





Note

Before using this information and the product it supports, read the information in "Notices" on page 83.

December 2008

This edition applies to version 6, release 2, modification 0 of IBM WebSphere Adapter for PeopleSoft Enterprise 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 PeopleSoft Enterprise 1

Hardware and software requirements	2
Technical overview	2
Outbound processing	3
Inbound processing	4
Business objects	5
Java data bindings	6
J2C Bean wizard	6
Standards Compliance	7

Chapter 2. Planning for adapter

implementation 9

Before you begin	9
Security	9
Support for protecting sensitive user data in log and trace files	9
User authentication	10
Deployment options	11
WebSphere Adapters in clustered environments	13

Chapter 3. Configuring the module for deployment. 15

Creating a custom event project in PeopleTools	15
Launching the J2C Bean wizard	16
Configuring the connector dependencies.	17
Setting connection properties for the J2C Bean wizard	17
Configuring the module for outbound processing	19
Selecting business objects and services	19
Configuring the selected objects	20
Setting deployment properties and generating artifacts.	21
Generating the EJB or JSP project	22
Configuring the module for inbound processing	24
Selecting business objects and services	24
Configuring the selected objects	25
Setting deployment properties and generating artifacts.	26

Chapter 4. Configuring the application on WebSphere Application Server 29

Changing configuration properties for embedded adapters	29
Setting resource adapter properties for embedded adapters	29
Setting managed (J2C) connection factory properties for embedded adapters	30
Setting activation specification properties for embedded adapters.	31

Changing configuration properties for stand-alone adapters	32
Setting resource adapter properties for stand-alone adapters	32
Setting managed (J2C) connection factory properties for stand-alone adapters	33
Setting activation specification properties for stand-alone adapters	34
Adding dependency libraries to the deployed resource adapter.	35
Standalone deployment	35
EAR deployment	35

Chapter 5. Troubleshooting and support 37

Support for the Log and Trace Analyzer	37
Configuring logging and tracing	37
Configuring logging properties	38
Changing the log and trace file names	39
First-failure data capture (FFDC) support	40
XAResourceNotAvailableException	41
Self-help resources	41

Chapter 6. Reference information 43

Business object information	43
Application-specific information	43
Supported operations	44
PeopleCode for a custom event project	45
Outbound configuration properties	49
Connection properties for the J2C Bean wizard	51
Resource adapter properties	54
Managed connection factory properties	57
Interaction specification properties.	61
Inbound configuration properties	62
Connection properties for the J2C Bean wizard	64
Resource adapter properties	66
Activation specification properties	70
Globalization	79
Globalization and bidirectional data transformation	79
Properties enabled for bidirectional data transformation	80

Notices 83

Programming interface information	85
Trademarks and service marks	85

Index 87

Chapter 1. Overview of WebSphere Adapter for PeopleSoft Enterprise

With adapter for PeopleSoft Enterprise, you can create integrated processes that exchange information with PeopleSoft Enterprise through a standard interface. This interface shields the client application from analyzing lower level details about implementation or the application or data structures used on the PeopleSoft Enterprise server.

With the adapter, a client application can send and receive requests from a PeopleSoft Enterprise server. For example, the application can send a request to query an employee record in an HR table, or it can receive events such as notification about an updated employee record from the server.

To make it possible to communicate in a standard way, WebSphere Adapter for PeopleSoft Enterprise 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 PeopleSoft Enterprise server, interact with each other. WebSphere Adapter for PeopleSoft Enterprise enables JCA-compliant application servers to connect to and interact with the PeopleSoft Enterprise server. Clients running on the JCA-compliant server can then communicate with the PeopleSoft Enterprise server in a standard way using Java Beans.

For example, consider that you are setting up an adapter using Rational® Application Developer for WebSphere Software tooling on WebSphere Application Server, and your company uses PeopleSoft Enterprise to coordinate most of its business operations. PeopleSoft Enterprise includes a business function that returns a list of employees in response to a range of skill codes. A client application might be able to use this function as part of an overall business process. For example, the technical support department within the company dispatches employees to customer sites to troubleshoot and repair equipment and, as part of that process, must obtain a list of employees with certain skills within a defined geographic region.

To use this function, however, the client application that the technical support department uses must understand the low-level implementation details and data structures of the PeopleSoft function. Information technology resources and time must link the client application and the PeopleSoft function.

With WebSphere Adapter for PeopleSoft Enterprise, you can automatically generate an interface to the PeopleSoft function to hide the lower-level details of the function. Depending on your plans for using the adapter, you can either embed it with the deployed application or install it as a stand-alone component that more than one application uses.

The embedded adapter is part of an application that is deployed to WebSphere Application Server. The application component interacts with the adapter instead of interacting with the PeopleSoft function.

The component that you generate with the J2C Bean wizard of Rational Application Developer for WebSphere Software uses a standard interface and

standard business objects, taking the business object sent by the client application and calling the PeopleSoft function. It then returns a business object to the client application.

The client application does not deal directly with the PeopleSoft function. Instead, the component calls the function and returns the results.

In the example, the client application that needed the list of employees sends a business object with the range of skill codes to the PeopleSoft adapter component. The client application, in return, receives the results (the list of employees) in the form of a standard business object. The client application does not need information about the way the function worked or the structure of the data. The adapter component performs all interactions with the PeopleSoft function.

Similarly, the client application might be alerted about a change to the data on the PeopleSoft Enterprise server (for example, a change to the skills set of a specific employee). You can generate an adapter component that polls for such events on the PeopleSoft Enterprise server and notifies client applications with the update.

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

The adapter supports the exchange of business data between the PeopleSoft Enterprise server and WebSphere Application Server by connecting to two layers of PeopleTools application programming interface classes that reveal the underlying business data for integration.

The adapter establishes bidirectional connectivity with PeopleSoft Enterprise server by connecting to two PeopleTools application programming interfaces:

1. The adapter accesses the primary API layer to create a session instance and to connect to the application server through the Jolt port.

2. The adapter then accesses the PeopleSoft Component Interface API, which reveals underlying business objects, logic, and functionality.

In PeopleSoft, a component is a set of pages grouped together for a business purpose (such as an employee profile), and a component interface is an API that provides access to a component from an external application. After the adapter connects to the component interface, the following entities are exposed to the adapter and available for integration:

- All business objects in the component interface definition
- PeopleCode methods associated with the underlying components
- Records, except searches and menu-specific processing options

Outbound processing

The adapter supports synchronous outbound request processing, so that when the client application sends a request in the form of a business object to the adapter, the adapter processes the request and returns a business object that represents the result of the operation to the client application.

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

1. The adapter extracts metadata that identifies the appropriate PeopleSoft component interface to access.
2. The adapter extracts the outbound operation to perform.
3. After accessing the component interface, the adapter sets the keys from values specified in the business objects. If key values are not generated, for example with a Create operation, the PeopleSoft application generates key fields.
4. After retrieving the PeopleSoft objects, the adapter instantiates an existing component interface to delete, retrieve, update, or create a component interface.
5. If update operations (Create, Update) are to be performed, the adapter populates the component interface with data from the WebSphere business object hierarchy. If Delete operations are required, the adapter populates the component interface only with StatusColumnName and value information.

Note: For Create and Update operations only, the adapter processes attributes in the order defined in the business object. For example, if a complex attribute is between two simple attributes, the adapter processes the simple attribute at the first position, then the complex attribute, and then the simple attribute. After the changes are made, the component interface is saved to commit the data to the PeopleSoft database.

Supported outbound operations

WebSphere Adapter for PeopleSoft Enterprise supports the following outbound operations:

- Apply Changes
- Create
- Delete
- Exists
- Retrieve
- Retrieve All
- Update

The following table describes each of these operations.

Table 1. Supported outbound operations

Operation	Description
ApplyChanges	Updates, creates, or deletes the business object based on the delta information contained in the change summary.
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 PeopleSoft component and maps component data onto the business object.
RetrieveAll	Retrieves multiple instances of the PeopleSoft component and maps component data onto the business object.
Update	Updates the corresponding PeopleSoft component with the incoming business object.

Inbound processing

The adapter supports asynchronous inbound processing. This means that the adapter polls the PeopleSoft Enterprise server at specified intervals for events. When the adapter detects an event, it converts the event data into a business object and sends it to the client application.

