

IBM WebSphere Adapters
Version 7 Release 5 Fix Pack 1 (7.5.0.1)

*IBM WebSphere Adapter for IBM i User
Guide
Version 7 Release 5 Fix Pack 1
(7.5.0.1)*



IBM WebSphere Adapters
Version 7 Release 5 Fix Pack 1 (7.5.0.1)

*IBM WebSphere Adapter for IBM i User
Guide
Version 7 Release 5 Fix Pack 1
(7.5.0.1)*



Note

Before using this information and the product it supports, read the information in "Notices" on page 159.

November 2011

This edition applies to version 7, Release 5, Fix Pack 1 (7.5.0.1) 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 IBM Corporation 2006, 2011.

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	1
Hardware and software requirements	2
Technical overview of WebSphere Adapter for IBM i	3
Outbound processing	3
Inbound processing.	11
WebSphere Application Server environment variables	17
Business object structures.	18
The external service wizard	20
Log and Trace Analyzer	20
Business faults	21

Chapter 2. Planning for adapter implementation 23

Security	23
Support for protecting sensitive user data in log and trace files	23
User authentication.	24
Deployment options	24
WebSphere Adapters in clustered environments	27
Migrating to version 7.5 of WebSphere Adapter for IBM i	28
Migration considerations	28
Performing the migration.	30
Updating but not migrating a project.	32

Chapter 3. Samples and tutorials 33

Chapter 4. Configuring the module for deployment. 35

Road map for configuring the module	35
Creating an authentication alias	36
Creating the module	37
Creating the project.	38
Defining WebSphere Application Server environment variables.	39
Selecting the processing direction	41
Setting connection properties for the external wizard	42
Setting properties to connect to the IBM i system	42
Setting properties to connect to a local file system	44
Configuring the module for outbound processing	47
Discovering objects.	47
Accessing messages using simple or keyed IBM i data queue	48
Calling IBM i programs	58
Setting global properties for operations	71
Configuring the module for inbound processing	71
Pre requisites for inbound processing.	72
Discovering objects.	72
Accessing messages using simple or keyed IBM i data queue	74

Chapter 5. Changing interaction specification properties 89

Chapter 6. Deploying the module 91

Deployment environments	91
Deploying the module for testing	91
Generating and wiring a target component for testing inbound processing	91
Adding the module to the server	92
Testing the module for outbound processing using the test client.	93
Deploying the module for production	93
Installing the RAR file (for modules using stand-alone adapters only)	94
Exporting the module as an EAR file.	95
Installing the EAR file.	96

Chapter 7. Administering the adapter module 99

Changing configuration properties for embedded adapters	99
Setting resource adapter properties for embedded adapters	99
Setting managed (J2C) connection factory properties for embedded adapters	101
Setting activation specification properties for embedded adapters	103
Changing configuration properties for stand-alone adapters	105
Setting resource adapter properties for stand-alone adapters	105
Setting managed (J2C) connection factory properties for stand-alone adapters	106
Setting activation specification properties for stand-alone adapters	107
Starting the application that uses the adapter.	108
Stopping the application that uses the adapter	109
Monitoring performance using Performance Monitoring Infrastructure	110
Configuring Performance Monitoring Infrastructure	110
Viewing performance statistics.	112
Enabling tracing with the Common Event Infrastructure	113

Chapter 8. Troubleshooting and support 115

Configuring logging and tracing	115
Configuring logging properties	115
Changing the log and trace file names	117
First-failure data capture (FFDC) support	117
Avoiding stale connection problems in the IBM i adapter	118

Known error conditions for the Data queues and
 Call IBM i programs 119
 Self-help resources. 119

Chapter 9. Reference 121

Business object information. 121
 Business object attributes for operations . . . 121
 Supported operations. 123
 Naming conventions 124
 Fault business objects. 124
 Configuration properties 125
 Guide to information about properties . . . 125
 Outbound configuration properties 126
 Inbound configuration properties. 135
 Globalization 154

Globalization and bidirectional transformation 154
 Bidirectional transformation in business objects 155
 Properties enabled for bidirectional data
 transformation 156
 Adapter messages. 157
 Related information 157
 Video samples 158

Notices 159

Programming interface information 161
 Trademarks and service marks 161

Index 163

Chapter 1. Overview of WebSphere Adapter for IBM i

The WebSphere® Adapter for IBM® i creates a service that uses the program call markup language (PCML) standard to call a Report Program Generator (RPG) or COBOL program or service program (SRVPGM) on the target IBM i system without special coding. You can also send or receive messages to a Data queue using WebSphere Adapter for IBM i.

The adapter enables two-way communication between a module running on a supported application server or integration broker and an IBM i system. You can call a program (RPG or COBOL) or service program (SRVPGM) using PCML source file by providing the parameters for the program on an IBM i system. A service program is an Integrated Language Environment® (ILE) object that provides a means of packaging externally supported callable routines (functions or procedures) into a separate object. Bound programs and other service programs can access these routines by resolving their imports to the exports provided by a service program. The connections to these services are made when the calling IBM i programs are created. The service program is generated using a combination of language programs.

Note: With the prerequisite system set up, and a valid PCML generated for an SRVPGM object type, you can call the service program.

With WebSphere Adapter for IBM i, a module can also send requests to get data from a Data queue or put data on a Data queue without the need of special code. 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. 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 contains individual pieces of data or 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. On 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 call a service on an IBM i system. 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.

With the WebSphere Adapter for IBM i software, you can automatically generate an interface to the IBM i server to hide the lower-level details of the function. Based on how you want to use the adapter, you can embed it with the deployed module, or install the adapter as a stand-alone component, to be used by more than one application. The adapter is deployed to WebSphere Enterprise Service Bus or IBM Business Process Manager. The application component interacts with the adapter instead of with the IBM i server.

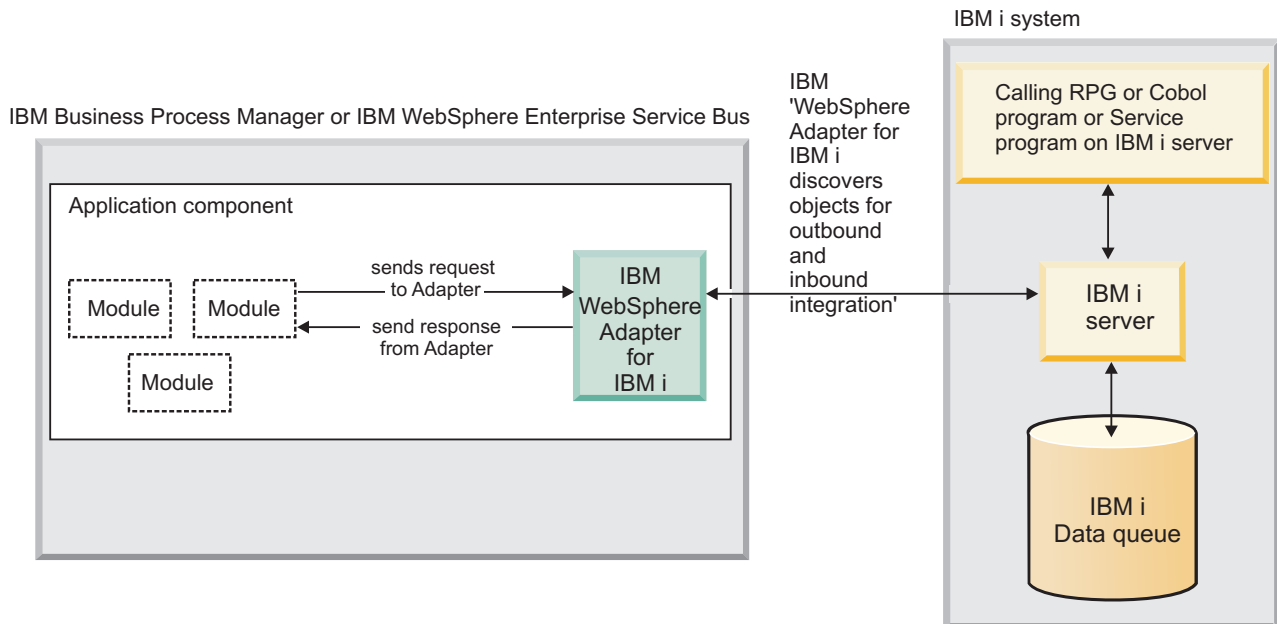


Figure 1. An application component calls the IBM i adapter, and the adapter interacts with the IBM i server to obtain the data

The adapter, which you generate with the external service wizard of Integration Designer, uses a standard interface and standard data objects. The adapter takes the standard data object sent by the application component and calls the IBM i server. The adapter then returns a standard data object to the application component. The application component does not have to deal directly with the IBM i server; it is the WebSphere Adapter for IBM i that calls the function and returns the results.

Hardware and software requirements

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

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: http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family.
- Technotes for WebSphere Adapters provide workaround 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 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 a data queue or calling a program such as Report Program Generator (RPG), COBOL or service program (SRVPGM) 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.

JTOpen API support: WebSphere Adapter for IBM i uses JTOpen version 6.6 APIs to connect to IBM i system. For more information about JTOpen, refer to <http://jt400.sourceforge.net>

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 a data queue or calling an IBM i program, such as Report Program Generator (RPG), COBOL, or service program (SRVPGM) 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 in the form of 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 process, such as messages on a data queue, or calls to an IBM i program, such as Report Program Generator (RPG) or COBOL or service program (SRVPGM) on an IBM i server.

Figure 2 on page 4 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.

Each outbound module contains components that form a service, including an import. 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 gets transported from the module. The references are bound to the physical location of the resource in the target operational environment at deployment. The IBM Integration Designer provides an interface that you use to set up the import.

Interfaces and references are used by the component and import 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 operations, which are GetQueue, PutQueue, or CallPGM for outbound requests. References declare what interface your service component calls. Each component in your module has one or more references. When you write your implementation for your service component using the IBM Integration Designer, you call a reference instead of directly calling the component itself. There by allows you to reassemble components in the future because you have not built dependencies into the code.

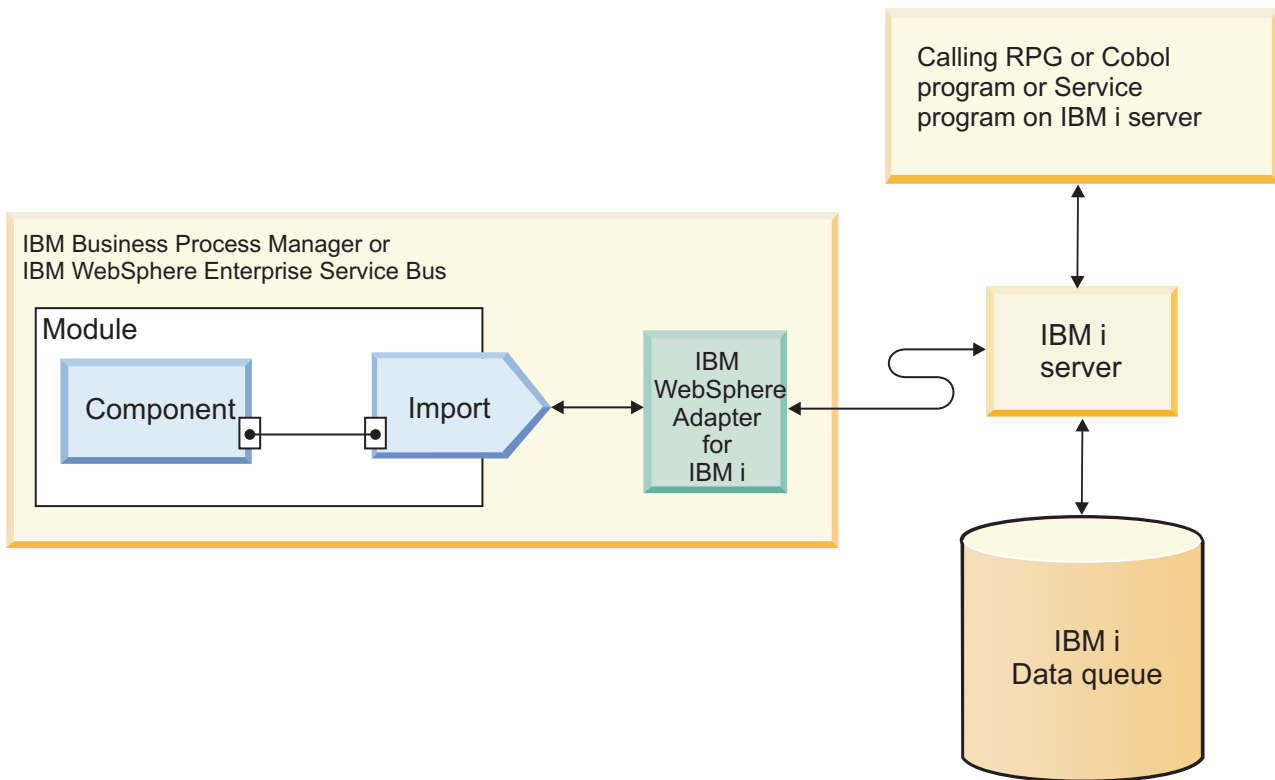


Figure 2. IBM i adapter as a part of an Outbound SOA implementation

Supported object types

The WebSphere Adapter for IBM i enable two-way communication between a module running on a supported application server or integration broker and an IBM i system.

WebSphere Adapter for IBM i supports the following object types:

- Data queues
- Call IBM i programs

Data queues: Data queues enables a two-way communication. Data queues are like a message queue. You can send messages from one program and another program can receive them from the message queue.

Call IBM i programs: Call IBM i programs or Program call is a call to an IBM i program or a service program.

Note: For more information about the operations supported by the WebSphere Adapter for IBM i, see the topics “Supported operations for data queues” on page 6 and “Supported operations for IBM i programs” on page 7 in this guide.

Data queues:

A Data queue is a system object that exists on IBM i system. Data queues are a cross-reference between data areas and message queues. These data queues help you to communicate asynchronously between programs, also known as control language programs.

A typical use for a data queue would be for a job in a batch system that is waiting for the data queue entry to be created while multiple other programs are requesting for entries to be created in the data queue. Each data queue object requires the integrated file system path name of the data queue. The system-recognized identifier is *DTAQ.

Characteristics of data queues: 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.

Advantages of data queues: The data queues:

- Provide a fast means of process communication, requiring minimal set up.
- Are designed to provide a flexible, highly efficient, yet temporary means of process communication.
- Interfaces require no communications programming and can be used either for connected or disconnected communication.
- Messages are merely described at the record-level, allowing you to define the field-level structure as required.

By default, when one program reads an entry in the queue, the entry is then deleted. Pointers to the queue entries are then updated to reflect the change in the record stack. A data queue can exist with no entries, a single entry, or multiple entries. Multiple concurrent jobs and programs can access data queues.

Benefits of using data queues: Data queues provide many benefits including:

- Contents of a data queue message are free-format (fields are not required), providing flexibility that is not provided by other system objects.
- Accessing data queues through control language (CL) command, which provides a straight forward means of developing client server applications. A control language (CL) command is a single statement to request a system function.

Ordering data queue messages: There are three ways to designate the order of messages on a data queue or message queue:

1. Sequential Last-in first-out (LIFO) - The last message (newest) placed on the data queue is the first message taken off the queue.
2. Sequential First-in first-out (FIFO) - The first message (oldest) placed on the data queue is the first message taken off the queue.
3. Keyed - Each message on the data queue has a key associated with it. A message can be taken off the queue only by requesting the key with which it is associated.

Call IBM i programs:

Call IBM i programs are stand alone programs that can take some parameters through program and run commands within the program and return results, if any.

Call IBM i programs or Program call is used to call any program (PGM) or service program (SRVPGM). Currently, the supported programs are RPG, COBOL, and SRVPGM. In some scenarios, the PGM can call other PGM or database script or application that are located in different library on the same IBM i machine.

Supported operations for data queues

Components use operations to perform actions such as putting a message on a data queue and getting a message from a data queue.

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 IBM i Data queue type of business object

Business objects supported	Operation
Business object corresponding to a simple data queue such as FIFO or LIFO or keyed data queue	GetQueue
	PutQueue

Note: For more details regarding the business object attributes for data queues, refer to section “Business object attributes for operations” on page 121.

GetQueue operation:

The GetQueue operation retrieves a message from either first-in first-out (FIFO) or last-in first-out (LIFO) data queue or a keyed 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 data queue is classified as a FIFO data queue, the adapter fetches the first message in the queue. If the data queue is classified as a LIFO queue, the adapter fetches the last message in 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

Supported operations for IBM i programs

Components use operations to perform actions such as calling an RPG or COBOL program (PGM) or service program (SRVPGM) on an 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 2. Outbound operations supported by IBM i program type of business object

Operation	Business objects supported
CallPGM	Business object corresponding to the IBM i program parameters

Note: For more details regarding the business object attributes for PGM operation, refer to section “Business object attributes for operations” on page 121.

Call IBM i Programs from the adapter:

Call IBM i program is a call to an IBM i program (RPG or COBOL) or service program (SRVPGM).

WebSphere Adapter for IBM i uses the PCML standard to help customers directly call an IBM i program, such as Report Program Generator (RPG) or COBOL or Service Program (SRVPGM) on the target IBM i system without any special coding. This program call can be achieved in three easy steps as described below.

1. Compile your program source (RPG or COBOL) or service program (SRVPGM) with the appropriate command and generate the PCML document.
2. Use the WebSphere Adapter for IBM i enterprise service discovery (ESD) wizard to discover the program through a generated PCML. The enterprise metadata discovery (EMD) auto generates all required artifacts.
3. Use the generated artifacts to run the underlying program or service program.

Report program generator (RPG) program:

RPG programs accept and return records in a fixed format.

The length of each parameter is important in sending a request and processing results. To promote consistency across the integrated language environment (ILE) languages, ILE RPG has enabled support for data types such as integers, floating point, variable length fields, parameters passed by value, return values, and so on.

Note: For more details regarding the business object attributes for PGM operation, refer to the section “Business object attributes for operations” on page 121.

COBOL program:

COBOL programs accept and return records in a fixed format.

WebSphere Adapter for IBM i leverages the PCML standard to help users directly call COBOL programs on the target IBM i system without any special coding. Currently, only simple data types are supported as parameters. The length of each parameter is important in sending a request and processing results. To promote consistency across the Integrated Language Environment (ILE) languages, ILE has enabled support for integers, floating point, parameters passed by value, return values, and so on.

Note: For more details regarding the business object attributes for PGM operation, refer to the section “Business object attributes for operations” on page 121.

Service program (SRVPGM):

A service program (SRVPGM) is an Integrated Language Environment (ILE) object that provides a means of packaging externally supported callable routines (functions or procedures) into a separate object.

Bound programs and other service programs can access these routines by resolving their imports to the exports provided by a service program. The connections to these services are made when the calling programs are created. This improves call performance to these routines without including the code in the calling program.

Note: For more details regarding the business object attributes for PGM operation, refer to the section “Business object attributes for operations” on page 121.

Service program conditions

The SRVPGM conditions are:

- The service program must be on an i5/OS® system running V5R4 or later.
- Up to seven parameters can be passed to the service program.
- By default, the adapter will create a Application-specific information (ASI) `passByType` with a `passByValue` as value for all the parameters associated with the service program. The supported values are `passByValue` and `passByReference`. You may need to set these values, corresponding to your service program design.
- The adapter creates a property, `procedureName`, on the interaction specification for the entry point found in the program call markup language (PCML) document. The value of this property is case sensitive. If the `CallPGM` operation fails with the message `CPF226E - Value for a parameter was not valid`, then the `procedureName` that is set in the `InteractionSpec` may be incorrect.
- The service program entry point name is converted from a Java String to an array of EBCDIC bytes before being sent to the system. By default, the adapter creates a property called Coded Character Set ID (CCSID) on `InteractionSpec` with `37` as value. You can overwrite this value based on your host environment.
- The service program is generated using a combination of multiple modules. With the prerequisite system set up, and a valid PCML being generated for an SRVPGM object type, you can call the service program.

Service program constraints

The programs or service programs created using the new features available to the V5R4 compilers can be executed only on a V5R4 OS/400® operating system. Though the compilers generate back-level objects, these objects are limited to the OS/400 and IBM i operating systems.

Pre IBM i and post IBM i commands:

Pre IBM i and post IBM i commands or library list is a system object that serves as a directory to other objects. IBM i adapter uses the PCML standard to help customers directly call program (RPG or COBOL) or service program (SRVPGM) on the target IBM i machine without any special coding.

A call to run an IBM i program such as RPG, COBOL, or SRVPGM program requires reference to other dependent IBM i programs such as RPG, COBOL, or SRVPGM program and database files. These files could be present in the same or different libraries (folders) on same IBM i machine. In order to support referring files and objects located in different locations, you can configure the pre IBM i commands and post IBM i commands as per your requirement. Pre IBM i commands are commands that you run before the program execution. Post IBM i commands are commands that you run after the program execution. You can choose to add any number of pre IBM i commands and post IBM i commands.

Configuring the library list

To configure the library list, you need to understand the structure of the data in the database and know what database objects the module needs to access. In addition, you need to know the following information:

- Database files and the dependent IBM i programs such as RPG or COBOL or SRVPGM
- Location of the files on the enterprise information system (EIS)
- Whether the required files are present in same or different libraries (folders)

Add Library List Entry and Remove Library List Entry command

In order to support referring files and objects located in different locations, you can add the location of the library to the current library list before executing the PRG program using the Add Library List Entry ADDLIBLE command. After executing the program, you need to remove these newly added entries from the current library list using the Remove Library List Entry RMVLIBLE command. For a detailed explanation of ADDLIBLE, see documentation at <http://publib.boulder.ibm.com/series/v5r1/ic2924/index.htm?info/cl/addlible.htm>

Library list example

The following example provides more information on using the ADDLIBLE command to configure the pre commands and post commands for RPG execution.

Example: Consider the RPG program name ABC.RPG and the PCML for this program is ABC.PCML. After executing the EMD wizard, the business object file for this PGM is generated as ABC.XSD. Also, considering ABC.PGM as call to object which is located in another library LIBINFO and considering there is a dependency on this file. This command adds the library LIBINFO to the end of the user portion of the library list. You can edit this file and update the changes in it. For more details, see the annotation provided in this example. The ADDLIBLE command is used to add libraries to the current library list. You can choose to add any number of pre commands and post commands. The attribute stoponerror can identify when a command fails and throws `ErrorCompletingRequestException`.

```
<appinfo source="http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi/metadata">
<ibmiasi:IBMiObjectLevelMetadata
xmlns:ibmiasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi/metadata">
<ibmiasi:precommands>
  <command>
    <ibmiasi:commandstring>ADDLIBLE LIBINFO *FIRST</ibmiasi:commandstring>
    <ibmiasi:stoponerror>>false</ibmiasi:stoponerror>
  </command>
</command>
  <ibmiasi:commandstring>ADDLIBLE LIBINFO *FIRST</ibmiasi:commandstring>
  <ibmiasi:stoponerror>>true</ibmiasi:stoponerror>
</command>
</ibmiasi:precommands>
<ibmiasi:postcommands>
  <command>
    <ibmiasi:commandstring>RMVLIBLE CREDITINFO</ibmiasi:commandstring>
    <ibmiasi:stoponerror>>false</ibmiasi:stoponerror>
  </command>
</ibmiasi:postcommands>
</ibmiasi:IBMiObjectLevelMetadata
</appinfo>
```

CallPGM operation:

The CallPGM operation invokes an RPG program or a COBOL program and a service program (SRVPGM) on an IBM i machine.

Programs on IBM i system are stand-alone programs that can take some input through program parameters and run commands within the program and return results, if any. CallPGM operation is used to call any program (PGM) or service program (SRVPGM) (currently supported program listed is RPG and COBOL). In some scenarios, the program can call other program or database script or

application that are located in a different library on the same IBM i system. The target library path needs to be added to the current library list for the smooth execution of the program.

Before CallPGM operation is invoked, the adapter will read the configured IBM i pre commands from the business object and will run the pre IBM i commands on the IBM i system. The decision to continue further execution or not is based on Stop On Error values. If command execution fails and Stop On Error is configured to be true, then adapter will generate an `ErrorCompletingRequestFault`.

The CallPGM operation is invoked and the adapter does the following:

1. Builds the parameter list for the program from the attribute values received in the request business object.
2. Establishes a connection with an IBM i system.
3. Searches the target application in the libraries listed in the current library list if there is a call referring to other application or to a database. The system-recognized identifier is *LIBL.
4. Runs the program on an IBM i system.

If CallPGM execution is successful then adapter will read the configured post IBM i commands from the business object and will run the post IBM i commands on the IBM i system. The decision to continue further execution or not is based on Stop On Error values. If command execution fails and Stop On Error is configured to be true, then adapter will generate an `ErrorCompletingRequestFault`.

Program do not return values directly. Instead, they take in all the input arguments provided by the adapter, complete the required processing, and write the result value into variables that 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.

Note: For more details regarding the business object attributes for program operation, refer to section "Business object attributes for operations" on page 121.

Inbound processing

Inbound processing refers to the actions taken by the adapter when polling an IBM i Data queue 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 a notification when an object is put on an IBM i Data queue. Figure 3 on page 12 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 IBM Business Process Manager or WebSphere Enterprise Service Bus or administrative console.

