WebSphere Adapters

Version 6 Release 1

WebSphere Adapter for JD Edwards EnterpriseOne User Guide
Version 6 Release 1

WebSphere® Adapters

IBM

Version 6 Release 1

WebSphere Adapter for JD Edwards EnterpriseOne User Guide
Version 6 Release 1

Note
Before using this information and the product it supports, read the information in "Notices" on page 105.

16 January 2007

This edition applies to version 6, release 1, modification 0 of IBM WebSphere Adapter for JD Edwards EnterpriseOne and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

Contents

Chapter 1. Overview of WebSphere Adapter for JD Edwards EnterpriseOne

With WebSphere Adapter for JD Edwards EnterpriseOne, you can create integrated processes that include the exchange of information with a JD Edwards EnterpriseOne server, without special coding.

The adapter provides a standard interface that eliminates the need for the component to understand the lower-level implementation details or data structures of the application. Using the adapter, a component (the program or piece of code that performs a specific business function) can send requests to the JD Edwards EnterpriseOne server (for example, to query a customer record in a JD Edwards EnterpriseOne table or to update an order document).

WebSphere Adapter for JD Edwards EnterpriseOne complies with the Java™ Connector Architecture (JCA). JCA standardizes the way application components, application servers, and enterprise information systems, such as a JD Edwards EnterpriseOne server, interact with each other. WebSphere Adapter for JD Edwards EnterpriseOne makes it possible for JCA-compliant application servers to connect to and interact with the JD Edwards EnterpriseOne server. Clients running on the JCA-compliant server can then communicate with the JD Edwards EnterpriseOne server in a standard way (using business objects or JavaBeans™).

The following example assumes you are setting up an adapter using WebSphere Integration Developer and deploying the module that contains the adapter toWebSphere Process Server.

Suppose a medium-sized retail company uses JD Edwards EnterpriseOne to coordinate most of its business operations. JD Edwards EnterpriseOne includes a business function that can return a real-time list of inventory items for its 100 stores located across the United States. An application component might be able to use this business function as part of an overall business process. For example, an employee of a retail company can access the real-time list of available inventory items, thus providing correct, real-time information to a customer.

New in this release

WebSphere Adapter for JD Edwards EnterpriseOne, Version 6.1.0 provides enhancements to the adapter. This release also includes some deprecated features.

Updates to this information are made available at the WebSphere Adapters product support Web site. To read updated or additional information, see: http://www.ibm.com/software/integration/wbiadapters/support/.

New in version 6.1.0

- The adapter now supports JD Edwards EnterpriseOne, version 8.9.6.
- Business graphs and verbs are now optional

  The business graph that contains each business object in version 6.0.2 is now optional. You need a business graph only for modules whose business objects were created in version 6.0.2.
- Changes to the enterprise service discovery wizard

The wizard has been renamed the external service wizard, and includes functional and usability improvements to make it easier for you to discover, create, and configure business objects and services for use with the adapter. The wizard now guides you through several tasks that were previously performed manually in the file system or in WebSphere Integration Developer, such as creating a project, importing dependency files into the project, and creating the module.

The wizard now provides default values for many properties, makes it easier to enter certain information, indicates which properties are required, and lets you configure the module without worrying about advanced properties.

- Simplified support for bidirectional script processing
- Support for node-level, or stand-alone, deployment of the adapter
- Support for business faults

  The adapter now generates business faults for business exceptions. This lets you easily assign a corrective action for those error conditions.
- The adapter RAR file is available in WebSphere Integration Developer; you do not need to install it separately. The wizard automatically copies the adapter files into the project for you.
- The adapter documentation is located on the WebSphere Integration Developer Information Center, in the Configuring and using adapters section.
- Support for a first-failure data capture (FFDC) construct that can be contained in a WebSphere Application Server symptom database to provide information and suggested actions to assist a diagnostic module in customizing the data that is logged.

Deprecated in version 6.1.0

A deprecated feature is one that is supported but no longer recommended and that might become obsolete. For a list of features from earlier versions of Adapter for JD Edwards EnterpriseOne that have been deprecated in version 6.1.0, see "Migration considerations" on page 16.

Hardware and software requirements

The hardware and software requirements for WebSphere Adapters are documented on the IBM® Web site at the location below.

Hardware and software requirements for WebSphere Adapters:
http://www.ibm.com/support/docview.wss?uid=swg27006249

Additional information

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

- The compatibility matrix for WebSphere Business Integration Adapters and WebSphere Adapters identifies the supported versions of required software for your adapter. To view this document, go to the WebSphere Adapters support page and click the link for the compatibility matrix under **Planning upgrades**: http://www.ibm.com/software/integration/wbiadapters/support/.
- Technotes for WebSphere Adapters document workarounds and additional information not included in the product documentation. To view the technotes for your adapter, go to the following Web page, select the name of your adapter from the **Product category** list, and click the search icon: http://www.ibm.com/

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

Technical overview of the Adapter for JD Edwards EnterpriseOne

IBM WebSphere Adapter for JD Edwards EnterpriseOne provides a way for applications to interact with data on JD Edwards EnterpriseOne servers. Outbound processing, which is the processing of requests from an application through the adapter to the JD Edwards EnterpriseOne server, is supported.

The adapter processes requests using one of two types of business objects: business functions and XML Lists. A business function is a business object container that can contain one or many business objects which can be processed as a single transaction. An XML List is a single business objects that can query a table and return multiple records.

You create business objects by using the external service wizard, a tool launched from WebSphere Integration Developer. The business objects generated by the external service 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, another tool launched from WebSphere Integration Developer, to changed the properties of the generated business object definition. You can also change or set configuration properties for business objects in the administration console of WebSphere Integration Developer.

Outbound processing

Adapter for JD Edwards EnterpriseOne supports synchronous outbound request processing. This means that when the adapter receives a request, in the form of a business object, from the module, it processes the request and returns the result, when applicable, in a business object.

When the adapter receives a request, it 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 operations:

- create
- delete
- execute
- retrieve
- update

XML Lists support the following operation:

- retrieveAll

Business objects

A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent either a business entity, such as an invoice or an employee record, or unstructured text. The adapter uses business objects to send data to or obtain data from the JD Edwards EnterpriseOne server.

How the adapter uses business objects

The adapter uses the JD Edwards EnterpriseOne Dynamic Java Connector APIs to communicate with the JD Edwards EnterpriseOne application. The adapter exchanges information with EnterpriseOne through business function and XML List calls.

The following figure illustrates how business objects are used by the adapter for JD Edwards EnterpriseOne.

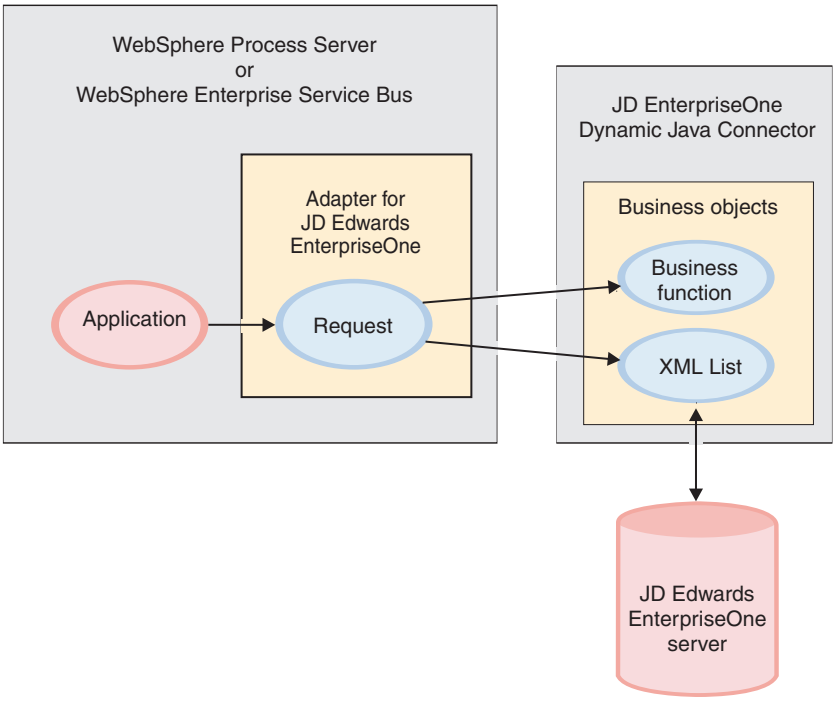


Figure 1. How the adapter for JD Edwards EnterpriseOne uses business objects

How business objects are created

You create business objects by using the external service wizard, which is launched from WebSphere Integration Developer. The external service wizard connects to the application, discovers data structures in the application, and generates business objects to represent them. It also generates other artifacts needed by the adapter.

The following figure illustrates the high-level flow of steps involved in creating business objects using the external service wizard.

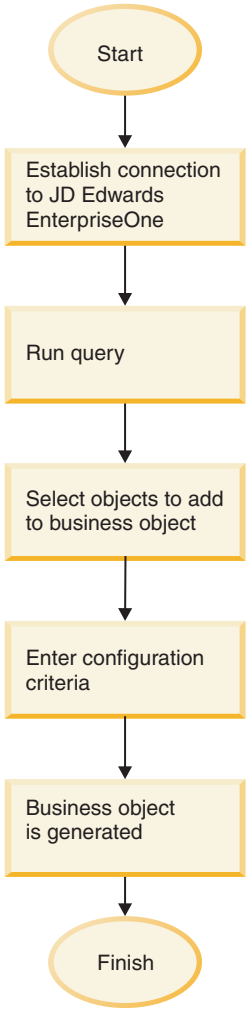


Figure 2. Creating business objects using the external service wizard

Business object structure

The adapter supports processing of hierarchical business objects. The top-level business object that comes under the business graph is a wrapper container business object. A container business object representing a JD Edwards EnterpriseOne operation is a wrapper object that contains single or multiple child business function objects, also called simple business function objects. Each business function object represents a specific function call in JD Edwards EnterpriseOne.

The following figures show diagram views of a business graph, a business object container, and a simple (or child) business object.

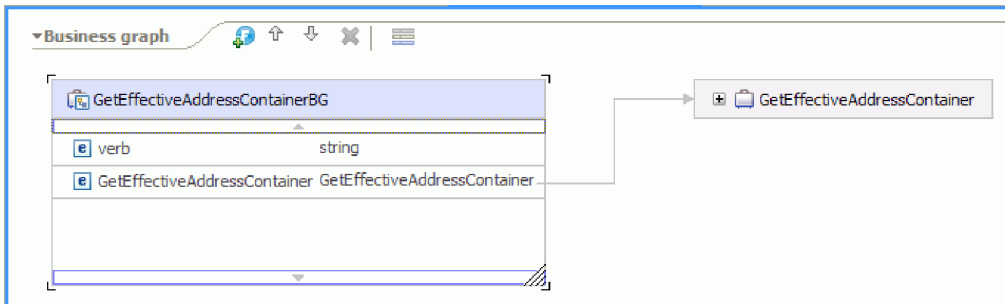


Figure 3. The structure of a business object graph, shown in a diagram view in WebSphere Integration Developer

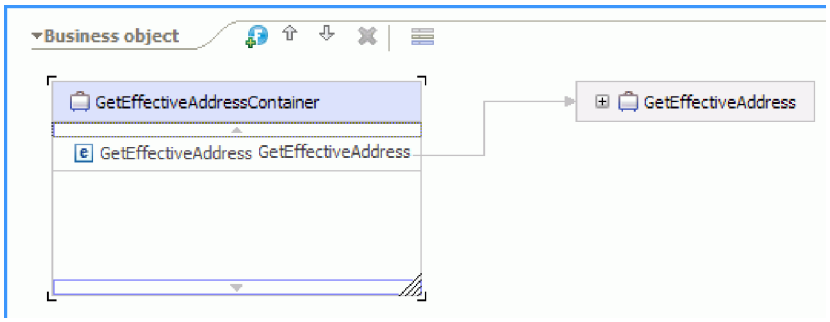


Figure 4. The structure of a business object container, shown in a diagram view in WebSphere Integration Developer

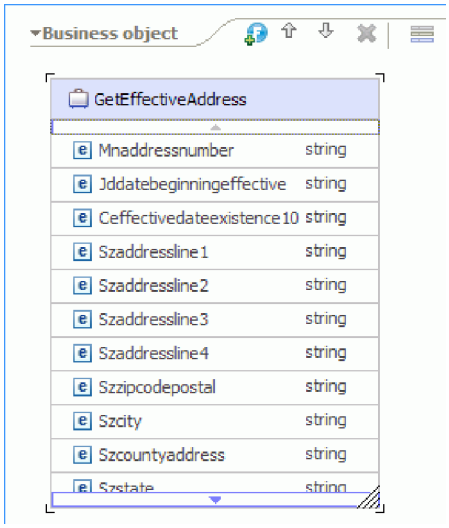


Figure 5. The structure of a child business object, shown in a diagram view in WebSphere Integration Developer

You can optionally choose, during adapter configuration, to generate a business graph. In version 6.0.2, each top-level business object is contained in a business graph, which includes a verb that an application can use in version 6.0.2 to specify additional information about the operation to be performed. In version 6.1.0, business graphs are optional; they are required only when you are adding business objects to a module created with a version of WebSphere Integration Developer

earlier than version 6.1.0. If business graphs exist, they are processed, but the verb is ignored.

The external service wizard

The external service wizard is a tool you use to configure your adapter before deploying it to WebSphere Process Server or WebSphere Enterprise Service Bus. The external service wizard establishes a connection to the JD Edwards EnterpriseOne server, discovers services (based on search criteria you provide), and generates business objects and interfaces based on the services discovered.

Using WebSphere Integration Developer, you establish a connection to the JD Edwards EnterpriseOne server to browse the database information on the JD Edwards EnterpriseOne server.

You specify connection information, such as the user name and password needed to access the server, as shown in the following figure.

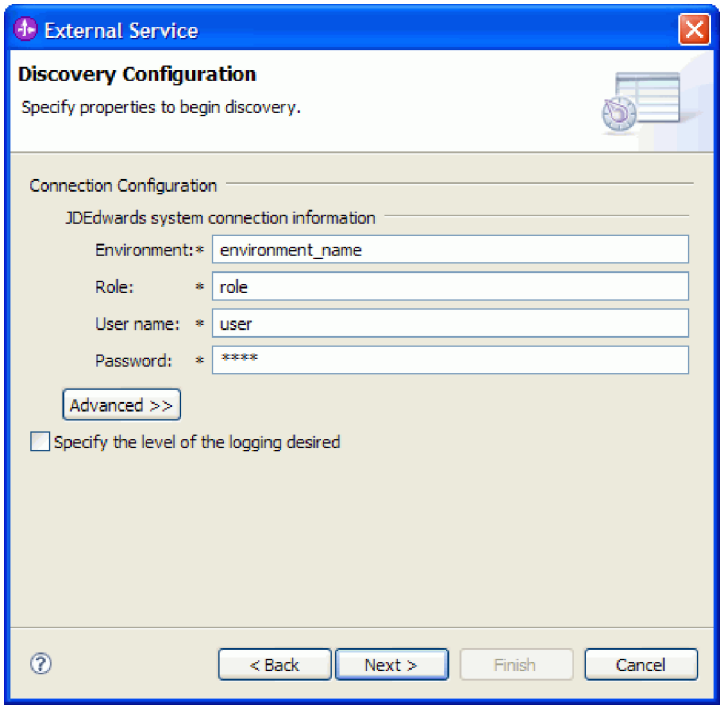


Figure 6. Configure Settings for Discovery Agent window

The result of running the external service wizard is a module that contains the interfaces and business objects along with the adapter. You deploy this module on WebSphere Process Server or WebSphere Enterprise Service Bus.

For example, if you run the external service wizard to create a module called BSFNSample, you can see, under **Data Types**, a list of generated business objects, including the objects associated with any faults that might be generated during processing.

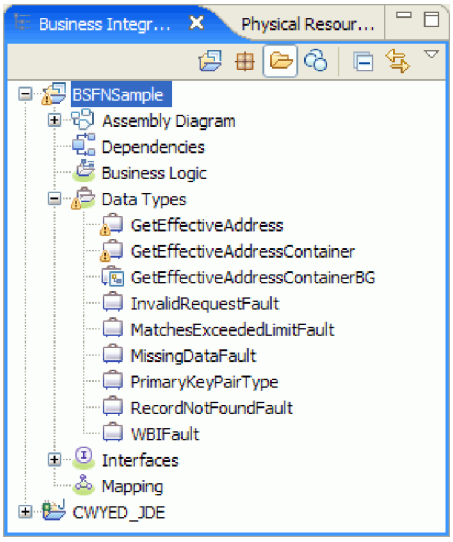


Figure 7. An example of the BSFNSample module generated by the external service wizard

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. WebSphere Adapters are 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.

Administration

The run time administrative console is the primary interface for deployment and administration of enterprise applications. The console is 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 utilizing 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.

External service wizard

The external service wizard is the primary component used to create modules. This wizard, which is implemented as an Eclipse plug-in that is available through WebSphere Integration Developer, is fully accessible.

Keyboard navigation

This product uses standard Microsoft Windows® navigation keys.

IBM and accessibility

See the *IBM Accessibility Center* web site http://www.ibm.com/able/ for more information about the commitment that IBM has to accessibility.

Internet Protocol Version 6 (IPv6)

WebSphere Process Server and WebSphere Enterprise Service Bus rely on WebSphere Application Server for Internet Protocol Version 6 (IPv6) compatibility.

IBM WebSphere Application Server, version 6.1.0 and later support dual-stack Internet Protocol Version 6.0 (IPv6).

For more information about this compatibility in WebSphere Application Server, see IPv6 support in the http://www.ibm.com/software/webservers/appserv/was/library/.

For more information about IPv6, see http://www.ipv6.org.

Chapter 2. Planning for adapter implementation

Before you configure WebSphere Adapter for JD Edwards EnterpriseOne, become familiar with all of the security and performance options that are required for your environment, including which deployment option to use. Also, if you are migrating from an earlier version of WebSphere Adapter for JD Edwards EnterpriseOne, perform any migration tasks.

Before you begin

Before you begin to set up and use the adapter, you should possess a thorough understanding of business integration concepts, the capabilities and requirements of the integration development tools and runtime environment you will use, and the JD Edwards EnterpriseOne environment where you will build and use the solution.

To configure and use WebSphere Adapter for JD Edwards EnterpriseOne, you should understand and have experience with the following concepts, tools, and tasks:

- The business requirements of the solution you are building.
- Business integration concepts and models, including the Service Component Architecture (SCA) programming model.
- The capabilities provided by the integration development tools you will use to build the solution. You should know how to use these tools to create modules, test components, and complete other integration tasks.
- The capabilities and requirements of the runtime environment you will use for the integration solution. You should know how to configure and administer the host server and how to use the administrative console to set and modify property definitions, configure connections, and manage events.

Security

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

User authentication

The adapter supports several methods for supplying the user name and password that are needed to connect to the JD Edwards EnterpriseOne server. Understand the features and limitations of each method to pick a method that provides the appropriate level of security and convenience for your application.

