





Note

Before using this information and the product it supports, read the information in "Notices" on page 133.

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
External service wizard	6

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	14
Migrating to version 6.2	15
Migration considerations	16
Performing the migration	17
Updating but not migrating a version 6.0.2.x or version 6.1.x project	18
Migrating WebSphere business integration applications for use with Version 6.2 WebSphere Adapters	19
Road map for migrating applications from WebSphere InterChange Server	20
Migration considerations for WebSphere business integration adapters	21
Migrating application artifacts from WebSphere InterChange Server	22
Migrating adapter-specific artifacts	22
Import, export, and WSDL file changes after migration	25

Chapter 3. Samples and tutorials 27

Chapter 4. Configuring the module for deployment. 29

Road map for configuring the module	29
Creating a custom event project in PeopleTools	31
Creating an authentication alias	32
Creating the project	33
Adding external software dependencies	34
Setting connection properties for the external service wizard	35
Configuring the module for outbound processing	38
Selecting business objects and services	38
Configuring the selected objects	39
Setting deployment properties and generating the service	40
Configuring the module for inbound processing	47
Selecting business objects and services	47

Configuring the selected objects	48
Setting deployment properties and generating the service	49

Chapter 5. Changing interaction specification properties using the assembly editor 57

Chapter 6. Deploying the module 59

Deployment environments	59
Deploying the module for testing	59
Adding external dependencies	59
Generating and wiring a target component for testing inbound processing	60
Adding the module to the server	62
Testing the module for outbound processing using the test client	63
Deploying the module for production	64
Adding external software dependencies on the server	64
Installing the RAR file (for modules using stand-alone adapters only)	64
Exporting the module as an EAR file	66
Installing the EAR file	66

Chapter 7. Administering the adapter module 69

Changing configuration properties for embedded adapters	69
Setting resource adapter properties for embedded adapters	69
Setting managed (J2C) connection factory properties for embedded adapters	71
Setting activation specification properties for embedded adapters	73
Changing configuration properties for stand-alone adapters	75
Setting resource adapter properties for stand-alone adapters	75
Setting managed (J2C) connection factory properties for stand-alone adapters	76
Setting activation specification properties for stand-alone adapters	78
Starting the application that uses the adapter	79
Stopping the application that uses the adapter	79
Monitoring performance using Performance Monitoring Infrastructure	80
Configuring Performance Monitoring Infrastructure	80
Viewing performance statistics	82
Enabling tracing with the Common Event Infrastructure (CEI)	83

Chapter 8. Troubleshooting and support 85

Support for the Log and Trace Analyzer 85
Configuring logging and tracing 86
 Configuring logging properties 86
 Changing the log and trace file names 87
First-failure data capture (FFDC) support 88
Business faults 89
Resolving memory limit exception during outbound processing 90
Self-help resources 90

Chapter 9. Reference information 93

Business object information 93
 Application-specific information 93
 Supported operations 94
PeopleCode for a custom event project 95
Outbound configuration properties 100
 Connection properties for the external service wizard 101
 Resource adapter properties 104

Managed connection factory properties 107
Interaction specification properties 111
Inbound configuration properties 112
 Connection properties for the external service wizard 114
 Resource adapter properties 116
 Activation specification properties 120
Globalization 129
 Globalization and bidirectional data transformation 129
 Properties enabled for bidirectional data transformation 131
Adapter messages 131
Related information 131

Notices 133

Programming interface information 135
Trademarks and service marks 135

Index 137

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.

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 a 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.

The adapter can run in such JCA-compliant environments as WebSphere Process Server or WebSphere Enterprise Service Bus, WebSphere Application Server, and WebSphere Message Broker. The way you set up and use the adapter depends upon the environment in which the adapter runs.

For example, consider that you are setting up an adapter using WebSphere Integration Developer tooling on WebSphere Process Server or WebSphere Enterprise Service Bus, 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 Process Server or WebSphere Enterprise Service Bus. The application component interacts with the adapter instead of interacting with the PeopleSoft function.

The component that you generate with the external service wizard of WebSphere Integration Developer 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 listens 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 Process Server or WebSphere Enterprise Service Bus 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. Inbound processing 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.

To use inbound processing, you must create a custom event project in PeopleSoft.

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 external service wizard, which is launched from WebSphere Integration Developer, 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.

Business graphs

You can optionally choose, during adapter configuration, to generate a business graph. In version 6.0.2, each top-level business object is contained in a business graph, which includes a verb that an application can use in version 6.0.2 to specify additional information about the operation to be performed. In version 6.2, business graphs are required only in these situations:

- If you need to use the outbound ApplyChanges operation
- When adding business objects to a module created with a version of WebSphere Integration Developer earlier than version 6.2

If business graphs exist, they are processed, but the verb is ignored for all operations except ApplyChanges.

External service wizard

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

Using WebSphere Integration Developer, you establish a connection to the PeopleSoft Enterprise server to browse the metadata repository on the PeopleSoft Enterprise server. You also specify connection information, such as the user name and password needed to access the server.

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

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.
- Business integration concepts and models, including the Service Component Architecture (SCA) programming model.
- 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 Process Server or WebSphere Enterprise Service Bus.

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 external service wizard or change the `HideConfidentialTrace` property. When this property is enabled, the adapter replaces the sensitive data with XXX's.

See “Managed connection factory properties” on page 107 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 external service 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 Process Server or WebSphere Enterprise Service Bus, when the adapter connects to the PeopleSoft Enterprise server to process outbound requests and inbound events.

Authentication in the wizard

The external service 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 external service wizard lets you configure the adapter to get the user information by using any of the following methods:

- Adapter properties
- Data source
- J2C authentication alias

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

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

Using a J2C authentication 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.

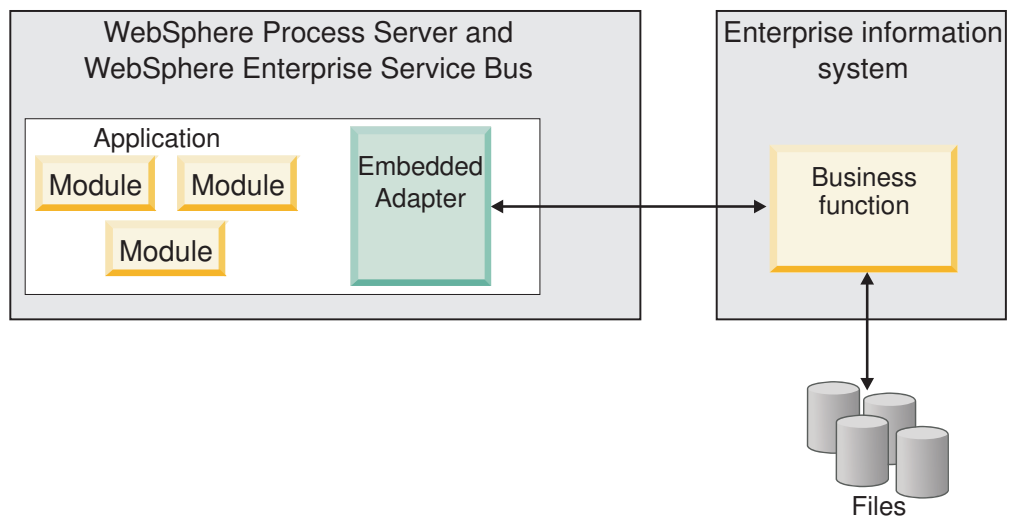
The deployment options are described below:

- **With module for use by single application.** With the adapter files embedded in the module, you can deploy the module to any application server. Use an

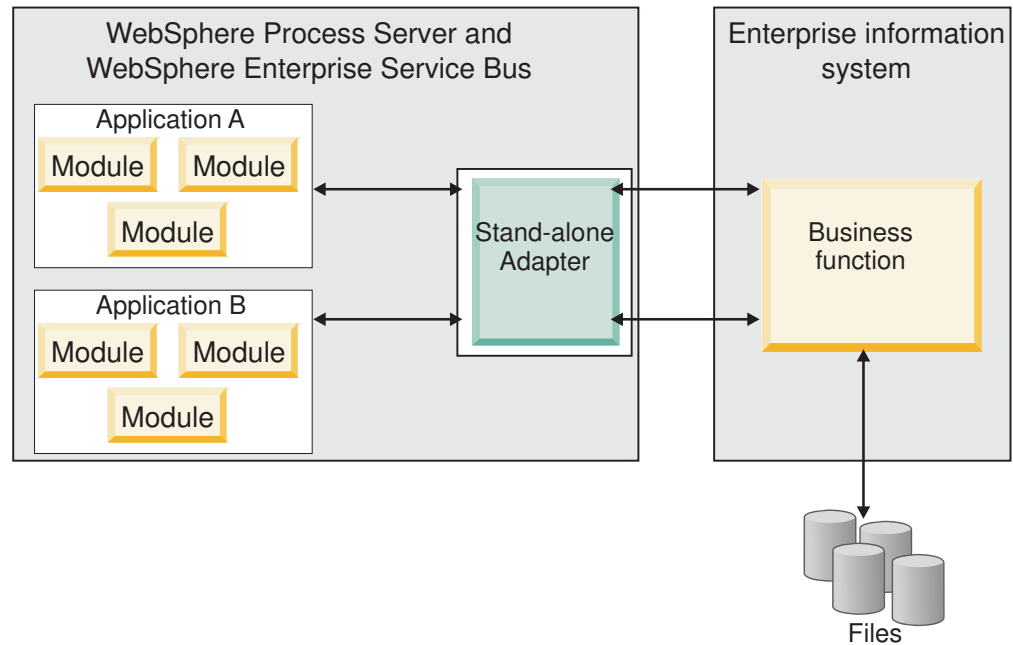
embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications.** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

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



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



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

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

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

Considerations for embedding an adapter in the application

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

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

Considerations for using a stand-alone adapter

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

- Stand-alone adapters have no class loader isolation.
Because stand-alone adapters have no class loader isolation, only one version of any given Java artifact is run and the version and sequence of that artifact is undetermined. For example, when you use a stand-alone adapter there is only *one* resource adapter version, *one* adapter foundation class (AFC) version, or *one* third-party JAR version. All adapters deployed as stand-alone adapters share a single AFC version, and all instances of a given adapter share the same code version. All adapter instances using a given third-party library must share that library.
- If you update any of these shared artifacts, all applications using the artifacts are affected.
For instance, if you have an adapter that is working with server version X, and you update the version of the client application to version Y, your original application might stop working.
- 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 Process Server or WebSphere Enterprise Service Bus, 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. For information about creating clustered environments, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_v61.html.

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

and configurations to evenly handle load variations. For information on the benefits of WebSphere Extended Deployment, see the following link: <http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1/index.jsp>.

In clustered environments, adapter instances 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 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. For more information on workload management, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Migrating to version 6.2

By migrating to version 6.2 of WebSphere Adapter for PeopleSoft Enterprise, you automatically upgrade from the previous version of the adapter. Additionally, you can migrate your applications that embed an earlier version of the adapter, so that the applications can utilize features and capabilities present in version 6.2.

Migration considerations

WebSphere Adapter for PeopleSoft Enterprise version 6.2 includes updates that might affect your existing adapter applications. Before migrating applications that will use WebSphere Adapter for PeopleSoft Enterprise, you must consider some factors that might affect your existing applications.

Compatibility with earlier versions

WebSphere Adapter for PeopleSoft Enterprise version 6.2 is fully compatible with version 6.0.2.x and version 6.1.x of the adapter and can work with custom business objects (XSD files) and data bindings.

Because version 6.2 of WebSphere Adapter for PeopleSoft Enterprise is fully compatible with version 6.0.2.x and version 6.1.x, any of your applications that used version 6.0.2.x or version 6.1.0.x of WebSphere Adapter for PeopleSoft Enterprise will run unchanged when you upgrade to version 6.2. However, if you want your applications to use features and functionality present in version 6.2 of the adapter, run the migration wizard.

The migration wizard replaces (upgrades) version 6.0.2.x or version 6.1.x of the adapter with version 6.2 and enables version 6.2 features and functionality for use with your applications.

Note: The migration wizard does not create components or modify existing components, such as mappers and mediators to work with version 6.2 of the adapters. If you are upgrading to version 6.2 and any of your applications embed an adapter that is version 6.1.x or earlier, and you want your applications to take advantage of the features and functions in version 6.2, you might need to make changes to those applications.

If artifacts are inconsistent with regard to versioning within a single module, the entire module is marked and will not be selectable for migration. Version inconsistencies are recorded in the workspace log, because they might indicate corruptions to the project.

If the WebSphere Integration Developer version does not match that of WebSphere Adapter for PeopleSoft Enterprise, the products cannot be run together.

The external service wizard cannot be run when the following product combinations are used:

Table 3. Product combinations not supported

WebSphere Integration Developer	WebSphere Adapter for PeopleSoft Enterprise
version 6.2	version 6.1.x
version 6.1.x	version 6.0.2.x
version 6.0.2.x	version 6.2
version 6.0.2.x	version 6.1.x

Deciding whether to upgrade or to upgrade and migrate

By default, the migration wizard upgrades the adapter and migrates the application artifacts so that the applications can use features and functions in

version 6.2 of the adapter. When you choose to upgrade the adapter by selecting a project, the wizard automatically selects the associated artifacts for migration.

If you decide that you want to upgrade the adapter from version 6.0.2.x or version 6.1.x to version 6.2, but you do not want to migrate the adapter artifacts, you can do so by deselecting the adapter artifacts from the appropriate area of the migration wizard.

Running the migration wizard without any adapter artifacts selected will install and upgrade your adapter, but your artifacts will not be migrated and your applications will not be able to take advantage of the features and capabilities that exist in version 6.2 of the adapter.

Running the migration wizard in a test environment first

Because adapter migration might require you to make changes to those applications that will use version 6.2 of WebSphere Adapter for PeopleSoft Enterprise, you should always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

The migration wizard is fully integrated with the development environment.

Deprecated features

A deprecated feature is one that is supported but no longer recommended and that might become obsolete. The following features from earlier versions of WebSphere Adapter for PeopleSoft Enterprise have been deprecated in version 6.1.x and version 6.2 and might require changes to your applications:

- The EnableCrossReferencing parameter is deprecated.
- Application-specific information maintained at the business object attribute level for delete operations is deprecated. It is now maintained at the business object level.

Performing the migration

You can migrate a project or EAR file to version 6.2, by using the adapter migration wizard. When the tool is finished, the migration is complete and you can work in the project or deploy the module.

Before you begin

Review the information in *Migration considerations*.

About this task

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

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

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

Procedure

1. Import the PI (project interchange) file for an existing project or the EAR (enterprise archive) file for a deployed application into the workspace.
2. Change to the Java EE perspective.
3. Right-click the module and select **Update** → **Update Connector Project**.

Note: Additionally, you can launch the adapter migration wizard in the following ways:

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

When you launch the migration wizard from the connector project context menu while in the Java EE perspective, by default all of the dependent artifact projects are selected. If you deselect a dependent artifact project, that project is not migrated. You will be allowed to migrate any projects that you deselect at a later time. Previously migrated projects, projects with a current version, and projects that contain errors are unavailable for migration and are not selected.

4. Review the tasks and warnings presented on the welcome page, and click **Next**.
5. A warning window opens with the message, "The properties that are not supported in the version of the target adapter will be removed during the migration." Click **OK** to continue to the next step.
6. On the Review changes window you can review the migration changes that will occur in each of the artifacts that you are migrating by clicking the + sign.
7. Click **Finish** to perform the migration.

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

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

8. Check the Problems view for messages from the migration wizard, which start with the string CWPAD.
9. If you are migrating an EAR file, optionally create a new EAR file with the migrated adapter and artifacts, and deploy it to WebSphere Process Server or WebSphere Enterprise Service Bus. For more information about exporting and deploying an EAR file, see the topics devoted to it in this documentation.

Results

The project or EAR file is migrated to version 6.2. You do not need to run the external service wizard after exiting the adapter migration wizard.