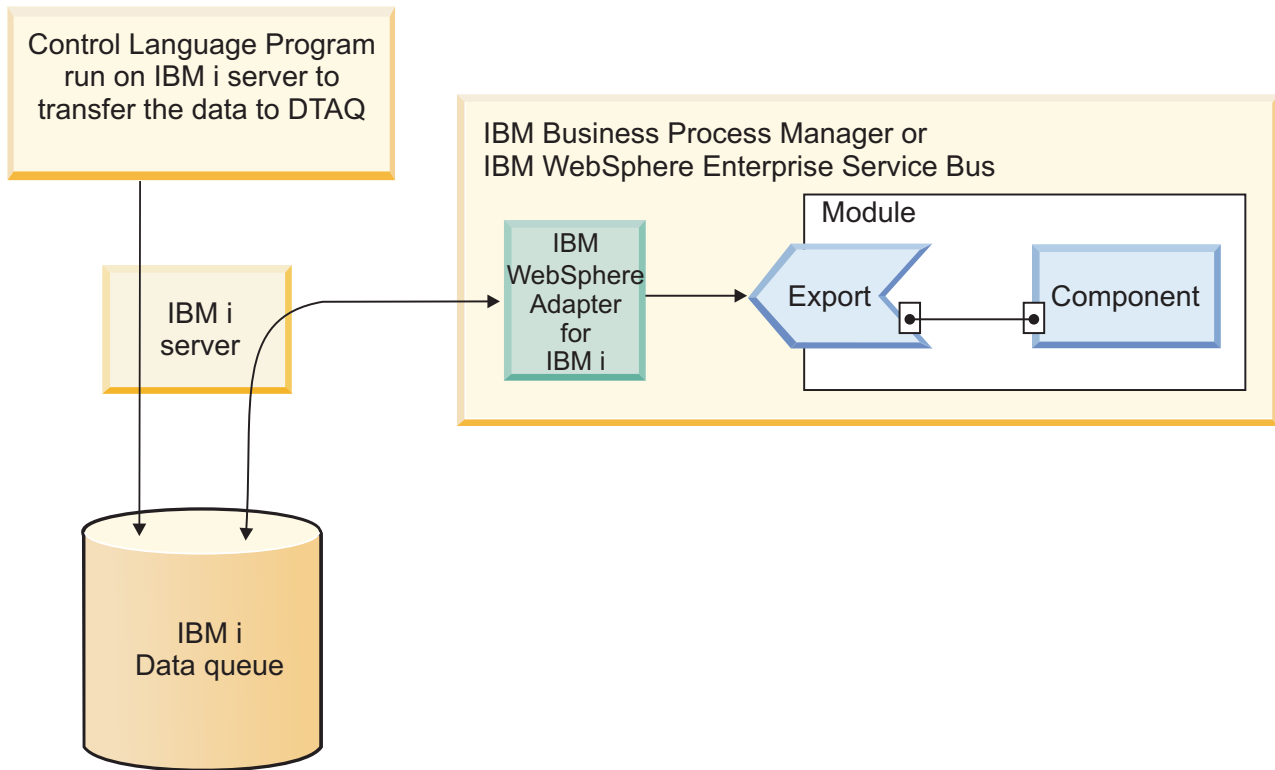


Figure 3. Overview of inbound processing

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 first-in first-out (FIFO) queue, the adapter fetches the oldest message. From a LIFO queue, the 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 the way the adapter processes events when you use the external service wizard to configure the adapter or at a later time by using the administrative console of the server to change the activation specification properties. For example, you can specify how often the adapter polls for events and how many events it retrieves during each polling period.

Before you can create data queues, you must copy the control language (CL) program located in the adapter installation directory to the IBM i system. The CL program is run on IBM i server to transfer the data to DTAQ. The adapter provides sample scripts CL programs I5INBOUND.SAVF and I5ARCHFAIL.SAVF for creating data queues. The CL programs contains the source code CL (.CLLP files) along with the compiled code program (.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 IBM Integration Designer, and `version` is the version of the adapter, for example, 7.5.

Assured-once delivery of events

The adapter provides a recovery feature called 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. Before the event store can be created, the data source must be created in IBM Business Process Manager. To use the recovery feature provided by IBM Business Process Manager, 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.

Supported object types

The WebSphere Adapter for IBM i enables a two-way communication between a module running on a supported application server or an integration broker and an IBM i system.

WebSphere Adapter for IBM i supports the Data queues object type.

Data queues: Data queues are similar to a message queue. You can send messages from one program and another can receive them from the message queue.

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

Data queues:

A Data queue is a system object that exists on IBM i system. Data queues are a cross-reference between data areas and message queues. These data queues help you to communicate asynchronously between programs, also known as control language programs.

A typical use for a data queue would be for a job in a batch system that is waiting for the data queue entry to be created while multiple other programs are requesting for entries to be created in the data queue. Each data queue object requires the integrated file system path name of the data queue. The system-recognized identifier is *DTAQ.

Characteristics of data queues: 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.

Advantages of data queues: The data queues:

- Provide a fast means of process communication, requiring minimal set up.
- Are designed to provide a flexible, highly efficient, yet temporary means of process communication.
- Interfaces require no communications programming and can be used either for connected or disconnected communication.
- Messages are merely described at the record-level, allowing you to define the field-level structure as required.

By default, when one program reads an entry in the queue, the entry is then deleted. Pointers to the queue entries are then updated to reflect the change in the record stack. A data queue can exist with no entries, a single entry, or multiple entries. Multiple concurrent jobs and programs can access data queues.

Benefits of using data queues: Data queues provide many benefits including:

- Contents of a data queue message are free-format (fields are not required), providing flexibility that is not provided by other system objects.
- Accessing data queues through control language (CL) command, which provides a straight forward means of developing client server applications. A control language (CL) command is a single statement to request a system function.

Ordering data queue messages: There are three ways to designate the order of messages on a data queue or message queue:

1. Sequential Last-in first-out (LIFO) - The last message (newest) placed on the data queue is the first message taken off the queue.
2. Sequential First-in first-out (FIFO) - The first message (oldest) placed on the data queue is the first message taken off the queue.

3. Keyed - Each message on the data queue has a key associated with it. A message can be taken off the queue only by requesting the key with which it is associated.

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 3.

Table 3. 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 <i>AbsolutePathOfQueue+TimeStamp</i> .
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: <ul style="list-style-type: none"> • 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 3. 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 4. 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: <ul style="list-style-type: none"> • 0 (Created) • 1 (Executed) • 3 (In Progress) • -1 (Failed)

Table 4. 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.

WebSphere Application Server environment variables

WebSphere Application Server environment variables can be used in the external service wizard to specify directory values. You can now change any string property values in inbound and outbound configuration by only changing the environment variables.

When you configure the adapter for inbound or outbound processing using the external service wizard, you set values for various required local files and directories. You can later change these values in the deployed application from the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

Starting with IBM Business Process Manager or WebSphere Enterprise Service Bus Version 6.1.0, instead of hard coding values for Queues and Program paths, you can declare them as WebSphere Application Server environment variables and specify the environment variable names when you run the external service wizard. When you deploy your application, the environment variable name is replaced with the actual value and used by the adapter. If you want to change the property value, you can just change the environment variable in the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

WebSphere Application Server environment variables can be used for all string property values (not Boolean or integer variables) that are set in inbound and outbound configurations.

When you create a WebSphere Application Server environment variable, you specify:

- The name of the environment variable, for example, PAYLOAD STAGING QUEUE, POLLING CTRL LANG PGM.
- The value that the symbolic name represents, for example: /QSYS.LIB/RAJAN.LIB/PAYLOAD.DTAQ, /QSYS.LIB/RAJAN.LIB/MSGQCL2.PGM.
- The scope for the environment variable, which determines the level at which the environment variable is visible in the administrative console. The scope level can be server, node, or cell:
 - Server scope limits visibility to the named server. The server scope is the most specific scope for defining environment variables.
 - Node scope limits visibility to all the servers on the named node. This is the default scope.
 - Cell scope limits visibility to all servers on the named cell.

Business object structures

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 IBM i programs such as RPG, COBOL or Service programs running on the IBM i system.

How the adapter uses business objects

Your component communicates with the adapter using business objects. The module uses business objects to access the IBM i data queues or to call IBM programs such as RPG, COBOL, or Service program 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 shows a business object 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.

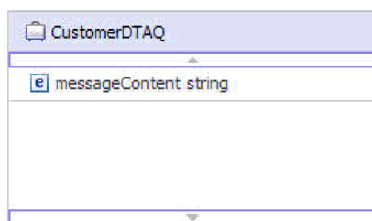


Figure 4. Business object for putting a message from a customer on a data queue

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

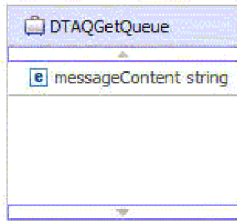


Figure 5. Business object for getting a message from a data queue

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.

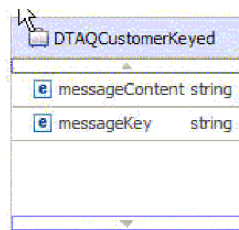


Figure 6. Business object for removing a customer message from a keyed data queue

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 IBM i program business objects

Figure 7 on page 20 shows a business object used to call an IBM i program such as RPG program or COBOL program or Service program on an IBM i system in the business object editor. To properly invoke an IBM i program on an IBM i system, the business object must contain the 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.

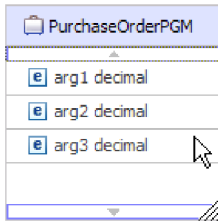


Figure 7. PurchaseOrderPGM business object

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 IBM i program business object you need to account for application-specific information such as the required fields for invoking an IBM i 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.

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, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter. For example, when 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to 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. It prevents 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, see the “Adapter ID (AdapterID)” on page 129 property.

Business faults

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Although IBM Business Process Manager and WebSphere Enterprise Service Bus support other types of faults, the WebSphere adapter generates only business faults. These business faults will be referred to as faults in this document. Not all exceptions are faults. Faults are generated for errors that have a recovery action and do not terminate the application. For example, during outbound processing, the adapter generates a fault when it receives a business object that does not contain the required data or when the adapter encounters certain errors.

Fault business objects

The external service wizard creates a business object for each fault that the adapter can generate. In addition, the wizard creates a WBIFault superset business object, which has information common to all faults, such as the message, errorCode, and primaryKeySet attributes, as shown in figure 1.

WBIFault	
message	string
errorCode	string
primaryKeySet	PrimaryKeyPairType []

Figure 8. The structure of the WBIFault business object

The WBIFault business object contains all the information needed to handle the fault.

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 the 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

Support for protecting sensitive user data in log and trace files

You can configure the adapter to prevent sensitive or confidential data, in the log and trace files, from being viewed by users 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 might 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 such situations, the adapter lets you specify whether 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 131 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. You can configure the adapter to get your user information, through 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 this 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 access that IBM i system. This includes 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 must be known only to that administrator, who can change the password in a single place, when a change is required.

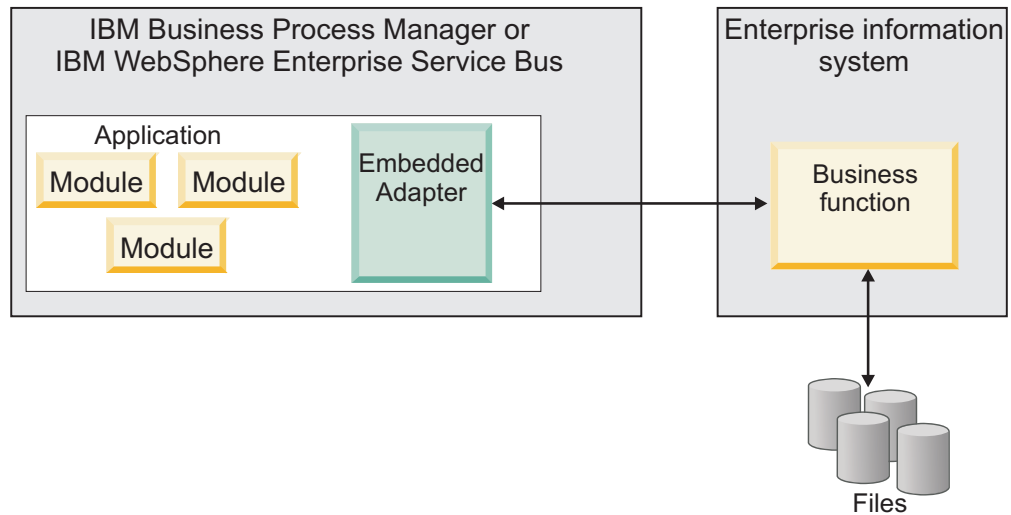
Deployment options

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 affect the type of deployment option you choose.

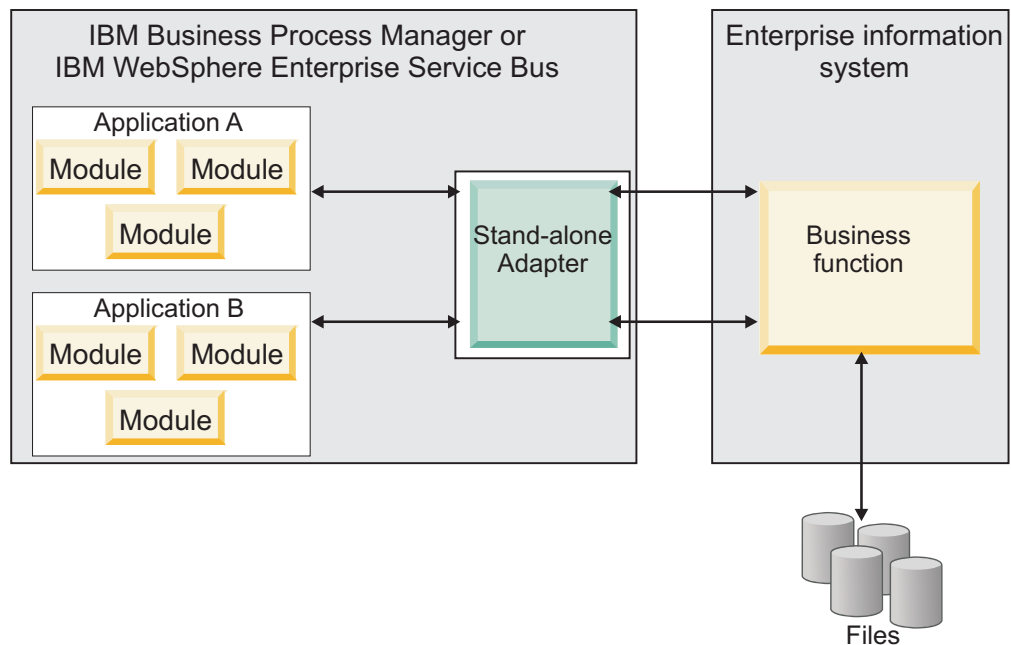
The following are the deployment options:

- **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.



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 IBM Integration Designer, you can choose how to package the adapter [either bundled with the (EAR) file or as a stand-alone (RAR) file]. Your choice affects how the adapter is used in the run time 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 to run 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

Consider the following items if you plan to embed 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 run time environments. *Class loader isolation* means that 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

Consider the following items if you plan to use 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.

Note:

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during runtime, the adapter will return the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for IBM i version 7.5, the following error message is displayed: IBM WebSphere Adapter for IBM i has loaded file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this

jar required is version 7.5. When you install multiple adapters with different CWYBS_AdapterFoundation.jar versions, the adapter returns the ResourceAdapterInternalException message, due to a version conflict. To overcome this conflict, you must migrate all adapters to the same version level. For further assistance, contact WebSphere Adapters Support for help.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying a module on a clustered server environment. Clusters are groups of servers that are managed together to balance workloads and to provide high availability and scalability.

The module you deployed is replicated across all servers in a cluster, regardless of whether you deploy the module using a stand-alone or an embedded adapter. The following IBM products support WebSphere Adapters in a clustered environment:

- IBM Business Process Manager or WebSphere Enterprise Service Bus
- WebSphere Application Server Network Deployment
- WebSphere Extended Deployment

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 an adapter instance. For information about creating clustered environments, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/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 instance instead of a static Workload Manager. The dynamic Workload Manager instance 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 systems with different capacities and configurations to handle load variations evenly. For information about the benefits of WebSphere Extended Deployment, see <http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1m1/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 Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (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.

Note: In the active-passive configuration mode of the adapters, the endpoint application of the passive adapter instance also listens to the events/messages even if the `enableHASupport` property is set to `True`. This is because the `alwaysactivateAllMDBs` property in the JMS activation specification is set to `True`. To stop the endpoint application of the passive adapter instance from listening to the events, you must set the `alwaysactivateAllMDBs` property value to `False`. For more information, see [Disabling end point applications of the passive adapter](#) .

Note: In clustered environments, when the adapter works in a Active-Active configuration, it provides both high availability and load balancing support. This functionality is useful in production environments where high performance is needed.

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 about workload management, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Migrating to version 7.5 of WebSphere Adapter for IBM i

By migrating to version 7.5 of WebSphere Adapter for IBM i, you automatically upgrade from the previous version of the adapter. Additionally, you can migrate your applications that embed an earlier version of the adapter, so that the applications can use features and capabilities present in version 7.5.

Migration considerations

WebSphere Adapter for IBM i version 7.5 may have some features and updates that might affect your existing adapter applications. Before migrating applications that use WebSphere Adapter for IBM i, you must consider some factors that might affect your existing applications.

Compatibility with earlier versions

WebSphere Adapter for IBM i version 7.5 is fully compatible with the custom business objects (XSD files) and data bindings that are created using the adapter version 7.0 and enables the existing business objects and data bindings to work well in the latest version of the adapter.

Run the migration wizard if you want your applications to use features and functionality present in version 7.5 of the adapter.

The migration wizard replaces (upgrades) version 7.0 of the adapter with version 7.5 and enables version 7.5 features and functionality for use with your applications.

Note: The migration wizard does not create components or modify existing components, such as mappers and mediators to work with version 7.5 of the adapters. If any of your applications embed an adapter that is version 7.0 or earlier and you are upgrading to version 7.5, and you want your applications to take advantage of the features and functions in version 7.5, you might need to change those applications.

If the artifacts within a module have inconsistent versions, the entire module is marked as unavailable for migration and cannot be selected for migration. Version inconsistencies are recorded in the workspace log, as this might be a symptom of project corruption.

If the IBM Integration Designer version is not equal to the WebSphere Adapter for IBM i version, they will fail to run together as depicted in the scenarios here.

The following scenario is not supported:

- Running the external service wizard in IBM Integration Designer version 7.5 with WebSphere Adapter for IBM i version 7.0.

When you migrate the business objects for the putqueue operation from IBM Integration Designer 7.0 to the current version, the following migration exception error message will appear in the problems pane of IBM Integration Designer 7.5 workspace:

"Response BO object not found. Add response BO to the project".

WSDL: The element `http://schemas.xmlsoap.org/wsdl/#fault` is not permitted as constrained by `"http://schemas.xmlsoap.org/wsdl/#fault"`.

This is because of a version mismatch between an earlier business object that has a putqueue operation, and IBM Integration Designer 7.5 version. You will encounter this error only with the putqueue operation and not with other operations, such as getqueue, RPG via PCML, COBOL via PCML, and so on. To resolve this error:

1. Right click the error message and select the **Quick Fix** menu option.
2. Select the fix for **Response BO object not found. Add Response BO to the project** in the Quick Fix screen.
3. Follow the remaining suggestions in this screen.

You can then migrate a business object from Integration Designer 7.0, to the Integration Designer 7.5 workspace.

Deciding whether to upgrade or to upgrade and migrate

The default processing of the migration wizard is to perform an upgrade of the adapter and to migrate the application artifacts so that the applications can use features and functions in version 7.5 of the adapter. When you choose to upgrade the adapter by selecting a project, the wizard automatically selects the associated artifacts for migration.

If you decide that you want to upgrade the adapter from version 7.0 to version 7.5, but you do not want to migrate the adapter artifacts, you can do so by deselecting the adapter artifacts from the appropriate area of the migration wizard.

Running the migration wizard without any adapter artifacts selected installs and upgrade your adapter, but your artifacts are not migrated and your applications cannot take advantage of the features and capabilities that exist in version 7.5 of the adapter.

Run the migration wizard in a test environment first

Because adapter migration might require you to change those applications that use version 7.5 of WebSphere Adapter for IBM i, you must always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

The migration wizard is fully integrated with the development environment.

Performing the migration

You can migrate a project or EAR file to version 7.5 using the adapter migration wizard. When the tool is finished, the migration is complete and you can work in the project or deploy the module.

Before you begin

Review the information in *Migration considerations*.

About this task

To perform the migration in IBM Integration Designer, complete the following steps.

Note: After migration is complete, the module is no longer compatible with previous versions of the IBM Business Process Manager or WebSphere Enterprise Service Bus runtimes or IBM Integration Designer.

The following steps describe how to run the adapter migration wizard from the connector project menu while in the Java EE perspective in IBM Integration Designer.

Procedure

1. Import the PI (project interchange) file for an existing project into the workspace.

Note: Ensure that you do not modify the contents of the RAR or copy the adapter JAR file outside the connector project.

2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrscart.html>.
3. Change to the Java EE perspective.
4. Right-click the module and select **Migrate connector project**. For example, the adapter RAR module.

You can also launch the adapter migration wizard in the following ways:

- Right-click the project in the Java EE perspective and select **Migrate adapter artifacts**.
 - From the Problems view, right-click a migration-specific message and select **Quick Fix** to correct the problem.
5. In the Select Projects window, perform the following steps:
 - a. The **Source connector** field displays the name of the connector project that you are migrating. If you are migrating a module project, this field lists all the connector projects in the module project. Select the source project from the list.
 - b. The **Target connector** field displays the name of the connector to which you are migrating. If you are working with more than one adapter version, this list displays the names of all the compatible connectors. Select the connector you want to migrate.
 - c. The **Target version** field displays the version corresponding to the target connector that you selected in the previous step.
 - d. The **Dependent artifacts project** area lists the adapter artifacts that are migrated. If you are migrating a module project, this area lists only the selected module project. If you are migrating a connector project within the module project, this area lists all projects which reference the selected connector project, including the module project. By default, all the dependent artifact projects are selected. If you do not select a dependent artifact project, that project is not migrated. You can migrate any project that you have not selected at a later time. Previously migrated projects, projects with a current version, and projects that contain errors are unavailable for migration and are not selected. For more information, see “Updating but not migrating a project” on page 32.
 - e. Click **Next**. A warning window is displayed with the message, “Properties that are not supported in this version of the target adapter will be removed during the migration”.
 - f. Click **OK**.
 6. In the Review Changes window, review the migration changes that occur in each of the artifacts that you are migrating. To view the details, expand each node by clicking the + sign.
 7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.

Before running the migration process, the wizard performs a backup of all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. If the migration fails for any reason, or if you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.

Upon completing the migration successfully, all backed up projects are deleted.

8. If you are migrating an EAR file, optionally create a new EAR file with the migrated adapter and artifacts, and deploy it to IBM Business Process Manager or WebSphere Enterprise Service Bus. For more information about exporting and deploying an EAR file, see the topics devoted to it in this documentation.

Results

The project or EAR file is migrated to version 7.5. You do not need to run the external service wizard after exiting the adapter migration wizard.

Updating but not migrating a project

You can upgrade the adapter from version 7.0 to version 7.5, while choosing not to migrate the adapter project artifacts.

About this task

Running the migration wizard without selecting any adapter artifacts installs and upgrades your adapter. As the artifacts are not migrated, your applications cannot take advantage of the features and capabilities that exist in version 7.5 of the adapter.

Procedure

1. Import the project interchange (PI) file into the workspace.
2. In the Java EE perspective, right-click the project name and click **Update Connector ProjectMigrate connector project**. The adapter migration wizard is displayed.
3. In the Select Projects window, clear the dependent artifact projects, and click **Next**. A warning window opens with the message, "The properties that are not supported in the version of the target adapter will be removed during the migration."
4. Click **OK** to continue to the next step.
5. In the Review changes window, review the migration changes that occur during updating the project. To view the details, expand each node by clicking the + sign.
6. Click **Finish** to perform the updating.

Results

The project can now be used with WebSphere Adapter for IBM i, version 7.5.

Chapter 3. Samples and tutorials

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

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

- From the welcome page of IBM Integration Designer, click **Go to Samples and Tutorials**. In the Samples and Tutorials pane, under More samples, click **Retrieve**. Browse the displayed categories to make your selection.
- From the Business Process Management Samples and Tutorials website:
<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 IBM Business Process Manager or WebSphere Enterprise Service Bus, use IBM Integration Designer 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 IBM Integration Designer. 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.

