WebSphere® Adapters

Version 6.0.2

Adapter for JD Edwards EnterpriseOne User Guide

Note

Before using this information, be sure to read the general information in "Notices" on page 119.

22December2006

This edition applies to version 6, release 0, modification 2 of WebSphere Adapter for JD Edwards EnterpriseOne (product number 5724-N41) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

Contents

Chapter 1. About this information

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

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

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

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

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

Chapter 2. Release notes

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

Release notes for this adapter can be found at the following Web site: Adapter for JD Edwards EnterpriseOne release notes

Chapter 3. Introduction to WebSphere Adapters

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

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

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

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

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

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

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

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

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

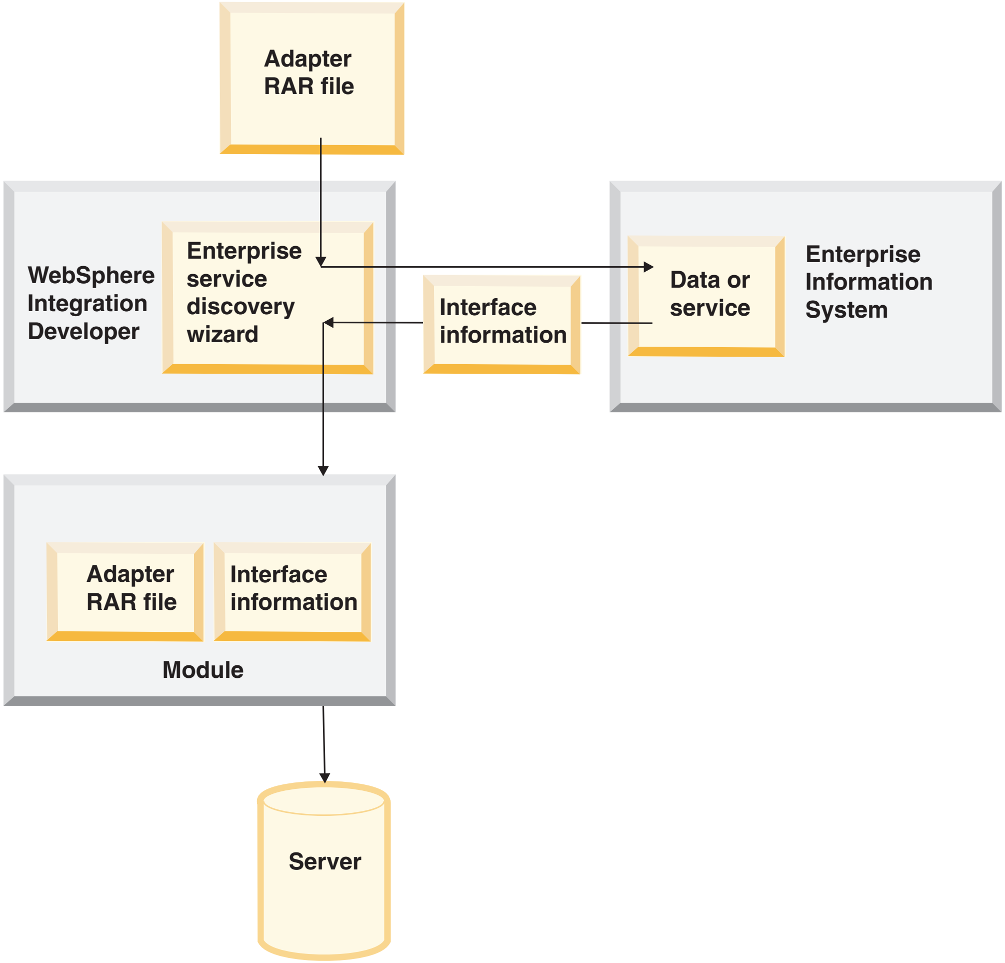


Figure 1. How a module is generated and deployed

Chapter 4. Introduction to the WebSphere Adapter for JD Edwards EnterpriseOne

WebSphere Adapter for JD Edwards EnterpriseOne acts as a component that is part of a system of components. Therefore, it is important that you become familiar with the various hardware and software requirements for installing, configuring, deploying, and running the adapter before you begin any of those tasks. In addition, all WebSphere Adapters comply with various government and industry standards, including accessibility standards and Internet protocol standards. Before beginning the adapter installation, you may want to become familiar with the accessibility features of the adapter. WebSphere Adapter for JD Edwards EnterpriseOne provides the means for clients (J2EE components) to send requests to the JD Edwards EnterpriseOne environment. For example, clients can update a customer record in a JD Edwards EnterpriseOne table or query data from that table. To learn more about how the adapter works, refer to the ″Technical overview of the Adapter for JD Edwards EnterpriseOne.″

Hardware and software requirements

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

Supported platforms for running the adapter installer

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

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

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

Standards compliance

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

Accessibility

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

Installation

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

Administration

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

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

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

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

Enterprise service discovery wizard

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

Keyboard navigation

This product uses standard Microsoft Windows® navigation keys.

IBM and accessibility

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

Internet Protocol Version 6.0

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

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

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

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

Technical overview of the Adapter for JD Edwards EnterpriseOne

WebSphere Adapter for JD Edwards EnterpriseOne enables bidirectional, real time integration between JD Edwards EnterpriseOne applications and requests received from any J2EE component. The adapter processes these outbound requests using one of two types of business objects: business functions (business object containers that can contain multiple business objects that can be processed as a single transaction) or XML Lists (a single business object that can return multiple records). You create business objects by using the enterprise service discovery wizard, a tool launched from WebSphere Integration Developer. The business objects generated by the enterprise service discovery wizard have predefined business object definitions. If you want to add or remove functionality from a generated business object, you can use the Business Object Editor, a tool launched from WebSphere Integration Developer, to changed the properties of the generated business object definition.

Outbound processing

WebSphere Adapter for JD Edwards EnterpriseOne provides outbound processing only. This means that various J2EE components can send requests to the adapter, and the adapter processes those requests by creating, retrieving, updating, or deleting data in the JD Edwards EnterpriseOne database.

When a J2EE component makes a request, the adapter processes the request using the JD Edwards EnterpriseOne Dynamic Java connector to invoke either a business function or an XML List. Business functions support the following types of updates: create, retrieve, update, delete, and execute, while XML Lists support only one kind of request: a ″request all.″ The following figure shows the flow of outbound requests from J2EE components to the JD Edwards EnterpriseOne server.

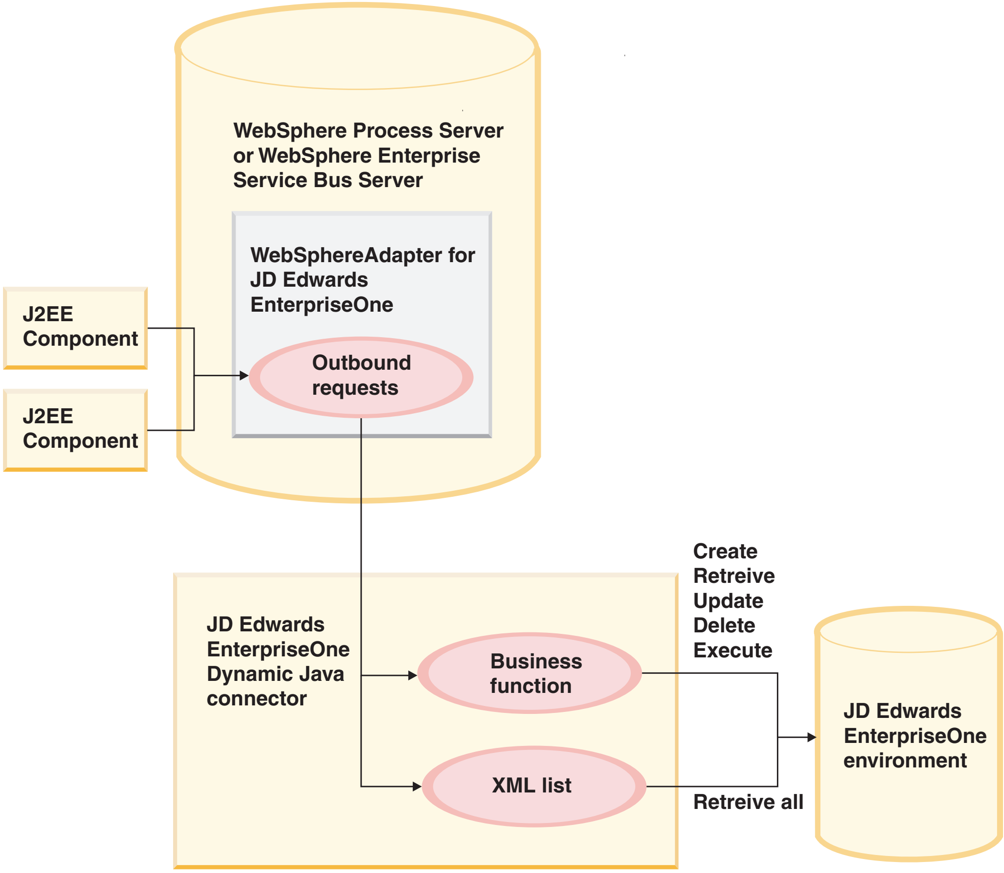


Figure 2. Flow of outbound requests from J2EE components to the JD Edwards EnterpriseOne environment

Business objects

WebSphere Adapter for JD Edwards EnterpriseOne supports two types of business objects: business functions and XML Lists. Business functions allow the adapter to process multiple business objects as a single transaction, while XML Lists allow the adapter to the retrieve multiple records from a single RetrieveAll transaction.

Business functions

Business functions are designed to support the execution of multiple JD Edwards EnterpriseOne business objects as a single transaction. Each business function definition contains references to the business objects inside the business function, and each business object contains a definition allowed by that business function. All business functions are represented by a business graph. The following figure illustrates the design of a business function.

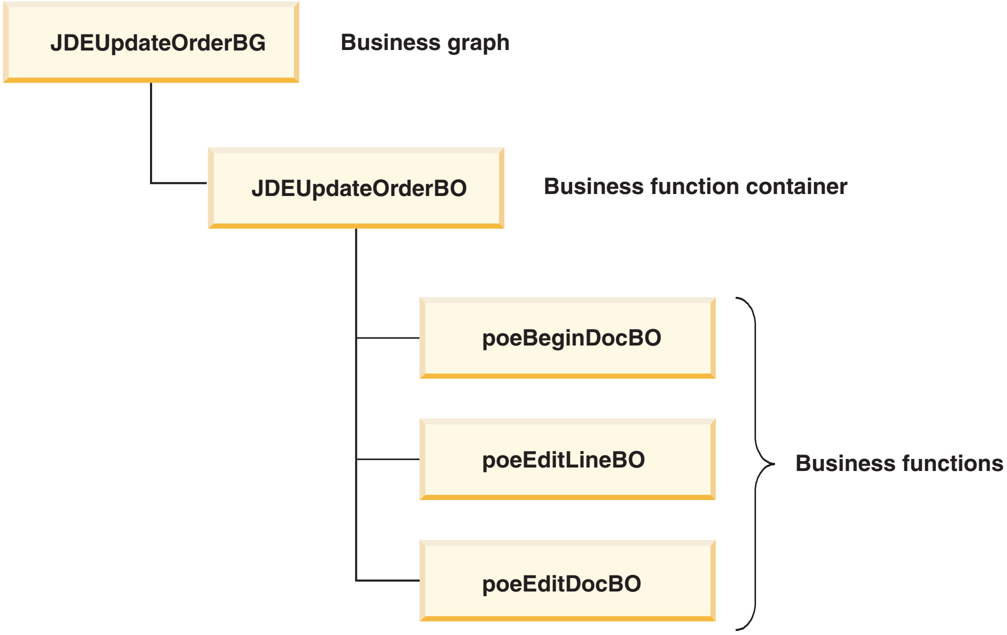


Figure 3. Design of a business function

Business functions support the following outbound operations:

- Create
- Retrieve
- Update
- Delete
- Execute

XML Lists

XML Lists are designed to support the retrieval of multiple records from a specific JD Edwards EnterpriseOne table or view. The adapter uses XML Lists to request and receive JD Edwards EnterpriseOne database information in chunks.

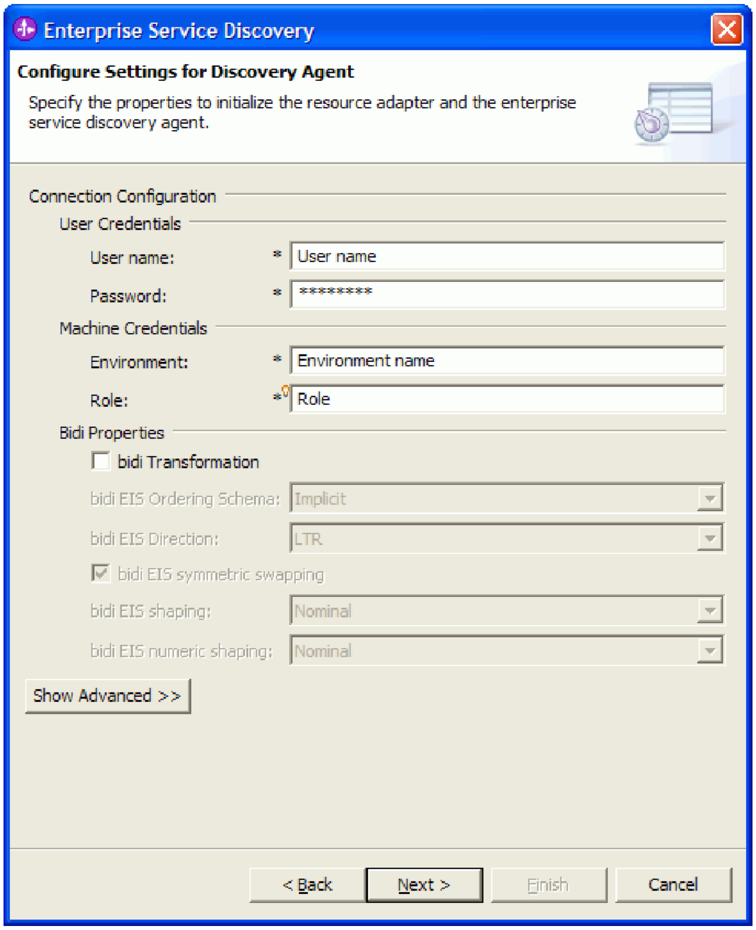
XML Lists support the following outbound operation:

- RetrieveAll

Enterprise service discovery

The enterprise service discovery wizard is a tool launched from WebSphere Integration Developer that you use to configure the adapter before it is deployed to WebSphere Process Server or WebSphere Enterprise Service Bus. The enterprise service discovery wizard connects to the JD Edwards EnterpriseOne environment, discovers services (based on search criteria you provide), then generates business objects.

Using the enterprise service discovery wizard, you establish an outbound connection to the JD Edwards EnterpriseOne environment for purposes of browsing the metadata repository. You specify connection information needed to access the server, such as the user name and password. The following figure illustrates the connection information required by the enterprise service discovery wizard.



Configure Settings for Discovery Agent

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

Connection Configuration

User Credentials

User name: * User name

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

Machine Credentials

Environment: * Environment name

Role: * Role

Bidi Properties

☐ bidi Transformation

bidi EIS Ordering Schema: Implicit

bidi EIS Direction: LTR

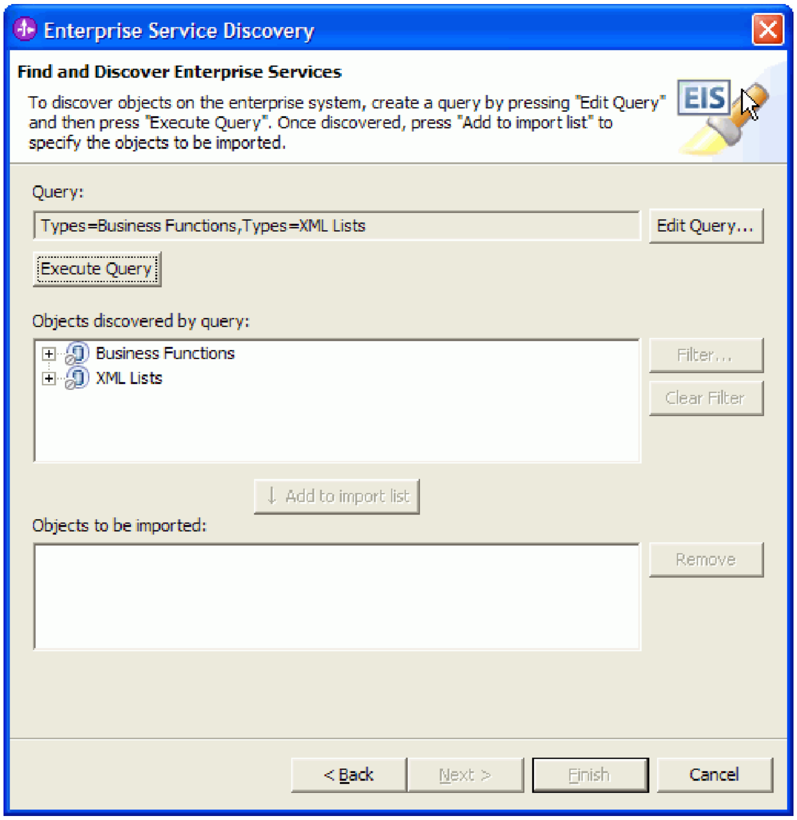
☑ bidi EIS symmetric swapping

bidi EIS shaping: Nominal

bidi EIS numeric shaping: Nominal

Figure 4. Enterprise service discovery wizard: Configure Settings for Discovery Agent window

After connecting to the JD Edwards EnterpriseOne environment, you search its database for the objects and services needed to generate the business object you want. You do this by executing a query from the Find and Discover Enterprise Services page. The following figure illustrates the Find and Discover Enterprise Services page of the enterprise services discovery wizard.


Enterprise Service Discovery

Find and Discover Enterprise Services

To discover objects on the enterprise system, create a query by pressing "Edit Query" and then press "Execute Query". Once discovered, press "Add to import list" to specify the objects to be imported.

Query: Types=Business Functions,Types=XML Lists

Objects discovered by query:

Figure 5. Enterprise service discovery wizard: Find and Discover Enterprise Services window

After executing the query, you can navigate to the object(s) or service(s) you need for generating the business object. For example, if you navigate to the GetEffectiveAddress business function (a container business object), the following objects are generated and displayed in the Business Integration view of WebSphere Integration Developer.

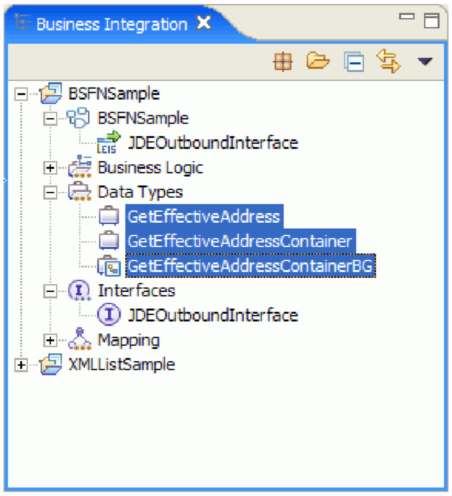


Figure 6. Example of objects generated by the enterprise service discovery wizard

Globalization and bidirectional transformation

The adapter is globalized to support single- and multi-byte character sets and deliver message text in the specified language. The adapter also performs bidirectional transformation, which refers to the task of processing data that contains both left-to-right (Hebrew or Arabic, for example) and right-to-left (a URL or file path, for example) semantic content within the same file.

Globalization

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

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

Bidirectional transformation

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

Bidirectional format

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

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

Table 1. Bidirectional format attributes

Letter position	Purpose	Values	Description	Default setting
1	Order schema	I or V	Implicit (Logical) or Visual	I
2	Direction	L R C D	Left-to-Right, Right-to-Left Contextual Left-to-Right Contextual Right-to-Left	L
3	Symmetric Swapping	Y or N	Symmetric Swapping is on or off	Y

Table 1. Bidirectional format attributes (continued)

Letter position	Purpose	Values	Description	Default setting
4	Shaping	S N I M F B	Text is shaped Text is not shaped Initial shaping Middle shaping Final shaping Isolated shaping	N
5	Numeric Shaping	H C N	Hindi Contextual Nominal	N

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

Using bidirectional properties

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

The following table describes four types of bidirectional properties.

Table 2. Bidirectional property types

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

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

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

Business object annotations

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

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

Property scope and lookup mechanism

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

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

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

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

Chapter 5. Planning for adapter implementation

Before you install WebSphere Adapter for JD Edwards EnterpriseOne, make sure you have a plan for implementing security. Also consider any performance implications of the installation, such as whether you want the adapter to run in a high-availability environment. Finally, review the ″Roadmap for installing, configuring, and deploying the adapter″ section to gain a high-level understanding of these tasks before performing them.

Security

The adapter is Java 2 security enabled and features user name and password authentication. In addition, you can configure additional security permissions by altering the application server's WAS.policy file and storing it in the meta-inf folder. For more details on configuring security details, see the security documentation for WebSphere Process Server or WebSphere Enterprise Service Bus.

WebSphere Adapters in clustered environments

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

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

High availability for outbound operations

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

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

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

Roadmap for installing, configuring, and deploying the adapter

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

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

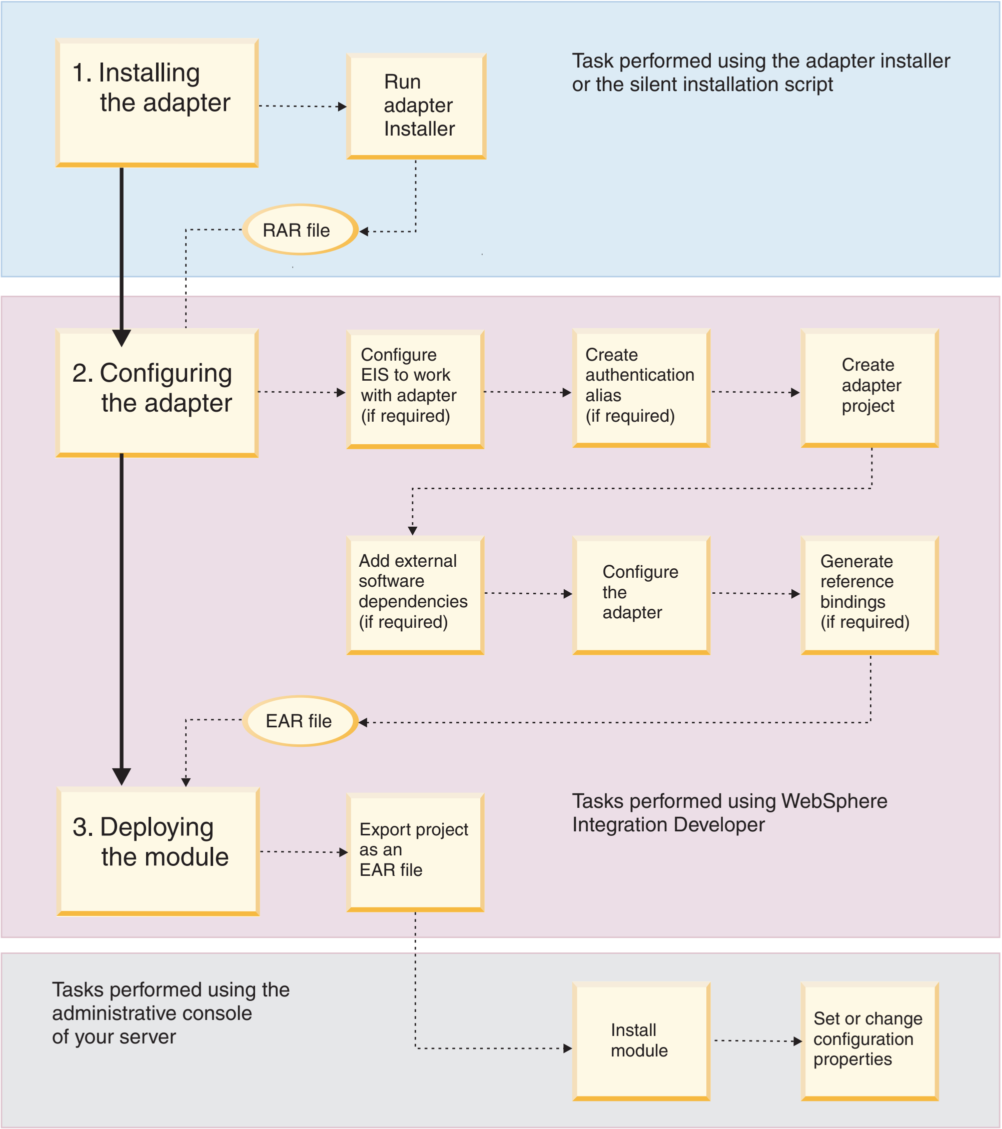

1. Installing the adapter
Run adapter Installer
Task performed using the adapter installer or the silent installation script
RAR file
2. Configuring the adapter
Configure EIS to work with adapter (if required)
Create authentication alias (if required)
Create adapter project
Add external software dependencies (if required)
Configure the adapter
Generate reference bindings (if required)
EAR file
3. Deploying the module
Export project as an EAR file
Tasks performed using WebSphere Integration Developer
Tasks performed using the administrative console of your server
Install module
Set or change configuration properties

Figure 7. Roadmap for installing, configuring, and deploying the adapter

1. **Installing the adapter**
   a. Use the installer (a graphical user interface) or a script that runs a silent installation. Either method installs a resource adapter archive (RAR) file on your workstation. You use this RAR file to configure the adapter.
2. **Configuring the adapter**
   a. (If required) Configure the enterprise information system (EIS) to work with your adapter. You perform this step from within the EIS application.
   b. (If required) Create an authentication alias to access the application.
   c. Create an adapter project in WebSphere Integration Developer (J2EE Perspective) by importing the adapter RAR file.
   d. (If required) Using WebSphere Integration Developer, add any external dependencies required by your adapter to the adapter project. These dependencies are also required as part of the bundled EAR file, which is exported when you deploy the adapter.

e. To configure the adapter, run the enterprise service discovery wizard from the Business Integration Perspective of WebSphere Integration Developer. The enterprise service discovery wizard generates business integration components and allows you to enter all the information necessary to configure the adapter for the first time. The output from the enterprise service discovery tool is saved to a business integration module project, which contains the business object, or objects, and the import or export file.

f. (If required) Use WebSphere Integration Developer to generate reference bindings for the component created by the enterprise service discovery wizard.

3. **Deploying the module**

   a. From the J2EE perspective in WebSphere Integration Developer, export a business integration module project as an EAR file.

   b. Install the module on WebSphere Process Server or WebSphere Enterprise Service Bus.

   c. (If required) In the server administrative console, set (or change) the following properties:

      - Resource adapter properties
      - Managed (J2C) connection factory properties
      - Activation specification properties for the EIS

Chapter 6. Installing the adapter

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

Installation prerequisites

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

Supported platforms for running the adapter installer

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

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

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

Additional jar files

If you are using WebSphere Integration Developer, version 6.0.1.1 or earlier, you must manually add three additional jar files to the classpath of the adapter project. For more information about how to do this, see "Adding jar files to WebSphere Integration Developer versions 6.0.1.1 and earlier" on page 115 in the Reference section.

Performing the installation

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

Before you begin

Review the installation prerequisites.

How to perform this task

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

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

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

Result

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

What to do next

Configure the adapter.

Uninstalling the adapter

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

About this task

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

Note: If you need to uninstall an adapter that is already deployed, refer to the ″Additional adapter-related information you might need″ section of "Related product information" on page 116.