Updating but not migrating a version 6.0.2.x or version 6.1.x project

You can upgrade the adapter from version 6.0.2.x or version 6.1.x to version 6.2, while choosing not to migrate the adapter project artifacts.

About this task

Because the internal name of the adapter changed in version 6.2, artifacts in a version 6.0.2.x or version 6.1.x project must be updated to use the new name before you can use the adapter wizard in WebSphere Integration Developer, version 6.2. Use the migration wizard to update a version 6.0.2.x or version 6.1.x project. Then use the Quick Fix feature of WebSphere Integration Developer to change the adapter name in project artifacts.

Procedure

1. Import the project interchange (PI) file into the workspace.
2. In the Java EE perspective, right-click the project name and click **Update** → **Update Connector Project**. The adapter migration wizard opens.
3. On the welcome page, click **Next**.
4. On the Select Projects window, deselect the dependent artifact projects, and then click **Finish**.
5. A warning window opens with the message, “The properties that are not supported in the version of the target adapter will be removed during the migration.” Click **OK** to continue to the next step.
6. In the Quick Fix window, make sure the fix **Rename the referenced adapter** is selected, and then click **OK**.
7. If the error remains visible, click **Project** → **Clean**, select the project you just updated, and then click **OK**.

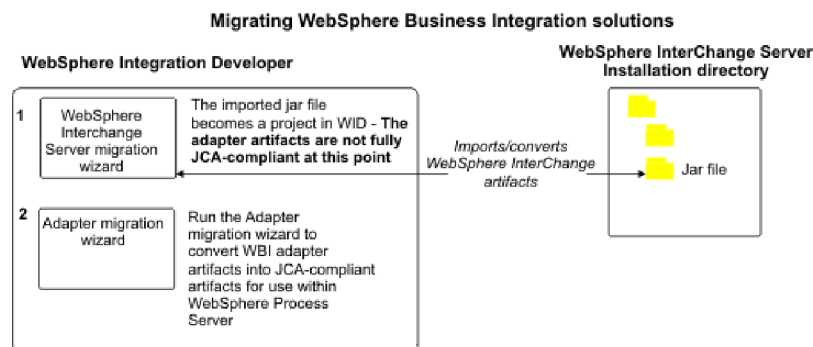
Results

The project can now be used with WebSphere Adapter for PeopleSoft Enterprise, version 6.2.

Migrating WebSphere business integration applications for use with Version 6.2 WebSphere Adapters

Migrating WebSphere business integration applications for use with Version 6.2 of your WebSphere adapter is a multistep process. The first step migrates and converts the artifacts from the WebSphere InterChange Server and creates a project for the artifacts in WebSphere Integration Developer. The remaining steps migrate adapter-specific artifacts and complete the conversion of the artifacts into the JCA-compliant format supported by Version 6.2 the adapter.

The following diagram shows the wizards that you use to migrate WebSphere business integration solutions from WebSphere InterChange Server, so that these applications can be used with Version 6.2 of your adapter.



Road map for migrating applications from WebSphere InterChange Server

To use Version 6.2 of WebSphere Adapter for PeopleSoft Enterprise with applications from WebSphere InterChange Server, you need to migrate the application artifacts and convert them so that they can be deployed and run on WebSphere Process Server or WebSphere Enterprise Service Bus. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

The following figure illustrates the flow of the migration task. The steps that follow the figure describe this task at a high level only. See the topics following this road map for the details on how to perform each of these steps.

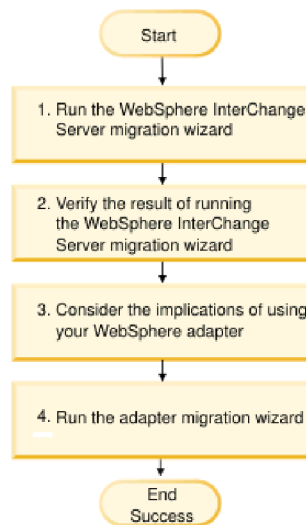


Figure 1. Road map for migrating applications from WebSphere InterChange Server

Migrating applications from WebSphere InterChange Server

This task consists of the following steps:

1. Run the WebSphere InterChange Server migration wizard.
The WebSphere InterChange Server migration wizard moves the application artifacts into the WebSphere Integration Developer. The migrated adapter artifacts are not fully JCA-compliant at the completion of this task.
2. Verify that the WebSphere InterChange Server migration is successful.
Review all messages from the Migration results window and take actions if required.
3. Consider the implications of using Version 6.2 of WebSphere Adapter for PeopleSoft Enterprise.
In addition to considerations for migrating WebSphere InterChange Server applications, you need to consider how Version 6.2 of WebSphere Adapter for PeopleSoft Enterprise works with the migrated applications. Some of the adapter operations supported by WebSphere InterChange Server applications might be supported and implemented differently with Version 6.2 of the adapter.
4. Run the adapter migration wizard.

Run the adapter migration wizard to update adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl files) for use with Version 6.2 of the adapter.

Migration considerations for WebSphere business integration adapters

After migration, you will have an adapter that is compliant with the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) and designed specifically for service-oriented architecture.

Note: WebSphere business integration adapter and JCA compliant adapter use different event Component Interface. You must change to the new event Component Interface supported by the JCA compliant adapter manually after the completion of migration.

Operations considerations

Several WebSphere business integration adapter operations are supported differently in WebSphere Adapter for PeopleSoft Enterprise Version 6.2. You might need to provide additional development effort when you plan to use the following operations:

DeltaUpdate

The DeltaUpdate is not supported in WebSphere Adapter for PeopleSoft Enterprise. You must use the ApplyChanges operation to implement delta processing of components.

Exists The Exists operation is implemented differently in WebSphere Adapter for PeopleSoft Enterprise.

RetrieveByContent

The RetrieveAll operation is supported in WebSphere Adapter for PeopleSoft Enterprise as an equivalent to the RetrieveByContent operation.

Run the WebSphere Interchange Server migration wizard before running the migration wizard

Before running the adapter migration wizard, use the WebSphere Interchange Server migration wizard to generate the application artifacts for the WebSphere business integration adapter, including the business objects, maps, and collaborations. Then you can run the adapter migration wizard to update the adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl) so that they are suitably converted into a format that is compliant with JCA.

Run the migration wizard in a test environment first

Because migrating from a WebSphere business integration adapter to WebSphere Adapter for PeopleSoft Enterprise might require changes to those applications that use Version 6.2 of WebSphere Adapter for PeopleSoft Enterprise, always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

Note: After migrating the WebSphere business integration adapters to become compliant with the J2EE Connector Architecture (JCA), you will not be able add a new application specific information (ASI) property to the business object in the

business object editor. To add a new ASI property after migration, open the business object with a text editor and add the ASI property manually.

Migrating application artifacts from WebSphere InterChange Server

To migrate the application artifacts into WebSphere Integration Developer, run the WebSphere InterChange Server migration wizard. The wizard imports and converts the majority of artifacts into a format that is compatible with WebSphere Process Server or WebSphere Enterprise Service Bus.

Before you begin

Launch the WebSphere InterChange Server migration wizard from within WebSphere Integration Developer to migrate the application artifacts from WebSphere InterChange Server format into artifacts that are compatible with WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

For information about how to prepare to migrate artifacts from WebSphere InterChange Server and for detailed instructions on performing the migration and verifying that the migration was successful, go to the IBM WebSphere Business Process Management information center and read the topic Migrating to WebSphere Process Server or WebSphere Enterprise Service Bus from WebSphere InterChange Server.

Running WebSphere InterChange Server migration wizard might not fully convert adapter-specific artifacts (such as service descriptors, service definitions, and business objects) into WebSphere Process Server or WebSphere Enterprise Service Bus compatible artifacts. To complete the migration of adapter-specific artifacts, run the adapter migration wizard after you have successfully run the WebSphere InterChange Server migration wizard.

Results

The project and application artifacts are migrated and converted into WebSphere Process Server compatible artifacts.

What to do next

Run the adapter migration wizard to migrate the adapter-specific artifacts.

Migrating adapter-specific artifacts

You can migrate a project using the adapter migration wizard. The adapter migration wizard updates adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl) for use with version 6.2 of the adapter. When you finish running the adapter migration wizard, the migration is complete and you can work in the project or deploy the module.

Before you begin

Before running the adapter migration wizard you should do the following:

- Review the information in the "Migration considerations" topic.

- Run the WebSphere InterChange Server migration wizard to migrate the project and convert data objects for use with WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

Note: After migration is complete, the module will work only with Version 6.2 of your adapter.

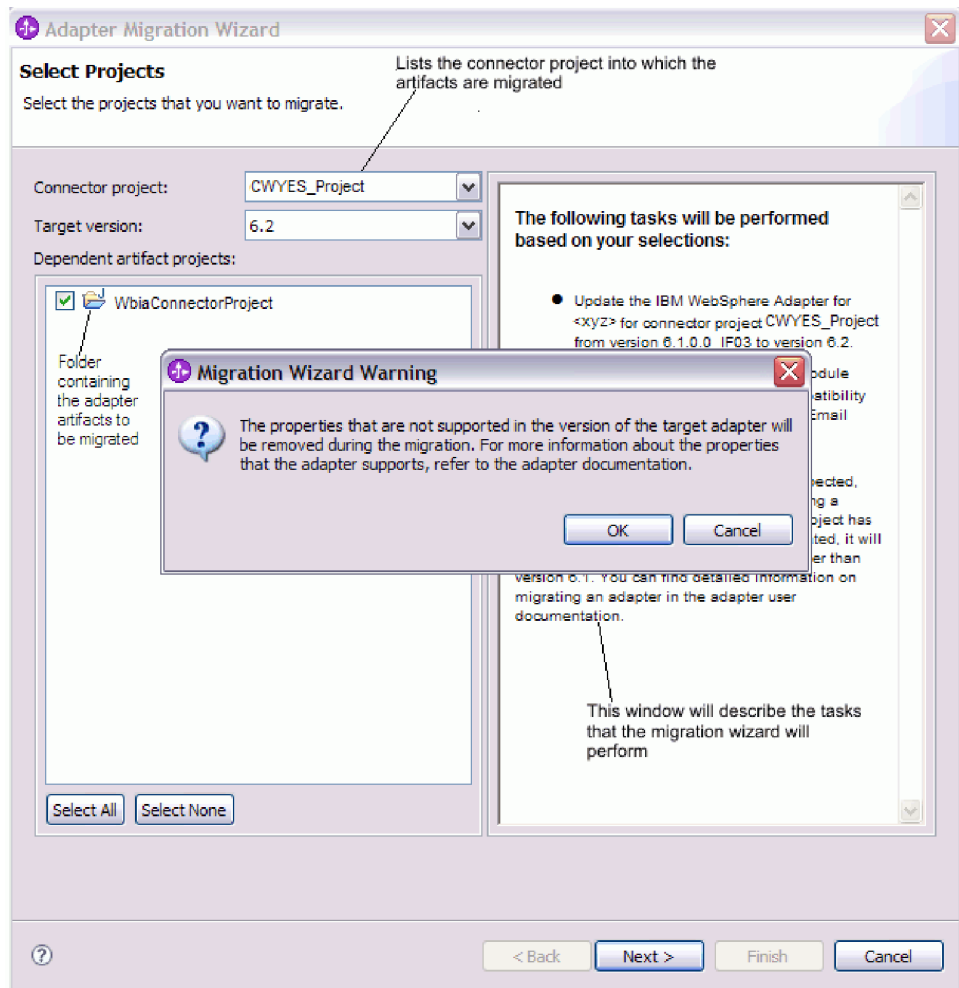
To perform the migration in WebSphere Integration Developer, complete the following steps.

Procedure

1. In the Java EE perspective, right-click the connector project and select **Update** → **Update Connector Project**.

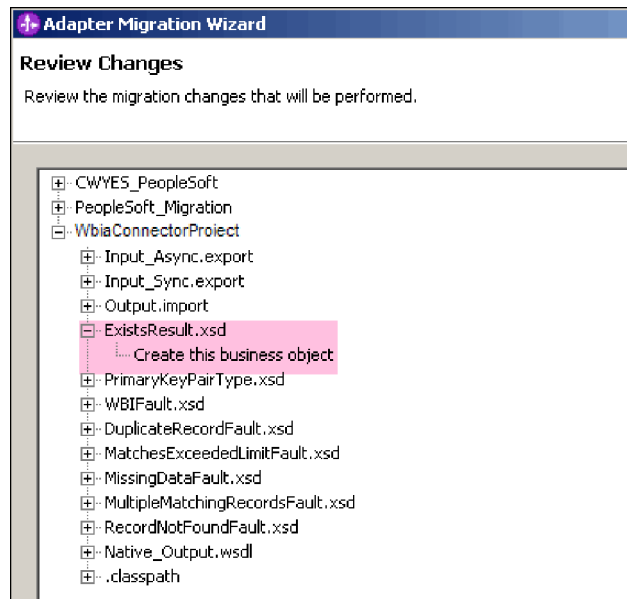
Note: Additionally, you can launch the adapter migration wizard by right-clicking the module project in the Java EE perspective and selecting **Update** → **Migrate Adapter Artifacts**.

The following figure describes the functional areas of the wizard.

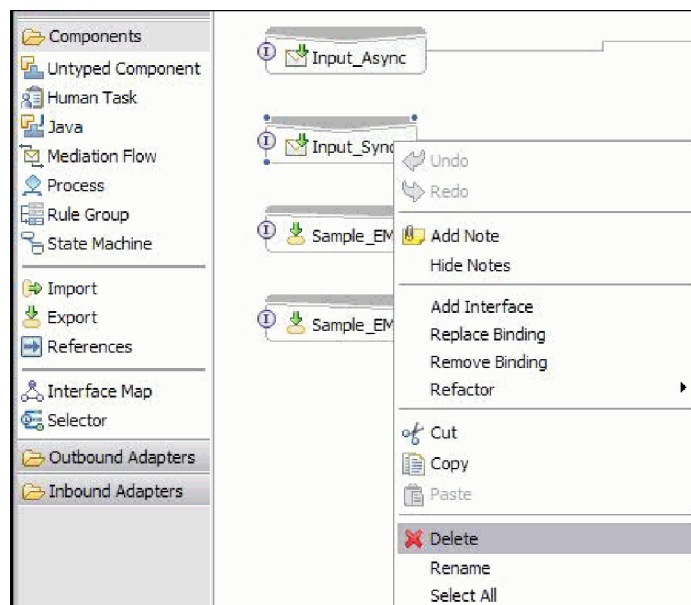


When you launch the migration wizard from the connector project context menu while in the Java EE perspective, by default all the dependent artifact projects are selected. If you deselect a dependent artifact project, that project is not migrated.

- On the Review changes window you can review the migration changes that will occur in each of the artifacts that you are migrating by clicking the + sign.



- Click **Finish** to perform the migration.
Before performing the migration process, the wizard backs up all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. If the migration fails for any reason, or if you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.
- Select **Project > Clean**, to refresh and rebuild the workspace for the changes to take effect.
- On successful migration, all backed up projects are deleted. Remove the Sync inbound flow manually as this flow is not used by the adapter. From the migrated project, select the Input_Sync inbound flow, right-click and select **Delete**.



6. If you are migrating an EAR file, create a new EAR file with the migrated adapter and artifacts, and deploy it to WebSphere Process Server or WebSphere Enterprise Service Bus. For information about exporting and deploying an EAR file, see “Deploying the module for production” on page 64.

Results

The project is migrated to Version 6.2. You do not need to run the external service wizard after exiting the adapter migration wizard.

Import, export, and WSDL file changes after migration

The WebSphere InterChange Server migration wizard moves the application artifacts into the WebSphere Integration Developer. The migrated adapter artifacts are not fully JCA-compliant at the completion of this task. You can complete the migration of the adapter-specific artifacts (such as service descriptors, service definitions, and business objects) to a JCA compatible format by running the adapter migration wizard. The changes by the migration wizard are reflected in the service definition files (.import, .export and, .wsdl).

