





**WebSphere Adapter for JD Edwards EnterpriseOne User Guide
Version 6 Release 2**

Note

Before using this information and the product it supports, read the information in "Notices" on page 85.

December 2008

This edition applies to version 6, release 2, 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.

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

© **Copyright International Business Machines Corporation 2006, 2008.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Overview of WebSphere

Adapter for JD Edwards EnterpriseOne . 1

Hardware and software requirements	1
Technical overview of the Adapter for JD Edwards EnterpriseOne	2
Outbound processing	2
Inbound processing	3
Business objects	5
Java data bindings	6
The J2C Bean wizard	7
Standards Compliance	7

Chapter 2. Planning for adapter implementation 9

Before you begin	9
Support for protecting sensitive user data in log and trace files	9
Security	10
User authentication	10
Deployment options	11
WebSphere Adapters in clustered environments	12

Chapter 3. Configuring the module for deployment. 15

Launching the J2C Bean wizard	15
Adding external software dependencies	16
Editing external dependency files	18
Setting connection properties for the J2C Bean wizard	19
Configuring the module for outbound processing	20
Generating business functions using external service discovery	20
Generating the EJB or JSP project	25
Generating XML Lists using external service discovery	26
Setting deployment properties and generating artifacts.	30
Configuring the module for inbound processing	31
Selecting business objects and services	31
Configuring the selected objects	33
Setting deployment properties and generating artifacts.	33

Chapter 4. Configuring the application on WebSphere Application Server . . . 35

Changing configuration properties for embedded adapters	35
Setting resource adapter properties for embedded adapters	35
Setting managed (J2C) connection factory properties for embedded adapters	36

Setting activation specification properties for embedded adapters.	37
Changing configuration properties for stand-alone adapters	38
Setting resource adapter properties for stand-alone adapters	38
Setting managed (J2C) connection factory properties for stand-alone adapters	39
Setting activation specification properties for stand-alone adapters	40
Adding dependency libraries to the deployed resource adapter.	41

Chapter 5. Troubleshooting and support 43

Support for the Log and Trace Analyzer	43
Configuring logging and tracing	43
Configuring logging properties	44
Changing the log and trace file names	45
First-failure data capture (FFDC) support	46
XAResourceNotAvailableException	47
Self-help resources	47

Chapter 6. Reference information . . . 49

Business object information	49
Application-specific information	49
Supported operations	51
Naming conventions	51
Outbound configuration properties	52
Setting connection properties for the J2C Bean wizard	54
Resource adapter properties	57
Managed connection factory properties	61
Interaction specification properties.	64
Inbound configuration properties	65
Setting connection properties for the J2C Bean wizard	67
Resource adapter properties	70
Activation specification properties.	74
Globalization	81
Globalization and bidirectional data transformation	81
Properties enabled for bidirectional data transformation	83

Notices 85

Programming interface information	87
Trademarks and service marks	87

Index 89

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™ 2 Platform, Enterprise Edition (J2EE) 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 Rational® Application Developer for WebSphere Software and deploying the module that contains the adapter to WebSphere Application 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.

Hardware and software requirements

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

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

Additional information

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

- The compatibility matrix for WebSphere Business Integration Adapters and WebSphere Adapters identifies the supported versions of required software for your adapter. To view this document, go to the WebSphere Adapters support

page and click **Compatibility Matrix** beneath the **Related** heading in the **Additional support links** section: <http://www.ibm.com/software/integration/wbiadapters/support/>.

- Technotes for WebSphere Adapters provide workarounds and additional information that are not included in the product documentation. To view the technotes for your adapter, go to the following Web page, select the name of your adapter from the **Product category** list, and click the search icon: <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

Technical overview of 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 J2C Bean wizard, a tool launched from Rational Application Developer for WebSphere Software. The business objects generated by the J2C Bean 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 Rational Application Developer for WebSphere Software, to change the properties of the generated business object definition. You can also change or set configuration properties for business objects in the administration console of Rational Application Developer for WebSphere Software.

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.

WebSphere Adapter for JD Edwards EnterpriseOne business applications supports synchronous outbound processing. This implies that the component sends a request in the form of a WebSphere business object hierarchy to the adapter, the adapter processes the request and returns a WebSphere business object hierarchy that represents the result of the operation.

When the adapter receives a WebSphere business object hierarchy, the adapter processes it as follows:

1. The adapter extracts metadata from the WebSphere business object hierarchy.
2. It identifies the appropriate JD Edwards EnterpriseOne objects to access (for example, JD Edwards EnterpriseOne business objects and business components, or JD Edwards EnterpriseOne business function, or JD Edwards EnterpriseOne XML List) depending on the objects against which the artifacts were generated.
3. The adapter extracts the outbound operation to perform from the WebSphere business object hierarchy.

4. After accessing the required JD Edwards EnterpriseOne objects, the adapter retrieves the data for XML List, or performs the corresponding business function method.
5. If there are updates (Create, Update, Delete), the adapter populates that JD Edwards EnterpriseOne object (business function or XML List hierarchy) with data from the hierarchy of WebSphere business objects.

Supported outbound operations

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

Table 1.

Operations	Description
Create	Creates the business object.
Delete	Deletes the business object and its children. Because the adapter supports only logical deletes, objects are marked as deleted but not removed.
Exists	Checks for the existence of incoming business objects.
Retrieve	Retrieves the JD Edwards EnterpriseOne component and maps component data onto the business object.
RetrieveAll	Retrieves multiple instances of the JD Edwards EnterpriseOne component and maps component data onto the business object.
Update	Updates the corresponding JD Edwards EnterpriseOne component with the incoming business object.

Inbound processing

WebSphere Adapter for JD Edwards EnterpriseOne supports asynchronous inbound processing. This means that the adapter polls the JD Edwards EnterpriseOne server at specified intervals for events. When the adapter receives an event, it converts the event data into a business object and sends the business object to the component.

WebSphere Adapter for JD Edwards EnterpriseOne supports real-time event. A real-time event is a business transaction that provides information from the JD Edwards EnterpriseOne server that can be used to interoperate with a third-party system. Real-time events can be generated wherever business functions run, such as HTML, WIN32, and enterprise servers. Real-time events are useful for producing notifications in real-time. The adapter supports both single and container real-time events.

When the adapter gets a real-time event from the JD Edwards EnterpriseOne transaction server by invoking JD Edwards EnterpriseOne Dynamic Java Connector API, it parses the content of this real-time event and converts it into a business object. Then, the adapter sends the business object to the event endpoints. For example, if a company is updated, the JD Edwards EnterpriseOne server captures this change immediately, and one real-time event is generated by JD Edwards EnterpriseOne transaction server. The adapter then communicates with the JD Edwards EnterpriseOne transaction server, retrieves this real-time event, and processes it. After converting it into a business object, the adapter delivers this business object to the event endpoint.

WebSphere Adapter for JD Edwards EnterpriseOne processes events as follows:

1. The adapter invokes JD Edwards EnterpriseOne Dynamic Java Connector API to get a real-time event.
2. The adapter parses the content of this real-time event.
3. The adapter populates the associated business object with the values it retrieves from the payload of this real-time event.
4. The adapter sends the generated business object to each registered application.

Note: Before inbound processing can occur, the JD Edwards EnterpriseOne server must be configured to support real-time events.

Event persistence

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

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

To prevent the loss of events in the case of server failure, the recommended approach is to use the database event store. To use the in-memory event persistence capability of the adapter, you must not set the `EP_DataSource_JNDIName` property.

When a failed event occurs and the file can not be written to disk the JDE adapter will print the content of this failed event to a JDE trace file, to avoid event loss. The content will be between two "#####", and the trace level is INFO. The content can be copied to into a new file and saved with the event id as the file name in the directory "FailedEventFolder". When the status of the failed event record is changed from -1 to 0 in the event table the event will be handled again.

Event store

The event store is a persistent cache where event records are saved until the polling adapter can process them. The adapter uses event stores to keep track of

inbound events as they make their way through the system. Each time a real-time event is received, the adapter updates the status of the event in an event store. The status of each event is continually updated by the adapter for recovery purposes until the events are delivered to the endpoint.

If the adapter detects that there is no event store for the inbound module in the WebSphere Application Server, it automatically creates one when the application is deployed to the runtime. Each event store created by the adapter is associated with a specific inbound module. The adapter does not support multiple adapter modules pointing to the same event store.

When the adapter polls the JD Edwards EnterpriseOne transaction server and receives a real-time event, it creates an entry in the event store for each event that matches the search criteria specified in the activation specification properties. The adapter records the status of each new entry as NEW.

If a real-time event is successfully delivered, the corresponding event store entries are deleted. For failed events, the entries remain in the event store.