Chapter 7. Configuring the adapter for deployment

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

Creating the authentication alias

To create the authentication alias on the server, use the administrative console. From the administrative console, configure the global security and set the password for the authentication alias, which is used to process outbound requests.

Before you begin

You must have access to the administrative console of the either WebSphere Process Server or WebSphere Enterprise Service Bus.

How to perform this task

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

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

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

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

Before you begin

Make sure you have installed Adapter for JD Edwards EnterpriseOne and that you have created an authentication alias.

About this task

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

To create an adapter project, use the following procedure.

How to perform this task

1. If WebSphere Integration Developer is not currently running, start it now.
   a. Click **Start** → **Programs** → **IBM WebSphere** → **Integration Developer V6.0.2** → **WebSphere Integration Developer V6.0.2**.
   b. If you are prompted to specify a workspace, accept the default value.

      The workspace is a directory where WebSphere Integration Developer stores your project.
   c. When the WebSphere Integration Developer window is displayed, close the Welcome page.
2. Switch to the J2EE perspective:
   a. Click **Window** → **Open Perspective** → **Other**.
   b. Click **J2EE**.

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

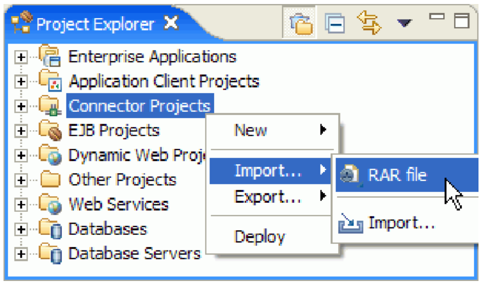
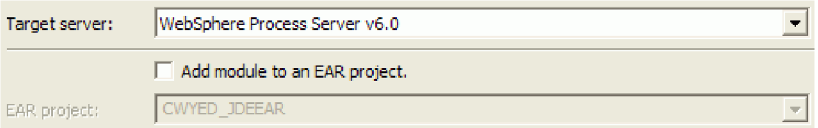


Figure 8. Importing the RAR file

4. From the Connector Import window, click **Browse** and navigate to the directory in which Adapter for JD Edwards EnterpriseOne was installed.
5. Click **CWYED_JDE.rar**.

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

   Note: If a project named CWYED_JDE.rar already exists in this workspace, the name in the Connector project field has a number appended to it (for example, CWYED_JDE1).
6. **Optional:** In the **Connector project** field, type another name for the project or accept the default value.
7. **Optional:** In the **Target server** field, select the server to which the adapter will be deployed or accept the default value.
8. Clear the **Add module to an EAR project** check box.



Target server: WebSphere Process Server v6.0

□ Add module to an EAR project.

EAR project: CWYED_JDEEAR

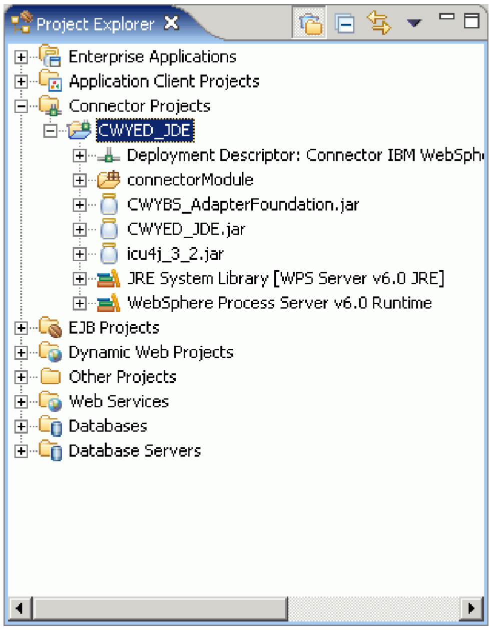
Figure 9. Clearing the Add module to an EAR project check box

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

9. Click **Finish**.

Result

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



Project Explorer

```
+ Enterprise Applications
+ Application Client Projects
- Connector Projects
    - CWYED_JDE
        + Deployment Descriptor: Connector IBM WebSph
        + connectorModule
        + CWYBS_AdapterFoundation.jar
        + CWYED_JDE.jar
        + icu4j_3_2.jar
        + JRE System Library [WPS Server v6.0 JRE]
        + WebSphere Process Server v6.0 Runtime
+ EJB Projects
+ Dynamic Web Projects
+ Other Projects
+ Web Services
+ Databases
+ Database Servers
```

Figure 10. The connector project in the Project Explorer window

What to do next

Add the required external dependencies to the project.

Adding external software dependencies

To add the required external dependency files to the adapter project, you first import the files to the connectorModule folder inside the adapter project, then you copy the files from the connectorModule folder of the adapter project to the project classpath.

Before you begin

Create the adapter project in WebSphere Integration Developer. Also, obtain the software dependency files. For information about obtaining the software dependency files, refer to the following JD Edwards EnterpriseOne document: *PeopleSoft EnterpriseOne Tools Connectors PeopleBook*.

About this task

The JD Edwards EnterpriseOne application requires that you add external software dependencies to the adapter project. These software dependencies enable the Adapter for JD Edwards EnterpriseOne to communicate with the JD Edwards EnterpriseOne environment. For a list of all software dependencies, refer to "External software dependencies" on page 109 in the Reference section.

How to perform this task

1. Import the JD Edwards EnterpriseOne software dependency files to the connectorModule folder inside the adapter project.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.
   d. In the ″File system″ window, click **Browse**, then navigate to the directory that contains the JD Edwards EnterpriseOne software dependencies and configuration files, then click **OK**.
   e. To select all of the files that are displayed in the right pan of the ″File system″ window, click **Select All** , then click **Finish**.
2. Import the following .jar files into the connectorModule folder: ffdcSupport.jar and aspectjrt.jar.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.
   d. In the ″File system″ window, click **Browse** to navigate to the following directory:*WebSphere_Integration_Developer_Installation_Directory*\runtimes\bi_v6\lib.

      Note: If you accepted the default installation directory when you installed WebSphere Integration Developer, the .jar files are located in the following directory: C:\Program Files\IBM\WebSphere\ID\6.0\runtimes\bi_v6\lib.
3. Add the software dependency files from the connectorModule folder to the adapter project classpath.
   a. Right-click the adapter project, then select **Properties**.
   b. In the Properties for CWYED_JDE window, select **Java Build Path** from the left pane.

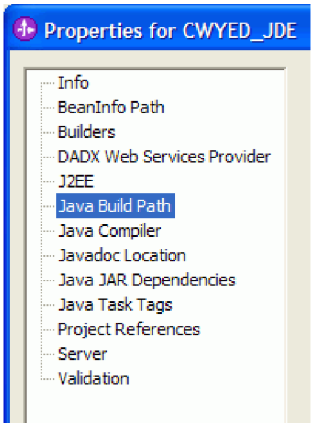


Figure 11. Selecting Java Build Path

c. In the Libraries page of the right pane, click **Add JARs**.

d. In the JAR Selection window, expand the adapter project folder (CWYED_JDE), then expand the connectorModule folder.

e. Highlight all of the JAR files listed under the connectorModule folder, then click **OK**.

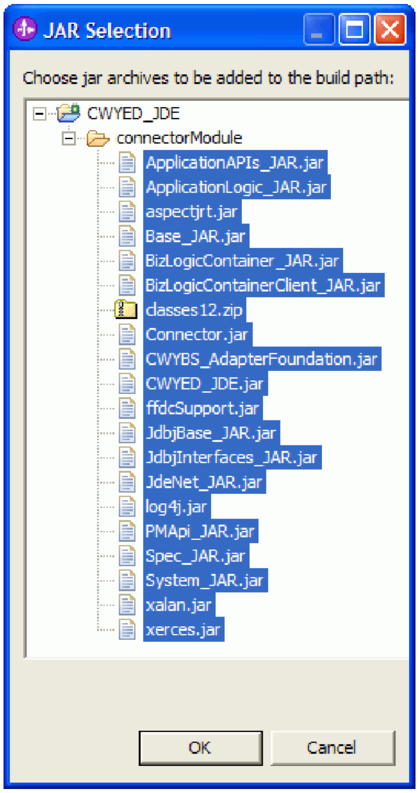
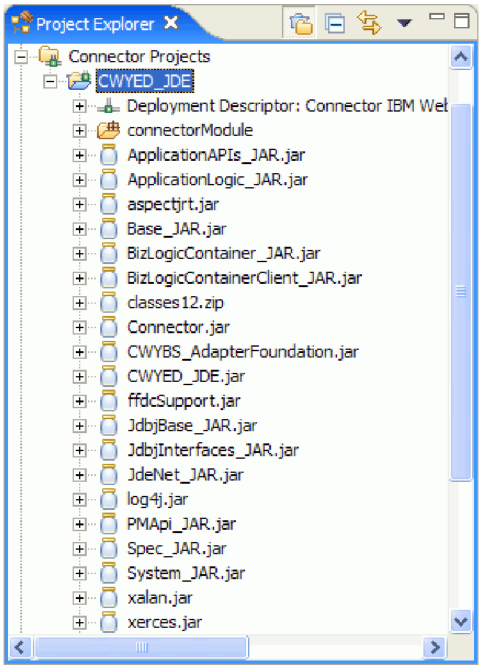


Figure 12. Selecting JAR files

f. In the Properties window, click **OK**.

Result

The external dependencies are displayed in the adapter project folder.



```
Connector Projects
  CWYED_JDE
    Deployment Descriptor: Connector IBM Wel
    connectorModule
    ApplicationAPIs_JAR.jar
    ApplicationLogic_JAR.jar
    aspectjrt.jar
    Base_JAR.jar
    BizLogicContainer_JAR.jar
    BizLogicContainerClient_JAR.jar
    classes12.zip
    Connector.jar
    CWYBS_AdapterFoundation.jar
    CWYED_JDE.jar
    ffdcSupport.jar
    JdbjBase_JAR.jar
    JdbjInterfaces_JAR.jar
    JdeNet_JAR.jar
    log4j.jar
    PMApi_JAR.jar
    Spec_JAR.jar
    System_JAR.jar
    xalan.jar
    xerces.jar
```

Figure 13. Adapter project folder showing the external dependency files

What to do next

Configure the adapter. The first step in configuring the adapter is to specify information about the JD Edwards EnterpriseOne environment, so that the enterprise service discovery wizard can connect to it.

Note: Depending on which version of JD Edwards EnterpriseOne you are using, you may be required to configure the following files before configuring the adapter:

- jdeinterop.ini
- jdelog.properties
- jdbj.ini
- tnsnames.ora

For instructions on configuring these files, refer to the following JD Edwards EnterpriseOne documents:

- *PeopleSoft EnterpriseOne Tools Connectors PeopleBook* (for jdeinterop.ini and jdelog.properties files)
- *PeopleSoft EnterpriseOne Tools HTML Server Installation PeopleBook* (for jdbj.ini and tnsnames.ora files)

Configuring the adapter using business functions

To configure WebSphere Adapter for JD Edwards EnterpriseOne using business functions, use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties for the enterprise service discovery, select business functions that are in the enterprise information system, and generate business object definitions and related artifacts for outbound processing.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the JD Edwards EnterpriseOne environment, specify such information as the user name and password you use to access the server as well as the environment name and role of the user.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the JD Edwards EnterpriseOne environment and discover its business objects and services.

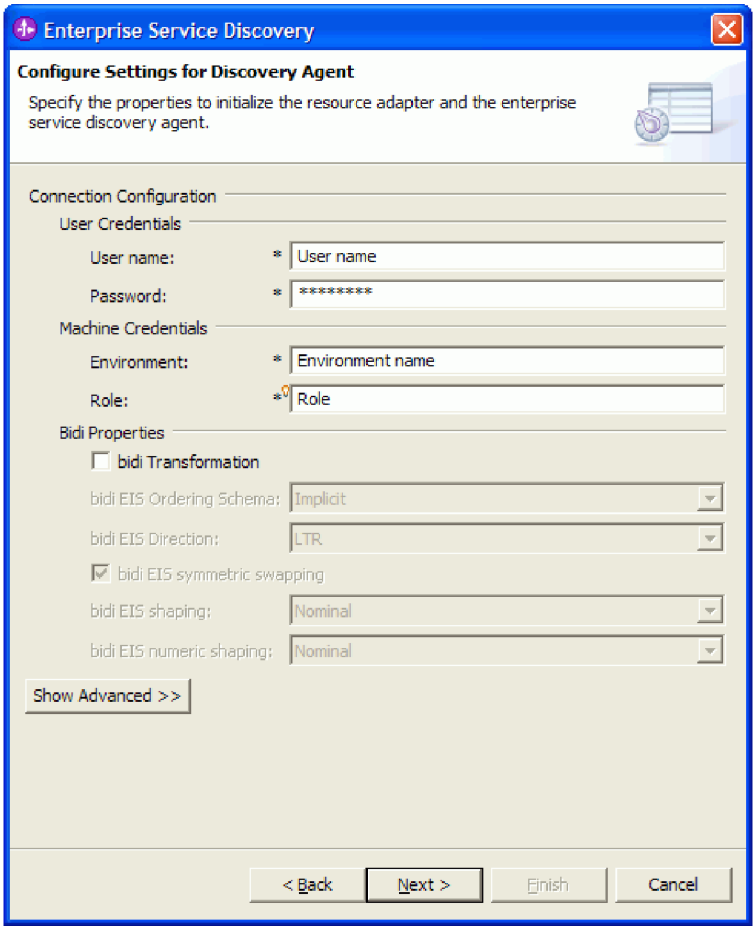
To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration perspective by clicking **Window** → **Open Perspective** → **Other**.
   b. In the Select Perspective window, select **Business Integration**, then click **OK**.
   c. Select **File** → **New** → **Enterprise Service Discovery**.
2. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for JD Edwards EnterpriseOne (version 6.0.2) from the CWYED_JDE Connector Project** , then click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the JD Edwards EnterpriseOne application you used the last time you ran the enterprise service discovery wizard.
3. Specify the configuration properties to initialize the discovery agent.

   Note: Properties marked with an asterisk (*) are required.
   a. Type the name and password you use to access the JD Edwards EnterpriseOne system.
   b. Type the environment name of your JD Edwards EnterpriseOne system.
   c. Type the role name you use to access the JD Edwards EnterpriseOne system.



Enterprise Service Discovery

Configure Settings for Discovery Agent

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

Connection Configuration

User Credentials

User name: * User name

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

Machine Credentials

Environment: * Environment name

Role: * Role

Bidi Properties

□ bidi Transformation

bidi EIS Ordering Schema: Implicit

bidi EIS Direction: LTR

☑ bidi EIS symmetric swapping

bidi EIS shaping: Nominal

bidi EIS numeric shaping: Nominal

Figure 14. The Configure Settings for Discovery Agent window

4. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See “Settings for controlling bidirectional transformation” on page 115 for more information about these properties.
5. **Optional:** To change the logging level, perform the following steps:
   a. At the bottom of the window, click **Show Advanced**.
   b. Set the **Logging Level**.

      In a test environment, select **FINEST**, which provides the highest level of logging. In a production environment, choose a level lower than **FINEST** to optimize the logging process.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the JD Edwards EnterpriseOne environment, using the information you provided, then the Find and Discover Enterprise Services window opens.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to discover business objects and services in the JD Edwards EnterpriseOne environment.

Selecting business objects and services

To select which business function to use for outbound processing, you provide information in the enterprise service discovery wizard.

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

About this task

Specify search criteria that the enterprise service discovery wizard uses to discover business functions on the JD Edwards EnterpriseOne server. The enterprise service discovery wizard returns a list of business functions.

To specify the search criteria and select an business functions, use the following procedure.

How to perform this task

1. To view the list of business functions available in the JD Edwards EnterpriseOne server, click **Execute Query** in the Find and Discover Enterprise Services window of the enterprise services discovery wizard.
2. In the Find and Discover Enterprise Services window of the enterprise services discovery wizard, click **Execute Query**. The results of the query are displayed in the ″Objects discovered by query″ field.
3. Navigate to the business object you want by expanding the **Business Functions** nodes.
4. Select the desired business function, then click **Add to import list**.
5. In the Configuration Parameters for *<business_object>* window, you can either keep the default name in the Business Object Name field or rename it, then click **OK**. The business object you selected is displayed in the ″Objects to be imported″ list.
6. Click **Next**.

Result

The business function you selected is imported from the JD Edwards EnterpriseOne server to the enterprise service discovery wizard. The Configure Objects window opens.

What to do next

Specify a name for the business object and the directory in which it should be stored.

Configuring the selected objects

To configure the business object, you specify information about the object, such a name for the business object and the directory in which it should be stored.

Before you begin

Make sure you have selected and imported the business function.

About this task

Configure the business object that was imported. Name the object, and indicate where the object should be stored.

To configure the business object, use the following procedure.

How to perform this task

1. In the Configure Objects window of the enterprise service discovery wizard, fill out the following fields:
   - **Namespace**: Use the default namespace (http://www.ibm.com/xmlns/prod/websphere/j2ca/jde) except in the following circumstance. If you are adding the business object to an existing module and the module already includes that business object (from an earlier run of the enterprise service discovery wizard), change the namespace value.
   - **Relative Path**: Specify the directory to store the business object.
   - **Maximum Number of Records**: Leave this field set to the default value of 100. You specify the maximum number of records to retrieve when processing a RetrieveAll operation, which is for XML Lists only.
   - **Timeout**: Leave this field empty. You specify a timeout value in milliseconds for XML Lists only. For business objects, the default timeout value is set to the global adapter value.
2. Select **Add Container BO**.
3. In the expanded window, select the business functions you want to add to the operation by clicking the **Add** button next to one of the following fields:
   - **Business Functions for Create**
   - **Business Functions for Retrieve**
   - **Business Functions for Update**
   - **Business Functions for Delete**
   - **Business Functions for Execute**
4. In the **Add** window, select the business function you imported, then click OK.
5. Type a name in the **Container Business Object Name** field. You can type any meaningful name. For example, for the GetEffectiveAddress business object, you may want to type **GetEffectiveAddressContainer**.
6. Click **Next**.

Result

You have associated an operation with the object and selected a name for the object. The Generate Artifacts window opens.

What to do next

Generate a deployable module that includes the adapter and the business object.

Note: If you want to use the Reference or RollbackOnWarning properties in your generated business object, use the Business Object Editor in WebSphere Integration Developer to edit the business object definitions you have created. For information about the Reference and RollbackOnWarning properties, see “Metadata of business functions” on page 107. For information about Business Object Editor, see “Related product information” on page 116.

Generating artifacts

To generate the module, which stores the outbound artifacts that are deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create a new module, include the adapter project in the module, and specify an alias used to authenticate the caller in the JD Edwards EnterpriseOne environment.

Before you begin

Make sure you have configured the business object. The Generate Artifacts window should be open.

About this task

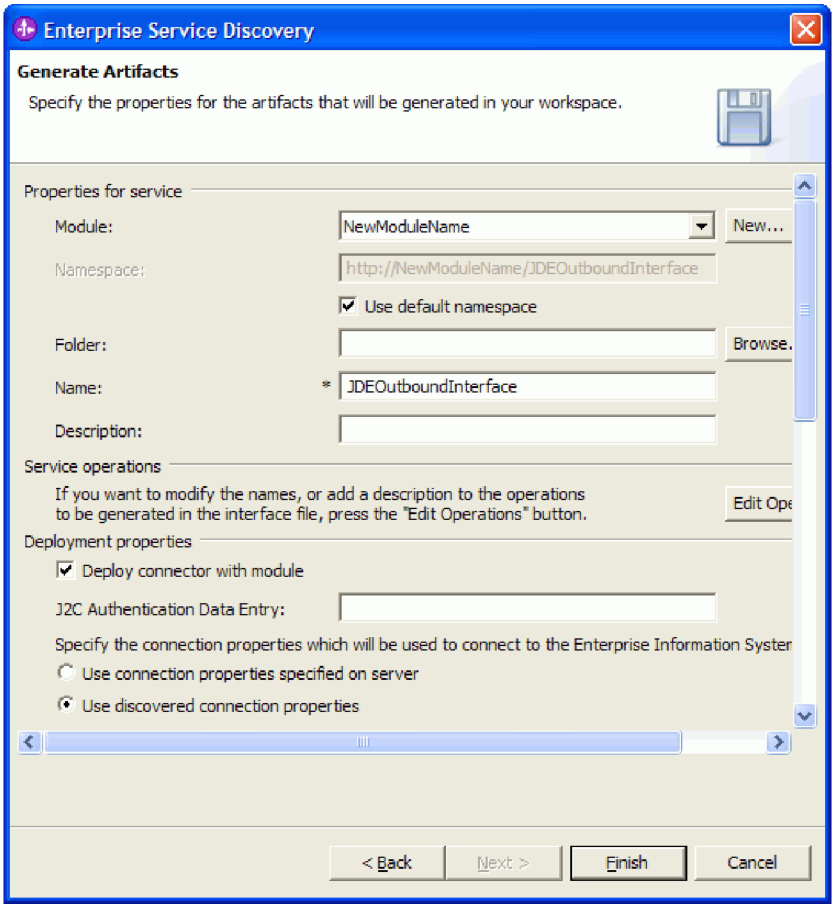
Generate the module, which includes the adapter and configured business object. The module is the artifact you deploy on the server.

To generate the module, use the following procedure.

How to perform this task

1. In the **Generate Artifacts** window, create a new module.
   a. Click **New**.
   b. In the New Integration Project window, select **Create a module project**, then click **Next**.
   c. In the New Module window, type a name for the module.

      Note: As you type the name, it is added to the workplace specified in the **Directory** field.
   d. Click **Finish**. The new module is created. When the creation process is finished, the New Module window closes, and the new module appears in the **Module** list in the Generate Artifacts window.
2. To complete the process of generating the module, make the following selections in the Generating Artifacts window.
   a. In the **Folder** field, specify the folder within the module where the service description should be saved.
   b. Select **Deploy connector with module**.
   c. In the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
   d. Select **Use discovered connection properties** to set properties at this time. Alternatively, if you select **Use connection properties specified on server**, you can configure properties later, using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.
   e. Specify the connection properties. Properties marked with an asterisk (*) are required. For more information about connection properties, see “Enterprise service discovery connection properties” on page 110.



Generate Artifacts

Specify the properties for the artifacts that will be generated in your workspace.

Properties for service

Module: NewModuleName

Namespace: http://NewModuleName/JDEOutboundInterface

☑ Use default namespace

Folder:

Name: * JDEOutboundInterface

Description:

Service operations

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

Deployment properties

☑ Deploy connector with module

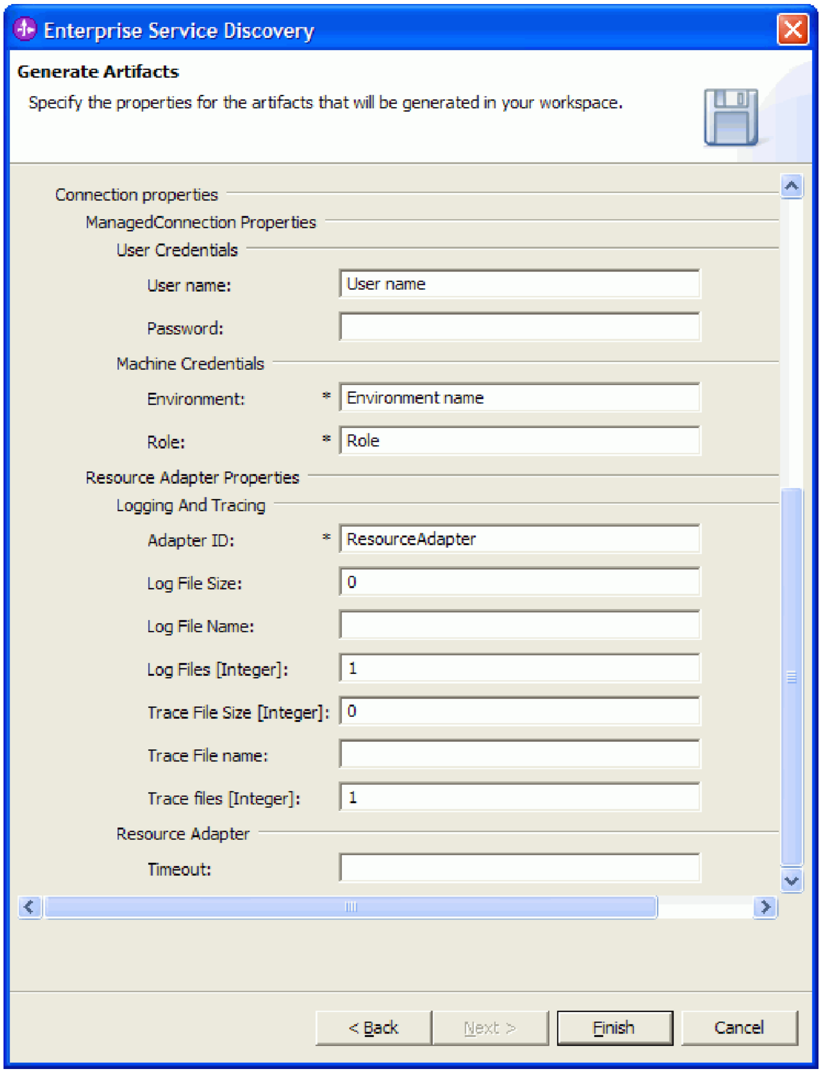
J2C Authentication Data Entry:

Specify the connection properties which will be used to connect to the Enterprise Information Syster

○ Use connection properties specified on server

● Use discovered connection properties

Figure 15. Generate Artifacts window (top portion) of the enterprise service discovery wizard



Generate Artifacts

Specify the properties for the artifacts that will be generated in your workspace.

Connection properties

ManagedConnection Properties

User Credentials

User name: User name

Password:

Machine Credentials

Environment: * Environment name

Role: * Role

Resource Adapter Properties

Logging And Tracing

Adapter ID: * ResourceAdapter

Log File Size: 0

Log File Name:

Log Files [Integer]: 1

Trace File Size [Integer]: 0

Trace File name:

Trace files [Integer]: 1

Resource Adapter

Timeout:

Figure 16. Generate Artifacts window (bottom portion) of the enterprise service discovery wizard

3. Click **Finish**.

Result

The new module is displayed in the J2EE perspective of WebSphere Integration Developer with "App" appended to its name, indicating that the module is a deployable application.

What to do next

Export the module as an EAR file for deployment.

Generating reference bindings

Reference bindings are used by external WebSphere Business Integration SCA components to access the adapter. You create a reference to the adapter from the project module so as to link the adapter to the other server processes. This is required in a standalone testing environment only. It is not necessary when deploying the adapter in a production environment.

Before you begin

Make sure you have created the project module.

About this task

Generating reference bindings is required if you are planning to run the adapter in a test environment before deploying it in a production environment.

To generate reference bindings, use the following procedure.

How to perform this task

1. In the Business Integration perspective of WebSphere Integration Developer, right-click the project module, then select **Open With** → **Assembly Editor**. The Assembly Diagram window opens and displays Import component of the module.
2. To create a new component, click the **Import** icon in the left-side (vertical) pane of the Assembly Diagram window. A new menu of icons is displayed.
3. Move the mouse pointer over each icon to display the hover Help and locate the **Standalone References** icon
4. Click the **Standalone References** icon.
5. Click the blank area (right-side pane) of the Assembly Diagram window to drop the new **Standalone References** component into that pane.
6. Click the new Standalone References component, then move the mouse pointer over the outline of the component until a yellow bulb is displayed on the right side of the component.
7. Drag and drop the yellow bulb to the import module. This draws a wire from the Import component to the new component and opens the Add Wire window.
8. In the Add Wire window, click **OK**.
9. When prompted to use Java interfaces, click **No**. The new Standalone Reference component is displayed in the Assembly Diagram window with a “wire” that connects it to the Import component of the module.
10. Click **File** → **Save** to save the assembly diagram.