Event store

The event store is a table that holds events that represent data changes until the polling adapter can process them. The adapter uses the event store to keep track of event entities.

To use inbound processing, you must use PeopleTools Application Designer to create a custom project for event notification. The custom project uses two PeopleCode functions that determine the way future events are processed, and the custom project creates the event store the adapter needs for inbound processing. Each time a business object is created, updated, or deleted, the PeopleCode function used in the project and then added to the component interface inserts a new record in the event store, with the appropriate object name, keys, and status value.

During inbound processing, the adapter polls the events from the event store at configured poll intervals. In each poll call, a configured number of events are processed by the adapter. The order of event processing is based on the ascending order of priority and the ascending order of the event time stamp. The events with the Ready for poll (0) status are picked up for polling in each poll cycle. The adapter uses the object name and object key to retrieve the corresponding business object.

To ensure that an event is delivered only once to the target application, set the activation specification property `AssuredOnceDelivery` to true. On setting this to true, an `XID` (transaction ID) value is set for each event in the event store. After an event is obtained for processing, the `XID` value for that event is updated in the event store. The event is then delivered to its corresponding export, and its status is updated to show that event delivery was completed. If the application is stopped before the event can be delivered to the export or if the delivery fails, the

event might not be processed completely. In this case, the XID value represents in-progress status, and the XID column ensures that the event is reprocessed and sent to the export. After the database connection is re-established or the adapter starts again, the adapter checks for events in the event table that have a value in the XID column of Ready for Poll (0). The adapter processes these events first, and then polls the other events during the poll cycles.

The adapter uses special handling for events that have status code (99) to indicate that those events will occur in the future. During a poll cycle, when the adapter retrieves events with a future status, it compares the system time with the time stamp on each event. If the timestamp on the event is earlier than or equal to the system time, the adapter processes the event and changes the event status to Ready for Poll (0).

Note: If you want the adapter to process events with a future status in the present time, use the function `IBM_PUBLISH_EVENT` instead of `IBM_FUTURE_PUBLISH_EVENT`. Doing so means that the event is identified as Ready to Poll (0) instead of Future (99).

As events are retrieved from the event store and processed, the status of the event changes to reflect its state, as shown in the following table.

Table 2. Event status values

Status short name	Description	Event table value
Error processing event	An error occurred during event processing.	-1
Ready for poll	The event is ready to be picked up.	0
Success	The event has been delivered to the event manager.	1
Deleted	The event has been processed successfully and should be removed from the event store.	4
Future Events	These events should be processed at a future date.	99

Business objects

To send data or to obtain data from PeopleSoft Enterprise, the adapter uses 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.

How business objects are created

Use the J2C Bean wizard, which is launched from Rational Application Developer for WebSphere Software, to create business objects. The 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 business objects that are hierarchically structured. The top-level business object must have a one-to-one correspondence with the PeopleSoft component interface, and collections that occur within the top-level object are children of it. Information that the adapter uses to process the business object is stored in the application-specific information for the object and in each of its attributes.

The following table describes the attributes that compose a business object.

Attribute property	Description
Name	Name of the business object attribute.
Type	Type of the business object attribute. The adapter uses character mapping between PeopleSoft component property types and the generated business object attribute types. PeopleSoft component property types map to generated attribute types in the following manner: <ul style="list-style-type: none">• CHAR maps to attribute type String.• NUMBER maps to attribute type Integer.• SIGNED NUMBER maps to attribute type Integer.• DATE maps to attribute type String.• TIME maps to attribute type String.• DTTM maps to attribute type String.
Key	Application-specific information for the primary key. Child business objects also inherit keys from their parent business object.
Cardinality	Single cardinality for simple attributes; multiple cardinality for container attributes.

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 PeopleSoft 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.

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 PeopleSoft Enterprise server, discovers business objects and services (based on search criteria you provide), and generates business objects based on the services discovered.

Using Rational Application Developer for WebSphere Software, you establish a connection to the PeopleSoft Enterprise server to browse the metadata repository on the PeopleSoft Enterprise 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 business objects and services along with the adapter. You deploy this module on WebSphere Application Server.

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 IBM WebSphere Adapter for PeopleSoft Enterprise, you must plan for inbound and outbound processing and consider security and performance requirements.

Before you begin

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

To configure and deploy WebSphere Adapter for PeopleSoft Enterprise, 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 development tools you 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 used 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.
- PeopleSoft integration architecture, including the relationship between component interfaces and underlying components.
- The PeopleTools version that supports your applications. You should be able to use PeopleSoft Application Designer to access the object definitions that compose a business application, and to create and add projects to components.
- The programming environment required by PeopleSoft (PeopleCode). You should understand the role of PeopleCode within components, and know how to add PeopleCode to projects you create in Application Designer.

Security

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

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 PeopleSoft Enterprise 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 57 for information about this optional property.

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

- The contents of a business object
- The contents of the object key of the event record
- User name and password
- The URL used to connect to the PeopleSoft Enterprise 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

User authentication

The adapter supports several methods for supplying the user name and password that are needed to connect to the PeopleSoft Enterprise 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 PeopleSoft Enterprise 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 PeopleSoft Enterprise server to process outbound requests and inbound events.

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 PeopleSoft Enterprise server, although the schema name must be the same in both databases. For example, while developing and integrating an application that uses Adapter for PeopleSoft Enterprise, 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 PeopleSoft Enterprise 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 PeopleSoft Enterprise 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 PeopleSoft Enterprise (product number 5724-T76). 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 can handle both inbound and outbound processes.

High availability for inbound processes

Inbound processes are based on events triggered as a result of updates to data in the PeopleSoft Enterprise server. WebSphere Adapter for PeopleSoft Enterprise 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 PeopleSoft Enterprise 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.

Creating a custom event project in PeopleTools

To perform asynchronous inbound event processing, the adapter requires an event project in PeopleSoft. Use PeopleTools to create the custom event project.

About this task

If your environment requires inbound event support, you must use a custom event project in PeopleSoft. A sample event project, IBM_EVENT_V600, is provided with the adapter. You can either modify and use the sample project, or you can create your own project using PeopleTools. If you create your own project, use the following steps:

Procedure

1. Use PeopleTools Application Designer to create and name a new project.
2. Create the fields for the new project as described in the following table:

Field name	Field description
IBM_EVENT_ID	Numeric value that is retrieved from the IBM_FETCH_ID record. This value is a unique ID for the event.
IBM_OBJECT_NAME	Name of the corresponding business graph.
IBM_OBJECT_KEYS	The get key property names in the Component Interface followed by the key values in name-value pairs. This value is used to retrieve the component from the enterprise information system (EIS).
IBM_EVENT_STATUS	Status of the event. The status is 0 if the event is ready to be polled. The IBMPublishEvent function is called.
IBM_OBJECT_VERB	Verb that is set on the business object graph that contains the retrieved business object.
IBM_EVENT_DTTM	Date on which the event is created. For a future dated event, this value is the effective date.
IBM_NEXT_EVENT_ID	Field that has the latest event ID under the record IBM_FETCH_ID. The value is incremented for each event that is added to the IBM_EVENT_TBL table. The IBM_EVENT_ID field is populated within that table.
IBM_XID	The transaction ID which ensures providing of assured event delivery.

Field name	Field description
IBM_CONNECTOR_ID	The unique identifier for the adapter instance that will receive a specific event.

3. Create a record named IBM_EVENT_TBL and add to it all fields you have just created except IBM_NEXT_EVENT_ID.
4. Create a record named IBM_FETCH_ID and add to it only the IBM_NEXT_EVENT_ID field.
5. Open the IBM_FETCH_ID record, select the IBM_NEXT_EVENT_ID field, view the PeopleCode, and select **fieldformula**.
6. Copy the PeopleCode for custom event project, found under the Reference section of this documentation, to the project you are creating.
7. Create a page under your project that contains the fields of the IBM_EVENT_TBL record at level 0. The page can have any name.
8. Create a component under your project that contains the page you just created. The component can have any name.
9. Create a Component Interface against this component and give it any name. Confirm that you want to use the properties based on the underlying component definition.
10. Build the entire project, selecting all create options.
11. Test and confirm that the Component Interface works, using the Component Interface tester.
12. Generate the Java APIs for the Component Interface, and then add the generated classes to the adapter class path. For more information about building a PeopleTools project and testing the PeopleSoft Component Interface, refer to the PeopleSoft documentation.