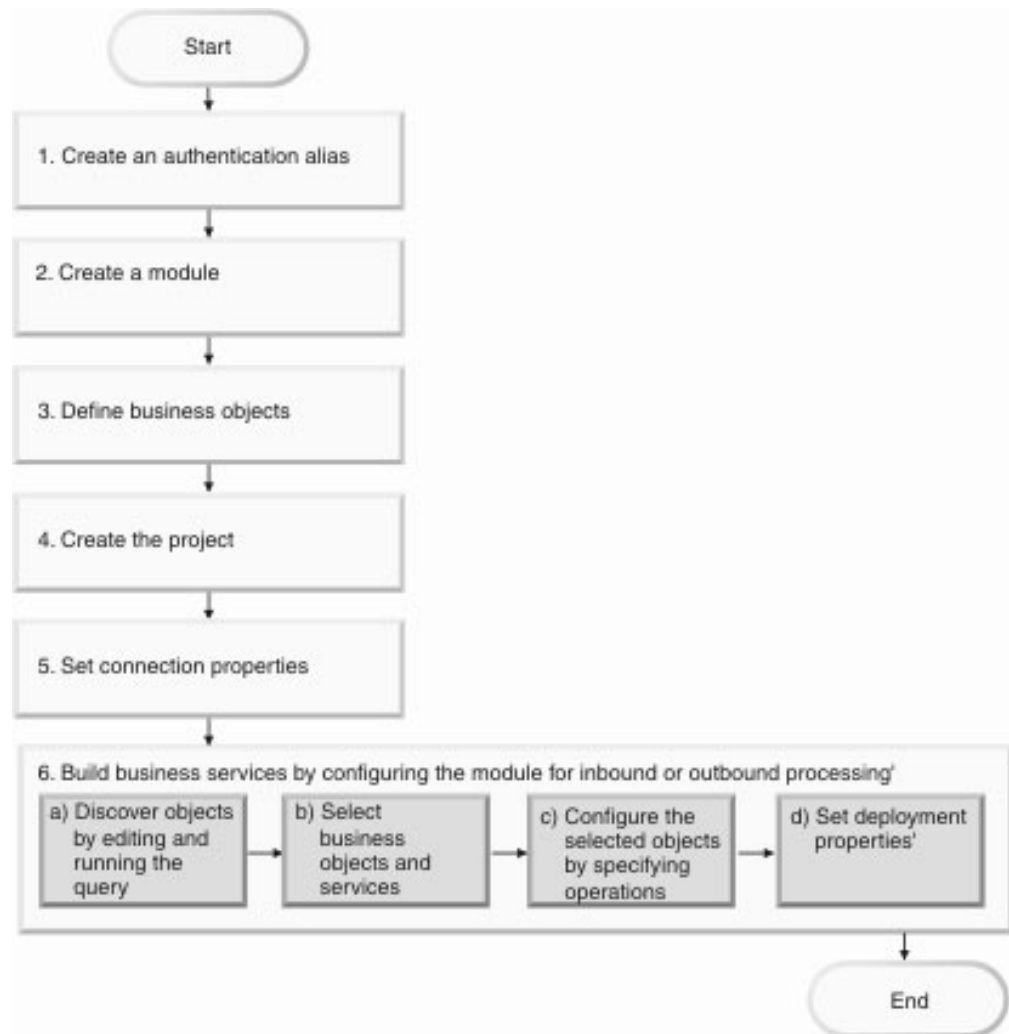


Figure 9. Road map for configuring the module

Configuring the module for deployment environment

This task consists of the following step:

1. Configure the IBM i system to work with your adapter. You perform this step from within the IBM i system.
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 IBM Integration Designer 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 processing 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.

Note: The required services on the IBM i server to create data queues must be running for inbound processing. Also, ensure that the required control language (CL) program files for inbound processing are copied in the correct directories and are running. The CL program is used to transfer data to DTAQ.

- b. Edit and run the query that discovers objects that you can access.
- c. Select and configure business objects for inbound processing or outbound processing.
- d. Configure the selected objects by specifying operations and other properties that apply to all the business objects.
- e. 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 file or export file, and the service interface.

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 IBM Business Process Manager 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 IBM Integration Designer. If you are using the administrative console directly (without going through IBM Integration Designer), log in to the administrative console and skip to step 2 on page 37.

About this task

Using an authentication alias eliminates the need for storing 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 IBM Integration Designer, perform the following steps:
 - a. In the Business Integration perspective of Integration Designer, click the **Servers** tab.
 - b. If the server does not show a status of **Started**, right-click the name of the server (for example, IBM Business Process Manager) 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 module

A module encapsulates the service in a reusable unit and consists of both a project in IBM Integration Designer and a unit of deployment to IBM Business Process Manager or WebSphere Enterprise Service Bus. The module is packaged and deployed to IBM Business Process Manager or WebSphere Enterprise Service Bus as an enterprise archive (EAR) file.

Procedure

1. If IBM Integration Designer is not currently running, start it now.
 - a. Click **Start > Programs > IBM > IBM Integration Designer > IBM Integration Designer 7.5**.
 - b. If you are prompted to specify a workspace, either accept the default value or select another workspace.

The workspace is a directory where IBM Integration Designer stores your project.
 - c. Optional: When the IBM Integration Designer window is displayed, click **Go to the Business Integration perspective**.
2. Right-click inside of the Business Integration section of the IBM Integration Designer window.

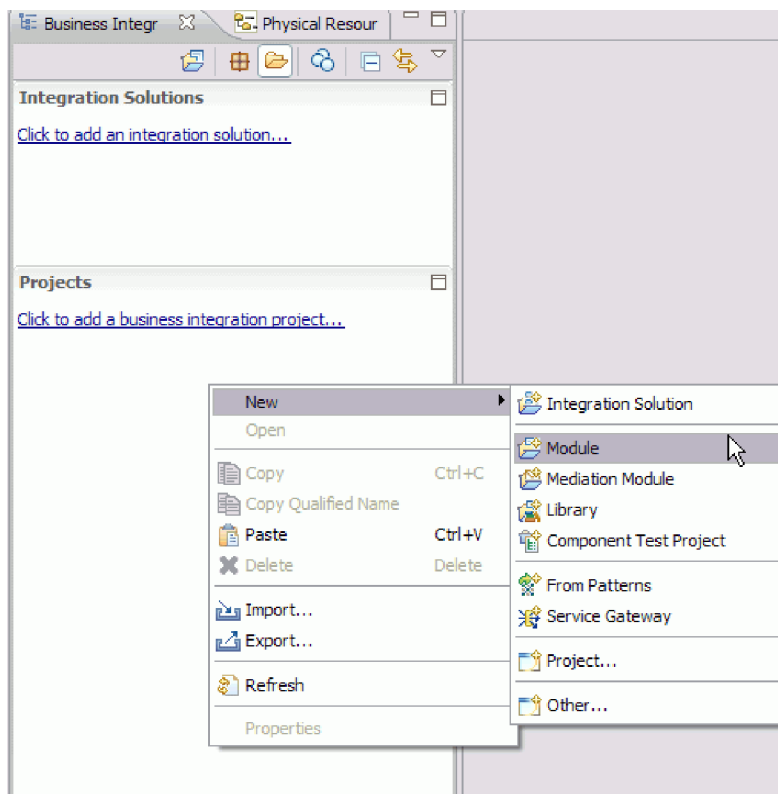


Figure 10. Business Integration section of the window

3. In the New Module window, type a name for the module in the **Module Name** field. For example, to configure a module for outbound processing, type **Outbound** as the module name. Click **Finish**.

Results

A new module is listed in the Business Integration window.

Creating the project

To begin the process of creating and deploying a module, you start the external service wizard in IBM Integration Designer. 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 IBM Integration Designer, and then click **File > New > External Service**.
2. In the New 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.
4. Click **Next**.
5. In the Import a RAR File 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 Locate 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.

Defining WebSphere Application Server environment variables

Use the administrative console of IBM Business Process Manager or WebSphere Enterprise Service Bus to define WebSphere Application Server environment variables.

About this task

To define a WebSphere Application Server environment variable, use the following procedure.

Procedure

1. Start the administrative console.
2. Select **Environment > WebSphere Variables**.
3. Select the scope for the environment variable. The scope specifies the level at which the resource definition is visible on the administrative console panel. The possible values are *server*, *node*, and *cell*. In the following figure, the scope is defined at the cell level as `Cell=Dmgr1Cell`.

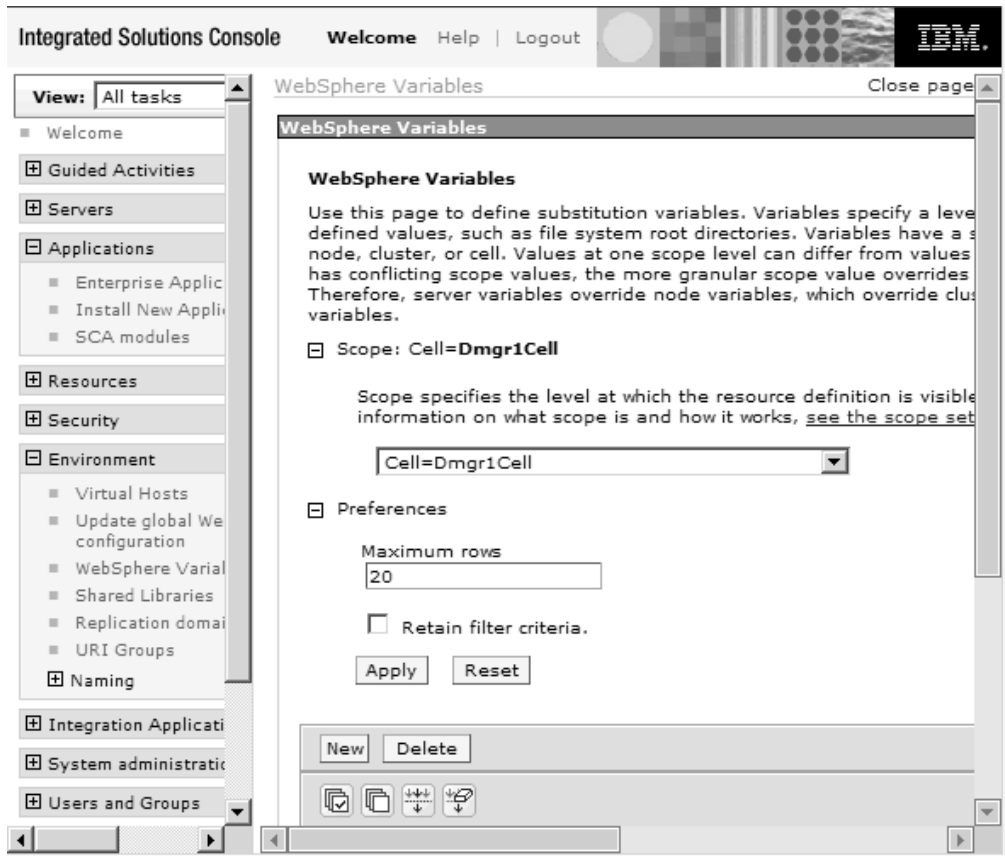


Figure 11. Setting the scope for the environment variable

4. Click **New** and provide a name and a value for the environment variable. The name is the symbolic name that represents a physical path. The value is the absolute path that the variable represents.
5. Click **OK** to save the changes.

Results

An environment variable with the specified name and a scope of Cell=Dmgr1Cell is defined. You can use it in the external service wizard whenever you need to specify the event directory.

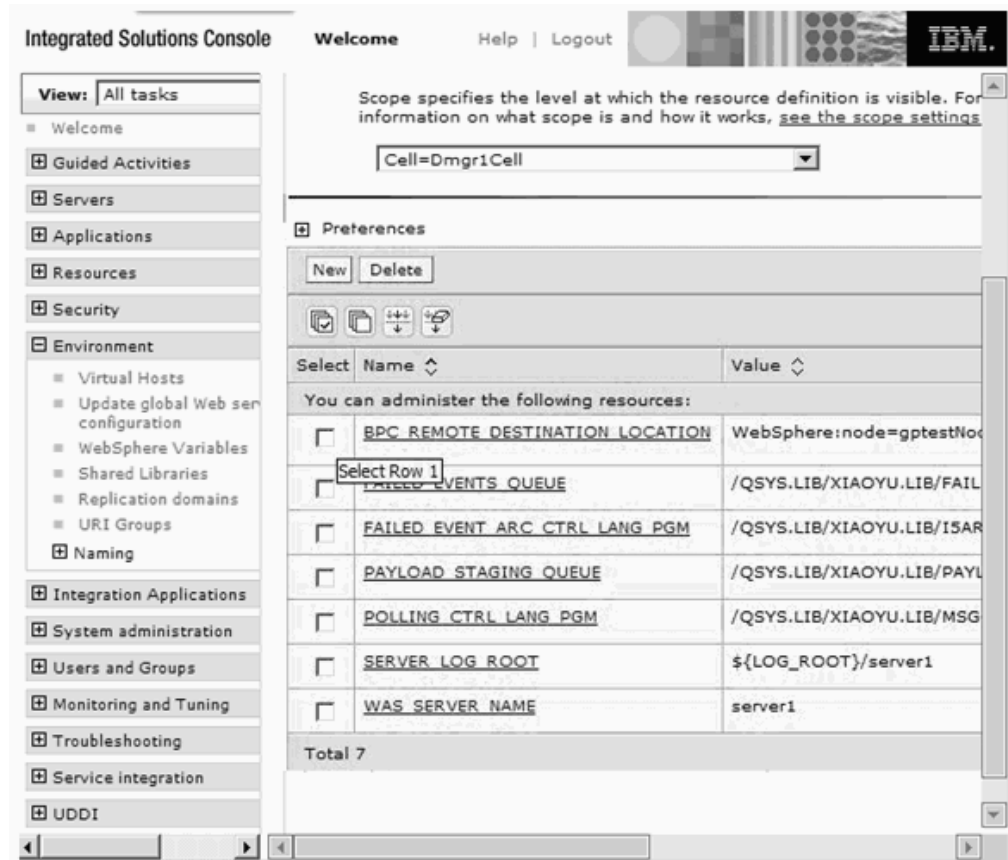


Figure 12. The new environment variable *PAYLOAD STAGING QUEUE* is displayed in the *WebSphere Variables* window

Selecting the processing direction

The adapter processing direction can be configured at run time environment. You can either configure your adapter for inbound processing or outbound processing.

Before you begin

Ensure that you are running the external service wizard in IBM Integration Designer to perform this task.

About this task

When you configure your adapter for inbound processing you are processing the request or response interaction from the adapter to your service export. When you configure your adapter for outbound processing you are processing the response from your service import to the adapter.

In the Select the Processing Direction window, select **Outbound** or **Inbound**, and then click **Next** to advance to the Connection properties window.

Results

You have defined the type of process you need for your adapter.

Setting connection properties for the external wizard

You must specify the connection properties to enable the external service wizard to discover business objects on the IBM i system.

About this task

You can set the connection properties for the external wizard in the following ways:

- Setting properties to connect to the IBM i system. For more information on this approach, refer “Setting properties to connect to the IBM i system”
- Setting properties to connect to a local file system. For more information on this approach, refer “Setting properties to connect to a local file system” on page 44

Results

The external service wizard connects to the IBM i server using the information you provided and the Find Objects in the Enterprise System window is displayed.

What to do next

Decide upon either of the approaches, to proceed with your discovery.

Setting properties to connect to the IBM i system

You must specify the connection properties to enable the external service wizard to discover business objects on the IBM i system.

About this task

The external service wizard requires these properties to connect to the IBM i system to discover and create a service description.

Procedure

1. Start the external service wizard and enter the appropriate information in the Select the Processing Direction window, as explained in the “Selecting the processing direction” on page 41 topic.
2. In the Specify the Discovery Properties 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. Select any one of the following options from the **Object type to discover** list. You have an option to select Data queue (default option), RPG via PCML, Cobol via PCML, and SRVPGM via PCML from the object type to discover list.
 - Data queue (default option)
 - RPG via PCML
 - COBOL via PCML
 - SRVPGM via PCML

- RPG Program
- COBOL Program

Note: Option to work directly with RPG Program and COBOL Program is not available during iterative discovery.

Note: For more details regarding the supported object types for IBM i adapter, refer to the following link: “Supported object types” on page 4.

- The **Path to library on IBM i for object discovery** field is populated with values based on the object type selected in the previous step. If you select Data queue, or RPG Program, or COBOL Program as the object type, the library path on the is populated with /QSYS.LIB/. If you select any of the Call IBM i programs, for example RPG via PCML, COBOL via PCML, SRVPGM via PCML, as one of the object types, the library path on the IBM i system is populated with /home/.
- 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 is TestFIFOQ.

The following figure shows the options to specify a discovery property, using the IBM i system.

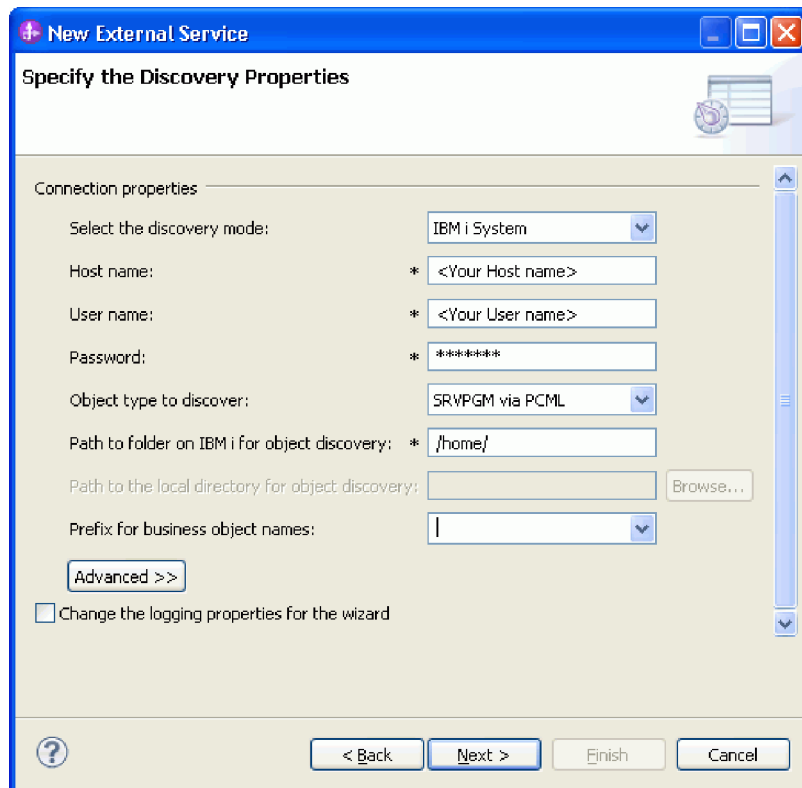


Figure 13. Specify the Discovery Properties window.

Connection information from this window is used by the adapter to connect to the IBM i system and discover business objects. In this same wizard at a later time, you can also change the connection information or an authentication alias to be used at run time.

- Optional: To enable bidirectional support for the adapter at run time:

- a. Click **Advanced**.
 - b. Select **Bidi transformation** from the **Bidi properties** list.
 - 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 logging 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.

The log information is only for the wizard. At run time, the adapter writes messages and trace information into the standard log and trace files for the server.
 5. After you provide the required connection information for the wizard, click **Next**. The **Path to folder at IBM i for object discovery** field is validated. If the path is invalid, a message to that effect appears. In addition, you can also check the logs, which are located in the directory specified in the **Log file output location**. Ensure that the connection information is correct.
 6. Click **Next** to proceed with your discovery.

Results

The external service wizard connects to the IBM i server using the information you provided and the Find Objects in the Enterprise System window is displayed.

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.

Setting properties to connect to a local file system

You must specify the connection properties to enable the external service wizard to discover business objects through your local system.

About this task

The external service wizard connects to a local file system, IBM i system to discover and create a service description.

Procedure

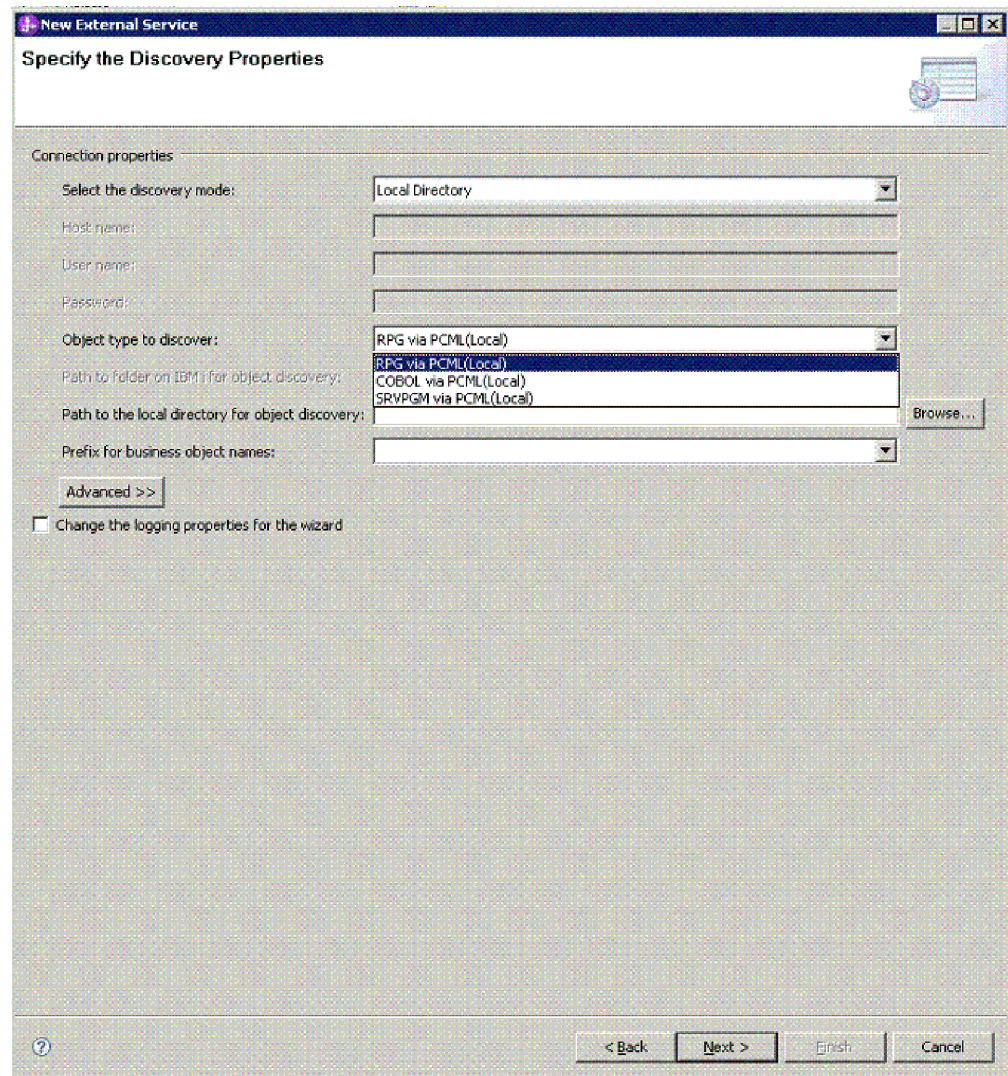
1. Start the external service wizard and enter the appropriate information in the Select the Processing Direction window, as explained in the “Selecting the processing direction” on page 41 topic.

Select any one of the following options from the Object type to discover list:

- RPG via PCML (Local)
- COBOL via PCML (Local)
- SRVPGM via PCML (Local)

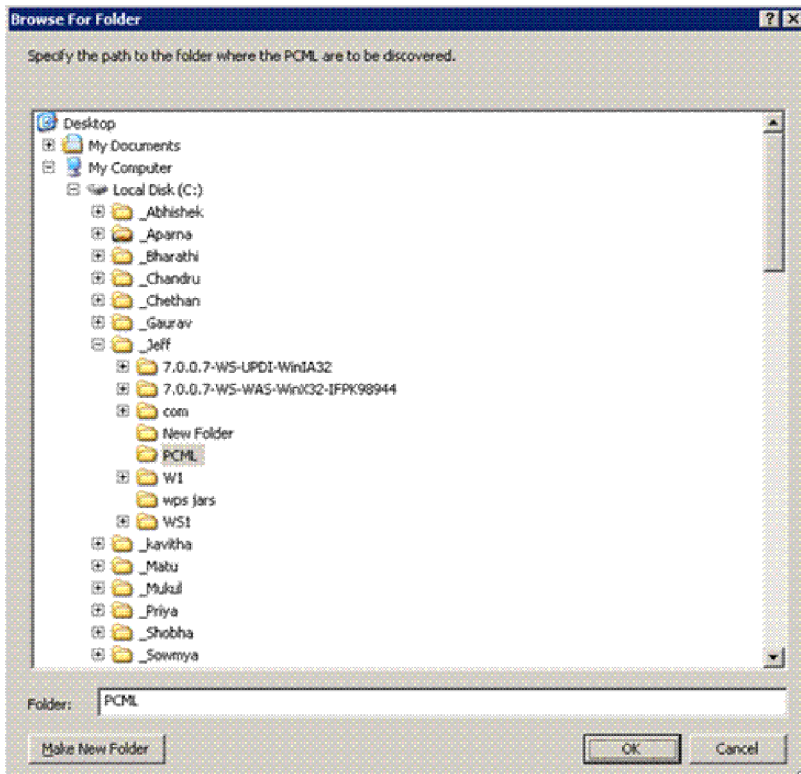
The Specify the Discovery Properties window displays, as shown in this figure.

Figure 14. Specify the Discovery Properties window



2. To set your connection properties:
 - In the **Select the discovery mode** field, select **Local Directory** from the list of options.
 - In the **Object type to discover** field, select **RPG via PCML (Local)** from the list of options.
3. Click **Browse**. The adapter uses this connection information to connect to your local file system, and discover business objects, as shown in this figure.