Result

You have created a reference from the project module to the adapter.

Configuring the adapter using XML Lists

To configure WebSphere Adapter for JD Edwards EnterpriseOne using XML Lists, use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties for the enterprise service discovery, select XML Lists that are in the enterprise information system, and generate business object definitions and related artifacts for outbound processing.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the JD Edwards EnterpriseOne environment, specify such information as the user name and password you use to access the server as well as the environment name and role of the user.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the JD Edwards EnterpriseOne environment and discover its business objects and services.

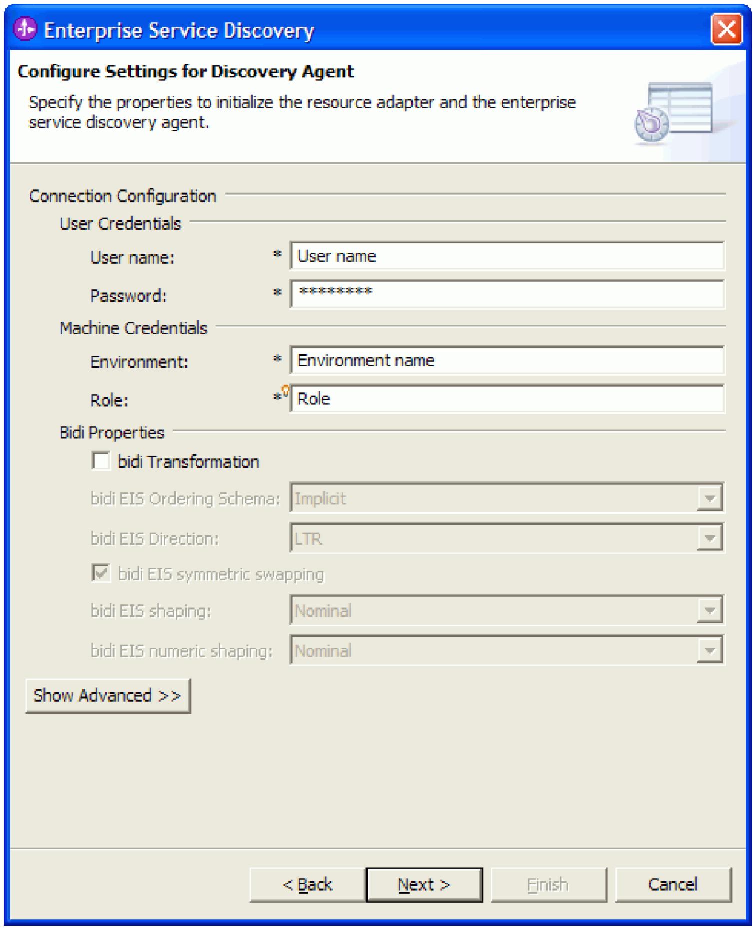
To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration perspective by clicking **Window** → **Open Perspective** → **Other**.
   b. In the Select Perspective window, select **Business Integration**, then click **OK**.
   c. Select **File** → **New** → **Enterprise Service Discovery**.
2. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for JD Edwards EnterpriseOne (version 6.0.2) from the CWYED_JDE Connector Project** , then click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the JD Edwards EnterpriseOne application you used the last time you ran the enterprise service discovery wizard.
3. Specify the configuration properties to initialize the discovery agent.

   Note: Properties marked with an asterisk (*) are required.
   a. Type the name and password you use to access the JD Edwards EnterpriseOne system.
   b. Type the environment name of your JD Edwards EnterpriseOne system.
   c. Type the role name you use to access the JD Edwards EnterpriseOne system.


Enterprise Service Discovery

Configure Settings for Discovery Agent

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

Connection Configuration

User Credentials

User name: * User name

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

Machine Credentials

Environment: * Environment name

Role: * Role

Bidi Properties

□ bidi Transformation

bidi EIS Ordering Schema: Implicit

bidi EIS Direction: LTR

☑ bidi EIS symmetric swapping

bidi EIS shaping: Nominal

bidi EIS numeric shaping: Nominal

Figure 17. The Configure Settings for Discovery Agent window

4. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See “Settings for controlling bidirectional transformation” on page 115 for more information about these properties.
5. **Optional:** To change the logging level, perform the following steps:
   a. At the bottom of the window, click **Show Advanced**.
   b. Set the **Logging Level**.

      In a test environment, select **FINEST**, which provides the highest level of logging. In a production environment, choose a level lower than **FINEST** to optimize the logging process.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the JD Edwards EnterpriseOne environment, using the information you provided, then the Find and Discover Enterprise Services window opens.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to discover business objects and services in the JD Edwards EnterpriseOne environment.

Selecting business objects and services

To specify which XML List you want to use for outbound processing, you provide information in the enterprise service discovery wizard.

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

About this task

Specify search criteria that the enterprise service discovery wizard uses to discover XML Lists in the JD Edwards EnterpriseOne environment. Before you can run the query for XML Lists on the server, however, you must specify the table name so that enterprise service discovery can retrieve metadata about that table. After executing the query, the enterprise service discovery wizard returns a list of XML Lists.

To specify the search criteria and select an XML Lists, use the following procedure.

How to perform this task

1. Specify the table to use in the XML List query.
   a. In the Find and Discover Enterprise Services window of the enterprise services discovery wizard, click **Edit Query**.
   b. In the Query Filter Properties window, click **Add** next to the Tables list.
   c. In the Add window that opens, type the name of the table in the JD Edwards EnterpriseOne application that contains the data, then click **OK**. The table name you typed is displayed in the Tables list.
   d. Click **OK**.
2. Execute the XML List query to find and discover the table in the JD Edwards EnterpriseOne application that matches the table you specified.
   a. In the Find and Discover Enterprise Services window, click **Execute Query**.
   b. Expand the **XML Lists** node, navigate to the table that matches the table you created, then click **Add to import list**.
3. Add search criteria to the query before importing the data from the JD Edwards EnterpriseOne environment. This allows you to specify query parameters such as table type, sorting conditions, and other query conditions.

   Note: When you make selections in the Configure Parameters for <*table_name*> window, the fields change causing the window to expand or collapse.

   a. In the Configuration Parameters for <*table_name*> window, you can either keep the default name in the Business Object Name field or rename it.
   b. In the Table Type field, select the type of table from the list. The following table types are available:
      - **OWTABLE**
      - **OWVIEW**
      - **FOREIGN_TABLE**
      - **TABLE_CONVERSION**

c. In the Queries section that expands, you can optionally add a sorting order or add conditions to the query.

   Note: If the Queries section does not automatically expand, expand it by clicking **Add Query**.

   - If you want to add sorting to the query, click **Add Sorting**, then select an attribute and a sorting order (ascending or descending).
   - If you want to add a condition, select **Add Condition**, then select conditions from the following fields.

     Note: If no conditions are specified, all records are retrieved.

     – **Attribute**: Select the attribute for the query.

     – **Clause**: Select the clause for the query condition. The default is **Where**.

     – **Operator**: Select the operator when comparing the attribute to the column value.

     – **Use Attribute Value**: Select an attribute to compare to.

     – **Default**: Specify the default value for the query condition.

d. Click **OK**. The XML List is displayed in the ″Objects to be imported″ list.

4. Click **Next**.

Result

The XML List you selected is imported from the JD Edwards EnterpriseOne environment to the enterprise service discovery wizard. The Configure Objects window opens.

What to do next

Specify a name for the business object and the directory in which it should be stored.

Configuring the selected objects

To configure the business object, you specify information about the object, such a name for the business object and the directory in which it should be stored.

Before you begin

Make sure you have selected and imported the XML List.

About this task

Configure the business object that was imported. Name the object, and indicate where the object should be stored.

To configure the business object, use the following procedure.

How to perform this task

1. In the Configure Objects window of the enterprise service discovery window, fill out the following fields:
   - **Namespace**: Use the default namespace (http://www.ibm.com/xmlns/prod/websphere/j2ca/jde) except in the following circumstance. If you are adding

the business object to an existing module and the module already includes that business object (from an earlier run of the enterprise service discovery wizard), change the namespace value.

- **Relative Path**: Specify the directory to store the business object.
- **Maximum Number of Records**: Specify the maximum number of records to retrieve when processing a RetrieveAll operation.
- **Timeout**: Specify a timeout value in milliseconds for XML Lists request execute call.

2. Leave the **Add Container BO** box de-selected, then click **Next**.

Result

You have associated an operation (RetrieveAll) with the object and selected a name for the object. The Generate Artifacts window opens.

What to do next

Generate a deployable module that includes the adapter and the business object.

Note: If needed, use the Business Object Editor in WebSphere Integration Developer to edit the business object definitions you have created. For information about Business Object Editor, see “Related product information” on page 116.

Generating artifacts

To generate the module, which stores the outbound artifacts that are deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create a new module, include the adapter project in the module, and specify an alias used to authenticate the caller in the JD Edwards EnterpriseOne environment.

Before you begin

Make sure you have configured the business object. The Generate Artifacts window should be open.

About this task

Generate the module, which includes the adapter and configured business object. The module is the artifact you deploy on the server.

To generate the module, use the following procedure.

How to perform this task

1. In the **Generate Artifacts** window, create a new module.
   a. Click **New**.
   b. In the New Integration Project window, select **Create a module project**, then click **Next**.
   c. In the New Module window, type a name for the module.

      Note: As you type the name, it is added to the workplace specified in the **Directory** field.

   d. Click **Finish**. The new module is created. When the creation process is finished, the New Module window closes, and the new module appears in the **Module** list in the Generate Artifacts window.

2. To complete the process of generating the module, make the following selections in the Generating Artifacts window.

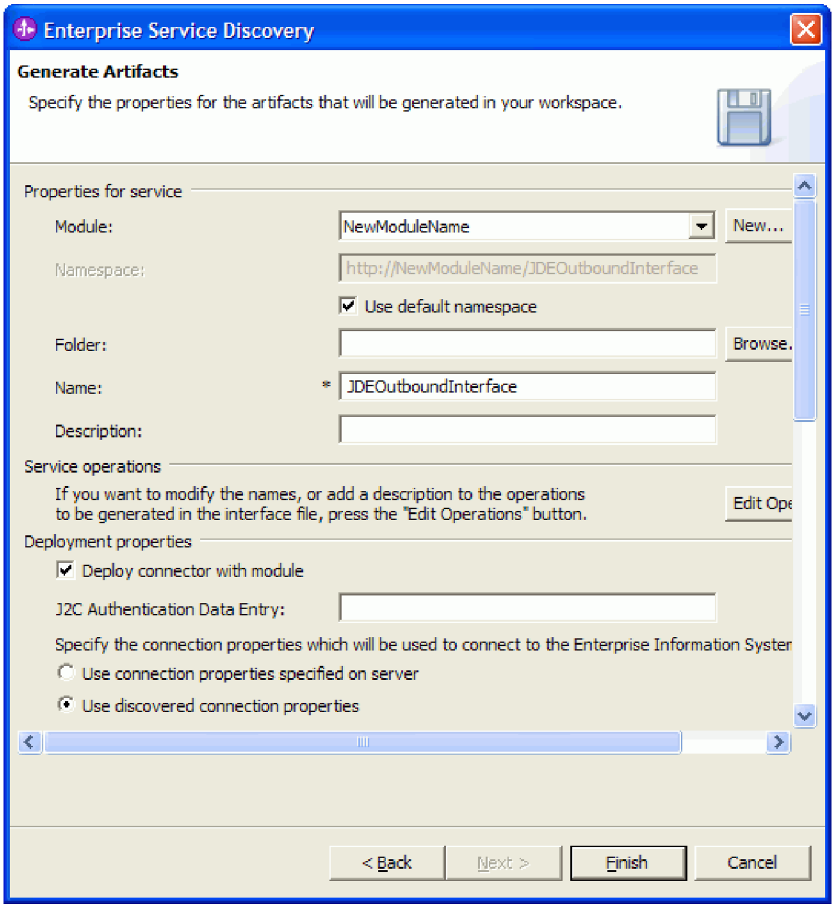
   a. In the **Folder** field, specify the folder within the module where the service description should be saved.

   b. Select **Deploy connector with module**.

   c. In the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.

   d. Select **Use discovered connection properties** to set properties at this time. Alternatively, if you select **Use connection properties specified on server**, you can configure properties later, using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

   e. Specify the connection properties. Properties marked with an asterisk (*) are required. For more information about connection properties, see "Enterprise service discovery connection properties" on page 110.



Enterprise Service Discovery

Generate Artifacts

Specify the properties for the artifacts that will be generated in your workspace.

Properties for service

Module: NewModuleName

Namespace: http://NewModuleName/JDEOutboundInterface

☑ Use default namespace

Folder:

Name: * JDEOutboundInterface

Description:

Service operations

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

Deployment properties

☑ Deploy connector with module

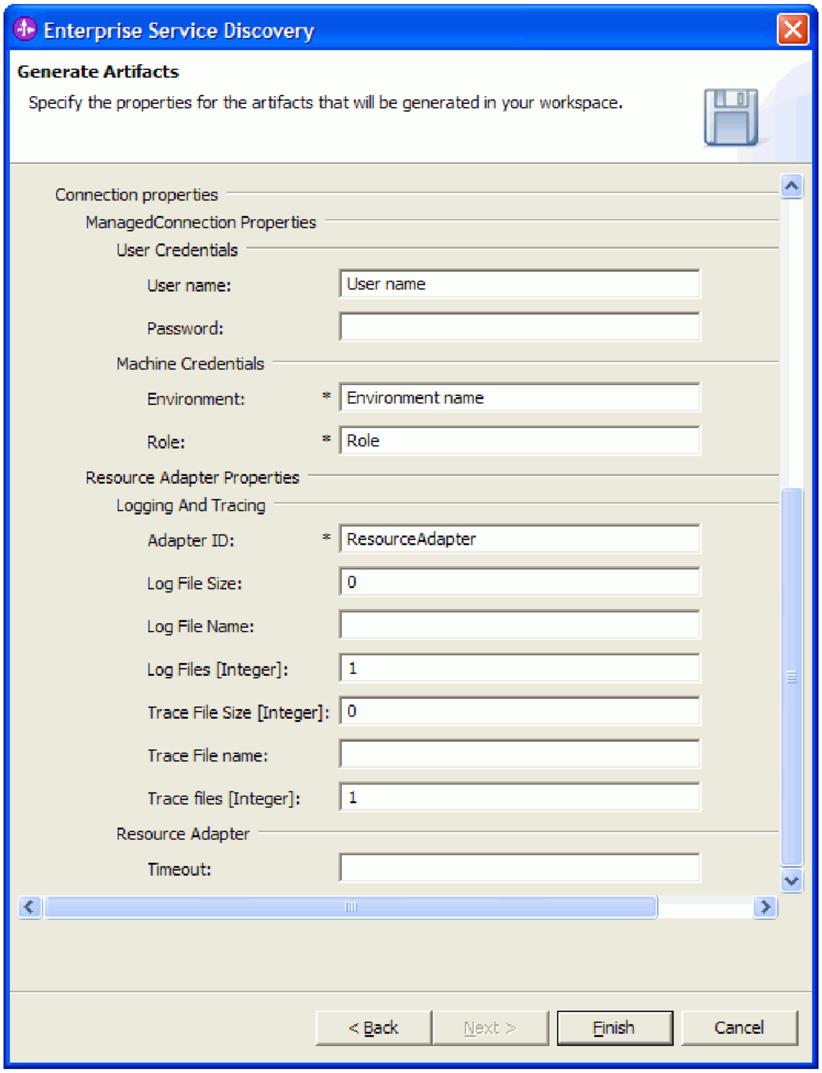
J2C Authentication Data Entry:

Specify the connection properties which will be used to connect to the Enterprise Information Syster

○ Use connection properties specified on server

● Use discovered connection properties

Figure 18. Generate Artifacts window (top portion) of the enterprise service discovery wizard



Generate Artifacts

Specify the properties for the artifacts that will be generated in your workspace.

Connection properties

ManagedConnection Properties

User Credentials

User name: User name

Password:

Machine Credentials

Environment: * Environment name

Role: * Role

Resource Adapter Properties

Logging And Tracing

Adapter ID: * ResourceAdapter

Log File Size: 0

Log File Name:

Log Files [Integer]: 1

Trace File Size [Integer]: 0

Trace File name:

Trace files [Integer]: 1

Resource Adapter

Timeout:

Figure 19. Generate Artifacts window (bottom portion) of the enterprise service discovery wizard

3. Click **Finish**.

Result

The new module is displayed in the J2EE perspective of WebSphere Integration Developer with "App" appended to its name, indicating that the module is a deployable application.

What to do next

Export the module as an EAR file for deployment.

Generating reference bindings

Reference bindings are used by external WebSphere Business Integration SCA components to access the adapter. You create a reference to the adapter from the project module so as to link the adapter to the other server processes. This is required in a standalone testing environment only. It is not necessary when deploying the adapter in a production environment.

Before you begin

Make sure you have created the project module.

About this task

Generating reference bindings is required if you are planning to run the adapter in a test environment before deploying it in a production environment.

To generate reference bindings, use the following procedure.

How to perform this task

1. In the Business Integration perspective of WebSphere Integration Developer, right-click the project module, then select **Open With** → **Assembly Editor**. The Assembly Diagram window opens and displays Import component of the module.
2. To create a new component, click the **Import** icon in the left-side (vertical) pane of the Assembly Diagram window. A new menu of icons is displayed.
3. Move the mouse pointer over each icon to display the hover Help and locate the **Standalone References** icon
4. Click the **Standalone References** icon.
5. Click the blank area (right-side pane) of the Assembly Diagram window to drop the new **Standalone References** component into that pane.
6. Click the new Standalone References component, then move the mouse pointer over the outline of the component until a yellow bulb is displayed on the right side of the component.
7. Drag and drop the yellow bulb to the import module. This draws a wire from the Import component to the new component and opens the Add Wire window.
8. In the Add Wire window, click **OK**.
9. When prompted to use Java interfaces, click **No**. The new Standalone Reference component is displayed in the Assembly Diagram window with a "wire" that connects it to the Import component of the module.
10. Click **File** → **Save** to save the assembly diagram.

Result

You have created a reference from the project module to the adapter.

Chapter 8. Deploying the module

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

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

To deploy the module, you must export it as an EAR file. The EAR file is created during the deployment process.

Before you begin

Make sure you have created a module. The module should be displayed in the J2EE perspective of WebSphere Integration Developer with ″App″ appended to its name, indicating that the module is a deployable application.

About this task

An EAR file is a specialized type of JAR file, defined by the J2EE standard, used to deploy J2EE applications to J2EE application servers. You create the EAR file when you export the module.

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

How to perform this task

1. From the J2EE perspective in WebSphere Integration Developer, expand the **Enterprise Applications** node.
2. Right-click the module, then select **Export** → **EAR file**.
3. In the EAR Export window, select the destination directory (the directory, including the EAR filename, where the project should be exported).
4. **Optional:** Select one or more of the following options.
   - Export source files
   - Overwrite existing file
   - Include project build paths and meta-data files
5. Click **Finish**.

Result

The EAR file is created and saved and is ready to be deployed.

What to do next

Install the application on WebSphere Process Server or WebSphere Enterprise Service Bus.

Installing the module

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

Before you begin

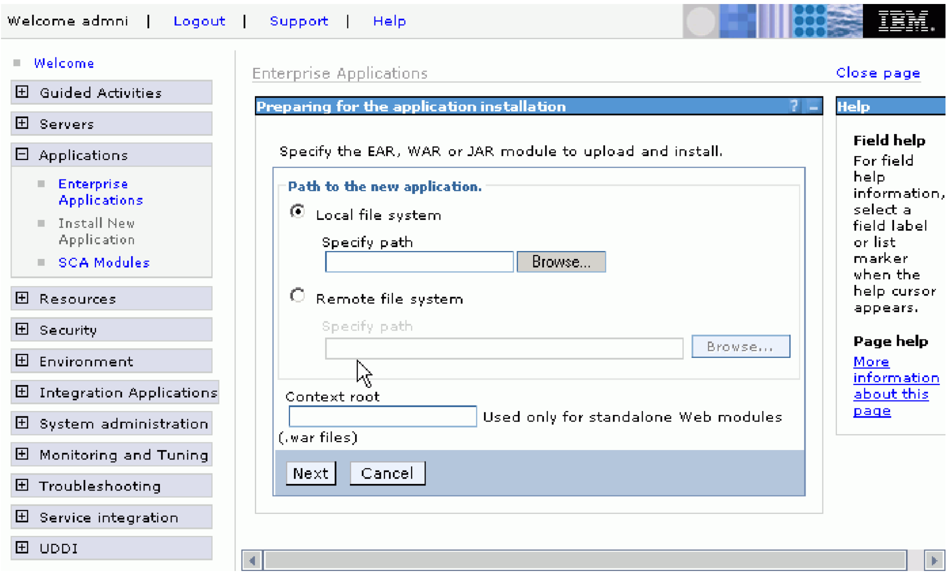
You must have exported your project module as an EAR file before installing the adapter project.

About this task

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

How to perform this task

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



Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

● Local file system

Specify path

○ Remote file system

Specify path

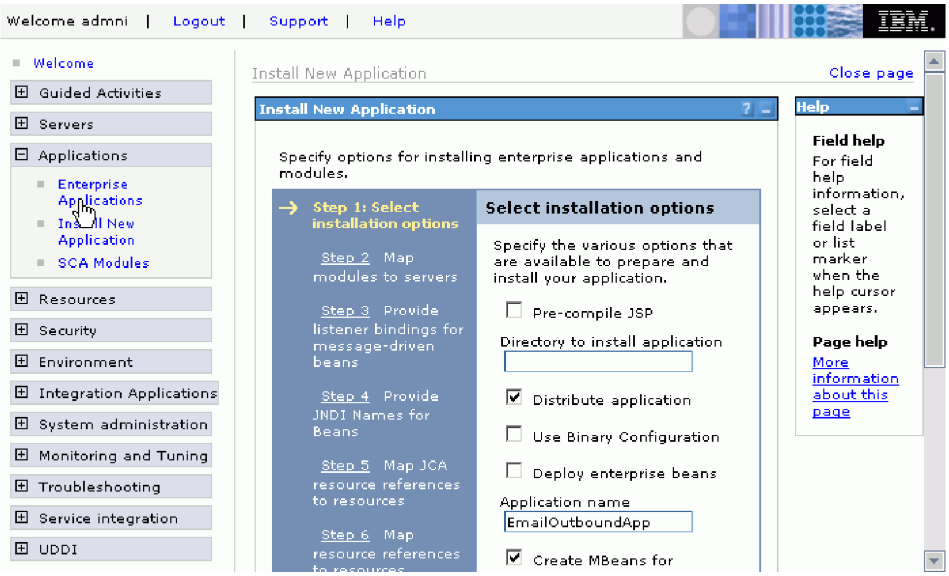
Context root

Used only for standalone Web modules (.war files)

For field help information, select a field label or list marker when the help cursor appears.

Preparing for the application installation window

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



Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options

Step 2 Map modules to servers

Step 3 Provide listener bindings for message-driven beans

Step 4 Provide JNDI Names for Beans

Step 5 Map JCA resource references to resources

Step 6 Map resource references to resources

Select installation options

Specify the various options that are available to prepare and install your application.

☐ Pre-compile JSP

Directory to install application

☑ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name: EmailOutboundApp

☑ Create MBeans for

Field help

For field help information, select a field label or list marker when the help cursor appears.

Page help

More information about this page

Install New Application window

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

Result

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

What to do next

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

Setting or changing configuration properties from the administrative console

To set or change configuration properties after you deploy a module, you use the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus. For example, you can update managed (J2C) connection factory properties.

Setting resource adapter properties

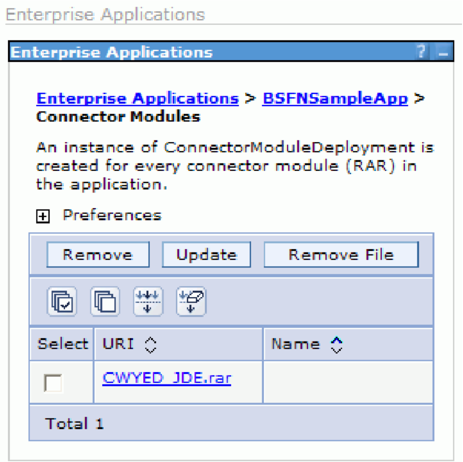
To set resource adapter properties for your adapter module after it has been deployed, use the administrative console. You select the name of the property you want to configure and then change or set the value as desired.

Before you begin

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

How to perform this task

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Click the **CWYED_JDE.rar** file.



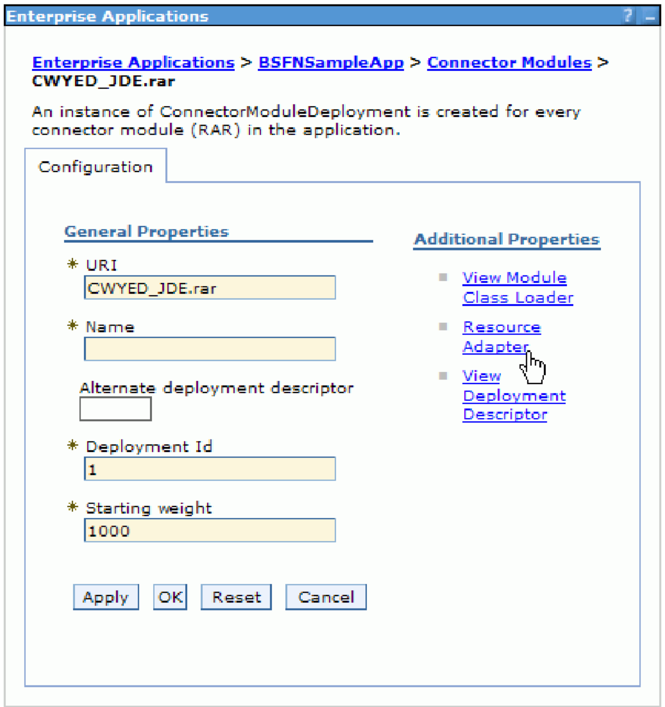
Enterprise Applications > BSFNSampleApp > Connector Modules

An instance of ConnectorModuleDeployment is created for every connector module (RAR) in the application.

Select	URI	Name
□	CWYED_JDE.rar	
Total 1		

Figure 20. Location of CWYED_JDE.rar file in Enterprise Applications window

6. From the Additional Properties list, click **Resource Adapter**.


Enterprise Applications

Enterprise Applications > BSFNSampleApp > Connector Modules > CWYED_JDE.rar

An instance of ConnectorModuleDeployment is created for every connector module (RAR) in the application.

Configuration

General Properties

* URI: CWYED_JDE.rar

* Name:

Alternate deployment descriptor:

* Deployment Id: 1

* Starting weight: 1000

Figure 21. Location of **Resource Adapter** *link on Configuration page of Enterprise Applications window*

7. From the Additional Properties list, click **Custom properties**. The following figure shows the location of the **Custom properties** link in the Additional Properties list.