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 PeopleSoft Enterprise server. For example, you need the name or IP address of the PeopleSoft Enterprise 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 PeopleSoft folder and select **IBM WebSphere Adapter for PeopleSoft Enterprise (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.

Configuring the connector dependencies

The J2C Bean wizard needs two files either copied or generated from the PeopleSoft Enterprise to be able to communicate with it. Use the J2C Bean wizard to specify the location of the JAR files. To add the required JAR files to your project, make sure the files are in the local machine and then use the Rational Application Developer for WebSphere Software to specify the location of the files.

Before you begin

You should be running the J2C Bean wizard in Rational Application Developer for WebSphere Software to perform this task.

About this task

To add the two required PeopleSoft dependency files, the PeopleSoft psjoa.jar and the component interface jar file, use the Connector Settings window of the J2C Bean wizard to specify the location of the files.

Procedure

1. From the Connector Settings window, specify the location of the files by clicking **Browse**
2. Click **Next**.

Results

The wizard has the files it needs to communicate with the PeopleSoft server.

Setting connection properties for the J2C Bean wizard

To set connection properties for the J2C Bean wizard so that it can access the PeopleSoft Enterprise server, specify information such as the user name and password you use to access the server as well as the name or IP address of the server.

Before you begin

Make sure you have successfully added the external dependency files (the psjoa.jar and associated files).

About this task

Specify the connection properties that the J2C Bean wizard needs to establish a connection to the PeopleSoft Enterprise server and discover functions or data.

To specify the connection properties, use the following procedure.

Procedure

1. In the Adapter Style window, select **Inbound** (if you are going to send data from the PeopleSoft Enterprise server) or **Outbound** (if you are going to send data to the PeopleSoft Enterprise server).
2. In the Discovery Configuration window, specify the configuration properties:
 - a. In the **Host name** field, type the name (or IP address) of your PeopleSoft Enterprise server.
 - b. In the **Port number** field, type the port number the adapter uses to access the PeopleSoft Enterprise server
 - c. Type the name and password you use to access the PeopleSoft Enterprise server.
The password is case-sensitive.
 - d. Click **Browse** and select the **Component Interface JAR file** that you added to the project as a dependency file.
 - e. Optionally, type a prefix to prepend to all objects associated with this module.
A prefix allows you to differentiate between objects generated against the same PeopleSoft component interface.
 - f. Optionally, select the **Prompt for additional configuration settings when adding business object** check box if you want to configure additional meta data when you generate objects. Use this option if you plan to use Delete and RetrieveAll operations and have the effective-dated fields and generated keys used in the operation.
3. To set additional advanced properties, click **Advanced**. The related properties are displayed.
4. 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.
5. 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.

6. Click **Next**.

Results

The J2C Bean wizard contacts the PeopleSoft Enterprise server, using the information you provided (such as user name and password) to log in. The Object Discovery and Selection window is displayed.

Specify search criteria that the J2C Bean wizard uses to discover functions or data on the PeopleSoft Enterprise 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 PeopleSoft Enterprise server, and to generate business object definitions and related artifacts.

Selecting business objects and services

To specify the objects you want to import, you use the J2C Bean wizard.

Before you begin

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

About this task

To find and select one or more objects, use the following procedure.

Procedure

1. In the Object Discovery and Selection window, click the object you want to work with, and then click the arrow to move it to the selected set.
To see children of an object, click the plus sign.
2. In the Configuration Parameters window, specify configuration properties for the object. (This window is available only if you select the **Prompt for additional configuration settings when adding a business object** check box on the Discovery Configuration window.)
 - a. Select the field name to use for performing logical deletes.
 - b. In the **Value used to indicate a deleted object** field, type the status value used to indicate that an object is no longer active.
 - c. Optional: If you want to use alternate keys during RetrieveAll operations, click **Add** and select the key fields you want to use.
3. Optional: To set additional parameters, click **Advanced**.
 - a. At the **Preserve objects in parent update operation** box, click **Add** and select the child objects to which this applies.
 - b. At the **Field that represents generated keys in PeopleSoft** box, click **Add** and select the fields to which this applies.
 - c. Click the **Effective date configuration** section, and select from the following effective date options:

- **Retrieve only current effective dated item:** The corresponding ASI is `GetCurrentItem`.
- **Insert at oldest effective date position:** The corresponding ASI is `InsAtOldestEffDtPos`.
- **Insert at current effective date position:** The corresponding ASI is `InsAtCurrentEffDtPos`.
- **Ignore effective sequence number:** The corresponding ASI is `IgnoreEffectiveSequence`.
- **Field that represents effective date value:** The corresponding ASI is `EffectiveDate`.
- **Field that represents effective date sequence:** The corresponding ASI is `EffectiveSequence`.

For more information about the effective date options, see “Application-specific information” on page 43 (ASI).

4. Click **OK** to add the object to the list of objects to be imported.
5. When you have selected all objects, click **Next**.

Results

You have selected one or more objects to work with.

What to do next

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

Configuring the selected objects

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

Before you begin

Make sure you have selected and imported the object.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Configure Composite Properties window, enter the following information.
 - a. Add or remove operations for the object that will be added to the service interface.
 - b. In the **Maximum number of records for RetrieveAll operations** , specify the maximum number of records to retrieve when processing. The default value is 100.
 - c. If you do not want to specify a namespace value, leave the default value for the **Business object namespace field**.
 - d. Optional: To indicate where the business object information should be stored, type the path to the location in the **Folder** field.
2. Click **Next**.

Results

The J2C Bean Creation and Deployment Configuration window is displayed.

What to do next

Generate a 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 PeopleSoft adapter RAR dependency such as CWYES_PeopleSoft.
 - 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, MyAdapterOutboundWEB EAR.

- e. Click **Next**.
- f. In the Java EE Module dependencies area of the Enterprise Application window, select the PeopleSoft adapter RAR dependency such as CWYES_PeopleSoft.
- 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.

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 PeopleSoft Enterprise server, and to generate business object definitions and related artifacts.

Selecting business objects and services

To specify which objects you want to process, you use the J2C Bean wizard.

Before you begin

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

About this task

To find and select one or more objects, use the following procedure.

Procedure

1. In the Object Discovery and Selection window, click or expand the object you want to work with.
2. In the Configuration Parameters, specify configuration properties for the object. (This window is available only if you select the **Prompt for additional configuration settings when adding a business object** check box on the Discovery Configuration window.)
 - a. Select the field name to use for performing logical deletes.
 - b. In the **Value used to indicate a deleted object** field, type the status value used to indicate that an object is no longer active.
 - c. Optional: If you want to use alternate keys during RetrieveAll operations, click **Add** and select the key fields you want to use.
3. Optional: To set additional parameters, click **Advanced**.

- a. At the **Preserve objects in parent update operation** box, click **Add** and select the child objects to which this applies.
- b. At the **Fields that represents generated keys in PeopleSoft** box, click **Add** and select the fields to which this applies.
- c. Click the **Effective date configuration** section, and select from the following effective date options:
 - **Retrieve only current effective dated item** : The corresponding ASI is `GetCurrentItem`.
 - **Insert at oldest effective date position**: The corresponding ASI is `InsAtOldestEffDtPos`.
 - **Insert at current effective date position**: The corresponding ASI is `InsAtCurrentEffDtPos`.
 - **Ignore effective sequence number**: The corresponding ASI is `IgnoreEffectiveSequence`.
 - **Field that represents effective date value**: The corresponding ASI is `EffectiveDate`.
 - **Field that represents effective date sequence**: The corresponding ASI is `EffectiveSequence`.

For more information about the effective date options, see “Application-specific information” on page 43 (ASI).

4. Click **OK**.
5. Click **Next**.

Results

You have selected one or more objects to work with.

What to do next

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

Configuring the selected objects

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

Before you begin

Make sure you have selected and imported the object.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Configure Composite Properties window, enter the following information.
 - a. Add or remove operations for the object that will be added to the service interface.
 - b. If you do not want to specify a namespace value, leave the default value for the **Business object namespace field**.

- c. Type a directory location to store the generated business objects.
2. Click Next.