Figure 15. Specify the path to the PCML Discovery window



4. The Specify the Discovery Properties window reappears, displaying the **Path to the local folder for object discovery** information, as well as the Connection properties section enabled. Click **Next**.

Note: : The external service wizard connects to a local file system only to discover and create a service description. You have to connect to the IBM i system during run time, to use the business objects that you generate from your local file system. A corresponding option will be created, to associate the configuration information with the business object you select for discovery

5. Specify the connection properties, such as the Host name, User name and Password in the **Specify the Service Generation and Deployment Properties** window.
6. 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.
7. Optional: To change the location of the logging 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.

The log information is only for the wizard. At run time, the adapter writes messages and trace information into the standard log and trace files for the server.

8. Click **Next** to proceed with your discovery.

Results

The external service wizard connects to the IBM i server using the information you provided, and the Find Objects in the Enterprise System window is displayed.

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 IBM Integration Designer to find and select business objects and services from the IBM i system, and to generate business object definitions and related artifacts.

Discovering objects

After connecting to the IBM i system, you can run a query to search for the required objects. You can browse the discovered objects navigation tree to understand the structure of objects in the IBM i system. You can also search to display only the objects you need to produce a service that typically accesses one or more Enterprise information systems (EIS).

Before you begin

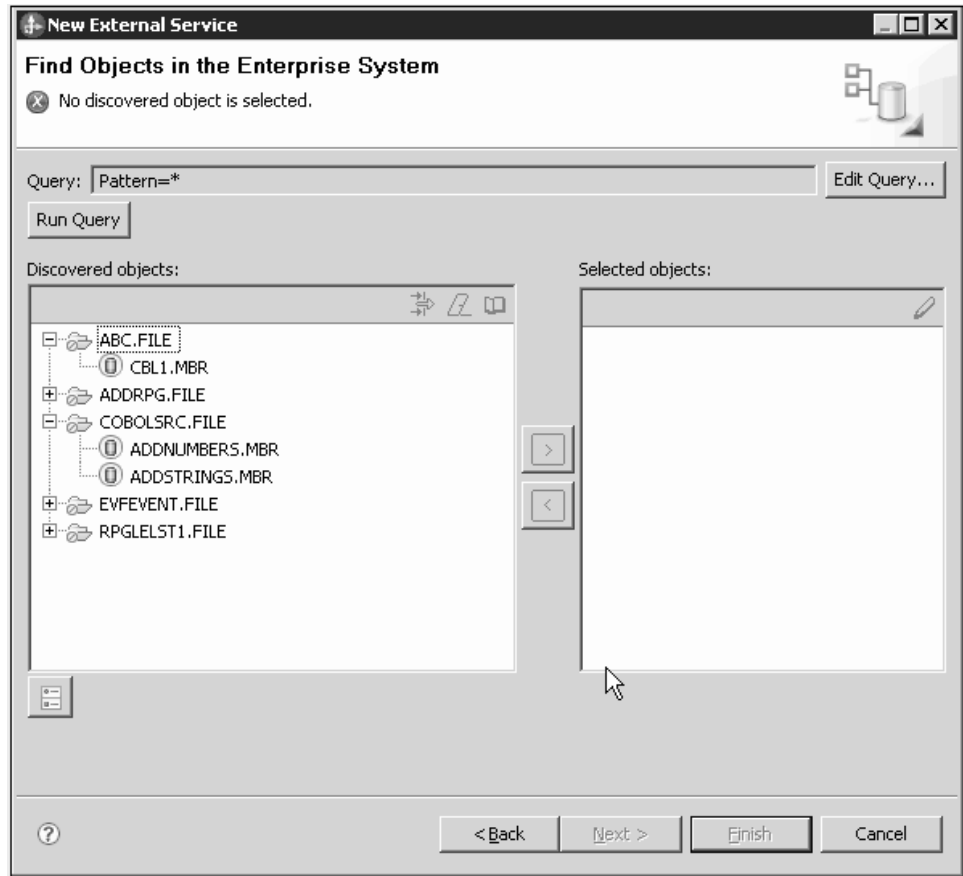
You must have successfully connected to the IBM i system.

About this task

To discover objects on the enterprise system, you can run the query to call the service in the Find Objects in the Enterprise System window of the external service wizard.


Procedure

1. In the Find Objects in the Enterprise System window, click **Edit Query** to modify a query, corresponding to your needs. The **Specify the Query Properties** window is displayed.
2. Specify a value to filter data queue or program matching the pattern. The acceptable characters in the search pattern are wildcard characters, such as asterisk (*) and question mark (?). Search result fetches files and is not case-sensitive. The result of running a typical query is shown in the following figure.



Note:

The **Discovered objects** pane lists all the objects that are discovered based on your query.

3. In the **Discovered objects** list, click + or the plus sign to expand a node and view the objects discovered by the wizard.
4. Select the required objects from the **Discovered objects** list and use the  button to move these items to the **Selected objects** list.

Results

The wizard displays the objects that you have selected, which can be used by the adapter.

What to do next

Continue working in the external service wizard. The next step is to select the objects you want to use in your module and configure each object.

Accessing messages using simple or keyed IBM i data queue

To configure a module to use the adapter for accessing messages using data queue, you use the external service wizard in IBM Integration Designer to find a simple IBM i data queue such as first-in first-out (FIFO) or last-in first-out (LIFO) or a keyed IBM i data queue. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services

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

Ensure you have set the connection properties for the external service wizard.

About this task

To find and select one or more objects, use the following procedure. The Find Objects in the Enterprise System window allows you to select and access non keyed IBM i Data queues such as first-in first-out (FIFO) or last-in first-out (LIFO) or a keyed IBM i Data queues. For more information on the properties in this topic, see Managed connection factory properties.

Procedure

1. Ensure you have selected Data queue option from the **Object type to discover** list while you are selecting an object type in the connection properties for accessing the data queues. In the Discovered objects list of the Find Objects in the Enterprise System window, select either a non keyed data queue or a keyed data queue as per your requirement and then click the  button to move it to the selected set.
2. In the Specify the Configuration Properties for 'object' window, specify configuration properties for the data queue.
3. **Optional:** To configure Additional properties (such as bidi-related properties or logging and tracing details), click **Advanced** and expand the Additional properties, Bidi properties, or Logging and tracing sections as needed.
4. **Optional:** Select the **Change logging properties for wizard** check box if you want to define the level of logging for this module.
5. Click **OK** to add the object to the list of objects to be imported.
6. When you have selected all objects, click **Next**. The Specify Composite Properties window is displayed.

Results

The business objects you configured for data queue are listed in the Find Objects in the Enterprise System window.

What to do next

From the Specify Composite Properties window, add or remove associated operations.

Setting global properties for operations

To configure the business object, you specify information about the object (such as the operation associated with the object). For data queue, the operation names are PutQueue and GetQueue.

About this task

The Specify Composite Properties window displays the operation name for selected business objects. Configure PutQueue and GetQueue operations for a non

keyed Data queue such as first-in first-out (FIFO) or last-in first-out (LIFO) or a keyed Data queue for use in your module by setting the configuration properties for the business object in the configuration properties window. To configure the business object, use the following procedure.

Procedure

1. The Specify Composite Properties window for a Data queue business object displays the PutQueue and GetQueue operations as shown in Figure 1. You can specify the properties for a non keyed Data queue such as first-in first-out (FIFO) or last-in first-out (LIFO) or a keyed data queue in the Specify Composite Properties window depending on your requirement. For example, choose PutQueue. You can then **Add** or **Remove** the selected operation.

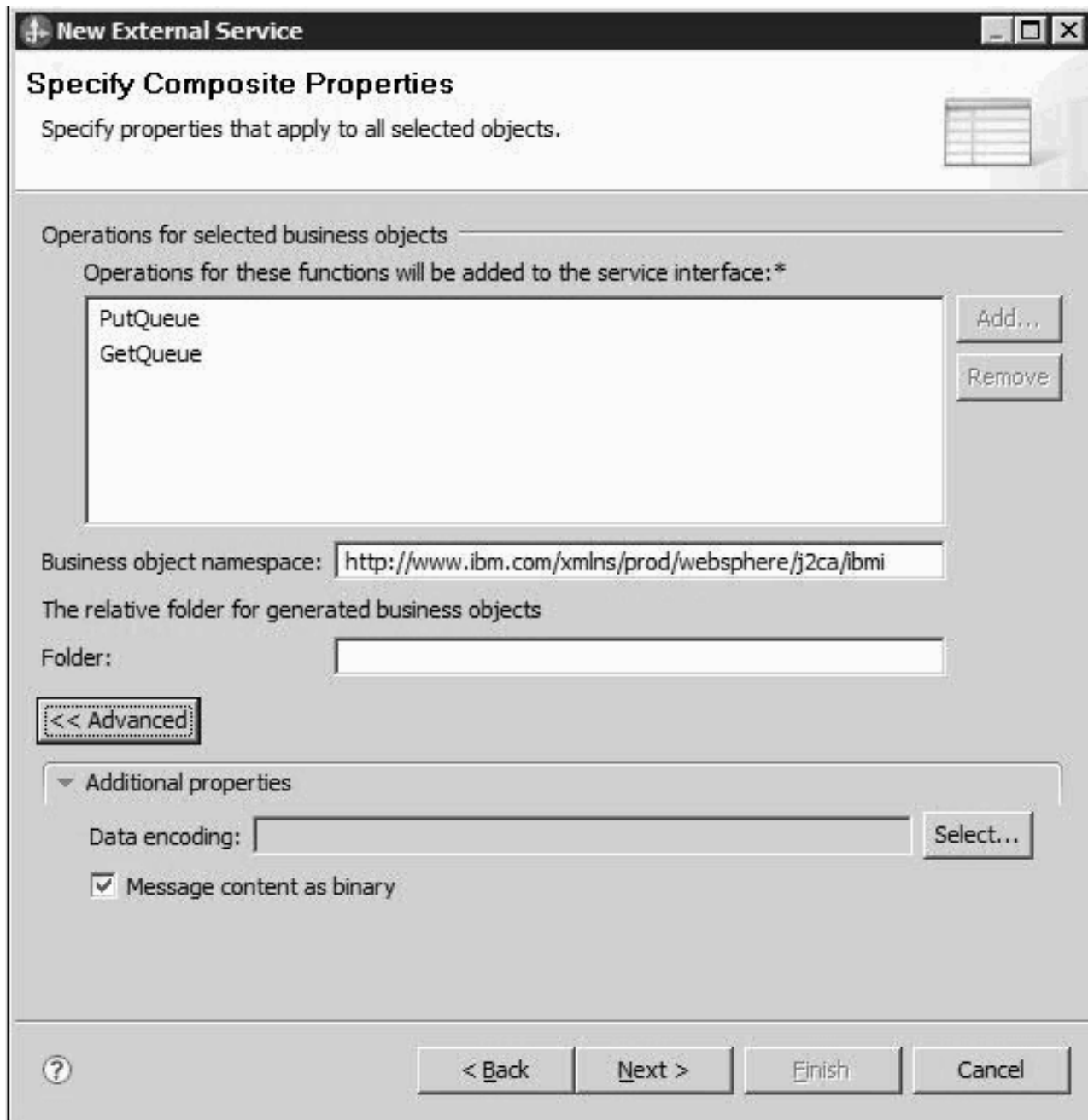


Figure 16. Specify Composite Properties window

2. Based on your requirement, use **Add** to add any of these operations or **Remove** to remove any of these operations. If you choose to remove any of these operations, the removed operation is made available in the **Add** operation for you to add back.
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.

5. Optional: In the **Data encoding** field, click the **Select** button to specify the type of character encoding to use.
6. Optional: Select the **Message content as binary** field if you use the message in binary format for the PutQueue and GetQueue operations. Selecting this option results in the business objects to put and get messages from the non keyed or keyed data queue in the binary format.

Note: For Keyed Data queue, with message content in binary format, the message key length must be same as the length supported by the Keyed Data queue.

7. 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 environment and generating the service description

After you select and configure business objects for your module, use the external service wizard to configure runtime environment that the adapter uses to connect to a IBM i system. The wizard creates a new business service module where all the artifacts and property values are saved. This task is performed using the Specify the Service Generation and Deployment Properties and Specify the Location Properties window of the external service wizard.

Results

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

Setting deployment properties for runtime environment:

The connection properties in this task are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as User name and Password here.

Before you begin

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

About this task

To generate artifacts for your module follow this procedure:

Procedure

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

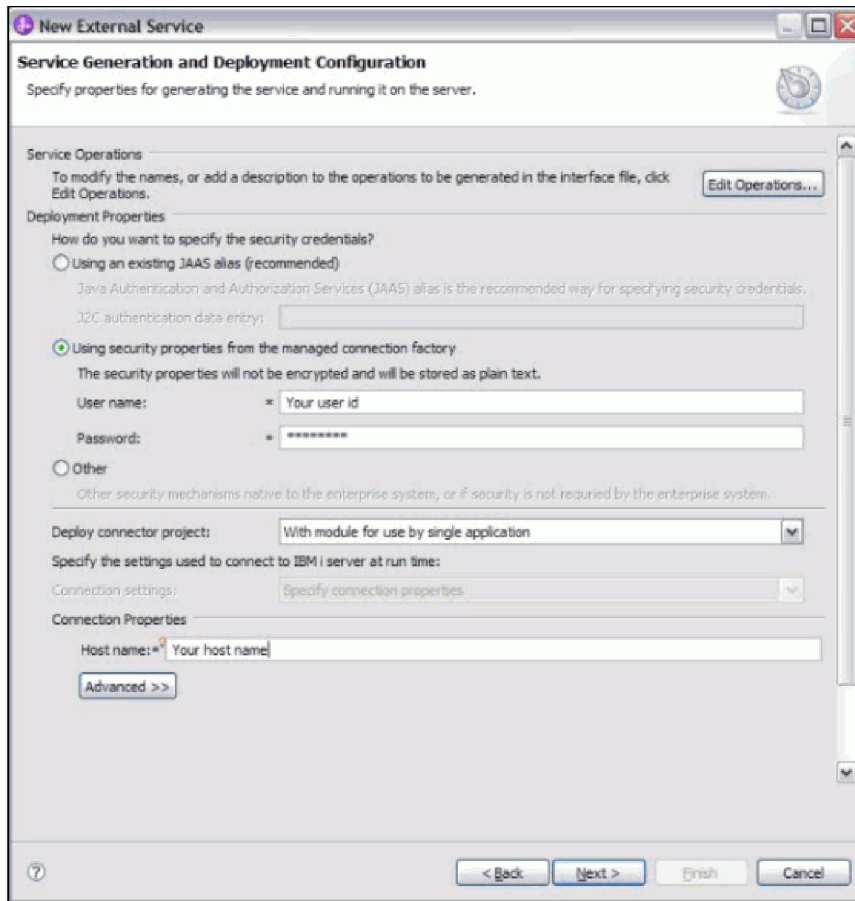


Figure 17. Service Generation and Deployment Configuration window

2. Specify how you want the adapter to connect to IBM i system. In the Deployment Properties area, specify how you want the adapter to get the user name and password at runtime environment.
 - To use a J2C authentication alias, select **Using an existing JAAS alias (recommended)** and type the name of the alias in **J2C Authentication Data Entry**. You can 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.
 - To use managed connection properties, select **Using security properties from the managed connection factory** and type the values in the **User name** and **Password** fields.
 - To administer the user name and password from other mechanism, click **Other**.
3. To use a J2C authentication alias, select **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** check box 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. 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 environment.
 - 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 settings**, 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**.
5. 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 runtime environment. 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.
 6. Specify advanced properties by clicking **Advanced** as shown in the following figure. 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.



- **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, see “Adapter ID (AdapterID)” on page 131
- **Additional properties:**
 - Set the Maximum number of retries in case of system connection failure to the appropriate positive integer.
 - a. If the property value is set to 0, then the adapter does not perform any EIS connection validation and executes the outbound operation. If the EIS connection is invalid, the outbound operation fails. Though the subsequent requests get executed successfully if the IBM i system is functional, the current request fails.
 - b. If the property value is set to greater than 0, then during each request, the adapter validates that the EIS connection is active/alive.
 - c. If the connection is valid, then the operation is completed.

- d. If the connection is invalid, the adapter invalidates the current managed connection so that a new managed connection is created (new physical connection).
- e. If the connection is created successfully, the outbound operation is completed. Otherwise a ResourceException error is thrown

Note: Set the Time interval between retries if the connection fails (milliseconds) field with the appropriate integer to indicate the time in milliseconds, in between retries. This property is enabled only when the connectionRetryLimit property has a value, greater than 0.

- **Globalization**

- Select the Globalization property, to encode the input and output values for the Call PGM operation.
- Specify the CCSID either in the PCML or in the EMD. If CCSID is present in both PCML and EMD, the EMD CCSID is recognized as the highest priority for both single byte and double byte character values. If no EMD value is provided, then the PCML CCSID is recognized as the default coded character set identifier.
- Enter the values for both these properties in the following screen



Figure 18. CCSID support for PCML and EMD encoding values (for Chinese characters)

- a. Connection CCSID- the CCSID used for the connection.
- b. Data CCSID- the CCSID used for the conversion of data (for DBCS graphic fields).

- **Process illegal XML character:**

From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement

- Select **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- Select **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML character in the trace file, at runtime.
- Select **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML character, and log the character in the trace file, at runtime.

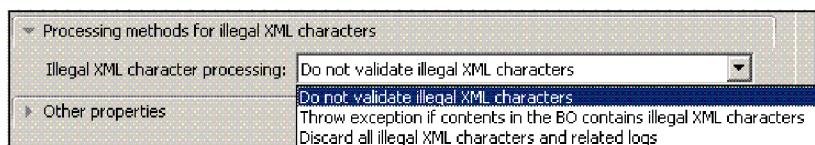


Figure 19. Process Illegal XML character window

7. Click **Next**. The Specify the Location Properties window is displayed.

Results

The Specify the Location Properties window is displayed.

Setting location properties for artifacts:

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Before you begin

This task is performed using the Specify the Location Properties window of the external service wizard.

About this task

To configure location properties for your module follow this procedure:

Procedure

1. In the Specify the 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 Select a Business Integration Project Type window, select **Create a module project** and click **Next**.
 - c. In the Create a 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 Integration Designer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
 - f. Click **Finish** to create the new module.
2. Specify the namespace you want to use for your business objects schema.
 - 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**.
 3. 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).
 4. In **Name**, accept the default import name or type a different name.

5. 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**.
6. In **Description**, type a descriptive comment about the module.
7. When you are finished setting properties, click **Finish**.

Note: If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The artifacts are generated.

Calling IBM i programs

To create a service that leverages the program call markup language (PCML) standard to call an RPG, COBOL, and service programs (SRVPGM) on the target IBM i machine without special coding, you use the enterprise service discovery (ESD) wizard to discover the program via generated PCML. The external service wizard in IBM Integration Designer auto generates all required artifacts. You can use the generated artifacts to run the underlying program or service program. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services

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


Ensure that you have set the connection properties for the External service wizard.


About this task

The adapter reads the configured **pre commands** from the business object and runs the **pre commands** on the IBM i system. To call a program, use the following procedure. The Find Objects in the Enterprise System window allows you to select and call programs. For more information about any of the properties in this topic, see the manage connection factory properties reference topic in this guide.

Procedure

1. Ensure that you have selected the required program via PCML option from the **Object type to discover** list while you are selecting an object type while setting the connection properties for calling the program in the Specify the Discovery Properties window.
 - **Calling RPG program:** Ensure that you have selected the correct RPG or RPG LE program and the PCML type in the **Discovered objects** list of the Find Objects in the Enterprise System. Select the required objects from the **Discovered objects** list and use the  button to move these items to the **Selected objects** list.
 - **Calling COBOL program:** Ensure that you have selected the correct COBOL or COBOL LE program and the PCML type in the **Discovered objects** list of

the Find Objects in the Enterprise System. Select the required objects from the **Discovered objects** list and use the  button to move these items to the **Selected objects** list.

- **Calling Service program:** Ensure that you have selected the correct service program (SRVPGM) and the PCML type in the **Discovered objects** list of the Find Objects in the Enterprise System. Select the required objects from the **Discovered objects** list and use the  button to move these items to the **Selected objects** list.

Note: For more details information about Call IBM i programs supported by the WebSphere Adapter for IBM i, see the topics “Call IBM i programs” on page 6 and “CallPGM operation” on page 10 in this guide.

2. When you click **Next** in the Specify the Discovery Properties window, only PCMLs of the program via PCML type selected by you previously in the metadata discovery tree in the Find Objects in the Enterprise System window are displayed. Invalid PCMLs are marked with an invalid icon indicating they cannot be selected and added to list of selected objects.

Note: For the number of programs in the PCML, only those many numbers of business objects are created. Invalid PCMLs are those PCML for which the path is null or the Call IBM i program object types such as RPG or RPG LE, COBOL or COBOL LE or service program (SRVPGM) does not exist.

3. The Find Objects in the Enterprise System window allows you to select and configure objects. The metadata object selected in the tree and on click of details button shows the PCML content as shown in the following figure.

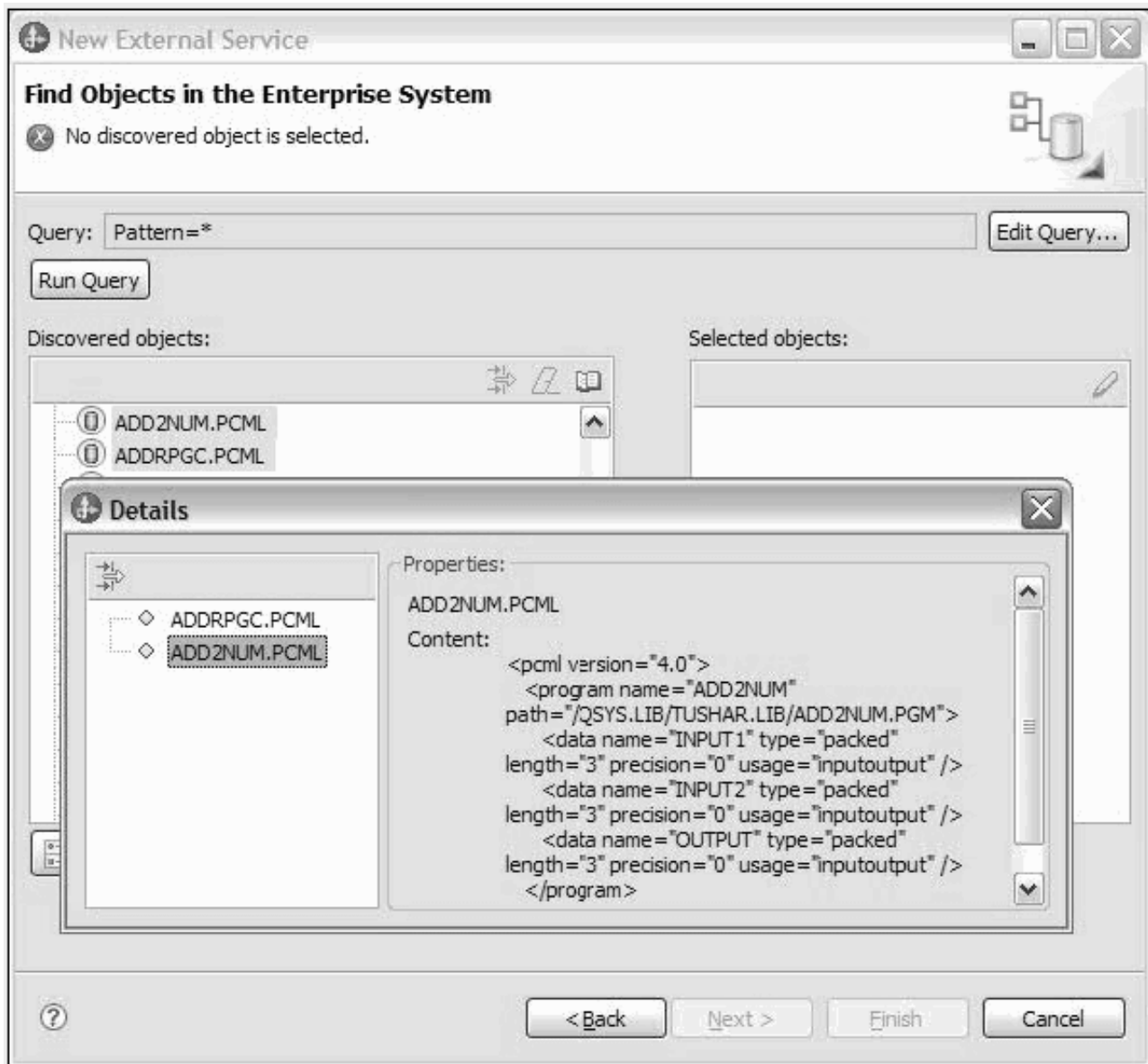



Figure 20. Find Objects in the Enterprise System window

4. Select the PCML as per your requirement, for example, if you selected RPG via PCML or RPG LE via PCML, select ADD2NUM.PCML and then click the  button to move it to the selected set. In the Specify the Configuration Properties for 'object' window, specify configuration properties for the object. Proceed to configure the **pre commands** and **post commands**. For more details on how to configure the pre commands and post commands, see the Configuring pre and post commands for PGM execution section.