To integrate an adapter into your application, a user name and password are needed at the following times:

- When the external service wizard connects to the JD Edwards EnterpriseOne server to extract, or *discover*, information about the objects and services that you can access with the adapter.

- At run time on WebSphere Process Server or WebSphere Enterprise Service Bus, when the adapter connects to the JD Edwards EnterpriseOne server to process outbound requests.

Authentication in the wizard

The external service wizard asks for connection information for both uses. You can use a different user name and password while running the wizard than you use when the application is deployed to the server. You can even connect to a different JD Edwards EnterpriseOne server, although the schema name must be the same in both databases. For example, while developing and integrating an application that uses Adapter for JD Edwards EnterpriseOne, you might not use the production database; using a test database with the same data format but fewer, simulated records lets you develop and integrate the application without impacting the performance of a production database and without encountering restrictions caused by the privacy requirements for customer data.

The wizard uses the user name and password that you specify for the discovery process only during the discovery process; they are not accessible after the wizard completes.

Authentication at run time

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

- Adapter properties
- Data source
- J2C authentication alias

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

Using a data source lets you use a connection already established for another application. For example, if multiple applications access the same database with the same user name and password, the applications can be deployed using the same data source. The user name and password can be known only to the first person who deploys an application to that data source or who defines a data source separately.

Using a J2C authentication alias created with the Java Authentication and Authorization Service (JAAS) is a robust, secure way to deploy applications. An administrator creates the authentication alias that is used by one or more

applications that need to access a system. The user name and password can be known only to that administrator, who can change the password in a single place when a change is required.

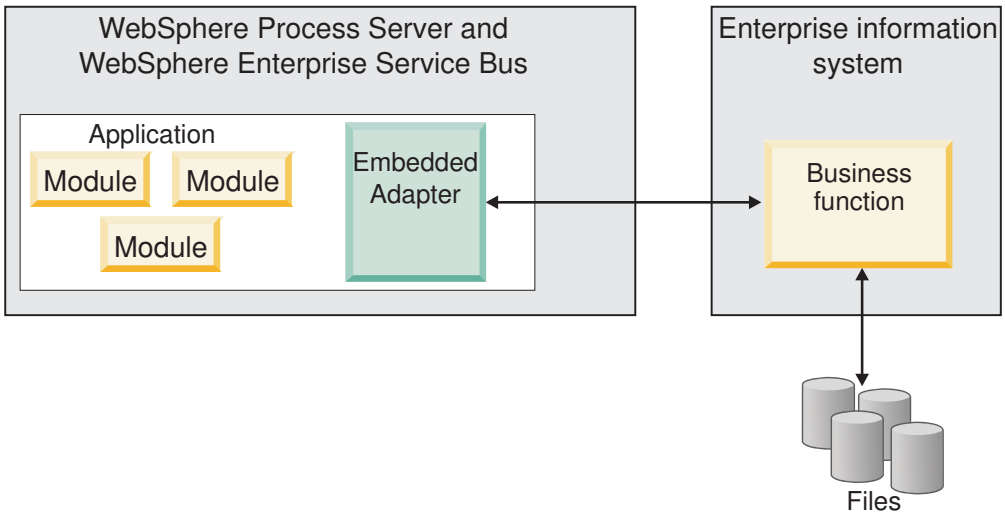
Deployment options

You can choose to embed the adapter to be part of the deployed application or you can choose to deploy the RAR file stand-alone.

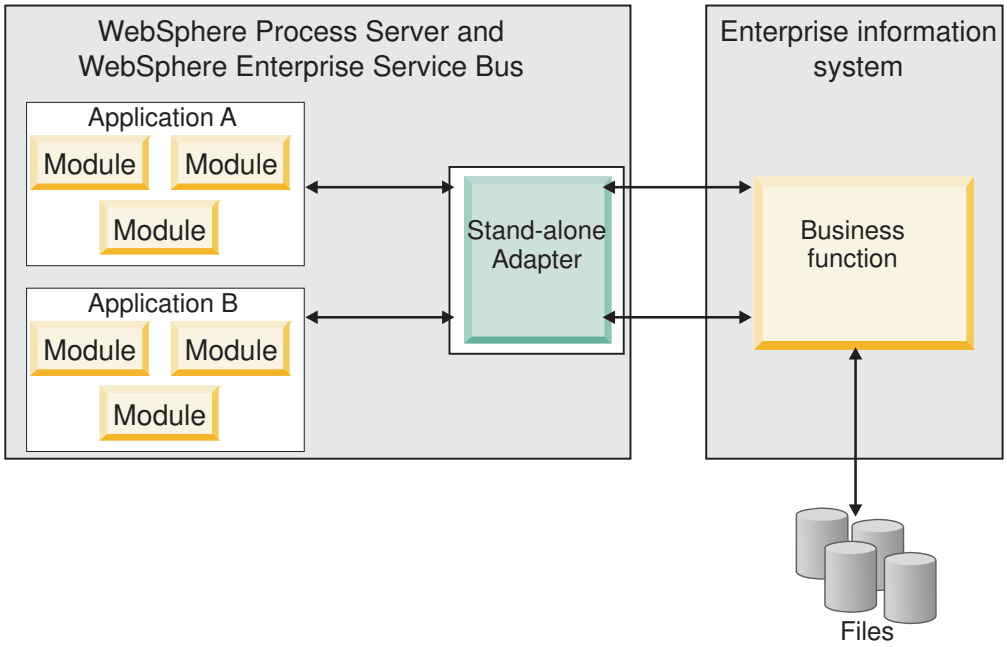
The deployment options are described below:

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

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



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



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

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

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

Considerations for embedding an adapter in the application

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

- An embedded adapter has class loader isolation.

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

Considerations for using a stand-alone adapter

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

- Stand-alone adapters have no class loader isolation.

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

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

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

WebSphere Adapters in clustered environments

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

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

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

benefits of WebSphere Extended Deployment, see the following link: http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1/index.jsp.

In clustered environments, adapter instances for WebSphere Adapter for JD Edwards EnterpriseOne can handle outbound processes only.

High availability for outbound processes

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

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

Migrating to version 6.1.0

By migrating to version 6.1 of WebSphere Adapter for JD Edwards EnterpriseOne, you automatically upgrade from the previous version of the adapter. Additionally, you can migrate your applications that embed an earlier version of the adapter, so that the applications can utilize features and capabilities present in version 6.1.

Migration considerations

WebSphere Adapter for JD Edwards EnterpriseOne version 6.1.0 includes updates that might affect your existing applications. Before migrating applications that will utilize WebSphere Adapter for JD Edwards EnterpriseOne, take into consideration the information in the sections that follow.

Compatibility with earlier versions

WebSphere Adapter for JD Edwards EnterpriseOne version 6.1.0 is fully compatible with version 6.0.2 of the adapter and can work with custom business objects (XSD files), and data bindings.

Because version 6.1 of WebSphere Adapter for JD Edwards EnterpriseOne is fully compatible with version 6.02, any of your applications that utilized version 6.0.2 of WebSphere Adapter for JD Edwards EnterpriseOne will run unchanged when you upgrade to version 6.1. However, if you want your applications to utilize features and functionality present in version 6.1 of the adapter, run the migration wizard.

The migration wizard replaces (upgrades) version 602 of the adapter with version 6.1 *and enables version 6.1 features and functionality for use with your applications*.

Note: The migration wizard does not create new or modify existing mitigating code, such as mappers and mediators to work with version 6.1 of the adapters. If any of your applications embed a 6.0.2.x or earlier version of an adapter and you are upgrading to version 6.1.0, and you want your applications to take advantage of the features and functions in 6.1, you might need to make changes to those applications.

If artifacts are inconsistent with regard to *versioning* within a single module, this module in its entirety will be marked as such, will not be selectable for migration. Version inconsistencies are recorded in the workspace log, as this may be a symptom of project corruption.

Deciding whether to upgrade or to upgrade and migrate

The default processing of the migration wizard is to perform an upgrade of the adapter and to migrate the application artifacts so that the applications can utilize features and functions in version 6.1 of the adapter. When you choose to upgrade the connector by selecting a connector project, the wizard automatically selects the associated artifacts for migration.

If you decide that you want to upgrade the adapter from version 6.0.2 to version 6.1, but you do not want to migrate the adapter artifacts, you can do so by deselecting the adapter artifacts from the appropriate page of the migration wizard.

Running the migration wizard without any adapter artifacts selected will install and upgrade your adapter, but your artifacts are not migrated and your applications will not be able to take advantage of the features and capabilities that exist in version 6.1 of the adapter.

Run the migration wizard in a test environment first

Because adapter migration may require you to make changes to those applications that will utilize version 6.1 of WebSphere Adapter for JD Edwards EnterpriseOne, you should always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

The migration wizard is fully integrated with the development environment.

Deprecated features

Become familiar with the deprecated features in version 6.1.0 and make any required changes to your applications.

A deprecated feature is one that is supported but no longer recommended and that might become obsolete. Features from earlier versions of WebSphere Adapter for JD Edwards EnterpriseOne that have been deprecated in version 6.1.0 include:

- In version 6.1.0, the Timeout property moved from a Resource Adapter property to a Managed Connection Factory (J2C) property. In order provide backward compatibility with artifacts that were generated with the Timeout property set as a Resource Adapter property (version 6.0.2), the Adapter for JD Edwards EnterpriseOne, version 6.1.0 behaves as follows:
  1. The adapter looks for the Timeout property setting in the Managed Connection Factory (J2C) properties.

2. If the Timeout property is not set as a Managed Connection Factory (J2C) property, the adapter looks for it in the Resource Adapter properties.
3. If the Timeout property is not set as either a Managed Connection Factory property or a Resource Adapter property, the adapter assigns the default setting for the Timout property of 30 seconds.

For more information about the Timeout property, refer to "Managed connection factory properties" on page 98.

Performing the migration

You can migrate a project or EAR file using the version 6.1.0, use the adapter migration wizard. When the tool is finished, the migration is complete and you can work in the project or deploy the module.

Before you begin

Review the information in *Migration considerations*.

About this task

To perform the migration in WebSphere Integration Developer, complete the following steps.

Note: After migration is complete, the module will no longer be compatible with previous versions of WebSphere Process Server, WebSphere Enterprise Service Bus, or WebSphere Integration Developer.

Note: The following steps describe how to run the adapter migration wizard from the connector project context menu while in the J2EE perspective in WebSphere Integration Developer.

Note: You can also migrate in one of the following ways:

- Right-click the project in the J2EE perspective and select **Migrate → Migrate project**.
- From the Problems view, right-click a migration-specific message and select **Quick Fix** to correct the problem.

Procedure

1. Import the PI (project interchange) file for an existing project or the EAR (enterprise archive) file for an deployed application into the workspace.
2. Change to the J2EE perspective.
3. Right-click the module and select **Migrate → Update Connector Project**.
4. Review the tasks and warnings presented on the welcome page, and then select **Next**.
5. On the Select Projects window, select **Next**.

   By default, the wizard migrates the connector project and any dependent projects. If your project has dependent projects and you do not want to migrate one or more of them at this time, clear their check boxes in the **Dependent adapter project** list. You can rerun the wizard to migrate the dependent project at a later time. Previously migrated projects, projects with a current version, and projects that contain errors are unavailable for migration and are not selected.

6. On the Adapter Migration window, optionally review the migration changes, but do not change any selections. Click **Finish**.
7. Check the Problems view for messages from the migration wizard, which start with the string CWPAD.
8. If you are migrating an EAR file, optionally create a new EAR file with the migrated adapter and artifacts, and deploy it to WebSphere Process Server or WebSphere Enterprise Service Bus. For more information about exporting and deploying an EAR file, see the topics devoted to it in this documentation.

Results

The project or EAR file is migrated to version 6.1.0. You do not need to run the external service wizard after exiting the adapter migration wizard.

Updating but not migrating a version 6.0.2 project

Before you can use a version 6.0.2 project, without migrating the complete project, with WebSphere Adapter for JD Edwards EnterpriseOne, version 6.1.0 in WebSphere Integration Developer, version 6.1.0, use the migration wizard to update the project, and then correct a problem.

About this task

Because the internal name of the adapter changed in version 6.1.0, artifacts in a version 6.0.2 project must be updated to use the new name before you can use the adapter wizard in WebSphere Integration Developer, version 6.1.0. Use the migration wizard to update a version 6.0.2 project. Then use the Quick Fix feature of WebSphere Integration Developer to change the adapter name in project artifacts.

Procedure

1. Import the project interchange (PI) file into the workspace.
2. In the J2EE perspective, right-click the project name and click **Migrate → Update Connector Project**. The adapter migration wizard opens.
3. On the welcome page, click **Next**.
4. On the Select Projects window, select none of the dependent artifact projects, and then click **Finish**.
5. In the Quick Fix window, make sure the fix **Rename the referenced adapter** is selected, and then click **OK**.
6. If the error remains visible, click **Project → Clean**, select the project you just updated, and then click **OK**.

Results

The project can now be used with WebSphere Adapter for JD Edwards EnterpriseOne, version 6.1.0.

Chapter 3. Samples and tutorials

The WebSphere Integration Developer online samples/tutorials gallery includes samples and tutorials to help you use WebSphere Adapters.

You can access the online samples/tutorials gallery as follows:

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

Chapter 4. Configuring the module for deployment

To configure the adapter so that it can be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, use WebSphere Integration Developer to create a module, which is exported as an EAR file when you deploy the adapter. You then specify the business objects you want to discover and the system on which you want to discover them. After completing these steps, you will have successfully created an external service.

Roadmap for configuring the module

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

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

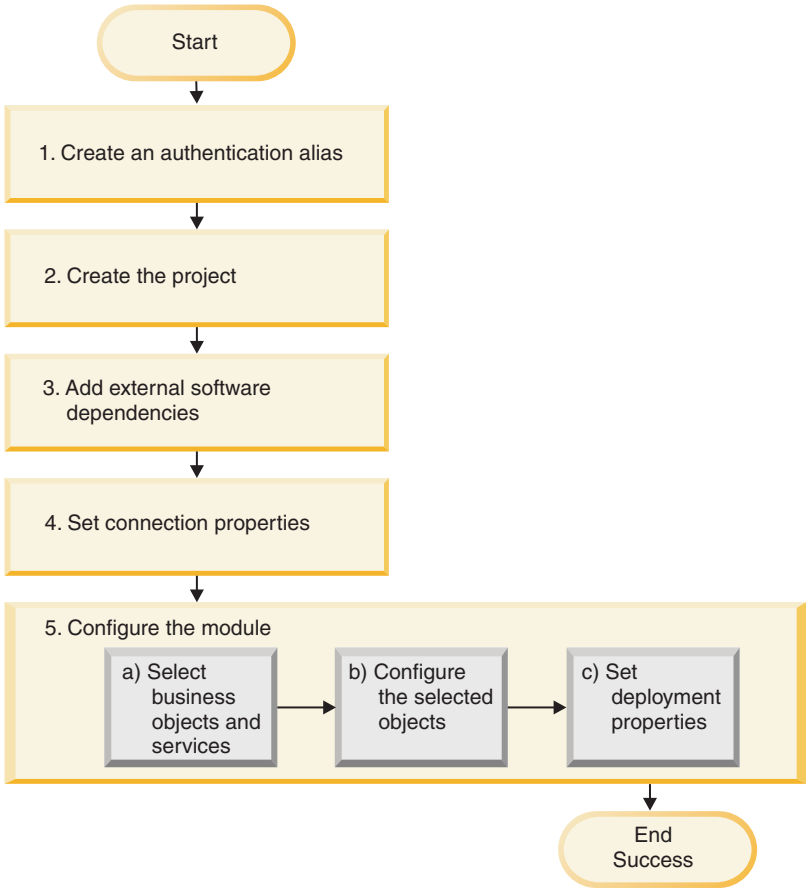


Figure 8. Roadmap for configuring the module

Configuring the module for deployment

This task consists of the following high-level steps:

1. Create an authentication alias to access the JD Edwards EnterpriseOne server with an encrypted password. This step is optional, depending on your policy for handling passwords and IDs. You perform this step using the administrative console on the server.
2. Create the project. First, start the external service wizard in WebSphere Integration Developer to begin the process of creating and deploying a module. The wizard creates a project that is used to organize the files associated with the module.
3. Add the external software dependencies required by WebSphere Adapter for JD Edwards EnterpriseOne to the project. These dependencies are also required when you export the module as an EAR file, and deploy the EAR file to the server.
4. Set connection properties that the external service wizard needs to connect to the JD Edwards EnterpriseOne server for discovery of objects and services.
5. Configure the module for outbound processing by using the external service wizard to find and select business objects and services from the JD Edwards EnterpriseOne server, and to generate business object definitions and related artifacts.
   a. Select business objects and services for outbound processing from the business integration components discovered by the external service wizard.
   b. Configure the selected objects by specifying operations and other properties that apply to all of the business objects.
   c. Set deployment properties that the adapter uses to connect to the JD Edwards EnterpriseOne server at run time. Then, generate the service by using the external service wizard to save the new module, which contains the business object or objects you configured, the import or export file, and the service interface.

Creating the authentication alias

An authentication alias is a feature that encrypts the password used by the adapter to access the JD Edwards EnterpriseOne server. After an authentication alias has been created, you can use it when you configure the adapter (instead of directly typing the user ID and password). Adapter properties are not encrypted, and if you directly type password, it is stored as clear text that can be viewed by others. Using the authentication alias is the default choice in the external service wizard.

Before you begin

To create an authentication alias, you must have access to the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus. The following procedure shows you how to gain access to the administrative console through WebSphere Integration Developer.

About this task

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

To create an authentication alias, use the following procedure.

Procedure

1. Start the administrative console.

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

   a. Start WebSphere Integration Developer by clicking **Start** → **Programs** → **IBM Software Development Platform** → **IBM WebSphere Integration Developer 6.1** → **IBM WebSphere Integration Developer 6.1**.

   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. If the WebSphere Integration Developer Welcome page is displayed, click **Go to the Business Integration perspective**.

      Note: If you minimize the Welcome page, WebSphere Integration Developer opens in the Business Integration perspective the next time you open it.

   d. Click the **Servers** tab.

   e. If the server does not show a status of **Started**, right-click the name of the server (for example, **WebSphere Process v6.1 Server**) and click **Start**.

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

   g. Log on to the administrative console. If your administrative console requires a user ID and password, type the ID and password and click **Log in**. If the user ID and password are not required, click **Log in**.

2. In the administrative console, click **Security** → **Secure administration, applications, and infrastructure**.

3. Under **Authentication**, expand **Java Authentication and Authorization Service**, then click **J2C authentication data**.