Results

The J2C Bean Creation and Deployment Configuration window is displayed.

What to do next

Generate a 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 **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), managed connection factory properties (used for outbound processing), and activation specification properties (used for inbound 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 PeopleSoft Enterprise**.
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 54 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 PeopleSoft Enterprise 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 PeopleSoft Enterprise**.
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 PeopleSoft Enterprise. 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 57 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 PeopleSoft Enterprise**.
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 70 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. If the adapter will be used for inbound operations, you create an activation specification 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 PeopleSoft Enterprise**.
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 PeopleSoft Enterprise 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 PeopleSoft Enterprise**.
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 PeopleSoft Enterprise. 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 57 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 PeopleSoft Enterprise**.
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 PSFTRA plus the value of the adapter ID property. For example, if the adapter ID property is set to 001, the component ID is PSFTRA001.

If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for PeopleSoft Enterprise to 001 and 002. The component IDs for those instances, PSFTRA001 and PSFTRA002, 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 PSFTRAIInstanc.

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 54.

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 PeopleSoft Enterprise only, select the **com.ibm.j2ca.peoplesoft.*** 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 PeopleSoft Enterprise.
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 PeopleSoft Enterprise, 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 PeopleSoft Enterprise 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

A business object is a structure that contains application-specific information about how the adapter should process the business object as well as the operation to be performed on the business object. The name of the business object is generated by the J2C Bean wizard in accordance with the naming convention for the adapter.

Application-specific information

Application-specific information (ASI) is metadata that specifies how to process business objects for adapter for PeopleSoft Enterprise.

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 PeopleSoft Enterprise uses application-specific information (ASI) to create queries for Create, Retrieve, Update, and Delete operations.

Metadata at the business-object-level

Metadata at the Business-object-level defines the top-level wrapper of the business object. The following table describes the elements of business-object metadata .

Table 3. Metadata for business objects

Metadata element	Description
ObjectName	The corresponding object in the PeopleSoft application; in other words, the component interface name as defined in PeopleTools. Set this for the top-level business object only.
GetCurrentItem	When set to true, only effective-dated items are retrieved during an operation.
IgnoreEffectiveSequence	When set to true, the effective sequence number is ignored and the record is inserted either at the lowest index, which has the current effective date or at the highest index, which has the oldest effective date. If you set this parameter, you must also set either <code>InsAtCurrentEffDtPos</code> or <code>InsAtOldestEffDtPos</code> . This parameter is used only for the effective dated items.
InsAtOldestEffDtPos	When set to true and an incorrect index (negative number) is returned by the PeopleTools API, <code>getEffectiveItemNum()</code> , the adapter inserts the row at the highest index, which has the oldest effective date. If you set this parameter, do not set <code>InsAtCurrentEffDtPos</code> .
InstAtCurrentEffDtPos	When set to true and an incorrect index (negative number) is returned by the PeopleTools API, <code>getEffectiveItemNum()</code> , the adapter inserts the row at the highest index (zero), which has the current effective date. If you set this parameter, do not set <code>InsAtOldestEffDtPos</code> .
StatusColumnName	A property used to perform deletes. Set this for the top-level business object only.

Table 3. Metadata for business objects (continued)

Metadata element	Description
StatusValue	Identifies a business object that is inactive or that has been deleted. Set this for the top-level business object only.

Metadata at the property-level

Metadata at the property level defines the property of a business object. The property can be a simple string type, or it can be a complex property type that represents child objects or an array of child objects. The following table describes the metadata elements of a complex property (a child) or a structure or table property (an array of child objects).

Table 4. Metadata for Property

Metadata element	Description
PrimaryKey	GetKey/CreateKey attribute in the component interface or a key in the child collection.
FindKey	FindKey attribute in the component interface.
EffectiveDate	EffectiveDate attribute in the component interface.
EffectiveSequence	EffectiveSequence attribute in the component interface.
KeepRelations	Affects the processing of child objects when the adapter performs an update operation. If this parameter is enabled and the adapter finds a missing child object of a parent during the operation, the child object is not deleted and the relationship between the objects is preserved.
UID	Determines whether the adapter will set values for key attributes within the business object or not. When this is set to true, the adapter does not set values for these keys but allows PeopleCode to automatically generate values for the keys.
Getter	The name of the get method used to read values for the property.
Setter	The name of the set method used to set values for the property.

Supported operations

An operation is the action that an adapter can perform on the PeopleSoft Enterprise 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 table defines the operations that the adapter supports.

Table 5. Supported operations of business objects

Operation	Definition
Create	The adapter accesses the PeopleSoft component and retrieves values from the attributes that have the primary key application-specific information set. It then instantiates the corresponding component interface (using the value provided for the ObjectName application-specific information). It sets the attribute values on the corresponding Create Keys in the component interface. An empty Component Interface is created, and the adapter maps all the business object data to the created component interface. When mapping the data, the adapter sends all data for simple attributes in the hierarchy, and it creates items that match each of the child objects in the hierarchy as well, including effective-dated and effective-sequenced child records.

Table 5. Supported operations of business objects (continued)

Operation	Definition
Retrieve	The adapter accesses the PeopleSoft component and retrieves values from the attributes that have the primary key application-specific information set. It then instantiates the corresponding component interface using the value provided for the ObjectName application-specific information. It sets the attribute values on the corresponding Get Keys in the component interface. The adapter then maps the component data onto the business object hierarchy. Child objects are included in the data mapping.
RetrieveAll	This functions as the Retrieve operation, except that it allows retrieval of multiple instances of the same PeopleSoft component.
Update	The adapter retrieves an object from PeopleSoft and compares it to the target business object. It processes the objects as follows: When the comparison reveals extra child objects in PeopleSoft, the children are deleted. When the comparison reveals missing children in PeopleSoft, the children are created. When the comparison reveals child objects that have been updated in PeopleSoft, they are updated.
Exists	The adapter processes an exist operation in the same way that it processes a retrieve operation, except that it does not populate the business object with retrieved data. It simply checks for the existence of an object in PeopleSoft.
Delete	Based on the values set for the application-specific metadata elements StatusColumnName and StatusValue, the adapter updates a business object to inactive. A delete operation can only be performed on a top level object. PeopleSoft does not allow an object to be physically deleted, so the inactive object remains in the PeopleSoft database.
Apply Changes	Apply changes Updates the PeopleSoft component based on the operation that was performed on it. The supported operations are create, update, and delete.

PeopleCode for a custom event project

Two PeopleCode functions are required to support inbound processing. If you create a custom event project in PeopleTools for inbound support, add the PeopleCode functions to the project.

The following PeopleCode contains the IBMPublishEvent and IBMPublishFutureDatedEvent functions that are used to publish events to the event table. Calls to these functions are made from the SavePostChange PeopleCode function in the PeopleSoft component of interest.

```

/* IBM event notification */
Component string &KEYSTRING;
Component string &KEYNAME;
Component array of string &KEYARRAY;
Component string &KEYDELIM;
Component string &IBMVERB;
Local Record &IBMREC;

Function IBMPublishFutureDatedEvent(&BO, &KEYS, &EFFDATE)
; /* == create a new record object for cw_event_tbl == */
&IBMREC = CreateRecord(Record.IBM_EVENT_TBL);
/* ===== KEYS ===== */
/* composing keys and values in name value format */
&KEYSTRING = "";
&KEYDELIM = ".";
&KEYARRAY = Split(&KEYS, &KEYDELIM);
&LEN = &KEYARRAY.Len;
For &I = 1 To &LEN;
/* get keys and values */
/* get rid of record name */
&POS1 = Find(".", &KEYARRAY [&I]);
&L1 = Len(&KEYARRAY [&I]);
&POS2 = &L1 - &POS1;
&KEYNAME = Right(&KEYARRAY [&I], &POS2);
/****The code below will remove special characters and****/
/****adjust the characters' case to ensure it is same as the****/
/****attribute name in the business object definition****/
/****Start****/
&LLen = Len(&KEYNAME);