Configuring pre and post commands for program execution:

You can specify pre and post commands for program execution such as RPG via PCML, COBOL via PCML, SRVPGM via PCML, RPG Program, and COBOL Program for use in your module. Specify the configuration properties for the selected business object in the Configuration Properties window.

Procedure

1. If you have selected the RPG Program or COBOL Program from the Object type to discover list, then continue with the following steps and skip step 2.
 - a. When you have selected the RPG Program or COBOL Program in the metadata object in the metadata tree, the corresponding MBR files of the program file is listed. You have an option to select either the RPG or COBOL metadata object and move to the right pane. A new window with the path to the newly compiled PCML file displays.

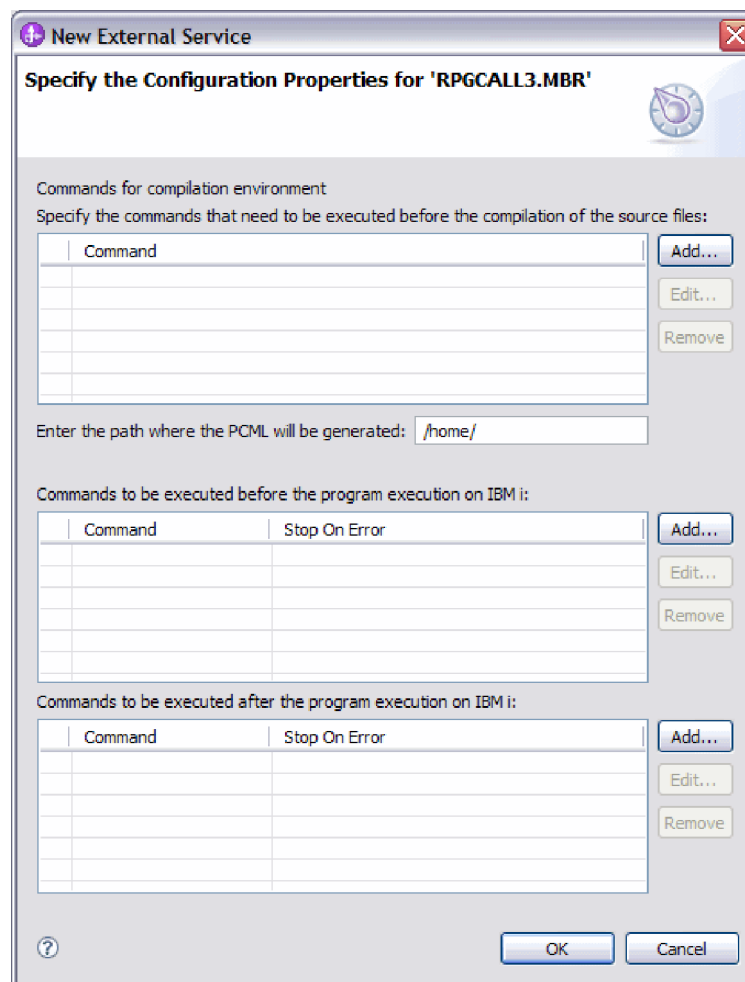


Figure 21. Specify the Configuration Properties for RPGCALL3.MBR window

- Note:** You will not get any exception message when there is a pre-compilation command failure. See the log files for troubleshooting.
- b. Enter the commands for the compilation environment. These commands run even before the adapter compiles the RPG or COBOL programs as PCMLs.
2. When you have selected the RPG via PCML or COBOL via PCML option from the **Object Type to discover** list, continue with the following steps.
 3. If you have selected the object type as RPG via PCML or COBOL via PCML, you can specify the path to store the generated PCML files in the IBM i system. You have an option to change the default folder path.

Note: The procedure varies, when you select the SRVPGM via PCML option in the **Object Type to discover** list.

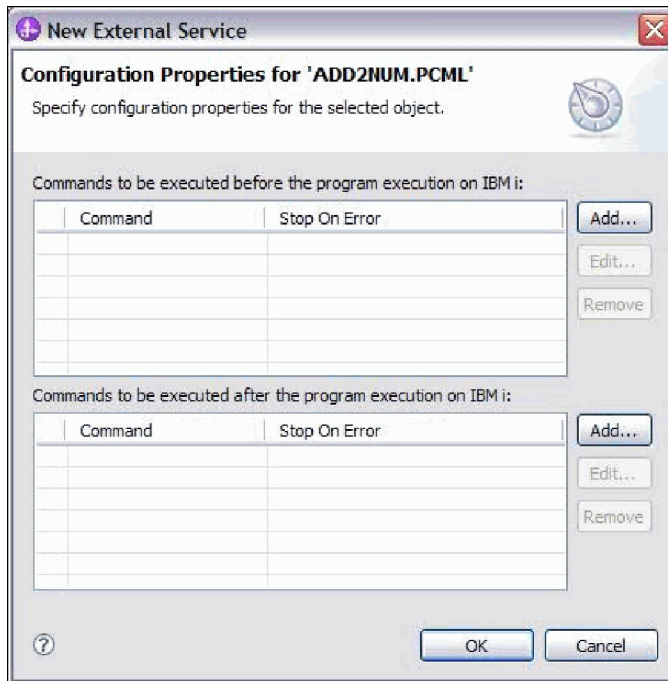


Figure 22. Configuration properties for ADD2NUM.PCML window

4. Enter the pre and post commands to be executed before the program execution on IBM i system. Pre commands are to be executed before the program execution on IBM i. Post commands are to be executed after the program execution on IBM i. You can add any number of pre commands and post commands. Stop On Error indicates whether the execution must stop or continue in case of any error during the command execution. If Stop On Error for a command is set to true and error occurs during execution of the command, then an `ErrorCompletingRequestFault` is generated and execution stops. If Stop On Error for a particular command is set to false and if an error occurs during command execution, then the command execution continues to execute the next commands and program, irrespective of the error.
5. Click **Add** to add commands to be executed before or after the program execution on IBM i. The following figure shows the Add/Edit window to add a new command.

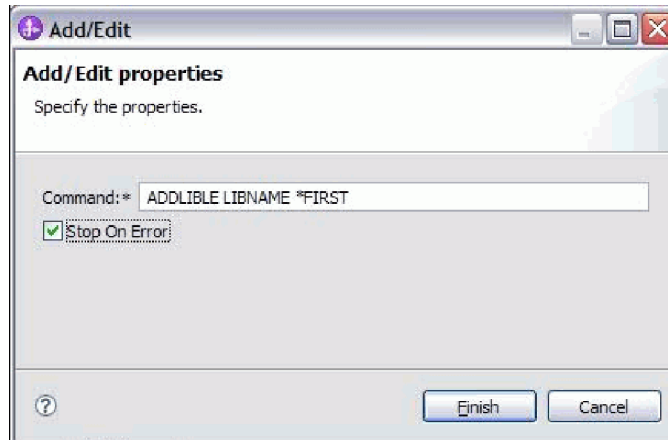


Figure 23. Add Edit Properties window

6. In the **Command** field, enter the command name.
7. Select the Stop On Error check box to stop the execution of the command if there are errors during the execution.
8. Click **Finish** to add the specified properties. A new row is appended to the table, as shown in the following figure.

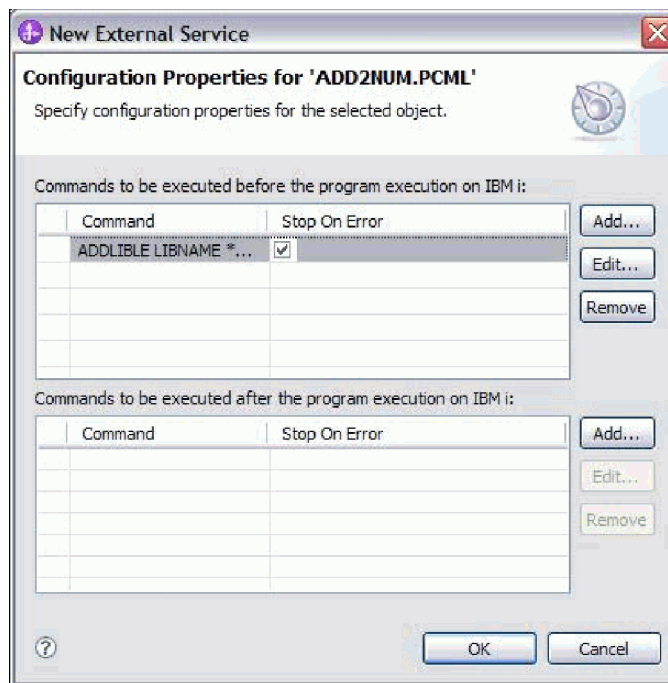


Figure 24. Configuration Properties for program window

9. You can also select multiple objects in the Find Objects in the Enterprise System window. When you have selected the SRVPGM via PCML option from the **Object type to discover** list, as shown in this figure, click **OK** and follow steps 3 to 9.
10. After you have selected the business objects you want to add to the module, click **Next**. The Specify Composite Properties window is displayed.
11. When you have selected the SRVPGM via PCML option from the Object type to discover list, click **OK** and follow steps 3 to 9. The Specify the Configuration

Properties for ADD2NUMSRV.PCML window appears.

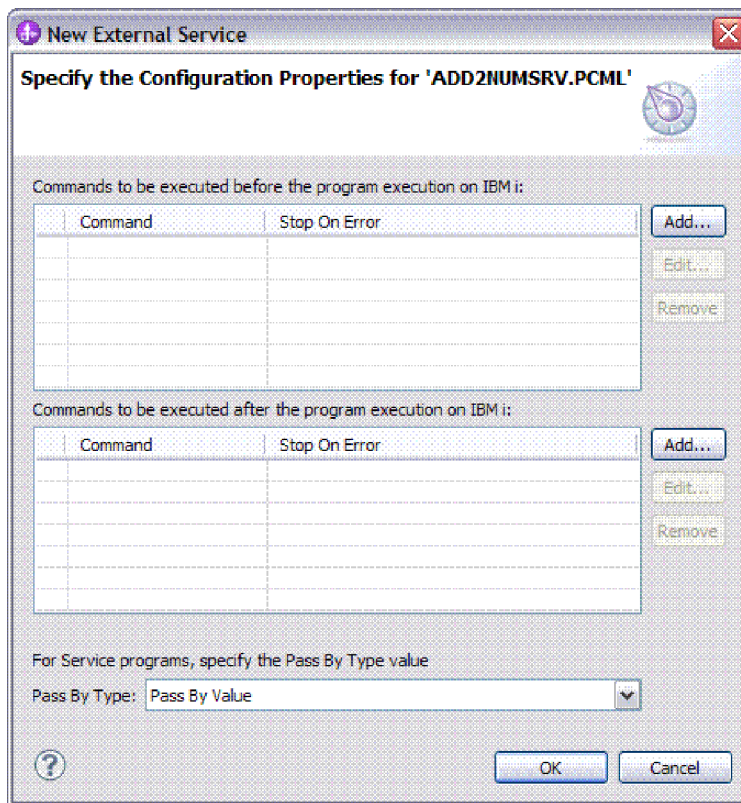


Figure 25. Specify the Configuration Properties for ADD2NUMSRV.PCML window

12. For this service program, you can specify the Pass By Type. You can choose between **Pass By Value** and **Pass By Reference**, and click **OK**.

Results

The business objects you configured are listed in the Find Objects in the Enterprise System window.

What to do next

From the Specify Composite Properties window, add or remove the associated operations.

Setting global properties for operations

To configure the business object, you specify information about the object such as the operation associated with the object.

Before you begin

Ensure you have selected and imported the object.

About this task

The Specify Composite Properties window displays the operation name for selected business objects. For RPG via PCML, COBOL via PCML, SRVPGM via PCML, the operation name is CallPGM.

Procedure

1. 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.
2. 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 description as a module.

Setting deployment environment and generating the service description

After you select and configure business objects for your module, use the external service wizard to configure runtime environment that the adapter uses to connect to a IBM i system. The wizard creates a new business service module where all the artifacts and property values are saved. This task is performed using the Specify the Service Generation and Deployment Properties and Specify the Location Properties window of the external service wizard.

Results

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

Setting deployment properties for runtime environment:

The connection properties in this task are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as User name and Password here.

Before you begin

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

About this task

To generate artifacts for your module follow this procedure:

Procedure

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

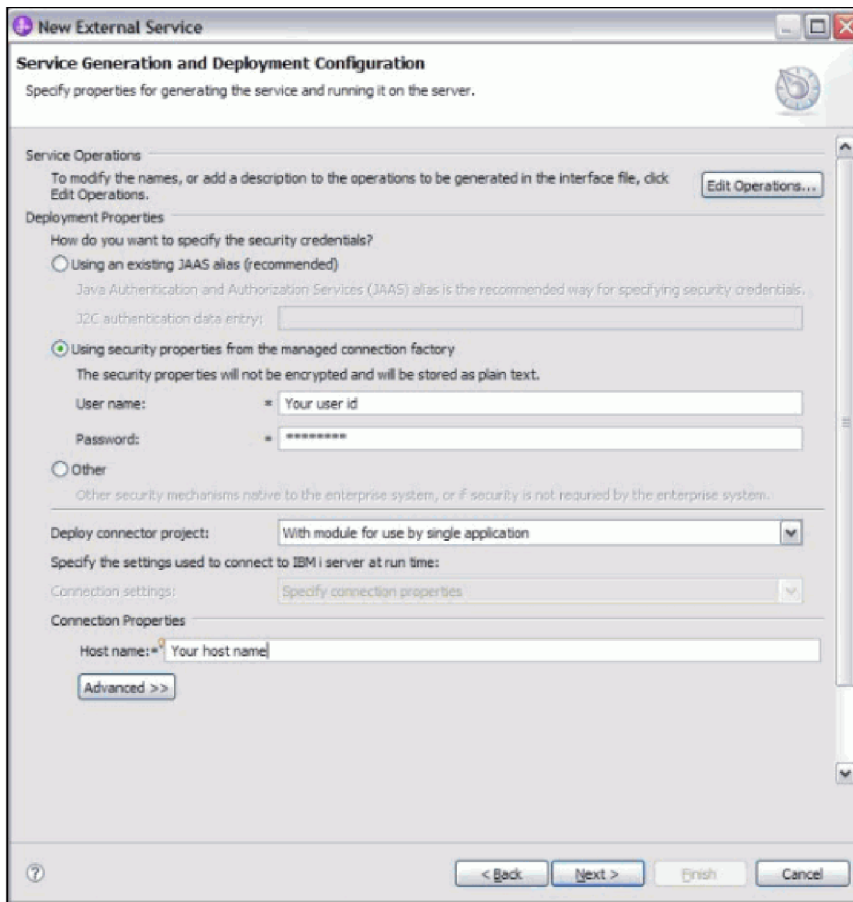


Figure 26. Service Generation and Deployment Configuration window

2. Specify how you want the adapter to connect to IBM i system. In the Deployment Properties area, specify how you want the adapter to get the user name and password at runtime environment.
 - To use a J2C authentication alias, select **Using an existing JAAS alias (recommended)** and type the name of the alias in **J2C Authentication Data Entry**. You can 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.
 - To use managed connection properties, select **Using security properties from the managed connection factory** and type the values in the **User name** and **Password** fields.
 - To administer the user name and password from other mechanism, click **Other**.
3. To use a J2C authentication alias, select **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** check box 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. 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 environment.
 - 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 settings**, 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**.
5. 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 runtime environment. 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.
6. Specify advanced properties by clicking **Advanced** as shown in the following figure. 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.



- **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, see “Adapter ID (AdapterID)” on page 131
- **Additional properties:**
 - Set the Maximum number of retries in case of system connection failure to the appropriate positive integer.
 - a. If the property value is set to 0, then the adapter does not perform any EIS connection validation and executes the outbound operation. If the EIS connection is invalid, the outbound operation fails. Though the subsequent requests get executed successfully if the IBM i system is functional, the current request fails.
 - b. If the property value is set to greater than 0, then during each request, the adapter validates that the EIS connection is active/alive.
 - c. If the connection is valid, then the operation is completed.

- d. If the connection is invalid, the adapter invalidates the current managed connection so that a new managed connection is created (new physical connection).
- e. If the connection is created successfully, the outbound operation is completed. Otherwise a ResourceException error is thrown

Note: Set the Time interval between retries if the connection fails (milliseconds) field with the appropriate integer to indicate the time in milliseconds, in between retries. This property is enabled only when the connectionRetryLimit property has a value, greater than 0.

- **Globalization**

- Select the Globalization property, to encode the input and output values for the Call PGM operation.
- Specify the CCSID either in the PCML or in the EMD. If CCSID is present in both PCML and EMD, the EMD CCSID is recognized as the highest priority for both single byte and double byte character values. If no EMD value is provided, then the PCML CCSID is recognized as the default coded character set identifier.
- Enter the values for both these properties in the following screen

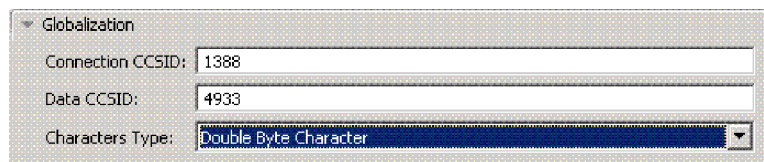


Figure 27. CCSID support for PCML and EMD encoding values (for Chinese characters)

- a. Connection CCSID- the CCSID used for the connection.
- b. Data CCSID- the CCSID used for the conversion of data (for DBCS graphic fields).

- **Process illegal XML character:**

From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement

- Select **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- Select **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML character in the trace file, at runtime.
- Select **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML character, and log the character in the trace file, at runtime.

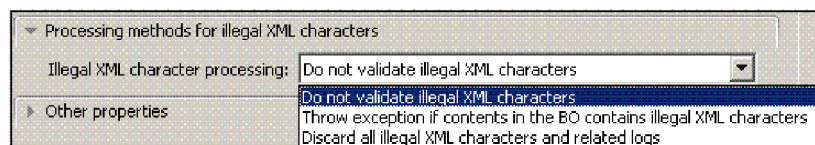


Figure 28. Process Illegal XML character window

7. Click **Next**. The Specify the Location Properties window is displayed.

Results

The Specify the Location Properties window is displayed.

Setting location properties for artifacts:

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Before you begin

This task is performed using the Specify the Location Properties window of the external service wizard.

About this task

To configure location properties for your module follow this procedure:

Procedure

1. In the Specify the 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 Select a Business Integration Project Type window, select **Create a module project** and click **Next**.
 - c. In the Create a 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 Integration Designer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
 - f. Click **Finish** to create the new module.
2. Specify the namespace you want to use for your business objects schema.
 - 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**.
 3. 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).
 4. In **Name**, accept the default import name or type a different name.

5. 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**.
6. In **Description**, type a descriptive comment about the module.
7. When you are finished setting properties, click **Finish**.

Note: If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The artifacts are generated.

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 Find Objects in the Enterprise System window contains all of the business objects you want to use in your application, click **Next**.
2. The Specify Composite Properties window displays the operation name, which is `Emit`. The `emit` operation is the only operation available during inbound processing.
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 definition as a module.

Configuring the module for inbound processing

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

Pre requisites for inbound processing

To configure the server for inbound processing, you must create two keyed data queues for each inbound Adapter, and extract the compiled control language (CL) programs which are used to complete a specific service. These CL programs are present in the installation directory for IBM Integration Designer.

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 command for creating the data queue are: `CRTDTAQ DTAQ(<library>/<QueueName>) TYPE(*STD) MAXLEN(128) SEQ(*KEYED) KEYLEN(18) TEXT('Description')`
2. The CL programs are located in the `WID_installation_dir/ResourceAdapters/Ios_version/scripts` directory, where `WID_installation_dir` is the installation directory for IBM Integration Designer, and `version` is the version of the adapter, for example, 7.0. 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. 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.

Discovering objects

After connecting to the IBM i system, you can run a query to search for the required objects. You can browse the discovered objects navigation tree to understand the structure of objects in the IBM i system. You can also search to display only the objects you need to produce a service that typically accesses one or more Enterprise information systems (EIS).

Before you begin

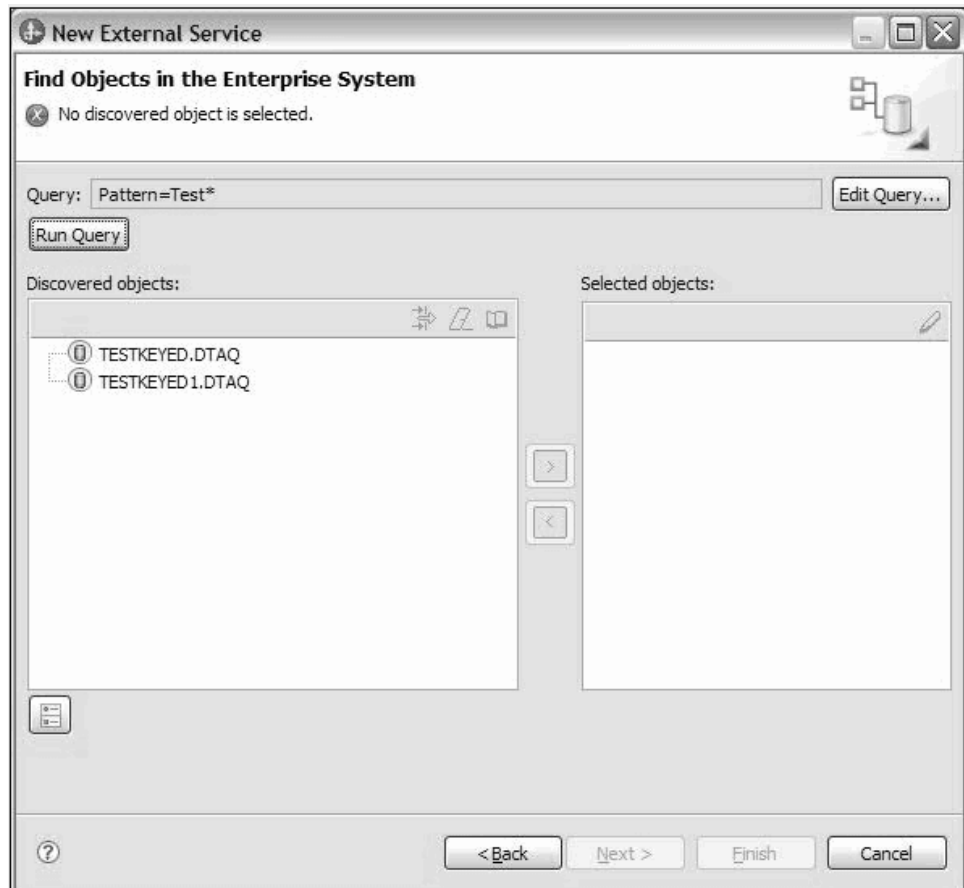
You must have successfully connected to the IBM i system.

About this task

To discover objects on the enterprise system, you can run the query to call the service in the Find Objects in the Enterprise System window of the external service wizard.


Procedure

1. In the Find Objects in the Enterprise System window, click **Edit Query** to modify a query, corresponding to your needs. The **Query Properties** window is displayed.
2. In the Find Objects in the Enterprise System window, click **Run Query** to search for discovered objects. The result of running a typical query is shown in the following figure.



Note: The acceptable characters in the search pattern are wildcard characters, such as asterisk (*) and question mark (?). Search result fetches files and is not case sensitive.

The **Discovered objects** pane lists all the objects that are discovered based on your query.

3. In the **Discovered objects** list, click + or the plus sign to expand a node and view the objects discovered by the wizard.
4. Select the required objects from the **Discovered objects** list and use the  button to move these items to the **Selected objects** list.

Results

The wizard displays the objects that you have selected, which can be used by the adapter.

What to do next

Continue working in the external service wizard. The next step is to select the objects you want to use in your module and configure each object.

Accessing messages using simple or keyed IBM i data queue

To configure a module to use the adapter for accessing messages using data queue, you use the external service wizard in IBM Integration Designer to find a simple IBM i data queue such as first-in first-out (FIFO) or last-in first-out (LIFO) or a keyed IBM i data queue. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services

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 IBM Integration Designer below the adapter project. For more information on creating the adapter project, see “Creating the project” on page 38

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**. The Discovery Configuration window is displayed.
2. Optional: For last-in first-out (LIFO) or first-in first-out (FIFO) data queue polling, in the Discovery Configuration window, type the values in the host name, user name, password and the path to folder on IBM i for object discovery.