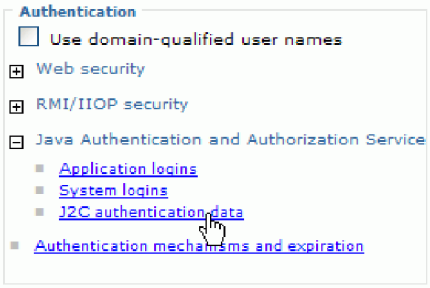


Figure 9. The Authentication section of the Secure administration, applications, and infrastructure window

4. Create an authentication alias.

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

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

   c. Type the user ID and password that are required to establish a connection to the JD Edwards EnterpriseOne server.

   d. Optional: Type a description of the alias.

   e. Click **OK**.

      The newly created alias is displayed.

      Important: The full name of the alias is *<node_name>/<alias_name>*. This full name is the one you use in subsequent configuration windows.

Example of full alias name:
widNode/JDE_auth_alias

f. Click **Save**, and then click **Save** again.

5. Click **New**.

Results

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

Creating the project

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

Before you begin

Make sure you have gathered the information you need to establish a connection to the JD Edwards EnterpriseOne server. For example, you need the role name and environment name for the JD Edwards EnterpriseOne environment on the JD Edwards EnterpriseOne server. You also need the user ID and password to access the JD Edwards EnterpriseOne server.

About this task

Start the external service wizard to create a project for the adapter in WebSphere Integration Developer. If you have an existing project, you can select it instead having the wizard creating one.

To start the external service wizard and create a project, use the following procedure.

Procedure

1. If WebSphere Integration Developer is not currently running, start it now.
   a. Click **Start** → **Programs** → **IBM Software Development Platform** → **IBM WebSphere Integration Developer 6.1** → **IBM WebSphere Integration Developer 6.1**.
   b. If you are prompted to specify a workspace, either accept the default value or select another workspace.

      The workspace is a directory where WebSphere Integration Developer stores your project.
   c. If the WebSphere Integration Developer Welcome page is displayed, click **Go to the Business Integration perspective**.

      Note: If you minimize the Welcome page, WebSphere Integration Developer opens in the Business Integration perspective the next time you open it.
2. To start the external service wizard, click **File** → **New** → **External Service**.
3. In the New external service window, make sure **Adapters** is selected, then click **Next**.

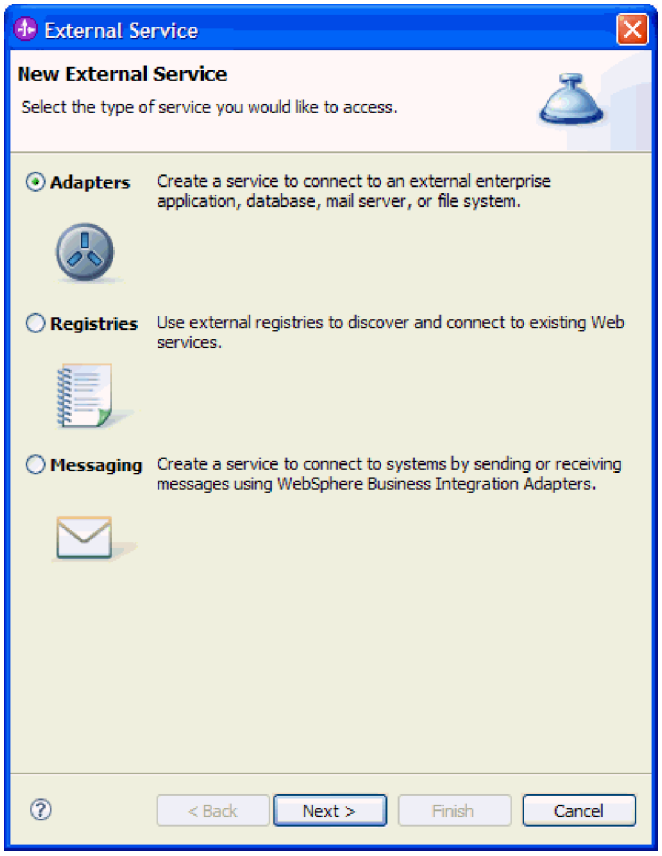


Figure 10. New external service window

4. From the Select an Adapter window, you can either create a new project or select an existing project.
   - To create a new project, perform the following steps:
     a. Select **IBM WebSphere Adapter for JD Edwards EnterpriseOne (IBM : 6.1.0)**, as shown in the following figure.

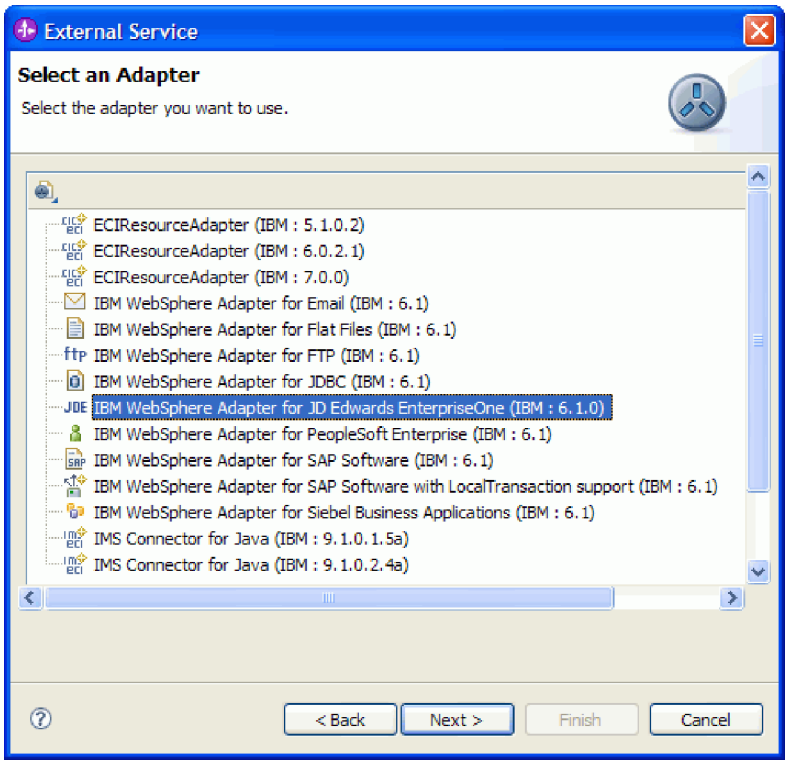


Figure 11. The Select an Adapter window

b. Click **Next**.

c. In the Adapter Import window, either accept the default name for the project (**CWYED_JDE**) or provide another name in the **Connector project** field, select the server from the **Target runtime** list (for example, **WebSphere Process Server v6.1**), then click **Next**.

- To select an existing project, perform the following steps:

  a. Expand **IBM WebSphere Adapter for JD Edwards EnterpriseOne (IBM : 6.1.0)**.

  b. Select a project.

  For example, if you have an existing project named CWYED_JDE, you can expand **IBM WebSphere Adapter for JD Edwards (IBM : 6.1.0)CWYED_JDE**, then select the connection information you used to create the project.

  c. Click **Next**.

Results

- If you created a new project, the project is created and is listed in the Business Integration window.
- If you selected an existing project, the project is selected.

What to do next

- If you created a new project, you will be prompted by the external service wizard to provide the location of the external software dependency files required to connect to the JD Edwards EnterpriseOne server.
- If you selected an existing project, and that project does not already have the required external software dependency files associated with it, you will be

prompted by the external service wizard to provide the location of the external software dependency files required to connect to the JD Edwards EnterpriseOne server.

- If you selected an existing project, and that project already has the required external software dependency files associated with it, you will be prompted by the external service wizard to specify the information required to establish a connection between the external service wizard and the JD Edwards EnterpriseOne server.

Adding external software dependencies

The JD Edwards EnterpriseOne application requires that you add external software dependencies to the project. These software dependencies enable the external service wizard to communicate with the JD Edwards EnterpriseOne environment.

Before you begin

Create the project, or select an existing project.

About this task

To obtain the required software dependency files and specify their location, use the following procedure.

Procedure

1. Obtain the JD Edwards EnterpriseOne software dependency files from your JD Edwards EnterpriseOne administrator. The files are listed in the following table.

   Note: The software dependencies differ, depending on which version of JD Edwards EnterpriseOne you use.

Table 1. External software dependency files required by 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, 8.96
connector.jar	Connector.jar	ApplicationAPIs_JAR.jar
database.jar	database.jar	ApplicationLogic_JAR.jar
jdeinterop.ini	jdeutil.jar	Base_JAR.jar
jdeLog.properties	jdbj.ini	BizLogicContainer_JAR.jar
kernel.jar	jdeinterop.ini	BizLogicContainerClient_JAR.jar
log4j.jar	jdelog.properties	Connector.jar
xalan.jar	kernel.jar	jdbj.ini
xerces.jar	log4j.jar	JdbjBase_JAR.jar
JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: • tnsnames.ora • classes12.zip	xalan.jar	JdbjInterfaces_JAR.jar
	xerces.jar	jdeinterop.ini

Table 1. External software dependency files required by 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, 8.96
	JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: • tnsnames.ora • classes12.zip	jdelog.properties
		JdeNet_JAR.jar
		log4j.jar
		PMApi_JAR.jar
		Spec_JAR.jar
		System_JAR.jar
		xalan.jar
		xerces.jar
		JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: • tnsnames.ora • classes12.zip

2. Copy the external dependency files to a temporary location. For example, copy them to C:\temp\JDE_dependencies\.

   Tip: If you put the JDBC driver files in their own folder, it makes them easier to browse to from the external service wizard. For example, if you are using an Oracle database server, you can put the tnsnames.ora and the classes12.zip files in the following location: C:\temp\JDE_dependencies\jdbc_driver\.
3. In the Required Files and Libraries window, use the following procedure to add the external dependency files to the project. The following figure shows the Required Files and Libraries window, with all of the required files selected for JD Edwards EnterpriseOne, version 8.96.

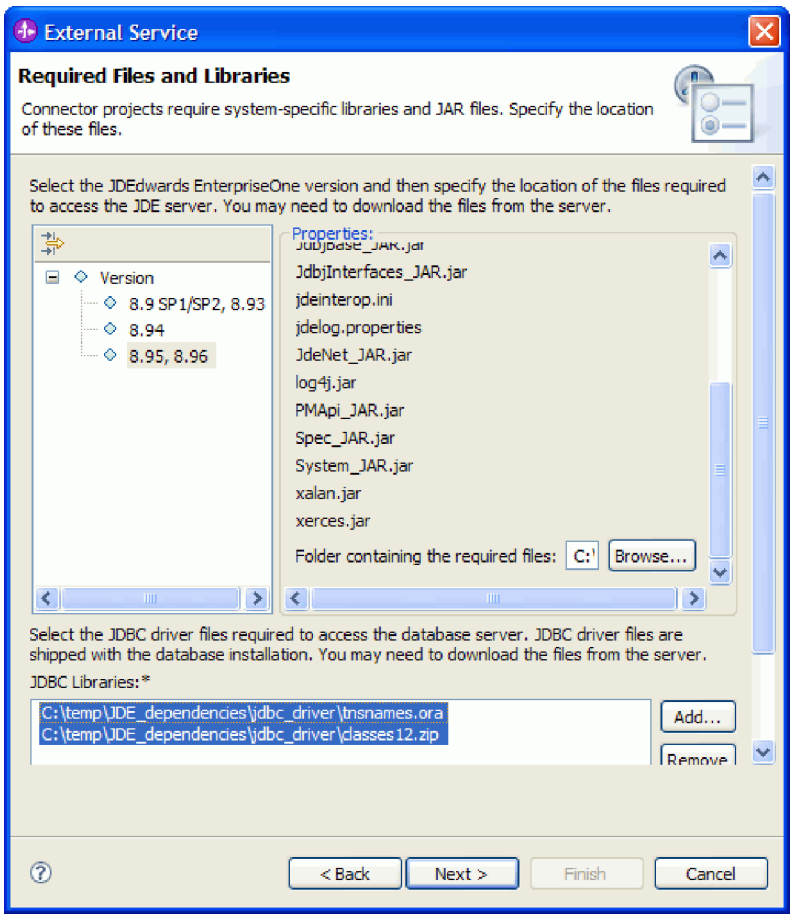


Figure 12. Required Files and Libraries window, showing the required files selected for JD Edwards EnterpriseOne, version 8.96

a. Select the version of JD Edwards EnterpriseOne from the left pane. The required files for the version you choose appear in the right pane.
b. To locate and select the required JAR, INI, and PROPERTIES files, click **Browse**, navigate to the folder that contains the files listed, then click **OK**.
c. To add the JDBC driver files, click **Add**, navigate to the location of the JDBC driver files, select the tnsnames.ora and classes12.zip files, then click **OK**.

4. Click **Next**.

Results

The external software dependency files are added to the project. The following figure shows the CWYED_JDE project after the dependency files have been added.

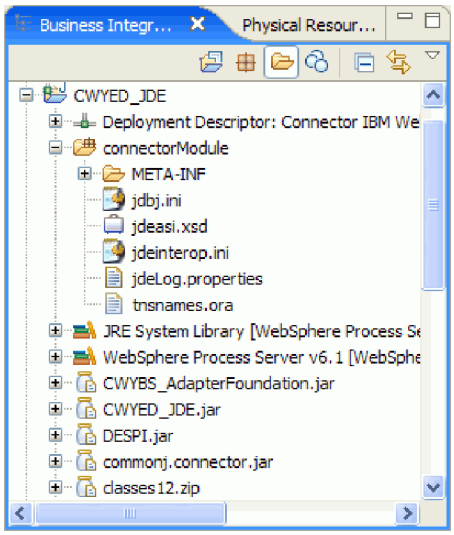


Figure 13. External software dependency files, after being added to the CWYED_JDE project

Verify or edit the connection information in some of the external software dependency files to make sure the external service wizard can connect with the JD Edwards EnterpriseOne server.

Editing external dependency files

Some of the external dependency files for JD Edwards EnterpriseOne contain editable information, such as the WebSphere Integration Developer workplace location and JD Edwards EnterpriseOne server name and port number. This type of information may need to be edited before you can establish a connection between the external service wizard and the JD Edwards EnterpriseOne server.

Before you begin

Make sure you have added the external dependency files to you project and that you know the location of the WebSphere Integration Developer workplace.

About this task

Verify that the information in you external dependency files correctly reflects the WebSphere Integration Developer workplace location and the connection information required by the JD Edwards EnterpriseOne environment. To do this, use the following procedure.

For further information about configuring the external software dependency files, refer to the *JD Edwards EnterpriseOne Tools Connectors* documentation for your version of JD Edwards EnterpriseOne.

Procedure

1. If the external service wizard is open, click **Cancel** to close it.
2. In the Business Integration pane, expand the project name, then expand **connecorModule**.
3. Verify the information in the jdbj.ini file.
   a. Double-click **jdbj.ini** to open it. The file opens in Notepad.
   b. Press **Ctrl+F** to search the file.

c. Enter `tns` in the Find what field.
d. Verify that the location listed for the tnsnames.ora file is the correct location for the WID workplace for this project. For example: tns=C:\IBM\wid6.1\workspace\CWYED_JDE\connectorModule\tnsnames.ora
e. Click **File** → **Save** to save any changes you made to the file.

4. Verify the information in the jdeinterop.ini file.
   a. Double-click **jdeinterop.ini** to open it. The file opens in Notepad.
   b. Verify that the server name and the port number are correct. You can obtain this information from the JD Edwards EnterpriseOne administrator.
   c. Click **File** → **Save** to save any changes you made to the file.
5. Verify the information in the jdelog.properties file.
   a. Double-click **jdelog.properties** to open it. The file opens in the right-pane of WebSphere Integration Developer.
   b. Verify that the information in this file is correct. This file specifies log levels and log file locations. You can obtain this information from the JD Edwards EnterpriseOne administrator.
   c. Click **File** → **Save** to save any changes you made to the file.

Results

The external software dependency files contain the correct information to allow a connection to be established between the external service wizard and the JD Edwards EnterpriseOne server.

In the Discovery and Configuration window, specify the connection properties that the external service wizard needs to connect to the JD Edwards EnterpriseOne environment.

Setting connection properties for the external service wizard

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

Before you begin

Make sure you have successfully added the external software dependency files, and that you have edited the connection information in the dependency files.

About this task

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

Note: You can optionally set bidirectional properties and logging properties in the same external service wizard window as you set the connection properties.

To specify the required connection properties and optional bidirectional and logging properties, use the following procedure.

Procedure

1. In the Discovery Configuration window, specify the configuration properties:
   a. In the Environment field, type the name of the JD Edwards EnterpriseOne environment.
   b. In the Role field, type the role name you use to access the JD Edwards EnterpriseOne environment.
   c. In the User name field, type the user name required to access the JD Edwards EnterpriseOne server.
   d. In the Password field, type the password you use to access the JD Edwards EnterpriseOne server.
2. Optional: If you need to set bidirectional properties, perform the following steps:
   a. Click **Advanced**.
   b. In the Bidi Properties section, select **Bidi transformation**.
   c. Set properties for your environment. See "Connection properties for the external service wizard" on page 92 for more information about these properties.

      The following figure shows an example of the Discovery Configuration window with the Advanced button selected and the Bidi Properties box selected.

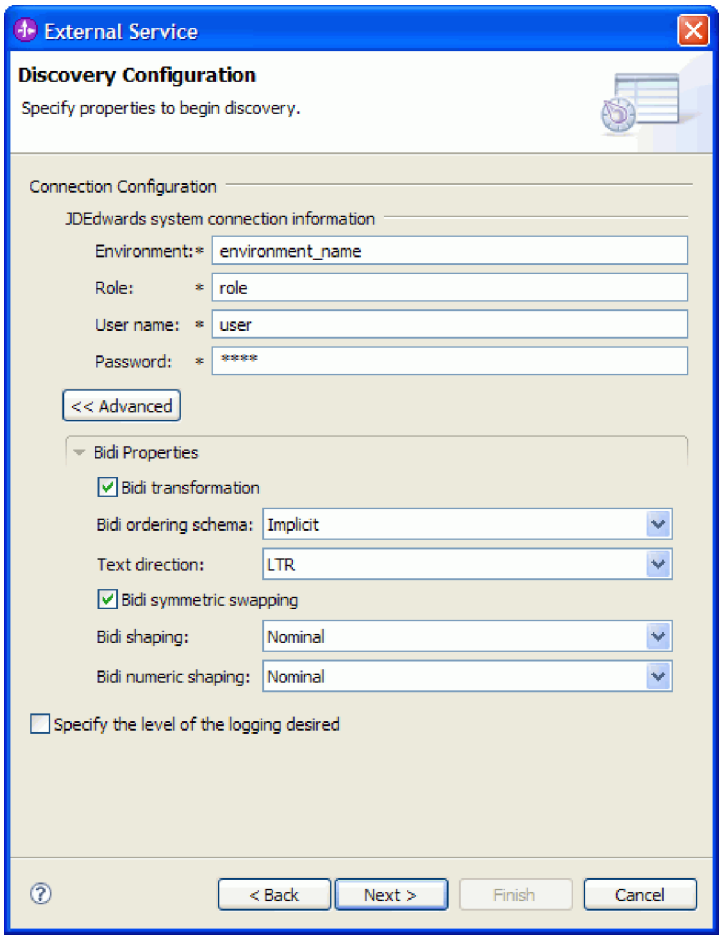


Figure 14. The Discovery Configuration window, showing optional Bidi Properties

3. Optional: To set logging properties for the external service wizard, perform the following steps:
   a. Select **Specify the level of the logging desired**.
   b. Change the location of the log file output location by clicking **Browse** and selecting a different location.
   c. Set the **Logging Level**.