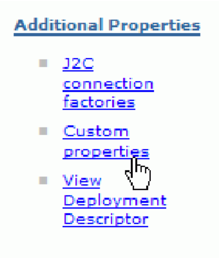
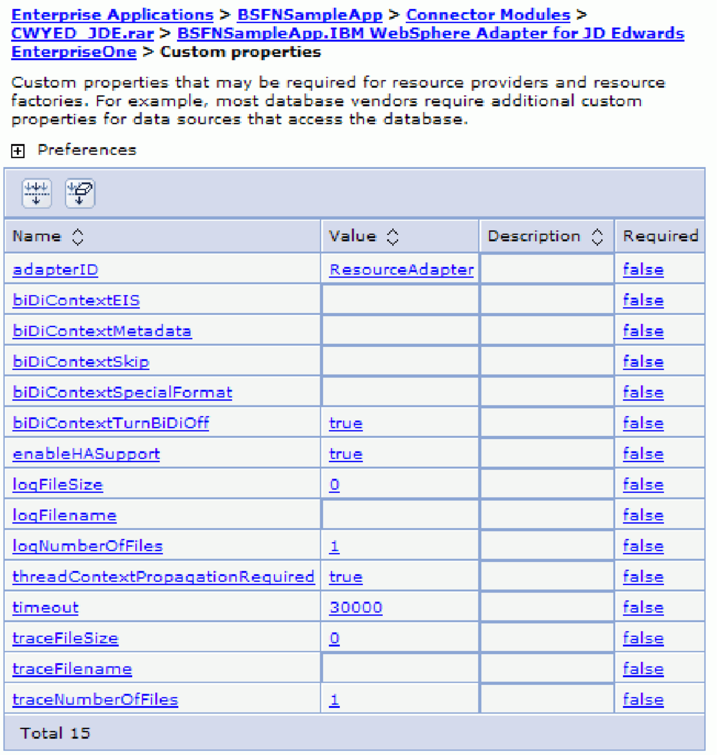


Figure 22. Location of the **Custom properties** *link in the Additional Properties list*

8. For each property you want to change, perform the following steps.
   a. Click the name of the property.



Enterprise Applications > BSFNSampleApp > Connector Modules > CWYED_JDE.rar > BSFNSampleApp.IBM WebSphere Adapter for JD Edwards EnterpriseOne > Custom properties

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

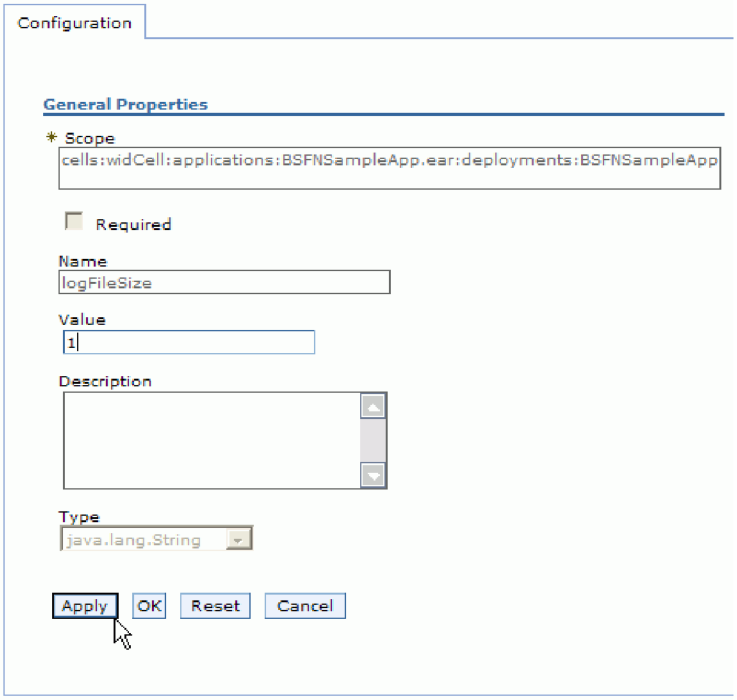
⊞ Preferences

Name	Value	Description	Required
adapterID	ResourceAdapter		false
biDiContextEIS			false
biDiContextMetadata			false
biDiContextSkip			false
biDiContextSpecialFormat			false
biDiContextTurnBiDiOff	true		false
enableHASupport	true		false
logFileSize	0		false
logFilename			false
logNumberOfFiles	1		false
threadContextPropagationRequired	true		false
timeout	30000		false
traceFileSize	0		false
traceFilename			false
traceNumberOfFiles	1		false

Total 15

Figure 23. Resource Adapter properties in Enterprise Applications window

b. In the Configuration page for the property, change the contents of the **Value** field value or type a value, if the field is empty. For information about resource adapter properties, see “Resource adapter properties” on page 112.



Configuration

General Properties

* Scope
cells:widCell:applications:BSFNSampleApp.ear:deployments:BSFNSampleApp

□ Required

Name
logFileSize

Value
1

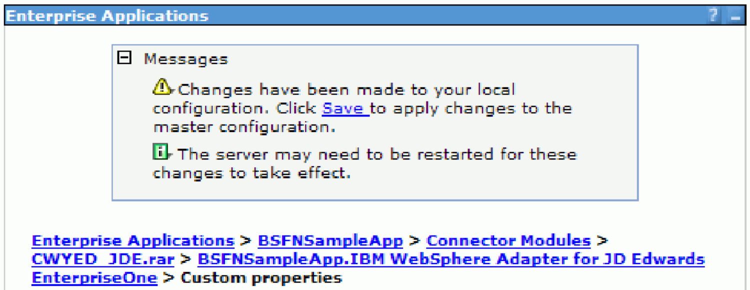
Description

Type
java.lang.String

Figure 24. Configuration page of the logFileSize property

c. Click **Apply**.

9. In the Messages box that appears at the top of the Enterprise Applications window, click the **Save** link if you want to save your changes to the master configuration.



Messages

Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

The server may need to be restarted for these changes to take effect.

Enterprise Applications > BSFNSampleApp > Connector Modules > CWYED_JDE.rar > BSFNSampleApp.IBM WebSphere Adapter for JD Edwards EnterpriseOne > Custom properties

Figure 25. Messages box that displays when applying changes to resource adapter properties

Result

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

Setting managed (J2C) connection factory properties

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

Before you begin

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

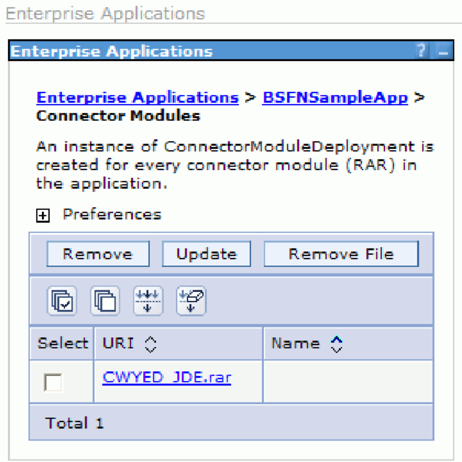
About this task

You use managed connection factory properties to configure the target JD Edwards EnterpriseOne environment.

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

How to perform this task

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Click the **CWYED_JDE.rar** file. The following figure illustrates the location of the CWYED_JDE.rar file in the Enterprise Applications window.


Enterprise Applications

Enterprise Applications

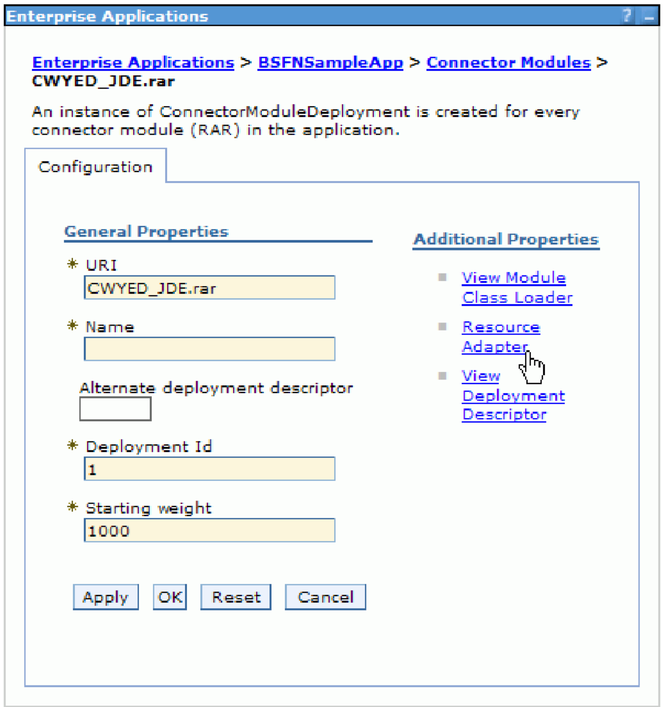
Enterprise Applications > BSFNSampleApp > Connector Modules

An instance of ConnectorModuleDeployment is created for every connector module (RAR) in the application.

Select	URI	Name
□	CWYED_JDE.rar	
Total 1		

Figure 26. Location of CWYED_JDE.rar file in Enterprise Applications window

6. From the Additional Properties list, click **Resource Adapter**. The following figure shows the location of the **Resource Adapter** link on the Configuration page of the Enterprise Applications window.


Enterprise Applications

Enterprise Applications > BSFNSampleApp > Connector Modules > CWYED_JDE.rar

An instance of ConnectorModuleDeployment is created for every connector module (RAR) in the application.

Configuration

General Properties

* URI
CWYED_JDE.rar

* Name

Alternate deployment descriptor

* Deployment Id
1

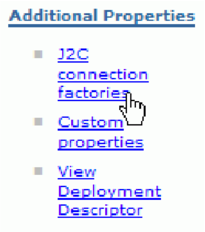
* Starting weight
1000

Additional Properties

- View Module Class Loader
- Resource Adapter
- View Deployment Descriptor

Figure 27. Location of **Resource Adapter** *link on Configuration page of Enterprise Applications window*

7. From the Additional Properties list, click **J2C connection factories**.

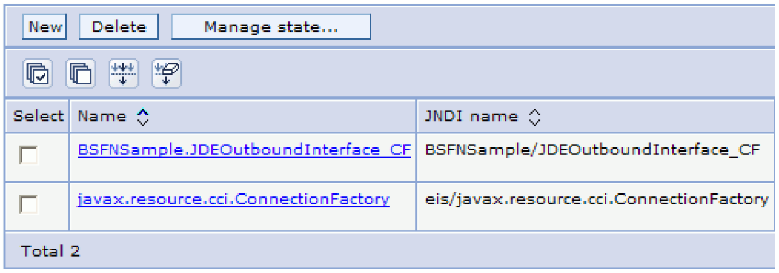


Additional Properties

- J2C connection factories
- Custom properties
- View Deployment Descriptor

Figure 28. Location of the **J2C connection factory** *link in the Additional Properties list*

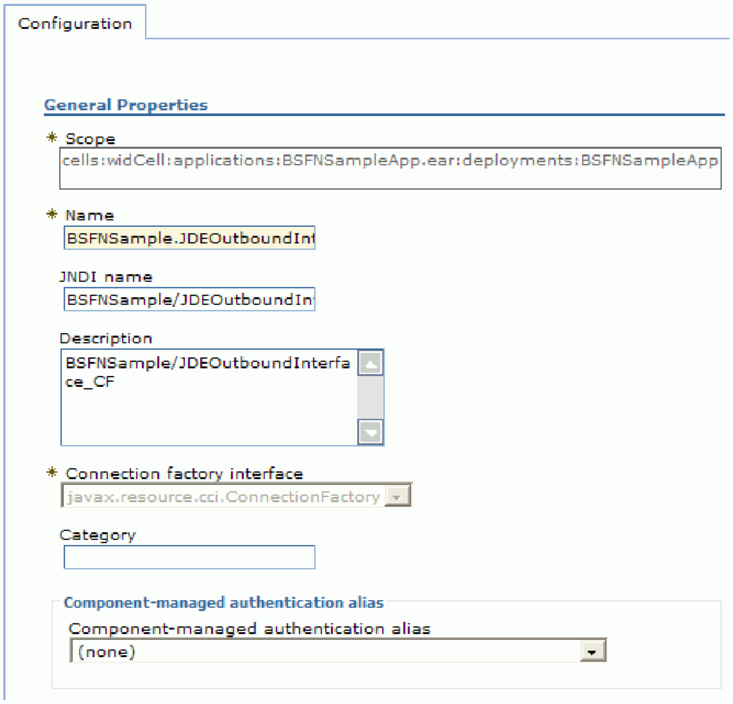
8. Click the name of the J2C connection factory you want to configure.



Select	Name	JNDI name
□	BSFNSample.JDEOutboundInterface_CF	BSFNSample/JDEOutboundInterface_CF
□	javax.resource.cci.ConnectionFactory	eis/javax.resource.cci.ConnectionFactory
Total 2		

Figure 29. Example of J2C connection factories listed in the Enterprise Applications window

9. For each connection factory property you want to change, perform the following steps. For more information about managed (J2C) connection factory properties, see “Managed (J2C) connection factory properties” on page 114.
   a. In the Configuration page of the chosen connection factory, change the contents of the fields that you want to change.



Configuration

General Properties

* Scope
cells:widCell:applications:BSFNSampleApp.ear:deployments:BSFNSampleApp

* Name
BSFNSample.JDEOutboundInt

JNDI name
BSFNSample/JDEOutboundIn

Description
BSFNSample/JDEOutboundInterface_CF

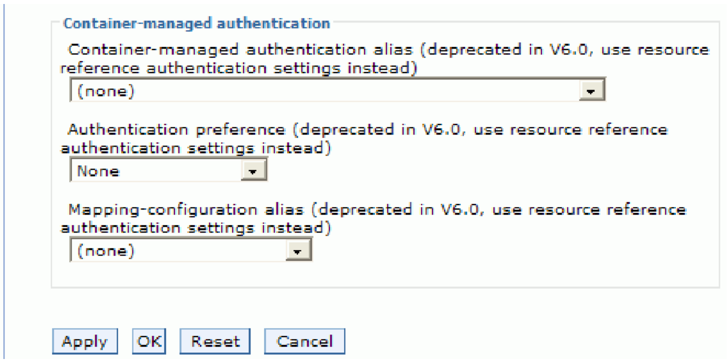
* Connection factory interface
javax.resource.cci.ConnectionFactory

Category

Component-managed authentication alias

Component-managed authentication alias
(none)

Figure 30. Configuration page (top portion) of a connection factory



Container-managed authentication

Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)

(none)

Authentication preference (deprecated in V6.0, use resource reference authentication settings instead)

None

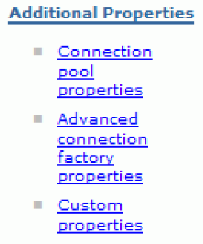
Mapping-configuration alias (deprecated in V6.0, use resource reference authentication settings instead)

(none)

Figure 31. Configuration page (bottom portion) of a connection factory

b. Click **Apply**.

Note: In the Additional properties list, **Connection pool** and **Advanced connection factory properties** are properties you configure if you are developing your own adapter, while **Custom properties** are those J2C connection factory properties that are unique to Adapter for JD Edwards EnterpriseOne Software.

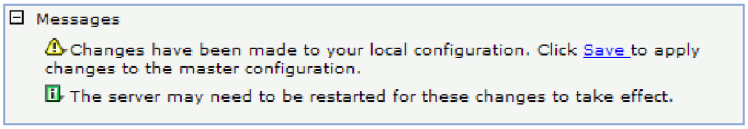


Additional Properties

- Connection pool properties
- Advanced connection factory properties
- Custom properties

Figure 32. Additional Properties list

10. In the Messages window at the top of the Enterprise Applications window, click the **Save** link to save your changes to the master configuration.



⊟ Messages

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

🛈 The server may need to be restarted for these changes to take effect.

Figure 33. Messages window

Result

The managed connection properties associated with your adapter application are changed.

Chapter 9. Configuring troubleshooting tools

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

Enabling tracing with the Common Event Infrastructure (CEI)

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

Before you begin

Before you enable tracing with CEI, complete the following tasks:
- Enable the diagnostic trace service.
- Publish the IBM WebSphere Adapters event definitions file to the CEI catalog before you can set these event definitions.

For instruction on how to do these tasks, refer to the CEI documentation located on the Web site for your server:
- For WebSphere Process Server: http://www.ibm.com/software/integration/wps
- For WebSphere Enterprise Service Bus: http://www.ibm.com/software/integration/wsesb

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

How to perform this task

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

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

5. Select the component that matches your adapter. Each adapter component has two subcomponents, one for logging and one for CEI. They are:

- *subcomponent_name*.log.*adapter_ID*
- *subcomponent_name*.cei.*adapter_ID*

For example, com.ibm.j2ca.siebel.cei.*adapter_ID1*. For each instance of a deployed adapter, the system shows a separate ID.

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

   For information on what each event content level means (Empty, Digest and Full), and for more information on using the Common Base Event model and the Common Event Infrastructure, refer to the documentation for your process server.

Configuring logging properties

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

About this task

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

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

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

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

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

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

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

How to perform this task

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

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

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

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

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

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

Changing the log and trace file names

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

About this task

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

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

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

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

How to perform this task

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

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

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

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

Installing or upgrading IBM Support Assistant

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

About this task

IBM Support Assistant provides the following services:

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

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

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

How to perform this task

1. Go to the IBM Support Assistant Web page at:

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

Chapter 10. Troubleshooting and support

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

Self help resources

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

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

- Flashes (alerts from technical support)
- Technotes

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

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

Contacting IBM Software Support

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

Before you begin

If you think your problem is defect-related, IBM Software Support provides assistance. Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli®, Lotus®, and Rational® products, as well as DB2® and WebSphere products that run on Windows, Linux®, or UNIX® operating systems), you must be enrolled in Passport Advantage®. You can enroll in one of the following ways:

  Online

  Go to the Passport Advantage Web page (http://www-306.ibm.com/software/support/pa.html), and click **How to Enroll**.

By phone

For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web (http://techsupport.services.ibm.com/guides/contacts.html), and click the name of your geographic region.

- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries®, pSeries®, and iSeries™ environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page (http://www-03.ibm.com/servers/eserver/techsupport.html).

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

About this task

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

To contact IBM Software Support, use the following procedure.

How to perform this task

1. Describe your problem and gather background information. When explaining a problem to a support specialist, be as specific as possible. Include all relevant background information so that the specialists can help you solve the problem efficiently. To save time, know the answers to these questions:
   - What software versions were you running when the problem occurred? Include the version of the operating system as well as related products.
   - Has the problem happened before, or is this an isolated problem?
   - What steps led to the failure?
   - Can the problem be recreated? If so, what steps led to the failure?
   - Have any changes been made to the system such as to the hardware, operating system, networking software, and so on?
   - Are you currently using a workaround for this problem? If so, be prepared to explain it when you report the problem.
   - Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
2. Determine the business impact of your problem. When you report a problem, you will be asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting. Use the criteria described in the following table.

Table 3. Severity criteria for problem reporting

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

3. Submit your problem to IBM Software Support. You can submit your problem in the following ways:
   - **Online**. Go to the Submit and track problems page on the IBM Software Support site http://www.ibm.com/software/support/probsub.html Enter your information into the appropriate problem submission tool.
   - **By phone**. For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web (http://techsupport.services.ibm.com/guides/contacts.html), and click the name of your geographic region.

Result

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

What to do next

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

Chapter 11. Quick start tutorials

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

Introduction

Each tutorial provides a complete set of instructions for configuring the adapter so that it can be used by a J2EE component to send requests to the JD Edwards EnterpriseOne environment.

In the tutorials, you use WebSphere Integration Developer (and its enterprise service discovery wizard) to configure the adapter, connect to the JD Edwards EnterpriseOne environment, and retrieve information about a service in the JD Edwards EnterpriseOne environment. Enterprise service discovery then creates the business objects and interface information needed to interact with the service. The business objects and interface information, along with the adapter, are built into a deployable module.

Two tutorials are provided.

- Retrieving data from the JD Edwards EnterpriseOne application using a business function
- Retrieving data from a JD Edwards EnterpriseOne table using an XML List query

Learning objectives

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

- Create an adapter project in WebSphere Integration Developer
- Discover services and associated business objects from the JD Edwards EnterpriseOne environment and make them part of the adapter project
- Create a deployable module that you install in the WebSphere Process Server test environment
- Test the module to ensure that it operates correctly and to see the results of running the module

Time required

The following table lists the approximate time required to complete each tutorial.

Table 4. Time required to complete the tutorials

Tutorial	Time to complete
Tutorial 1: Retrieving data from the JD Edwards EnterpriseOne application using a business function	30 minutes
Tutorial 2: Retrieving data from a JD Edwards EnterpriseOne table using an XML List query	30 minutes

Audience

The tutorials are intended for the integration developer who will be configuring Adapter for JD Edwards EnterpriseOne for deployment on WebSphere Process Server or WebSphere Enterprise Service Bus.

Prerequisites

Before you begin the tutorials, make sure you have performed the following tasks:

- Install all prerequisite software
- Install Adapter for JD Edwards EnterpriseOne

Also make sure you have all the information (such as user ID and password) needed to access the JD Edwards EnterpriseOne environment.

Tutorial 1: Retrieving data from the JD Edwards EnterpriseOne application using a business function

To create a module that uses a business function to retrieve data from the JD Edwards EnterpriseOne database, you create an adapter project, use the enterprise service discovery wizard to generate the business function, and create a module that contains WebSphere Adapter for JD Edwards EnterpriseOne and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create the authentication alias on the server, use the administrative console. From the administrative console, configure the global security and set the password for the authentication alias, which is used to process outbound requests.

Before you begin

You must have access to the administrative console of the either WebSphere Process Server or WebSphere Enterprise Service Bus.

How to perform this task

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

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

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with a JD Edwards EnterpriseOne service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) contains the adapter itself plus other related artifacts. You create the project by importing the RAR file, which was copied to your local file system during the adapter installation, into WebSphere Integration Developer.

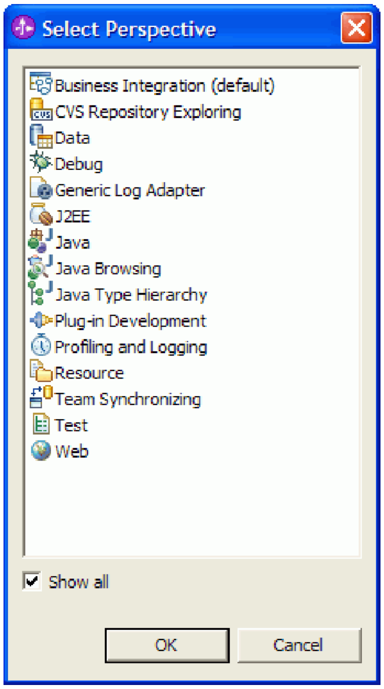
About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

1. In WebSphere Integration Developer, switch to the J2EE perspective:
   a. Click **Window** → **Open Perspective** → **Other**.
   b. Click **J2EE**.

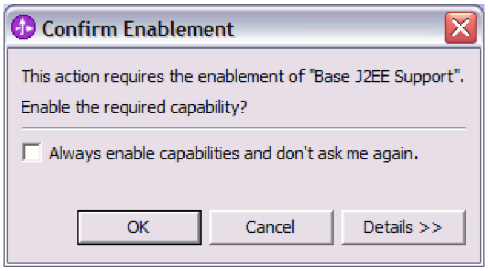
      If **J2EE** is not displayed, select the **Show all** check box, click **J2EE**, and click **OK**.



☑ Show all

Figure 34. Selecting J2EE from the Select Perspective list

   c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.



Confirm Enablement

This action requires the enablement of "Base J2EE Support". Enable the required capability?

□ Always enable capabilities and don't ask me again.

Figure 35. The Confirm Enablement window

d. Click **OK**.

2. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

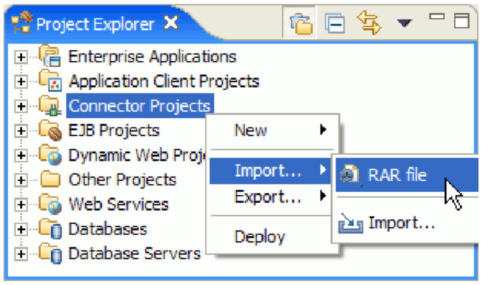
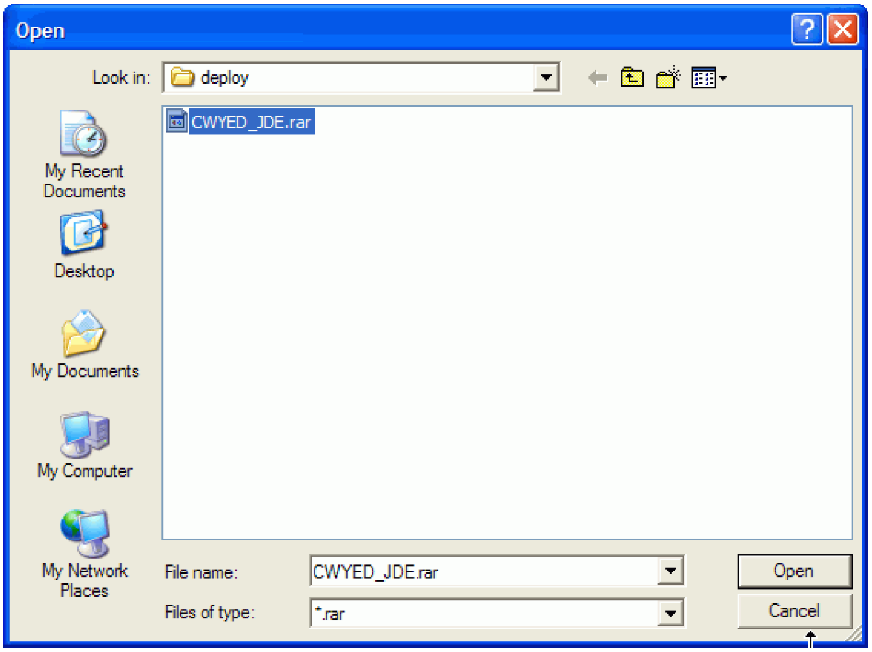


Figure 36. Importing the RAR file

3. Find the RAR file on your local file system by clicking **Browse** and navigating to the directory in which Adapter for JD Edwards EnterpriseOne was installed.



Open

Look in: deploy

CWYED_JDE.rar

File name: CWYED_JDE.rar

Files of type: *.rar

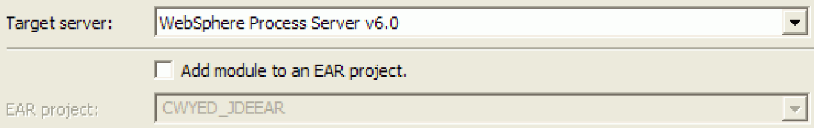
Figure 37. Selecting the RAR file from the installation directory

4. Accept the default setting (**CWYED_JDE.rar**) for **Connector project**. The connector project has the same name as the RAR file.

Note: If a project named CWYED_JDE.rar already exists in this workspace, the name in the Connector project field has a number appended to it (for example, CWYED_JDE1).

5. Accept the default value in the **Target server** field.

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



Target server: WebSphere Process Server v6.0

□ Add module to an EAR project.

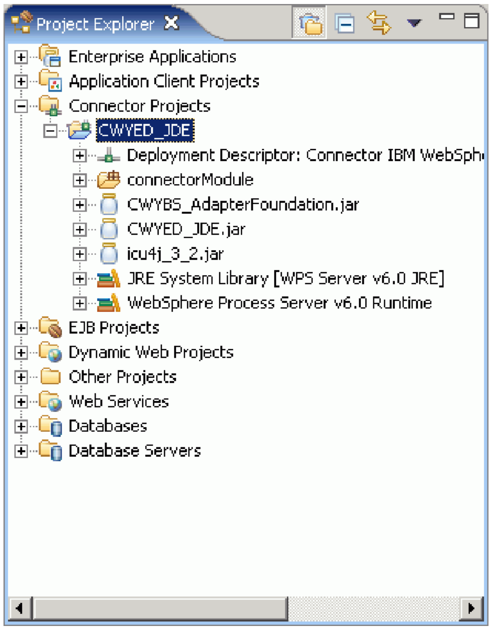
EAR project: CWYED_JDEEAR

Figure 38. Clearing the Add module to an EAR project check box

   Notice that the EAR project field becomes unavailable after you remove the check mark.
7. Click **Finish**.