Assured-once delivery

The JD Edwards EnterpriseOne transaction server provides guaranteed event delivery quality of service. This means that all of the real-time events the adapter subscribes to are delivered to the adapter without any loss. Because it is possible for the JD Edwards EnterpriseOne transaction server to send duplicate real-time events to the adapter, the adapter provides assured-once event delivery. This means that each event is delivered once and only once. To enable assured-once delivery, you must set the `AssuredOnceDelivery` activation specification property to `True`.

Note: When you set the `AssuredOnceDelivery` activation specification property to `True`, you must set the `AutoAcknowledge` activation specification property to `False`. Otherwise, the assured-once delivery feature will not work.

When you set the `AssuredOnceDelivery` activation specification property to `True`, the adapter stores an `XID` (transaction ID) value for each event in the event store.

When a real-time event is obtained, it is processed as follows:

1. The `XID` value for the event is updated in the event store.
2. The event is delivered to its corresponding endpoint.
3. The event entry is deleted from the event store.
4. An acknowledgement is issued to the JD Edwards EnterpriseOne transaction server.

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 functions, XML List calls and the real-time event mechanism.

The following figures illustrate how business objects are used by the adapter for JD Edwards EnterpriseOne for outbound and inbound processing.

How business objects are created

You create business objects by using the J2C Bean wizard, which is launched from Rational Application Developer for WebSphere Software. The J2C Bean 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.

Business object structure

The adapter supports processing of hierarchical business objects. 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 the JD Edwards EnterpriseOne application.

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.x, business graphs are optional; they are required only when you are adding business objects to a module created with a version of Rational Application Developer for WebSphere Software earlier than Version 6.1.x. If business graphs exist, they are processed, but the verb is ignored.

Java data bindings

The business data exchanged between the client application and the resource adapter is represented as Java Data Bindings. The metadata describing the business data is defined as business objects and represented as the XSD schemas. The Java Data Bindings are generated from these XSDs and are the realization of the business objects.

A java data binding is a structure that consists of data and, in some cases, metadata with additional instructions, for processing the data. It is a generated, hierarchical, java objects implementing Record interface. The data can represent a business entity, such as an invoice or an employee record.

You create java data bindings by using the J2C Bean wizard, launched from Rational Application Developer Connector Tools. The wizard connects to the JD Edwards system, discovers data structures in the EIS, and generates data bindings to represent them. The adapter supports records that are hierarchically structured. Information about the processed object is stored in the application-specific information for the object and each of its attributes.

The J2C Bean wizard

The J2C Bean wizard is a tool you use to configure your adapter before deploying it to WebSphere Application Server. The J2C Bean 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 Rational Application Developer for WebSphere Software, you establish a connection to the JD Edwards EnterpriseOne server to browse the database information on the JD Edwards EnterpriseOne server. In the wizard, you select the business objects you need. The wizard automatically generates the XSD schemas that represent the business objects and the corresponding Java data bindings.

The result of running the J2C Bean wizard is a module that contains the interfaces and business objects along with the adapter. You deploy this module on WebSphere Application Server.

For example, if you run the J2C Bean 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.

Standards Compliance

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

Accessibility

Administration

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

- Use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Use voice recognition software, such as IBM via Voice, 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.

J2C Bean wizard

The J2C Bean wizard is the primary component used to create application accessing EIS systems. This wizard is implemented as an Eclipse plug-in that is available through Rational Application Developer for WebSphere Software 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)

IBM WebSphere Application Server, version 6.1.0 and later and its JavaMail component support dual-stack Internet Protocol Version 6.0 (IPv6). For more information about this compatibility in WebSphere Application Server, see IPv6 support in the WebSphere Application Server information center. For more information about IPv6, see <http://www.ipv6.org>.

Chapter 2. Planning for adapter implementation

To implement the application using IBM® WebSphere® Adapter for JD Edwards EnterpriseOne, you must plan for request processing and consider security and performance requirements.

Before you begin

Before you begin

Before you begin to set up and use the adapter, you should possess a thorough understanding of 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.
- 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.

Support for protecting sensitive user data in log and trace files

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

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

To protect the data in situations like this, the adapter lets you specify whether or not you want to prevent confidential user data from displaying in the adapter log and trace files. You can select this option in the J2C Bean wizard or change the HideConfidentialTrace property. When this property is enabled, the adapter replaces the sensitive data with XXX's.

See “Managed connection factory properties” on page 61 for information about this optional property.

The following types of information are considered potentially sensitive data and are disguised:

- The contents of a business object
- The contents of the object key of the event record

- User name, Password, Environment, and Role
- The URL used to connect to the JD Edwards EnterpriseOne server

The following types of information are not considered user data and are not hidden:

- The contents of the event record that are not part of the event record object key, for example, the *XID*, event ID, business object name, and event status
- Business object schemas
- Transaction IDs
- Call sequences

Security

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

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. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

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

- When the J2C Bean 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 Application Server, when the adapter connects to the JD Edwards EnterpriseOne server to process outbound requests.

Authentication in the wizard

The J2C Bean wizard asks for connection information for the discovery process, and then reuses it as the default values of the adapter properties that specify the connection information used at run time. 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 J2C Bean wizard lets you configure the adapter to get the user information using any of the following methods:

- Adapter properties
- J2C authentication alias

Saving the user name and password in adapter properties is a direct way to provide this information at run time. You provide this user name and password when you use the J2C Bean 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 J2C authentication data entry, or authentication alias, created with the Java Authentication and Authorization Service (JAAS) feature of Java 2 security is a robust, secure way to deploy applications. An administrator creates the authentication alias that is used by one or more applications that need to access a system. The user name and password can be known only to that administrator, who can change the password in a single place when a change is required.

Deployment options

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

The deployment options are described below:

- When you deploy the adapter as an embedded component, the adapter is included within an enterprise application archive (EAR) file and is available only to the application in the EAR file.
- When you deploy the adapter as a stand-alone component, the adapter is represented by a stand-alone resource adapter archive (RAR) file. When it is deployed, it is available to all applications deployed in the server instance.

While creating the project for your application using Rational Application Developer for WebSphere Software, 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. To deploy the RAR file to the application server, you must obtain and install Adapter for JD Edwards EnterpriseOne (product number 5724-T75). This provides the RAR file that you install following instructions supplied with WebSphere Application Server.

Considerations for embedding an adapter in the application

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

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

Considerations for using a stand-alone adapter

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

- Stand-alone adapters have no class loader isolation.
Because stand-alone adapters have no class loader isolation, only one version of any given Java artifact is run and the version and sequence of that artifact is undetermined. For example, when you use a stand-alone adapter there is only *one* resource adapter version, *one* adapter foundation class (AFC) version, or *one* third-party JAR version. All adapters deployed as stand-alone adapters share a single AFC version, and all instances of a given adapter share the same code version. All adapter instances using a given third-party library must share that library.
- If you update any of these shared artifacts, all applications using the artifacts are affected.
For instance, if you have an adapter that is working with server version X, and you update the version of the client application to version Y, your original application might stop working.
- Adapter Foundation Classes (AFC) is compatible with previous versions, but the latest AFC version must be in every RAR file that is deployed in a stand-alone manner.
If more than one copy of any JAR file is in the class path in a stand-alone adapter, the one that is used is random; therefore, they all must be the latest version.

WebSphere Adapters in clustered environments

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

WebSphere Application 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 Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) container to activate the adapter instance. The JCA container provides a runtime environment for adapter instances.

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.

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

High availability for inbound processes

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

When you deploy a module to a cluster, the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) container checks the enableHASupport resource adapter property. If the value for the enableHASupport property is true, which is the default setting, all of the adapter instances are registered with the HAManager with a policy 1 of N. This policy means that only one of the adapter instances starts polling for events. Although other adapter instances in the cluster are started, they remain dormant with respect to the active event until the active adapter instance finishes processing the event. If the server on which the polling thread was started shuts down for some reason, an adapter instance that is running on one of the backup servers is activated.

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

High availability for outbound processes

In clustered environments, multiple adapter instances are available to perform outbound process requests. Accordingly, if your environment has multiple applications that interact with WebSphere Adapter for 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.

Chapter 3. Configuring the module for deployment

To configure the adapter so that it can be deployed on WebSphere Application Server, use Rational Application Developer for WebSphere Software 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.

Launching the J2C Bean wizard

To begin the process of creating and deploying a module, you start the J2C Bean wizard in Rational Application Developer for WebSphere Software. The wizard creates a connector project, which is used to organize the files associated with the module.

Before you begin

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

About this task

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

Procedure

1. To start the J2C Bean wizard, go to the Enterprise Explorer of Rational Application Developer for WebSphere Software and click **File** → **New** → **Other** → **J2C** → **J2C Bean**.
2. Click **Next**.
3. In the Resource Adapter Selection window, expand the JD Edwards EnterpriseOne folder and select **IBM WebSphere Adapter for JD Edwards EnterpriseOne (IBM : *version*)**, where *version* is the version of the adapter you want to use, for example, 6.2.0.0
4. Click **Next**.
5. In the window, accept the default project name in the **Connector project** field or type a different name.
6. In the **Target server** field, select the type of server where you will deploy the module. The wizard creates the artifacts that are appropriate to that server.
7. Click **Next**. The Connector Settings window is displayed.