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

      Note: This log pertains to the external service wizard only, not to the operation of the adapter.
4. Click **Next**.

Results

The external service wizard contacts the JD Edwards EnterpriseOne server, using the information you provided (such as user name and password) to log in. You see the Object Discovery and Selection window.

Specify search criteria that the external service wizard uses to discover functions or data on the JD Edwards EnterpriseOne server.

Configuring the module for outbound processing

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

Generating business functions using external service discovery

To configure WebSphere Adapter for JD Edwards EnterpriseOne using business functions, use the external service wizard in WebSphere Integration Developer to select business functions that are in the JD Edwards EnterpriseOne server, and generate business object definitions and related artifacts for outbound processing.

Selecting business objects and services

To specify which business function you want to call and which data you want to process, you provide information in the external service wizard.

Before you begin

Make sure you have set the connection properties for the external service wizard.

About this task

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

To specify the search criteria and select a business function, use the following procedure.

Procedure

1. In the Object Discovery and Selection window, make sure the query is set up to find business objects in the JD Edwards EnterpriseOne server. In the Query field, make sure you see **Business Functions=true**. If you see **Business Functions=false**, use the following procedure to set Business Functions to true.
   a. Click **Edit Query**.
   b. In the Query Properties window, select **Business Functions**.

      Note: You can optionally clear the check box for **XML Lists** so the query finds only business functions. If you leave **XML Lists** selected, the query looks for both XML Lists and business functions.
   c. Click **OK**.
2. Click **Run Query**.
3. In the Discovered objects list, indicate which business function you want to work with.
   a. Click **Business functions**. This enables the filter button.

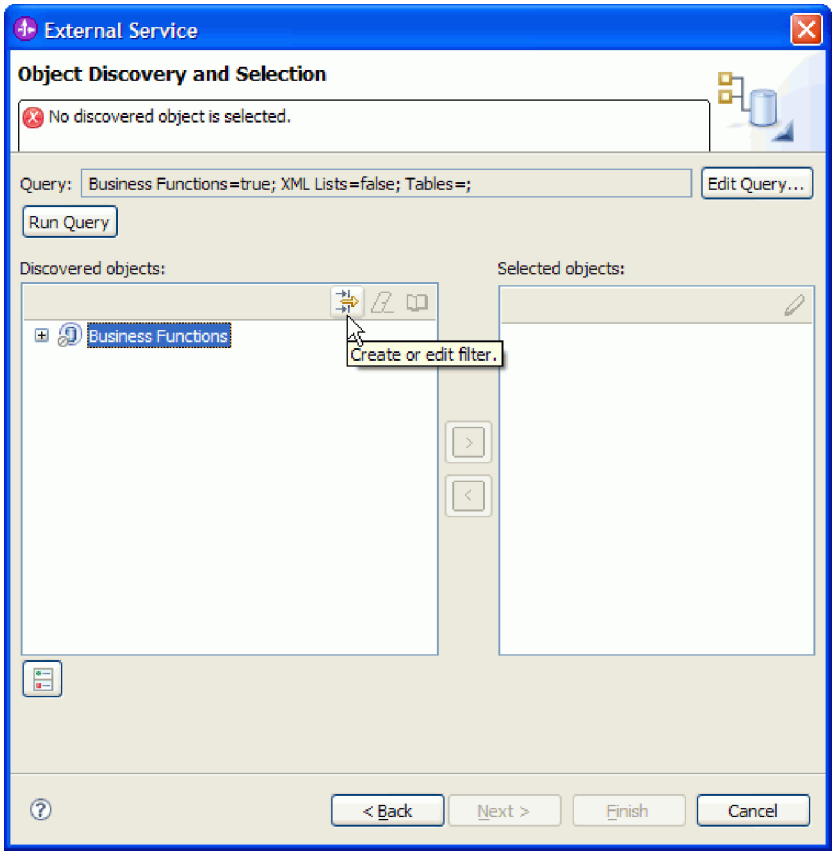


Figure 15. Object Discovery and Selection window, showing the filter button enabled

   b. Search for business functions using one of the following methods.
      - Click the filter button if you know the Library name (for example, **CFIN**), the C File name (for example, **B01000033**), and the business function name (for example, GetEffectiveAddress) for the business function you want.

- If you do not know the Library name, C File name, or business function name, expand **Business functions**, expand the Library name, expand the C File name, then select the business function.

  Tip: The mouseover text indicates the purpose of each Library name and C File name, as indicated in the following two figures.

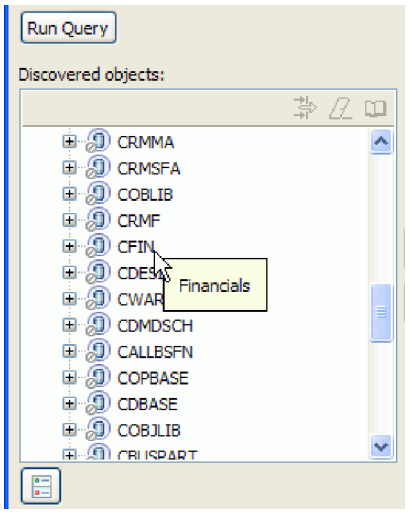


Figure 16. Object Discovery and Selection window, showing discovered business functions with the mouseover text for the CFIN Library displayed

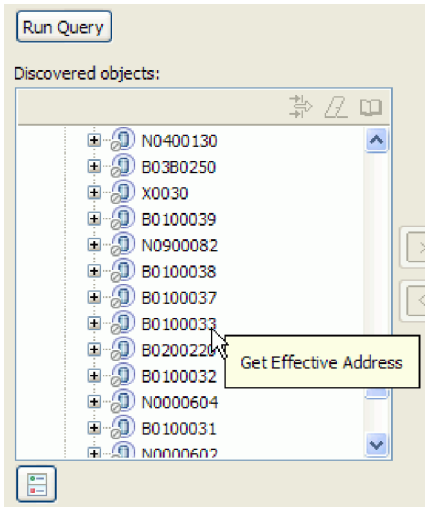


Figure 17. Object Discovery and Selection window, showing discovered business functions with the mouseover text for the B0100033 C File displayed

4. Select the business function. For example, if you navigated to **Business Functions** → **CFIN** → **B0100033**, select **GetEffectiveAddress**.
5. Click the arrow button to add the business function to the **Selected objects** list.

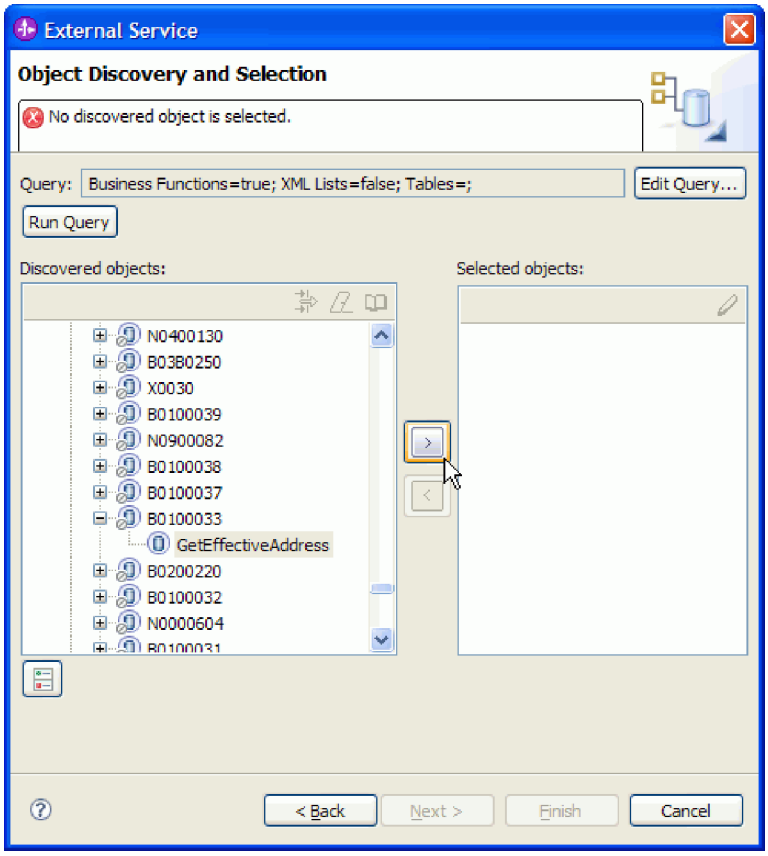


Figure 18. Clicking arrow to add business function to Selected objects list

6. In the Configuration Properties window, either accept the default business object name or type a different name. The default name is the name of the business function.

   Note: The business object name has no semantic value, so you can give it a meaningful name, and the name you assign it will not impact how the business object functions.
7. Click **OK** to add the business function to the list of business objects to be imported.
8. Click **Next**.

Results

You have selected the business function you want to work with and selected a name for it.

From the Configure Composite Properties window, specify a business object container name and associated operation. Optionally specify a namespace and directory to which the generated business object will be stored and indicate whether you want a business graph generated.

Configuring the selected objects

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

Before you begin

Make sure you have selected and imported the business function.

About this task

To configure the business function, use the following procedure.

Procedure

1. Optional: In the Configure Composite Properties window, enter the following information.
   a. If you want the business function to be enclosed within a business graph, leave **Business Graph** selected. Otherwise, remove the check.
   b. In the **Business object namespace** field, 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 external service wizard), change the namespace value. For example, you could change the namespace to http://www.ibm.com/xmlns/prod/websphere/j2ca/jde1.
   c. In the **Relative path** field, specify the directory to store the business object.
   d. In the **Maximum number of records** field, specify the maximum number of records to retrieve when processing a RetrieveAll operation. The default value is 100.
   e. In the **Timeout (milliseconds)** field, specify a timeout value in milliseconds. If no value is set, the default value is 30,000 milliseconds (or 30 seconds).
2. Required: Add a business object container to the business function, and assign a name to the business object container. All business functions require that you add a business object container.

   Note: You can add one or many business functions inside a single business object container.
   a. Click **Add container business object**.

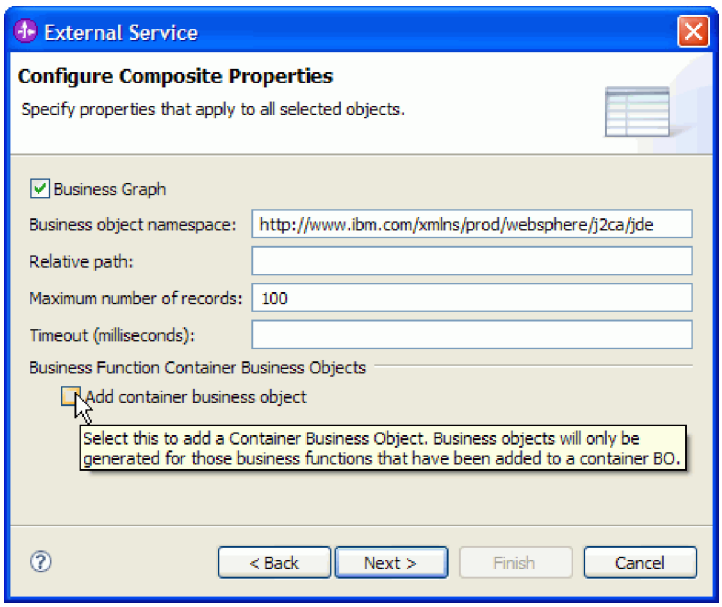


Figure 19. Configure Composite Properties window, showing the required "Add container business object" check box

b. Type a name in the **Container business object name field**. You can type any meaningful name. For example, if the business function is called GetEffectiveAddress, you can type **GetEffectiveAddressContainer**.

3. Required: Associate the business functions with an operation.
   a. Click the **Add** button next to one of the following fields. Each field represents an available operation that you can associate with the business function.
      - **Business functions for create**
      - **Business functions for retrieve**
      - **Business functions for update**
      - **Business functions for delete**
      - **Business functions for execute**
   b. In the **Add** window, select the business function you imported, then click **OK**. The business function appears in the operation field you selected.

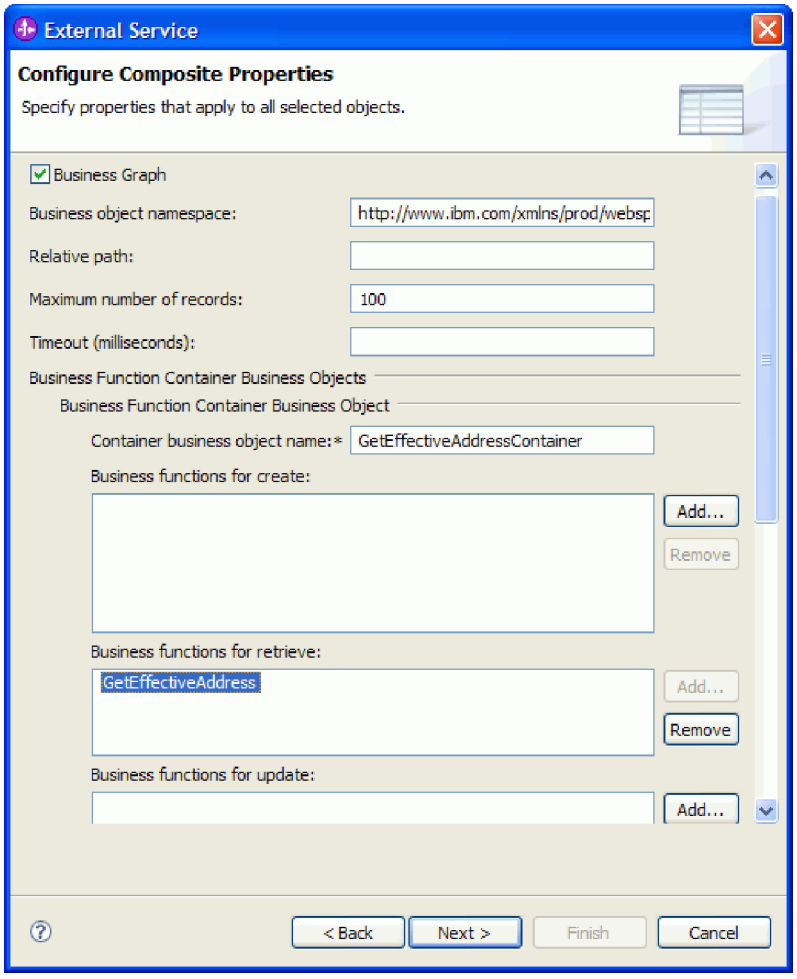


Figure 20. Configure Composite Properties window, showing the business function after it is associated with the "retrieve" operation

4. Click **Next**.

Results

You specified a name for the business object container and selected an operation for the business function. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create the module, and associate the adapter with the module.

Before you begin

Make sure you have configured the business function. The Service Generation and Deployment Configuration window should be displayed.

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.

Procedure

1. Optional: To change the default operation name, use the following procedure.
   a. Click **Edit Operations**.
   b. In the Edit Operation Names window, select the operation you want to edit, then click **Edit**.
   c. In the Add/Edit properties window, type a new operation name and optional description, then click **Finish**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the JD Edwards EnterpriseOne server:
   - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
   - If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.
   - **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
   - **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:
   - Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
   - Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

     If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties.

If you want to change the resource adapter name in the import or export, use the assembly editor in WebSphere Integration Developer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

a. Type a value for **JNDI Lookup Name**.
b. Click **Next**.
c. Go to step 9.

5. In the Connection properties section, set or change any connection properties that apply to your configuration.

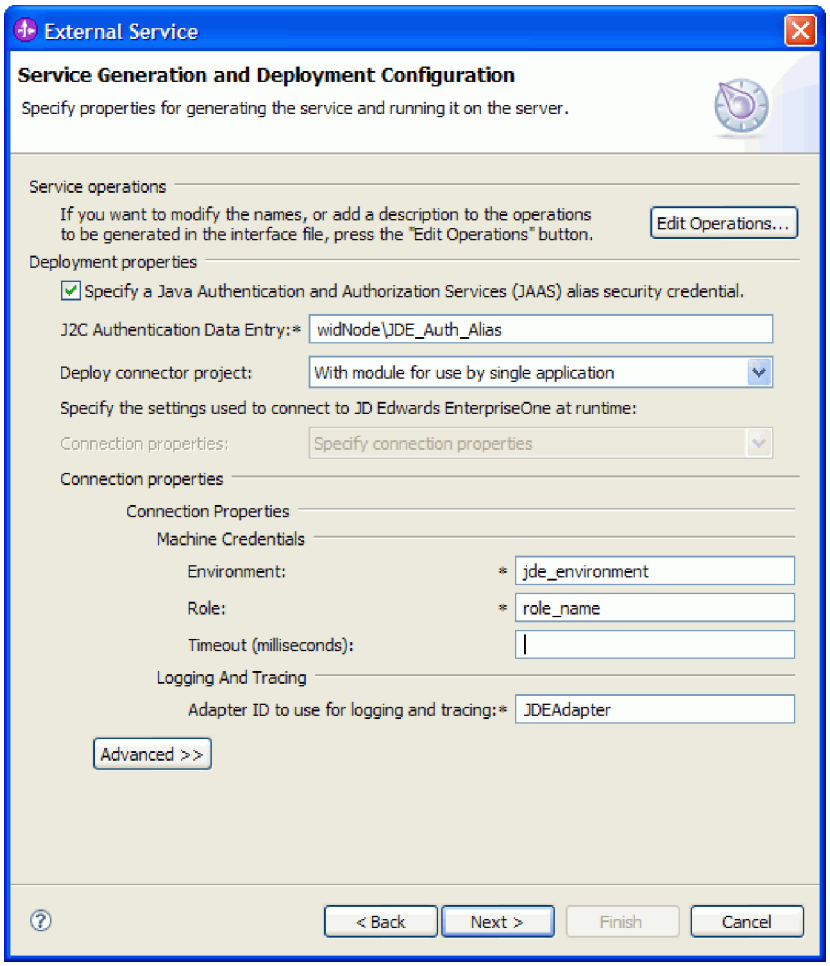


Figure 21. Service Generation and Deployment Configuration window, showing the **Connection properties** *section*

See "Managed connection factory properties" on page 98 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. In the Logging and Tracing section, keep the default Adapter ID (JDEAdapter) in the **Adapter ID to use for logging and tracing** field, or type a new Adapter ID.
7. To set additional properties, click **Advanced**.
8. Click **Next**.
9. Create a new module.

a. In the **Service Location Properties** window, click **New** in the **Module** field.
b. In the Integration Project window, select **Create a module project** or **Create a mediation 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 directory specified in the **Location** field.