Result

A new J2EE adapter project, named CWYED_JDE, is created. To see its contents, expand **CWYED_JDE**.



- Enterprise Applications
- Application Client Projects
- Connector Projects
  - CWYED_JDE
    - Deployment Descriptor: Connector IBM WebSph
    - connectorModule
    - CWYBS_AdapterFoundation.jar
    - CWYED_JDE.jar
    - icu4j_3_2.jar
    - JRE System Library [WPS Server v6.0 JRE]
    - WebSphere Process Server v6.0 Runtime
- EJB Projects
- Dynamic Web Projects
- Other Projects
- Web Services
- Databases
- Database Servers

Figure 39. The CWYED_JDE connector project in the Project Explorer window

Adding external software dependencies

To add the required external dependency files to the adapter project, you first import the files to the connectorModule folder inside the adapter project, then you copy the files from the connectorModule folder of the adapter project to the project classpath.

Before you begin

Create the adapter project in WebSphere Integration Developer. Also, obtain the software dependency files. For information about obtaining the software dependency files, refer to the following JD Edwards EnterpriseOne document: *PeopleSoft EnterpriseOne Tools Connectors PeopleBook*.

About this task

The JD Edwards EnterpriseOne application requires that you add external software dependencies to the adapter project. These software dependencies enable the Adapter for JD Edwards EnterpriseOne to communicate with the JD Edwards EnterpriseOne environment. For a list of all software dependencies, refer to "External software dependencies" on page 109 in the Reference section.

How to perform this task

1. Import the JD Edwards EnterpriseOne software dependency files to the connectorModule folder inside the adapter project.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.
   d. In the ″File system″ window, click **Browse**, then navigate to the directory that contains the JD Edwards EnterpriseOne software dependencies and configuration files, then click **OK**.
   e. To select all of the files that are displayed in the right pan of the ″File system″ window, click **Select All** , then click **Finish**.
2. Import the following .jar files into the connectorModule folder: ffdcSupport.jar and aspectjrt.jar.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.
   d. In the ″File system″ window, click **Browse** to navigate to the following directory:*WebSphere_Integration_Developer_Installation_Directory*\runtimes\bi_v6\lib.

      Note: If you accepted the default installation directory when you installed WebSphere Integration Developer, the .jar files are located in the following directory: C:\Program Files\IBM\WebSphere\ID\6.0\runtimes\bi_v6\lib.
3. Add the software dependency files from the connectorModule folder to the adapter project classpath.
   a. Right-click the adapter project, then select **Properties**.
   b. In the Properties for CWYED_JDE window, select **Java Build Path** from the left pane.

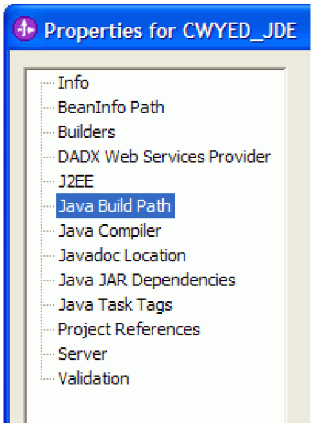
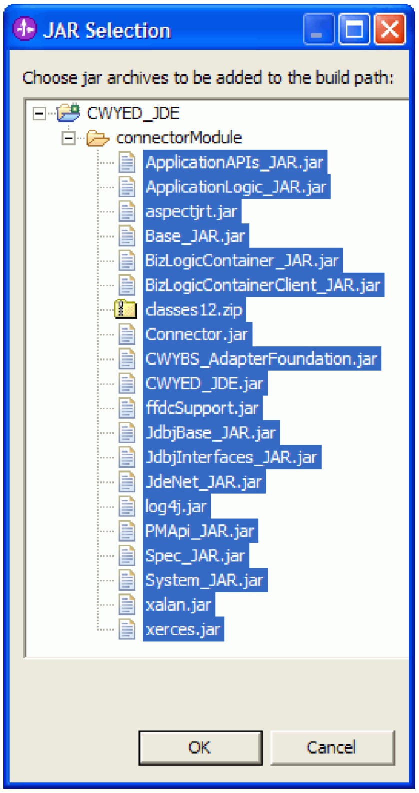


Figure 40. Selecting Java Build Path

c. In the Libraries page of the right pane, click **Add JARs**.

d. In the JAR Selection window, expand the adapter project folder (CWYED_JDE), then expand the connectorModule folder.

e. Highlight all of the JAR files listed under the connectorModule folder, then click **OK**.



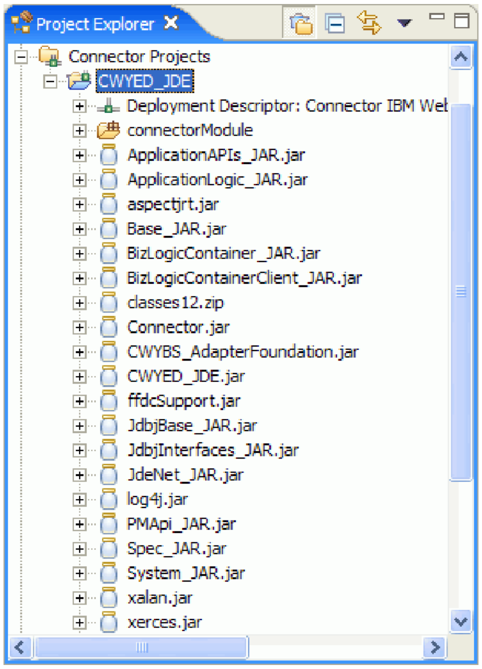
Choose jar archives to be added to the build path:

Figure 41. Selecting JAR files

f. In the Properties window, click **OK**.

Result

The external dependencies are displayed in the adapter project folder.



Project Explorer

- Connector Projects
  - CWYED_JDE
    - Deployment Descriptor: Connector IBM Wel
    - connectorModule
    - ApplicationAPIs_JAR.jar
    - ApplicationLogic_JAR.jar
    - aspectjrt.jar
    - Base_JAR.jar
    - BizLogicContainer_JAR.jar
    - BizLogicContainerClient_JAR.jar
    - classes12.zip
    - Connector.jar
    - CWYBS_AdapterFoundation.jar
    - CWYED_JDE.jar
    - ffdcSupport.jar
    - JdbjBase_JAR.jar
    - JdbjInterfaces_JAR.jar
    - JdeNet_JAR.jar
    - log4j.jar
    - PMApi_JAR.jar
    - Spec_JAR.jar
    - System_JAR.jar
    - xalan.jar
    - xerces.jar

Figure 42. Adapter project folder showing the external dependency files

What to do next

Configure the adapter. The first step in configuring the adapter is to specify information about the JD Edwards EnterpriseOne environment, so that the enterprise service discovery wizard can connect to it.

Note: Depending on which version of JD Edwards EnterpriseOne you are using, you may be required to configure the following files before configuring the adapter:

- jdeinterop.ini
- jdelog.properties
- jdbj.ini
- tnsnames.ora

For instructions on configuring these files, refer to the following JD Edwards EnterpriseOne documents:

- *PeopleSoft EnterpriseOne Tools Connectors PeopleBook* (for jdeinterop.ini and jdelog.properties files)
- *PeopleSoft EnterpriseOne Tools HTML Server Installation PeopleBook* (for jdbj.ini and tnsnames.ora files)

Configuring the adapter for outbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the GetEffectiveAddress business function and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the JD Edwards EnterpriseOne environment, specify such information as the user name and password you use to access the server as well as the environment name and role of the user.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the JD Edwards EnterpriseOne environment and discover its business objects and services.

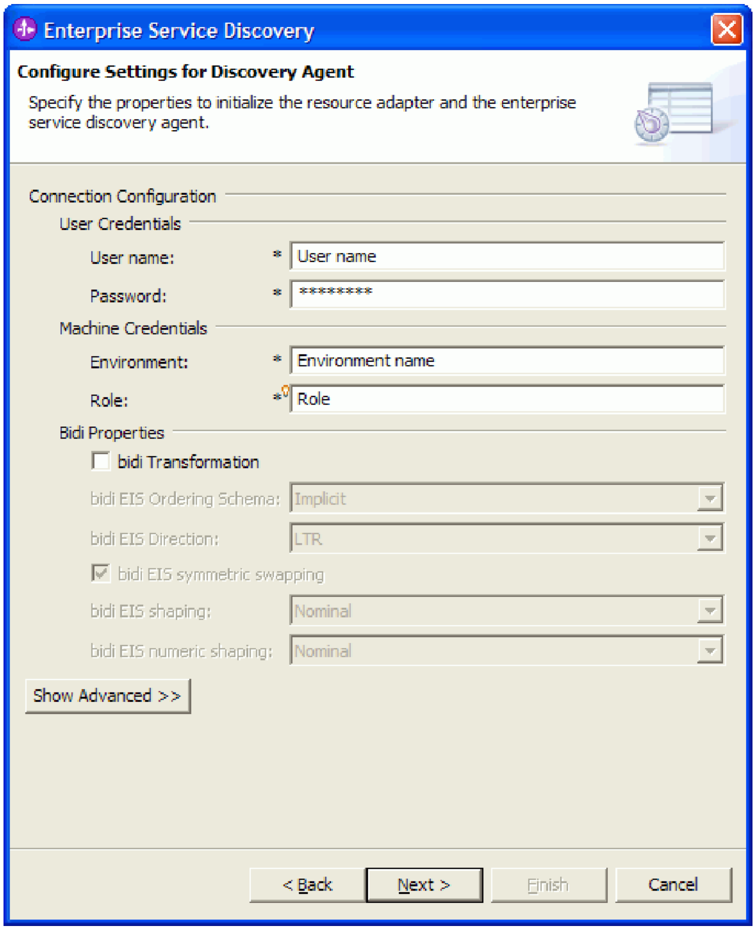
To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration perspective by clicking **Window** → **Open Perspective** → **Other**.
   b. In the Select Perspective window, select **Business Integration**, then click **OK**.
   c. Select **File** → **New** → **Enterprise Service Discovery**.
2. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for JD Edwards EnterpriseOne (version 6.0.2) from the CWYED_JDE Connector Project** , then click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the JD Edwards EnterpriseOne application you used the last time you ran the enterprise service discovery wizard.
3. Specify the configuration properties to initialize the discovery agent.

   Note: Properties marked with an asterisk (*) are required.
   a. Type the name and password you use to access the JD Edwards EnterpriseOne system.
   b. Type the environment name of your JD Edwards EnterpriseOne system.
   c. Type the role name you use to access the JD Edwards EnterpriseOne system.



Enterprise Service Discovery

Configure Settings for Discovery Agent

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

Connection Configuration

User Credentials

User name: * User name

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

Machine Credentials

Environment: * Environment name

Role: * Role

Bidi Properties

□ bidi Transformation

bidi EIS Ordering Schema: Implicit

bidi EIS Direction: LTR

☑ bidi EIS symmetric swapping

bidi EIS shaping: Nominal

bidi EIS numeric shaping: Nominal

Figure 43. The Configure Settings for Discovery Agent window

4. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See “Settings for controlling bidirectional transformation” on page 115 for more information about these properties.
5. **Optional:** To change the logging level, perform the following steps:
   a. At the bottom of the window, click **Show Advanced**.
   b. Set the **Logging Level**.

      In a test environment, select **FINEST**, which provides the highest level of logging. In a production environment, choose a level lower than **FINEST** to optimize the logging process.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the JD Edwards EnterpriseOne environment, using the information you provided, then the Find and Discover Enterprise Services window opens.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to discover business objects and services in the JD Edwards EnterpriseOne environment.

Selecting the business objects and services

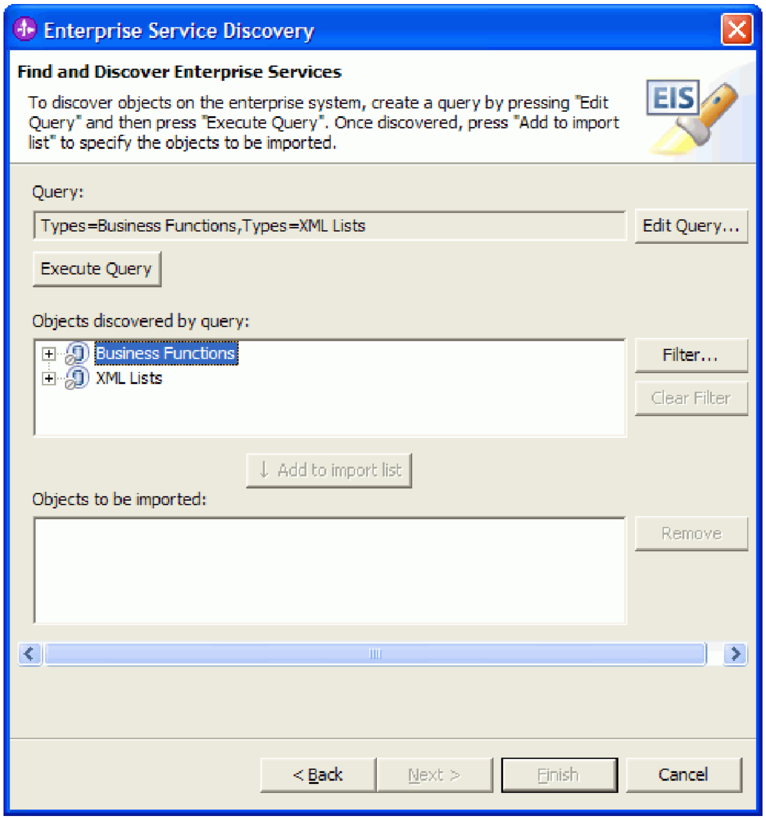
To select the GetEffectiveAddress business function to use for outbound processing, you provide information in the enterprise service discovery wizard.

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

How to perform this task

1. In the Find and Discover Enterprise Services window of the enterprise services discovery wizard, click **Execute Query**. The results of the query are displayed in the "Objects discovered by query" field.



Enterprise Service Discovery

Find and Discover Enterprise Services

To discover objects on the enterprise system, create a query by pressing "Edit Query" and then press "Execute Query". Once discovered, press "Add to import list" to specify the objects to be imported.

Query: Types=Business Functions,Types=XML Lists

Objects discovered by query:
- Business Functions
- XML Lists

Objects to be imported:

Figure 44. Results of executing a query in the Find and Discover Enterprise Services window of the enterprise service discovery wizard

2. Navigate to the **GetEffectiveAddress** business function by expanding the following nodes: **Business Functions** → **CFIN** → **B0100033**.
3. Click **Add to import list**.
4. In the Configuration Parameters for GetEffectiveAddress window, keep **GetEffectiveAddress** in the Business Object Name field, then click **OK**. The business object, GetEffectiveAddress, is displayed in the "Objects to be imported" field.

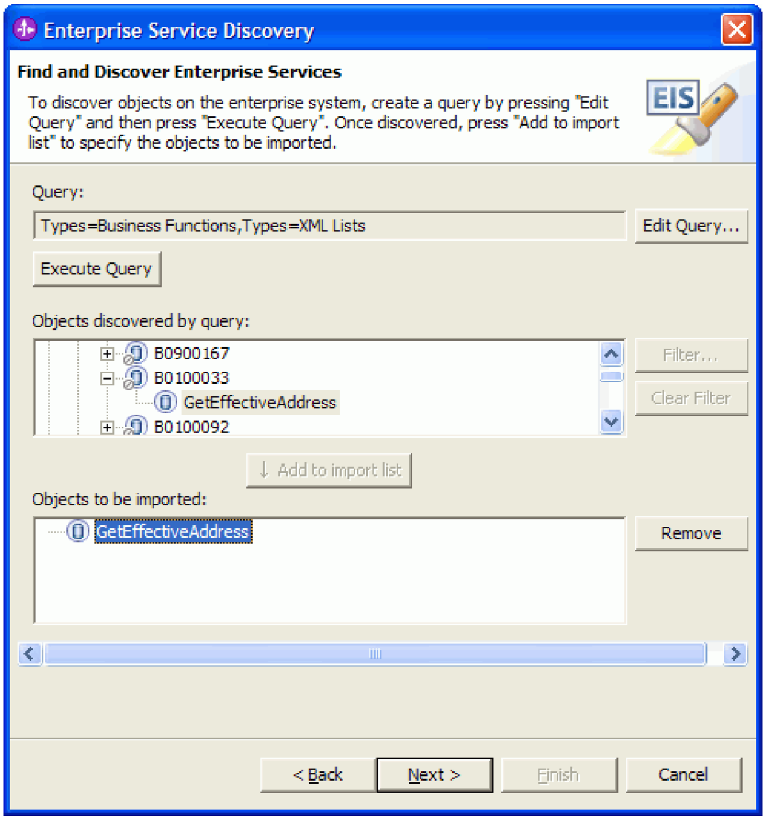

Enterprise Service Discovery
Find and Discover Enterprise Services
To discover objects on the enterprise system, create a query by pressing "Edit Query" and then press "Execute Query". Once discovered, press "Add to import list" to specify the objects to be imported.
EIS
Query:
Types=Business Functions,Types=XML Lists
Edit Query...
Execute Query
Objects discovered by query:
B0900167
B0100033
GetEffectiveAddress
B0100092
Filter...
Clear Filter
Add to import list
Objects to be imported:
GetEffectiveAddress
Remove
< Back
Next >
Finish
Cancel

Figure 45. Adding GetEffectiveAddress to the import list

5. Click **Next**.

Result

The business function, GetEffectiveAddress, is imported from the JD Edwards EnterpriseOne application to the enterprise service discovery wizard. The Configure Objects window opens.

What to do next

Specify a name for the business object and the directory in which it should be stored.

Configuring the selected objects

To configure the business function, you specify a name and the directory in which it should be stored.

Before you begin

Make sure you have selected and imported the GetEffectiveAddress business function.

How to perform this task

1. In the Configure Objects window of the enterprise service discovery wizard, select the **Add Container BO** check box. The window expands to include fields for container business object information.

2. In the **Container Business Object Name** field, type **GetEffectiveAddressContainer**.
3. Click **Add** next to the Business Function for Retrieve field.
4. In the Add window, select **GetEffectiveAddress**, leaving the Value field empty, then click **OK**. The GetEffectiveAddress business function is displayed in the **Business Functions for Retrieve** field.
5. Click **Next**.

Result

You have associated the Retrieve operation with the business function and given it the following name: GetEffectiveAddressContainer. The Generate Artifacts window opens.

What to do next

Generate a deployable module that includes the adapter and the business function.

Generating artifacts

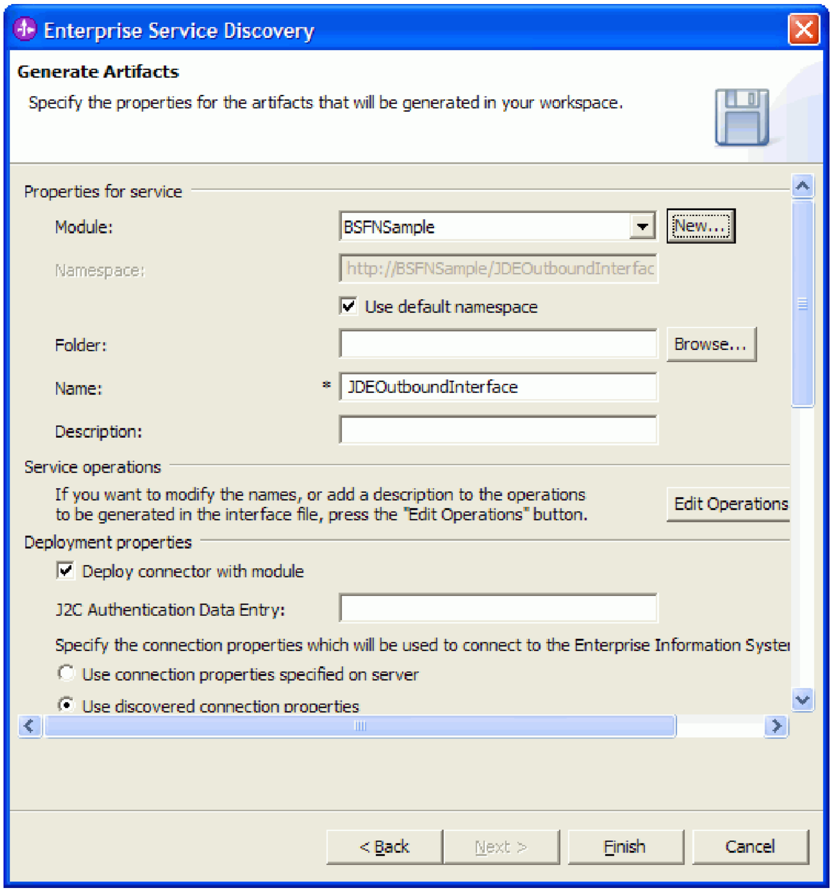
To generate the module, which stores the outbound artifacts that are deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create a new module, include the adapter project in the module, and specify an alias used to authenticate the caller in the JD Edwards EnterpriseOne environment.

Before you begin

Make sure you have configured the GetEffectiveAddresss business function. The Generate Artifacts window should be open.

How to perform this task

1. In the Generate Artifacts window, click **New** next to the Module field to create a new module.
2. In New Integration Project window, select **Create a module project**, then click **Next**.
3. In the New Module window, type **BSFNSample** in the Module Name field, then click **Finish**.
4. In the Generate Artifacts window, select the **Use discovered connection properties** option.



Enterprise Service Discovery

Generate Artifacts

Specify the properties for the artifacts that will be generated in your workspace.

Properties for service

Module: BSFNSample

Namespace: http://BSFNSample/JDEOutboundInterfac

☑ Use default namespace

Folder:

Name: * JDEOutboundInterface

Description:

Service operations

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

Deployment properties

☑ Deploy connector with module

J2C Authentication Data Entry:

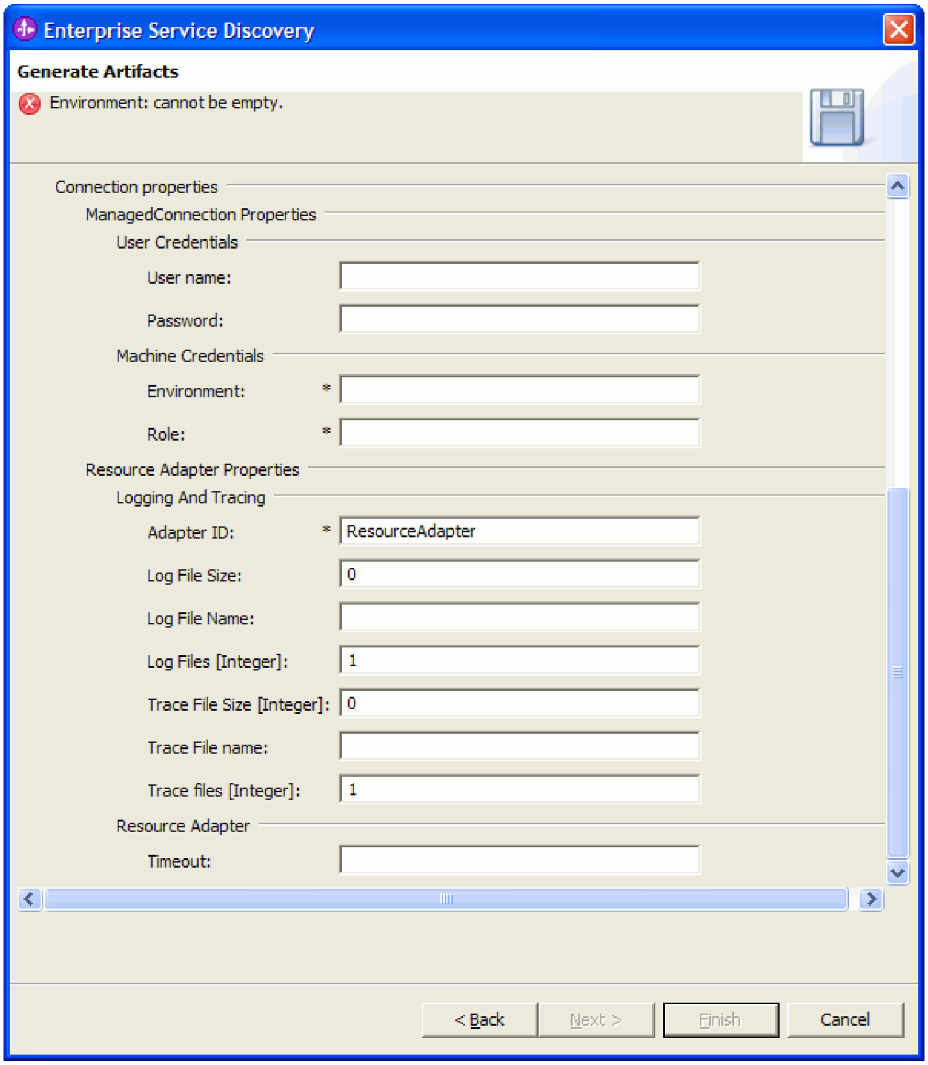
Specify the connection properties which will be used to connect to the Enterprise Information Syste

○ Use connection properties specified on server

● Use discovered connection properties

Figure 46. Generate Artifacts window (top portion) of the enterprise service discovery wizard

5. Scroll down to enter information in the **Environment** and **Role** fields. You are required to fill out these two fields, as indicated by the asterisk (*). For more information about the **Environment** and **Role** properties, see “Managed (J2C) connection factory properties” on page 114.



Enterprise Service Discovery

Generate Artifacts

Environment: cannot be empty.

Connection properties

ManagedConnection Properties

User Credentials

User name:

Password:

Machine Credentials

Environment: *

Role: *

Resource Adapter Properties

Logging And Tracing

Adapter ID: * ResourceAdapter

Log File Size: 0

Log File Name:

Log Files [Integer]: 1

Trace File Size [Integer]: 0

Trace File name:

Trace files [Integer]: 1

Resource Adapter

Timeout:

Figure 47. Generate Artifacts (bottom portion) window of the enterprise service discovery wizard

6. Click **Finish**.

Result

The BSFNSample module is displayed in the J2EE perspective of WebSphere Integration Developer with "App" appended to its name, indicating that the module is a deployable application.

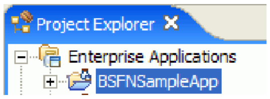


Figure 48. BSFNSample module (BSFNSampleApp) displayed in the Project Explorer window (J2EE perspective) of WebSphere Integration Developer

What to do next

Deploy the module for testing.

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server or WebSphere Enterprise Service Bus, you add BSFNSampleApp to the server. If the server is not already started, it starts when you add the module.

Before you begin

You must have exported BSFNSampleApp as an EAR file before installing it on the server.

About this task

You use WebSphere Integration Developer to access the server and deploy the module to the test environment of the server.

To deploy the module, use the following procedure.

How to perform this task

1. Select the test environment server.
   a. In WebSphere Integration Developer, click the **Server** tab.
   b. Right-click the server, then click **Add and remove projects**.