Results

A new connector project is created, which contains the adapter RAR file. The project is listed in the Enterprise Explorer view of the Java EE perspective.

What to do next

Continue working in the J2C Bean wizard. The next step is to add database-specific files to the project.

Adding external software dependencies

The JD Edwards EnterpriseOne application requires that you add external software dependencies to the project. These software dependencies enable the J2C Bean 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 necessary files are listed in the following table.

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

Table 2. External software dependency files required by JD Edwards EnterpriseOne Tools

JD Edwards EnterpriseOne Tools, version 8.9 (SP1, SP2), 8.93	JD Edwards EnterpriseOne Tools, version 8.94	JD Edwards EnterpriseOne Tools, version 8.95, 8.96
connector.jar	Common_Jar.jar	ApplicationAPIs_JAR.jar
database.jar	Connector.jar	ApplicationLogic_JAR.jar
jdeinterop.ini	database.jar	Base_JAR.jar
jdeLog.properties	EventProcessor_EJB.jar	BizLogicContainer_JAR.jar
kernel.jar	jdeutil.jar	BizLogicContainerClient_JAR.jar
log4j.jar	jdbj.ini	bootstrap.jar
owra.jar	jdeinterop.ini	castor.jar
xalan.jar	jdelog.properties	Connector.jar
xerces.jar	kernel.jar	ecutils.jar
JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: <ul style="list-style-type: none"> • tnsnames.ora • ojdbc14.jar 	log4j.jar	EventProcessor_JAR.jar
	xalan.jar	EventProcessor_EJB.jar
	xerces.jar	jdbj.ini

Table 2. External software dependency files required by JD Edwards EnterpriseOne Tools (continued)

JD Edwards EnterpriseOne Tools, version 8.9 (SP1, SP2), 8.93	JD Edwards EnterpriseOne Tools, version 8.94	JD Edwards EnterpriseOne Tools, version 8.95, 8.96
	JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: <ul style="list-style-type: none"> • tnsnames.ora • ojdbc14.jar 	JdbjBase_JAR.jar
		JdbjInterfaces_JAR.jar
		jdeinterop.ini
		jdelog.properties
		JdeNet_JAR.jar
		Improxy.jar
		log4j.jar
		messagingClient.jar
		naming.jar
		PMApi_JAR.jar
		Spec_JAR.jar
		System_JAR.jar
		urlprotocols.jar
		xalan.jar
		xerces.jar
	JDBC driver files For example, if you are using an Oracle database server, use the following JDBC driver files: <ul style="list-style-type: none"> • tnsnames.ora • ojdbc14.jar 	

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 J2C Bean wizard. For example, if you are using an Oracle database server, you can put the tnsnames.ora and the ojdbc14.jar 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.
 - a. Select the version of JD Edwards EnterpriseOne Tools from the left pane. The required files for the version you choose are displayed 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 ojdbc14.jar files, then click **OK**.
4. Click **Next**.

Results

The adapter is configured with the names of the files it needs to communicate with the JD Edwards EnterpriseOne server.

Verify or edit the connection information in some of the external software dependency files to make sure the J2C Bean 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 Rational Application Developer for WebSphere Software 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 J2C Bean wizard and the JD Edwards EnterpriseOne server.

Before you begin

Make sure you have added the external dependency files to your project and that you know the location of the Rational Application Developer for WebSphere Software workplace.

About this task

Verify that the information in your external dependency files correctly reflects the Rational Application Developer for WebSphere Software 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 J2C Bean 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 Rational Application Developer for WebSphere Software 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 Rational Application Developer for WebSphere Software.
 - 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 wizard has the files it needs to connect to the JD Edwards EnterpriseOne server.

In the Discovery and Configuration window, specify the connection properties that the J2C Bean wizard needs to communicate with the JD Edwards EnterpriseOne environment.

Setting connection properties for the J2C Bean wizard

To set connection properties for the J2C Bean 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 J2C Bean 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 J2C Bean 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. To set additional advanced properties, click **Advanced**. The related properties are displayed.

3. If you need to set bidirectional properties, perform the following steps:
 - a. Expand **Bidi properties** and select the **Bidi transformation** check box.
 - b. Set the properties, ordering schema, text direction, symmetric swapping, character shaping, and numeric shaping to control how bidirectional transformation is performed.
4. To change the location of the wizard's log files or the amount of information included in the logs, click the **Specify the level of the logging desired** check box. Provide the following information:
 - a. In the **Log file output location** field, specify the location of the log file for the wizard. Click **Browse** if you want to select a different location.
 - b. In the **Logging Level** field, specify the severity of errors that you want logged. This log information is for the wizard only; at run time, the adapter writes the error messages and the trace information into the standard log and trace files for the server.

Note: This log pertains to the J2C Bean wizard only, not to the operation of the adapter.
5. Click **Next**.

Results

The J2C Bean 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 J2C Bean 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 J2C Bean wizard in Rational Application Developer for WebSphere Software 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 J2C Bean wizard in Rational Application Developer for WebSphere Software 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 J2C Bean wizard.

Before you begin

Make sure you have set the connection properties for the J2C Bean wizard.

About this task

Specify search criteria that the J2C Bean wizard uses to discover business functions on the JD Edwards EnterpriseOne server. The J2C Bean 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, clear the check box for XML Lists so that query finds only business functions. The Add button in the Query Properties window is grayed out.
 - c. Click **OK**.
 2. Click **Run Query**.
 3. In the Discovered objects list, indicate which business function you want to work with.
 - a. Expand **Business functions**. This enables the filter button.
 - 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.
4. Select the business function. For example, if you navigated to **Business Functions** → **CFIN** → **B01000033**, select **GetEffectiveAddress**.
 5. Click the arrow button to add the business function to the **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 J2C Bean 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 the **Add** button.
 - b. In the **Add window**, type a name in the **Value** field. You can type any meaningful name. For example, if the business function is called `GetEffectiveAddress`, you can type **GetEffectiveAddressContainer**.
3. Required: Specify the operation to be associated with the business object.
 - a. Click the **Add** button.
 - b. In the **Add window**, select the operation to import, then click **OK**.
 4. Required: Associate the business functions with an operation.
 - a. Click the **Add** button to create a business function.
 - b. In the **Add window**, select the corresponding the business function, then click **OK**.

5. Optional: Execute a business function multiple times.
 - a. Select the corresponding business function.
 - b. Click the **Add** button to specify the properties for the business function.
 - c. If you want the business function to be executed multiple times select the **Array** check box.
 - d. If you want the business function to be executed even after the business function returns warnings, check the **RollBackOnWarning** check box.
6. 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 artifacts

After you select and configure business objects for your module, use the J2C Bean wizard to configure properties that the adapter uses to connect to a specific database. The wizard creates a new Java project where all the artifacts and property values are saved.

About this task

This task is performed through the J2C Bean Creation and Deployment Configuration window of the J2C Bean wizard.

Procedure

1. In the **Project name** field, select or create a new project into which the J2C Bean is generated.
 - To select an existing project, click **Browse**. If the desired project name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new project:
 - a. Click **New**.
 - b. In the New Source Project Creation window, select **Java project**.
 - c. In the Create a new project window, type a name for the project. For example, MyApdapterOutbound.
 - d. Accept the default values for the other fields.
 - e. Click **Finish**.
2. In the **Package name** field, select or create a new package into which the J2C Bean is generated.
 - To select an existing package, click **Browse**. If the desired package name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new package:
 - a. Click **New**.
 - b. In the Java Package window, click **Java project**.
 - c. In the Create a new project window, type a name for the package. For example, myadapteroutboundpkg.

- d. Accept the default values for the other fields.
 - e. Click **Finish**.
3. In the **Interface name** field, specify the interface name you want to use for your business objects. For example, MyAdapterOutboundInterface. The implementation name is automatically generated by suffixing "Impl" to the interface name and the name is displayed in the **Implementation name** field. For example, MyAdapterOutboundInterfaceImpl.
 4. Optional: Select the **Enable generate Command Bean** check box, to generate a command bean for every operation that you have selected. If you create command bean, you need to specify the command bean name as well as the input and output names.
 5. In the Connection properties area, specify how you want the adapter to connect to the database.
 - To obtain the connection through JNDI, select the **Managed Connection (recommended)** check box. This type of connection is managed by the application server.
 - To obtain the connection directly from the resource adapter, select the **Non-managed Connection** check box.
 6. If you select **Managed Connection (recommended)** check box, specify how you want the adapter to specify the connection properties.
 - To select an existing name, click **Browse**.
 - Otherwise, create a new name.
 - a. Click **New**.
 - b. In the Server selection window, specify the server on which to deploy the adapter and click **Next**.
 - c. In the New J2C Connection Factory window, specify the name in the **JNDI Name** field. For example, com/eis/AdapterCF.
 - d. Click **Finish**.
 7. If you select the **Non-managed Connection** check box, the Database system connection information area is expanded to show the connection information. Review the connection information and change the values if required. You can change the values to specify a different user name and password at run time. You can also connect to an alternate database, although the schema names must be the same in both databases. The format of the connection properties is database-specific.
 8. Optional: Specify advanced properties by clicking **Advanced**. Expand each of the advanced sections to review the properties.
 9. When you are finished setting properties, click **Finish**.