   This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location, or click **Browse** and select the location.
d. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
e. 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 Service Location Properties window.

10. In the Service Location Properties window, you can specify deployment properties for the new module by performing the following steps.
    a. If you want to change the default namespace, clear the **Use Default Namespace** check box and type a new path in the **Namespace** field.
    b. Specify the folder within the module where the service description should be saved by typing a name in the **Folder** field or by browsing for a folder.
    c. Change the name of the interface.

       The default name is `JDEOutboundInterface.` You can change it to a more descriptive title if you prefer.
    d. If you want to save the business objects so that they can be used by another application, click Save business objects to a library, then select a library from the list, or click **New** to create a new library.
    e. Type a description of the module.
11. Click **Finish**.

Results

The new module is created, configured and saved to the Business Integration perspective of WebSphere Integration Developer.

What to do next

Export the module as an EAR file for deployment.

Generating XML Lists using external service discovery

To configure WebSphere Adapter for JD Edwards EnterpriseOne using XML Lists, use the external service wizard in WebSphere Integration Developer to select XML Lists that are in the JD Edwards EnterpriseOne server, and generate business object definitions and related artifacts for outbound processing.

Selecting business objects and services

To specify which data you want to process for the XML List, you provide information in the external service wizard.

Before you begin

Make sure you have set the connection properties for the external service wizard. Also, make sure you have gathered the table information, including table names and table types, for the XML Lists you want to create.

Tip: Use the JD Edwards EnterpriseOne Universal Table Browser (UTB) to gather the table information for the XML Lists you want to create. For information about using the JD Edwards EnterpriseOne UTB, refer to the *JD Edwards EnterpriseOne Tools 8.96 System Administration Guide*.

About this task

Specify search criteria that the external service wizard uses to discover database table information on the JD Edwards EnterpriseOne server. The external service wizard returns a list of objects that meet the search criteria.

To specify the search criteria for creating an XML List, use the following procedure.

Procedure

1. In the Object Discovery and Selection window, edit the query to prepare it to find the table information that is required for creating the XML List.
   a. Click **Edit Query**.
   b. In the Query Properties window, select **XML Lists**.

      Note: You can optionally clear the check box for **Business Functions** so the query finds only XML Lists. If you leave **Business Functions** selected, the query returns both XML Lists and business functions.
   c. To add the table name, click **Add**.
   d. In the **Add** window, enter the name of the database table you want to add to the query. For example, enter `F0116` for the database table. If you do not know the name of the table you need, you can use the JD Edwards EnterpriseOne Universal Table Browser (UTB) to find it on the JD Edwards EnterpriseOne server.
   e. Click **OK** in the Add window. The table name you entered is displayed in the Tables list.
   f. Click **OK** in the Query Properties window.

   The query is now ready to discover the JD Edwards EnterpriseOne database table you specified in order to create the XML List.

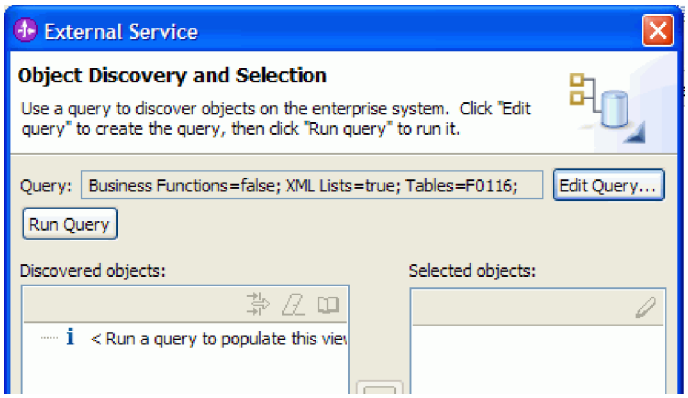


Figure 22. Object Discovery and Selection window, showing the Query field ready to run an XML List query

2. Run the XML List query to find and discover the table in the JD Edwards EnterpriseOne server that matches the table you specified.
   a. Click **Run Query**. The external service wizard queries the JD Edwards EnterpriseOne server. The results of the query appear in the Discovered objects list.
   b. In the Discovered objects list, expand the **XML Lists**.
   c. Navigate to the table that matches the table you created.
   d. Click the arrow button to add the table to the **Selected objects** list.

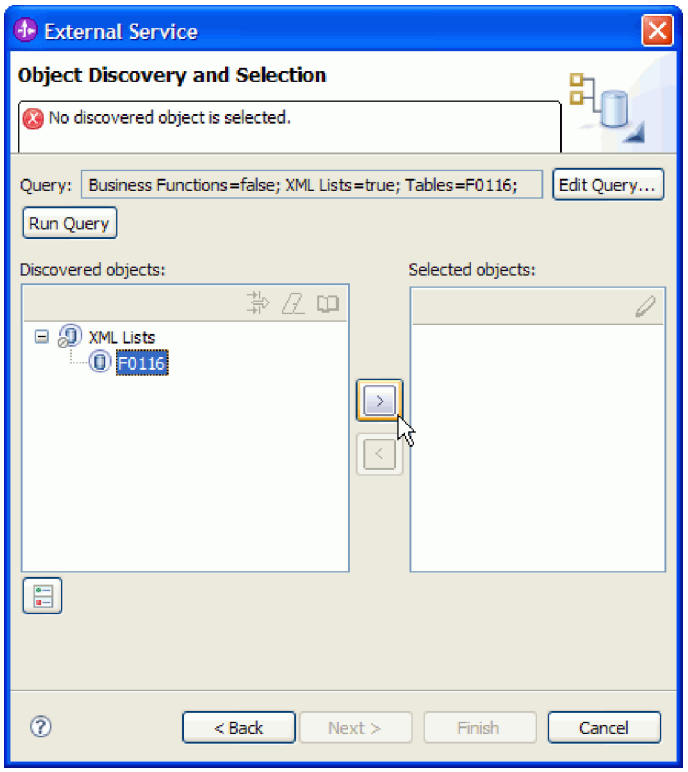


Figure 23. Clicking arrow to add the discovered table to Selected objects list

3. In the Configuration Parameters window, add search criteria to the XML List query before importing the data from the JD Edwards EnterpriseOne server. This allows you to specify query parameters such as table type and sorting conditions.
   a. In the Business Object Name field, you can either keep the default business object name or rename it to suit your needs. The default name is the name of the table.

      Note: The business object name has no semantic value, so you can give it a meaningful name, and the name you assign it will not impact how the business object functions.
   b. In the Table type field, select the type of table from the list. For example, select OWTABLE table type for the F0116 table.

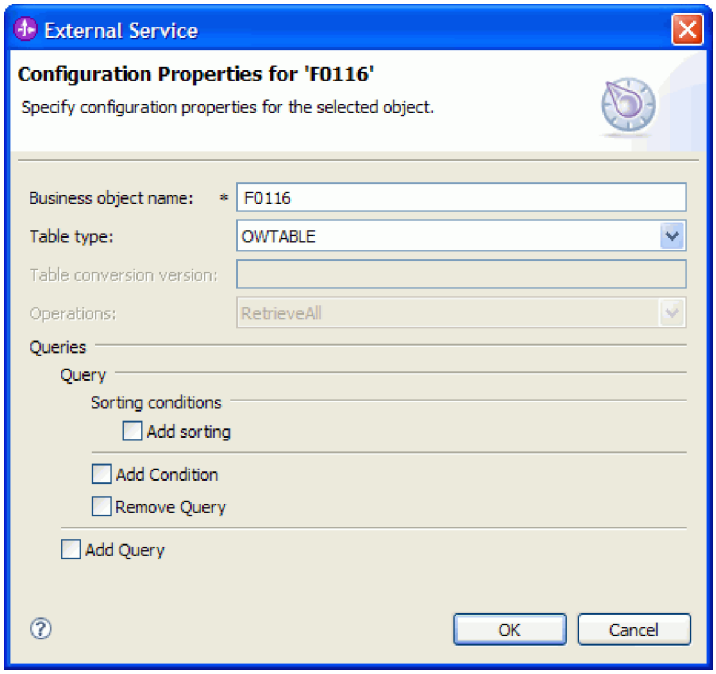


Figure 24. Configuration Properties window, showing OWTABLE selected in the Table type field

The following table types are available.

Table 2. Table types for XML Lists

Table type	Description	When to use
OWTABLE	A JD Edwards EnterpriseOne database table	Use this table type if the table you want is located in the JD Edwards EnterpriseOne database.
OWVIEW	A business view used to define the relationship between two or more tables and joins the data into a single view	Use this business view as input to the TABLE_CONVERSION table type.
FOREIGN_TABLE	A non-JD Edwards EnterpriseOne database table that resides in a database supported by JD Edwards EnterpriseOne, such as Oracle, Access, iSeries®, or SQL Server	Use this table type if the table you want is located in a non-JD Edwards EnterpriseOne database.

Table 2. Table types for XML Lists (continued)

Table type	Description	When to use
TABLE_CONVERSION	A table type that uses batch processes which allow you to rapidly manipulate data in tables. You can set up table conversions as templates, running them multiples times, then revising them using different versions to suit the needs of your environment.	Use this table type if you plan to manipulate batches of data in one of the following ways: • Data Conversion: allows you to transfer data from an input table or business view into output tables; also allows you to update records in a table or business view • Data Copy: allows you to copy tables from one data source or environment to another data source or environment when the tables are identical • Data Copy with Table Input: allows you to copy tables based on information from an input table. • Batch Delete: allows you to delete records from a table or business view

c. To add a sorting condition to the query, click **Add Sorting**, click **Select** to select the attribute you want sorted, then select **ASCENDING** or **DESCENDING** from the Sorting list.

   Note: If you want to remove a sorting condition, click **Remove Sorting Condition**.

d. To add other conditions to the query, click **Add Condition**, then select one of the following conditions.
   - **Attribute**: Select the attribute for which you want to add a condition.
   - **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.

   Note: If no conditions are specified, all records are retrieved. If you want to add multiple conditions to the query, click **Add Condition** again. If you want to remove any unwanted conditions, click **Remove condition**.

e. Click **OK**. The table name is displayed in the Selected objects list.

4. Click **Next**.

Results

You have selected the table you want to work with and configured it for the XML List you want to create.

From the Configure Composite Properties window, you can optionally specify a namespace and directory to which the generated XML List will be stored.

Configuring the selected objects

To configure the XML List, you specify information about the object, such as the maximum number of records to retrieve.

Before you begin

Make sure you have selected and imported the XML List.

About this task

To configure the XML List, use the following procedure.

Procedure

1. Optional: In the Configure Composite Properties window, enter the following information.

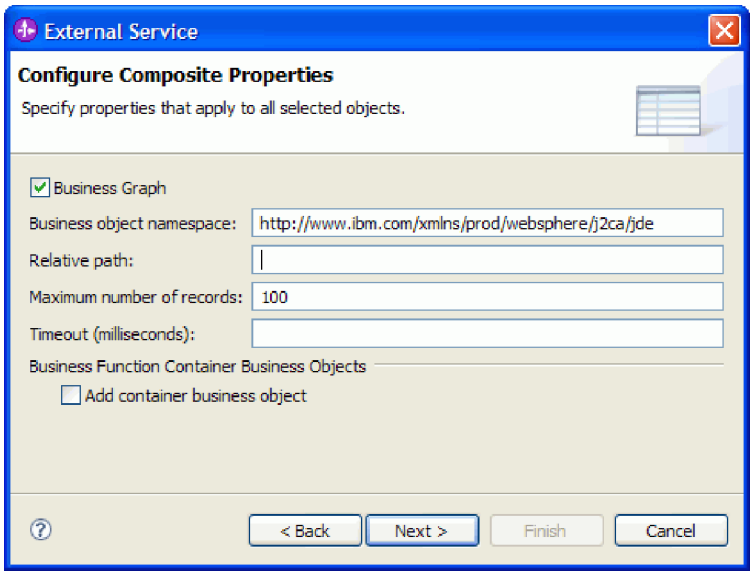


Figure 25. Configure Composite Properties window, showing optional configuration settings for XML Lists

a. Leave **Business Graph** selected.

   Note: This is required for XML Lists, so that multiple records can be retrieved.

b. In the **Business object namespace** field, use the default namespace (http://www.ibm.com/xmlns/prod/websphere/j2ca/jde) except in the following circumstance: if you are adding the XML List to an existing module and the module already includes that XML List (from an earlier run of the external service wizard), change the namespace value. For example, you could change the namespace to http://www.ibm.com/xmlns/prod/websphere/j2ca/jde1.

c. In the **Relative path** field, specify the directory to store the XML List.

d. In the **Maximum number of records** field, specify the maximum number of records to retrieve when processing a RetrieveAll operation. The default value is 100.

e. In the **Timeout (milliseconds)** field, specify a timeout value in milliseconds. If no value is set, the default value is 30,000 milliseconds (or 30 seconds).

Important: You do not need to add a business object container to or associate an operation with the XML List object. The business object container name and associated operation for the XML List is set by default as follows:

- Business object container name: <*XML_List_object*>Container
- Operation: RetrieveAll

2. Click **Next**.

Results

You have set optional configuration parameters for the XML List object. The external service wizard automatically assigned the business object container name and associated operation for the XML List. The Generate Artifacts window opens.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create the module, and associate the adapter with the module.

Before you begin

Make sure you have configured the XML List. The Service Generation and Deployment Configuration window should be displayed.

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.

Procedure

1. Optional: To change the default operation name, use the following procedure.
   a. Click **Edit Operations**.
   b. In the Edit Operation Names window, select the operation you want to edit, then click **Edit**.
   c. In the Add/Edit properties window, type a new operation name and optional description, then click **Finish**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the JD Edwards EnterpriseOne server:
   - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
   - If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

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

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:
   - Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
   - Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

     If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in WebSphere Integration Developer to change the value in the import or export.

     When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

     a. Type a value for **JNDI Lookup Name**.

     b. Click **Next**.

     c. Go to step 9.
5. In the Connection properties section, set or change any connection properties that apply to your configuration.

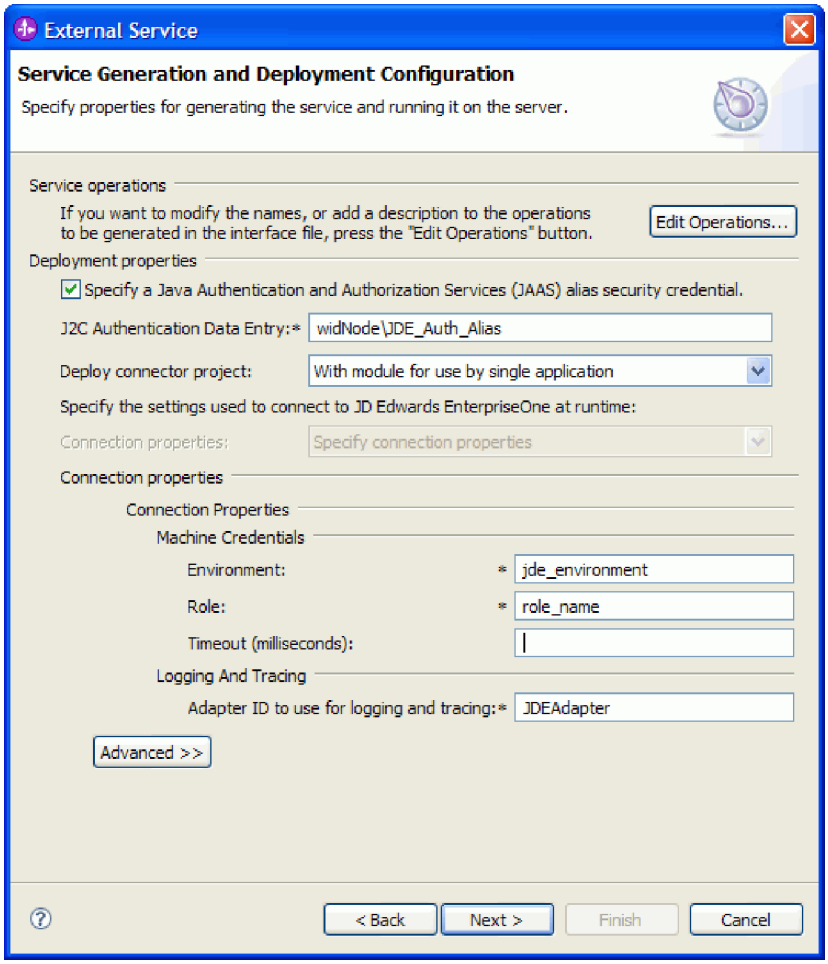


Figure 26. Service Generation and Deployment Configuration window, showing the **Connection properties** *section*

See "Managed connection factory properties" on page 98 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. In the Logging and Tracing section, keep the default Adapter ID (JDEAdapter) in the **Adapter ID to use for logging and tracing** field, or type a new Adapter ID.
7. To set additional properties, click **Advanced**.
8. Click **Next**.
9. Create a new module.
   a. In the **Service Location Properties** window, click **New** in the **Module** field.
   b. In the Integration Project window, select **Create a module project** or **Create a mediation 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 directory specified in the **Location** field.

This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location, or click **Browse** and select the location.

d. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.

e. 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 Service Location Properties window.

10. In the Service Location Properties window, perform the following steps to complete the process of generating the module.

    a. If you want to change the default namespace, clear the **Use Default Namespace** check box and type a new path in the **Namespace** field.

    b. Specify the folder within the module where the service description should be saved by typing a name in the **Folder** field or by browsing for a folder.

    c. Change the name of the interface.

    The default name is `JDEOutboundInterface.` You can change it to a more descriptive title if you prefer.

    d. If you want to save the business objects so that they can be used by another application, click Save business objects to a library, then select a library from the list, or click **New** to create a new library.

    e. Type a description of the module.

11. Click **Finish**.

Results

The new module is created, configured and saved to the Business Integration perspective of WebSphere Integration Developer.

What to do next

Export the module as an EAR file for deployment.

Chapter 5. Changing interaction specification properties using the assembly editor

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

Before you begin

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

About this task

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

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

Procedure

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

Results

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

What to do next

Deploy the module.

Chapter 6. Deploying the module

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

Deployment environments

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

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

Deploying the module for testing

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

Adding the module to the server

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

About this task

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

Procedure

1. Add the module to the server
   a. Switch to the servers view. In WebSphere Integration Developer, select **Windows → Show View → Servers**
   a. Start the server. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click on the server, and then select **Start**.
2. When the server status is *Started*, right-click on the server, and select **Add and remove projects**.
3. In the Add and Remove Projects screen, select your project and click **Add**. The project moves from the **Available projects** list to the **Configured projects** list.
4. Click **Finish**. This deploys the module on the server.

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

What to do next

Test the functionality of your module and the adapter.

Testing the module for outbound processing using the test client

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

Before you begin

You need to add the module to the server first.

About this task

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

Procedure

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

What to do next

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

Deploying the module for production

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

Adding external software dependencies to the server runtime environment

You must copy the required external software dependency files to your runtime environment before you can run your adapter applications.