Changes to the import file after migration

During migration, the affected module artifacts are migrated to an import file. The existing JMS Binding property before the migration changes to EIS Binding property in the import file. The other property details information added in the import file include information about data binding configuration, changes to connection information in Managed Connection Factory Properties, and several new method bindings.

Changes to the export file after migration

During migration, the affected module artifacts are migrated to an export file. The existing JMS Binding property before the migration changes to EIS Binding property in the export file. The other property details information added in the export file include information about data binding configuration, changes to connection information in Managed Connection Factory Properties, and several new method bindings.

Changes to the WSDL file after migration

During migration, the affected module artifacts are migrated to corresponding WSDL files that include PeopleSoft specific service description WSDL artifacts. The service description files become JCA compatible. The WSDL files will have input and output type for each operation. Both the inbound and outbound operations work on their specific input types to produce corresponding output types after the operations execution.

Note:

- When you migrate multiple inbound top level business objects in the project, only the first top level business object inbound feature works correctly. For the other top level business object inbound to work correctly, you must manually modify the "emit + [verb name] + afterimage + [business object name]" method in the Input_Processing.java and Input_Async_Processing.java class to call the correct destination services.
- During inbound processing, a dynamic method called emit is generated. A business object name is appended to emit to create an actual method. The

function selector determines the business object name to append to the emit method. For example, if the business object name is Customer, the actual method name will be emitCustomer. Emit is the only operation available during inbound processing.

- The WebSphere business integration adapter properties (such as *config*) that are not valid or not supported by WebSphere Adapter for PeopleSoft Enterprise are removed from the migrated artifacts.

Chapter 3. Samples and tutorials

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

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

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

Chapter 4. Configuring the module for deployment

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

Road map for configuring the module

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

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

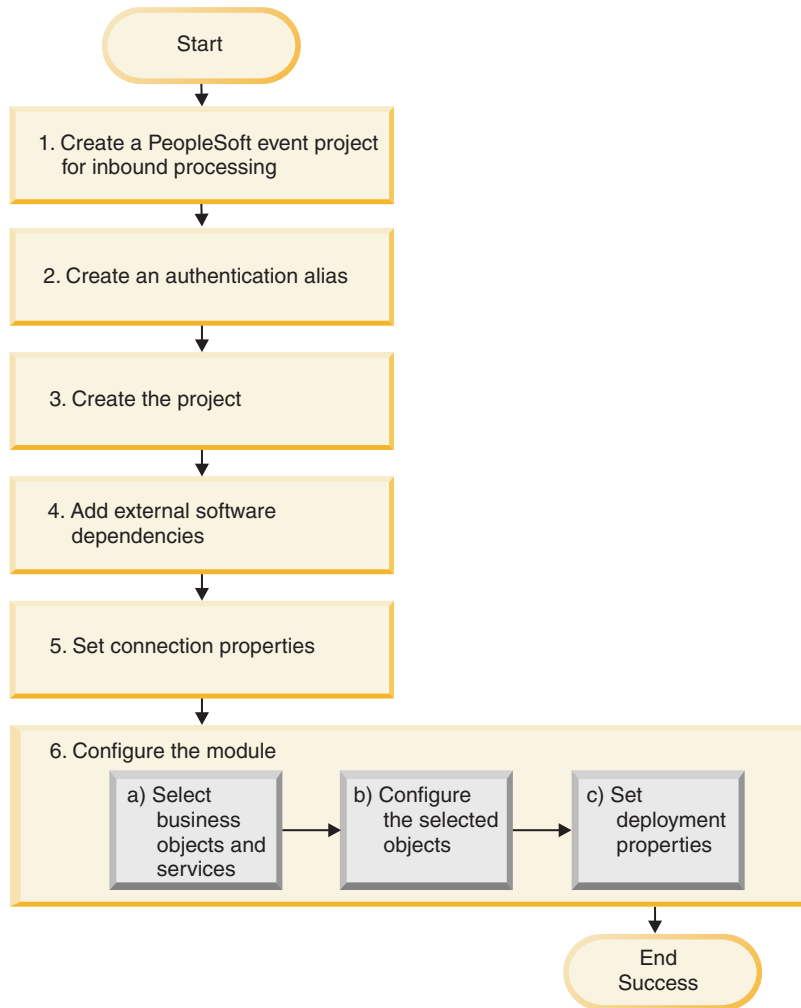


Figure 2. Road map for configuring the module

Configuring the module for deployment

This task consists of the following step:

1. Create a PeopleSoft event project for inbound processing.
2. Create an authentication alias to access the PeopleSoft Enterprise server with an encrypted password. This step is optional, depending on your policy for handling passwords and IDs. You perform this step using the server.
3. Create the project. First, start the external service wizard in WebSphere Integration Developer to begin the process of creating and deploying a module. The wizard creates a project that is used to organize the files associated with the module.
4. Add the external software dependencies required by WebSphere Adapter for PeopleSoft Enterprise to the project. These dependencies are also required when you export the module as an EAR file, and deploy the EAR file to the server.
5. Set connection properties that the external service wizard needs to connect to the PeopleSoft Enterprise server for discovery of objects and services.
6. Configure the module for inbound or outbound processing by using the external service wizard to find and select business objects and services from the PeopleSoft Enterprise server, and to generate business object definitions and related artifacts.

- a. Select business objects and services for inbound or outbound processing from the business integration components discovered by the external service wizard.
- b. Configure the selected objects by specifying operations and other properties that apply to all of the business objects.
- c. Set deployment properties that the adapter uses to connect to the PeopleSoft Enterprise server at run time. Then, generate the service by using the external service wizard to save the new module, which contains the business object or objects you configured, the import or export file, and the service interface.

Creating 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.

Field name	Field description
IBM_XID	The transaction ID which ensures providing of assured event delivery.
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.

Creating an authentication alias

An authentication alias is a feature that encrypts the password used by the adapter to access the PeopleSoft Enterprise server. The adapter can use it to connect to the PeopleSoft Enterprise server instead of using a user ID and password stored in an adapter property.

Before you begin

To create an authentication alias, you must have access to the administrative console. The following procedure shows you how to gain access to the administrative console through WebSphere Integration Developer. If you are using the administrative console directly (without going through WebSphere Integration Developer), log in to the administrative console and skip to step 2 on page 33.

About this task

An authentication alias is a feature that encrypts the password used by the adapter to access the PeopleSoft Enterprise server. Using an authentication alias eliminates the need to store the password in clear text in an adapter configuration property, where it might be visible to others.

To create an authentication alias, use the following procedure.

Procedure

1. Start the administrative console.
To start the administrative console from WebSphere Integration Developer, perform the following steps:
 - a. In the Business Integration perspective of WebSphere Integration Developer, click the **Servers** tab.
 - b. If the server does not show a status of **Started**, right-click the name of the server (for example, **WebSphere Process Server**) and click **Start**.
 - c. Right-click the name of the server, click **Administration** and click **Run administrative console**.
 - d. Log on to the administrative console. If your administrative console requires a user ID and password, type the ID and password and click **Log in**. If the user ID and password are not required, click **Log in**.
2. In the administrative console, click **Security** → **Secure administration, applications, and infrastructure**.
3. Under **Authentication**, expand **Java Authentication and Authorization Service** and click **J2C authentication data**.
4. Create an authentication alias:
 - a. In the list of J2C authentication aliases that is displayed, click **New**.
 - b. In the **Configuration** tab, type the name of the authentication alias in the **Alias** field.
 - c. Type the user ID and password that are required to establish a connection to the PeopleSoft Enterprise server.
 - d. Optionally type a description of the alias.
 - e. Click **OK**.
The newly created alias is displayed.
The full name of the alias contains the node name and the authentication alias name you specified. For example, if you create an alias on the node `widNode` with the name `ProductionServerAlias`, then the full name is `widNode/ProductionServerAlias`. This full name is the one you use in subsequent configuration windows.
 - f. Click **Save**, and then click **Save** again.

Results

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

Creating the project

To begin the process of creating and deploying a module, you start the external service wizard in WebSphere Integration Developer. You then create a project, which is used to organize the files associated with the adapter.

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 the PeopleSoft Enterprise server.

About this task

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

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

Procedure

1. To start the external service wizard, go to the Business Integration perspective of WebSphere Integration Developer, and then click **File** → **New** → **External Service**.
2. In the New external service window, make sure **Adapters** is selected, and click **Next**.
3. From the Select an Adapter window, create a project or select an existing project.
 - To create a project, perform the following steps:
 - a. Select **IBM WebSphere Adapter for PeopleSoft Enterprise**, and click **Next**.
 - b. In the Adapter Import window, provide another name for the project (if you want to use a name other than **CWYES_PeopleSoft**), select the server (for example, **WebSphere Process Server v6.2**), and click **Next**.
 - To select an existing project, perform the following steps:
 - a. Expand **IBM WebSphere Adapter for PeopleSoft Enterprise**.
 - b. Select a project.
For example, if you have an existing project named **CWYES_PeopleSoftAdapter**, you can expand **IBM WebSphere Adapter for PeopleSoft Enterprise** and select the project.
 - c. Click **Finish**.

Results

A new project is created and is listed in the Business Integration window.

What to do next

Provide the location of the psjoa.jar file and other required files.

Adding external software dependencies

The external service wizard needs two files either copied or generated from the PeopleSoft Enterprise to be able to communicate with it. Use the external service 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 WebSphere Integration Developer to specify the location of the files.

Before you begin

You should be running the external service wizard in WebSphere Integration Developer 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 Required Files and Libraries window of the external service wizard to specify the location of the files.

Procedure

1. From the Required Files and Libraries 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 external service wizard

To set connection properties for the external service wizard so that it can access the PeopleSoft Enterprise server, specify such information 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 external service 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. From the Processing Direction 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. From 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.

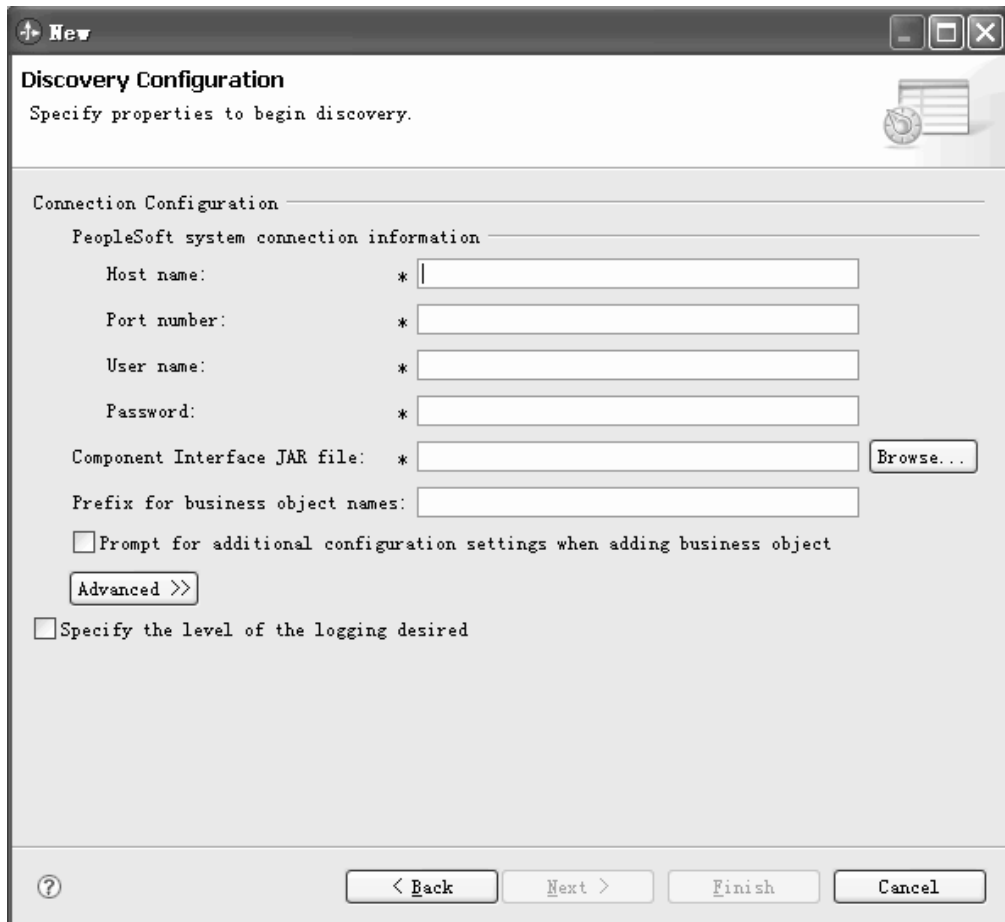


Figure 3. The Discovery Configuration window

- 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 (bidirectional properties), click **Advanced**.

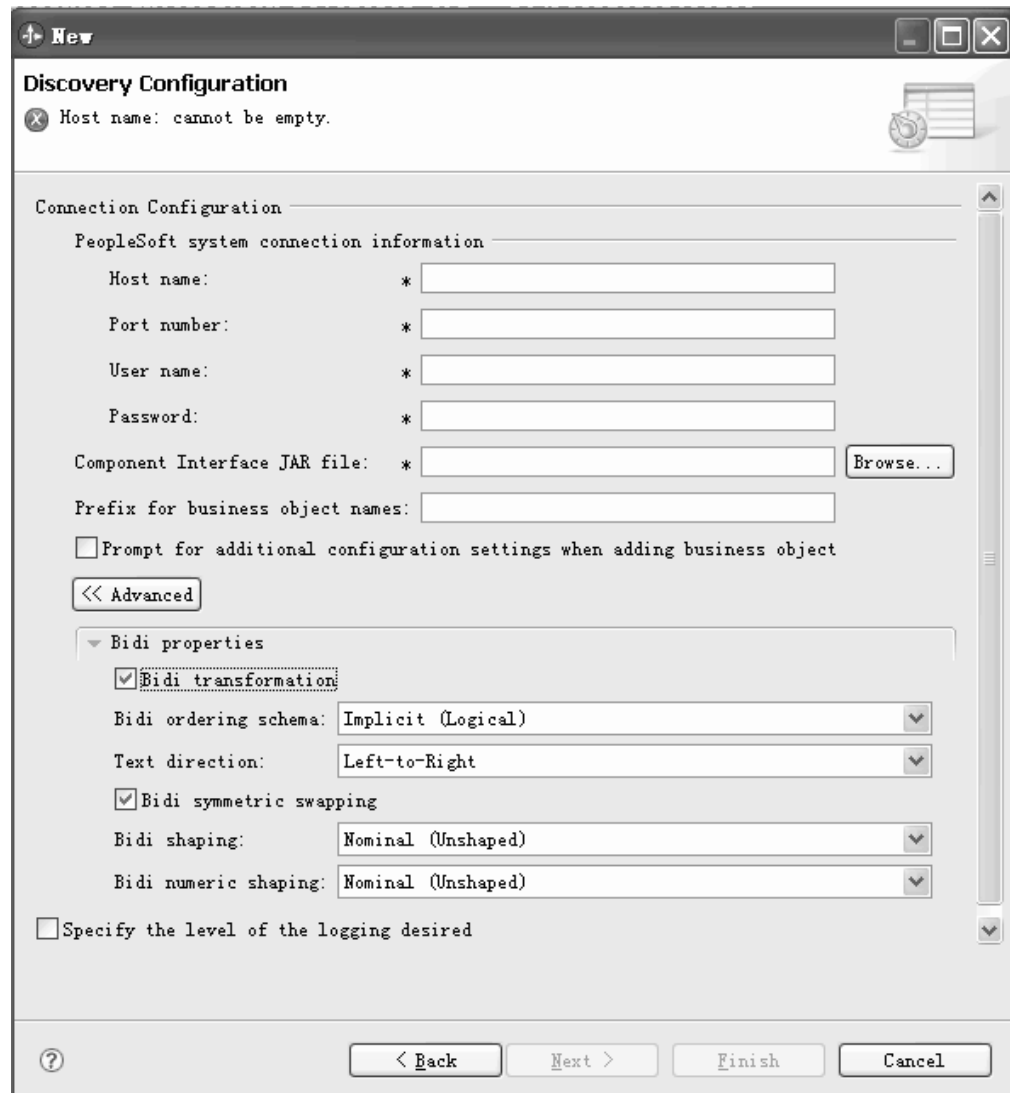


Figure 4. The Discovery Configuration window, showing the additional advanced property sections after clicking the **Advanced** button

4. If you need to set bidirectional properties, perform the following steps:
 - a. Expand **Bidi Properties** and select **Bidi transformation**.
 - b. Set properties for your environment.
5. To set logging properties for the external service wizard, perform the following steps:
 - a. Select the **Change logging properties for wizard** check box.
 - b. Change the location of the log file output location by clicking **Browse** and selecting a different location.
 - c. Set the **Logging Level**.