```

```

&sOrigString = &KEYNAME;
&sNewString = "";
&lCtr2 = 1;
&isSpecialChar = "true";
For &lCtr = 1 To &Len;
    &sChar = Substring(&sOrigString, &lCtr, 1);
    If (&sChar = "A" Or
        &sChar = "a" Or
        &sChar = "B" Or
        &sChar = "b" Or
        &sChar = "C" Or
        &sChar = "c" Or
        &sChar = "D" Or
        &sChar = "d" Or
        &sChar = "E" Or
        &sChar = "e" Or
        &sChar = "F" Or
        &sChar = "f" Or
        &sChar = "G" Or
        &sChar = "g" Or
        &sChar = "H" Or
        &sChar = "h" Or
        &sChar = "I" Or
        &sChar = "i" Or
        &sChar = "J" Or
        &sChar = "j" Or
        &sChar = "K" Or
        &sChar = "k" Or
        &sChar = "L" Or
        &sChar = "l" Or
        &sChar = "M" Or
        &sChar = "m" Or
        &sChar = "N" Or
        &sChar = "n" Or
        &sChar = "O" Or
        &sChar = "o" Or
        &sChar = "P" Or
        &sChar = "p" Or
        &sChar = "Q" Or
        &sChar = "q" Or
        &sChar = "R" Or
        &sChar = "r" Or
        &sChar = "S" Or
        &sChar = "s" Or
        &sChar = "T" Or
        &sChar = "t" Or
        &sChar = "U" Or
        &sChar = "u" Or
        &sChar = "V" Or
        &sChar = "v" Or
        &sChar = "W" Or
        &sChar = "w" Or
        &sChar = "X" Or
        &sChar = "x" Or
        &sChar = "Y" Or
        &sChar = "y" Or
        &sChar = "Z" Or
        &sChar = "z" Or
        &sChar = "1" Or
        &sChar = "2" Or
        &sChar = "3" Or
        &sChar = "4" Or
        &sChar = "5" Or
        &sChar = "6" Or
        &sChar = "7" Or
        &sChar = "8" Or
        &sChar = "9" Or
        &sChar = "0") Then
        If (&isSpecialChar = "true") Then
            &sNewString = &sNewString | Upper(&sChar);
            &isSpecialChar = "false";
        Else
            &sNewString = &sNewString | Lower(&sChar);
        End-If;
    Else
        &isSpecialChar = "true";
    End-If;
End-For;
&KEYNAME = &sNewString;
/*****End*****/
&KEYSTRING = &KEYSTRING | &KEYNAME | "=" | @&KEYARRAY [&I] | &KEYDELIM
End-For;
&KEYSTRING = RTrim(&KEYSTRING, "=");
&IBMREC.IBM_OBJECT_KEYS.Value = &KEYSTRING;
/***** VERB *****/
/* verb determination uses variable &IBMVERB */
Evaluate %Mode
When = "A"
    &IBMVERB = "Create";
    Break;
When = "U"

```

```

        &IBMVERB = "Update";
        Break;
    When = "L"
        &IBMVERB = "Update";
        Break;
    When = "C"
        &IBMVERB = "Update";
        Break;
    When-Other
        &IBMVERB = "Retrieve";
    End-Evaluate;
    &IBMREC.IBM_OBJECT_VERB.Value = &IBMVERB;
    /* ===== EVENT_ID GEN ===== */
    /* create event_id */
    &NEWNUM = GetNextNumber(IBM_FETCH_ID.IBM_NEXT_EVENT_ID, 99999);
    /* only use newnum if no error generating next number */
    If &NEWNUM > 0 Then
        &IBMREC.IBM_EVENT_ID.Value = &NEWNUM;
    Else
        &IBMREC.IBM_EVENT_ID.Value = %Datetime;
    End-If; /*Support for Future Effective Date - The adapter will poll such events when the date arrives*/
    If &EFFDATE > %Datetime Then
        &IBMREC.IBM_EVENT_DTTM.Value = &EFFDATE;
        &IBMREC.IBM_EVENT_STATUS.Value = "99";
    Else
        &IBMREC.IBM_EVENT_DTTM.Value = %Datetime;
        &IBMREC.IBM_EVENT_STATUS.Value = "0";
    End-If; /*===== INSERT EVENT INTO IBM_EVENT_TBL =====*/
    /* insert row into table using record object*/
    &IBMREC.IBM_OBJECT_NAME.Value = &B0;
    &IBMREC.Insert();
End-Function;

```

```

Function IBMPublishEvent(&B0, &KEYS);
    /* == create a new record object for cw_event_tbl == */
    &IBMREC = CreateRecord(Record.IBM_EVENT_TBL);

```

```

    /* ===== KEYS ===== */
    /* composing keys and values in name value format */
    &KEYSTRING = "";
    &KEYDELIM = ":";
    &KEYARRAY = Split(&KEYS, &KEYDELIM);
    &LEN = &KEYARRAY.Len;

```

```

For &I = 1 To &LEN;
    /* get keys and values */
    /* get rid of record name */
    &POS1 = Find(".", &KEYARRAY [&I]);
    &L1 = Len(&KEYARRAY [&I]);
    &POS2 = &L1 - &POS1;
    &KEYNAME = Right(&KEYARRAY [&I], &POS2);

```

```

    /*****The code below will remove special characters and
    /*****adjust the characters' case to ensure it is same as the
    /*****attribute name in the business object definition****/
    /*****Start****/

```

```

    &lLen = Len(&KEYNAME);
    &sOrigString = &KEYNAME;
    &sNewString = "";
    &lCtr2 = 1;
    &isSpecialChar = "true";
    For &lCtr = 1 To &lLen;
        &sChar = Substring(&sOrigString, &lCtr, 1);
        If (&sChar = "A" Or
            &sChar = "a" Or
            &sChar = "B" Or
            &sChar = "b" Or
            &sChar = "C" Or
            &sChar = "c" Or
            &sChar = "D" Or
            &sChar = "d" Or
            &sChar = "E" Or
            &sChar = "e" Or

```

```

&sChar = "F" Or
&sChar = "f" Or
&sChar = "G" Or
&sChar = "g" Or
&sChar = "H" Or
&sChar = "h" Or
&sChar = "I" Or
&sChar = "i" Or
&sChar = "J" Or
&sChar = "j" Or
&sChar = "K" Or
&sChar = "k" Or
&sChar = "L" Or
&sChar = "l" Or
&sChar = "M" Or
&sChar = "m" Or
&sChar = "N" Or
&sChar = "n" Or
&sChar = "O" Or
&sChar = "o" Or
&sChar = "P" Or
&sChar = "p" Or
&sChar = "Q" Or
&sChar = "q" Or
&sChar = "R" Or
&sChar = "r" Or
&sChar = "S" Or
&sChar = "s" Or
&sChar = "T" Or
&sChar = "t" Or
&sChar = "U" Or
&sChar = "u" Or
&sChar = "V" Or
&sChar = "v" Or
&sChar = "W" Or
&sChar = "w" Or
&sChar = "X" Or
&sChar = "x" Or
&sChar = "Y" Or
&sChar = "y" Or
&sChar = "Z" Or
&sChar = "z" Or
&sChar = "1" Or
&sChar = "2" Or
&sChar = "3" Or
&sChar = "4" Or
&sChar = "5" Or
&sChar = "6" Or
&sChar = "7" Or
&sChar = "8" Or
&sChar = "9" Or
&sChar = "0") Then
If (&isSpecialChar = "true") Then
&sNewString = &sNewString | Upper(&sChar);
&isSpecialChar = "false";
Else
&sNewString = &sNewString | Lower(&sChar);
End-If;
Else
&isSpecialChar = "true";
End-If;
End-For;
&KEYNAME = &sNewString;