About this task

To copy the required files to WebSphere Process Server or WebSphere Enterprise Service Bus, use the following procedure.

Procedure

1. Obtain the external software dependency files from your JD Edwards EnterpriseOneadministrator. The files are listed in Table 3 on page 59.

Table 3. Files to be copied

JD Edwards EnterpriseOne, version 8.9 (SP1, SP2), 8.93	**JD Edwards EnterpriseOne, version 8.94**	**JD Edwards EnterpriseOne, version 8.95, 8.96**
connector.jar	Connector.jar	ApplicationAPIs_JAR.jar
database.jar	database.jar	ApplicationLogic_JAR.jar
jdeinterop.ini	jdeutil.jar	Base_JAR.jar
jdeLog.properties	jdbj.ini	BizLogicContainer_JAR.jar
kernel.jar	jdeinterop.ini	BizLogicContainerClient_JAR.jar
log4j.jar	jdelog.properties	Connector.jar
xalan.jar	kernel.jar	jdbj.ini
xerces.jar	log4j.jar	JdbjBase_JAR.jar
JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: • tnsnames.ora • classes12.zip	xalan.jar	JdbjInterfaces_JAR.jar
	xerces.jar	jdeinterop.ini
	JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: • tnsnames.ora • classes12.zip	jdelog.properties
		JdeNet_JAR.jar
		log4j.jar
		PMApi_JAR.jar
		Spec_JAR.jar
		System_JAR.jar
		xalan.jar
		xerces.jar
		JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: • tnsnames.ora • classes12.zip

2. Copy the files listed in Table 3 to the installation directory of WebSphere Process Server or WebSphere Enterprise Service Bus. The installation directory is typically in the runtimes\bi_v6 directory of theWebSphere Integration Developer installation directory.

Results

The external software dependency files for JD Edwards EnterpriseOne are now part of your runtime environment.

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

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

Before you begin

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

About this task

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

Procedure

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

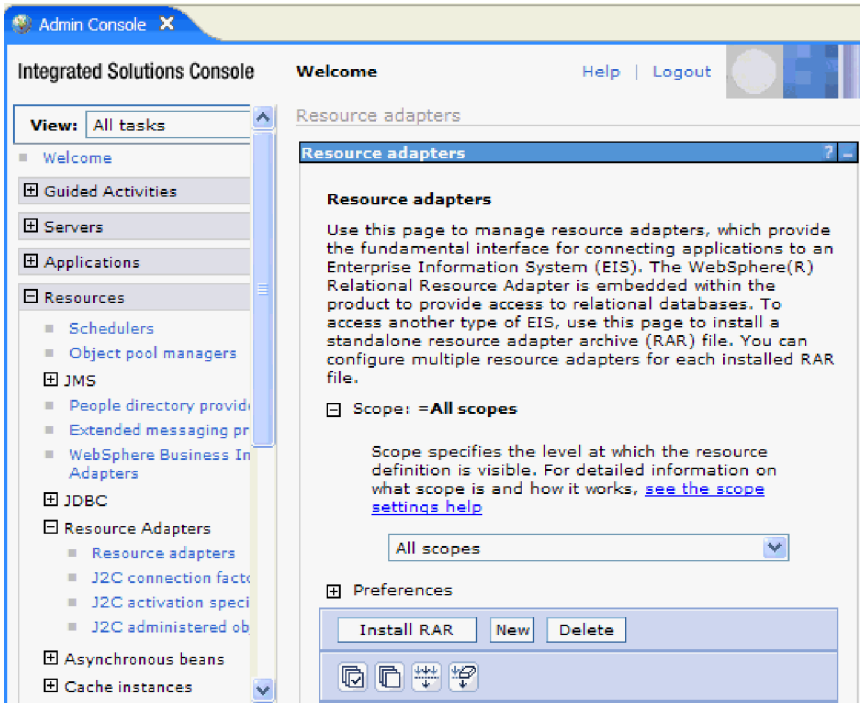


Figure 27. The Install RAR button on the Resource adapters page

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

   The RAR files are typically installed in the following path:
   WID_installation_directory/ResourceAdapters/*adapter_name*/deploy/*adapter*.rar
5. Click **Next**.

6. From the Resource adapters page, optionally change the name of the adapter and add a description.
7. Click **OK**.
8. Click **Save** in the **Messages** box at the top of the page.

What to do next

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

Exporting the module as an EAR file

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

Before you begin

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

About this task

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

Procedure

1. Right-click the module and select **Export**.
2. In the Select window, expand **J2EE**.
3. Select **EAR file** and click **Next**.

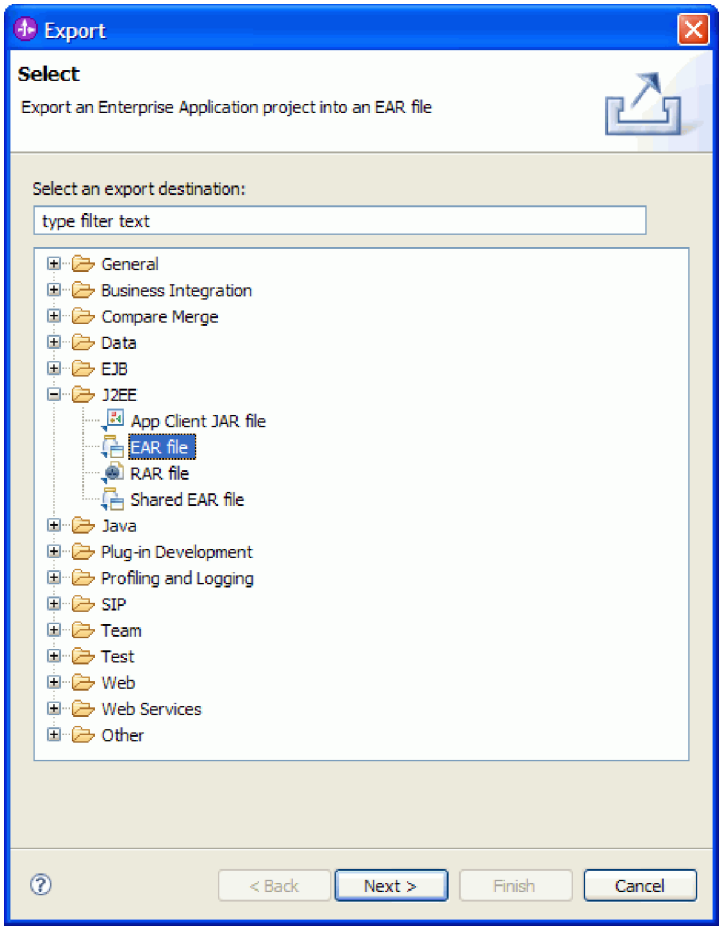


Figure 28. Selecting **EAR file** *from the Select window*

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

Results

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

What to do next

Install the module in the administrative console. This deploys the module to WebSphere Process Server.

Installing the EAR file

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

Before you begin

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

About this task

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

Procedure

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

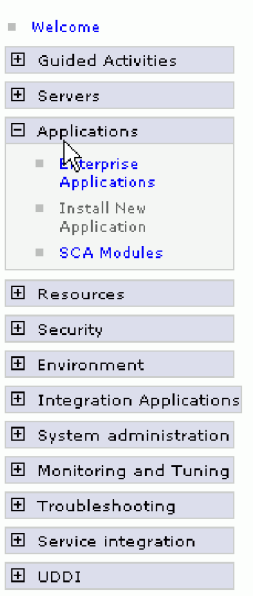


Figure 29. Preparing for the application installation window

3. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by "App."
4. Optional: If you are deploying to a clustered environment, complete the following steps.
   a. On the **Step 2: Mapping modules to servers** window, select the module.
   b. Select the name of the server cluster.
   c. Click **Apply**.
5. Click **Next** to open the Summary. Verify that all settings are correct and click **Finish**.
6. Optional: If you are using an authentication alias, complete the following steps:

a. Expand **Security** and select **Business Integration Authentication Aliases**.
b. Select the authentication alias that you want to configure. You must have administrator or operator authority to make changes to authentication alias configurations.
c. Optional: If it is not already filled in, type the **User name**.
d. If it is not already filled in, type the **Password**.
e. If it is not already filled in, type the password again in the **Confirm Password** field.
f. Click **OK**.

Results

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

What to do next

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

Chapter 7. Administering the adapter module

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

Changing configuration properties for embedded adapters

To change configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment. You can update resource adapter properties (used for general adapter operation) and managed connection factory properties (used for outbound processing).

Setting resource adapter properties for embedded adapters

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

Before you begin

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

About this task

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

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

Procedure

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

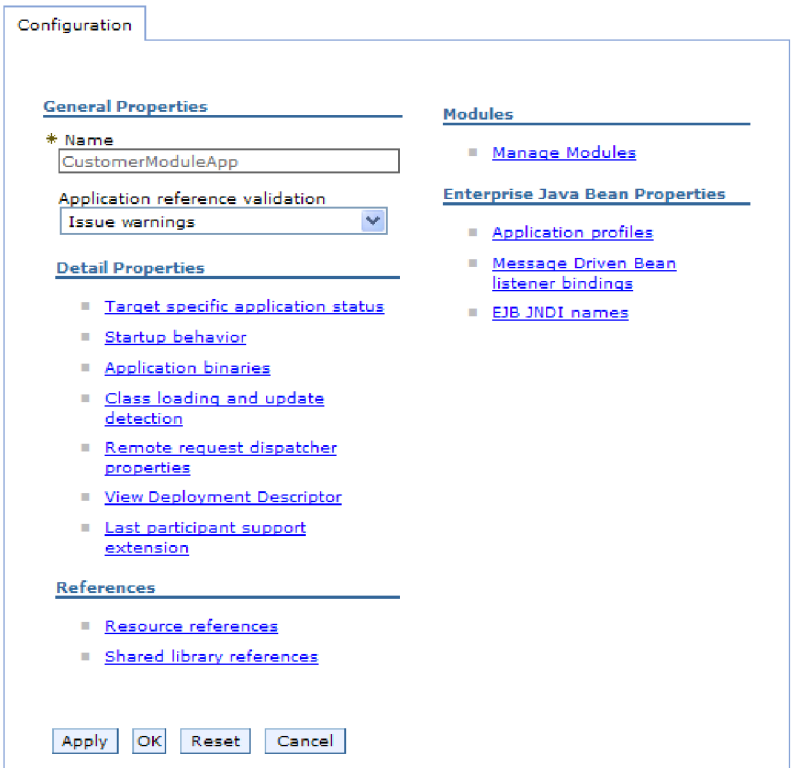


Figure 30. The Manage Modules selection in the Configuration tab

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

   Note: See "Resource adapter properties" on page 95 for more information about these properties.

   a. Click the name of the property.
   b. Change the contents of the **Value** field or type a value, if the field is empty. For example, if you click **logNumberOfFiles**, you see the following page:

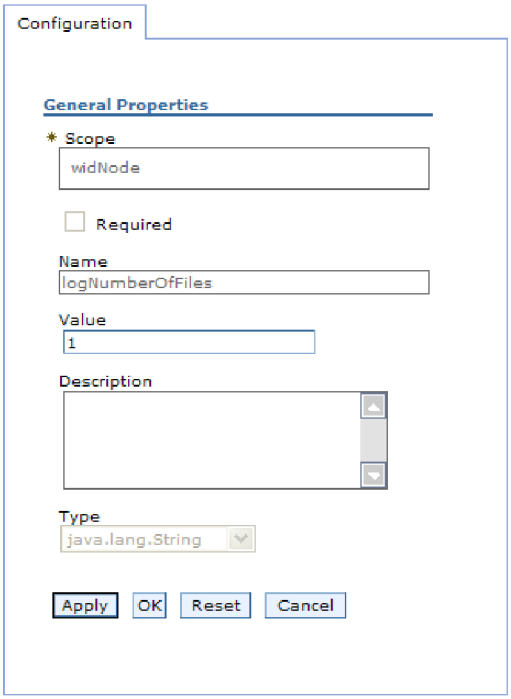


Figure 31. The Configuration tab for the logNumberOfFiles property

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

c. Click **OK**.

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

Results

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

Setting managed (J2C) connection factory properties for embedded adapters

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

Before you begin

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

About this task

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

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

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

Procedure

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

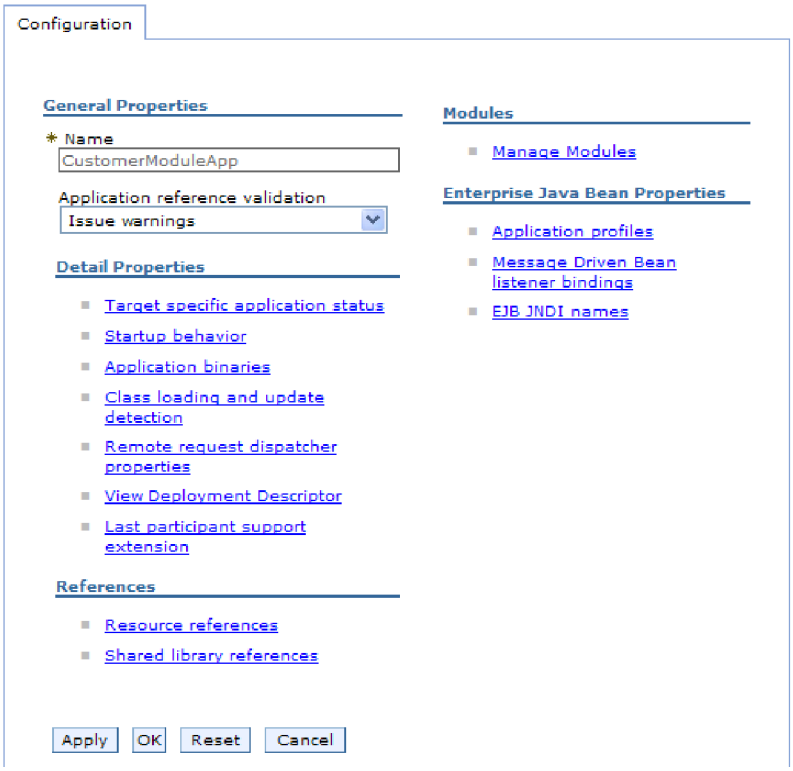


Figure 32. The Manage Modules selection in the Configuration tab

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

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

    Note: See "Managed connection factory properties" on page 98 for more information about these properties.

    a. Click the name of the property.

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

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

Results

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

Changing configuration properties for stand-alone adapters

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

Setting resource adapter properties for stand-alone adapters

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

Before you begin

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

About this task

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

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

Procedure

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

   Note: See "Resource adapter properties" on page 95 for more information about these properties.

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

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

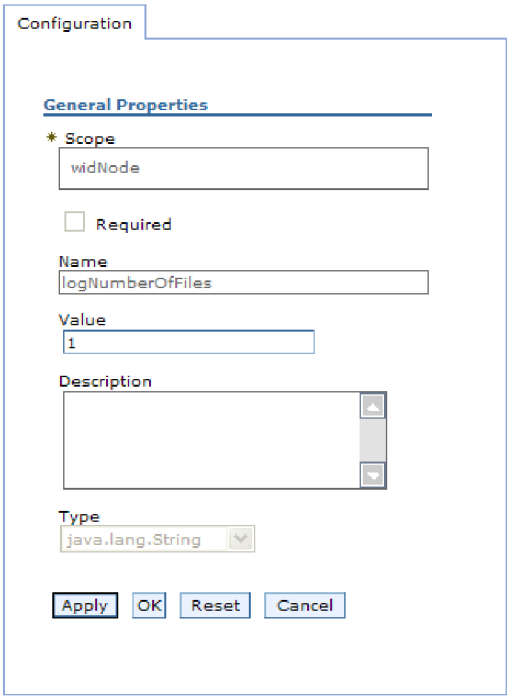


Figure 33. The Configuration tab for the logNumberOfFiles property

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

c. Click **OK**.

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

Results

The resource adapter properties associated with your adapter are changed.

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

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

Before you begin

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

About this task

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

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

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

Procedure

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. From the Resource adapters page, click **IBM WebSphere Adapter for JD Edwards EnterpriseOne**.
4. From the **Additional Properties** list, click **J2C connection factories**.
5. If you are going to use an existing connection factory, skip ahead to step 6.

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

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

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

      The newly created connection factory is displayed.

Figure 34. The list of connection factories

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

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

   Note: See “Managed connection factory properties” on page 98 for more information about these properties.

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

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

Results

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

Starting the application that uses the adapter

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

About this task

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

Procedure

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

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

Results

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

Stopping the application that uses the adapter

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

About this task

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

Procedure

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

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

Results

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

Monitoring performance using Performance Monitoring Infrastructure

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

About this task

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

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

Configuring Performance Monitoring Infrastructure

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

Before you begin

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

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

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

   For instructions on setting the trace level, refer to "Enabling tracing with the Common Event Infrastructure (CEI)" on page 76.
2. Generate at least one outbound request to produce performance data that you can configure.

Procedure

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

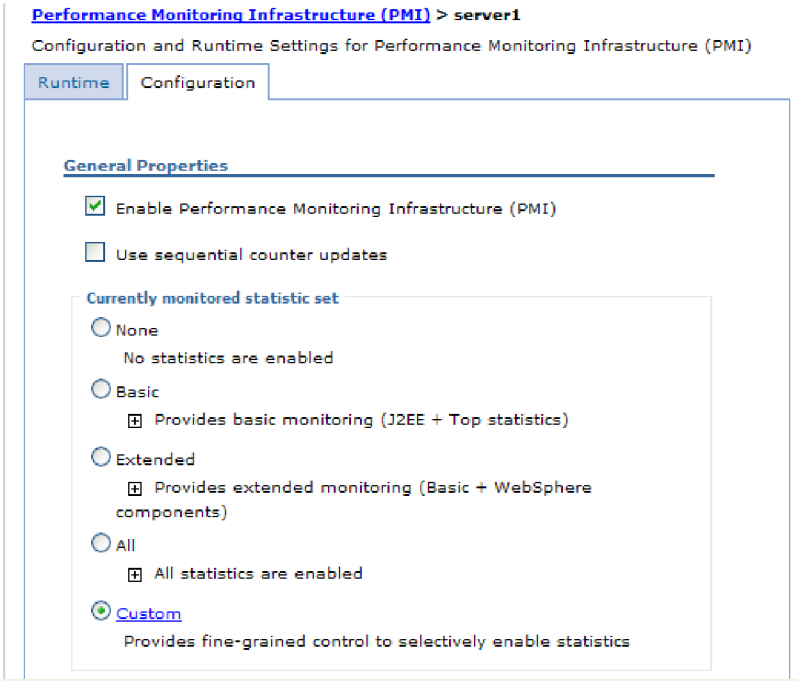


Figure 35. Enabling Performance Monitoring Infrastructure

e. Click **Apply** or **OK**.

f. Click **Save**. PMI is now enabled.

2. Configure PMI for your adapter.

   a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.

   b. From the list of servers, click the name of your server.

   c. Select **Custom**.

   d. Select the **Runtime** tab. The following figure shows the Runtime tab.