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

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

6. Click **Next**.

Results

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

Specify search criteria that the external service wizard uses to discover functions or data on the PeopleSoft Enterprise server.

Configuring the module for outbound processing

To configure a module to use the adapter for outbound processing, use the external service wizard in WebSphere Integration Developer to find and select business objects and services from the 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 external service wizard.

Before you begin

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

About this task

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

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 Properties window, specify configuration properties for the object. (This window is available only if you select the **Prompt for additional configuration settings when adding business objects** 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 93 (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. If you want the business object contained within a business graph, leave the check box selected.
2. Click Next.

Results

The Publishing Object Configuration Properties window is displayed.

What to do next

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

Setting deployment properties and generating the service

After you select and configure business objects for your module, you can use the external service wizard to configure properties that the adapter uses to connect to a specific enterprise information system. The wizard creates a new business integration module where all the artifacts and property values are saved.

Before you begin

Ensure that you have configured the business object.

About this task

Use the Service Generation and Deployment Configuration and Service Location Properties windows of the external service wizard to perform this task. The connection properties in this task are initialized to the values that the wizard used to connect to the enterprise information system. To configure the module to use other values, change the values here.

Procedure

1. In the Service Generation and Deployment Configuration window, click **Edit Operations** if you want to review or modify the names, or add a description for the operations to the business objects that you are creating. The following figure shows the Service Generation and Deployment Configuration window.

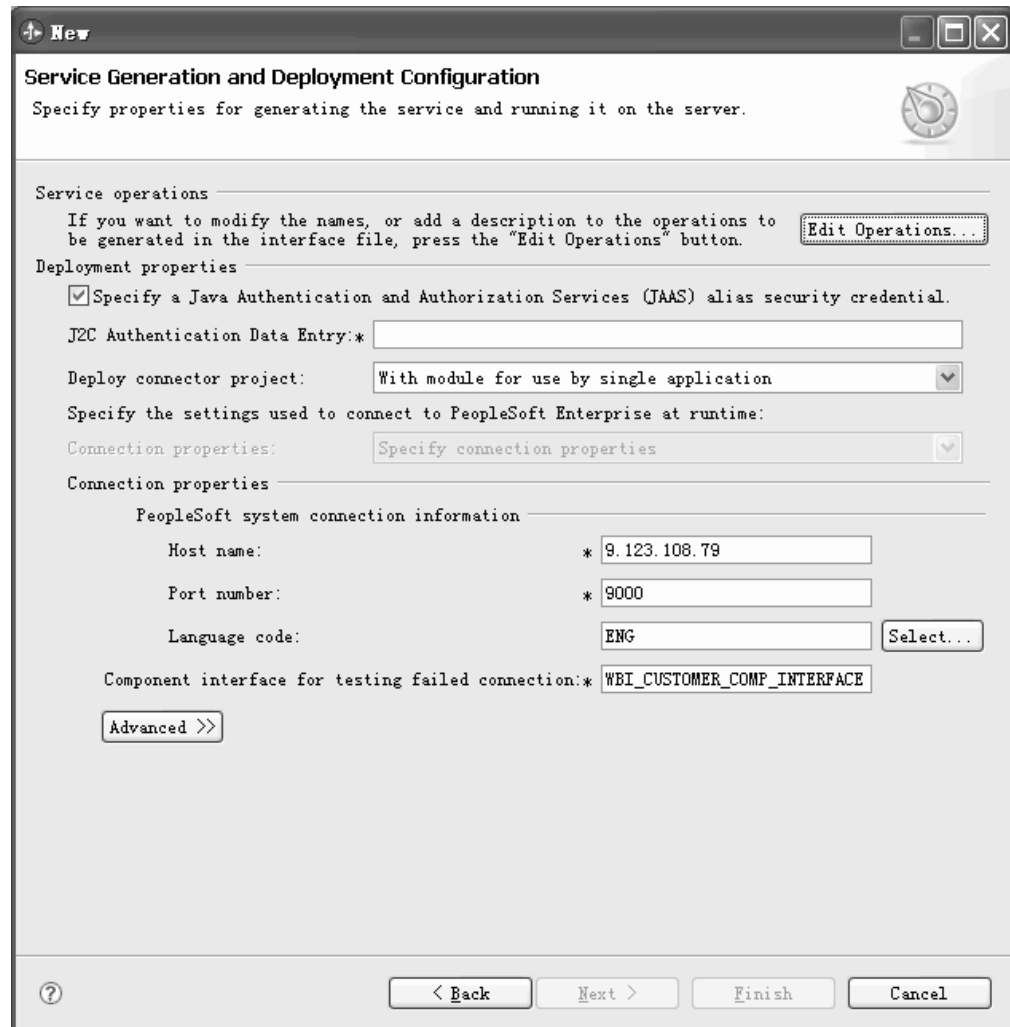


Figure 5. Service Generation and Deployment Configuration window

- a. In the Edit Operations Name window, select the operation you want to edit, and then click **Edit**.
- b. In the Add/Edit properties window, type a new operation name and description, and then click **Finish**.
2. Specify how you want the adapter to get the user name and password at run time. To use a J2C authentication alias, select **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** and type the name of the alias in **J2C Authentication Data Entry**. You can specify an existing authentication alias or create one at any time before deploying the module. The name is case-sensitive and includes the node name. If you did not create an authentication alias, leave this field blank.
3. In the **Deploy connector project** field, specify whether to include the adapter files in the module. Select one of the following values:
 - **With module for use by single application**. When the adapter files are embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules must run different versions of the adapter. You can use an embedded adapter to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications.** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
4. If you selected **On server for use by multiple applications** in the previous step, indicate how you want to specify the connection properties.
 - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same enterprise information system using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source. The following figure shows the **JNDI Lookup Name** field.
 - a. In **Connection properties**, select **Use predefined connection properties**.
 - b. In **JNDI Lookup Name**, type the name of the JNDI data source for an existing managed connection factory or activation specification.
 - c. Click **Next** to complete this task.

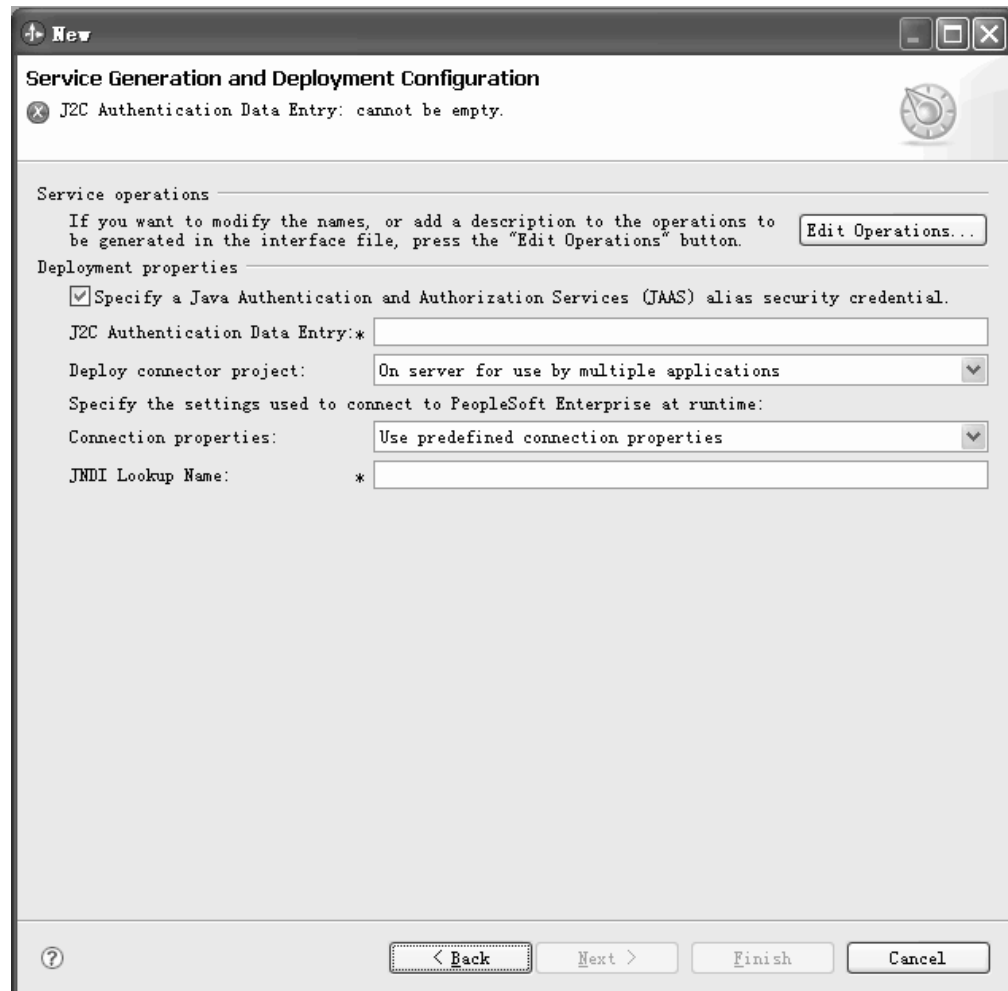


Figure 6. Specify a JNDI Lookup name

- If this application is the first that connects to the enterprise information system with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.
5. Review and, if necessary, change the values of the required connection properties. The **Host name**, **Port number**, and **Language code** fields are initialized with the connection information that you specified when you started the wizard. You can also connect to an alternate enterprise information system, although the schema names must be the same in both. The format of the connection properties is specific to the enterprise information system. For more information about the properties, see “Managed connection factory properties” on page 107.
 6. Optionally, you can change the values to specify a different user name and password at run time if you click **Advanced** and then enter the new values in those fields. The **User name** and **Password** field labels are shown italicized. The following figure shows the screen after you select **Advanced**.

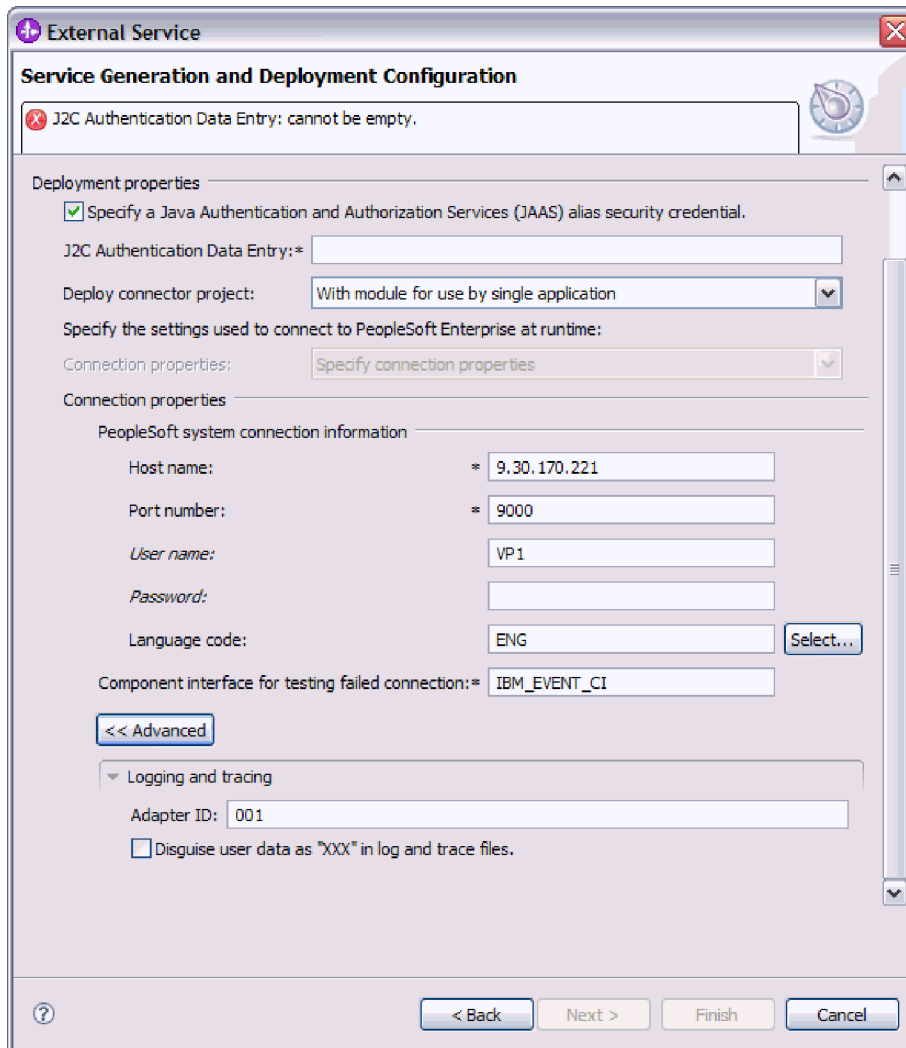


Figure 7. Specify a different user name and password at run time

7. Enter the name of the **Component interface for testing failed connection**.
8. After clicking **Advanced**, you can specify advanced options in the **Logging and tracing** area:
 - If you have multiple instances of the adapter, set Adapter ID to a value that is unique for this instance. For more information, see “Adapter ID (AdapterID)” on page 104.
 - You can select **Disguise user data as ‘XXX’ in log and trace files** if you want to prevent sensitive user data from being written to log and trace files. For more information, see “Disguise user data as “XXX” in log and trace files (HideConfidentialTrace) property” on page 109.
9. Click **Next**. The Service Location Properties window is displayed. The following figure shows the Service Location Properties window.

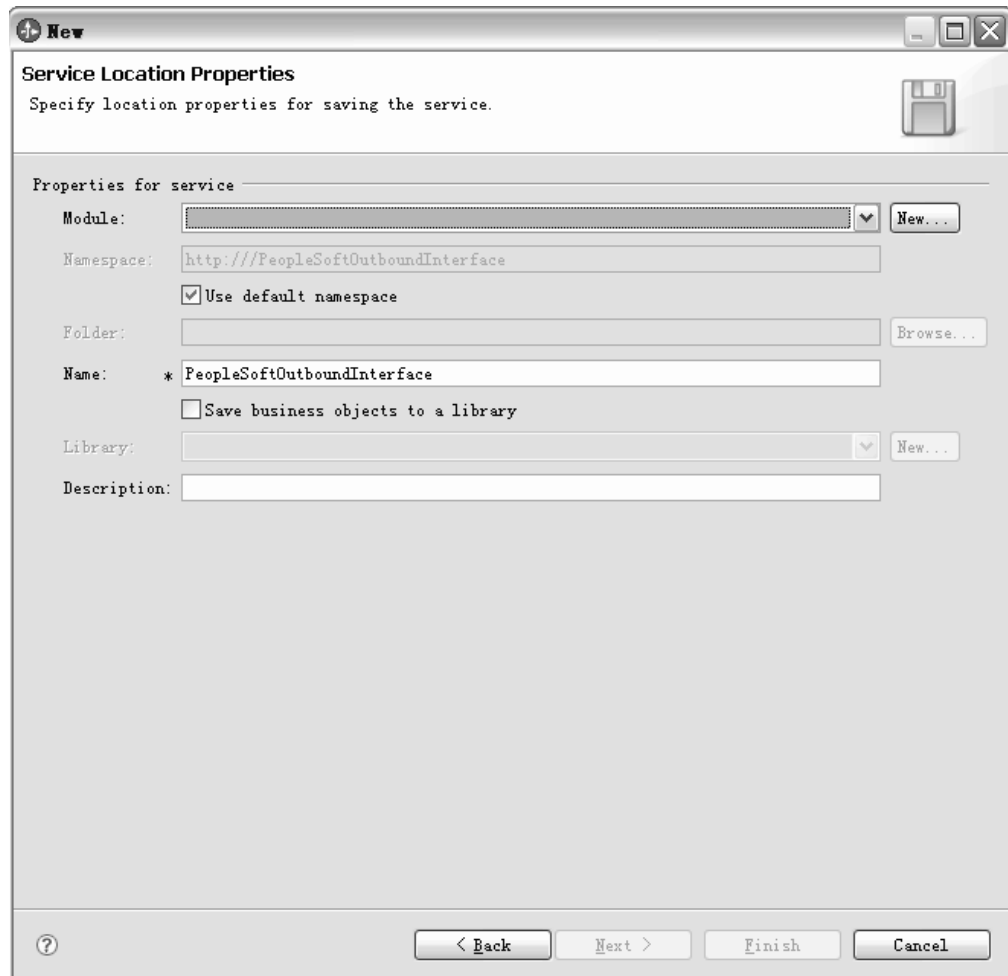


Figure 8. Service Location Properties window

10. In the Service Location Properties window, specify the name of the module that you want to create. This module can be new or existing.
 - If the module name is displayed in the **Module** list, select its name.