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

3. Click **Next**. The Find Objects in the Enterprise System window is displayed as shown in the following figure.

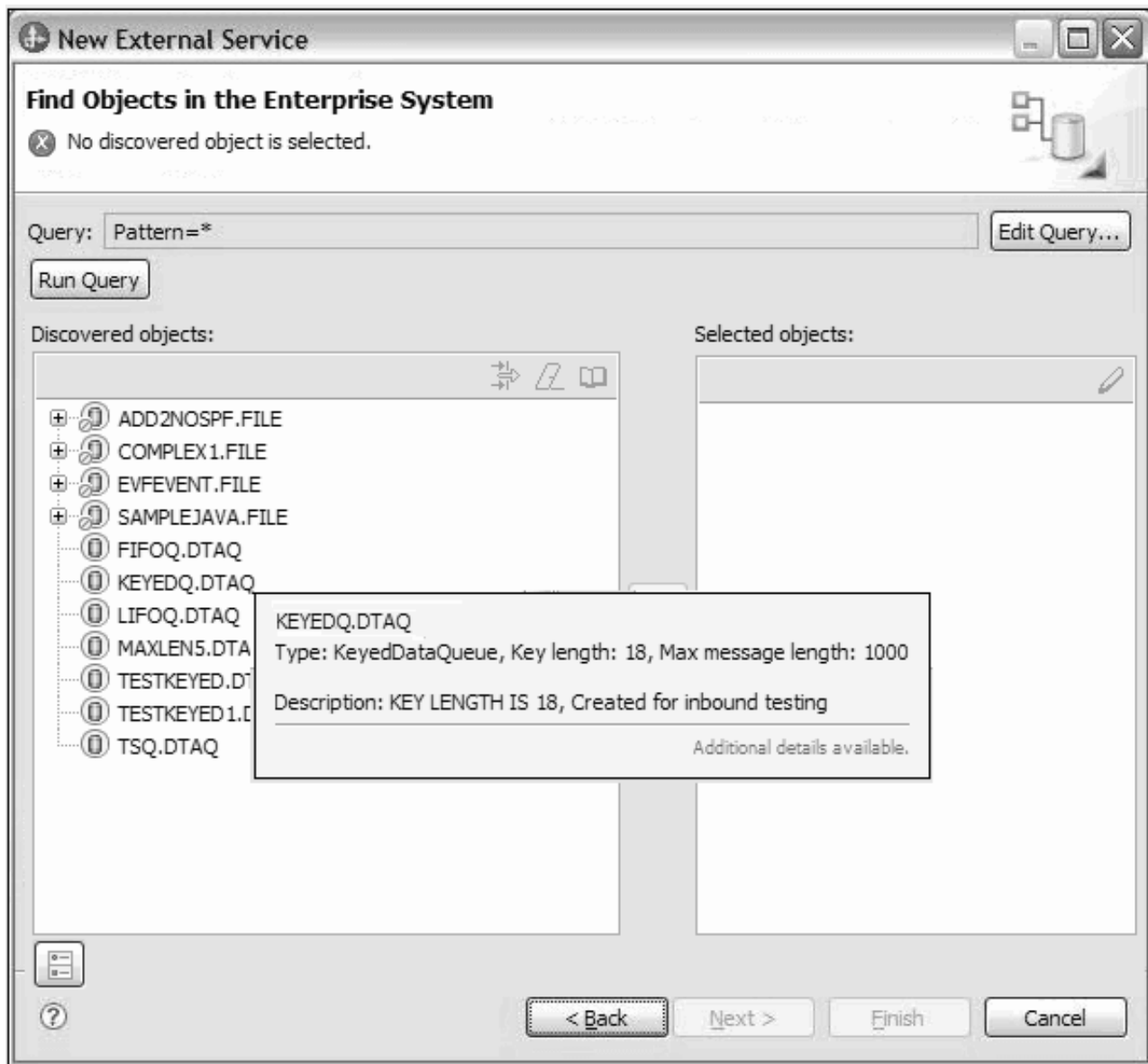


Figure 29. Find Objects in the Enterprise System window

4. Optional: In the Find Objects in the Enterprise System window, select an object from the **Discovered objects** pane. For LIFO or FIFO data queue polling, select appropriate data queue object. For example in the above figure one of the FIFO data queues is FIFOQ.DTAQ. When you mouse over the data queue, the details such as data queue type, message maximum length and description and so on are displayed as tool tip. Click **Next**.
5. 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.

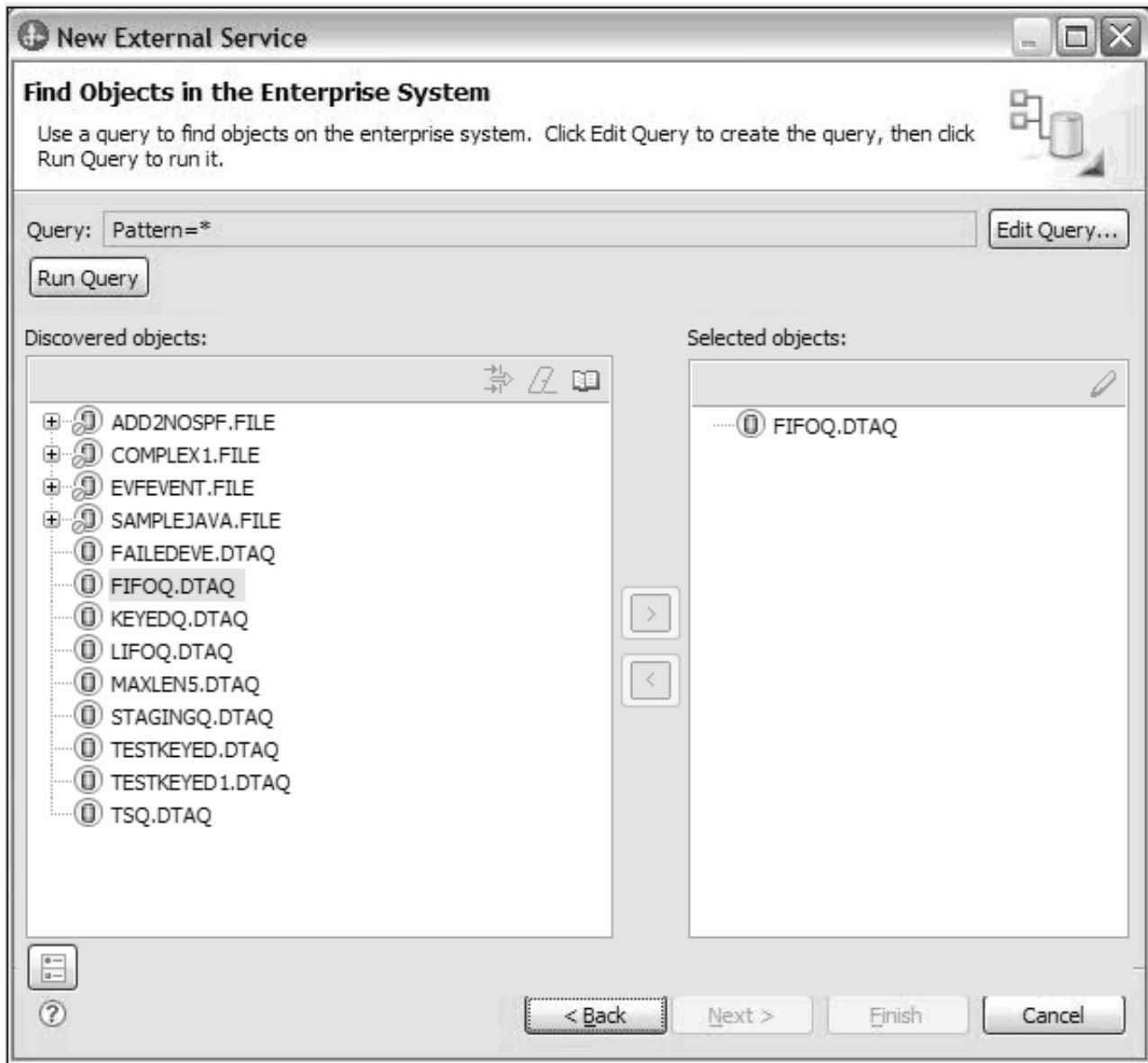


Figure 30. Find Objects in the Enterprise window

6. Optional: For a keyed data queue polling, in the Find Objects in the Enterprise System window, select an object from the **Discovered objects** list. Click **Next**.
7. Select KEYEDQ.DTAQ from the Discovered objects pane. Click the > (Add) button. The added objects are displayed in the Selected objects pane. Click **Next**.

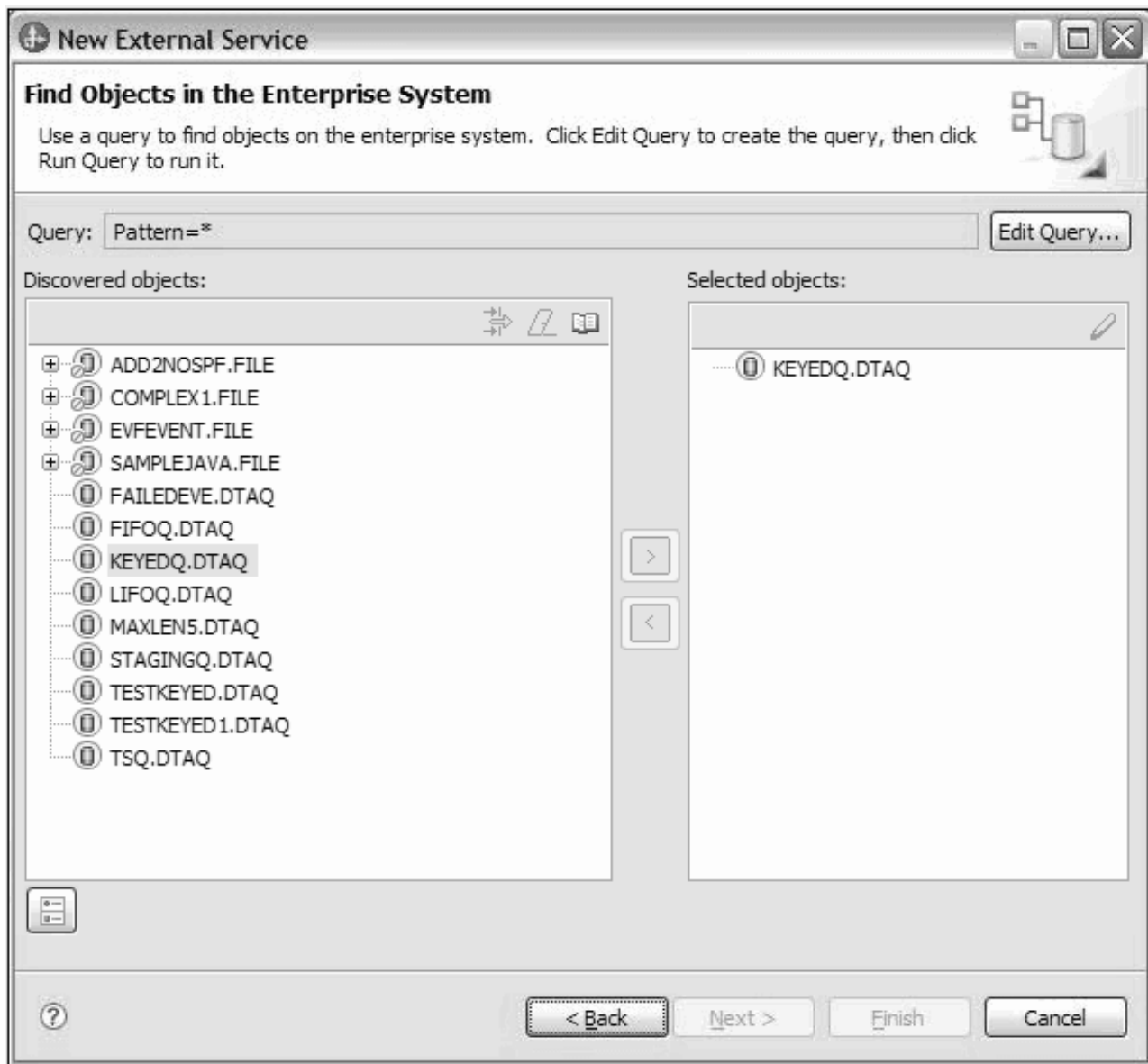


Figure 31. Find Objects in the Enterprise System window

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 Find Objects in the Enterprise System window contains all of the business objects you want to use in your application, click **Next**.

2. The Specify Composite Properties window displays the operation name, which is Emit. The emit operation is the only operation available during inbound processing.
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 definition as a module.

Setting deployment environment and generating the service description

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 a IBM i system. The wizard creates a new business integration module where all the artifacts and property values are saved. This task is performed using the Specify the Service Generation and Deployment Properties and Specify the Location Properties windows of the external service wizard.

Results

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

Setting deployment properties for runtime environment:

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.

Before you begin

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

About this task

To generate artifacts for your module follow this procedure:

Procedure

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

Figure 32. 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. **For a simple data queue:** Specify how you want the adapter to connect to IBM i system at runtime environment for non keyed data queues such as FIFO and LIFO, as shown in the preceding figure. Poll queue path field is disabled for simple data queue polling. In the Deployment Properties area, specify how you want the adapter to get the user name and password at runtime environment.
 - To use a J2C authentication alias, select **Using an existing JAAS alias (recommended)** and type the name of the alias in **J2C Authentication Data**

Entry. You can 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.

- To use activation specification properties, select **Using security properties from the activation specification** and type the values in the **User name** and **Password** fields.
 - To administer the user name and password from other mechanism, click **Other**.
3. **For a keyed data queue** - For example, as shown in following figure, specify how you want the adapter to connect to IBM i system at runtime environment. 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. In the Deployment Properties area, specify how you want the adapter to get the user name and password at run time.
- To use a J2C authentication alias, select **Using an existing JAAS alias (recommended)** and type the name of the alias in **J2C Authentication Data Entry**. You can 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.
 - To use activation specification properties, select **Using security properties from the activation specification** and type the values in the **User name** and **Password** fields.
 - To administer the user name and password from other mechanism, click **Other**.

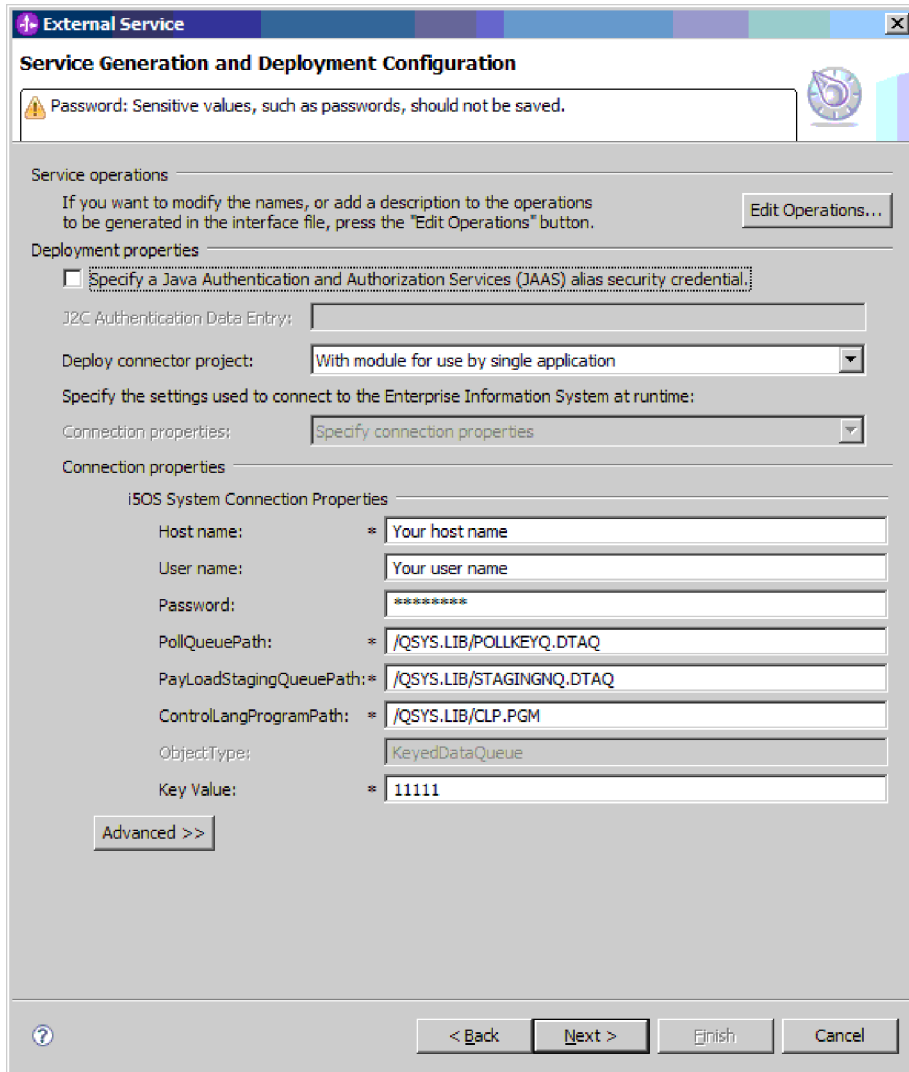


Figure 33. Service Generation and Deployment Configuration window

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 the **Deploy connector project** field, specify whether to include the adapter files in the module. Select 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 run time.
 - 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 settings**, select **Specify 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. 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.
10. **For Last-in first-out (LIFO) or First-in first-out (FIFO) simple data 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 Specify the Service Generation and Deployment Properties window.

Figure 34. Service generation and deployment configuration advanced window

- **Event polling configuration**
 - 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 try to connect to the server 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.
- **Event delivery configuration**
 - 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. In the **Number of connections for event delivery** field, specify the minimum and maximum number of connections to use to deliver events. Any value less than 1 is treated as 1 by the adapter.
- By selecting the **Additional 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 on 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 IBM Business Process Manager.
- d. In the **User name** field, enter your database user name.
- e. In the **Password** field, enter your password.
- f. In the **Database schema name** field, enter the name of the database used by event persistence processing.
- **Logging and tracing**
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
- **Processing methods for illegal XML characters**

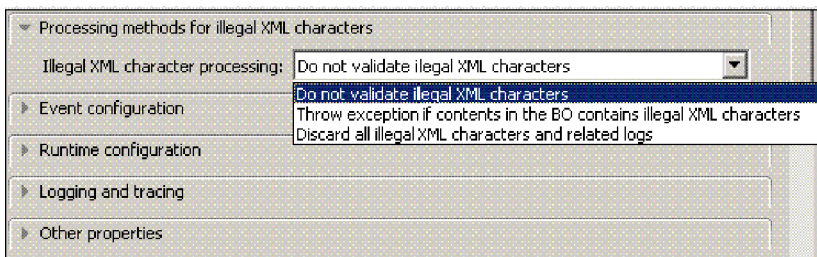


Figure 35. Illegal character processing window

From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.
11. **For a keyed data 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.

Figure 36. Service generation and deployment configuration window

For more details of the Advanced properties such as Event polling configuration, Event delivery configuration, Event persistence properties and Logging and tracing refer to the section in step 10.

- Select the calendar based scheduling option to create calendar based polling for inbound activities. You can schedule your business activities, when you create a new calendar in IBM Integration Designer. The option of working with the calendar based scheduling feature is only possible with IBM Integration Designer as the tooling environment. The following figure helps you to schedule a calendar polling option.

Figure 37. Polling based on calendar

You can either select a blank calendar or create a new calendar for a module or library. When you select a blank calendar, you will not be able to set pre-defined time intervals. You have to define your time intervals. When you create a calendar using a pre-defined template, you can define time intervals for each template.

- a. Click **New** to create a new calendar entry for a module or library.
 - You can choose an existing calendar, or create a new calendar instance.
 - Click **Browse** to select an existing calendar module. Or click **New** to create a module for the new calendar.
 - Click **Browse** to choose a folder for the calendar. (Optional).
 - Enter a name for the new calendar.
 - Click **Next** if you want to generate the calendar, through a predefined template. Or, click **Finish**, to create a non template calendar.
- b. Click **Browse** to select an existing calendar for a module or library. In the **Select a Business Calendar** screen you can search for all currently existing calendar files (*.cal) in the IBM Integration Designer workspace.
 - In the **Name** field, type the calendar name or click the calendar in the **Matching business calendars** screen. Click **OK** to open the external service wizard.
 - In the **WebSphere Integration workspace**, select the **Calendar** module, and browse **Integration logic->Calendars**, to view or modify the calendar schedules. You can modify the intervals and exceptions, or add new entries for these elements. For more information, refer to the information at <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/index.jsp?topic=/com.ibm.wbpm.wid.bpel.doc/topics/cbuscal.html>.

Note: You must deploy the Business Calendar module to the same IBM Business Process Manager or WebSphere Enterprise Service Bus instance, along with the inbound application. If you do not map these two connections to the same server instance, the inbound application using the business calendar will by default, poll as there is no calendar configured.

12. Click **Next**. The Specify the Location Properties window is displayed.

Results

The Specify the Location Properties window is displayed.

Setting location properties for artifacts:

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Before you begin

This task is performed using the Service Location Properties window of the external service wizard.

About this task

To configure location properties for your module follow this procedure:

Procedure

1. In the Specify the 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 Select a Business Integration Project Type window, select **Create a module project** and click **Next**.
 - c. In the Create a 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 Integration Designer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
 - f. Click **Finish** to create the new module.
2. Specify the namespace you want to use for your business objects schema.
 - 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**.
3. 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).
4. In **Name**, accept the default import name or type a different name.

5. 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**.
6. In **Description**, type a descriptive comment about the module.
7. When you are finished setting properties, click **Finish**.

Note: If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The artifacts are generated.

Chapter 5. Changing interaction specification properties

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

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 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 IBM Integration Designer, expand the module name.
2. Expand **Assembly Diagram** and double-click the interface.
3. Click the interface in the assembly editor. The module properties are displayed.
4. Click the **Properties** tab. You can also right-click the interface in the assembly 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 IBM Integration Designer, the integrated test environment features runtime support for IBM Business Process Manager 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 IBM Integration Designer, 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 IBM Business Process Manager or WebSphere Enterprise Service Bus as EAR files using the administrative console or command-line tools.

Deploying the module for testing

In IBM Integration Designer, 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.

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 IBM Integration Designer. 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 IBM Integration Designer, 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.
 - 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 > Java**.
 - 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 IBM Integration Designer, 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, generate and wire a *target component* to which the adapter sends the 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 servers settings.
 - d. Click **Finish** to publish the server.

2. Add the module to the server.
 - a. Switch to the servers view. In IBM Integration Designer, select **Windows > Show View > Servers**.
 - a. Start the server. In the **Servers** tab in the lower-right pane of the IBM Integration Designer screen, right-click the server, and then select **Start**.
3. When the server status is *Started*, right-click 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 IBM Integration Designer integration test client.

Before you begin

You need to add the module to the server first.

About this task

Testing a module is 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 about testing a module using the test client, see the *Testing modules and components* topic in the IBM Integration Designer 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 IBM Business Process Manager or WebSphere Enterprise Service Bus in a production environment is a two-step process. First, you export the module in IBM Integration Designer as an enterprise archive (EAR) file. Second, you deploy the EAR file using the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

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 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 set **Deploy connector project** to **On server for use by multiple adapters** in the Specify the Service Generation and Deployment Properties 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 run time.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **Install RAR**.

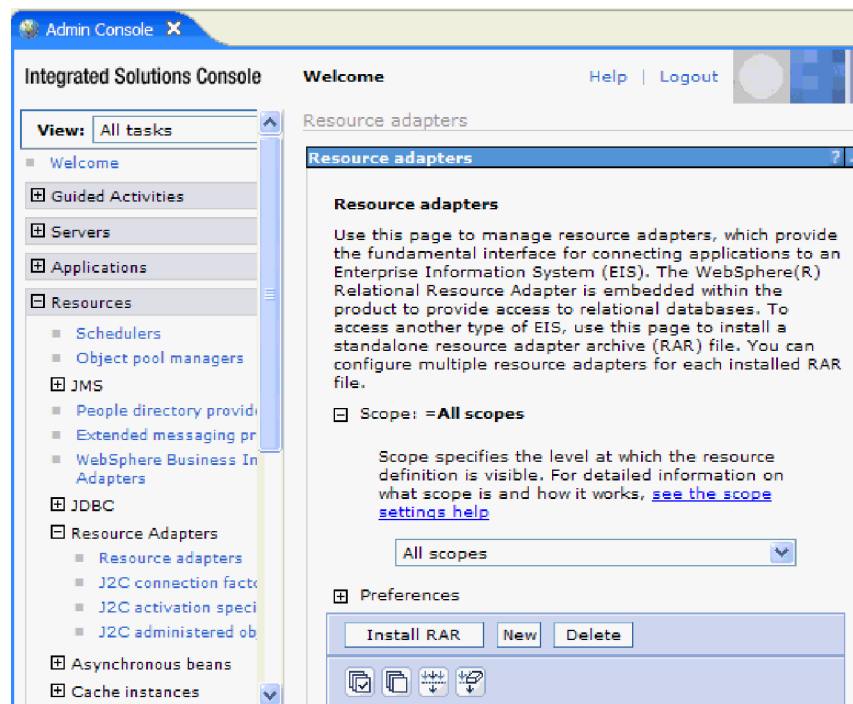


Figure 38. The Install RAR button on the Resource adapters page

6. 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:
`IID_installation_directory/ResourceAdapters/adapter_name/adapter.rar`
7. Click **Next**.
8. Optional: In the Resource adapters page, change the name of the adapter and add a description.
9. Click **OK**.
10. 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 IBM Integration Designer, 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 IBM Business Process Manager 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 IBM Integration Designer 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. To export the source files, select the **Export source files** check box. 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 existing file**.
8. Click **Finish**.

Results

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

What to do next

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

About this task

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > New Application > New Enterprise Application**.