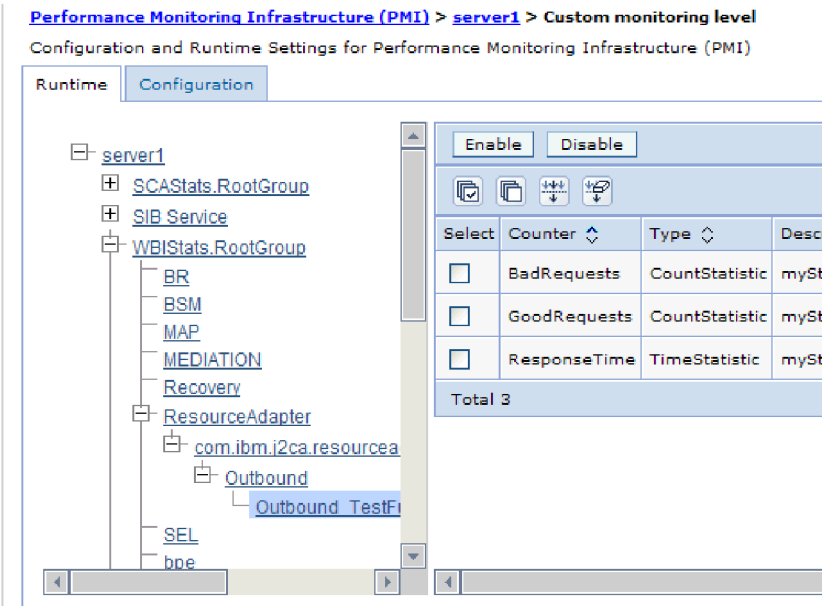


Figure 36. Runtime tab used for configuring PMI

e. Click **WBIStats.RootGroup**. This is a PMI submodule for data collected in the root group. This example uses the name WBIStats for the root group.

f. Click **ResourceAdapter**. This is a submodule for the data collected for the JCA adapters.

g. Click the name of your adapter, and select the processes you want to monitor.

h. In the right pane, select the check boxes for the statistics you want to gather, and then click **Enable**.

Results

PMI is configured for your adapter.

What to do next

Now you can view the performance statistics for your adapter.

Viewing performance statistics

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

Before you begin

Configure Performance Monitoring Infrastructure for your adapter.

Procedure

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

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

Results

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

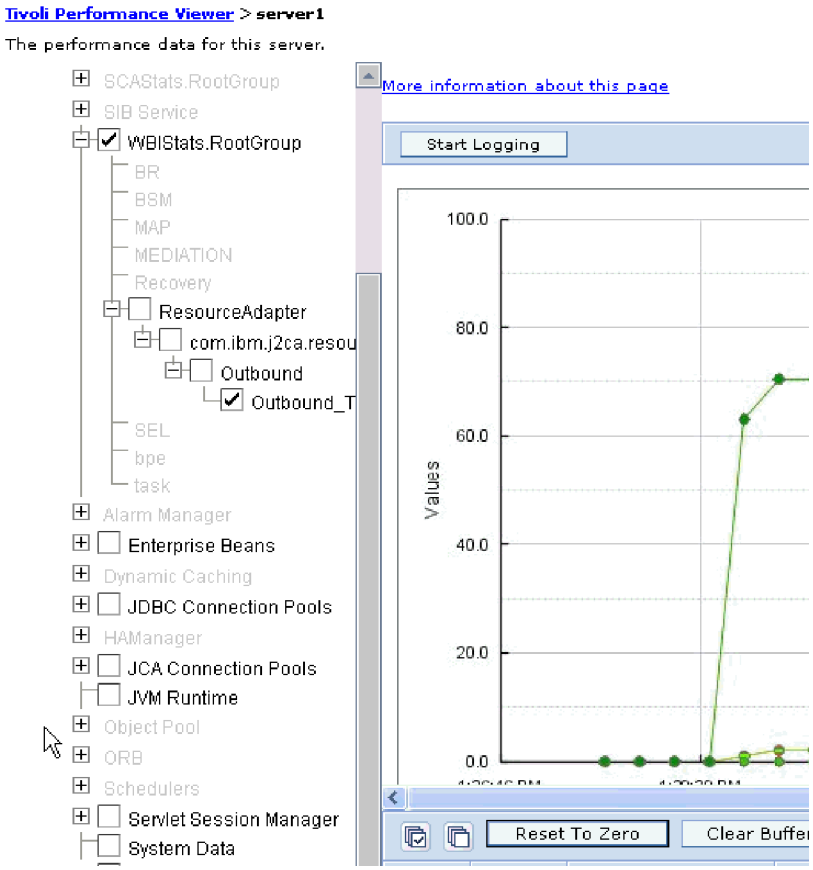


Figure 37. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure (CEI)

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

Procedure

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. In the list of servers, click the name of your server.

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

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

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

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

Results

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

Troubleshooting and support

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

Configuring logging and tracing

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

Configuring logging properties

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

About this task

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

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

  If you set the output for log analyzer format, you can open trace output using the Log Analyzer tool, which is an application included with your 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.

Procedure

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

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

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.

Results

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

Changing the log and trace file names

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

Before you begin

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

About this task

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

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

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

Procedure

1. In the navigation pane of the administrative console, select **Applications > Enterprise Applications**.
2. In the Enterprise Applications list, click the name of the adapter application. This is the name of the EAR file for the adapter, but without the .ear file extension. For example, if the EAR file is named Accounting_OutboundApp.ear, then click **Accounting_OutboundApp**.
3. In the Configuration tab, in the Modules list, click **Manage Modules**.
4. In the list of modules, click IBM WebSphere Adapter for JD Edwards EnterpriseOne.
5. In the Configuration tab, under Additional Properties, click **Resource Adapter**.
6. In the Configuration tab, under Additional Properties, click **Custom properties**.
7. In the Custom Properties table, change the file names.

a. Click either **logFilename** to change the name of the log file or **traceFilename** to change the name of the trace file.
b. In the Configuration tab, type the new name in the **Value** field. By default, the log file is called SystemOut.log and the trace file is called trace.log.
c. Click **Apply** or **OK**. Your changes are saved on your local machine.
d. To save your changes to the master configuration on the server, use one of the following procedures:
   - **Static change**: Stop and restart the server. This method allows you to make changes, but those changes do not take effect until you stop and start the server.
   - **Dynamic change**: Click the **Save** link in the Messages box above the Custom properties table. Click **Save** again when prompted. This method allows you to make changes that take effect right away.

First-failure data capture (FFDC) support

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

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

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

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

Business faults

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

Although WebSphere Process Server and WebSphere Enterprise Service Bus support other types of faults, the adapter generates only business faults, which are called simply *faults* in this documentation. Not all exceptions become faults. Faults are generated for errors that are actionable, that is, errors that can have a recovery action that does not require the termination of the application. For example, the adapter generates a fault when it receives a business object for outbound processing that does not contain the required data or when the adapter encounters certain errors during outbound processing.

Fault business objects

The external service wizard creates a business object for each fault that the adapter can generate. In addition, the wizard creates a WBIFault superset business object, which has information common to all faults, such as the message, errorCode, and primarySetKey attributes as shown in Figure 38 on page 81.

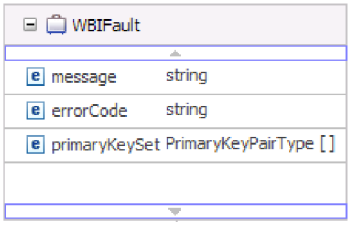


Figure 38. The structure of the WBIFault business object

Some faults contain the matchCount attribute, to provide additional information about the error. For others, WBIFault contains all the information needed to handle the fault.

The wizard creates the following fault business objects:

- InvalidRequestFault

  The adapter throws this fault when it detects an error in a property value in a business object before sending the business object to the JD Edwards EnterpriseOne server. For example, this fault is thrown when a date value string does not match the supported date format (yyyy-MM-dd for JDEDate property types and yyyy-MM-dd'T'hh:mm for JDEUTime property types).
- MatchesExceededLimitFault

  When processing the processing of an RetrieveAll operation, the adapter throws this fault if the number of records returned from the database query exceed the maximum number of records property in the interaction specification.

  To increase the number of records that can be returned, increase the value of the MaxRecords property in the interaction specification properties for the RetrieveAll operation.

  The business object for this fault has one property, matchCount, which is a string that contains the number of matches.
- MissingDataFault

  If the business object that is passed to the outbound operation does not have all the required attributes, then the adapter throws this fault.

  For example, the adapter throws this exception in the following situations:

  – If the requiredType application-specific information is `Yes` and ioType is not `Default` and the property is not set

  – If the reference application-specific information is set and requiredType application-specific information is `Yes` and the referenced property is not set
- RecordNotFoundFault

  When processing a data retrieval operation, the adapter throws this fault if the record is not found in the database for the keys specified. This fault can occur for the Delete, Update, Retrieve, and RetrieveAll operations.

Configuring the module for fault processing

Before you can configure your module to support business faults, you must have used the external service wizard to configure your module.

To enable fault processing, you must modify the .import and WSDL files for your module. You can configure faults at either the binding level or the method level. If the changes are made at binding level, they apply to all methods in the import. If the changes are made at the method binding level, you can configure a different fault for each method.

Table 4 lists the fault name and fault binding for each fault. Use the fault name and fault binding class when you configure the module.

Table 4. The fault name and fault binding class for each fault

Fault name	Associated fault binding class
INVALID_REQUEST	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl
MATCHES_EXCEEDED_LIMIT	com.ibm.j2ca.extension.emd.runtime.MatchingFaultDataBinding
MISSING_DATA	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl
RECORD_NOT_FOUND	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl

1. Edit the .import file to configure the fault at either the binding or the method level.
   - To configure the faults at the binding level:
     a. In the binding section, add the faultSelector attribute and the name of the fault selector. The name of the fault selector is com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl.
     b. For each fault that you want to enable, add a <faultBinding> element. In the element, specify the fault name and the fault data binding class name from Table 4.

        The following .import file shows the INVALID_REQUEST fault configured for all methods. **Bold face type** indicates changes made to enable fault handling.

```
<esbBinding xsi:type="eis:EISImportBinding"
 dataBindingType="com.ibm.j2ca.jde.emd.runtime.JDEDataBindingGenerator"
 faultSelector="com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl">
  <resourceAdapter
     name="BSFNSampleApp.IBM WebSphere Adapter for JD Edwards EnterpriseOne"
     type="com.ibm.j2ca.jde.JDEResourceAdapter" version="6.1">
       <properties/>
  </resourceAdapter>
  <faultBinding fault="INVALID_REQUEST"
    faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
```

   - To configure the faults at the method level:
     a. In method binding section for the method you want to associate with the fault, add the name of the fault selector. The value for fault selector is com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl.
     b. Add the fault binding elements in the method binding section. Use the fault name and the corresponding fault data binding class name from Table 4.

        The following .import file shows the INVALID_REQUEST fault configured for the retrieveGetEffectiveAddressContainer method. **Bold face type** indicates changes made to enable fault handling.

```
<methodBinding
 inDataBindingType="com.ibm.xmlns.prod.websphere.j2ca.jde.geteffectiveaddresscontainer
bg.GetEffectiveAddressContainerBGDataBinding"
 method="retrieveGetEffectiveAddressContainer"
 outDataBindingType="com.ibm.xmlns.prod.websphere.j2ca.jde.geteffectiveaddresscontaine
rbg.GetEffectiveAddressContainerBGDataBinding"
 faultSelector="com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl">
      <interaction>
        <properties>
          <functionName>Retrieve</functionName>
        </properties>
      </interaction>
      <faultBinding fault="INVALID_REQUEST"
        faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
</methodBinding>
```

2. Determine the target namespaces for your faults. For each fault that you want to enable, determine the namespace as follows:

a. Open the fault schema (XSD file) in a text editor.
b. Locate the target namespace. The target namespace is shown in **bold face type** in the following portion of a fault schema:

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://com/ibm/j2ca/fault/afcfault"
   xmlns:basefault="http://com/ibm/j2ca/fault">
<import namespace="http://com/ibm/j2ca/fault" schemaLocation="WBIFault.xsd"/>

. . .
```

The faults can all have the same target namespace or they can have different target namespaces.

3. Edit the WSDL file to declare the faults for the service. A sample WSDL file with these changes highlighted is shown at the end of the list.
   a. In the <definitions> element, add a namespace for each fault namespace, using the information you obtained from the fault schema files. If all your fault schemas have the same targetNamespace, add only one alias. If they have different targetNamespaces, add an alias for each unique namespace.
   b. Create an <xsd:import> element to import the schema for each fault you want to enable.
   c. Declare import statements for each fault type. Make sure that you are using the correct alias defined in step 3a to resolve the complex type in type=*alias*:*faultBOName*.xsd.
   d. Declare the message tags for each of the fault types.
   e. Add the fault declaration to each method where faults should be handled.

   The following WSDL file defines the INVALID_REQUEST fault. **Bold face type** indicates changes made to enable fault handling.

```
<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:GetEffectiveAddressContainerBG="http://www.ibm.com/xmlns/prod/websphere/j2ca/jde/g
eteffectiveaddresscontainerbg"
  xmlns:intf="http://BSFNSample/JDEOutboundInterface"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

Step 3a

```
  xmlns:fault="http://com/ibm/j2ca/fault/afcfault"
     name="JDEOutboundInterface.wsdl"
     targetNamespace="http://BSFNSample/JDEOutboundInterface">
  <types>
    <xsd:schema xmlns:tns="http://BSFNSample/JDEOutboundInterface"
       xmlns:xsd1="http://www.ibm.com/xmlns/prod/websphere/j2ca/jde/geteffectiveaddr
esscontainerbg"
       elementFormDefault="qualified"
       targetNamespace="http://BSFNSample/JDEOutboundInterface"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     <xsd:import
       namespace="http://www.ibm.com/xmlns/prod/websphere/j2ca/jde/geteffectiveaddresscont
ainerbg"
       schemaLocation="GetEffectiveAddressContainerBG.xsd"/>
```

Step 3b

```
  <xsd:import
     namespace="http://com/ibm/j2ca/fault/afcfault"
     schemaLocation="InvalidRequestFault.xsd"/>

. . .
```

Step 3c on page 83

```
<xsd:element name="invalidRequestFaultX">
    <xsd:complexType>
     <xsd:sequence>
     <xsd:element name="invalidRequestFaultElement"
                  type="fault:InvalidRequestFoundFault"/>
      </xsd:sequence>
    </xsd:complexType>
 </xsd:element>
  </xsd:schema>
</types>

. . .
```

Step 3d on page 83

```
<message name="invalidRequestFault">
    <part element="intf:invalidRequestFaultX"
          name="invalidRequestFaultPart"/>
</message>
<operation name=" retrieveGetEffectiveAddressContainer">
<input message="intf:retrieveGetEffectiveAddressContainerRequest"
       name="retrieveGetEffectiveAddressContainerRequest"/>
<output message="intf:retrieveGetEffectiveAddressContainerResponse"
        name="retrieveGetEffectiveAddressContainerResponse"/>
```

Step 3e on page 83

```
<fault message="intf:invalidRequestFault"
       name="invalidRequestFaultFault" />
</operation>
</portType>
</definitions>
```

XAResourceNotAvailableException

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

Symptom:

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

com.ibm.ws.Transaction.XAResourceNotAvailableException

Problem:

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

Solution:

To correct this problem, use the following procedure:

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

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

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

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

3. Start the process server.

Self-help resources

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

Support Web site

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

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

Recommended fixes

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

Technotes

Technotes provide the most current documentation about the Adapter for JD Edwards EnterpriseOne, including the following topics:

- Problems and their currently available solutions
- Answers to frequently asked questions
- How-to information about installing, configuring, using, and troubleshooting the adapter
- *IBM Software Support Handbook*

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

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

Plug-in for IBM Support Assistant

Adapter for JD Edwards EnterpriseOne provides a plug-in for IBM Support Assistant, which is a free, local software serviceability workbench. For information about installing or using IBM Support Assistant, visit this address:

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

Chapter 8. Reference information

To support you in your tasks, reference information includes details about business objects that are generated by the external service wizard and information about adapter properties, including those that support bidirectional transformation. It also includes pointers to adapter messages and related product information.

Business object information

You can determine the purpose of a business object by examining both the application-specific information within the business object definition file and the name of the business object. The application-specific information dictates what operations can be performed on the JD Edwards EnterpriseOne server. The name typically reflects the operation to be performed and the structure of the business object.

Application-specific information

Application-specific information (ASI) is metadata that specifies adapter-dependent information about how to process business objects for the adapter for JD Edwards EnterpriseOne.

When the external service wizard generates a business object, it automatically generates a business object definition, which is saved as an XSD (XML Schema Definition) file. The business object definition contains the application-specific information for that business object. If you want to change the generated ASI, you can modify the metadata values either from the Properties tab in the Business Integration perspective of WebSphere Integration Developer or by using the business object editor.

The adapter for JD Edwards EnterpriseOne uses application-specific information (ASI) to create queries for Create, Retrieve, Update, and Delete operations. ASI is generated by the external service wizard at three levels: the business-object level, the property level, and the operation level.

Application-specific information at the business-object-level

Application-specific information (ASI) at the business-object level is typically used to specify the name of the corresponding database table and to provide information necessary to perform a physical or logical delete operation. The following table describes the ASI at the business-object level.

Table 5. Application-specific information at the business-object level

Application-specific information	Description
Name	Name of operation
BSFN	List of business functions associated with the operation

Application-specific information at the property level

Application-specific information (ASI) at the property level is typically used to specify the metadata for a property. ASI at the property level represents either

child objects or an array of child objects. The following table describes the ASI of a complex property (a child) or a structure or table property (an array of child objects).

Table 6. Application-specific information at the property level

Application-specific information	Description	Possible values
Name	The business function parameter name as represented in JD Edwards EnterpriseOne	BSFNName
Type	The type of the business function parameter as it exists in JD Edwards EnterpriseOne	BSFN
IOType	The type of the business function parameter as it exists in JD Edwards EnterpriseOne	• `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	• `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	None

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	`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`.

Application-specific information at the operation level

Application-specific information (ASI) at the operation level is used by the adapter to perform operations, such as to retrieve or update information in the JD Edwards EnterpriseOne server. The following table describes the ASI at the operation level.

Table 7. Application-specific information at the operation level

Application-specific information	Description	Value
Name	The name of the business object operation	• Create • Retrieve • Update • Delete • RetrieveAll
BSFN.Name	The name of the business functions to process	• Name • RollbackOnWarnings
BSFN.RollbackOnWarnings	Indicates if the adapter needs to rollback the current transaction when the business function returns with warnings	False (default setting)

Supported operations

An operation is the action that an adapter can perform on the JD Edwards EnterpriseOne server during outbound processing. The name of the operation typically indicates the type of action that the adapter takes, such as *create* or *update*.