Important: If the module contains an interface or business object with the same name as any that you are now configuring, the original interface or business object in the module is replaced by the new version.
 - Otherwise, in the Service Location Properties, create a new module:
 - a. Click **New**.
 - b. In the Integration Project window, select **Create a module project** and click **Next**. The following figure shows the Integration Project window.

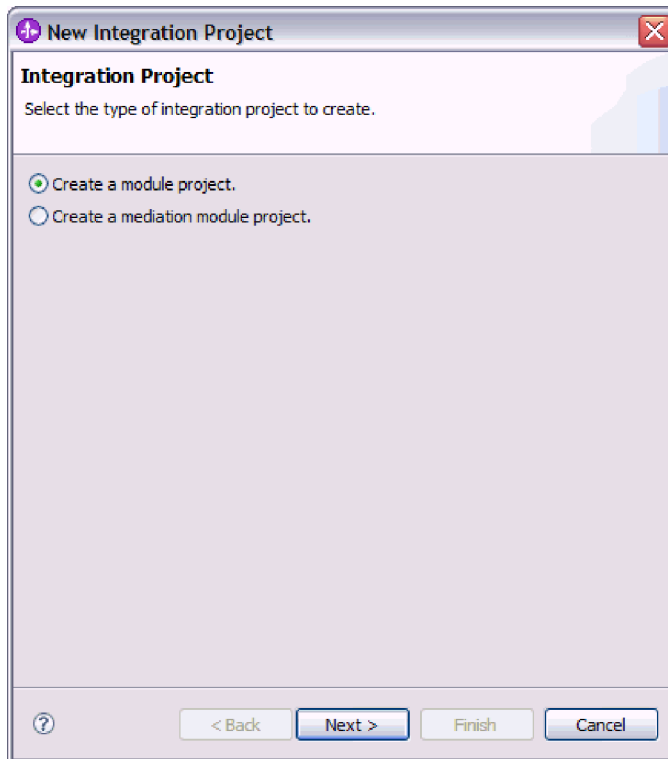


Figure 9. Create a module project in the Integration Project window

- c. In the Module window, type a name for the module. For example, PeopleSoftOutboundInterface.
 - d. If you want the service description files (the .import and .wsdl files) to be located in the default folder in the module, leave **Use default location** selected. If you want to specify a different folder in the module, clear the option and then click **Browse** to specify a different folder in **Location**.
 - e. If you want the module to be opened automatically in the assembly diagram in WebSphere Integration Developer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
 - f. Click **Finish** to create the new module.
11. Specify the namespace that you want to use for your business objects.
 - If you want the business objects in the module to use the default derived namespace, leave **Use default namespace** selected.
 - To specify a different namespace, clear the option and type a different value in **Namespace**.
 12. Optionally, specify the folder within the new module where the service description will be saved. In **Folder**, type the folder name or browse to an existing folder. If you do not specify a folder name, the artifacts (the import, XSD, and WSDL files) are stored in the root folder of the module, that is, the folder with the module name.
 13. In **Name**, accept the default import name or type a different name.
 14. Optionally, if you want to save the business objects in a library where they can be used by other modules, select **Save business objects to a library** and specify the location of the library in **Library**.
 15. Optionally, in **Description**, type a descriptive comment about the module.

16. When you are finished setting properties, click **Finish**.
17. If the Model Changed window is displayed, click **Yes**.

Results

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

What to do next

You might need to use the assembly editor to complete the configuration. Subsequently, you can test or deploy your module.

Configuring the module for inbound processing

To configure a module to use the adapter for inbound processing, use the external service wizard in WebSphere Integration Developer 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 external service wizard.

Before you begin

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

About this task

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

Procedure

1. In the Object Discovery and Selection window, click or expand the object you want to work with.
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 business objects** 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 93 (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 Objects 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 Objects 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.
 - d. If you want the business object contained within a business graph, leave the check box selected.
2. Click **Next**.

Results

The Publishing Object Configuration Properties window is displayed.

What to do next

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

Setting deployment properties and generating the service

After you select and configure business objects for your module, you can use the external service wizard to configure properties that the adapter uses to connect to a specific enterprise information system. The wizard creates a new business integration module where all the artifacts and property values are saved.

Before you begin

Ensure that you have configured the business object.

About this task

Use the Service Generation and Deployment Configuration and Service Location Properties windows of the external service wizard to perform this task. The connection properties in this task are initialized to the values that the wizard used to connect to the enterprise information system. To configure the module to use other values, change the values here.

Procedure

1. In the Service Generation and Deployment Configuration window, click **Edit Operations** if you want to review or modify the names, or add a description for the operations to the business objects that you are creating.

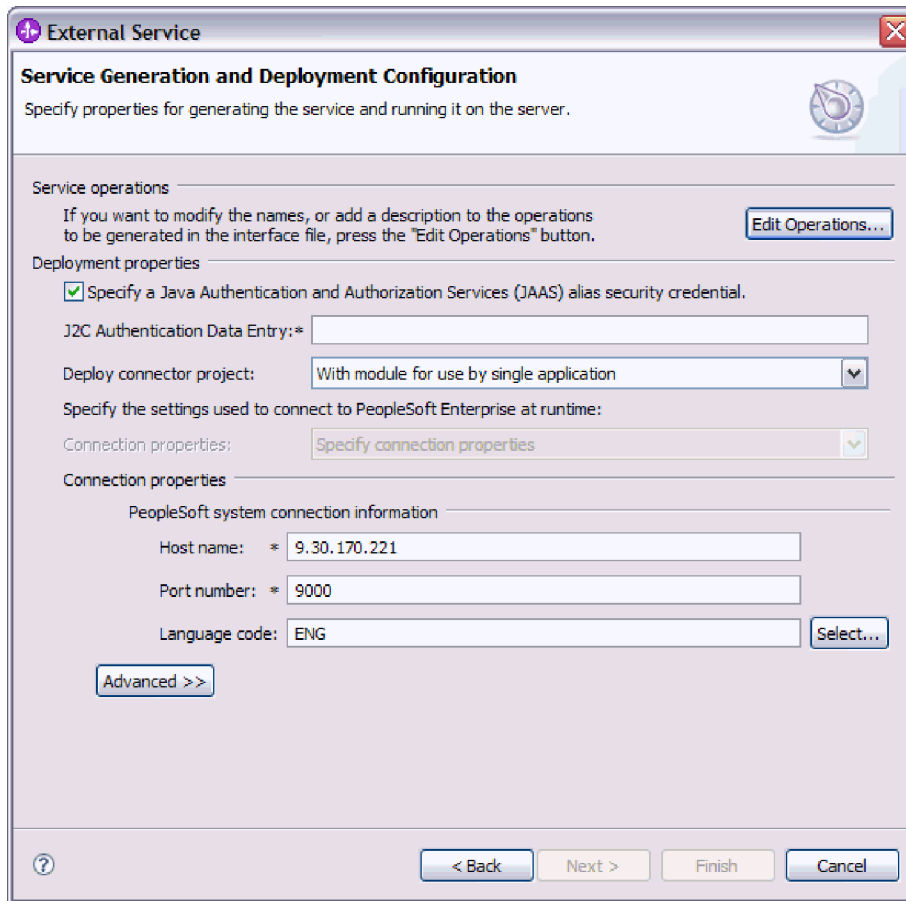


Figure 10. Service Generation and Deployment Configuration window

- a. In the Edit Operations Name window, select the operation you want to edit, and then click **Edit**.
 - b. In the Add/Edit properties window, type a new operation name and description, and then click **Finish**.
2. Specify how you want the adapter to get the user name and password at run time. To use a J2C authentication alias, select **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** and type the name of the alias in **J2C Authentication Data Entry**. You can specify an existing authentication alias or create one at any time before deploying the module. The name is case-sensitive and includes the node name. If you did not create an authentication alias, leave this field blank.
 3. In the **Deploy connector project** field, specify whether to include the adapter files in the module. Select one of the following values:
 - **With module for use by single application.** When the adapter files are embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules must run different versions of the adapter. You can use an embedded adapter to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
 - **On server for use by multiple applications.** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a

stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, indicate how you want to specify the connection properties.
 - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same enterprise information system using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source. The following figure shows the **JNDI Lookup Name** field.
 - a. In **Connection properties**, select **Use predefined connection properties**.
 - b. In **JNDI Lookup Name**, type the name of the JNDI data source for an existing managed connection factory or activation specification.
 - c. Click **Next** to complete this task.
 - If this application is the first that connects to the enterprise information system with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.
5. Review and, if necessary, change the values of the required connection properties. The **Host name**, **Port number**, and **Language code** fields are initialized with the connection information that you specified when you started the wizard. You can also connect to an alternate enterprise information system, although the schema names must be the same in both. The format of the connection properties is specific to the enterprise information system. For more information about the properties, see “Managed connection factory properties” on page 107.
6. Optionally, you can change the values to specify a different user name and password at run time if you click **Advanced** and then enter the new values in those fields. The **User name** and **Password** field labels are shown italicized. The following figure shows the screen after you select **Advanced**.

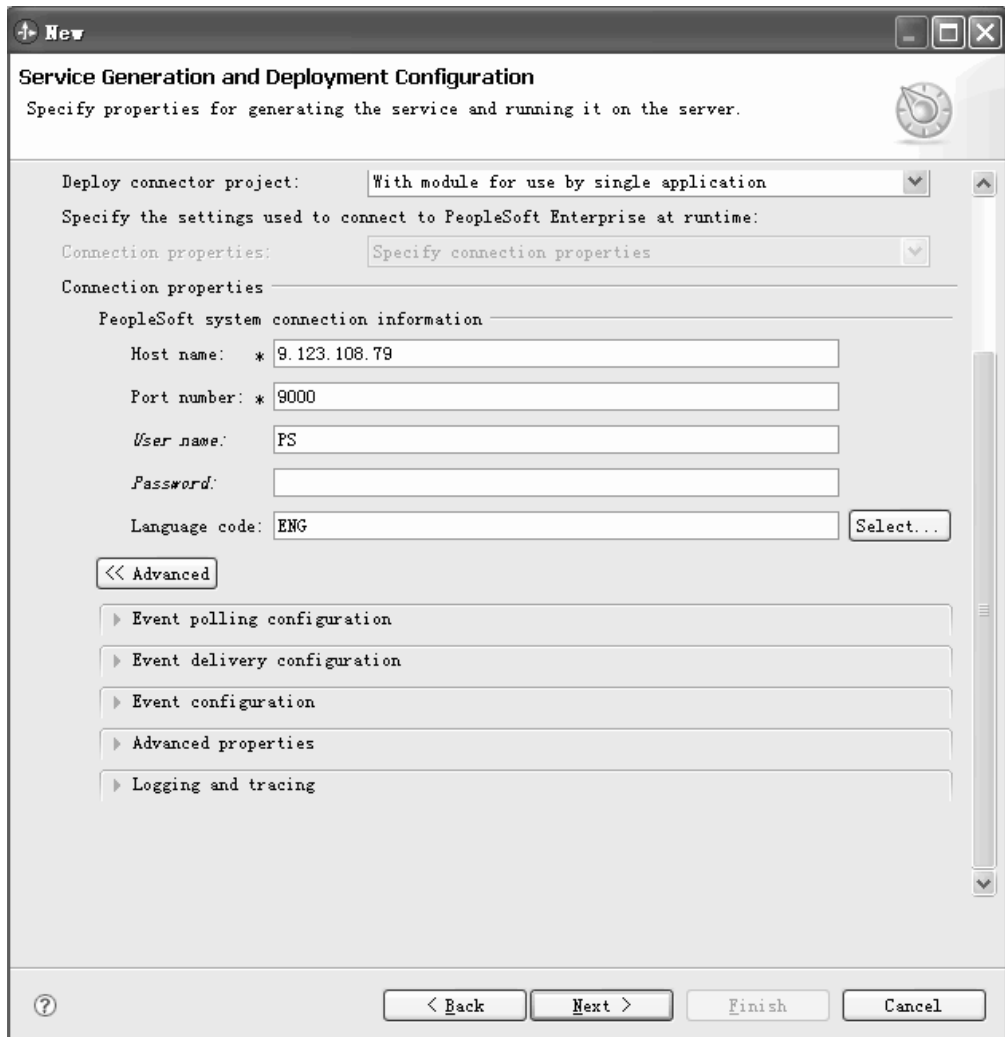


Figure 11. Change user name and password at run time

7. Each of the following areas under **Advanced** can be expanded to reveal additional fields:
 - Event polling configuration
 - Event delivery configuration
 - Event configuration
 - Advanced properties
 - Logging and tracing properties

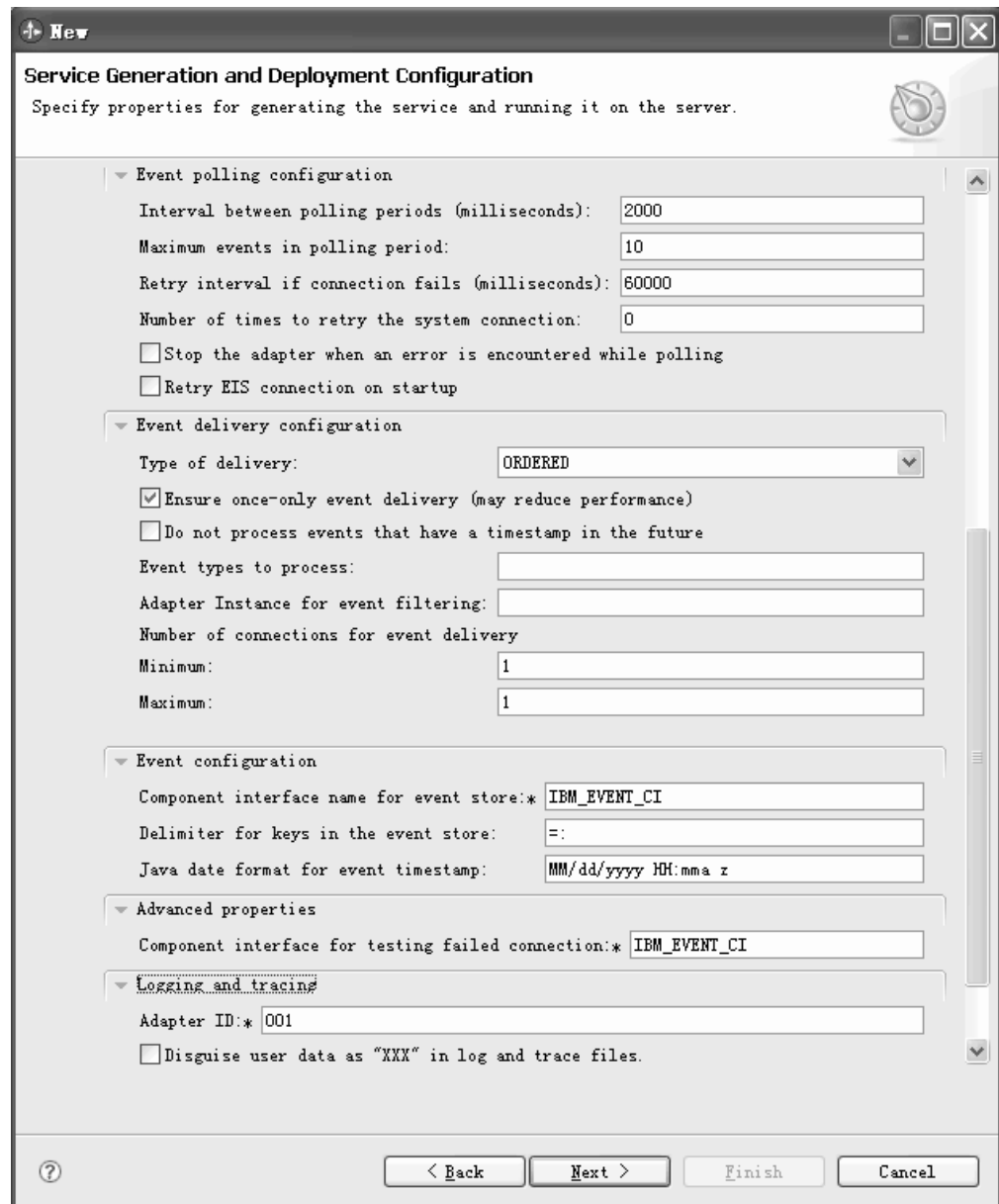


Figure 12. Advanced options for inbound processing

- **Event polling configuration**
 - a. In **Interval between polling periods**, type the number of milliseconds that the adapter waits between polling periods. For more information, see “Interval between polling periods (PollPeriod)” on page 127.
 - b. In **Maximum events in polling period**, type the number of events to deliver in each polling period. For more information, see “Maximum events in polling period (PollQuantity)” on page 127.
 - c. In **Retry interval if connection fails**, type the number of milliseconds to wait before trying to connect after a connection failure during polling. For more information, see “Retry interval if connection fails (RetryInterval)” on page 128.