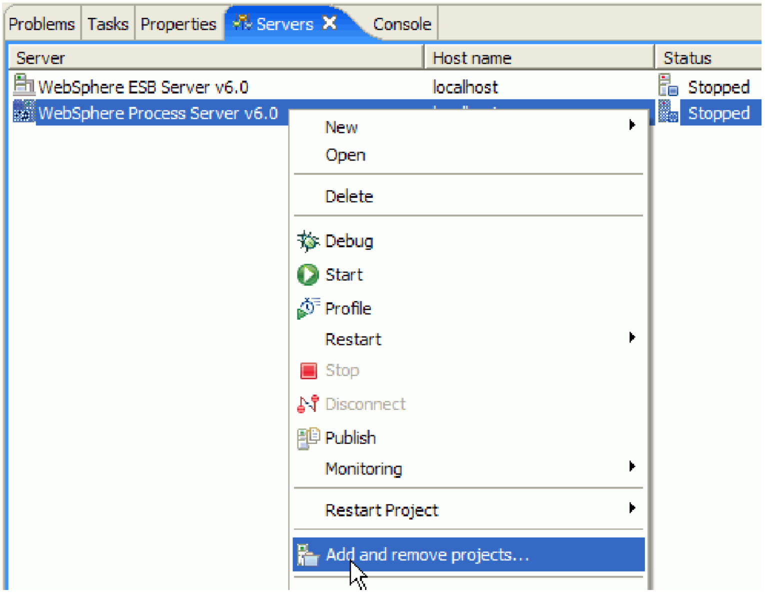


Figure 49. Selecting **Add and remove projects**

2. In the Add and Remove Projects window, select **BSFNSampleApp**, then click **Add**.

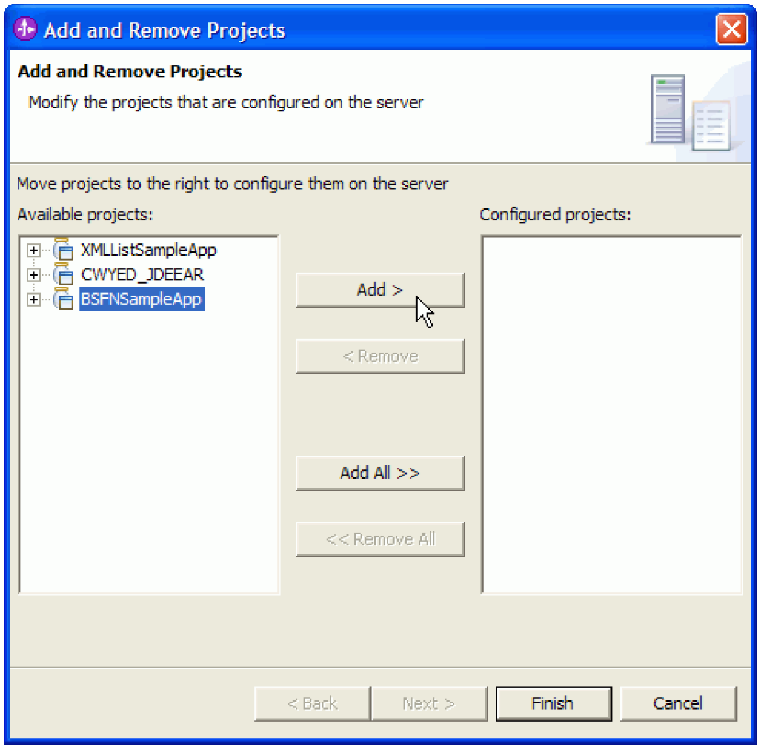


Figure 50. Add and Remove Projects window

3. When **BSFNSampleApp** is displayed in the Configured projects list, click **Finish**.

Result

Your server is started, if it was not already started. Then, the Console tab displays the status of the module as it is deployed.

Testing the module

Test the module to make sure you can retrieve data from the JD Edwards EnterpriseOne server. You enter search criteria, and the data meeting that criteria is returned.

Before you begin

Make sure you have deployed the BSFNSampleApp module to the server.

How to perform this task

1. In the Business Integration perspective of WebSphere Integration Developer, right-click BSFNSample, then select **Test** > **Test Module**.

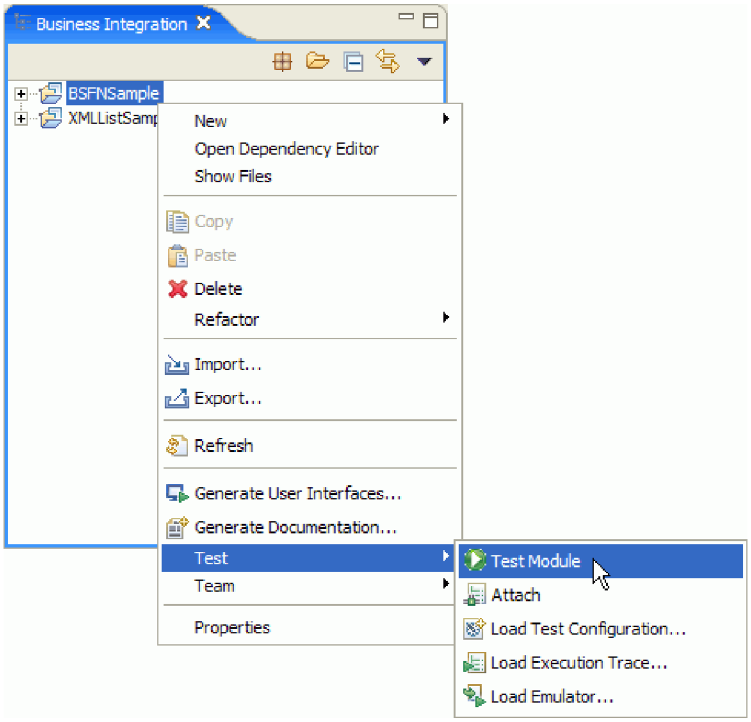
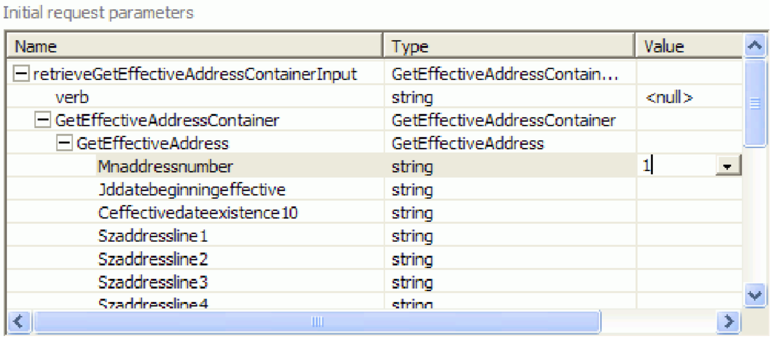


Figure 51. Selecting the BSFNSample module to test

2. In the Initial request parameters table of the Events page, populate the **Mnaddressnumber** attribute of the **GetEffectiveAddress** business object with a number of your choice. For example, type **1**.

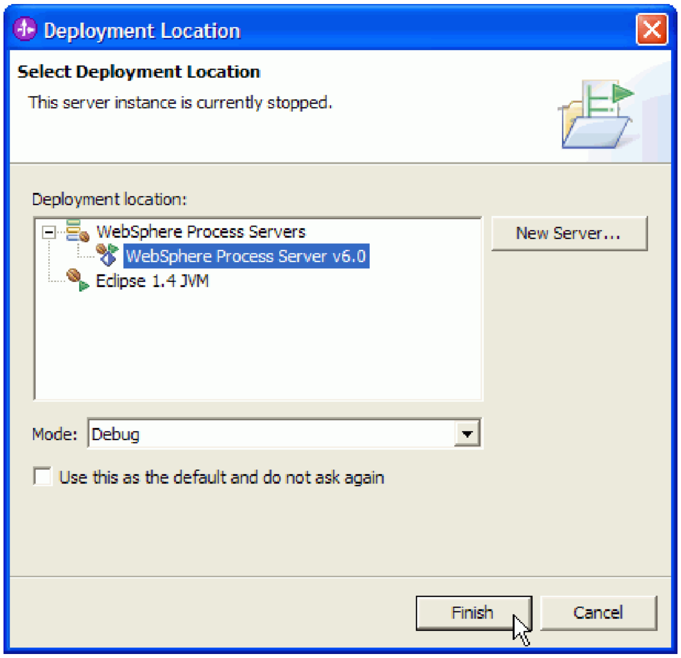


Initial request parameters

Name	Type	Value
⊟ retrieveGetEffectiveAddressContainerInput	GetEffectiveAddressContain...	
verb	string	<null>
⊟ GetEffectiveAddressContainer	GetEffectiveAddressContainer	
⊟ GetEffectiveAddress	GetEffectiveAddress	
Mnaddressnumber	string	1
Jddatebeginningeffective	string	
Ceffectivedateexistence10	string	
Szaddressline1	string	
Szaddressline2	string	
Szaddressline3	string	
Szaddressline4	string	

Figure 52. Adding a value to the Mnaddressnumber attribute in the Initial request parameters window

3. Execute the service by clicking **Continue**.
4. In the Select Deployment Location window, select the server, then click **Finish**.



Select Deployment Location

This server instance is currently stopped.

Deployment location:

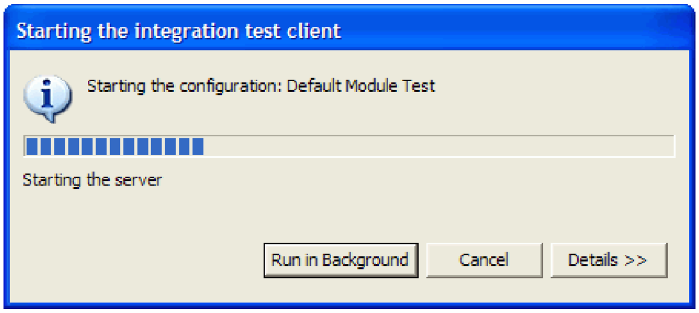
- WebSphere Process Servers
  - WebSphere Process Server v6.0
- Eclipse 1.4 JVM

Mode: Debug

□ Use this as the default and do not ask again

Figure 53. Deployment Location window

5. In the Starting the integrated test client window, click **Run in Background** if you prefer to run the server in the background.



Starting the configuration: Default Module Test

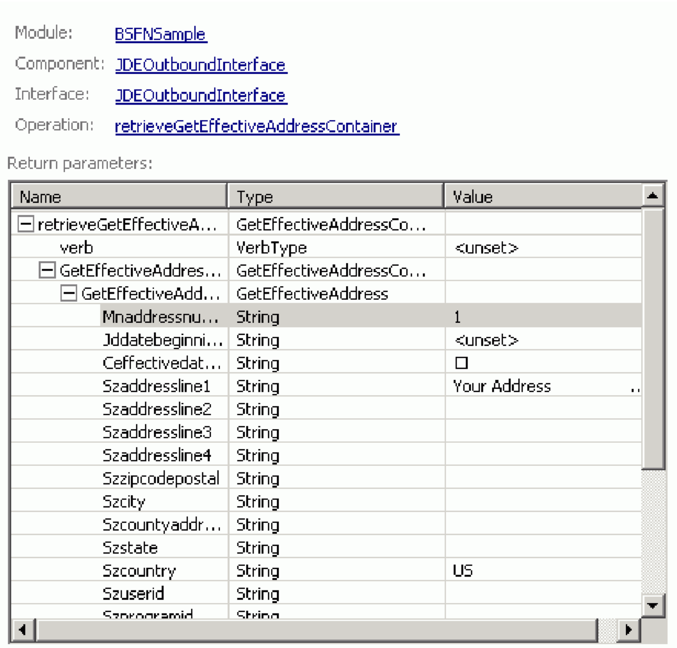
Starting the server

Figure 54. Starting the integration test client window

6. Click **Finish**. The integration test client invokes BSFNSampleApp.

Result

The output of the service and the data in the JD Edwards EnterpriseOne application match the data you entered for the Mnaddressnumber attribute of the GetEffectiveAddress business object. This indicates that the BSNFSampleApp test successfully completed.


Module: BSFNSample

Component: JDEOutboundInterface

Interface: JDEOutboundInterface

Operation: retrieveGetEffectiveAddressContainer

Return parameters:

Name	Type	Value
⊟ retrieveGetEffectiveA...	GetEffectiveAddressCo...	
verb	VerbType	<unset>
⊟ GetEffectiveAddres...	GetEffectiveAddressCo...	
⊟ GetEffectiveAdd...	GetEffectiveAddress	
Mnaddressnu...	String	1
Jddatebeginni...	String	<unset>
Ceffectivedat...	String	□
Szaddressline1	String	Your Address ..
Szaddressline2	String	
Szaddressline3	String	
Szaddressline4	String	
Szzipcodepostal	String	
Szcity	String	
Szcountyaddr...	String	
Szstate	String	
Szcountry	String	US
Szuserid	String	
Szprogramid	String	

Figure 55. Test results in the Return parameters window

Tutorial 2: Retrieving data from the JD Edwards EnterpriseOne table using an XML List query

To create a module that uses an XML List query to retrieve data from a JD Edwards EnterpriseOne database table, you create an adapter project, use the enterprise service discovery wizard to generate the XML List query, and create a module that contains WebSphere Adapter for JD Edwards EnterpriseOne and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create the authentication alias on the server, use the administrative console. From the administrative console, configure the global security and set the password for the authentication alias, which is used to process outbound requests.

Before you begin

You must have access to the administrative console of the either WebSphere Process Server or WebSphere Enterprise Service Bus.

How to perform this task

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

Note: The user ID and password that you type will be used to establish a connection to the enterprise information system for outbound processing.

5. Click **OK**, click **Save**, and then click **Save** again.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with a JD Edwards EnterpriseOne service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) contains the adapter itself plus other related artifacts. You create the project by importing the RAR file, which was copied to your local file system during the adapter installation, into WebSphere Integration Developer.

About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

1. In WebSphere Integration Developer, switch to the J2EE perspective:
   a. Click **Window** → **Open Perspective** → **Other**.
   b. Click **J2EE**.

      If **J2EE** is not displayed, select the **Show all** check box, click **J2EE**, and click **OK**.

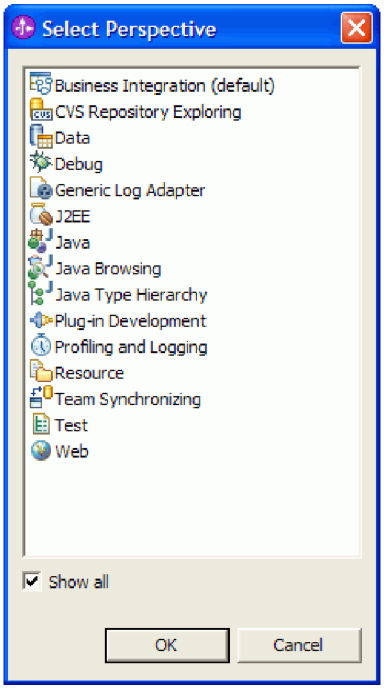
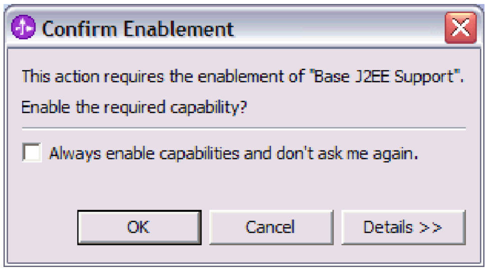


Figure 56. Selecting J2EE from the Select Perspective list

c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.


Confirm Enablement

This action requires the enablement of "Base J2EE Support". Enable the required capability?

☐ Always enable capabilities and don't ask me again.

Figure 57. The Confirm Enablement window

d. Click **OK**.

2. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

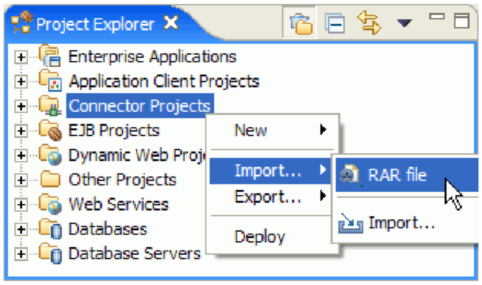
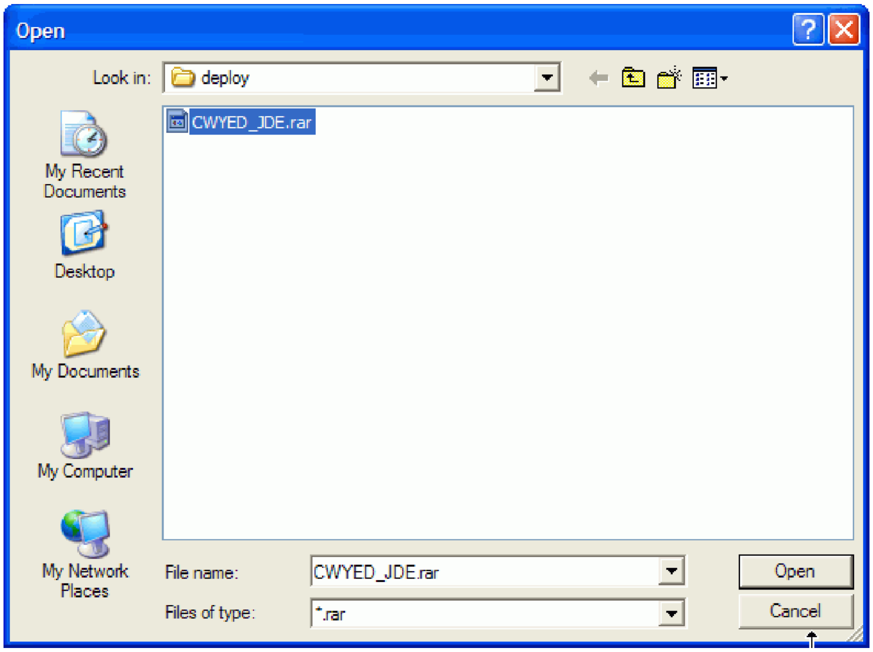


Figure 58. Importing the RAR file

3. Find the RAR file on your local file system by clicking **Browse** and navigating to the directory in which Adapter for JD Edwards EnterpriseOne was installed.



Open

Look in: deploy

CWYED_JDE.rar

File name: CWYED_JDE.rar

Files of type: *.rar

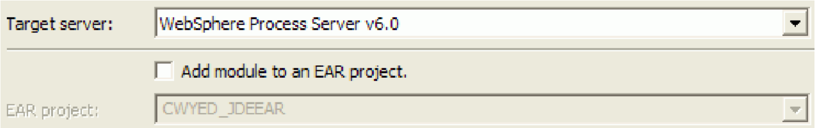
Figure 59. Selecting the RAR file from the installation directory

4. Accept the default setting (**CWYED_JDE.rar**) for **Connector project**.

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

   Note: If a project named CWYED_JDE.rar already exists in this workspace, the name in the Connector project field has a number appended to it (for example, CWYED_JDE1).
5. Accept the default value in the **Target server** field.

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



Target server: WebSphere Process Server v6.0

□ Add module to an EAR project.

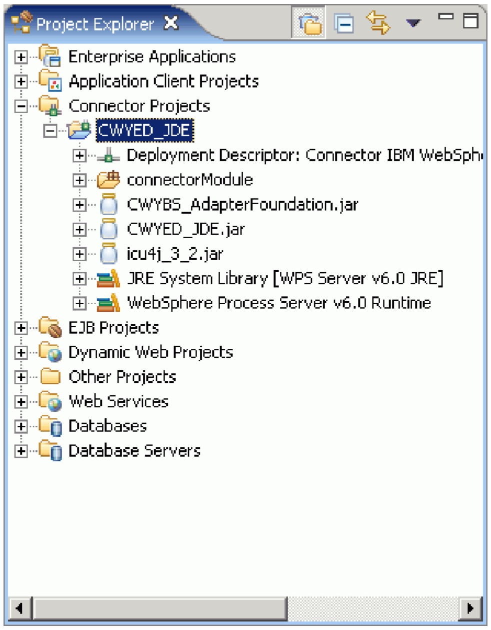
EAR project: CWYED_JDEEAR

Figure 60. Clearing the Add module to an EAR project check box

   Notice that the EAR project field becomes unavailable after you remove the check mark.
7. Click **Finish**.

Result

A new J2EE adapter project, named CWYED_JDE, is created. To see its contents, expand **CWYED_JDE**.



Project Explorer

- Enterprise Applications
- Application Client Projects
- Connector Projects
  - CWYED_JDE
    - Deployment Descriptor: Connector IBM WebSph
    - connectorModule
    - CWYBS_AdapterFoundation.jar
    - CWYED_JDE.jar
    - icu4j_3_2.jar
    - JRE System Library [WPS Server v6.0 JRE]
    - WebSphere Process Server v6.0 Runtime
- EJB Projects
- Dynamic Web Projects
- Other Projects
- Web Services
- Databases
- Database Servers

Figure 61. The CWYED_JDE connector project in the Project Explorer window

Adding external software dependencies

To add the required external dependency files to the adapter project, you first import the files to the connectorModule folder inside the adapter project, then you copy the files from the connectorModule folder of the adapter project to the project classpath.

Before you begin

Create the adapter project in WebSphere Integration Developer. Also, obtain the software dependency files. For information about obtaining the software dependency files, refer to the following JD Edwards EnterpriseOne document: *PeopleSoft EnterpriseOne Tools Connectors PeopleBook*.

About this task

The JD Edwards EnterpriseOne application requires that you add external software dependencies to the adapter project. These software dependencies enable the Adapter for JD Edwards EnterpriseOne to communicate with the JD Edwards EnterpriseOne environment. For a list of all software dependencies, refer to "External software dependencies" on page 109 in the Reference section.

How to perform this task

1. Import the JD Edwards EnterpriseOne software dependency files to the connectorModule folder inside the adapter project.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.

d. In the "File system" window, click **Browse**, then navigate to the directory that contains the JD Edwards EnterpriseOne software dependencies and configuration files, then click **OK**.

e. To select all of the files that are displayed in the right pan of the "File system" window, click **Select All** , then click **Finish**.

2. Import the following .jar files into the connectorModule folder: ffdcSupport.jar and aspectjrt.jar.

a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.

b. Right-click the **connectorModule** folder, then select **Import**.

c. In the Import window, select **File system** from the list of import sources, then click **Next**.

d. In the "File system" window, click **Browse** to navigate to the following directory:*WebSphere_Integration_Developer_Installation_Directory*\runtimes\bi_v6\lib.

Note: If you accepted the default installation directory when you installed WebSphere Integration Developer, the .jar files are located in the following directory: C:\Program Files\IBM\WebSphere\ID\6.0\runtimes\bi_v6\lib.

3. Add the software dependency files from the connectorModule folder to the adapter project classpath.

a. Right-click the adapter project, then select **Properties**.

b. In the Properties for CWYED_JDE window, select **Java Build Path** from the left pane.

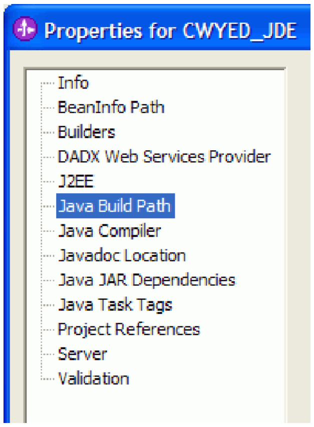
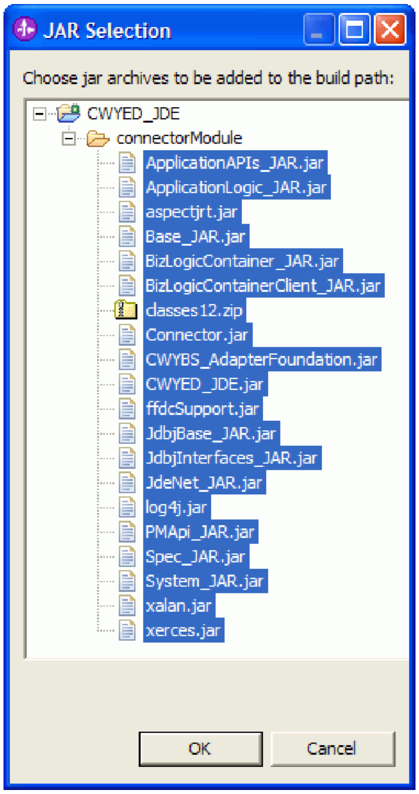


Figure 62. Selecting Java Build Path

c. In the Libraries page of the right pane, click **Add JARs**.

d. In the JAR Selection window, expand the adapter project folder (CWYED_JDE), then expand the connectorModule folder.

e. Highlight all of the JAR files listed under the connectorModule folder, then click **OK**.



Choose jar archives to be added to the build path:

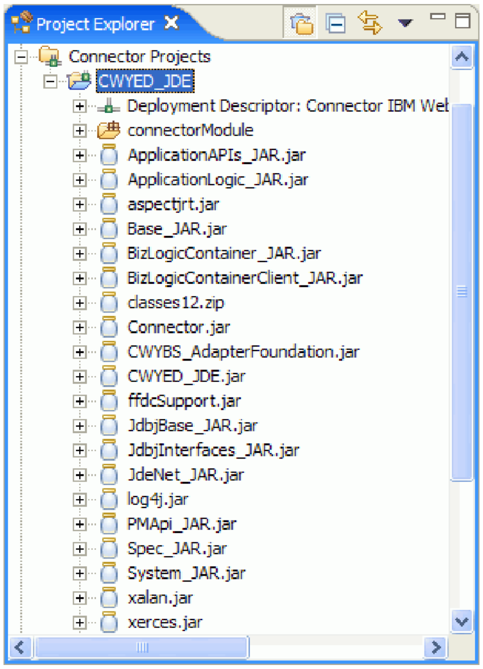
```
CWYED_JDE
  connectorModule
    ApplicationAPIs_JAR.jar
    ApplicationLogic_JAR.jar
    aspectjrt.jar
    Base_JAR.jar
    BizLogicContainer_JAR.jar
    BizLogicContainerClient_JAR.jar
    classes12.zip
    Connector.jar
    CWYBS_AdapterFoundation.jar
    CWYED_JDE.jar
    ffdcSupport.jar
    JdbjBase_JAR.jar
    JdbjInterfaces_JAR.jar
    JdeNet_JAR.jar
    log4j.jar
    PMApi_JAR.jar
    Spec_JAR.jar
    System_JAR.jar
    xalan.jar
    xerces.jar
```

Figure 63. Selecting JAR files

f. In the Properties window, click **OK**.

Result

The external dependencies are displayed in the adapter project folder.



```
Project Explorer
Connector Projects
  CWYED_JDE
    Deployment Descriptor: Connector IBM Wel
    connectorModule
    ApplicationAPIs_JAR.jar
    ApplicationLogic_JAR.jar
    aspectjrt.jar
    Base_JAR.jar
    BizLogicContainer_JAR.jar
    BizLogicContainerClient_JAR.jar
    classes12.zip
    Connector.jar
    CWYBS_AdapterFoundation.jar
    CWYED_JDE.jar
    ffdcSupport.jar
    JdbjBase_JAR.jar
    JdbjInterfaces_JAR.jar
    JdeNet_JAR.jar
    log4j.jar
    PMApi_JAR.jar
    Spec_JAR.jar
    System_JAR.jar
    xalan.jar
    xerces.jar
```

Figure 64. Adapter project folder showing the external dependency files

What to do next

Configure the adapter. The first step in configuring the adapter is to specify information about the JD Edwards EnterpriseOne environment, so that the enterprise service discovery wizard can connect to it.