The following tables defines the operations that the adapter for JD Edwards EnterpriseOne supports during outbound processing for business functions and XML Lists.

Table 8. Supported operations of business functions

Operation	Definition
Create	The top-level business object and all contained children are created.
Update	The top-level business object is modified. This operation can include adding and deleting child objects.
Delete	The top-level business object and any contained children are deleted.
Retrieve	The top-level business object and any contained children are retrieved.

Table 9. Supported operations of XML Lists

Operation	Definition
RetrieveAll	Retrieves all records from the JD Edwards EnterpriseOne server that correspond to the query values specified in the XML List. Returns a result set in the form of a container of JD Edwards EnterpriseOne query business objects, which represent the data for each row retrieved from the table.

Naming conventions

When the external service wizard generates a business object, it provides a name for the business object based on the name of the object in the JD Edwards EnterpriseOne server that it uses to build the business object.

When the external service wizard provides a name for the business object, it converts the name of the object to mixed case, which means that it removes any separators, such as spaces or underscores, and then capitalizes the first letter of each word. For example, if the external service wizard uses a JD Edwards EnterpriseOne server object called CUSTOMER_ADDRESS to generate a business object, it generates a business object called CustomerAddress.

The generated business object name can indicate the structure of the business object. However, business objects names have no semantic value to the adapter. This means that if you change the business object name, the behavior of the business object remains the same.

Important: If you choose to rename a business object, use the refactoring functionality in WebSphere Integration Developer to ensure that you update all of the business object dependencies. For instructions on using refactoring to rename business objects, refer to the following link: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wbit.help.refactor.doc/topics/trenameboatt.html.

Table 10. Naming conventions

Element	Naming convention	Example
Name of the business graph	The business graph that contains the parent business object is named for the contained business object, followed by the string ″BG.″	GetEffectiveAddressContainerBG
Name of the business object container	<*name_of_business_object*>Container	GetEffectiveAddressContainer
Name of the business function	Name of the business function discovered by the external service wizard	GetEffectiveAddress
Name of the XML List	Name of the XML List table discovered by the external service wizard	F0116

Note: Business graph generation is optional and is supported for WebSphere Process Server only.

Configuration properties

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

Guide to information about properties

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

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

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

Connection properties for the external service wizard

External service discovery connection properties establish a connection between the external service wizard of WebSphere Integration Developer, a tool that is used to create business objects, and the JD Edwards EnterpriseOne server. The properties you configure in the external service wizard specify such things as connection configuration, bidi properties, and logging and tracing options.

Once a connection between the external service wizard and the JD Edwards EnterpriseOne server is established, the external service wizard is able to access the metadata it needs from the JD Edwards EnterpriseOne server to create business objects.

Note: Some of the properties that you set in the external service wizard are used as the initial value for the resource adapter and managed connection factory properties that you can specify at a later time in the wizard.

The external service connection properties and their purposes are described in the following table. A complete description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 90.

Important: If you set any of these connection properties using bidirectional script, you must set values that identify the format of the bidirectional script entered for that property.

Table 11. External service connection properties for Adapter for JD Edwards EnterpriseOne

Property name	Description
"Environment"	Specifies the JD Edwards EnterpriseOne environment name
"Log file output location" on page 93	Specifies the location of the log file for external service
"Logging level" on page 93	Specifies the type error for which logging will occur during external service
"Password" on page 94	Password of the adapter user account on the JD Edwards EnterpriseOne environment
"Role" on page 94	Name of the role that is associated with the user name used to access the JD Edwards EnterpriseOne environment.
"User name" on page 95	Name of the adapter user account on the JD Edwards EnterpriseOne environment

Environment

This property specifies the JD Edwards EnterpriseOne environment name.

Table 12. Environment details

Required	Yes
Default	No default value
Property type	String
Usage	A JD Edwards EnterpriseOne environment is a user-defined pointer that indicates the location of data and objects on a JD Edwards EnterpriseOne server. Users can be authorized to use multiple JD Edwards EnterpriseOne environments on a single JD Edwards EnterpriseOne server.
Globalized	Yes

Table 12. Environment details (continued)

Bidi supported	Yes

Log file output location

This property specifies the location of the log file for external service.

Table 13. Log file output location details

Required	Yes
Default	The .metadata directory of the workspace
Property type	String
Usage	Use this directory to hold the log file that will list the errors that occur during the discovery process. The type of discovery errors for which logging occurs is controlled by the Logging level property.
Example	C:\IBM\wid6.1.0\workspace\.metadata\JDEMetadataDiscovery.log
Globalized	Yes
Bidi supported	No

Logging level

This property specifies the type error for which logging will occur during external service.

Table 14. Logging level details

Required	No

Possible values	`ALL`
	`OFF`
	`FINE`
	`FINER`
	`FINEST`
	`CONFIG`
	`INFO`
	`SEVERE`
	`WARNING`

Default	`SEVERE`
Property type	String
Usage	Use this property to tailor tracing capabilities. By specifying an error type, you are indicating that trace operations will occur only for errors of the type specified.

Table 14. Logging level details (continued)

Example	Accepting the default value of SEVERE will provide trace information on errors that fall into the SEVERE category. Severe errors mean that an operation cannot continue, though the adapter can still function. Severe errors also include error conditions that indicate an impending fatal error, i.e., reporting on situations that strongly suggest that resources are on the verge of being depleted. Other error descriptions are as follows: • Fatal – Adapter cannot continue. Adapter cannot function • Warning – Potential error or impending error. This also includes conditions that indicate a progressive failure, for example, the potential leaking of resources. • Audit – Significant event affecting adapter state or resources • Info – General information outlining overall operation progress • Config – Configuration change or status • Detail – General information detailing operation progress
Globalized	Yes
Bidi supported	No

Password

This property specifies the password of the adapter user account on the JD Edwards EnterpriseOne environment.

Table 15. Password details

Required	Yes
Default	No default value
Property type	String
Usage	Passwords are created and named by the JD Edwards EnterpriseOne administrator. There are no restrictions on the type of characters used, the number of characters used, or the case of the characters used in passwords.
Globalized	No
Bidi supported	Yes

Role

This property specifies the name of the role that is associated with the user name used to access the JD Edwards EnterpriseOne environment.

Table 16. Role details

Required	Yes
Default	No default value
Property type	String
Usage	Roles define what authority users have. Users can have multiple roles. A user's access to applications, forms, table columns, data sources, and so on, is based on one or more roles to which the user is assigned. Roles are created and named by the JD Edwards EnterpriseOne administrator.
Examples	• System administrator • Human resources • Accounting

Table 16. Role details (continued)

Globalized	No
Bidi supported	Yes

User name

This property specifies the name of the adapter user account on the JD Edwards EnterpriseOne environment.

Table 17. User name details

Required	Yes
Default	No default value
Property type	String
Usage	User names are created by the JD Edwards EnterpriseOne administrator. There are no restrictions on the type of characters used, the number of characters used, or the case of the characters used in user names.
Globalized	Yes
Bidi supported	Yes

Resource adapter properties

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

The following properties for logging and tracing are no longer required in version 6.1.0, but are supported for compatibility with previous versions.

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

The following property that was specified as a resource adapter property in version 6.0.2 applies to the managed connection factory property group in version 6.1.0.

- Timeout

The following table lists and describes the resource adapter properties. A more detailed description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 90.

Table 18. Resource adapter properties for Adapter for JD Edwards EnterpriseOne

Name		
In the wizard	**In the administrative console**	**Description**
Adapter ID	AdapterID	Identifies the adapter instance for CEI and PMI events with respect to logging and tracing.
(Not available)	enableHASupport	Do not change this property.
(Not available)	LogFileMaxSize	Supported for compatibility with earlier versions
(Not available)	LogFilename	Supported for compatibility with earlier versions
(Not available)	LogNumberOfFiles	Supported for compatibility with earlier versions
(Not available)	TraceFileMaxSize	Supported for compatibility with earlier versions
(Not available)	TraceFileName	Supported for compatibility with earlier versions
(Not available)	TraceNumberOfFiles	Supported for compatibility with earlier versions

Adapter ID to use for logging and tracing (AdapterID)

Use this property to identify a specific deployment, or instance, of the adapter.

Table 19. Adapter ID to use for logging and tracing details

Required	Yes
Default	CWYED_JDE
Property type	String
Usage	This property is used to identify the adapter instance for PMI events. If you are deploying multiple instances of an adapter, set this property to a unique value for each adapter instance. For inbound processing this property is retrieved from the resource adapter properties. For outbound processing, it is retrieved form the managed connection factory properties.
Globalized	Yes
Bidi supported	No

Enable high availability support (enableHASupport)

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

Log file maximum size (LogFileMaxSize)

This property specifies the size of the log files in kilobytes.

Table 20. Log file maximum size details

Required	No
Default	`0`
Property type	Integer
Usage	When the log file reaches it maximum size, the adapter start using a new log file. If the file size is specified as `0` or no maximum size is specified, the file does not have a maximum size.
Globalized	Yes
Bidi supported	No

Log file name (LogFilename)

This property specifies the full path name of the log file.

Table 21. Log file name details

Required	No
Default	No default value
Property type	String
Usage	This property is deprecated.
Globalized	Yes
Bidi supported	Yes

Log number of files (LogNumberOfFiles)

This property specifies the number of log files.

Table 22. Log number of files details

Required	No
Default	1
Property type	Integer
Usage	When a log file reaches its maximum size, the adapter starts using another log file. If no value is specified, the adapter creates a single log file.
Globalized	Yes
Bidi supported	No

Trace file maximum size (TraceFileMaxSize)

This property specifies the size of the trace files in kilobytes.

Table 23. Trace file maximum size details

Required	No
Default	0
Property type	Integer
Usage	If no value is specified, then the trace file has no maximum size.
Globalized	Yes
Bidi supported	No

Trace file name (TraceFilename)

This property specifies the full path of the trace file.

Table 24. Trace file name details

Required	No
Default	No default value
Unit of measure	Kilobytes
Property type	String
Usage	This property is deprecated.

Table 24. Trace file name details (continued)

Globalized	Yes
Bidi supported	Yes

Trace number of files (TraceNumberOfFiles)

This property specifies the number of trace files to use. When a trace file reaches its maximum size, the adapter starts using another trace file.

Table 25. Trace number of files details

Required	No
Default	1
Property type	Integer
Usage	If no value is specified, the adapter uses a single trace file.
Globalized	Yes
Bidi supported	No

Managed connection factory properties

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the JD Edwards EnterpriseOne server.

You set the managed connection factory properties using either the external service wizard or the administrative console (after deployment).

The following table lists and describes the managed connection factory properties. A more detailed description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 90.

Note: The external service wizard refers to these properties as managed connection factory properties and WebSphere Process Server or WebSphere Enterprise Service Bus administrative console refers to these as (J2C) connection factory properties.

Table 26. Managed connection factory properties for Adapter for JD Edwards EnterpriseOne

Property name		
In the wizard	**In the administrative console**	**Description**
"Environment" on page 99	environment	Specifies the JD Edwards EnterpriseOne environment name
"Password" on page 99	password	Password of the adapter user account on the JD Edwards EnterpriseOne environment
"Role" on page 99	role	Name of the role that is associated with the user name used to access the JD Edwards EnterpriseOne environment.
"Timeout" on page 100	timeout	This property is the global timeout value, in milliseconds, set on the XML List request execute call.
"User name" on page 100	userName	Name of the adapter user account on the JD Edwards EnterpriseOne environment

Environment

This property specifies the JD Edwards EnterpriseOne environment name.

Table 27. Environment details

Required	Yes
Default	No default value
Property type	String
Usage	A JD Edwards EnterpriseOne environment is a user-defined pointer that indicates the location of data and objects on a JD Edwards EnterpriseOne server. Users can be authorized to use multiple JD Edwards EnterpriseOne environments on a single JD Edwards EnterpriseOne server.
Example	
Globalized	Yes
Bidi supported	Yes

Password

This property specifies the password of the adapter user account on the JD Edwards EnterpriseOne environment.

Table 28. Password details

Required	Yes
Default	No default value
Property type	String
Usage	Passwords are created and named by the JD Edwards EnterpriseOne administrator. There are no restrictions on the type of characters used, the number of characters used, or the case of the characters used in passwords.
Example	
Globalized	No
Bidi supported	Yes

Role

This property specifies the name of the role that is associated with the user name used to access the JD Edwards EnterpriseOne environment.

Table 29. Role details

Required	Yes
Default	No default value
Property type	String
Usage	Roles define what authority users have. Users can have multiple roles. A user's access to applications, forms, table columns, data sources, and so on, is based on one or more roles to which the user is assigned. Roles are created and named by the JD Edwards EnterpriseOne administrator.
Examples	• System administrator • Human resources • Accounting

Table 29. Role details (continued)

Globalized	No
Bidi supported	Yes

Timeout

This property specifies the timeout value, in milliseconds, set on the XML List request call.

Table 30. Timeout details

Required	Yes
Default	30,000
Unit of measure	Milliseconds
Property type	Integer
Usage	Use the Timeout property to specify the amount of time the adapter should take to perform a RetrieveAll operation using an XML List. If no value is specified, the adapter will time out after 30 seconds (30,000 milliseconds).
Globalized	Yes
Bidi supported	No

User name

This property specifies the name of the adapter user account on the JD Edwards EnterpriseOne environment.

Table 31. User name details

Required	Yes
Default	No default value
Property type	String
Usage	User names are created by the JD Edwards EnterpriseOne administrator. There are no restrictions on the type of characters used, the number of characters used, or the case of the characters used in user names.
Example	
Globalized	Yes
Bidi supported	Yes

Interaction specification properties

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, some properties for outbound operations can be changed by the user.

One reason to change the interaction specification properties is to increase the value of the property that specifies the maximum number of records to be returned by a RetrieveAll operation, if your RetrieveAll operations do not return complete information. To change these properties after the application is deployed, use the assembly editor in WebSphere Integration Developer. The properties reside in the method binding of the import.

The following tables list and describe the interaction specification property that you can set. For information about how to read the property details table, see "Guide to information about properties" on page 90.

Table 32. Interaction specification property for the Adapter for JD Edwards EnterpriseOne

Property name		
In the wizard	**In the assembly editor**	**Description**
Maximum records for RetrieveAll operation	Maximum number of records	Maximum number of records to return during a RetrieveAll operation

Maximum number of records for RetrieveAll operation

This property specifies the maximum number of records to return for a RetrieveAll operation.

Table 33. Maximum records for RetrieveAll operation details

Required	Yes
Default	100
Usage	If the number of hits in the database exceeds the value of the Maximum number of records property, the adapter returns the error MatchesExceededLimitException and MatchesExceededLimitFault. Use this property to avoid out-of-memory issues.
Property type	Integer
Globalized	No
Bidi supported	No

Globalization

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

Globalization and bidirectional data 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 script data transformation, which refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Globalization

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, WebSphere Integration Developer, WebSphere Process Server, and WebSphere Enterprise Service Bus are written in Java. The Java runtime environment within the Java virtual machine (JVM) represents data in the Unicode character code set. Unicode contains encodings for characters in most known character code sets (both single- and multi-byte). Therefore, when data is transferred between these integration system components, there is no need for character conversion.

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

Bidirectional script data 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 data, standards are used to display and process it. Bidirectional script data transformation applies only to string type data. WebSphere Process Server orWebSphere Enterprise Service Bus uses the Windows standard format, but applications or file systems that exchange data with the server might use a different format. The adapter transforms bidirectional script data passed between the two systems so that it is accurately processed and displayed on both sides of a transaction. It transforms the script data by using a set of properties that defines the format of script data, as well as properties that identify content or metadata to which transformation applies.

Bidirectional script data formats

WebSphere Process Server or WebSphere Enterprise Service Bus uses the bidirectional format of ILYNN (implicit, left-to-right, on, off, nominal). These five attributes comprise the format used by Windows. If an application or file system that sends or receives data from the server uses a different format, the adapter converts the format prior to introducing the data to the server. For the conversion to occur, you use the external service wizard to set attribute values that represent the bidirectional format used by the sending application or file system. This is done when you deploy the adapter for the first time.

Bidirectional data format attributes and values are listed in the following table.

Table 34. Bidirectional data format attributes and values

Letter position	Purpose	Values	Description	Default setting
1	Order schema	I or V	Implicit (Logical) or Visual	I
2	Direction	L R C D	Left-to-Right, Right-to-Left Contextual Left-to-Right Contextual Right-to-Left	L
3	Symmetric Swapping	Y or N	Symmetric Swapping is on or off	Y
4	Shaping	S N I M F B	Shaped text Unshaped text Initial shaping Middle shaping Final shaping Isolated shaping	N
5	Numeric Shaping	H C N	Hindi Contextual Nominal	N

Bidirectional properties that identify data for transformation

To identify business data subject to transformation, set the BiDiContextEIS property. Do this by specifying values for each of the five bidirectional format attributes (listed in Table 1) for the property. The BiDiContextEIS property can be set for the managed connection factory and the activation specification.

To identify application-specific data for transformation, annotate the BiDiContextEIS property and the BiDiMetadata property within a business object. Do this by using the business object editor within WebSphere Integration Developer to add the properties as application-specific elements of a business object.

Properties enabled for bidirectional data transformation

Bidirectional data transformation properties enforce the correct format of bidirectional script data exchanged between an application and integration tools and runtime environments. Once these properties are set, bidirectional script data is correctly processed and displayed in WebSphere Integration Developer and WebSphere Process Server or WebSphere Enterprise Service Bus.

Managed connection properties

The following managed connection properties control bidirectional script data transformation.

- Username
- Password
- Environment
- Role

Adapter messages

View the messages issued by WebSphere Adapter for JD Edwards EnterpriseOne at the following location.

Link to messages: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.610.help.messages.doc/messages.html

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

- Messages with the prefix CWYED are issued by WebSphere Adapter for JD Edwards EnterpriseOne
- Messages with the prefix CWYBS are issued by the adapter foundation classes, which are used by all the adapters.

Related information

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

Samples and tutorials

The WebSphere Integration Developer online samples/tutorials gallery includes samples and tutorials to help you use WebSphere Adapters. You can access the online samples/tutorials gallery as follows:

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

Information resources

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

Information about related products

- WebSphere Business Process Management, version 6.1.0, information center, which includes WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Integration Developer information: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp
- WebSphere Adapters, version 6.0.2, information center: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsadapters602.doc/welcome_top_wsa602.html
- WebSphere Adapters, version 6.0, information center: http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/topic/com.ibm.wsadapters.doc/welcome_wsa.html
- WebSphere Business Integration Adapters information center: http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbi_adapters.doc/welcome_adapters.htm

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

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

Notices

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

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

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

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

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

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

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

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

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

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

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

IBM Corporation
Department 2Z4A/SOM1
294 Route 100
Somers, NY 10589-0100
U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of

this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

Programming interface information

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

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

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

Warning:

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

Trademarks and service marks

IBM, the IBM logo, developerWorks, iSeries, Redbooks, Tivoli, ViaVoice, and WebSphere are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

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

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

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

Index

T

W

X

Printed in USA