- d. In **Number of times to retry the system connection**, type the number of times to retry the connection before reporting a polling error. For more information, see “Number of times to retry the system connection (RetryLimit)” on page 127.
 - e. If you want the adapter to stop if polling errors occur, select **Stop the adapter when an error is encountered while polling**. If you do not select this option, the adapter logs an exception but continues to run. For more information, see “Stop the adapter when an error is encountered while polling (StopPollingOnError)” on page 129.
 - f. Optionally, you can select **Retry EIS connection on startup** if you want the adapter to retry a failed connection when starting. For more information, see “Retry EIS connection on startup (RetryConnectionOnStartup)” on page 128.
- **Event delivery configuration**
 - a. In **Type of delivery**, select the delivery method. The methods are described in “Type of delivery (DeliveryType)” on page 124.
 - b. If you want to ensure that events are delivered only once and to only one export, select **Ensure once-only delivery**. This option might reduce performance but does not result in duplicate or missing event delivery. For more information, see “Ensure assured-only event delivery (AssuredOnceDelivery)” on page 122.
 - c. By default, the adapter processes all events that it finds when it polls. If you do not want it to process events that have timestamps later than the current time, select the **Do not process events that have a timestamp in the future** check box. For more information, see “Do not process events that have a timestamp in the future (FilterFutureEvents)” on page 124
 - d. In **Event types to process**, type a comma-separated list of the business objects for which you want events delivered. Leave this field blank to receive events for all business object types.

For example, if you want to receive events only when the Customer and Order tables, but not other tables, are changed in the enterprise information system, set this field to Customer,Order.

For more information, see “Event types to process (EventTypeFilter)” on page 125.
 - e. In **Adapter Instance for event filtering**, type a string value that determines whether this adapter instance processes specific events in the event store. For more information, see “Adapter instance for event filtering (AdapterInstanceEventFilter)” on page 121.
 - f. Under **Number of connections for event delivery**, specify the minimum and maximum number of connections to use to deliver events. For more information, see “Minimum connections (MinimumConnections)” on page 126 and “Maximum connections (MaximumConnections)” on page 126.
 - **Event configuration**
 - a. In **Component interface name for event store**, specify the component interface the adapter uses for event notification. For more information, see “Component interface name for event notification (EventCIName)” on page 123.
 - b. In **Delimiter for keys in the event store**, specify the name and value for an object key in the event table. For more information, see “Delimiter for keys in the event store (EventKeyDelimiter)” on page 124.

- c. In **Java date format for event timestamp**, specify the format used to create the event timestamp. For more information, see “Java date format for event timestamp (DateFormat)” on page 126.
- **Advanced properties**
In **Component interface for testing failed connection**, enter the name of the interface. For more information, see “Component interface for testing failed connection (PingCompInterface)” on page 123.
- **Logging and tracing properties**
 - a. If you have multiple instances of the adapter, set Adapter ID to a value that is unique for this instance. For more information, see “Adapter ID (AdapterID)” on page 104.
 - b. Select **Disguise user data as 'XXX' in log and trace files** if you want to prevent sensitive user data from being written to log and trace files. For more information, see “Disguise user data as “XXX” in log and trace files (HideConfidentialTrace) property” on page 109.
- 8. Click **Next**. The Service Location Properties window is displayed.
- 9. In the Service Location Properties window, create a new module.
 - a. Click **New** in the **Module** field.
 - b. In the Integration Project window, select **Create a module project** or **Create a mediation module project**, then click **Next**.
 - c. In the New Module window, type a name for the module.
 - d. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - e. Click **Finish**. The new module is created. When the creation process is finished, the New Module window closes, and the new module appears in the Module list in the Service Location Properties window.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Chapter 5. Changing interaction specification properties using the assembly editor

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

Before you begin

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

About this task

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

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

Procedure

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

Results

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

What to do next

Deploy the module.

Chapter 6. Deploying the module

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

Deployment environments

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

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

Deploying the module for testing

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

About this task

Adding external dependencies

Dependent JARs have to be added to the libraries directory or packaged in the EAR.

About this task

The JARs are set in the class path and these dependent libraries have to be made available for run time when the module is deployed. There are two ways to make the dependent libraries available, one for either stand alone deployment or embedded deployment and the other for embedded deployment only.

Adding external software dependencies on the server

The adapter needs its specific JAR files installed on the WebSphere Process Server or WebSphere Enterprise Service Bus server to be able to communicate with the PeopleSoft Enterprise server.

About this task

You do not need to perform this task if the PeopleSoft Enterprise server is installed on the same computer system as WebSphere Process Server or WebSphere Enterprise Service Bus. The files are already available to the adapter.

Procedure

1. Obtain the psjoa.jar dependency JAR file from the workstation on which the PeopleSoft server is installed and generate the other Java API jar file for Component Interfaces using the PeopleSoft tools. Depending on the version of PeopleSoft Enterprise you use, the software dependencies differ.

Note: See the PeopleSoft documentation for more information on using the PeopleSoft tools.

2. Copy the files to the server.
 - In a testing environment in WebSphere Integration Developer, copy the files to the `${WAS_INSTALL_ROOT}/runtimes/bi_v62/lib/ext` directory.
 - In a production environment, copy the files to the `${WAS_INSTALL_ROOT}/lib/ext` directory of WebSphere Process Server or WebSphere Enterprise Service Bus.

Adding external software dependencies when the adapter is bundled

You must copy the dependent JAR files to the EAR application before you can run your adapter applications. You must use this method only for embedded deployment.

About this task

To obtain the required files and copy them to the EAR application, use the following procedure:

Procedure

1. From the appropriate module, go to the workspace and copy the JAR files to the directory. For example if the name of the module is `ModuleName`, then go to the workspace and copy the JAR files to the `ModuleNameApp/EarContent` directory.
2. Modify the adapter RAR's manifest file, `manifest.mf`, with the list of JAR files required by the adapter. Add the JAR files in the following format : `Class-Path: dependantjar1.jar, dependantjar2.jar`
3. Copy the native libraries to the run time bin directory and deploy the application.

Results

The third party libraries are now a part of your run time environment

Generating and wiring a target component for testing inbound processing

Before deploying to the test environment a module that includes an adapter for inbound processing, you must first generate and wire a target component. This target component serves as the *destination* to which the adapter sends events.

Before you begin

You must have generated an export module, using the external service wizard.

About this task

Generating and wiring a target component for inbound processing is required in a testing environment only. It is not necessary when deploying the adapter in a production environment.

The target component receives events. You *wire* the export to the target component (connecting the two components) using the assembly editor in WebSphere Integration Developer. The adapter uses the wire to pass event data (from the export to the target component).

Procedure

1. Create the target component
 - a. From the Business Integration perspective of WebSphere Integration Developer, expand **Assembly Diagram** and double-click the export component. If you did not change the default value, the name of the export component is the name of your adapter + **InboundInterface**.

An interface specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions. The **InboundInterface** contains the operations required by the adapter to support inbound processing and is created when you run the external service wizard.
 - b. Create a new component by expanding **Components**, selecting **Untyped Component**, and dragging the component to the Assembly Diagram. The cursor changes to the placement icon.
 - c. Click the component to have it displayed in the Assembly Diagram.
2. Wire the components.
 - a. Click and drag the export component to the new component. This draws a wire from the export component to the new component, as shown in the following figure:
 - b. Save the assembly diagram. Click **File** → **Save**
3. Generate an implementation for the new component.
 - a. Right-click on the new component and select **Generate implementation**.

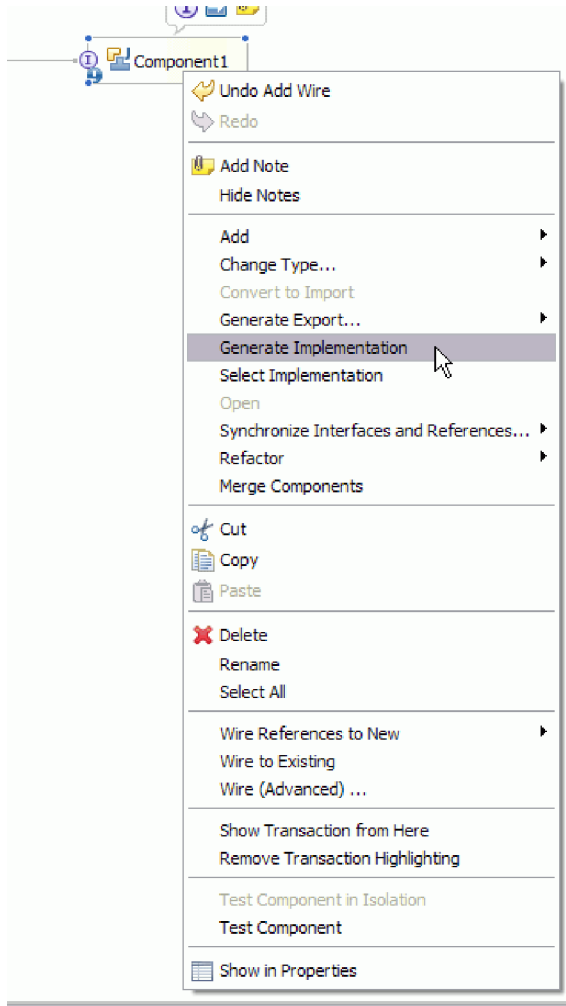


Figure 13. Generating a Java implementation

- b. Select **(default package)** and click **OK**. This creates an endpoint for the inbound module.
The Java implementation is displayed in a separate tab.
- c. **Optional:** Add print statements to print the data object received at the endpoint for each of the endpoint methods.
- d. Click **File** → **Save** to save the changes.

What to do next

Continue deploying the module for testing.

Adding the module to the server

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

Before you begin

If the module you are testing uses an adapter to perform inbound processing, you need to generate and wire a *target component* to which the adapter will send events.

About this task

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

Procedure

1. *Conditional:* If there are no servers in the **Servers view**, add and define a new server by performing the following steps:
 - a. Place your cursor in the **Servers view**, right click and select **New** → **server**
 - b. From the Define a New Server window, select the server type.
 - c. Configure server's settings.
 - d. Click **Finish** to publish the server.
2. Add the module to the server
 - a. Switch to the servers view. In WebSphere Integration Developer, select **Windows** → **Show View** → **Servers**
 - a. Start the server. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click on the server, and then select **Start**.
3. When the server status is *Started*, right-click on the server, and select **Add and remove projects**.
4. In the Add and Remove Projects screen, select your project and click **Add**. The project moves from the **Available projects** list to the **Configured projects** list.
5. Click **Finish**. This deploys the module on the server.

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

What to do next

Test the functionality of your module and the adapter.

Testing the module for outbound processing using the test client

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

Before you begin

You need to add the module to the server first.

About this task

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

Procedure

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

What to do next

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

Deploying the module for production

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

About this task

Adding external software dependencies on the server

The adapter needs its specific JAR files installed on the WebSphere Process Server or WebSphere Enterprise Service Bus server to be able to communicate with the PeopleSoft Enterprise server.

About this task

You do not need to perform this task if the PeopleSoft Enterprise server is installed on the same computer system as WebSphere Process Server or WebSphere Enterprise Service Bus. The files are already available to the adapter.

Procedure

1. Obtain the psjoa.jar dependency JAR file from the workstation on which the PeopleSoft server is installed and generate the other Java API jar file for Component Interfaces using the PeopleSoft tools. Depending on the version of PeopleSoft Enterprise you use, the software dependencies differ.

Note: See the PeopleSoft documentation for more information on using the PeopleSoft tools.

2. Copy the files to the server.
 - In a testing environment in WebSphere Integration Developer, copy the files to the `${WAS_INSTALL_ROOT}/runtimes/bi_v62/lib/ext` directory.
 - In a production environment, copy the files to the `${WAS_INSTALL_ROOT}/lib/ext` directory of WebSphere Process Server or WebSphere Enterprise Service Bus.

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

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

Before you begin

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

About this task

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

Procedure

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

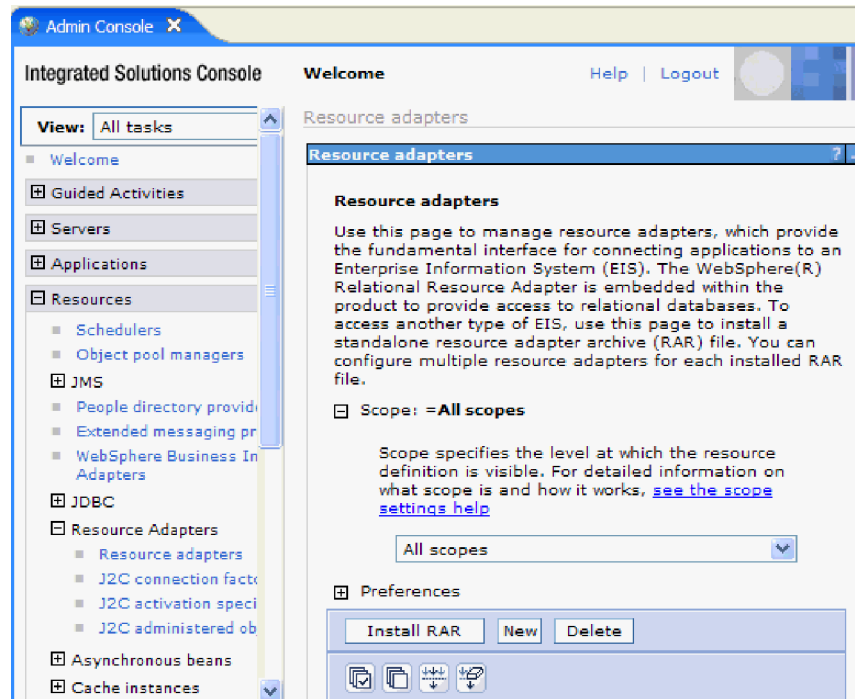


Figure 14. The Install RAR button on the Resource adapters page

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

The RAR files are typically installed in the following path:

WID_installation_directory/ResourceAdapters/adapter_name/deploy/adapter.rar

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

What to do next

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

Exporting the module as an EAR file

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

Before you begin

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

About this task

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

Procedure

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

Results

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

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

Installing the EAR file

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

Before you begin

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

About this task

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

Procedure

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

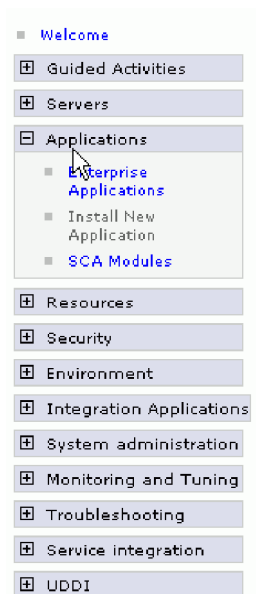


Figure 15. Preparing for the application installation window

3. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by "App."
4. Optional: If you are deploying to a clustered environment, complete the following steps.
 - a. On the **Step 2: Mapping modules to servers** window, select the module.
 - b. Select the name of the server cluster.
 - c. Click **Apply**.
5. Click **Next** to open the Summary. Verify that all settings are correct and click **Finish**.
6. Optional: If you are using an authentication alias, complete the following steps:
 - a. Expand **Security** and select **Business Integration Authentication Aliases**.
 - b. Select the authentication alias that you want to configure. You must have administrator or operator authority to make changes to authentication alias configurations.
 - c. Optional: If it is not already filled in, type the **User name**.
 - d. If it is not already filled in, type the **Password**.
 - e. If it is not already filled in, type the password again in the **Confirm Password** field.
 - f. Click **OK**.