Note: Depending on which version of JD Edwards EnterpriseOne you are using, you may be required to configure the following files before configuring the adapter:

- jdeinterop.ini
- jdelog.properties
- jdbj.ini
- tnsnames.ora

For instructions on configuring these files, refer to the following JD Edwards EnterpriseOne documents:

- *PeopleSoft EnterpriseOne Tools Connectors PeopleBook* (for jdeinterop.ini and jdelog.properties files)
- *PeopleSoft EnterpriseOne Tools HTML Server Installation PeopleBook* (for jdbj.ini and tnsnames.ora files)

Configuring the adapter for outbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the XML List and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the JD Edwards EnterpriseOne environment, specify such information as the user name and password you use to access the server as well as the environment name and role of the user.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the JD Edwards EnterpriseOne environment and discover its business objects and services.

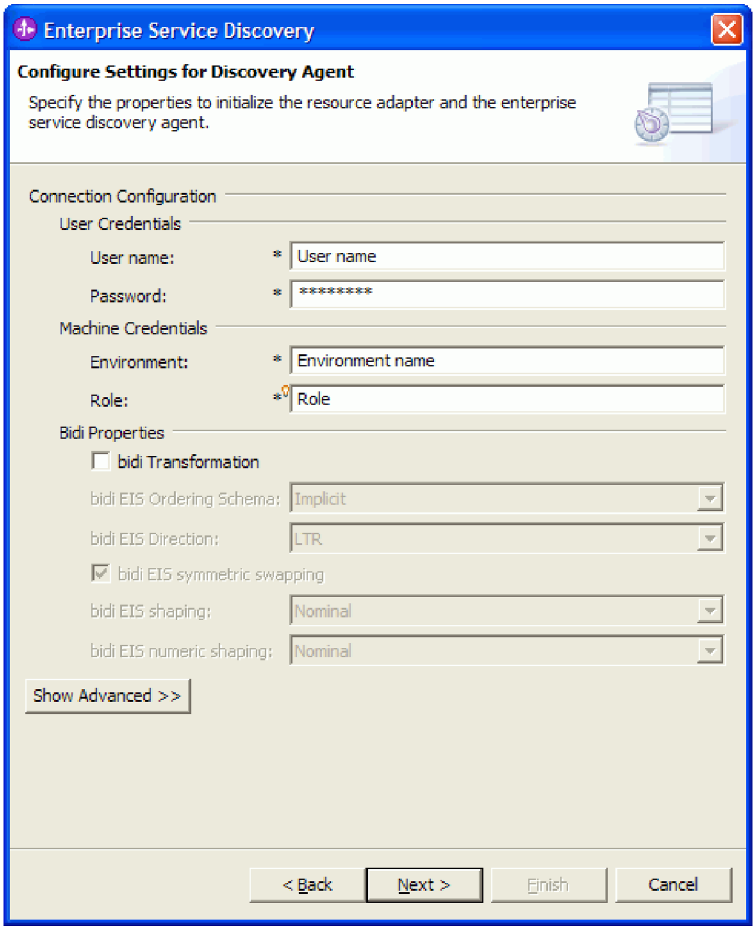
To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration perspective by clicking **Window** → **Open Perspective** → **Other**.
   b. In the Select Perspective window, select **Business Integration**, then click **OK**.
   c. Select **File** → **New** → **Enterprise Service Discovery**.
2. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for JD Edwards EnterpriseOne (version 6.0.2) from the CWYED_JDE Connector Project** , then click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the JD Edwards EnterpriseOne application you used the last time you ran the enterprise service discovery wizard.
3. Specify the configuration properties to initialize the discovery agent.

   Note: Properties marked with an asterisk (*) are required.
   a. Type the name and password you use to access the JD Edwards EnterpriseOne system.
   b. Type the environment name of your JD Edwards EnterpriseOne system.
   c. Type the role name you use to access the JD Edwards EnterpriseOne system.



Enterprise Service Discovery

Configure Settings for Discovery Agent

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

Connection Configuration

User Credentials

User name: * User name

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

Machine Credentials

Environment: * Environment name

Role: * Role

Bidi Properties

☐ bidi Transformation

bidi EIS Ordering Schema: Implicit

bidi EIS Direction: LTR

☑ bidi EIS symmetric swapping

bidi EIS shaping: Nominal

bidi EIS numeric shaping: Nominal

Figure 65. The Configure Settings for Discovery Agent window

4. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See “Settings for controlling bidirectional transformation” on page 115 for more information about these properties.
5. **Optional:** To change the logging level, perform the following steps:
   a. At the bottom of the window, click **Show Advanced**.
   b. Set the **Logging Level**.

      In a test environment, select **FINEST**, which provides the highest level of logging. In a production environment, choose a level lower than **FINEST** to optimize the logging process.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the JD Edwards EnterpriseOne environment, using the information you provided, then the Find and Discover Enterprise Services window opens.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to discover business objects and services in the JD Edwards EnterpriseOne environment.

Selecting the business objects and services

To select an XML List to use for outbound processing, you provide information in the enterprise service discovery wizard. .

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

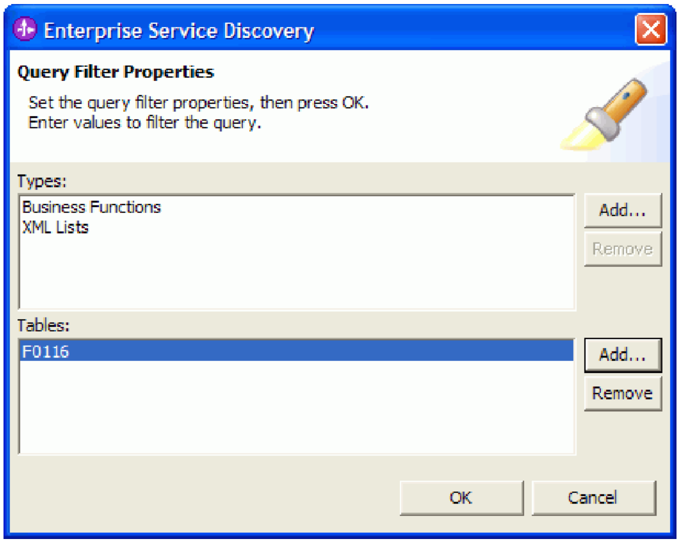
About this task

Specify search criteria that the enterprise service discovery wizard uses to discover XML Lists in the JD Edwards EnterpriseOne environment. Before you can run the query for XML Lists on the server, however, you must specify the table name so that enterprise service discovery wizard can retrieve metadata about that table. After executing the query, the enterprise service discovery wizard returns a list of XML Lists.

To specify the search criteria and select an XML Lists, use the following procedure.

How to perform this task

1. Specify the F0116 table to use in the XML List query.
   a. In the Find and Discover Enterprise Services window of the enterprise services discovery wizard, click **Edit Query**.
   b. In the Query Filter Properties window, click **Add** next to the Tables list.
   c. In the Add window that opens, type the `F0116`, then click **OK**. The F0116 table name is displayed in the Tables list.



Enterprise Service Discovery

Query Filter Properties

Set the query filter properties, then press OK.
Enter values to filter the query.

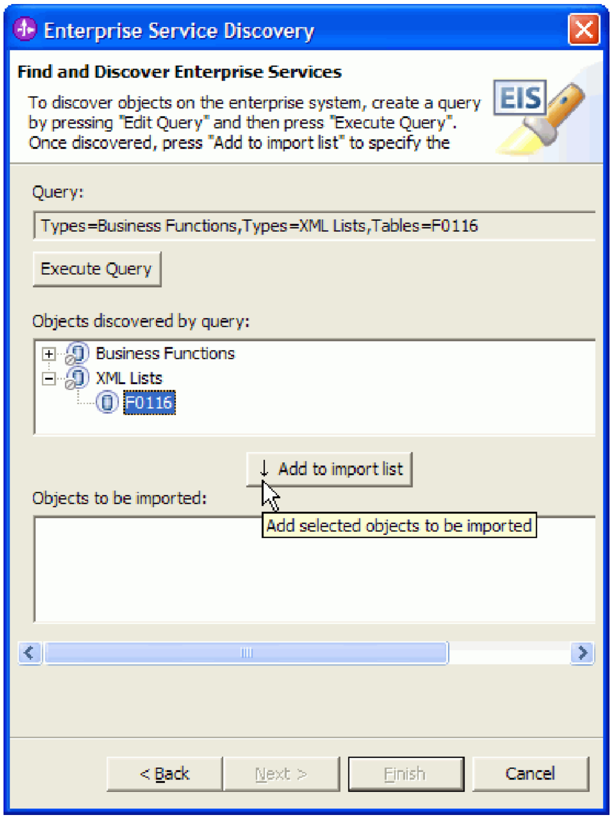
Types:
Business Functions
XML Lists

Tables:
F0116

Figure 66. Query Filter Properties window with the F0116 table added

   d. Click **OK**.
2. Execute the XML List query to find and discover the F0116 table in the JD Edwards EnterpriseOne application.
   a. In the Find and Discover Enterprise Services window, click **Execute Query**.

b. Expand the **XML Lists** node, navigate to the F0116 table, then click **Add to import list**.



Enterprise Service Discovery

Find and Discover Enterprise Services

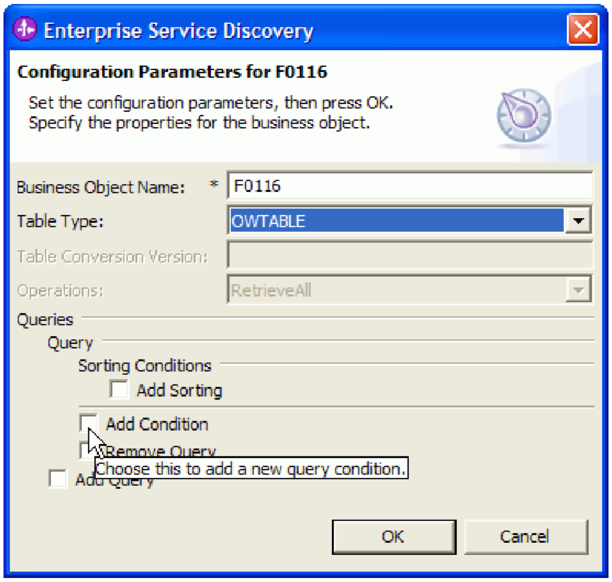
To discover objects on the enterprise system, create a query by pressing "Edit Query" and then press "Execute Query". Once discovered, press "Add to import list" to specify the

Query: Types=Business Functions,Types=XML Lists,Tables=F0116

Objects discovered by query:

Figure 67. Find and Discover Enterprise Services window, showing the query result of the **F0116** *table listed under* **XML Lists**

3. Configure the XML List query for import into the service description.
   a. In the Configuration Parameters for F0116 window, select **OWTABLE** from the Table Type list.
   b. Select **Add Condition**.



Enterprise Service Discovery

Configuration Parameters for F0116

Set the configuration parameters, then press OK.
Specify the properties for the business object.

Business Object Name: * F0116

Table Type: OWTABLE

Table Conversion Version:

Operations: RetrieveAll

Queries

Query

Sorting Conditions

□ Add Sorting

□ Add Condition

□ Remove Query

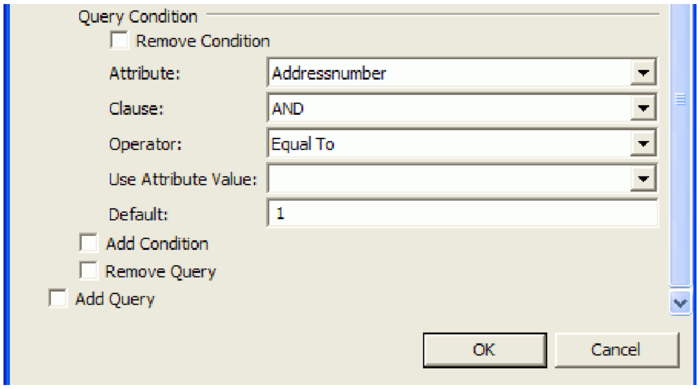
□ Add Query

Choose this to add a new query condition.

Figure 68. Selecting the ***OWTABLE*** *table type in the Configuration Parameters for F0116 window*

c. Scroll down to select **Addressnumber** in the Attribute list.

d. In the Default field, enter **1**.



Query Condition

□ Remove Condition

Attribute: Addressnumber

Clause: AND

Operator: Equal To

Use Attribute Value:

Default: 1

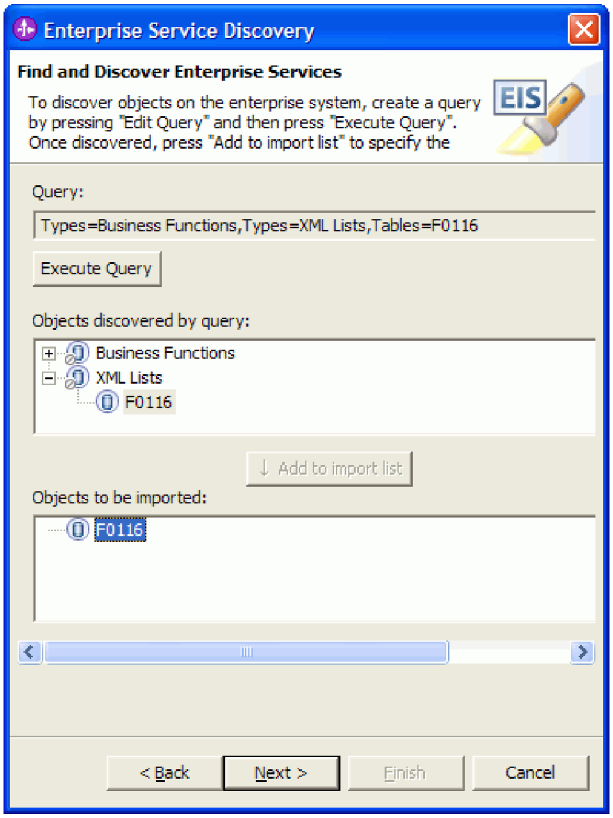
□ Add Condition

□ Remove Query

□ Add Query

Figure 69. Configuration Parameters for F0116 window, showing the ***Attribute*** *and* ***Default*** *fields*

e. Click **OK**. The configured XML List query business object is displayed in the "Objects to be imported" list of the Find and Discover Enterprise Services window.



Enterprise Service Discovery

Find and Discover Enterprise Services

To discover objects on the enterprise system, create a query by pressing "Edit Query" and then press "Execute Query". Once discovered, press "Add to import list" to specify the

Query:

Types=Business Functions,Types=XML Lists,Tables=F0116

Objects discovered by query:

Objects to be imported:

Figure 70. Find and Discover Enterprise Services window showing the F0116 table in the ***Objects to be imported*** *list*

4. Click **Next**.

Result

The XML List is imported from the JD Edwards EnterpriseOne application to the enterprise service discovery wizard. The Configure Objects window opens.

What to do next

Specify a name for the XML List and the directory in which it should be stored.

Configuring the selected objects

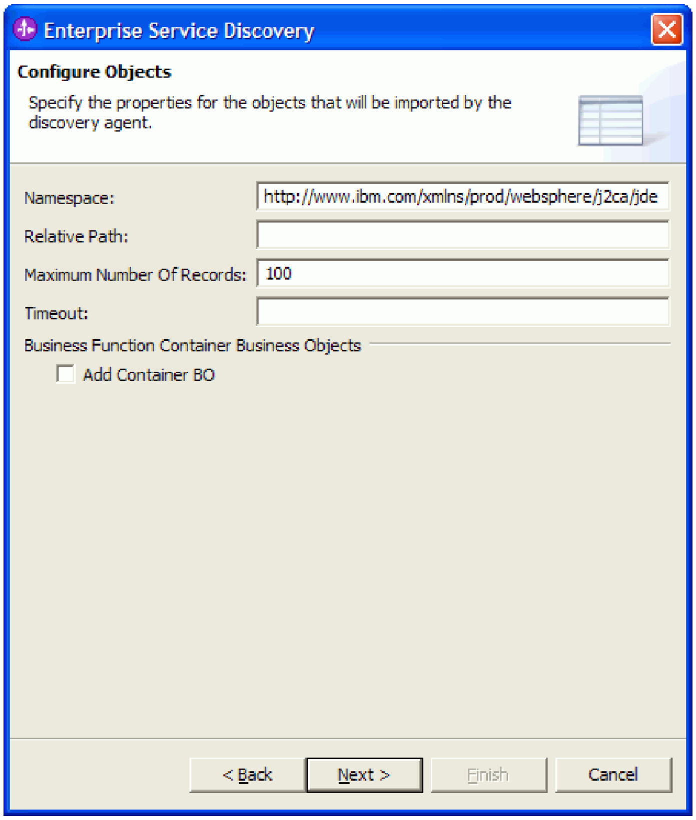
To configure the XML List, you specify a name and the directory in which it should be stored.

Before you begin

Make sure you have selected and imported the XML List.

How to perform this task

In the Configure Objects window of the enterprise service discovery wizard, accept the default values and click **Next**.



Configure Objects

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

Namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/jde

Relative Path:

Maximum Number Of Records: 100

Timeout:

Business Function Container Business Objects

□ Add Container BO

Figure 71. Configure Objects window of the enterprise services discovery wizard

Result

You have associated an operation (RetrieveAll) with the object. The Generate Artifacts window opens.

What to do next

Generate a deployable module that includes the adapter and the XML List.

Generating artifacts

To generate the module, which stores the outbound artifacts that are deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create a new module, include the adapter project in the module, and specify an alias used to authenticate the caller in the JD Edwards EnterpriseOne environment. In this tutorial, the module you create contains an XML List that contains the F0116 table.

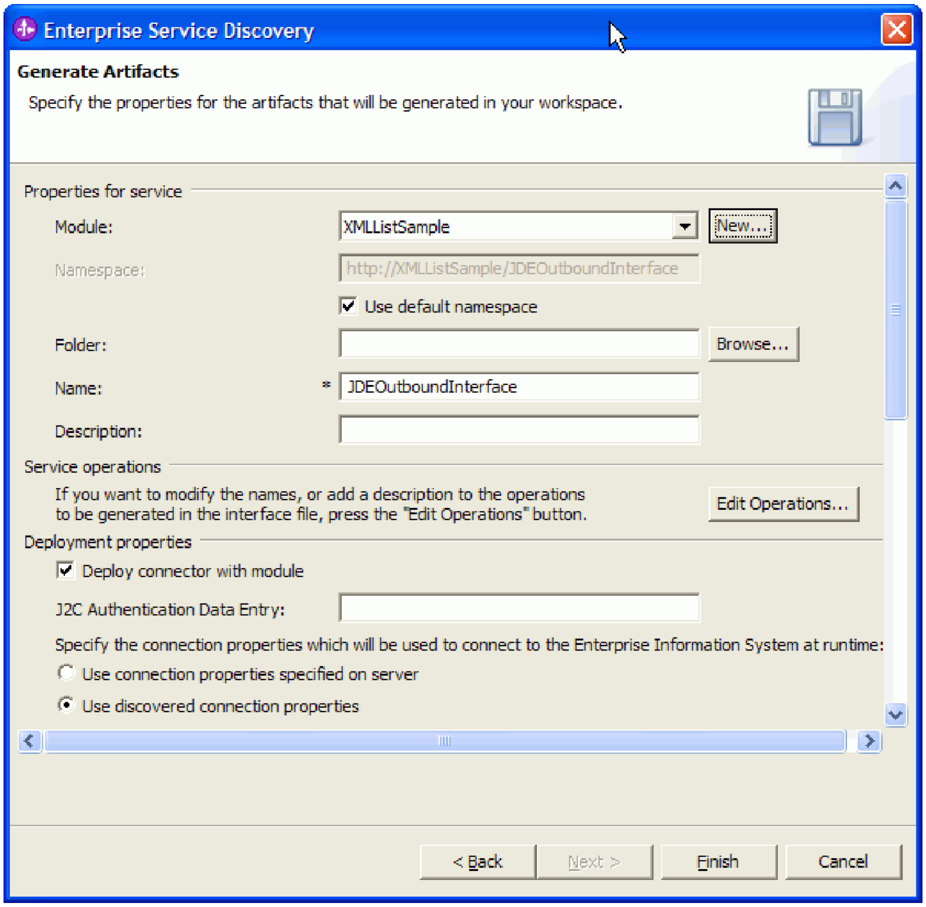
Before you begin

Make sure you have configured the GetEffectiveAddresss business function. The Generate Artifacts window should be open.

How to perform this task

1. In the Generate Artifacts window, click **New** next to the Module field.
2. In the New Integration Project window, select **Create a module project**, then click **Next**.
3. In the New Module window, type **XMLListSample** in the Module Name field, then click **Finish**.

4. In the Generate Artifacts window, select **Use discovered connection properties**.



Enterprise Service Discovery

Generate Artifacts

Specify the properties for the artifacts that will be generated in your workspace.

Properties for service

Module: XMLListSample

Namespace: http://XMLListSample/JDEOutboundInterface

☑ Use default namespace

Folder:

Name: * JDEOutboundInterface

Description:

Service operations

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

Deployment properties

☑ Deploy connector with module

J2C Authentication Data Entry:

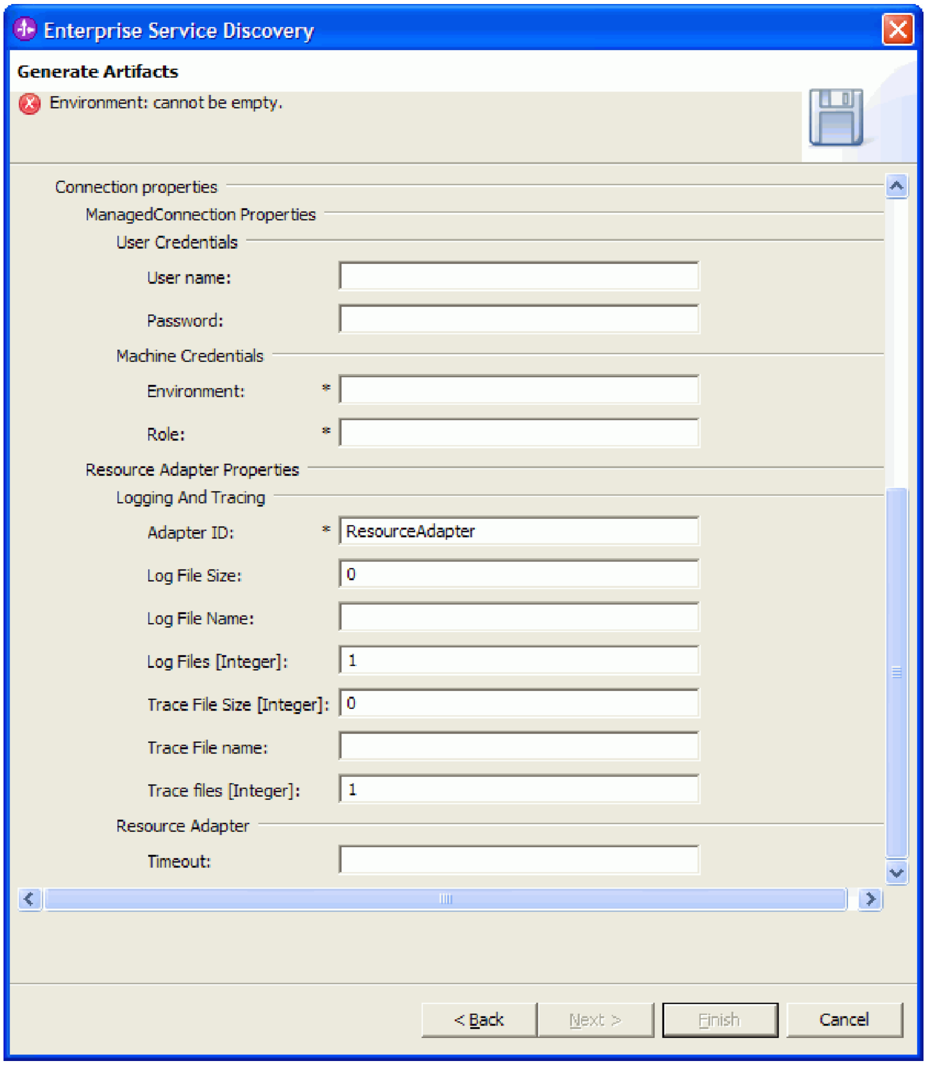
Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

○ Use connection properties specified on server

● Use discovered connection properties

Figure 72. Generate Artifacts window (top portion) of the enterprise service discovery wizard

5. Scroll down to enter information in the **Environment** and **Role** fields. You are required to fill out these two fields, as indicated by the asterisk (*). For more information about the **Environment** and **Role** properties, see “Managed (J2C) connection factory properties” on page 114.



Enterprise Service Discovery

Generate Artifacts

Environment: cannot be empty.

Connection properties

ManagedConnection Properties

User Credentials

User name:

Password:

Machine Credentials

Environment: *

Role: *

Resource Adapter Properties

Logging And Tracing

Adapter ID: * ResourceAdapter

Log File Size: 0

Log File Name:

Log Files [Integer]: 1

Trace File Size [Integer]: 0

Trace File name:

Trace files [Integer]: 1

Resource Adapter

Timeout:

Figure 73. Generate Artifacts window (bottom portion) of the enterprise service discovery wizard

6. Click **Finish**.

Result

The XMLListSample module is displayed in the J2EE perspective of WebSphere Integration Developer with "App" appended to its name, indicating that the module is a deployable application.

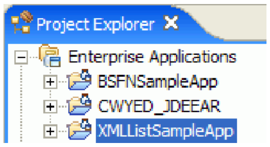


Figure 74. XMLListSample module displayed in the Project Explorer window (J2EE perspective) of WebSphere Integration Developer

What to do next

Deploy the module for testing.

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server or WebSphere Enterprise Service Bus, you add XMLListSampleApp to the server. If the server is not already started, it starts when you add the module.

Before you begin

You must have exported XMLListSampleApp as an EAR file before installing it on the server.

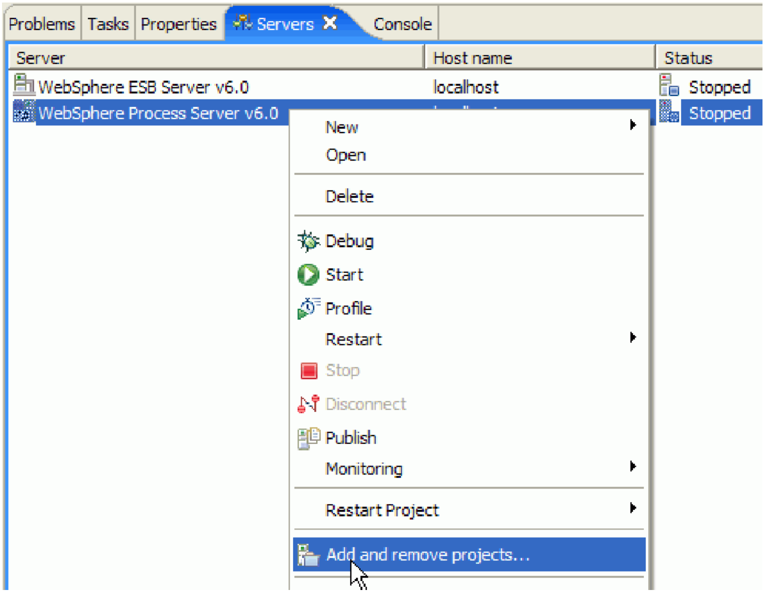
About this task

You use WebSphere Integration Developer to access the server and deploy the module to the test environment of the server.

To deploy the module, use the following procedure.