Results

The new project is added to the Enterprise Explorer perspective. The module is created in the project and artifacts are generated.

The generated artifacts allow you to build an enterprise application that accesses the EIS. You can use the J2C Bean and Java data bindings directly, in the non-managed mode or generate JSP or EJB that uses the J2C Bean. Rational Application Developer provides tools to automate this generation. You can access these tools from the **New>Others** menu, for their description refer to the Rational Application Developer documentation.

Generating the EJB or JSP project

After you create the Java project, use the Web Page, Web Service, or EJB from J2C Java wizard to create the EJB or JSP project.

About this task

Procedure

1. To start the J2C Bean wizard, go to the Enterprise Explorer of Rational Application Developer for WebSphere Software and click **New>Other>J2C>Web Page, Web Service, or EJB from J2C Java Bean**.
2. Click **Next**.
3. In the J2C Java bean selection window, click **Browse**.
4. In the Find J2C Bean window, type the first letter of the implementation name generated earlier or type the full name and press Enter.
5. Select the implementation name from the list and click **OK**.
6. Click **Next**.
7. In the Deployment Information window, select the Java EE Resource Type as **EJB** or **Sample JSP** and click **Next**.

Note: In the Deployment Information window, the **Check Configure Resource Adapter Deployment** check box is available for selection only if you have selected the **Non-managed Connection** check box when specifying the deployment settings.

8. If you select **EJB**, the EJB Creation wizard is displayed. This wizard creates the Java project as an EJB project.
 - To select an existing project, click **Browse**. If the desired project name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new project.
 - a. Click **New**.
 - b. In the EJB Project window, type an EJB project name. For example, MyAdapterOutboundEJB.
 - c. In the EAR Membership area, click **New** to create a new ear project name.
 - d. In the EAR Application Project window, type an EAR project name. For example, MyAdapterOutboundEJB.EAR.
 - e. Click **Next**.
 - f. In the Enterprise Application window, select the JD Edwards adapter RAR dependency such as CWYED_JDE.
 - g. Select the **Generate Deployment Descriptor** check box.
 - h. Click **Finish** to return to the EJB Project window.
 - i. Click **Finish** to return to the EJB Creation window.
 - j. Click **Next**.

Note: In the EJB Creation window, the **Next** button is available for selection only if you have selected the **Configure Resource Adapter Deployment** check box when specifying the deployment information. After you create the EJB and EAR project and return to EJB Creation window, you can directly click **Finish** to complete the process. The adapter is deployed as stand alone by default.

- k. In Resource Adapter Deployment window, select **Deploy within EAR** and select the EJB EAR Project you created.
 - l. Click **Finish**.
9. If you select **Sample JSP**, the Simple JSP Creation wizard is displayed. .
- To select an existing project, click **Browse**. If the desired project name appears in the Project Selection list, select its name.
 - Otherwise, create a new project.
 - a. Click **New**.
 - b. In Dynamic Web Project window, type a project name. For example, MyAdapterOutboundWEB.
 - c. In EAR Membership area, click **New** to create a new ear project name.
 - d. In EAR Application Project window, type an EAR project name. For example, MyAdapterOutboundWEBEAR.
 - e. Click **Next**.
 - f. In the Java EE Module dependencies area of the Enterprise Application window, select the JD Edwards adapter RAR dependency such as CWYED_JDE.
 - g. Select the **Generate Deployment Descriptor** check box.
 - h. Click **Finish** to return to the Dynamic Web Project window.
 - i. Click **Finish** to return to the Simple JSP Creation window.
 - j. Click **Next**.
- Note:** In the EJB Creation window, the **Next** button is available for selection only if you have selected the **Configure Resource Adapter Deployment** check box when specifying the deployment information. After you create the WEB and EAR project and return to Simple JSP Creation window, you can directly click **Finish** to complete the process. The adapter is deployed as stand alone by default.
- k. In Resource Adapter Deployment window, select **Deploy within EAR** and select the Web EAR Project you created.
 - l. Click **Finish**.
10. Export the project 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 J2C Bean wizard in Rational Application Developer for WebSphere Software 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 J2C Bean wizard.

Before you begin

Make sure you have set the connection properties for the J2C Bean 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 J2C Bean wizard uses to discover database table information on the JD Edwards EnterpriseOne server. The J2C Bean 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.

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 J2C Bean 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.
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.

The following table types are available.

Table 3. 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_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: <ul style="list-style-type: none"> • 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.
 - 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 J2C Bean 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: 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 J2C Bean 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 artifacts

After you select and configure business objects for your module, use the J2C Bean wizard to configure properties that the adapter uses to connect to a specific database. The wizard creates a new Java project where all the artifacts and property values are saved.

About this task

This task is performed through the J2C Bean Creation and Deployment Configuration window of the J2C Bean wizard.

Procedure

1. In the **Project name** field, select or create a new project into which the J2C Bean is generated.
 - To select an existing project, click **Browse**. If the desired project name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new project:
 - a. Click **New**.
 - b. In the New Source Project Creation window, select **Java project**.
 - c. In the Create a new project window, type a name for the project. For example, MyApdapterOutbound.
 - d. Accept the default values for the other fields.
 - e. Click **Finish**.
2. In the **Package name** field, select or create a new package into which the J2C Bean is generated.
 - To select an existing package, click **Browse**. If the desired package name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new package:
 - a. Click **New**.
 - b. In the Java Package window, click **Java project**.
 - c. In the Create a new project window, type a name for the package. For example, myadapteroutboundpkg.
 - d. Accept the default values for the other fields.
 - e. Click **Finish**.
3. In the **Interface name** field, specify the interface name you want to use for your business objects. For example, MyAdapterOutboundInterface. The implementation name is automatically generated by suffixing "Impl" to the interface name and the name is displayed in the **Implementation name** field. For example, MyAdapterOutboundInterfaceImpl.
4. Optional: Select the **Enable generate Command Bean** check box, to generate a command bean for every operation that you have selected. If you create command bean, you need to specify the command bean name as well as the input and output names.
5. In the Connection properties area, specify how you want the adapter to connect to the database.

- To obtain the connection through JNDI, select the **Managed Connection (recommended)** check box. This type of connection is managed by the application server.
 - To obtain the connection directly from the resource adapter, select the **Non-managed Connection** check box.
6. If you select **Managed Connection (recommended)** check box, specify how you want the adapter to specify the connection properties.
 - To select an existing name, click **Browse**.
 - Otherwise, create a new name.
 - a. Click **New**.
 - b. In the Server selection window, specify the server on which to deploy the adapter and click **Next**.
 - c. In the New J2C Connection Factory window, specify the name in the **JNDI Name** field. For example, com/eis/AdapterCF.
 - d. Click **Finish**.
 7. If you select the **Non-managed Connection** check box, the Database system connection information area is expanded to show the connection information. Review the connection information and change the values if required. You can change the values to specify a different user name and password at run time. You can also connect to an alternate database, although the schema names must be the same in both databases. The format of the connection properties is database-specific.
 8. Optional: Specify advanced properties by clicking **Advanced**. Expand each of the advanced sections to review the properties.
 9. When you are finished setting properties, click **Finish**.

Results

The new project is added to the Enterprise Explorer perspective. The module is created in the project and artifacts are generated.

The generated artifacts allow you to build an enterprise application that accesses the EIS. You can use the J2C Bean and Java data bindings directly, in the non-managed mode or generate JSP or EJB that uses the J2C Bean. Rational Application Developer provides tools to automate this generation. You can access these tools from the **New>Others** menu, for their description refer to the Rational Application Developer documentation.

Configuring the module for inbound processing

To configure a module to use the adapter for inbound processing, use the J2C Bean wizard in Rational Application Developer for WebSphere Software to find and select business objects and services from the JD Edwards EnterpriseOne server, and to generate business object definitions and related artifacts.

Selecting business objects and services

To specify which real-time events you want to work with, you provide information in the J2C Bean wizard.

Before you begin

Make sure you have set the connection properties correctly for the J2C Bean wizard.