Results

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

What to do next

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

Chapter 7. Administering the adapter module

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

Changing configuration properties for embedded adapters

To change configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment. You can update resource adapter properties (used for general adapter operation), 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 Process Server or WebSphere Enterprise Service Bus.

About this task

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

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

Procedure

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change. The Configuration page is displayed.

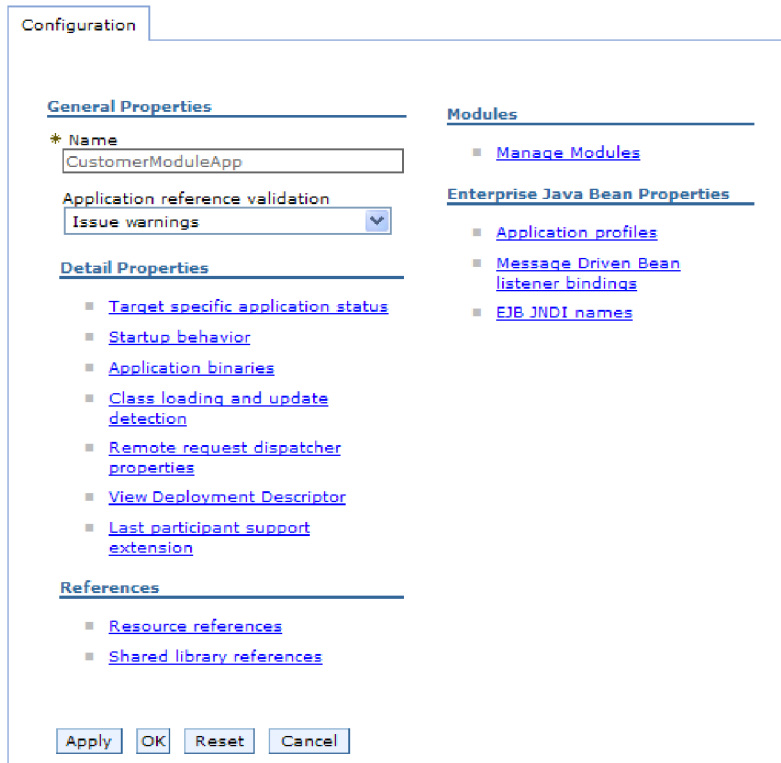


Figure 16. The Manage Modules selection in the Configuration tab

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 104 for more information about these properties.

- a. Click the name of the property. The **Configuration** page for the selected property is displayed. For example, if you click the **logNumberOfFiles** property, you see the page shown in the figure below.

The image shows a configuration dialog box titled "Configuration" with a tab labeled "Configuration". Under the "General Properties" section, there are several fields:

- * Scope:** A text box containing "widNode".
- Required:** An unchecked checkbox.
- Name:** A text box containing "logNumberOfFiles".
- Value:** A text box containing "1".
- Description:** An empty text area with scrollbars.
- Type:** A dropdown menu showing "java.lang.String".

 At the bottom of the dialog are four buttons: "Apply", "OK", "Reset", and "Cancel".

Figure 17. The Configuration tab for the logNumberOfFiles 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**.
9. Click the **Save** link in the **Messages** box at the top of the window.

Results

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

Setting managed (J2C) connection factory properties for embedded adapters

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

Before you begin

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

About this task

You use managed connection factory properties to configure the target 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**, select **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**.

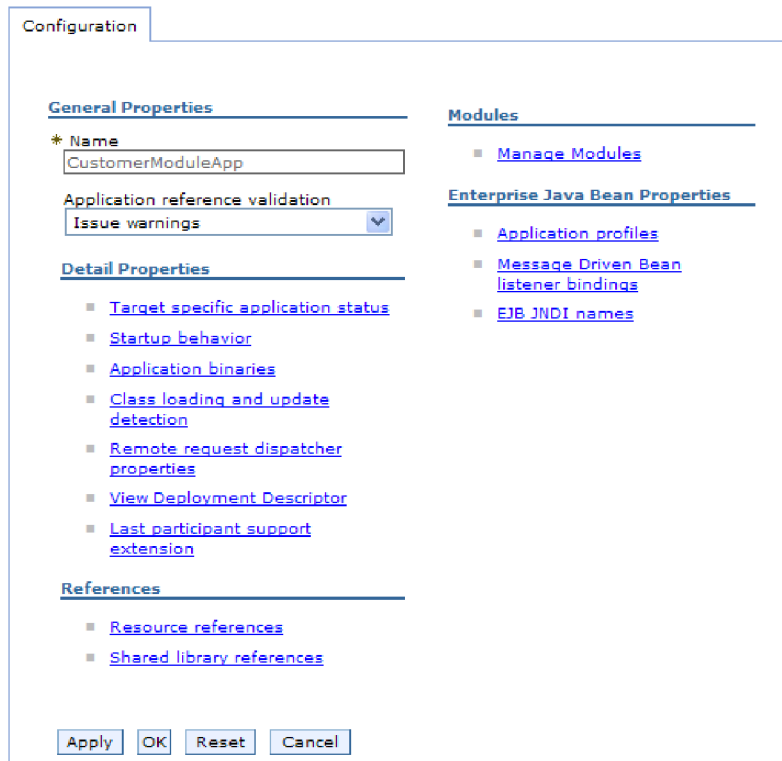


Figure 18. The Manage Modules selection in the Configuration tab

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 107 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 Process Server or WebSphere Enterprise Service Bus.

About this task

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

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

Procedure

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

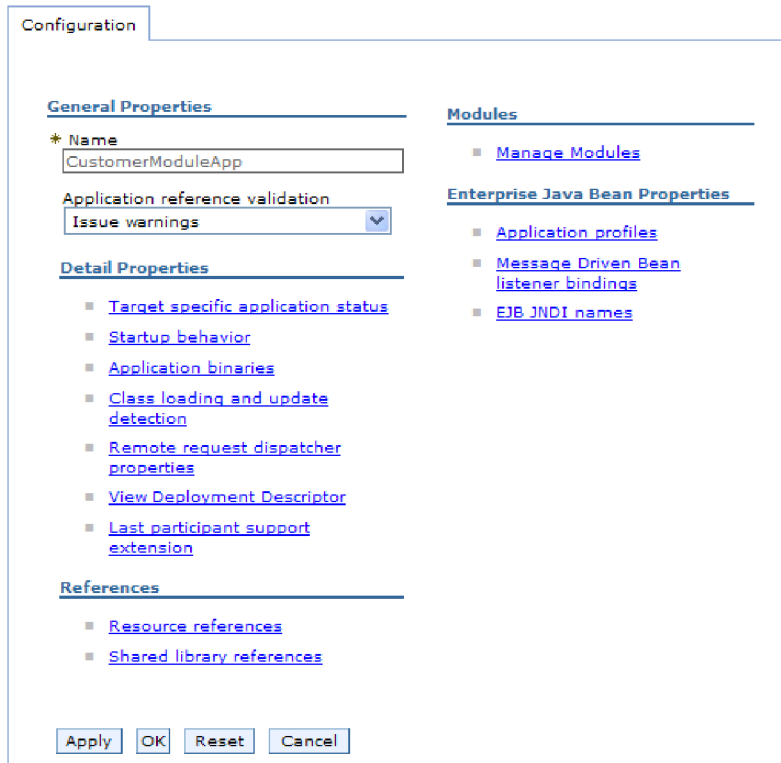


Figure 19. The Manage Modules selection in the Configuration tab

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 120 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 Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

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

About this task

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

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

Procedure

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. 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.

Note: See “Resource adapter properties” on page 104 for more information about these properties.

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

The image shows a configuration dialog box titled "Configuration" with a tab labeled "Configuration". Under the heading "General Properties", there are several fields:

- * Scope:** A text box containing "widNode".
- Required:** An unchecked checkbox.
- Name:** A text box containing "logNumberOfFiles".
- Value:** A text box containing "1".
- Description:** A large empty text area with scroll bars.
- Type:** A dropdown menu showing "java.lang.String".

 At the bottom of the dialog are four buttons: "Apply", "OK", "Reset", and "Cancel".

Figure 20. The Configuration tab for the logNumberOfFiles property

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

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

Results

The resource adapter properties associated with your adapter are changed.

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

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

Before you begin

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

About this task

You use managed connection factory properties to configure the target 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 external service wizard to configure the adapter module, you do not need to create a connection factory.

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

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

The newly created connection factory is displayed.

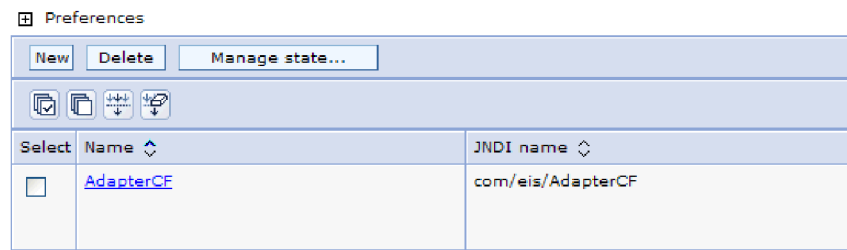


Figure 21. User defined connection factories for use with the resource adapter

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 107 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 Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

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

About this task

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

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

Procedure

1. 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 external service 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. 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.

Note: See “Activation specification properties” on page 120 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 page.

Results

The activation specification properties associated with your adapter are set.

Starting the application that uses the adapter

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

About this task

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

Procedure

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

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

2. Select the check box of the application that you want to start. The application name is the name of the EAR file you installed, without the .EAR file extension.
3. Click **Start**.

Results

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

Stopping the application that uses the adapter

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

About this task

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

stops when the application stops. For an application that uses a stand-alone adapter, the adapter stops when the application server stops.

Procedure

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

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

2. Select the check box of the application that you want to stop. The application name is the name of the EAR file you installed, without the .EAR file extension.
3. Click **Stop**.

Results

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

Monitoring performance using Performance Monitoring Infrastructure

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

About this task

You can monitor the performance of your adapter by having PMI collect data at the following points:

- At outbound processing to monitor outbound requests
- At inbound event retrieval to monitor the retrieval of an event from the event table
- At inbound event delivery to monitor the delivery of an event to the endpoint or endpoints

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

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

Configuring Performance Monitoring Infrastructure

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

Before you begin

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

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

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

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

For instructions on setting the trace level, refer to “Enabling tracing with the Common Event Infrastructure (CEI)” on page 83.

2. Generate at least one outbound request or inbound event to produce performance data that you can configure.

Procedure

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

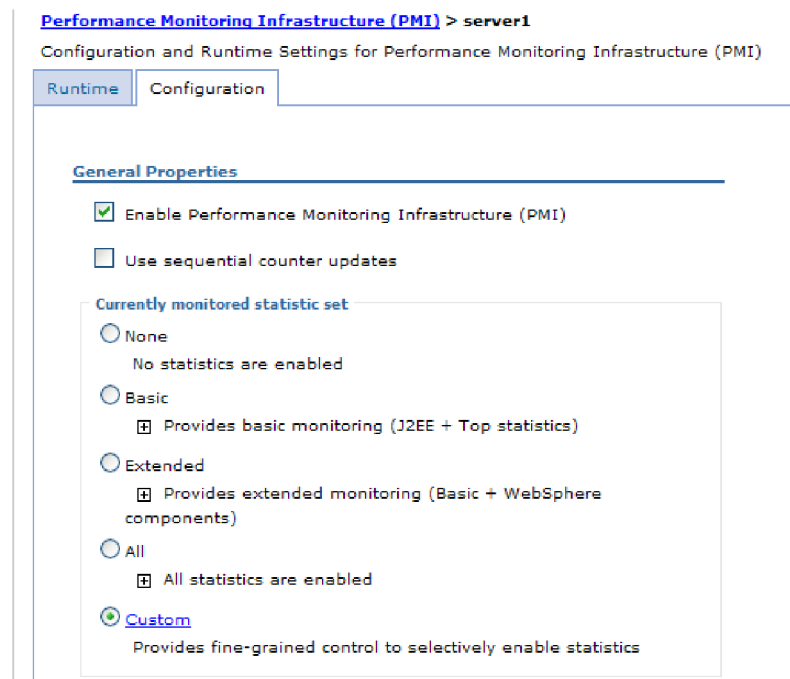


Figure 22. Enabling Performance Monitoring Infrastructure

- e. Click **Apply** or **OK**.
 - f. Click **Save**. PMI is now enabled.
2. Configure PMI for your adapter.
 - a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
 - b. From the list of servers, click the name of your server.
 - c. Select **Custom**.

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

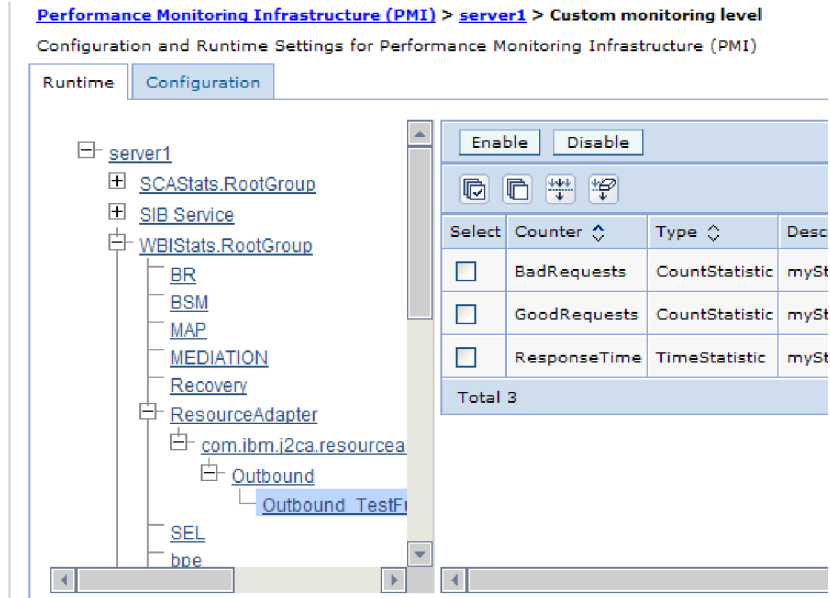


Figure 23. Runtime tab used for configuring PMI

- e. Click **WBISStats.RootGroup**. This is a PMI sub module for data collected in the root group. This example uses the name WBISStats for the root group.
- f. Click **ResourceAdapter**. This is a sub module for the data collected for the JCA adapters.
- g. Click the name of your adapter, and select the processes you want to monitor.
- h. In the right pane, select the check boxes for the statistics you want to gather, and then click **Enable**.