How to perform this task

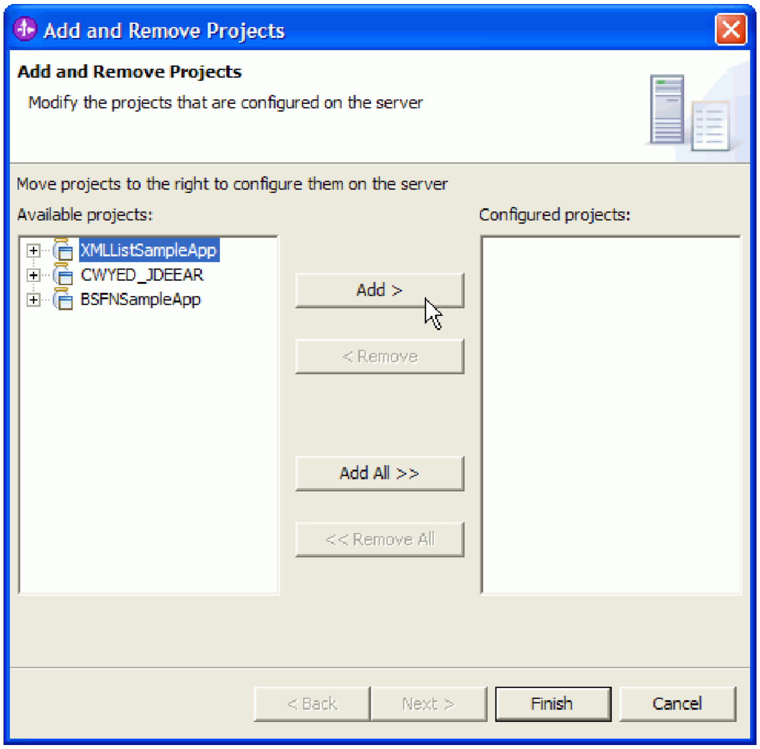
1. Select the test environment server.
   a. In WebSphere Integration Developer, click the **Server** tab.
   b. Right-click the server, then click **Add and remove projects**.



Server	Host name	Status
WebSphere ESB Server v6.0	localhost	Stopped
WebSphere Process Server v6.0	localhost	Stopped

Figure 75. Selecting **Add and remove projects**

2. In the Add and Remove Projects window, select **XMLListSampleApp**, then click **Add**.



Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

- XMLListSampleApp
- CWYED_JDEEAR
- BSFNSampleApp

Configured projects:

Figure 76. Add and Remove Projects window

3. When **XMLListSampleApp** is displayed in the Configured projects list, click **Finish**.

Result

Your server is started, if it was not already started. Then, the Console tab displays the status of the module as it is deployed.

Testing the module

Test the module to make sure you can retrieve data from the JD Edwards EnterpriseOne server. You enter search criteria, and the data meeting that criteria is returned.

Before you begin

Make sure you have deployed the XMLListSample module to the server.

How to perform this task

1. In the Business Integration perspective of WebSphere Integration Developer, right-click XMLListSample, then select **Test** > **Test Module**.

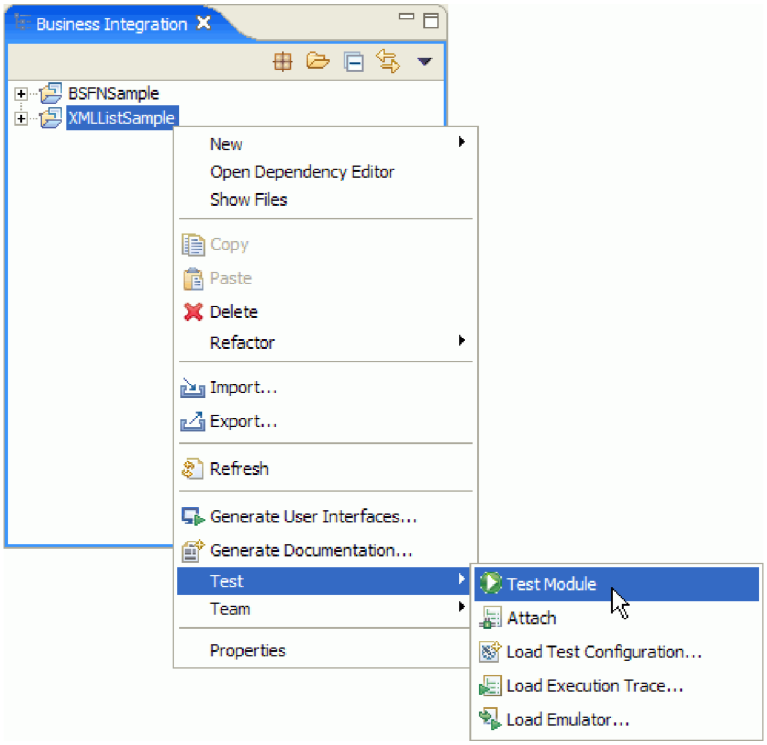
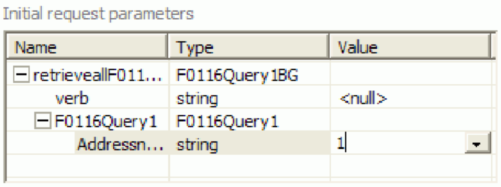


Figure 77. Selecting the XMLListSample module to test

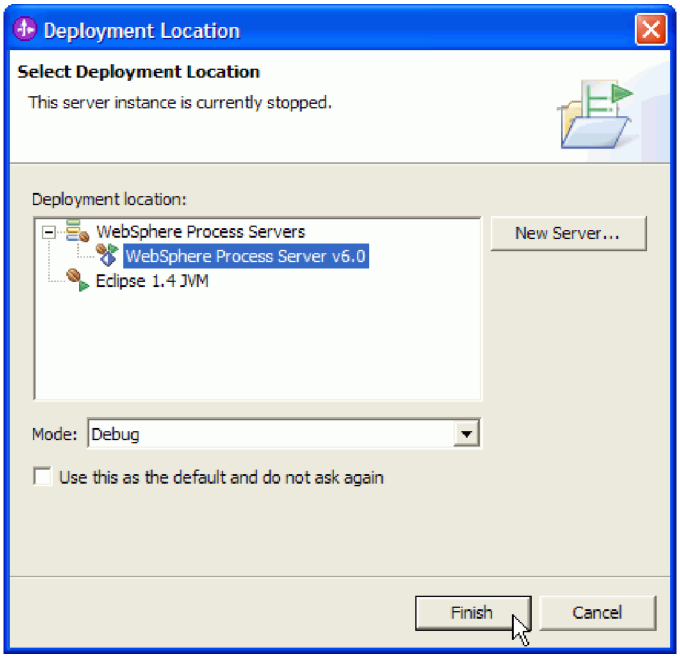
2. In the Initial request parameters table of the Events page, populate the **Addressnumber** attribute of the **F0116Query** business object with **1**.


Initial request parameters

Name	Type	Value
retrieveallF011...	F0116Query1BG	
verb	string	<null>
F0116Query1	F0116Query1	
Addressn...	string	1

Figure 78. Adding a value to the Addressnumber attribute in the Initial request parameters window

3. Execute the service by clicking **Continue**.
4. In the Deployment Location window, select the server, then click **Finish**.



Deployment Location

Select Deployment Location

This server instance is currently stopped.

Deployment location:

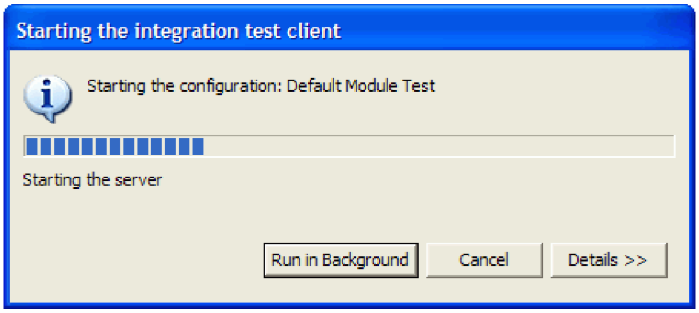
- WebSphere Process Servers
  - WebSphere Process Server v6.0
- Eclipse 1.4 JVM

Mode: Debug

□ Use this as the default and do not ask again

Figure 79. Deployment Location window

5. In the Starting the integrated test client window, click **Run in Background** if you prefer to run the server in the background.



Starting the integration test client

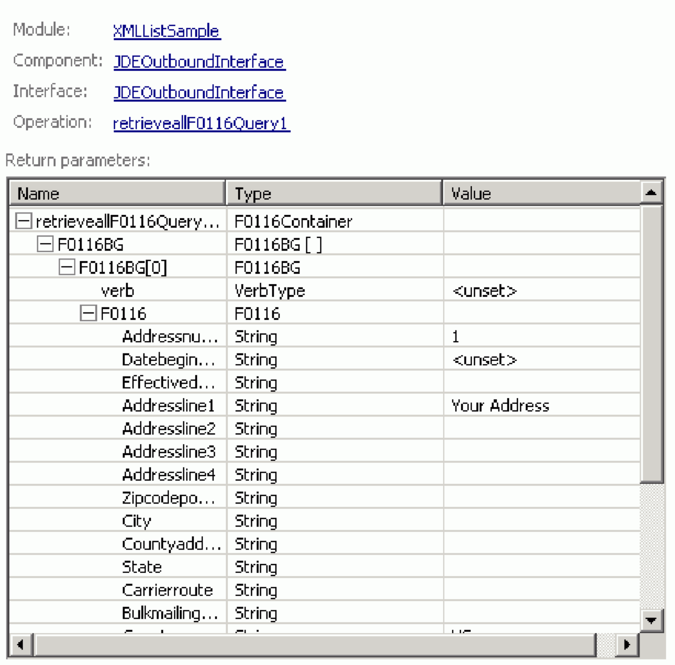
Starting the configuration: Default Module Test

Starting the server

Figure 80. Starting the integration test client window

Result

The output of the service and the data in the JD Edwards EnterpriseOne application match the data you entered for the **Addressnumber** attribute of the F0116 Query business object. This indicates that the XMLListSampleApp test successfully completed.



Module: XMLListSample

Component: JDEOutboundInterface

Interface: JDEOutboundInterface

Operation: retrieveallF0116Query1

Return parameters:

Name	Type	Value
⊟ retrieveallF0116Query...	F0116Container	
⊟ F0116BG	F0116BG [ ]	
⊟ F0116BG[0]	F0116BG	
verb	VerbType	<unset>
⊟ F0116	F0116	
Addressnu...	String	1
Datebegin...	String	<unset>
Effectived...	String	
Addressline1	String	Your Address
Addressline2	String	
Addressline3	String	
Addressline4	String	
Zipcodepo...	String	
City	String	
Countyadd...	String	
State	String	
Carrierroute	String	
Bulkmailing...	String	

Figure 81. Test result of XMLListSample

Viewing the sample adapter artifacts

To view the sample artifacts for each tutorial, import into IBM WebSphere Integration Developer the quick start reference files included with the adapter. Note that these artifacts are for reference only. They probably will not execute in your enterprise information system environment. If you have not stepped through the tutorials, you can still use the reference files to view examples of correctly-generated artifacts before you create your own.

Before you begin

Locate the quick start reference files in the referencefiles subdirectory of the samples directory. There is a project interchange zip file for each quick start tutorial. For instance, Tutorial1.zip is for quick start tutorial 1.

Important: Do not modify or use the artifacts provided in the quick start reference files. They are provided exclusively for viewing.

Reference files do not include third-party libraries. When imported into IBM WebSphere Integration Developer, the reference files might generate compilation error messages because dependent libraries are missing. The artifacts in the reference files may not be compatible with the enterprise information system (EIS) you are using. They vary based on EIS version and configuration.

The artifacts were generated with version 8.95 of JD Edwards EnterpriseOne Tools and version 8.11 of JD Edwards EnterpriseOne.

About this task

Import the quick start reference files into WebSphere Integration Developer to view sample artifacts associated with each quick start tutorial.

How to perform this task

1. In the Business Integration perspective of WebSphere Integration Developer, click **File** → **Import**.
2. In the Import window, select **Project Interchange** , then click **Next**.
3. Select the project interchange file containing the tutorial artifacts you want to view.
4. Import all the projects in the project interchange file by clicking **Select All** .
5. Click **Finish**.

Result

A business integration module is created with the following artifacts:

- Service import and export definitions
- Business objects (service data objects)
- Interfaces.

Troubleshooting the tutorials

If you experience problems deploying the adapter to the integrated test client of the server, you can try deploying the adapter from the administrative console of the server.

About this task

In the quick start tutorials, you use WebSphere Integration Developer to deploy the module by adding it to the to the integrated test client of the server. This method is used in the tutorials because it is a fast and efficient way to deploy the modules for the purposes of testing them. If you have any problems using this method to deploy the module, you can use the administrative console of the server to deploy the module. Using the administrative console to deploy the module is the method used in a production environment.

How to perform this task

To deploy the module using the administrative console of the server, follow the instructions in the Chapter 8, “Deploying the module,” on page 45 section.

Chapter 12. Reference information

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

Business objects

A business object contains application-specific information (metadata) about processing the business object as well as the operation to be performed on the business object. The enterprise service discovery wizard automatically generates an XSD file that contains metadata for business objects. You can view and, if necessary, modify the values of the metadata

Metadata of business functions

The metadata generated by enterprise service discovery provides the adapter with instructions on how to process business functions. Metadata is generated at the business-object level, the operation level, and the property level.

You can view (and modify) the metadata values listed associated with the business object. Use Business Object Editor to modify the values. For more information about Business Object Editor, see "Related product information" on page 116.

Note: Do not change the metadata element name.

Business-object-level metadata

Business-object-level metadata for business functions defines the top-level container.

The following table describes the business-object metadata elements of a business function container objects.

Table 5. Metadata for business function containers

Metadata element	Description
Type	The business object type value is set to BSFN.
Operation	The specified operation metadata is defined in the JDEBFNOperationTypeMetadata tag and contains the following: • Name: Name of the operation. • BSFN: List of Business Functions associated with the operation.

The following table describes the business-object metadata elements of a business function.

Table 6. Metadata for business functions

Metadata element	Description
BSFNName	Name of the Business Function as it exists in JD Edwards EnterpriseOne

Property-level metadata

Property-level metadata can represent child objects or an array of child objects.

The following table describes the metadata elements of a property.

Table 7. Property-level metadata

Metadata element	Description
Name	The business function parameter name as represented in JD Edwards EnterpriseOne.
Type	The type of the business function parameter as it exists in JD Edwards EnterpriseOne.
IOType	The type of the business function parameter as it exists in JD Edwards EnterpriseOne. Possible values are: • `IN`: the parameter is mapped from the business object to the business function. • `OUT`: the parameter is mapped from the business function to the business object. • `INOUT`: the parameter is mapped both ways. • `DEFAULT`: the parameter is mapped using the default JD Edwards EnterpriseOne value. For adapter purposes, it is processed as INOUT.
RequiredType	Identifies if the parameter is required. Possible values are: • `YES`: the parameter is required. • `NO`: the parameter is not required. • `DEFAULT`: the parameter is using the JD Edwards EnterpriseOne value. For adapter purposes, it is processed as `NO`.
Length	The maximum possible length for the parameter value.

Reference	The xpath of the business object property that is used to obtain the value of this attribute. The xpath expression starts at the business function level. For example:
	`BusinessFunctionContainer`
	` BusinessFunction1`
	`  Prop1`
	` BusinessFunction2`
	`  Prop2`
	If `BusinessFunction2/Prop2` property needs to be set with the value of `BusinessFunction1/Prop1`, the value of **Reference** for `BusinessFunction2/Prop2` needs to be set to `BusinessFunction1/Prop1`.

Operation-level metadata

The metadata for an operation specifies the list of business function names of the business functions in the JD Edwards EnterpriseOne system. This name is used by the adapter to execute the business functions.

The following table describes the operation-level metadata elements of a business function container

Table 8. Operation-level metadata

Metadata element	Description
Name	The name of the business object operation.
BSFN	List of business functions associated with the operation. • `Name`: Name of the business function • `RollbackOnWarnings`: A boolean that indicates if the adapter needs to rollback the current transaction when the business function returns with execution warnings. The default setting is `false`.

External software dependencies

WebSphere Adapter for JD Edwards EnterpriseOne requires external dependencies libraries to communicate with the JD Edwards EnterpriseOne environment.

To satisfy the external software dependencies of the Adapter for JD Edwards EnterpriseOne, you must copy the following files to the specified locations during adapter configuration. The software dependencies differ, depending on which version of JD Edwards EnterpriseOne you use.

For information about how to obtain these files, refer to the following document JD Edwards EnterpriseOne document: *PeopleSoft EnterpriseOne Tools Connectors PeopleBook*.

Note: Depending on which version of JD Edwards EnterpriseOne you are using, you may be required to configure the following files:

- jdeinterop.ini
- jdelog.properties
- jdbj.ini
- tnsnames.ora

For instructions on configuring these files, refer to the following JD Edwards EnterpriseOne documents:

- *PeopleSoft EnterpriseOne Tools Connectors PeopleBook* (for jdeinterop.ini and jdelog.properties files)
- *PeopleSoft EnterpriseOne Tools HTML Server Installation PeopleBook* (for jdbj.ini and tnsnames.ora files)

External dependencies for WebSphere Adapter for JD Edwards EnterpriseOne.

Table 9. External software dependencies for different versions of JD Edwards EnterpriseOne

JD Edwards EnterpriseOne, version 8.9 (SP1, SP2), 8.93	JD Edwards EnterpriseOne, version 8.94	JD Edwards EnterpriseOne, version 8.95
kernel.jar	kernel.jar	Connector.jar
connector.jar	Connector.jar	JdbjBase_JAR.jar
database.jar	database.jar	JdbjInterfaces_JAR.jar
log4j.jar	log4j.jar	JdeNet_JAR.jar
xerces.jar	xerces.jar	Spec_JAR.jar
xalan.jar	xalan.jar	System_JAR.jar

Table 9. External software dependencies for different versions of JD Edwards EnterpriseOne (continued)

JD Edwards EnterpriseOne, version 8.9 (SP1, SP2), 8.93	JD Edwards EnterpriseOne, version 8.94	JD Edwards EnterpriseOne, version 8.95
jdeinterop.ini	jdeutil.jar	Base_JAR.jar
jdeLog.properties	jdbj.ini	log4j.jar
JDBC driver	jdeinterop.ini	xerces.jar
	jdelog.properties	xalan.jar
	JDBC driver	PMApi_JAR.jar
		BizLogicContainer_JAR.jar
		BizLogicContainerClient_JAR.jar
		ApplicationAPIs_JAR.jar
		ApplicationLogic_JAR.jar
		jdeinterop.ini
		jdbj.ini
		jdelog.properties
		JDBC driver

Adapter configuration properties

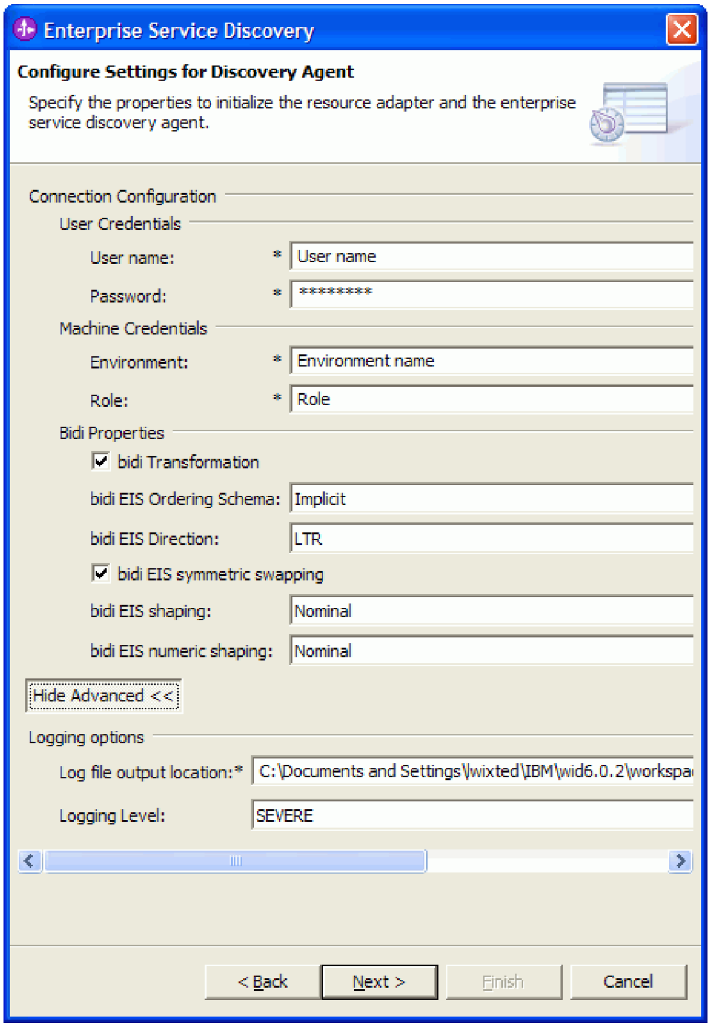
WebSphere Adapter for JD Edwards EnterpriseOne has several categories of configuration properties, some of which you set during the enterprise service discovery process and some of which you can set or change after you deploy the adapter application to the WebSphere Process Server or WebSphere Enterprise Service Bus.

Enterprise service discovery connection properties

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

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

Figure 82. Enterprise service discovery wizard: connection properties



Enterprise Service Discovery

Configure Settings for Discovery Agent

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

Connection Configuration

User Credentials

User name: * User name

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

Machine Credentials

Environment: * Environment name

Role: * Role

Bidi Properties

☑ bidi Transformation

bidi EIS Ordering Schema: Implicit

bidi EIS Direction: LTR

☑ bidi EIS symmetric swapping

bidi EIS shaping: Nominal

bidi EIS numeric shaping: Nominal

Hide Advanced <<

Logging options

Log file output location:* C:\Documents and Settings\wixted\IBM\wid6.0.2\workspa

Logging Level: SEVERE

Table 10. Enterprise service discovery connection properties for the Adapter for JD Edwards EnterpriseOne

Property	Type	Description	Default value
User name	String	Name of the adapter user account on the JD Edwards EnterpriseOne environment.	
Password	String	Password of the adapter user account on the JD Edwards EnterpriseOne environment.	
Environment	String	JD Edwards EnterpriseOne environment name	
Role	String	Name of the role that is associated with the user name that is allowed to access the JD Edwards EnterpriseOne environment	
bidi Transformation	Boolean	Turns bidirectional support on or off.	False (bi-di support is turned off)
bidi Ordering Schema	String	Determines the type of text schema used–either implicit (logical) or visual.	Implicit

Table 10. Enterprise service discovery connection properties for the Adapter for JD Edwards EnterpriseOne (continued)

Property	**Type**	**Description**	**Default value**
bidi EIS Direction	String	Determines the text direction used. Possible values are LTR (left to right), RTL (right to left), ContextualLTR (contextual left to right), and ContextualRTL (contextual right to left.)	LTR
bidi EIS symmetric swapping	Boolean	Determines whether systemic swapping is turned on or off.	True (systemic swapping is turned on)
bidi EIS shaping	String	Determines the bi-di format used by the enterprise service discovery wizard while it communicates with the JD Edwards EnterpriseOne application. Possible values are Nominal, National, and Contextual.	Nominal
bidi numeric shaping	String	Determines the bi-di format used by the enterprise service discovery wizard while it communicates with the JD Edwards EnterpriseOne application. Possible values are Nominal, National, and Contextual.	Nominal
Log file output location	String		Workspace location of WebSphere Integration Developer

Resource adapter properties

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

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

Table 11. Resource adapter properties for the Adapter for JD Edwards EnterpriseOne

Property name	Type	Description	Default value
adapterID	String		ResourceAdapter
biDiContextEIS	String	The bi-di format used by JD Edwards EnterpriseOne for its business data. The adapter normalizes the JD Edwards EnterpriseOne bi-di data from the application server's format back to JD Edwards EnterpriseOne bi-di format for outbound communication.	
biDiContextMetadata	String	Specifies the bi-directional format of meta configuration data in the JD Edwards EnterpriseOne environment.	

Table 11. Resource adapter properties for the Adapter for JD Edwards EnterpriseOne (continued)

biDiContextSkip	Boolean	Controls invocation of bi-di transformation. Acceptable values: true or false. A blank value invokes the lookup mechanism.	
biDiContextSpecialFormat	String	Signifies a category of values that are subject to special treatment during invocation of bi-di transformation to ensure accurate transformation of the category. Categories are predefined (for example, FTP URLs and e-mail addresses).	
biDiContextTurnBidiOff	Boolean	A flag used to turn off (explicitly exclude) bi-di support. This property takes precedence over the BiDiSkip property, and it allows users who do not require bidirectional script data support to turn it off.	`True`
enableHASupport	String	When the enableHASupport property is set to true, only one of the replicated adapter instances actively polls for events while other instances are in standby mode. If the enableHASupport property is set to false, all of the adapter instances replicated on cluster members actively poll for events. This may result in event duplication. Do not change the value of enableHASupport to false for single server environments.	`True`
logFilename	String	The full path of the log file. This property is required.	
logFileSize	Integer	Size of the log files in kilobytes. If no value is specified, the file has no maximum size. This property is not required.	`0`
logNumberofFiles	Integer	The number of log files to use. When a log file reaches its maximum size, the adapter starts using another log file. If no value is specified, the number is set to 1. This property is not required.	`1`
threadContextPropagation Required	Boolean		`True`
timeout	Integer	Global timeout value, in milliseconds, set on the XML List request execute call.	`30000` (30 seconds)
traceFileName	String	The full path to the trace file. This property is required.	
traceFileSize	Integer	Size of the trace files in kilobytes. If no value is specified, the file has no maximum size. This property is not required.	

Table 11. Resource adapter properties for the Adapter for JD Edwards EnterpriseOne (continued)

traceNumberOfFiles	Integer	The number of trace files to use. When a trace file reaches its maximum size, the adapter starts using another trace file. If no values is specified, the number is set to 1. This property is not required.	

Managed (J2C) connection factory properties

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

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

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

Table 12. Managed (J2C) connection factory properties

Property	**Type**	**Description**
Username	String	User name to use when logging into the JD Edwards EnterpriseOne Server
Password	String	Password to use with the corresponding user name
Environment	String	JD Edwards EnterpriseOne environment name for the corresponding user name. **Note:** Multiple JD Edwards EnterpriseOne environments can be present on a single machine.
Role	String	Role for the corresponding user name

Interaction specification properties

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

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

Interaction Specification properties for the Adapter for JD Edwards EnterpriseOne

Property	**Type**	**Description**
Maximum Number of Records	String	The maximum number of records to retrieve during the execution of a RetrieveAll operation.

Property	Type	Description
timeout	String	The global timeout value, in milliseconds, set on the XML List request execute call. The default is `30000` (30 seconds).

Adding jar files to WebSphere Integration Developer versions 6.0.1.1 and earlier

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

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

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

Settings for controlling bidirectional transformation

Within each category of adapter properties, certain properties can be set to control bidirectional transformation of content or metadata. Properties controlling bidirectional transformation can be set for either the resource adapter or the managed connection factory.

Resource adapter properties

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

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

Managed (J2C) connection factory properties

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

- BiDiContextCPuserNamEIS
- BiDiContextCPuserNameSkip
- BiDiContextCPpasswordEIS
- BiDiContextCPpasswordSkip
- BiDiContextCPenvironmentEIS
- BiDiContextCPenvironmentSkip
- BiDiContextCProleEIS
- BiDiContextCProleSkip

Messages

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

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

Related product information

The following links, information centers, Redbooks, and Web pages contain related information for the IBM WebSphere Adapter for JD Edwards EnterpriseOne.

Additional information you might need

Table 13. WebSphere Adapters information you might need

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

Information for related products

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

Redbooks

- WebSphere Adapter Development Redbook
- WebSphere Redbooks domain

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

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

Chapter 13. Glossary

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

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

Notices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

Programming interface information

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

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

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

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

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

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

Index

Printed in USA