About this task

Specify search criteria that the J2C Bean wizard uses to discover real-time events on the JD Edwards EnterpriseOne server. The J2C Bean wizard returns a list of business functions that meet the search criteria.

To specify the search criteria and select a real-time event, use the following procedure.

Procedure

1. In the Object Discovery and Selection window, make sure the query is set up to find real-time events in the JD Edwards EnterpriseOne server. In the Query field, make sure you see the following query criteria: **Real-Time Events=true**. If you see **Real-Time=false**, use the following procedure to set it to true.
 - a. Click **Edit Query**.
 - b. In the Query Properties window, select **Real-Time Events**.
 - c. Click **OK**.
2. Click **Run Query**.
3. In the Discovered objects list, expand the following root node: **Real-Time Events** and select one or more real-time events you want to work with.
4. Click the arrow button to add the real-time event to the **Selected objects** list.
5. In the Configuration Properties window, you can associate an operation with the action needed for the selected real-time event. This is required for real-time events which do not already have an operation field to store the action for the real-time event. If the real-time event already has an operation field, then you need only specify the name of the operation, not the field containing the associated action.
 - To associate an operation field with the selected real-time event, use the following procedure.
 - a. In the Selection field, select **Config with Operation Field**. The Configuration Properties window expands.
 - b. In the expanded window, enter information in the following fields:
 - List Children: All single real-time events which are contained by the container real-time event or the single real-time event itself
 - Operation Field: The operation field stores the action information associated with the operation.
 - 1) Click **Select** to select a value for the operation field.
 - 2) In the Select window, select the action for the operation, then click **OK**.
 - User-defined: Create: Leave the default number ("1") to use for mapping the Create operation to the selected action, or specify a new number.
 - User-defined: Update: Leave the default number ("2") to use for mapping the Update operation to the selected action, or specify a new number.
 - User-defined: Delete: Leave the default number ("3") to use for mapping the Delete operation to the selected action, or specify a new number.
 - c. Click **OK**.
 - To select the real-time event without associating an operation field, use the following procedure.

- a. In the Selection field, select **Configuration without Operation Field**.
- b. In the Configuration Properties window, specify the operation to use with the selected real-time event.
- c. Click **OK**.

The selected real-time event is added to the Selected objects list.

6. Click **Next**.

Results

You have selected the real-time event you want to work with and configured the appropriate operation mapping relationship for it.

What to do next

From the Configure Composite Properties window, you specify the directory where to store the real-time events.

Configuring the selected objects

To configure the real-time event, you specify information about the object.

Before you begin

Make sure you have selected and imported the real-time event.

About this task

To configure the real-time event, use the following procedure.

Procedure

1. In the Configure Composite Properties window, specify the directory to store the business object in the **Relative path** field.
2. Click **Next**.

Results

You specified a location to store the real-time event. The Service Generation and Deployment Configuration window is displayed.

What to do next

Generate a deployable module that includes the adapter and the real-time event.

Setting deployment properties and generating artifacts

After you select and configure business objects for your module, use the J2C Bean wizard to configure properties that the adapter uses to connect to a specific database. The wizard creates a new Java project where all the artifacts and property values are saved.

About this task

This task is performed through the J2C Bean Creation and Deployment Configuration window of the J2C Bean wizard.

Procedure

1. In the **EJB Project name** field, select or create a new EJB project.
 - To select an existing project, click **Browse**. If the desired project name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new project:
 - a. Click **New**.
 - b. In the EJB Project window, type a project name. For example, MyAdapterInboundEJB.
 - c. In the EAR Membership area, click **New** to create a new ear project.
 - d. In EAR Application Project window, type an EAR project name. For example, MyAdapterInboundEJB EAR
 - e. Click **Finish** to return to the EJB Project window.
 - f. Click **Finish**.
2. In the **Package name** field, select or create a new package into which the J2C Bean is generated.
 - To select an existing package, click **Browse**. If the desired package name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new package:
 - a. Click **New**.
 - b. In the New Java Package window, type a name for the package. For example, myadapterinboundejbpkg.
 - c. Click **Finish**.
3. In the **Stateless Session EJB's local business interface name** field, specify the interface name you want to use for your business object. For example, MyAdapterInboundInterface. The interface name is suffixed with "MDB" and it is displayed automatically in the **Message Driven EJB name** field. For example, MyAdapterInboundInterfaceMDB. Similarly, the interface name is suffixed with "SB" and it is displayed automatically in the **Stateless Session EJB name** field. For example, MyAdapterInboundInterfaceSB.
4. In the Inbound Connection configuration area, specify the JNDI name, which JNDI data source for an existing activation specification in WAS server or you can create it later. For example, com/eis/AdapterAS.
5. Click **Finish**.

Chapter 4. Configuring the application on WebSphere Application Server

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 Application Server.

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>Application Types**, select **WebSphere enterprise applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change. The **Configuration** page is displayed.
4. Under **Modules**, click **Manage Modules**.
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 57 for more information about these properties.

- a. Click the name of the property. The **Configuration** page for the selected property is displayed.
- b. Change the contents of the **Value** field or type a value, if the field is empty.

- 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 Application Server.

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>Application Types**, select **WebSphere enterprise applications**.
3. In the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change.
4. Under **Modules**, click **Manage Modules**.
5. Click **IBM WebSphere Adapter for JD Edwards EnterpriseOne**.
6. In the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
8. Click the name of the connection factory associated with your adapter module.
9. In the **Additional Properties** list, click **Custom properties**.

Custom properties are those J2C connection factory properties that are unique to Adapter for 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 61 for more information about these properties.

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

Results

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

Setting activation specification properties for embedded adapters

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

Before you begin

Your adapter module must be deployed on WebSphere Application Server.

About this task

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

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

Procedure

1. Start the administrative console.
2. Under **Applications>Application Types**, select **WebSphere 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**.
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 activation specifications**.
8. Click the name of the activation specification associated with the adapter module.
9. From the **Additional Properties** list, click **J2C activation specification custom properties**.
10. For each property you want to change, perform the following steps.

Note: See “Activation specification properties” on page 74 for more information about these properties.

- a. Click the name of the property.

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

Results

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

Changing configuration properties for stand-alone adapters

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

Setting resource adapter properties for stand-alone adapters

To set resource adapter properties for your stand-alone adapter after it has been installed on WebSphere Application Server, 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 Application Server.

About this task

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

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

Procedure

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for JD Edwards EnterpriseOne**.
4. In the **Additional Properties** list, click **Custom properties**.
5. For each property you want to change, perform the following steps.
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty. You can change the number in the **Value** field and add a description of the property.
 - c. Click **OK**.
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 Application Server, 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 Application Server.

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. In the Resource adapters page, click **IBM WebSphere Adapter for JD Edwards EnterpriseOne**.
4. In the **Additional Properties** list, click **J2C connection factories**.
5. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

Note: If you selected **Use predefined connection properties** when you used the J2C Bean 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. Optional: 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.
6. In the list of connection factories, click the one you want to use.
 7. In the **Additional Properties** list, click **Custom properties**.

Custom properties are those J2C connection factory properties that are unique to Adapter for 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 61 for more information about these properties.
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
9. After you have finished setting properties, click **Apply**.
10. Click **Save** in the **Messages** box at the top of the window.

Results

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

Setting activation specification properties for stand-alone adapters

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

Before you begin

Your adapter must be installed on WebSphere Application Server.

About this task

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

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

Procedure

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

Note: If you selected **Use predefined connection properties** when you used the J2C Bean wizard to configure the adapter module, you do not need to create an activation specification.

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

- a. Click **New**.
- b. In the **General Properties** section of the **Configuration** tab, type a name for the activation specification. For example, you could type AdapterAS.
- c. Type a value for **JNDI name**. For example, you could type com/eis/AdapterAS.

- d. Optional: Select an authentication alias from the **Authentication alias** list.
- e. Select a message listener type.
- f. Click **OK**.
- g. Click **Save** in the **Messages** box at the top of the page.
The newly created activation specification is displayed.
6. In the list of activation specifications, click the one you want to use.
7. In the Additional Properties list, click **J2C activation specification custom properties**.
8. For each property you want to set, perform the following steps.
 - 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 page.

Results

The activation specification properties associated with your adapter are set.

Adding dependency libraries to the deployed resource adapter

The deployed resource adapter running in the WebSphere Application Server requires the same dependency libraries as it does in Rational Application Developer for WebSphere Software to process requests. The method for adding these library files depends on the mode of the resource adapter deployment: standalone or embedded in the EAR file.

Standalone deployment

The dependency libraries may be added to the resource adapter deployed standalone either during initial deployment of the RAR file or by configuring the Resource Adapter properties after deployment.

About this task

To set the values during initial deployment of the RAR file, specify Class path and Native path locations. Class path is used to point to JAR files, and Native path is used to point to native libraries, such as *.dll, *.so.