Figure 39. Preparing for the application installation window

5. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by "App."

6. Optional: If you are deploying to a clustered environment, complete the following steps.
 - a. On the **Step 2: Map modules to servers** window, select the module and click **Next**.
 - b. Select the name of the server cluster.
 - c. Click **Apply**.
7. Click **Next**. In the Summary page, verify the settings and click **Finish**.
8. Optional: If you are using an authentication alias, complete the following steps:
 - a. Expand **Security** and select **Business Integration Security**.
 - b. Select the authentication alias that you want to configure. You must have administrator or operator rights to change the authentication alias configurations.
 - c. Optional: If it is not already specified, type the **User name**.
 - d. If it is not already specified, type the **Password**.
 - e. If it is not already specified, 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 the 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all IBM WebSphere Adapters.

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change. The **Configuration** page is displayed.

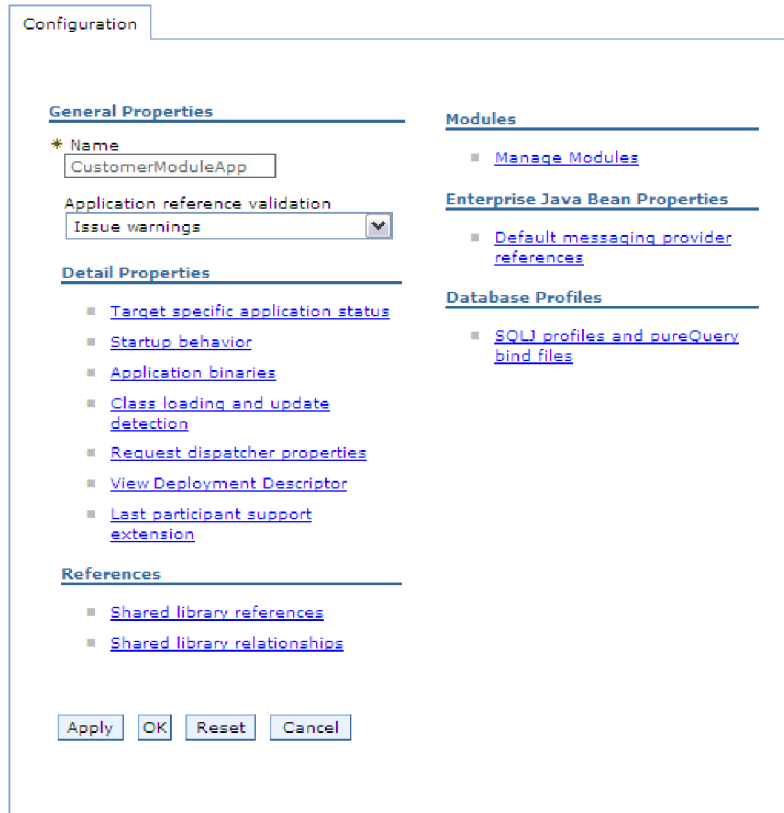


Figure 40. The Manage Modules selection in the Configuration tab

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

Note: See “Resource adapter properties” on page 128 for more information about these properties.

- a. Click the name of the property. The **Configuration** page for the selected property is displayed.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. In the Messages area, click **Save**.

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 IBM Business Process Manager 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. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. In the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

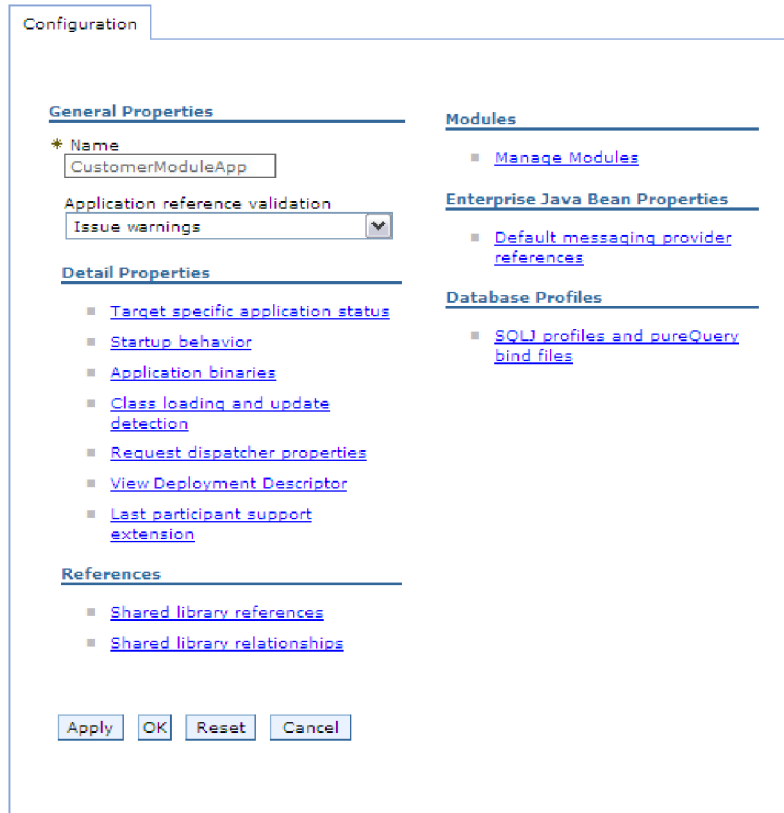


Figure 41. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for IBM i**.
8. In the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
10. Click the name of the connection factory associated with your adapter module.
11. In the **Additional Properties** list, click **Custom properties**.
Custom properties are those J2C connection factory properties that are unique to IBM WebSphere Adapter for IBM i. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
12. For each property you want to change, perform the following steps.
 - Note:** See “Managed connection factory properties” on page 131 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**.
13. In the Messages area, click **Save**.

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 IBM Business Process Manager 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. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

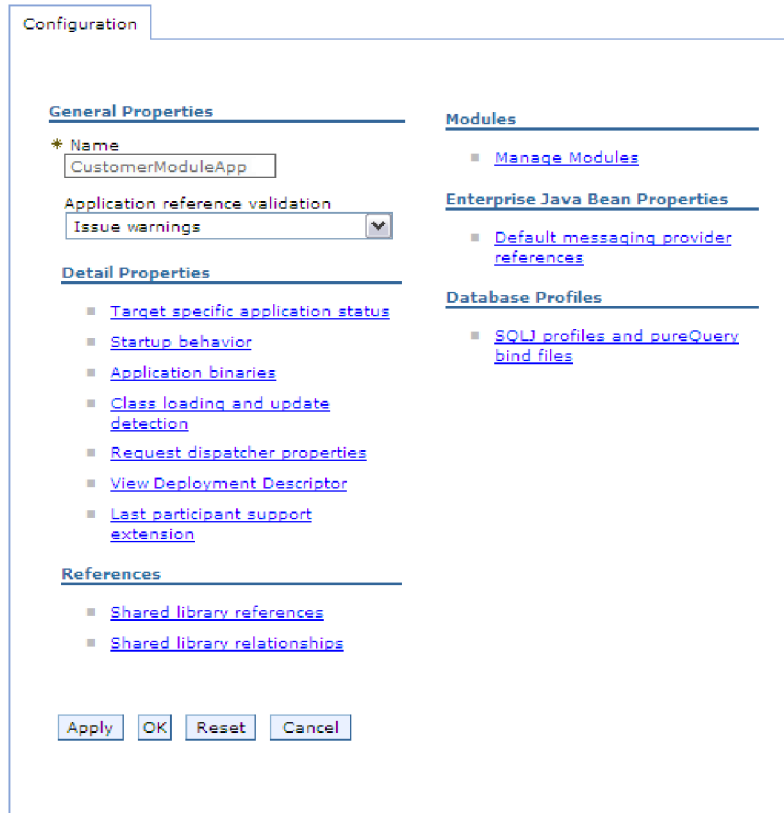


Figure 42. The Manage Modules selection in the Configuration tab

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

Note: See “Activation specification properties” on page 140 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**.
13. In the Messages area, click **Save**.

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, use the administrative console of the runtime environment. Provide the general information about the adapter and then set the resource adapter properties (which are used for general adapter operation). If the adapter is used for outbound operations, create a connection factory and then set the 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 IBM Business Process Manager 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all IBM WebSphere Adapters.

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for IBM i**.
6. In the **Additional Properties** list, click **Custom properties**.
7. For each property you want to change, perform the following steps.

Note: See “Resource adapter properties” on page 128 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**.
8. In the Messages area, click **Save**.

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 IBM Business Process Manager 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 IBM Business Process Manager 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. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for IBM i**.
6. In the **Additional Properties** list, click **J2C connection factories**.
7. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

Note: If you have selected **Specify connection properties** when you use 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 can type AdapterCF.
- c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterCF.
- d. Optional: Select an authentication alias from the **Component-managed authentication alias** list.
- e. Click **OK**.
- f. In the Messages area, click **Save**.

The newly created connection factory is displayed.

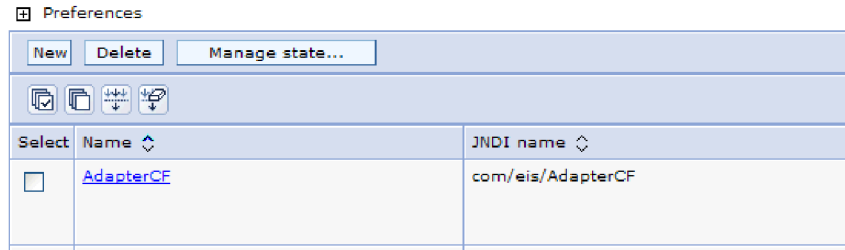


Figure 43. User-defined connection factories for use with the resource adapter

8. In the list of connection factories, click the one you want to use.

9. In the **Additional Properties** list, click **Custom properties**.

Custom properties are those J2C connection factory properties that are unique to WebSphere 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 131 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. After you have finished setting properties, click **Apply**.

12. In the Messages area, click **Save**.

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 IBM Business Process Manager 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 IBM Business Process Manager 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. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for IBM i**.
6. In the **Additional Properties** list, click **J2C activation specifications**.
7. If you are going to use an existing activation specification, skip ahead to select from an existing list of activation specifications.

Note: If you have selected **Use predefined connection properties** when you use the external service wizard to configure the adapter module, you must 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 can type AdapterAS.
 - c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterAS.
 - d. Optional: 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.
8. In the list of activation specifications, click the one you want to use.
 9. In the Additional Properties list, click **J2C activation specification custom properties**.
 10. For each property you want to set, perform the following steps.

Note: See “Activation specification properties” on page 140 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. After you have finished setting properties, click **Apply**.
 12. In the Messages area, click **Save**.

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. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to start. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. 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. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to stop. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. Click **Stop**.

Results

The status of the application changes to Stopped, and a message stating that the application has stopped is displayed 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 IBM WebSphere 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Before you 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 IBM Business Process Manager or WebSphere Enterprise Service Bus website:

<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 to gather the 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, see “Enabling tracing with the Common Event Infrastructure” on page 113.

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.

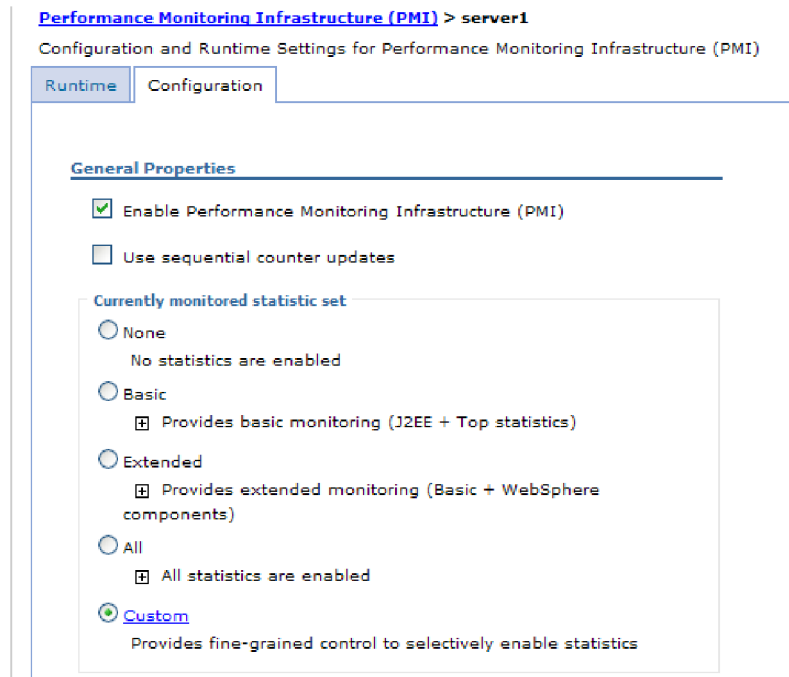


Figure 44. Enabling Performance Monitoring Infrastructure

- 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.

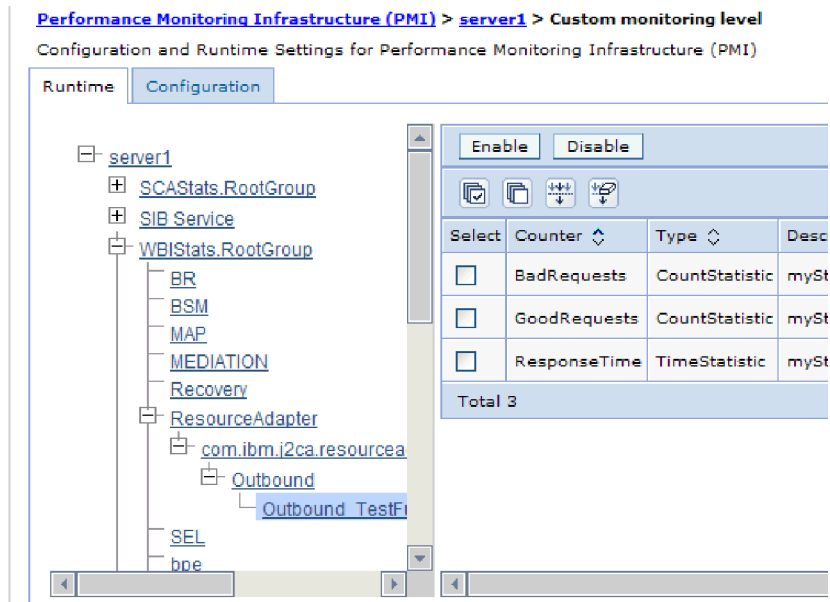


Figure 45. Runtime tab used for configuring PMI

- e. Click **WBISStats.RootGroup**. This is a PMI sub module for data collected in the root group. This example uses the name WBISStats 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

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.

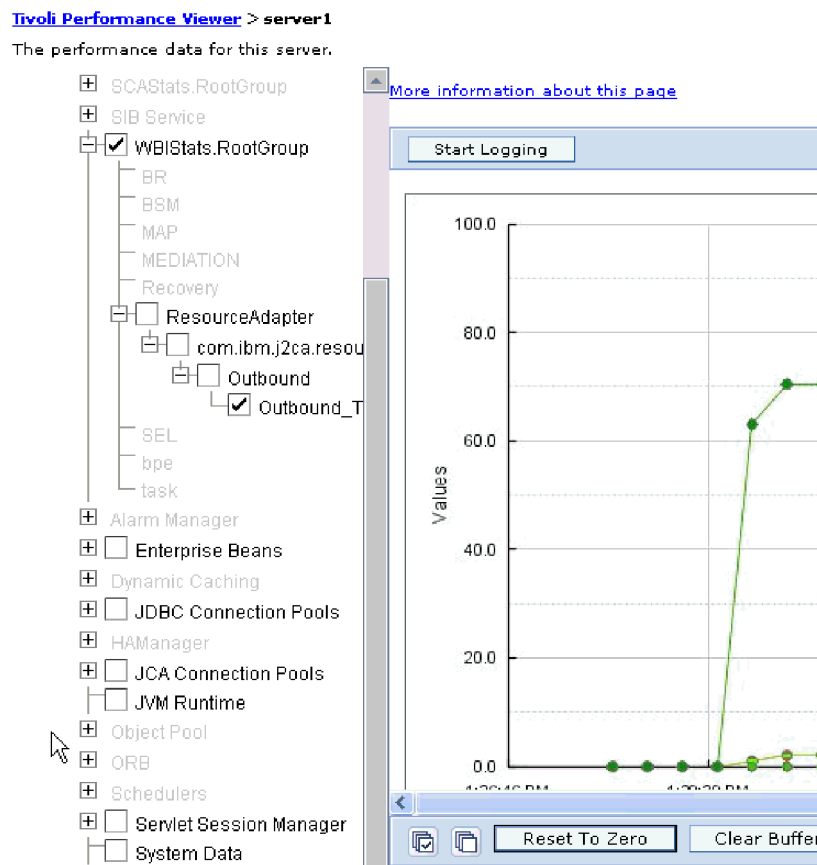


Figure 46. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure

The adapter can use the Common Event Infrastructure (CEI), 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

Use this procedure to report CEI entries in the trace log file by using the Common Base Event Browser within the administrative console.

Procedure

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. From 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, select one of the following options:
 - 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.

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.

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 IBM 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.

Note: For more information about monitoring on a IBM Process Server, including service components and event points, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.admin.doc/topics/welcome_wps_mon.html.

You can change the log configuration statically or dynamically. Static configuration takes effect when you start or restart the application server. Dynamic or run time 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 child logs, which recursively propagate the change to their child log, 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, select **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 WebSphere 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 IBM 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 IBM Process Server is written to the `SystemOut.log` and `trace.log` files.

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.

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 IBM Business Process Manager 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 IBM Business Process Manager or WebSphere Enterprise Service Bus documentation.

Avoiding stale connection problems in the IBM i adapter

Connection related problems can be resolved in WebSphere Adapter for IBM i using two new properties, **connectionRetryLimit** and **connectionRetryInterval** defined in the Managed connection factory of the IBM i adapter.

Before you begin

The two properties are used to provide two features during outbound communication of the adapter and are optional by default.

About this task

In the service generation and deployment configuration window explained in the “Setting deployment properties for runtime environment” on page 78 topic for the appropriate interface,

Procedure

1. Click on the **Advanced -> Additional connection configuration**
2. Set the **Maximum number of retries in case of system connection failure** to the appropriate positive integer
 - if the property value is set to 0, then adapter does not perform any EIS connection validation and executes the outbound operation. If the EIS connection is invalid, the outbound operation fails. Though the subsequent requests get executed successfully provided the IBM i system is functional, the current request fails.
 - if the property value is set to greater than 0, then during each request the adapter validates that the EIS connection is active/alive .
 - If connection is valid then operation is completed. if connection is invalid, the adapter invalidates the current managed connection so that a new managed connection is created (new physical connection).
 - If the connection is created successfully, the outbound operation is completed otherwise a `ResourceException` error is thrown.
3. Set the **Time interval between retries if connection fails (milliseconds)** field with the appropriate integer to indicate time in milliseconds in between retries. This property is enabled only when the **connectionRetryLimit** property has a value greater than 0.

Results

The two new features take care of connections which are timed out or become stale after the EIS restarts. Even though this option solves most of connection related problems, the adapter is not guaranteed to work 100% error free for connection problems.

Known error conditions for the Data queues and Call IBM i programs

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 program (PGM) that is not existing
- An attempt is made to run an PGM when the connection to the IBM i host is broken
- When not all the "in" or "input" attributes are populated with the 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 PGM

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 website

The WebSphere Adapters software support website at http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters, including:

- Flashes (alerts about the product)
- Technical information including the product information center, manuals, IBM Redbooks®, and whitepapers

- Educational offerings
- Technotes

Recommended fixes

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

Technotes

Technotes provide the most current documentation about WebSphere 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

WebSphere 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/>.

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 Data queue operations

Table 5 lists the properties of a business object attribute for the Data queue operations and describes their interpretation and settings.

Table 5. Attribute properties

Property name	Description
maxLength	Integer. This property indicates maximum length for a message added to an IBM i data queue.
messageContent	String
messageKey	String Indicates the key for a message as a string on an IBM i data queue. Note: This property is applicable for keyed data queues.
name	This property represents the unique name of the attribute.
messageKeyAsBinary	Indicates the key for a message as hexBinary on an IBM i data queue. Note: This property is applicable for keyed data queues.
messageContentAsBinary	The message content as hexBinary.

Business object attributes for PGM operations

Table 6 on page 122 lists the properties of a business object attribute and describes their interpretation and settings for running programs on an IBM i system.

Table 6. PGM business object attribute properties

Property name	Description	Value
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 7 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.	NA
passDirection	Determines whether an attribute is of in or out or inout. If a property is set as out or inout the adapter populates the result in that property. If a property is set as in, the adapter does not populate it in response.	String
totalDigits	The total number of digits in the field ('length' in PCML)	Integer
fractionDigits	The number of fractional digits ('precision' in PCML)	Integer
length	The length of the attribute. Applicable only for String types.	String
totalBytes	The field length in bytes ('length' in PCML).	String
count	The total number of elements in the array ('count' in PCML). This property is applicable only for array types.	Integer
count	The total number of elements in the array ('count' in PCML). This property is applicable only for array types.	String

In PCML, when you make a dynamic reference within a different structure, use an absolute path and provide the value in the format `programname.structurename.variablename`. When you work in the same structure, you can also use a relative path and provide the value for the `variablename`.

Table 7 lists the XPCML parameter type and the corresponding business object attribute type.

Table 7. XPCML parameter type and business object attribute type list

XPCML Parameter name	Description
shortParm	short
intParm	int
longParm	long
hexBinaryParm	hexBinary
floatParm	float
doubleParm	double
packedDecimalParm	decimal
stringParm	string
unsignedShortParm	unsignedShort

Table 7. XPCML parameter type and business object attribute type list (continued)

XPCML Parameter name	Description
unsignedIntParm	unsignedInt
zonedDecimalParm	decimal
struct	Defines a named structure that you can specify as an argument to a program or as a field within another named structure
arrayOfStruct	Defines an array of structs
structParm	type=struct and denotes reference to the struct type definition of the specified parameter
arrayOfStringParm	type=string and 'count'=number of elements in the array
arrayOfPackedDecimalParm	type=decimal and 'count'=number of elements in the array
arrayOfZonedDecimalParm	type=decimal and 'count'=number of elements in the array
arrayOfUnsignedShortParm	type=unsignedShort and 'count'=number of elements in the array
arrayOfUnsignedIntParm	type=unsignedShort and 'count'=number of elements in
arrayOfFloatParm	type=float and 'count'=number of elements in the array
arrayOfDoubleParm	type=double and 'count'=number of elements in the array
arrayOfHexBinaryParm	type=hexBinary and 'count'=number of elements in the array
arrayOfStructParm	type=struct and denotes reference to the struct type definition of the specified parameter and 'count'=number of elements in the array
arrayOfLongParm	type=long and 'count'=number of elements in the array
arrayOfShortParm	type=short and 'count'=number of elements in the array
arrayOfIntParm	type=int and 'count'=number of elements in the array

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/iadthelp/v7r1/index.jsp?topic=/com.ibm.etools.iseries.pgmmd.doc/c0925075211.htm>

Supported operations

Each business object is associated with an operation. 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 8. Supported operations

Operation	Description
CallPGM	The CallPGM operation invokes an RPG or COBOL or service program (SRVPGM) 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

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.

Naming convention for Data queue object type - The adapter names business objects according to the <PREFIX> <Queue name> naming convention, where the prefix is DTAQ (for queue 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.

Naming convention for IBM i program (RPG or Cobol) or service program (SRVPGM) object type - For program (RPG or Cobol) or SRVPGM, the adapter names business objects according to the <PREFIX> <PCML name> <Program name>, where the prefix is PGM. For Struct, the adapter names business objects according to the <PREFIX> <PCML name> <Struct name>, where the prefix is PGM. For example, the business object representing the QSYS.LIB/MYLIB.LIB/MSG1PGM1.PGM program is named PGMP1MSG1 by the adapter. In this example, for a PCML say P1.PCML and inside this PCML there are program tags MSG1, MSG2 and MSG3. The program MSG1 have the PGM name as QSYS.LIB/MYLIB.LIB/MSG1PGM1.PGM, so the business object name is PGMP1MSG1.

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 Integration Designer. For more details, refer to the Integration Designer documentation.

Fault business objects

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

The WBIFault business object contains information that is required to handle the fault.

- **IBMiMessageNotFoundFault**
The adapter generates this fault while attempting to get a message from the IBM i data queue using a specified key as input. The adapter is unable to get any message with the specified key.
- **IBMiObjectNotFoundFault**
The adapter generates this fault while attempting to run a PGM that does not exist or when the data queue with the given name is not found in the specified path.
- **IBMiInvalidObjectTypeFault**
The adapter generates this fault while attempting to get or put a message; from a simple data queue on the IBM i machine whereas message is from the keyed data queue.
- **IBMiInvalidRequestDataFault**

The adapter generates this fault when it detects a data format error in an attribute value for a business object. A data format error occurs when an invalid value is set for any of the parameters. For example, a null or empty message is sent to a data queue.

- IBMiSecurityFault

The adapter generates this fault when the operation fails due to security reasons. For example, an unauthorized user tries to process an operation on an object without the required permissions.

- IBMiOperationExecutionFault

The adapter generates this fault when the outbound operation returns a failure code or when there are exceptions or errors generated from an underlying PGM program.