```

```

        /*****End*****/
        &KEYSTRING = &KEYSTRING | &KEYNAME | "=" | @&KEYARRAY [&I] | &KEYDELIM
End-For;
&KEYSTRING = RTrim(&KEYSTRING, ":");

&IBMREC.IBM_OBJECT_KEYS.Value = &KEYSTRING;

/*===== VERB =====*/
/* verb determination uses variable &IBMVERB */
Evaluate %Mode
When = "A"
    &IBMVERB = "Create";
    Break;
When = "U"
    &IBMVERB = "Update";
    Break;
When = "L"
    &IBMVERB = "Update";
    Break;
When = "C"
    &IBMVERB = "Update";
    Break;
When-Other
    &IBMVERB = "Retrieve";
End-Evaluate;

&IBMREC.IBM_OBJECT_VERB.Value = &IBMVERB;

/* ===== EVENT_ID GEN ===== */
/* create event_id */

&NEWNUM = GetNextNumber(IBM_FETCH_ID.IBM_NEXT_EVENT_ID, 99999);

/* only use newnum if no error generating next number */

If &NEWNUM > 0 Then
    &IBMREC.IBM_EVENT_ID.Value = &NEWNUM;
Else
    &IBMREC.IBM_EVENT_ID.Value = %Datetime;
End-If;

&IBMREC.IBM_EVENT_DTTM.Value = %Datetime;

/* ===== EVENT_STATUS =====*/
/* Validate and set event status &IBMSTATUS - list values if date is ok*/
&IBMREC.IBM_EVENT_STATUS.Value = "0";

/*===== INSERT EVENT INTO IBM_EVENT_TBL =====*/
/* insert row into table using record object*/

&IBMREC.IBM_OBJECT_NAME.Value = &B0;

&IBMREC.Insert();

End-Function;

```

Outbound configuration properties

WebSphere Adapter for PeopleSoft Enterprise 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 PeopleSoft Enterprise 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

Row	Explanation
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>

Connection properties for the J2C Bean wizard

Connection properties for the J2C Bean wizard are used to establish a connection between the J2C Bean wizard and the application from which the wizard will obtain metadata. These properties specify such things as connection configuration, bidirectional (bidi) properties, and logging options.

Note: 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.

The J2C Java Bean connection properties and their purpose are described in the following table. A complete description of each property is provided in the sections that follow the table.

Table 6. J2C Java Bean connection properties

Property name	Description
"Component interface JAR file" on page 52	Specifies the PeopleSoft Enterprise component interface used by the adapter to establish a connection to the PeopleSoft components that are targets of integration transactions.
"Host name " on page 52	Specifies the name or address of the server that hosts PeopleSoft Enterprise.
"Password " on page 52	Specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.

Table 6. J2C Java Bean connection properties (continued)

“Port number” on page 53	The port number at which PeopleSoft Enterprise is configured to listen for client requests.
“Prefix for business object names” on page 53	Specifies a prefix to be added to generated business objects.
“User name” on page 53	The name of the user account the adapter uses on the PeopleSoft Enterprise server.

Component interface JAR file

This property specifies the PeopleSoft Enterprise component interface that the adapter uses to establish a connection to the PeopleSoft components that are targets of integration transactions.

Table 7. Component interface JAR file details

Required	Yes
Default	No default
Property type	String
Usage	The name of the JAR file that the adapter uses to connect to the PeopleSoft Enterprise components of interest.
Example	CWYES_PeopleSoft\connectorModule\WbiEvent.jar
Globalized	No
Bidi supported	No

Host name

This property specifies the name or address of the server that hosts PeopleSoft Enterprise.

Table 8. Host name details

Required	Yes
Default	No default value
Property type	String
Usage	Identifies the server, either by name or IP address, that hosts PeopleSoft Enterprise.
Example	9.26.248.202
Globalized	No
Bidi supported	No

Password

This property specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.

Table 9. Password details

Required	Yes
Default	No default value
Property type	String

Table 9. Password details (continued)

Usage	The restrictions (case, length, and character) are determined by the PeopleSoft Enterprise version.
Globalized	Yes
Bidi supported	Yes

Port number

The port number at which PeopleSoft Enterprise is configured to listen for client requests.

Table 10. Port number details

Required	Yes
Default	The port number entered when you run the J2C Bean wizard.
Property type	Integer
Example	9000
Globalized	No
Bidi supported	No

Prefix for business object names

This property specifies a prefix to be added to generated business objects.

Table 11. Prefix details

Required	No
Default	No default
Property type	String
Usage	This property allows you to distinguish between different business objects generated against the same PeopleSoft component interface.
Example	If you used IB as a prefix, all business objects generated by this service are named using this prefix.
Globalized	Yes
Bidi supported	No

User name

This property specifies the name of the user account the adapter uses on the PeopleSoft Enterprise server.

Table 12. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the PeopleSoft Enterprise version.
Example	DV1
Globalized	Yes

Table 12. User name details (continued)

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 table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table.

Table 13. Resource adapter properties for the Adapter for PeopleSoft Enterprise

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
(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 14. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 14. Adapter ID details (continued)

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, PSFTRA, 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 PSFTRA001.</p> <p>If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter.</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 PeopleSoft Enterprise to 001 and 002. The component IDs for those instances, PSFTRA001 and PSFTRA002, 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 PSFTRAIinstanc.</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 15. 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
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 16. Log file maximum size details

Required	No
Default	0
Property type	Integer
Usage	When the log file reaches its maximum size, the adapter starts 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 17. Log file name details

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

Log number of files (LogNumberOfFiles)

This property specifies the number of log files.

Table 18. 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 19. Trace file maximum size details

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

Table 19. Trace file maximum size details (continued)

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 20. 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 21. 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 with the PeopleSoft Enterprise server.

The following table lists and describes the managed connection factory properties for outbound communication. You set the managed connection factory properties using either the J2C Bean wizard or the administrative console (after deployment).

A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property details tables in the sections that follow, see Guide to understanding property details.

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

Table 22. Managed connection factory properties

In the wizard	In the administrative console	Description
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Component interface for testing failed connection	PingCompInterface	Specifies the component interface the adapter uses to validate a connection to the PeopleSoft Enterprise server.
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of X's instead of user data in log and trace files
Host name	HostName	Specifies the name or address of the server that hosts PeopleSoft Enterprise.
"Language (Language)" on page 60	Language	Specifies the language code the adapter uses to log on to the PeopleSoft Enterprise server.
"Password (Password)" on page 61	Password	Specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.
"Port number (Port)" on page 61	Port	The port number the adapter uses to access the PeopleSoft Enterprise server.
"User name (UserName)" on page 61	UserName	The name of the user account the adapter uses on the PeopleSoft Enterprise server.

Adapter ID (AdapterID)

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

Table 23. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 23. Adapter ID details (continued)

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, PSFTRA, 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 PSFTRA001.</p> <p>If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter.</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 PeopleSoft Enterprise to 001 and 002. The component IDs for those instances, PSFTRA001 and PSFTRA002, 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 PSFTRAIinstanc.</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

Component interface for testing failed connection (PingComplInterface)

This property specifies the name of the PeopleSoft Enterprise component interface that the adapter uses to validate a connection to the PeopleSoft Enterprise server.

Table 24. Component interface for testing failed connection details

Required	Yes
Default	The name of the first component interface in the list.
Property type	String
Usage	Specify a component interface name that already exists within your PeopleSoft Enterprise applications.
Example	WBI_CUSTOMER_CI
Globalized	No
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 25. Disguise user data as "XXX" in log and trace files details

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

Table 25. Disguise user data as "XXX" in log and trace files details (continued)

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

Host name (HostName)

This property specifies the name or address of the server that hosts PeopleSoft Enterprise.

Table 26. Host name details

Required	Yes
Default	No default value
Property type	String
Usage	Identifies, either by name or IP address, the server that hosts PeopleSoft Enterprise.
Example	9.26.248.202
Globalized	No
Bidi supported	No

Language (Language)

This property specifies the language code the adapter uses to log on to the PeopleSoft Enterprise server.

Table 27. Language details

Required	Yes
Default	The default value for the Language property is based on the system locale.
Property type	String
Usage	Each of the supported languages is preceded by a three character language code. The language itself is presented in parentheses.
Example	If the system locale is English, the value for this property is ENG (English).
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.

Table 28. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the version of PeopleSoft Enterprise.
Globalized	No
Bidi supported	No

Port number (Port)

This property specifies the port number the adapter uses to access the PeopleSoft Enterprise server.

Table 29. Port number details

Required	Yes
Default	The port number entered when you use the J2C Bean wizard to discover objects and services.
Property type	Integer
Example	9000
Globalized	No
Bidi supported	No

User name (UserName)

This property specifies the name of the user account the adapter uses on the PeopleSoft Enterprise server.

Table 30. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the PeopleSoft Enterprise version.
Example	DV1
Globalized	No
Bidi supported	No

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. For example, you might increase the value of the interaction specification 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.