To set the dependency library path files after the adapter has been installed on WebSphere Application Server, use the administrative console to modify the values for the Resource Adapter.

EAR deployment

For the rare case when the connector needs to be embedded in the EAR file, the dependant libraries are added as shared libraries. Define the appropriate shared library containing external dependencies and associate them with the EAR file.

About this task

There are two methods to do this task:

- Using enhanced EAR editor in Rational Application Developer for WebSphere Software

- Using administrative console of the WebSphere Application Server

Using enhanced EAR editor:

You can use the EAR editor in Rational Application Developer for WebSphere Software to add the dependency libraries.

About this task

To create shared libraries using the EAR editor, use the following procedure.

Procedure

1. Open Enhanced EAR editor.
2. Click **Deployment** tab.
3. Expand **Shared Library** Shared Library section.
4. Click **Add** to add new shared library.
5. Specify Shared library parameters and click **OK**.
6. Deploy the EAR to the server.

Using administrative console of the WebSphere Application Server:

You can use the administrative console of the WebSphere Application Server to add the dependency libraries.

Before you begin

Make sure dependent files are available on the server machine in the separate folder. If needed, copy dependent files on the server machine.

Procedure

1. Define WebSphere variables to point to appropriate folders.
2. Define the shared library through the server administrative console; you can specify it using WebSphere variables defined in above step 1.
3. Deploy the EAR to the server.
4. Configure the EAR to reference defined shared library.

Chapter 5. Troubleshooting and support

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

Support for the Log and Trace Analyzer

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

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

The adapter's component ID for filtering and highlighting is a string composed of the characters JDERA plus the value of the adapter ID property. For example, if the adapter ID property is set to 001, the component ID is JDERA001.

If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for JD Edwards EnterpriseOne to 001 and 002. The component IDs for those instances, JDERA001 and JDERA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to JDERAInstance.

For outbound processing, the adapter ID property is located in both the resource adapter and managed connection factory property groups. If you update the adapter ID property after using the J2C Bean wizard to configure the adapter for outbound processing, be sure to set the resource adapter and managed connection factory properties consistently, to prevent inconsistent marking of the log and trace entries. For inbound processing, the adapter ID property is located only in the resource adapter properties, so this consideration does not apply.

For more information about the adapter ID property, see “Adapter ID (AdapterID)” on page 58.

Configuring logging and tracing

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

About this task

Configuring logging properties

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

About this task

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

Use the administrative console to perform the following tasks:

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

If you set the output for log analyzer format, you can open trace output using the Log Analyzer tool, which is an application included with your application 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 an application server, including service components and event points, see the documentation for your application server.

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

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

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

Procedure

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

- For the Adapter for 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.
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 application 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 application 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 > Application Types > WebSphere application servers**.
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 Application Server.

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 Application Server documentation.

XAResourceNotAvailableException

When the application 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 application server log file:

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

Problem:

A resource was removed while the application 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 application 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 Rational Application Developer for WebSphere Software, 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\server_name\  
transaction\tranlog
```

3. Start the application 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. The plug-in supports the dynamic trace feature. For information about installing or using IBM Support Assistant, visit this address:

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

Chapter 6. Reference information

To support you in your tasks, reference information includes details about business objects that are generated by the J2C Bean 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 J2C Bean 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.

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 J2C Bean 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 4. 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 5. 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> YES: the parameter is required. N0: the parameter is not required. DEFAULT: the parameter is using the JD Edwards EnterpriseOne value. For adapter purposes, it is processed as N0.
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 6. Application-specific information at the operation level

Application-specific information	Description	Value
Name	The name of the business object operation	<ul style="list-style-type: none"> Create Retrieve Update Delete RetrieveAll

Table 6. Application-specific information at the operation level (continued)

Application-specific information	Description	Value
BSFN.Name	The name of the business functions to process	<ul style="list-style-type: none"> 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 7. 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 8. 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 J2C Bean 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 J2C Bean 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 J2C Bean 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 Rational Application Developer for WebSphere Software 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 9. 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 J2C Bean wizard	GetEffectiveAddress
Name of the XML List	Name of the XML List table discovered by the J2C Bean wizard	F0116

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

Outbound configuration properties

WebSphere Adapter for JD Edwards EnterpriseOne has several categories of outbound connection configuration properties, which you set with the J2C Bean wizard while generating or creating objects and services. You can change the resource adapter and managed connection factory properties after you deploy the module to WebSphere Application Server using Rational Application Developer for WebSphere Software or the administrative console, but connection properties for the J2C Bean wizard cannot be changed after deployment.

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	<p>A required field (property) must have a value in order for the adapter to work. Sometimes the J2C Bean wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the J2C Bean wizard <i>will not change that default value</i>. When a required field contains no value at all, the J2C Bean wizard will process the field using its assigned default value, and that default value will also be displayed on the administrative console.</p> <p>Possible values are Yes and No.</p> <p>Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example,</p> <ul style="list-style-type: none"> • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	<p>The predefined value that is set by the J2C Bean wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	<p>Describes the property type. Valid property types include the following:</p> <ul style="list-style-type: none"> • Boolean • String • Integer
Usage	<p>Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented:</p> <p>For Rational Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case sensitive • Can be up to 40 characters in length. <p>This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.</p>
Example	<p>Provides sample property values, for example:</p> <p>"If Language is set to JA (Japanese), Codepage number is set to 8000".</p>
Globalized	<p>If a property is globalized, it has national language support, meaning that you can set the value in your national language.</p> <p>Valid values are Yes and No.</p>
Bidi supported	<p>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.</p> <p>Valid values are Yes and No.</p>

Setting connection properties for the J2C Bean wizard

J2C Java Bean connection properties establish a connection between the J2C Bean wizard of Rational Application Developer for WebSphere Software, a tool that is used to create business objects, and the JD Edwards EnterpriseOne server. The properties you configure in the J2C Bean wizard specify such things as connection configuration, bidi properties, and logging and tracing options.

Once a connection between the J2C Bean wizard and the JD Edwards EnterpriseOne server is established, the J2C Bean 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 52.

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 10. 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 55	Specifies the location of the log file for external service
“Logging level” on page 55	Specifies the type error for which logging will occur during external service
“Password” on page 56	Password of the adapter user account on the JD Edwards EnterpriseOne environment
“Role” on page 56	Name of the role that is associated with the user name used to access the JD Edwards EnterpriseOne environment.
“User name” on page 57	Name of the adapter user account on the JD Edwards EnterpriseOne environment

Environment

This property specifies the JD Edwards EnterpriseOne environment name.

Table 11. 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 11. Environment details (continued)

Bidi supported	Yes
----------------	-----

Log file output location

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

Table 12. 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 13. 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 13. Logging level details (continued)

Example	<p>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.</p> <p>Other error descriptions are as follows:</p> <ul style="list-style-type: none"> • 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 14. 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 15. 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	<ul style="list-style-type: none"> • System administrator • Human resources • Accounting

Table 15. 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 16. 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 J2C Bean wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

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

The following property that was specified as a resource adapter property in Version 6.0.2.x applies to the managed connection factory property group in Version 6.1.x.

- 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 52.

Table 17. Resource adapter properties for Adapter for JD Edwards EnterpriseOne

Name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing

Table 17. Resource adapter properties for Adapter for JD Edwards EnterpriseOne (continued)

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

Adapter ID (AdapterID)

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

Table 18. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, JDERA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is JDERA001.</p> <p>If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter.</p> <p>To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for JD Edwards EnterpriseOne to 001 and 002. The component IDs for those instances, JDERA001 and JDERA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to JDERAInstance.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	Yes
Bidi supported	No

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

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

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

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.
Globalized	No
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

Table 21. Log file name details (continued)

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.
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 J2C Bean 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 52.

Note: The J2C Bean wizard refers to these properties as managed connection factory properties and WebSphere Application Server administrative console refers to these as (J2C) connection factory properties.

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

Property name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Disguise user data as “XXX” in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of X’s instead of user data in log and trace files
“Environment” on page 63	environment	Specifies the JD Edwards EnterpriseOne environment name
“Password” on page 63	password	Password of the adapter user account on the JD Edwards EnterpriseOne environment
“Role” on page 63	role	Name of the role that is associated with the user name used to access the JD Edwards EnterpriseOne environment.
“Timeout” on page 64	timeout	This property is the global timeout value, in milliseconds, set on the XML List request execute call.
“User name” on page 64	userName	Name of the adapter user account on the JD Edwards EnterpriseOne environment

Adapter ID (AdapterID)

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

Table 27. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, JDERA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is JDERA001.</p> <p>If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter.</p> <p>To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for JD Edwards EnterpriseOne to 001 and 002. The component IDs for those instances, JDERA001 and JDERA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to JDERAInstance.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	Yes
Bidi supported	No

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

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

Table 28. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	No

Table 28. Disguise user data as "XXX" in log and trace files details (continued)

Bidi supported	No
----------------	----

Environment

This property specifies the JD Edwards EnterpriseOne environment name.

Table 29. 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 30. 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 31. 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.

Table 31. Role details (continued)

Examples	<ul style="list-style-type: none"> • System administrator • Human resources • Accounting
Globalized	No
Bidi supported	Yes

Timeout

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

Table 32. 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 33. 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 J2C Bean 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 Rational Application Developer for WebSphere Software. 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 52.

Table 34. Interaction specification property for the Adapter for JD Edwards EnterpriseOne

Property name		Description
In the wizard	In the assembly editor	
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 35. 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

Inbound configuration properties

WebSphere Adapter for JD Edwards EnterpriseOne has several categories of inbound connection configuration properties, which you set with the J2C Bean wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using Rational Application Developer for WebSphere Software or the administrative console, but connection properties for the J2C Bean wizard cannot be changed after deployment.

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	<p>A required field (property) must have a value in order for the adapter to work. Sometimes the J2C Bean wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the J2C Bean wizard <i>will not change that default value</i>. When a required field contains no value at all, the J2C Bean wizard will process the field using its assigned default value, and that default value will also be displayed on the administrative console.</p> <p>Possible values are Yes and No.</p> <p>Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example,</p> <ul style="list-style-type: none"> • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	<p>The predefined value that is set by the J2C Bean wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	<p>Describes the property type. Valid property types include the following:</p> <ul style="list-style-type: none"> • Boolean • String • Integer
Usage	<p>Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented:</p> <p>For Rational Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case sensitive • Can be up to 40 characters in length. <p>This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.</p>
Example	<p>Provides sample property values, for example:</p> <p>"If Language is set to JA (Japanese), Codepage number is set to 8000".</p>
Globalized	<p>If a property is globalized, it has national language support, meaning that you can set the value in your national language.</p> <p>Valid values are Yes and No.</p>
Bidi supported	<p>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.</p> <p>Valid values are Yes and No.</p>

Setting connection properties for the J2C Bean wizard

J2C Java Bean connection properties establish a connection between the J2C Bean wizard of Rational Application Developer for WebSphere Software, a tool that is used to create business objects, and the JD Edwards EnterpriseOne server. The properties you configure in the J2C Bean wizard specify such things as connection configuration, bidi properties, and logging and tracing options.

Once a connection between the J2C Bean wizard and the JD Edwards EnterpriseOne server is established, the J2C Bean 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 52.

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 36. 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 68	Specifies the location of the log file for external service
“Logging level” on page 68	Specifies the type error for which logging will occur during external service
“Password” on page 69	Password of the adapter user account on the JD Edwards EnterpriseOne environment
“Role” on page 69	Name of the role that is associated with the user name used to access the JD Edwards EnterpriseOne environment.
“User name” on page 70	Name of the adapter user account on the JD Edwards EnterpriseOne environment

Environment

This property specifies the JD Edwards EnterpriseOne environment name.

Table 37. 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 37. Environment details (continued)

Bidi supported	Yes
----------------	-----

Log file output location

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

Table 38. 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 39. 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 39. Logging level details (continued)

Example	<p>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.</p> <p>Other error descriptions are as follows:</p> <ul style="list-style-type: none"> • 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 40. 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 41. 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	<ul style="list-style-type: none"> • System administrator • Human resources • Accounting

Table 41. 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 42. 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 J2C Bean wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

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

The following property that was specified as a resource adapter property in Version 6.0.2.x applies to the managed connection factory property group in Version 6.1.x.

- 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 52.

Table 43. Resource adapter properties for Adapter for JD Edwards EnterpriseOne

Name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing

Table 43. Resource adapter properties for Adapter for JD Edwards EnterpriseOne (continued)

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

Adapter ID (AdapterID)

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

Table 44. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, JDERA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is JDERA001.</p> <p>If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter.</p> <p>To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for JD Edwards EnterpriseOne to 001 and 002. The component IDs for those instances, JDERA001 and JDERA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to JDERAInstance.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	Yes
Bidi supported	No

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

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

Table 45. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.
Globalized	No
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 46. 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 47. Log file name details

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

Table 47. Log file name details (continued)

Bidi supported	Yes
----------------	-----

Log number of files (LogNumberOfFiles)

This property specifies the number of log files.

Table 48. 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 49. 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 50. Trace file name details