Configuration properties

IBM WebSphere Adapter for IBM i has several categories of configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter, managed connection factory, and activation specification properties after you deploy the application to IBM Business Process Manager or WebSphere Enterprise Service Bus.

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	<p>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.</p> <p>Removing a default value from a required field on the external service wizard <i>will not change that default value</i>. When a required field contains no value at all, the external service wizard processes the field using its assigned default value, and that default value is displayed on the administrative console.</p> <p>Possible values are Yes and No.</p> <p>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,</p> <ul style="list-style-type: none"> • 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	<p>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.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>

Row	Explanation
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: <ul style="list-style-type: none"> • 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: <p>For Rational® Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case sensitive • Can be up to 40 characters in length. <p>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.</p>
Example	Provides sample property values, for example: <p>"If Language is set to JA (Japanese), code page number is set to 8000".</p>
Globalized	If a property is globalized, it has national language support, meaning that you can set the value in your national language. <p>Valid values are Yes and No.</p>
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file. <p>Valid values are Yes and No.</p>

Outbound configuration properties

IBM 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 IBM Business Process Manager or WebSphere Enterprise Service Bus using IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

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 125.

Table 9. 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 128	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 10. 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	No
Bidi supported	No

Password (Password)

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

Table 11. 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	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 12. Host name details

Required	Yes
----------	-----

Table 12. Host name details (continued)

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

Object type to discover

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

Table 13. 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: <ul style="list-style-type: none"> • Data queue • RPG via PCML • COBOL via PCML • SRVPGM via PCML • RPG Program • COBOL Program
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 14. 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 125.

Table 15. Resource adapter properties for the Adapter for IBM i

Name		Description
In the wizard	In the administrative console	
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 X strings instead of user data in the log and trace files.
(Not available)	enableHASupport	Do not change this property.
(Not available)	LogFileSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
(Not available)	TraceFileSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

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

Table 16. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 16. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the 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.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter.</p> <p>For example, when 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to ISESRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

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

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

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>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.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No

Table 17. Disguise user data as "XXX" in log and trace files details (continued)

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 Integration Designer assembly editor or after deployment through the IBM Business Process Manager 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 125.

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 18. Managed connection factory properties for Adapter for IBM i

Property name		Description
In the wizard	In the administrative console	
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 X strings instead of user data in the 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 19. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 19. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the 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.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter.</p> <p>For example, when 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to ISESRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

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

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

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>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.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No

Table 20. Disguise user data as "XXX" in log and trace files details (continued)

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 21. 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	No
Bidi supported	No

Password (Password)

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

Table 22. 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	No
Bidi supported	No

User name (UserName)

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

Table 23. 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	No
Bidi supported	No

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 IBM Integration Designer. 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 125.

Table 24. 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 25. 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	<ul style="list-style-type: none"> • 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 26. Object type to discover details

Required	Yes
Default	No default value
Property type	String

Table 26. Object type to discover details (continued)

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: <ul style="list-style-type: none"> • Data queue • RPG via PCML • COBOL via PCML • SRVPGM via PCML • RPG Program • COBOL Program
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 27. 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 IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

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 125.

Table 28. 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 137	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 29. 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	No
Bidi supported	No

Password (Password)

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

Table 30. 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	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 31. 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 31. Host name details (continued)

Globalized	No
Bidi supported	No

Object type to discover

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

Table 32. 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: <ul style="list-style-type: none"> • Data queue • RPG via PCML • COBOL via PCML • SRVPGM via PCML • RPG Program • COBOL Program
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 33. 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 125.

Table 34. Resource adapter properties for the Adapter for IBM i

Name		Description
In the wizard	In the administrative console	
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 X strings instead of user data in the log and trace files.
(Not available)	enableHASupport	Do not change this property.
(Not available)	LogFileSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
(Not available)	TraceFileSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

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

Table 35. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 35. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the 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.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter.</p> <p>For example, when 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to ISESRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

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

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

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>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.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No

Table 36. Disguise user data as "XXX" in log and trace files details (continued)

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 Integration Designer assembly editor or after deployment through the IBM Business Process Manager 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 125.

Table 37. Activation specification properties for Adapter for IBM i

Property name		Description
In the wizard	In the administrative console	
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.
Maximum number of retries in case of system connection failure	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
Poll quantity	PollQuantity	The number of events the adapter delivers to the export during each poll period.

Table 37. Activation specification properties for Adapter for IBM i (continued)

Property name		Description
In the wizard	In the administrative console	
Retry EIS connection on startup	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the IBM i system if it cannot connect at startup.
Time between retries in case of system connection failure (milliseconds)	RetryInterval	The length of time that the adapter waits between attempts to reestablish connection after an error during inbound operations.
Retry limit for failed events	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 149	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 150	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 153	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.
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 38. Business object namespace property characteristics

Required	No
Default	<code>http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi</code>
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: <code>http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi/Schema1Customer</code>
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 39. Data source JNDI name details

Required	No
Default	No default value
Property type	String

Table 39. Data source JNDI name details (continued)

Usage	<p>Use this property to specify the JNDI name of a data source in IBM Business Process Manager or WebSphere Enterprise Service Bus that specifies connection information for the target database.</p> <p>To improve the performance of inbound or outbound operations, specify the name of a data source that is enabled for prepared statement caching.</p> <p>If the user name and password properties are also set, they override the user name and password in the data source.</p> <p>The properties for connecting to the database are used in the following order:</p> <ol style="list-style-type: none"> 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. <p>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:</p> <ul style="list-style-type: none"> • Data source JNDI name <ul style="list-style-type: none"> – 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 <ul style="list-style-type: none"> – 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

Delivery type (DeliveryType)

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

Table 40. Delivery type details

Required	No
Possible values	ORDERED UNORDERED

Table 40. Delivery type details (continued)

Default	ORDERED
Property type	String
Usage	The following values are supported: <ul style="list-style-type: none"> • 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 41. 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 once-only event delivery (AssuredOnceDelivery)

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

Table 42. Ensure once-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.

Table 42. Ensure once-only event delivery details (continued)

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 43. 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 events related to the Customer and Order business objects only, 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 44. Retry limit for failed events details

Required	No
Possible values	Integers
Default	5
Property type	Integer
Usage	<p>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:</p> <p>Default If this property is not set, the adapter tries five additional times before marking the event as failed.</p> <p>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.</p> <p>> 0 For integers greater than zero, the adapter retries the specified number of times before marking the event as failed.</p> <p>< 0 For negative integers, the adapter does not retry failed events.</p>
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 45. Maximum 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.
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 46. 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.
Globalized	No
Bidi supported	No

Password (Password)

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

Table 47. 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	No
Bidi supported	No

Interval between polling periods (PollPeriod)

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

Table 48. Interval between polling periods details

Required	Yes
----------	-----

Table 48. Interval between polling periods details (continued)

Possible values	Integers greater than or equal to 0.
Default	2000
Unit of measure	Milliseconds
Property type	Integer
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 49. 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 might 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 50. 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	No
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 51. Retry EIS connection on startup details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>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:</p> <ul style="list-style-type: none">• Set the property to <code>False</code> 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 <code>Stopped</code>. After you resolve the connection problem, start the adapter manually.• Set the property to <code>True</code> 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 <code>RetryInterval</code> property to determine how frequently to retry and the value of the <code>RetryLimit</code> property to retry multiple times until that value is reached. The administrative console shows the application status as <code>Started</code>.
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 reestablish a connection.

Table 52. Retry interval details

Required	Yes
Default	2000
Unit of measure	Milliseconds
Property type	Integer
Usage	<p>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.</p>
Globalized	No
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 53. Stop the adapter when an error is encountered while polling details

Required	No
Possible values	True False
Default	False
Property type	Boolean
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 54. 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	No
Bidi supported	No

Polling queue

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

Table 55. 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 56. Host name details

Required	Yes
Default	No default value

Table 56. Host name details (continued)

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

Object type (ObjectType)

This property determines the object type that the adapter polls.

Table 57. 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 58. 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 59. Payload staging queue path details

Required	Yes
Default	NoneNo 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 60. Event recovery data source (JNDI) name details

Required	Yes, if the AssuredOnceDelivery property is set to True
Default	No default value
Property type	String
Usage	This value is case-sensitive. The external service wizard does not create the data source on IBM Business Process Manager 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 61. 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 62. 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 63. Automatically create event table details

Required	No
Possible values	True False
Default	True
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 64. 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 65. 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 66. Failed events queue path details

Required	Yes
Default	NoneNo default value
Property type	String
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 67. Failed event archiving control language program path details

Required	Yes
Default	NoneNo 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 68. Polling control language program path details

Required	Yes
Default	NoneNo 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 right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Globalization

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, IBM Integration Designer, IBM Business Process Manager, 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 encoding 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. IBM Business Process Manager 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

IBM Business Process Manager 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

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

Table 69. 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 IBM Business Process Manager 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">
<xsd:annotation xml:space="preserve">
<xsd:appinfo>
<appinfo source="http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi/metadata">
<ibmi:asi: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>
</xsd:appinfo>
</xsd: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 70. 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/v7r5mx/topic/com.ibm.wbpm.ref.doc/topics/welc_ref_msg_wbpm.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 WebSphere Adapter for IBM i.

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

- IBM Business Process Manager, version 7.5, information center, which includes IBM Business Process Manager, IBM WebSphere Enterprise Service Bus, and IBM Integration Designer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/index.jsp>.
- IBM Business Process Manager, version 7.0, information center, which includes IBM Business Process Manager, IBM WebSphere Enterprise Service Bus, and IBM Integration Designer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp>.
- WebSphere Adapters, version 6.2.x, information center: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp>.
- IBM WebSphere Adapters, version 7.5 installation on WebSphere Application Server, version 8.0 information: <http://www-01.ibm.com/support/docview.wss?rs=695&uid=swg27011040>.

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Video samples

To help you use WebSphere Adapters, sample videos are available to detail some of the scenarios.

You can find several Adapter video demos at this FTP location
<ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/>.

To view these video samples:

1. Download the zip file of the required sample from the FTP location.
2. Extract the downloaded zip file to your system.
3. From the extracted folder, double-click the HTML file.
4. The video sample will open in your default Web browser.

Note: The video has a built in audio narration. If you do not want to listen to the audio, you can lower its tab to the minimum, and slide the tab to an audio level, whenever you want the audio back.

The following demos are currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/>

- AFC: Scheduling Calendar based polling
- AFC: Overcoming an AFC version conflict situation
- Lotus® Domino®: Configuring notification rules for forms in database paths, and generating their events
- IBM i: Supporting CCSID in the RPG, COBOL, and service programs
- IBM i: Discovering the PCMLS in a local file system
- JDE: Retrieving data using multiple business functions in outbound processing
- Oracle: Integrating support for an Oracle AQ activity
- Oracle: Integrating an XML gateway through JMS in Oracle E-Business Suite
- PeopleSoft: Handling a RecordNotFound fault behavior
- PeopleSoft: Handling an event filtering
- SAP: Monitoring Tivoli ITCAM for a BPM Portfolio requirement
- SAP: Adding random numbers to the end of duplicate field names
- Siebel: Adding codepage properties to both EMD and runtime for an inbound Business Object scenario
- Siebel: Supporting more than one output argument for a Siebel Business Services Interface in an outbound scenario
- WAT: Creating a custom adapter using WebSphere Adapter Toolkit
- WAT: Creating a custom adapter outbound interface in WebSphere Adapter Toolkit
- WAT: Testing a module in WebSphere Adapter Toolkit

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](http://www.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

A

- Accessing messages using data
 - queues 5, 14
- activation specification properties
 - list of 140
 - setting in administrative console 103, 107
- Active-Passive 27
- adapter 1, 11, 35, 41, 42, 44, 49, 58, 72, 74, 86, 123, 124
 - configure 35
 - information resources 157
 - messages 157
 - package files 115
 - performance 110
 - project, create 39
 - related information 157
 - samples and tutorials 157
 - support and assistance 157
 - technotes 157
- adapter application
 - starting 109
 - stopping 109
- Adapter for IBM i module
 - exporting as EAR file 95
 - installing EAR file on server 96
 - starting 109
 - stopping 109
- adapter implementation
 - security 23
- alias, authentication 36
- application 1
- application server
 - application server 1
 - server, application 1
- Application-specific information 8
- artifacts 52, 57, 58, 65, 70, 78, 86
- ASI 8
- Assured once-only delivery property 17
- assured-once 11
- asynchronous 5, 14
- attribute properties 121
- authentication 1, 35
 - description 24
 - in the wizard 24
 - runtime environment 24
- authentication alias 35, 36
 - J2C 24

B

- batch 5, 14
- Batch Processing 27
- bidi 49
- bidirectional 49
- binary format 49
- BQPROC field 17
- BQTOTAL field 17
- broker 1
- broker, integration 1

- business faults 21, 124
- business object 6, 11, 35, 42, 44, 49, 52, 57, 58, 61, 64, 65, 70, 74, 78, 86
- business object information 121
- business object structure 18
- business object, predefine 38
- business objects 18
 - attributes 121
 - how to view 48, 73
 - naming conventions 124
- Business objects
 - supported operations 123
- business service 52, 65, 78

C

- Call IBM i program 4, 8, 13, 58
- Call IBM i programs 42, 44
- calling IBM i programs 1
- CallPGM operation 6, 10
- CEI (Common Event Infrastructure) 113
- CL 11
- CL (control language) 35
- CL commands 5, 14
- CL concepts 5, 14
- clustered environment
 - adapters version conflict 27
 - deployment 27
 - inbound process 27
 - inbound processes 27
 - load balancing 27
 - outbound process 27
 - outbound processes 28
- COBOL 1, 58, 123
- COBOL program 8
- Common Event Infrastructure (CEI) 113
- communication 1
 - compatibility matrix 2
 - component 1
- confidential data, disguising 23
- confidential tracing 23
- configuration overview 36
- configuration properties
 - inbound 135
- configuring
 - logging properties 115
 - Performance Monitoring Infrastructure (PMI) 110
- Configuring the business object 49
 - connection 42, 44, 58
- connector project 39
- control language 5, 11, 14, 78
- control language elements 5, 14
- custom properties
 - activation specification 103, 107
 - managed connection factory 101, 106
 - resource adapter 99, 105

D

- data 11, 124
- data object 1
- data queue 5, 11, 14, 49, 74, 78, 124
- Data queue 5, 14
- data queues 42, 44
- Data queues 6, 49
- data queues set up 72
- data source
 - overview 16
- data types 8
- database 11, 52, 65
- debugging
 - self-help resources 119
- deployment 52, 65, 78, 86, 96
 - production environment 94
 - test environment 91
- deployment environment 35, 91
- deployment options 24
- developerWorks 158
- discover 58
- disguising confidential data 23
- DTAQ 11

E

- EAR file
 - exporting 95
 - installing on server 96
- EIS 1, 47
- embedded adapter
 - activation specification properties, setting 103
 - considerations for using 26
 - managed connection factory properties, setting 101
 - resource adapter properties, setting 99
 - usage considerations 24
- embedded adapters
 - changing configuration properties 99
 - setting activation specification properties 103
 - setting managed (J2C) connection factory properties 101
 - setting resource adapter properties 99
- EMD 8, 78
- enableHASupport property 27
- enterprise information system 1
- Enterprise information systems 47
- enterprise metadata discovery 8
- enterprise service discovery 8, 58
- EP_CreateTable property
 - description 16
- error 42, 44, 124
- ESD 8, 58
- event 11
- event delivery 11, 144
- event persistence 11

- event polling 11
- event processing 12
- event recovery table, Inbound 16
- event store 11, 15
- EVNTDATA field 17
- EVNTID field 16
- EVNTSTAT field 16
- exceptions 124
- export file 35
- exporting module as EAR file 95
- external service connection
 - properties 127, 135
- external service wizard
 - connection properties 42, 44
 - start up 42, 44

F

- Failed events queue path, Event recovery 17
- fault queue 11
- faults
 - description 21, 124
- FFDC (first-failure data capture) 118
- FIFO 1, 5, 6, 11, 14, 49, 78
- files
 - SystemOut.log log file 117
 - trace.log trace file 117
- first-failure data capture (FFDC) 118
- first-in first-out 1, 5, 11, 14, 49
- First-in first-out 6, 14
- folder 42, 44
- foreign key 121
- format 5, 14

G

- GetQueue operation 6, 7

H

- HA Active-Active 27
- hardware and software requirements 2
- hardware requirements 2
- High Availability (HA)
 - clustered environments 157
- high-availability environment 27
 - Active-Active 27
 - Active-Passive 27
 - deployment 27
 - inbound processes 27
 - outbound processes 28
- host 42, 44
- host name 42, 44

I

- IBM Business Process Manager
 - information 157
- IBM Business Process Manager or WebSphere Enterprise Service Bus
 - deploying to 94
- IBM Business Process Manager, version 7.0, information 157
- IBM i adapter 41, 42, 44

- IBM i Data queue 11, 49
- IBM i programs 1
- IBM i server 11
- IBM Integration Designer
 - information 157
 - starting 38
 - test environment 91
- IBM WebSphere Adapter for IBM i
 - administering 99
- IBM WebSphere Adapter Toolkit 158
 - developerWorks resources 157
- IBM WebSphere Enterprise Service Bus
 - information 157
- identifier 5, 14
- implementation, Java 92
- import file 35
- inbound 1
 - configuration properties 135
- Inbound 11
- inbound processing 11, 35, 41, 71, 77
- installing EAR file 96
- Integrated Language Environment 1
- integration broker 1
- interaction specification properties
 - changing 89
- interaction specification property 134
- interface 1, 35, 52, 57, 65, 70, 78, 86

J

- J2C connection factory
 - see managed connection factory 131
- Java 2 security 23, 24
- Java implementation 92
- JNDI 52, 65, 78
- job 5, 14
- JTOpen APIs 3

K

- key 5, 11, 14, 49, 124
- Keyed 6, 15
- keyed data queue 1
- keyed Data queues 49
- keyed IBM i Data queue 49

L

- last-in first-out 1, 5, 14, 49
- Last-in first-out 6, 14
- Library list 9
- LIFO 1, 5, 6, 14, 49, 78
- load balancing 27
- log 11
- Log Analyzer 115
- log and trace
 - configure 115
- Log and Trace Analyzer, support for 20
- log and trace files 20
- log files
 - changing file name 117
 - disabling 115
 - enabling 115
 - level of detail 115
 - location 117
 - SystemOut.log 117

- logging 42, 44, 49, 52, 65
 - configuring properties with administrative console 115
- logging level 115
- Lotus Domino driver files 41

M

- managed (J2C) connection factory
 - properties
 - setting in administrative console 101, 106
- managed connection factory properties
 - details 131
- matrix, compatibility 2
- message 1, 5, 11, 14, 42, 44, 49, 124
- message content 49
- message key length 49
- message queue 5, 14
- messages, adapter 157
- metadata 58
- Metadata selection properties
 - business object 71, 77
 - deployment 71, 77
 - how to specify (inbound) 71, 77
 - metadata 71, 77
 - module 71, 77
 - namespace 71, 77
 - service definition 71, 77
- metadata tree 58
- migration
 - performing migration 30
- migration considerations 28
- module 1, 35, 47, 49, 52, 57, 64, 65, 70, 72, 74, 78, 86
 - adding to the server 92
 - configuring for deployment
 - overview 35
 - configuring inbound processing 72
 - deploy for testing 91
- module, creating 38
- monitoring performance 110
- multiple connection 144

N

- namespace 49, 57, 64, 70
- naming conventions for business objects 124
- non keyed Data queues 49
- non keyed IBM i Data queue 49

O

- object 42, 44, 47
 - object 1
- object types 58
- operation 8, 52, 61, 64, 65
- operations
 - arguments 10
 - ASI 8
 - Bound programs 8
 - business object 10
 - business object attributes 8
 - Call IBM i program 8
 - Calling IBM i program 8

- operations (*continued*)
 - Calling IBM i programs 10
 - Calling program 8
 - CallPGM 6, 10
 - COBOL 8, 10
 - code 8
 - coding 8
 - compilers 8
 - data types 8
 - design 8
 - EBCDIC 8
 - GetQueue 6, 7
 - i5/OS system 8
 - IBM i operating system 8
 - ILE 8
 - Integrated Language Environment 8
 - iSeries 8
 - iSeries box 8
 - module 4
 - multiple modules 8
 - object types 4
 - OS/400 8
 - parameter 8, 10
 - PCML 8
 - performance 8
 - PGM 8, 10
 - post IBM i command 10
 - pre IBM i command 10
 - program call 4, 13
 - program call markup language 8
 - PutQueue 7
 - report program generator 8
 - response 8
 - routines 8
 - RPG 8, 10
 - service program 4, 8, 10
 - Service program conditions 8
 - Service program constraints 8
 - services 8
 - SRVPGM 8, 10
- outbound 1
- outbound configuration properties 126
- Outbound operation 6
- Outbound operations supported by IBM i 6
- outbound processing 3, 35, 41
 - testing the module 93

P

- package files for adapters 116
- path 1, 5, 14, 42, 44
- PCML 1, 58, 61
- performance monitoring
 - infrastructure 110
 - configuring 110
 - performance statistics 112
- Performance Monitoring Infrastructure (PMI)
 - configuring 110
 - description 110
 - viewing performance statistics 112
- performance statistics 112
- permission 124
- persistence 11
- PGM 124

- PMI (Performance Monitoring Infrastructure)
 - configuring 110
 - viewing performance statistics 112
- polling 15
- post IBM i commands
 - Add Library List Entry 9
 - Calling IBM i programs 9
 - COBOL 9
 - Configuring Library list 9
 - EIS 9
 - enterprise information system 9
 - Library list example 9
 - PCML 9
 - pre IBM i commands 9
 - program 9
 - program call markup language 9
 - Remove Library List Entry 9
 - RPG 9
 - service program 9
 - SRVPGM 9
- problem determination
 - self-help resources 119
- process 41
- program 11, 58, 61
- program call markup language 1, 58, 61
- project 35, 52, 65, 78
- project interchange (PI) file
 - artifact 32
 - project interchange files 32
 - projects 32
 - updating without migrating 32
- properties
 - activation specification 103, 107
 - list of 140
 - configuration properties
 - inbound 135
 - outbound 126
 - external service connection 127, 135
 - inbound configuration 135
 - managed (J2C) connection
 - factory 101, 106
 - outbound configuration 126
 - resource adapter 99, 105
- properties information
 - guide 125
- PutQueue operation 7

Q

- query 47, 72
- queue 11

R

- RAR (resource adapter archive) file
 - description 94
 - installing on server 94
- recommended fixes 119
- Redbooks, IBM WebSphere
 - Adapters 157
- related products, information 157
- report program generator 58
- Report program generator 8
- Report Program Generator 1
- request 41

- requirements
 - hardware 2
 - software 2
- resource adapter archive (RAR) file
 - description 94
 - installing on server 94
- resource adapter properties
 - details 129, 138
 - setting in administrative console 99, 105
- response 41
- retrieve message 11
- Retry limit property 147
- roadmap
 - configuring the module 35
- RPG 1, 58, 61, 123
- runtime environment 41, 42, 44, 52, 65, 78
 - deploying EAR file 94

S

- samples 33, 158
- security 124
 - disguising sensitive data 23
 - user authentication 24
- security features
 - adapter 23
 - Java 2 security 23
- security, Java 2 24
- self-help resources 119
- sensitive data, disguising 23
- service 1, 41, 42, 44, 47, 52, 65, 78, 86
- Service 49
- service description 42, 44, 52, 64, 65, 78
- service export 41
- service import 41
- service interface queue 36
- service program 1, 58, 61, 123
- Service program 8
- service-oriented architecture 1
- services 49
- SOA 1
- software dependencies 41
- software requirements 2
- SRVPGM 1, 58, 123
- staging queue 11
- stand-alone adapter 107
 - considerations for using 26
 - managed connection factory
 - properties, setting 106
 - resource adapter properties,
 - setting 105
 - usage considerations 24
- stand-alone adapters
 - changing configuration
 - properties 105
 - setting activation specification
 - properties 107
 - setting managed (J2C) connection
 - factory properties 106
 - setting resource adapter
 - properties 105
 - starting adapter applications 109
 - stopping adapter applications 109
 - structure of business objects 18

support
 overview 115
 plug-in for IBM support assistant 119
 self-help resources 119
 web site 119
Supported operations 6, 123
synchronize 5, 14
SystemOut.log file 117

X

XID field 17

T

table
 business object structure 18
target component 91
technotes 2, 119
test environment 91
 adding module to 92
 deploying to 92
 testing modules 93
trace files
 changing file name 117
 disabling 115
 enabling 115
 level of detail 115
 location 117
 trace.log 117
tracing 11
 configuring properties with
 administrative console 115
tRFC protocol 16
troubleshooting
 overview 115
 self-help resources 119
tutorials 33, 158

U

UNORDERED 144

V

validation 42, 44
view
 business object structure 18

W

warning message 11
WebSphere Adapters, version 6.0,
 information 157
WebSphere Adapters, version 6.2.x,
 information 157
WebSphere Application Server
 environment variables 17
WebSphere Application Server
 environment variables, defining 39
WebSphere Application Server
 information 157
WebSphere Business Integration Adapters
 information 157
WebSphere Extended Deployment 27
wiring components 91



Printed in USA