Table 31 lists and describes the interaction specification property that you set. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 50.

Table 31. Interaction specification property for the Adapter for PeopleSoft Enterprise

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

Maximum number of records for RetrieveAll operation (MaxRecords)

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

Table 32. Maximum number of records for RetrieveAll operation details

Required	Yes
Default	100
Usage	If the number of hits in PeopleSoft Enterprise exceeds the value of the Maximum number of records for RetrieveAll operation property, the adapter returns an error. The adapter uses this property to help avoid out-of-memory issues.
Property type	Integer
Globalized	No
Bidi supported	No

Inbound configuration properties

WebSphere Adapter for PeopleSoft Enterprise 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 PeopleSoft Enterprise 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>

Connection properties for the J2C Bean wizard

Connection properties for the J2C Bean wizard are used to establish a connection between the J2C Bean wizard and the application from which the wizard will obtain metadata. These properties specify such things as connection configuration, bidirectional (bidi) properties, and logging options.

Note: 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.

The J2C Java Bean connection properties and their purpose are described in the following table. A complete description of each property is provided in the sections that follow the table.

Table 33. J2C Java Bean connection properties

Property name	Description
"Component interface JAR file"	Specifies the PeopleSoft Enterprise component interface used by the adapter to establish a connection to the PeopleSoft components that are targets of integration transactions.
"Host name "	Specifies the name or address of the server that hosts PeopleSoft Enterprise.
"Password " on page 65	Specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.
"Port number" on page 65	The port number at which PeopleSoft Enterprise is configured to listen for client requests.
"Prefix for business object names" on page 65	Specifies a prefix to be added to generated business objects.
"User name" on page 66	The name of the user account the adapter uses on the PeopleSoft Enterprise server.

Component interface JAR file

This property specifies the PeopleSoft Enterprise component interface that the adapter uses to establish a connection to the PeopleSoft components that are targets of integration transactions.

Table 34. Component interface JAR file details

Required	Yes
Default	No default
Property type	String
Usage	The name of the JAR file that the adapter uses to connect to the PeopleSoft Enterprise components of interest.
Example	CWYES_PeopleSoft\connectorModule\WbiEvent.jar
Globalized	No
Bidi supported	No

Host name

This property specifies the name or address of the server that hosts PeopleSoft Enterprise.

Table 35. Host name details

Required	Yes
Default	No default value
Property type	String
Usage	Identifies the server, either by name or IP address, that hosts PeopleSoft Enterprise.
Example	9.26.248.202
Globalized	No
Bidi supported	No

Password

This property specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.

Table 36. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the PeopleSoft Enterprise version.
Globalized	Yes
Bidi supported	Yes

Port number

The port number at which PeopleSoft Enterprise is configured to listen for client requests.

Table 37. Port number details

Required	Yes
Default	The port number entered when you run the J2C Bean wizard.
Property type	Integer
Example	9000
Globalized	No
Bidi supported	No

Prefix for business object names

This property specifies a prefix to be added to generated business objects.

Table 38. Prefix details

Required	No
Default	No default
Property type	String
Usage	This property allows you to distinguish between different business objects generated against the same PeopleSoft component interface.

Table 38. Prefix details (continued)

Example	If you used IB as a prefix, all business objects generated by this service are named using this prefix.
Globalized	Yes
Bidi supported	No

User name

This property specifies the name of the user account the adapter uses on the PeopleSoft Enterprise server.

Table 39. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the PeopleSoft Enterprise version.
Example	DV1
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 table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table.

Table 40. Resource adapter properties for the Adapter for PeopleSoft Enterprise

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

Table 40. Resource adapter properties for the Adapter for PeopleSoft Enterprise (continued)

Name		Description
In the wizard	In the administrative console	
(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 41. 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, PSFTRA, 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 PSFTRA001.</p> <p>If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter.</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 PeopleSoft Enterprise to 001 and 002. The component IDs for those instances, PSFTRA001 and PSFTRA002, 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 PSFTRAIinstanc.</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 42. 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 43. 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 44. Log file name details

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

Table 44. Log file name details (continued)

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

Log number of files (LogNumberOfFiles)

This property specifies the number of log files.

Table 45. 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 46. 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 47. 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 48. 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 an export. You set activation specification properties either through the J2C Bean wizard or the administrative console.

The following table lists the activation specification properties for inbound communication. 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 50.

Table 49. Activation specification properties

Descriptive name	Property name	Purpose
Adapter instance for event filtering	AdapterInstanceEventFilter	Identifier that determines whether this adapter instance processes specific events in the event store
Ensure once-only event delivery	AssuredOnceDelivery	Specifies whether the adapter provides assured once delivery of events
Do not process events that have a timestamp in the future	FilterFutureEvents	Specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time
Component interface for testing failed connection	PingCompIntfc	Specifies the component interface the adapter uses to validate a connection to the PeopleSoft Enterprise server
Component interface name for event notification	EventCIName	Specifies the component interface the adapter uses for event notification
Delivery type	DeliveryType	Determines the order in which events are delivered by the adapter to the export
Delimiter for keys in the event store	EventKeyDelimiter	Specifies the name and value for an object key in the event table
Event types to process	EventTypeFilter	A delimited list of event types that indicates to the adapter which events it should deliver
Java date format for event timestamp	DateFormat	Specifies the format used to create the event timestamp
Maximum connections	MaximumConnections	The maximum number of connections that the adapter can use for inbound event delivery
Minimum connections	MinimumConnections	The minimum number of connections that the adapter can use for inbound event delivery

Table 49. Activation specification properties (continued)

Number of times to retry the system connection	RetryLimit	The number of times the adapter tries to reestablish an inbound connection after an error
Interval between polling periods	PollPeriod	The length of time that the adapter waits between polling periods
Poll quantity	PollQuantity	The number of events that the adapter delivers to the export during each poll period
Retry EIS connection on startup	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the PeopleSoft Enterprise 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

Adapter instance for event filtering (AdapterInstanceEventFilter)

This property controls whether this adapter instance processes specific events in the event store.

Table 50. Adapter instance for event filtering details

Required	No
Default	null
Property type	String
Usage	<p>This property helps you migrate from WebSphere Business Integration Adapter for PeopleSoft to WebSphere Adapter for PeopleSoft Enterprise. WebSphere Business Integration Adapter for PeopleSoft allows you to perform load balancing on high-volume event types by allowing multiple adapter instances to process events of the same type. When load balancing is not required, a single adapter instance processes all events of a given type. This property is to enable seamless migration for WBI customers to JCA for customers who are currently taking advantage of the connectorID filtering.</p> <p>WebSphere Adapter for PeopleSoft Enterprise typically does not require load balancing in this way, but supports it so that you can migrate without modifying the database triggers or other mechanisms that write events to the event store.</p> <p>The AdapterInstanceEventFilter property corresponds to the ConnectorID property of the WebSphere Business Integration Adapter for PeopleSoft.</p> <p>To use this feature, the PeopleCode that create events in the event store must assign the appropriate value to the IBM_CONNECTOR_ID column.</p> <p>Table 51 on page 72 shows the interaction between the AdapterInstanceEventFilter property and the value in the IBM_CONNECTOR_ID column in the event store.</p> <p>If the EventTypeFilter and AdapterInstanceEventFilter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose IBM_CONNECTOR_ID column matches the AdapterInstanceEventFilter property.</p>

Table 50. Adapter instance for event filtering details (continued)

Example	See Table 51.
Globalized	Yes
Bidi supported	Yes

Table 51. Interaction of the AdapterInstanceEventFilter property with the IBM_CONNECTOR_ID column in the event store

AdapterInstanceEventFilter property	IBM_CONNECTOR_ID column of an event	Result
null	null	The adapter processes the event
null	Instance1	The adapter processes the event, because the IBM_CONNECTOR_ID column is not checked
Instance1	Instance1	The adapter processes the event
Instance1	Instance2	The adapter does not process the event, because the instance IDs do not match
Instance1	null	The adapter does not process the event, because the instance IDs do not match

Ensure once-only event delivery (AssuredOnceDelivery)

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

Table 52. Ensure once-only event delivery details

Required	Yes
Possible values	True False
Default	True
Property type	Boolean
Usage	<p>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.</p> <p>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.</p> <p>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.</p>
Globalized	No
Bidi supported	No

Component interface for testing failed connection (PingComInterface)

This property specifies the name of the PeopleSoft Enterprise component interface that the adapter uses to validate a connection to the PeopleSoft Enterprise server.