Required	No
Default	No default value
Unit of measure	Kilobytes
Property type	String
Usage	This property is deprecated.
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 51. 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

Activation specification properties

Activation specification properties hold the inbound event processing configuration information for a message endpoint. You can set activation specification properties using either the external service wizard or the administrative console.

Activation specification properties are used during endpoint activation to notify the adapter of eligible event listeners. During inbound processing, the adapter uses these event listeners to receive events before forwarding them to the endpoint.

The following table lists the activation specification properties for inbound communication. You set the activation specification properties using the external service wizard and can change them using the Rational Application Developer for WebSphere Software Assembly Editor, or after deployment through the administrative console. A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 52.

Table 52. Activation specification properties

Property name		Description
In the wizard	In the administrative console	
Auto Acknowledge	AutoAcknowledge	Specifies the event acknowledge mode that is used.
Delivery type	DeliveryType	Determines the order in which events are delivered by the adapter to the export.
Ensure once-only event delivery	AssuredOnceDelivery	Specifies whether the adapter provides assured once delivery of events.
“Failed events folder” on page 76	FailedEventsFolder	The absolute path to the file folder on the local system where unsuccessfully processed events are archived in file format.
Retry limit for failed events	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed
Interval between polling periods	PollPeriod	The length of time that the adapter waits between polling periods.
Maximum connections	MaximumConnections	The maximum number of connections that the adapter can use for inbound event delivery.
Maximum events in polling period	PollQuantity	The number of events that the adapter delivers to the export during each poll period.
Minimum connections	MinimumConnections	The minimum number of connections that the adapter can use for inbound event delivery.

Table 52. Activation specification properties (continued)

Property name		Description
In the wizard	In the administrative console	
No Wait	NoWait	Specifies whether the adapter waits for a time interval to get an event from the JD Edwards EnterpriseOne transaction server by invoking the Dynamic Java Connector API.
Number of times to retry the system connection	RetryLimit	The number of times the adapter tries to reestablish an inbound connection after an error.
Retry EIS connection on startup	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the JD Edwards EnterpriseOne server if it cannot connect at startup
Retry interval if connection fails	RetryInterval	The length of time that the adapter waits between attempts to establish a new connection after an error during inbound operations.
Stop the adapter when an error is encountered while polling	StopPollingOnError	Specifies whether the adapter stops polling for events when it encounters an error during polling.
Wait Time	WaitTime	Specifies the waiting time if the No Wait property is false.

Auto acknowledge (AutoAcknowledge)

This property specifies the event acknowledge mode that is used. You can specify either the auto acknowledge mode or the client acknowledge mode.

Table 53. Auto acknowledge details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	Specifies the event acknowledge mode, which is either the auto acknowledge mode or the client acknowledge mode.
Example	False
Globalized	No
Bidi supported	Yes

Delivery type (DeliveryType)

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

Table 54. Delivery type details

Required	No
Possible values	ORDERED UNORDERED
Default	ORDERED

Table 54. Delivery type details (continued)

Property type	String
Usage	The following values are supported: <ul style="list-style-type: none"> • ORDERED: The adapter delivers events to the export one at a time. • UNORDERED: The adapter delivers all events to the export at once.
Globalized	No
Bidi supported	No

Ensure once-only event delivery (AssuredOnceDelivery)

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

Table 55. Ensure once-only event delivery details

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

Failed events folder

This property specifies the file folder on the local system where unsuccessfully processed events are archived in file format.

Table 56. Failed events folder details

Required	No
Possible values	No default value
Default	Null
Property type	String
Usage	Manually create this folder on the same system before the adapter is run. If no failed events folder is specified, the adapter does not archive unsuccessfully processed events.
Globalized	Yes
Bidi supported	No

Retry limit for failed events (FailedEventRetryLimit)

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

Table 57. Retry limit for failed events details

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

Interval between polling periods (PollPeriod)

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

Table 58. Interval between polling periods details

Required	Yes
Possible values	Integers greater than or equal to 0.
Default	2000
Unit of measure	Milliseconds
Property type	Integer
Usage	<p>The poll period is established at a fixed rate, which means that if running the poll cycle is delayed for any reason (for example, if a prior poll cycle takes longer than expected to complete) the next poll cycle will occur immediately to make up for the lost time caused by the delay.</p>
Globalized	No
Bidi supported	No

Maximum connections (MaximumConnections)

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

Table 59. Maximum connections details

Required	No
----------	----

Table 59. Maximum connections details (continued)

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

Maximum events in polling period (PollQuantity)

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

Table 60. Maximum events in polling period details

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

Minimum connections (MinimumConnections)

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

Table 61. Minimum connections details

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

No wait (NoWait)

This property specifies whether the adapter waits for a time interval to get an event from the JD Edwards EnterpriseOne transaction server by invoking the Dynamic Java Connector API.

Table 62. No wait details

Required	No
Possible values	True False
Default	True

Table 62. No wait details (continued)

Property type	Boolean
Usage	Specifies whether the adapter waits for a time interval to get an event from the JD Edwards EnterpriseOne transaction server by invoking the Dynamic Java Connector API.
Example	True
Globalized	No
Bidi supported	Yes

Number of times to retry the system connection (RetryLimit)

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

Table 63. Number of times to retry the system connection details

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

Retry EIS connection on startup (RetryConnectionOnStartup)

This property controls whether the adapter attempts to connect again to the JD Edwards EnterpriseOne server if it cannot connect at startup.

Table 64. Retry EIS connection on startup details

Required	No
Possible values	True False
Default	False
Property type	Boolean

Table 64. Retry EIS connection on startup details (continued)

Usage	<p>This property indicates whether the adapter should retry the connection to the JD Edwards EnterpriseOne server if the connection cannot be made when the adapter is started:</p> <ul style="list-style-type: none"> • Set the property to <code>False</code> when you want immediate feedback about whether the adapter can establish a connection to the JD Edwards EnterpriseOne server, for example, when you are building and testing the application that receives events from the adapter. If the adapter cannot connect, the adapter writes log and trace information and stops. The administrative console shows the application status as <code>Stopped</code>. After you resolve the connection problem, start the adapter manually. • Set the property to <code>True</code> if you do not need immediate feedback about the connection. If the adapter cannot connect during startup, it writes log and trace information, and then attempts to reconnect, using the <code>RetryInterval</code> property to determine how frequently to retry and the value of the <code>RetryLimit</code> property to retry multiple times until that value is reached. The administrative console shows the application status as <code>Started</code>.
Globalized	No
Bidi supported	No

Retry interval if connection fails (RetryInterval)

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

Table 65. Retry interval details

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

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

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

Table 66. Stop the adapter when an error is encountered while polling details

Required	No
Possible values	<code>True</code> <code>False</code>
Default	<code>False</code>
Property type	Boolean
Usage	<p>If this property is set to <code>True</code>, the adapter stops polling when it encounters an error.</p> <p>If this property is set to <code>False</code>, the adapter logs an exception when it encounters an error during polling and continues polling.</p>

Table 66. Stop the adapter when an error is encountered while polling details (continued)

Globalized	No
Bidi supported	No

Wait time (WaitTime)

This property specifies the waiting time if the No Wait property is false.

Table 67. Wait time details

Required	No
Possible values	Any positive integer. Any negative integer will be treated as the default value (3000 milliseconds)
Default	3000
Unit of measure	Millisecond
Property type	Integer
Usage	This property specifies the waiting time if the No Wait property is false.
Example	5000
Globalized	No
Bidi supported	Yes

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, Rational Application Developer for WebSphere Software, and WebSphere Application Server are written in Java. The Java run time 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 Application Server use 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 Application Server use 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 J2C Bean 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.

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

Table 68. Bidirectional format attributes

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

Table 68. Bidirectional format attributes (continued)

Letter position	Purpose	Values	Description	Default setting
5	Numeric Shaping	H	National (Hindi)	N
		C	Contextual shaping	
		N	Numbers are not shaped (Nominal)	

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 the preceding table) 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 to Rational Application Developer for WebSphere Software 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 Rational Application Developer for WebSphere Software and WebSphere Application Server.

Managed connection properties

The following managed connection properties control bidirectional script data transformation.

- Username
- Password
- Environment
- Role

Activation specification properties

The following activation specification properties are enabled for bidirectional script data transformation:

- Auto acknowledge
- Guaranteed event delivery
- No wait
- Wait time

Notices

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

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

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Programming interface information

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

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

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

Warning:

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

Trademarks and service marks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol ([®] or [™]), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

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

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

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

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

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

Index

A

- activation specification properties
 - list of 74
 - setting in administrative console 37, 40
 - setting with external service wizard 74
- adapter
 - project, create 15
- Adapter for JD Edwards EnterpriseOne
 - administering 35
 - security 10
- application-specific information
 - business-object-level 49
 - operation-level 50
 - property-level 49
- AssuredOnceDelivery property 74
- authentication
 - description 10
 - J2C Bean wizard 10
 - run time 11
- authentication alias 11
- AutoAcknowledge property 74

B

- business objects 6, 49
 - application-specific information 49
 - generating business functions using J2C Bean wizard 20
 - generating XML Lists using J2C Bean wizard 26
 - naming conventions 51
 - supported operations 51

C

- clustered environment
 - deploying in 13
 - description 13
 - inbound processes 13
 - outbound processes 13
- compatibility matrix 1
- confidential data, disguising 9
- confidential tracing 9
- configuring
 - logging 43
 - tracing 43
- connection properties, J2C Bean wizard 19
- connector project 15
- custom properties
 - activation specification 37, 40
 - managed connection factory 36, 39
 - resource adapter 35, 38

D

- debugging
 - self-help resources 47
 - XAResourceNotAvailableException exception 47
- DeliveryType property 74
- deployment
 - options 11

E

- embedded adapter
 - activation specification properties, setting 37
 - considerations for using 12
 - description 11
 - managed connection factory properties, setting 36
 - resource adapter properties, setting 35
- enableHASupport property 13
- event store
 - overview 5
- EventTypeFilter property 74
- exceptions
 - XAResourceNotAvailableException 47
- external dependencies, adding 16
- external dependencies, editing 18
- external service
 - properties, connection 54, 67

F

- FFDC (first-failure data capture) 46
- files
 - SystemOut.log log file 45
 - trace.log trace file 45
- first-failure data capture (FFDC) 46

H

- hardware and software requirements 1
- hardware requirements 1
- high-availability environment
 - deploying in 13
 - description 13
 - inbound processes 13
 - outbound processes 13

I

- inbound configuration properties 65
- inbound processing
 - business graph 3
 - overview 3
- interaction specification property 65

J

- J2C Bean wizard
 - authentication in 10
 - overview 7
 - setting connection properties 19
- JAR files, adding external 16
- Java 2 security 11

L

- Log Analyzer 44
- Log and Trace Analyzer, support for 43
- log and trace files 43

- log files
 - changing file name 45
 - disabling 44
 - enabling 44
 - level of detail 44
 - location 45
- logging
 - configuring properties with administrative console 44

M

- managed (J2C) connection factory properties
 - list of 61
 - setting in administrative console 36, 39
- matrix, compatibility 1
- MaximumConnections property 74
- metadata 49
 - business-object-level 49
 - operation-level 50
 - property-level 49
- MinimumConnections property 74

N

- naming conventions for business objects 51
- NoWait property 74

O

- outbound configuration properties 52
- outbound processing 2
 - overview 2

P

- package files for adapters 44
- PollPeriod property 74
- PollQuantity property 74
- problem determination
 - self-help resources 47
 - XAResourceNotAvailableException exception 47
- properties
 - activation specification 37, 40
 - configuration properties
 - inbound 65
 - outbound 52
 - external service connection 54, 67
 - inbound configuration 65
 - managed (J2C) connection factory 36, 39
 - list of 61
 - outbound configuration 52
 - resource adapter 35, 38
 - list of 57, 70

R

- requirements, hardware and software 1
- resource adapter properties
 - list of 57, 70
 - setting in administrative console 35, 38
- Retry limit property 79
- RetryInterval property 74
- RetryLimit property 74
- runtime environment
 - authentication in 11

S

- security
 - disguising sensitive data 9
- security features, adapter 10
- security, Java 2 11
- self-help resources 47
- sensitive data, disguising 9
- setting connection properties 19
- software dependencies, adding external 16
- software dependencies, editing external 18
- software requirements 1
- stand-alone adapter
 - activation specification properties, setting 40
 - considerations for using 12
 - description 11
 - managed connection factory properties, setting 39
 - resource adapter properties, setting 38
- StopPollingOnError property 74
- support
 - overview 43
 - self-help resources 47
- supported operations 51
- SystemOut.log file 45

T

- technotes 1, 47
- trace files
 - changing file name 45
 - disabling 44
 - enabling 44
 - level of detail 44
 - location 45
- trace.log file 45
- tracing
 - configuring properties with administrative console 44
- troubleshooting
 - overview 43
 - self-help resources 47
 - XAResourceNotAvailableException exception 47

W

- WaitTime property 74
- WebSphere Extended Deployment 13

X

- XAResourceNotAvailableException 47
- XML Lists
 - generating using J2C Bean wizard 26



Printed in USA