Table 53. Component interface for testing failed connection details

Row	Explanation
Required	Yes

Table 53. Component interface for testing failed connection details (continued)

Row	Explanation
Default	The name of the first component interface in the list
Property type	String
Usage	The name of the component interface that the adapter uses to test connectivity to the PeopleSoft Enterprise server. Specify a component interface name that already exists within your PeopleSoft Enterprise applications.
Globalized	No
Bidi supported	No

Component interface name for event notification (EventCIName)

This property specifies the name of the PeopleSoft Enterprise component interface that the adapter uses for inbound processing.

Table 54. Component interface name for event notification details

Row	Explanation
Required	Yes
Default	IBM_EVENT_CI
Property type	String
Usage	The name of the component interface that the adapter uses for inbound processing. To use inbound processing, you must create a component interface specifically for event notification within PeopleSoft Enterprise.
Globalized	No
Bidi supported	No

Delivery type (DeliveryType)

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

Table 55. Delivery type details

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

Delimiter for keys in the event store (EventKeyDelimiter)

This property specifies the delimiter for the object key name-value pair in the event table.

Table 56. Delimiter for keys in the event store details

Row	Explanation
Required	No
Default	=:
Property type	String
Usage	This property allows you to specify an object name and value to be used as an object key in the event store.
Example	CustomerID=2001
Globalized	No
Bidi supported	No

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

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

Table 57. Do not process events that have a timestamp in the future details

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

Event types to process (EventTypeFilter)

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

Table 58. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types
Default	null
Property type	String
Usage	Events are filtered by business object type. If the property is set, the adapter delivers only those events that are in the list. A value of null indicates that no filter will be applied and that all events will be delivered to the export.

Table 58. Event types to process details (continued)

Example	To receive only events relating to the Customer and Order business objects, specify this value: Customer,Order If the EventTypeFilter and AdapterInstanceEventFilter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose IBM_CONNECTOR_ID column matches the AdapterInstanceEventFilter property.
Globalized	No
Bidi supported	No

Retry limit for failed events (FailedEventRetryLimit)

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

Table 59. Retry limit for failed events details

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

Java date format for event timestamp (DateFormat)

This property specifies the format that is used for the event timestamp.

Table 60. Java date format for event timestamp details

Row	Explanation
Required	Yes
Default	MM/dd/yy
Property type	String
Usage	This property is used to format the date values obtained from the PeopleSoft Enterprise server.
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 61. Maximum connections details

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

Minimum connections (MinimumConnections)

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

Table 62. 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

Interval between polling periods (PollPeriod)

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

Table 63. 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	The poll period is established at a fixed rate, which means that if running the poll cycle is delayed for any reason (for example, if a prior poll cycle takes longer than expected to complete) the next poll cycle will occur immediately to make up for the lost time caused by the delay.
Globalized	No
Bidi supported	No

Maximum events in polling period (PollQuantity)

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

Table 64. Maximum events in polling period details

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

Number of times to retry the system connection (RetryLimit)

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

Table 65. Number of times to retry the system connection details

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

Retry EIS connection on startup (RetryConnectionOnStartup)

This property controls whether the adapter attempts to connect again to the PeopleSoft Enterprise server if it cannot connect at startup.

Table 66. Retry EIS connection on startup details

Required	No
Possible values	True False
Default	False
Property type	Boolean

Table 66. Retry EIS connection on startup details (continued)

Usage	<p>This property indicates whether the adapter should retry the connection to the PeopleSoft Enterprise 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 PeopleSoft Enterprise 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 67. 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 68. 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 68. Stop the adapter when an error is encountered while polling details (continued)

Globalized	No
Bidi supported	No

Globalization

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

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

Bidirectional script data transformation

Languages such as Arabic and Hebrew are written from right to left, yet they contain embedded segments of text that are written left to right, resulting in bidirectional script. When software applications handle bidirectional script data, standards are used to display and process it. Bidirectional script data transformation applies only to string type data. WebSphere 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). This is the format used by Windows. If an enterprise information system uses a different format, the adapter converts the format before introducing the data to WebSphere Application Server.

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 69. Bidirectional format attributes

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

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.

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.

Connection properties for the J2C Bean wizard

The following managed connection properties control bidirectional script data transformation.

- Password
- Username

Resource adapter properties

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

- Log file name
- Trace file name

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 70
 - setting in administrative console 31, 34
- adapter
 - project, create 16
- adapter architecture 2
- Adapter for PeopleSoft Enterprise
 - administering 29
- Adapter implementation 9
- application-specific information 43
- authentication
 - description 10
 - J2C Bean wizard 10
 - run time 11
- authentication alias 11

B

- business object information 43
- business objects 5
 - supported data operations 44
- business-object-level metadata 43

C

- clustered environment
 - deploying in 13
 - description 13
 - inbound processes 13
 - outbound processes 13
- compatibility matrix 2
- confidential data, disguising 9
- confidential tracing 9
- configuring
 - logging 37
 - tracing 37
- connector project 16
- custom properties
 - activation specification 31, 34
 - managed connection factory 30, 33
 - resource adapter 29, 32

D

- data operations 44
- debugging
 - self-help resources 41
 - XAResourceNotAvailableException exception 41
- deployment
 - options 11

E

- embedded adapter
 - activation specification properties, setting 31
 - considerations for using 12
 - description 11
 - managed connection factory properties, setting 30

- embedded adapter (*continued*)
 - resource adapter properties, setting 29
- enableHASupport property 13
- event store 4
- exceptions
 - XAResourceNotAvailableException 41
- external dependencies, adding 17
- External service connection properties 51, 64

F

- FFDC (first-failure data capture) 40
- files
 - SystemOut.log log file 39
 - trace.log trace file 39
- first-failure data capture (FFDC) 40

H

- hardware and software requirements 2
- hardware requirements 2
- high-availability environment
 - deploying in 13
 - description 13
 - inbound processes 13
 - outbound processes 13

I

- importing into project 17
- inbound configuration properties 62
- interaction specification property 62

J

- J2C Bean wizard
 - authentication in 10
- JAR file, adding external 17
- Java 2 security 11

L

- Log Analyzer 38
- Log and Trace Analyzer, support for 37
- log and trace files 37
- log files
 - changing file name 39
 - disabling 38
 - enabling 38
 - level of detail 38
 - location 39
- logging
 - configuring properties with administrative console 38

M

- managed (J2C) connection factory properties
 - setting in administrative console 30, 33

- matrix, compatibility 2
- metadata 43
 - business-object-level 43
 - property-level 44

O

- outbound configuration properties 50

P

- package files for adapters 38
- polling 4
- problem determination
 - self-help resources 41
 - XAResourceNotAvailableException exception 41
- properties
 - activation specification 31, 34
 - list of 70
 - configuration properties
 - inbound 62
 - outbound 50
 - inbound configuration 62
 - managed (J2C) connection factory 30, 33
 - outbound configuration 50
 - resource adapter 29, 32
- property-level metadata 44

R

- requirements, hardware and software 2
- resource adapter properties
 - details 54, 66
 - setting in administrative console 29, 32
- Retry limit property 77
- runtime environment
 - authentication in 11

S

- security
 - disguising sensitive data 9
- Security 9
- security, Java 2 11
- self-help resources 41
- sensitive data, disguising 9
- software dependencies, adding external 17
- software requirements 2
- stand-alone adapter
 - activation specification properties, setting 34
 - considerations for using 12
 - description 11
 - managed connection factory properties, setting 33
 - resource adapter properties, setting 32
- support
 - overview 37
 - self-help resources 41
- supported data operations 44
- SystemOut.log file 39

T

- Technical overview 2
- technotes 2, 41

- trace files
 - changing file name 39
 - disabling 38
 - enabling 38
 - level of detail 38
 - location 39
- trace.log file 39
- tracing
 - configuring properties with administrative console 38
- troubleshooting
 - overview 37
 - self-help resources 41
 - XAResourceNotAvailableException exception 41

W

- WebSphere Extended Deployment 13

X

- XAResourceNotAvailableException 41



Printed in USA