Results

PMI is configured for your adapter.

What to do next

Now you can view the performance statistics for your adapter.

Viewing performance statistics

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

Before you begin

Configure Performance Monitoring Infrastructure for your adapter.

Procedure

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

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

Results

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

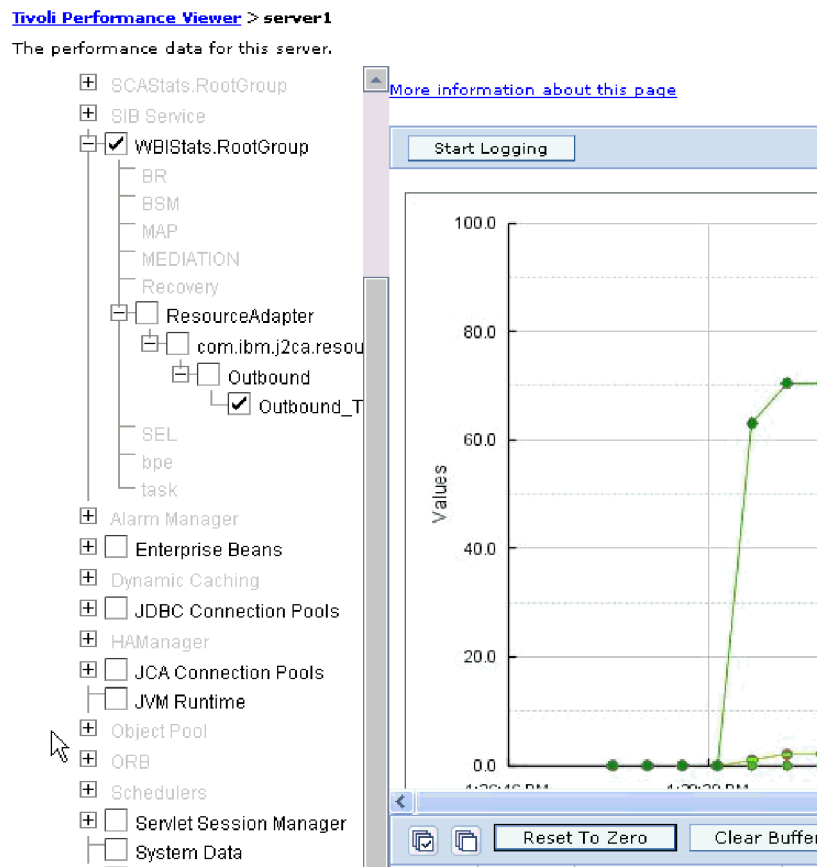


Figure 24. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure (CEI)

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

About this task

Procedure

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.

3. In the list of servers, click the name of your server.
4. In the **Change Log Detail Levels** box, click the name of the CEI database (for example, `WBIEventMonitor.CEI.ResourceAdapter.*`) or the trace log file (for example, `WBIEventMonitor.LOG.ResourceAdapter.*`) to which you want the adapter to write event data.
5. Select the level of detail about business events that you want the adapter to write to the database or trace log file, and (optionally) adjust the granularity of detail associated with messages and traces.
 - **No Logging.** Turns off event logging.
 - **Messages Only.** The adapter reports an event.
 - **All Messages and Traces.** The adapter reports details about an event.
 - **Message and Trace Levels.** Settings for controlling the degree of detail the adapter reports about the business object payload associated with an event. If you want to adjust the detail level, choose one of the following:
 - Fine.** The adapter reports the event but none of the business object payload.
 - Finer.** The adapter reports the event and the business object payload description.
 - Finest.** The adapter reports the event and the entire business object payload.
6. Click **OK**.

Results

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

Chapter 8. Troubleshooting and support

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

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 external service 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 104. For more information about the Log and Trace Analyzer, see <http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/org.eclipse.hyades.log.ui.doc.user/concepts/cltaviews.htm>.

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 process server. This is useful if you are trying to correlate traces from two different server processes, because it allows you to use the merge capability of the Log Analyzer.

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

You can change the log configuration statically or dynamically. Static configuration 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** → **Application Servers**.
2. Click the name of the server that you want to work with.
3. Under **Troubleshooting**, click **Logs and trace**.
4. Click **Change Log Detail Levels**.
5. Specify when you want the change to take effect:

- For a static change to the configuration, click the **Configuration** tab.
 - For a dynamic change to the configuration, click the **Runtime** tab.
6. Click the names of the packages whose logging level you want to modify. The package names for WebSphere Adapters start with **com.ibm.j2ca.***:
 - For the adapter base component, select **com.ibm.j2ca.base.***.
 - For the adapter base component and all deployed adapters, select **com.ibm.j2ca.***.
 - For the 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 process server.

Results

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

Changing the log and trace file names

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

Before you begin

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

About this task

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

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

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

Procedure

1. In the navigation pane of the administrative console, select **Applications > Enterprise Applications**.
2. In the Enterprise Applications list, click the name of the adapter application. This is the name of the EAR file for the adapter, but without the .ear file extension. For example, if the EAR file is named Accounting_OutboundApp.ear, then click **Accounting_OutboundApp**.
3. In the Configuration tab, in the Modules list, click **Manage Modules**.
4. In the list of modules, click IBM WebSphere Adapter for 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 Process Server or WebSphere Enterprise Service Bus.

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

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

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

Business faults

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

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

Fault business objects

The external service wizard creates a business object for each fault that the adapter can generate. In addition, the wizard creates a WBIFault superset business object, which has information common to all faults, such as the message, errorCode, and primaryKeySet attributes as shown in Figure 25.

WBIFault	
message	string
errorCode	string
primaryKeySet	PrimaryKeyPairType []

Figure 25. The structure of the WBIFault business object

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

The WebSphere Adapter for PeopleSoft Enterprise enables any faults for you. Manual configuration of faults is not required. The adapter provides the following fault business objects that the wizard creates:

- DuplicateRecordFault
When processing a Create operation, the adapter throws this fault if a duplicate record is found.
- MatchesExceededLimitFault
When processing a RetrieveAll operation, the adapter throws this fault if the number of records returned from the database query exceeds the value for the MaxRecords property in the interaction specification.
To increase the number of records that are returned, increase the value of the MaxRecords property in the interaction specification properties for the RetrieveAll operation.
The business object for this fault has one property, matchCount, which is a string that contains the number of matches.
- MissingDataFault

If the business object that is passed to the outbound operation does not have all the required attributes, the adapter throws this fault. This fault can occur for the Create, Delete, Update, Retrieve, ApplyChanges and Exists operations.

- **MultipleMatchingRecordsFault**

When processing a Retrieve or Update operation, the adapter throws this fault if the query returns more than one record for the keys specified. The business object for this fault has one property, `matchCount`, which is a string that contains the number of matches.

- **RecordNotFoundFault**

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

Resolving memory limit exception during outbound processing

A memory limit exception encountered during outbound processing can be resolved by increasing the JVM maximum heap memory size.

About this task

When concurrently processing fifty outbound requests, the adapter may produce an out of memory error. To avoid or resolve this error, increase the JVM maximum heap default value.

Procedure

1. From the administrative console, click on the name of your server.
2. In the configuration tab, under Server Infrastructure, click **Java and Process Definition>Process Definition**.
3. Click **Java Virtual Machine**.
4. In the **Maximum Heap Size** field, type 1024.
5. Click **OK**.
6. Click **Save**.
7. Log out, and then restart the server.

Results

The maximum heap size default value is increased.

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 white papers
- 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 9. Reference information

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

Business object information

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

The adapter for 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 4. 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> .

Table 4. Metadata for business objects (continued)

Metadata element	Description
InstAtCurrentEffDtPos	When set to true and an incorrect index (negative number) is returned by the PeopleTools API, getEffectiveItemNum(), the adapter inserts the row at the highest index (zero), which has the current effective date. If you set this parameter, do not set InsAtOldestEffDtPos.
StatusColumnName	A property used to perform deletes. Set this for the top-level business object only.
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 5. 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 6. 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.
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 &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; /*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 external service wizard while generating or creating objects and services. You can change the resource adapter and managed connection factory properties after you deploy the module to WebSphere Process Server or WebSphere Enterprise Service Bus using WebSphere Integration Developer or the administrative console, but connection properties for the external service 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 external service wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the external service wizard <i>will not change that default value</i>. When a required field contains no value at all, the external service wizard 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 external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
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 external service wizard

Connection properties for the external service wizard are used to establish a connection between the external service 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 external service 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 7. External service connection properties

Property name	Description
"Component interface JAR file" on page 102	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 102	Specifies the name or address of the server that hosts PeopleSoft Enterprise.
"Password " on page 102	Specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.

Table 7. External service connection properties (continued)

“Port number” on page 103	The port number at which PeopleSoft Enterprise is configured to listen for client requests.
“Prefix for business object names” on page 103	Specifies a prefix to be added to generated business objects.
“User name” on page 103	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 8. 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 9. 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 10. Password details

Required	Yes
Default	No default value
Property type	String

Table 10. 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 11. Port number details

Required	Yes
Default	The port number entered when you run the external service 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 12. 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 13. 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 13. 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 external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

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

The 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 14. 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 15. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 15. 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 external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace) 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 16. 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 external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No

Table 16. Disguise user data as "XXX" in log and trace files details (continued)

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

Enable high availability support (enableHASupport)

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

Log file maximum size (LogFileMaxSize)

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

Table 17. 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 18. 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 19. 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 20. 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 21. 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 22. 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 external service 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 external service wizard refers to these properties as managed connection factory properties and the administrative console refers to them as (J2C) connection factory properties.

Table 23. 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 110	Language	Specifies the language code the adapter uses to log on to the PeopleSoft Enterprise server.
"Password (Password)" on page 111	Password	Specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.
"Port number (Port)" on page 111	Port	The port number the adapter uses to access the PeopleSoft Enterprise server.
"User name (UserName)" on page 111	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 24. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 24. 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 external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.</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 25. 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 26. 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 external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No
Bidi supported	No

Host name (HostName)

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

Table 27. 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 28. 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 29. 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 30. Port number details

Required	Yes
Default	The port number entered when you use the external service 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 31. 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 external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, some properties for outbound operations can be changed by the user.

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 WebSphere Integration Developer. The properties reside in the method binding of the import.

Table 32 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 100.

Table 32. 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 33. 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 external service wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using WebSphere Integration Developer or the administrative console, but connection properties for the external service 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 external service wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the external service wizard <i>will not change that default value</i>. When a required field contains no value at all, the external service wizard 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 external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
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 external service wizard

Connection properties for the external service wizard are used to establish a connection between the external service 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 external service 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 34. External service 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 " on page 115	Specifies the name or address of the server that hosts PeopleSoft Enterprise.
"Password " on page 115	Specifies the password of the user account of the adapter on the PeopleSoft Enterprise server.
"Port number" on page 115	The port number at which PeopleSoft Enterprise is configured to listen for client requests.
"Prefix for business object names" on page 115	Specifies a prefix to be added to generated business objects.
"User name" on page 116	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 35. 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 36. 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 37. 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 38. Port number details

Required	Yes
Default	The port number entered when you run the external service 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 39. Prefix details

Required	No
Default	No default

Table 39. Prefix details (continued)

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 40. 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 external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

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

The 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 41. 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

Table 41. Resource adapter properties for the Adapter for PeopleSoft Enterprise (continued)

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

Adapter ID (AdapterID)

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

Table 42. 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 external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace) 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 43. 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 external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.</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 44. 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 45. Log file name details

Required	No
Default	No default value

Table 45. Log file name details (continued)

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 46. 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 47. 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 48. 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 49. 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 external service 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 100.

Table 50. 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

Table 50. Activation specification properties (continued)

Minimum connections	MinimumConnections	The minimum number of connections that the adapter can use for inbound event delivery
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 51. Adapter instance for event filtering details

Required	No
Default	null
Property type	String

Table 51. Adapter instance for event filtering details (continued)

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 WBIA 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 52 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>
Example	See Table 52.
Globalized	Yes
Bidi supported	Yes

Table 52. 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 assured-only event delivery (AssuredOnceDelivery)

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

Table 53. Ensure assured-only event delivery details

Required	Yes
Possible values	True False
Default	True

Table 53. Ensure assured-only event delivery details (continued)

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 (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 54. Component interface for testing failed connection details

Row	Explanation
Required	Yes
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 55. 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

Type of delivery (DeliveryType)

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

Table 56. Type of delivery 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 57. 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 58. 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.

Table 58. Do not process events that have a timestamp in the future details (continued)

Globalized	No
Bidi supported	No

Event types to process (EventTypeFilter)

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

Table 59. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types
Default	null
Property type	String
Usage	Events are filtered by business object type. If the property is set, the adapter delivers only those events that are in the list. A value of null indicates that no filter will be applied and that all events will be delivered to the export.
Example	To receive 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 60. 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

Table 60. Retry limit for failed events details (continued)

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

Java date format for event timestamp (DateFormat)

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

Table 61. 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 62. 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 63. 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 64. 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 65. 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 66. 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.

Table 66. Number of times to retry the system connection details (continued)

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 67. Retry EIS connection on startup details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>This property indicates whether the adapter should retry the connection to the 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 68. 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 69. Stop the adapter when an error is encountered while polling details

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

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

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

Bidirectional script data transformation

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

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

Table 70. 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 WebSphere Integration Developer and WebSphere Process Server or WebSphere Enterprise Service Bus.

Connection properties for the external service 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

Adapter messages

View the messages issued by WebSphere Adapter for PeopleSoft Enterprise at the following location.

Link to messages: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.wbit.620.help.messages.doc/messages.html>

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

- Messages with the prefix CWYES are issued by WebSphere Adapter for PeopleSoft Enterprise
- Messages with the prefix CWYBS are issued by the adapter foundation classes, which are used by all the adapters

Related information

The following information centers, IBM Redbooks, and Web pages contain related information for the WebSphere Adapter for PeopleSoft Enterprise.

Samples and tutorials

To help you use WebSphere Adapters, samples and tutorials are available from the Business Process Management Samples and Tutorials Web site. You can access the samples and tutorials in either of the following ways:

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

Information resources

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

Information about related products

- WebSphere Business Process Management, version 6.2, information center, which includes WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Integration Developer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp>
- WebSphere Adapters, version 6.1.x, information center: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.612.help.adapter.emd.ui.doc/topics/tcreatecmps.html>
- WebSphere Business Integration Adapters information center: http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbi_adapters.doc/welcome_adapters.htm

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

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

Notices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Programming interface information

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

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

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

Warning:

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

Trademarks and service marks

IBM, the IBM logo, 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 120
 - setting in administrative console 73, 78
- adapter application
 - starting 79
 - stopping 79
- adapter architecture 2
- Adapter for PeopleSoft Enterprise
 - administering 69
- Adapter for PeopleSoft Enterprise module
 - exporting as EAR file 66
 - installing EAR file on server 66
 - starting 79
 - stopping 79
- Adapter implementation 9
- adapter messages 131
- adapter performance 80
- adapter project, creating 33
- adapter technotes 132
- application-specific information 93
- authentication
 - description 10
 - external service wizard 10
 - run time 11
- authentication alias 11, 32

B

- backward compatibility
 - project interchange files 18
 - projects 18
- business faults 89
- business integration adapters to JCA-compliant adapters 19
- business object information 93
- business objects 5
 - supported data operations 94
- business-object-level metadata 93

C

- CEI (Common Event Infrastructure) 83
- clustered environment
 - deploying in 14
 - description 14
 - inbound processes 15
 - outbound processes 15
- Common Event Infrastructure (CEI) 83
- compatibility matrix 2
- confidential data, disguising 9
- confidential tracing 9
- configuration overview 30
- configuring
 - logging 86
 - Performance Monitoring Infrastructure (PMI) 80
 - tracing 86
- custom properties
 - activation specification 73, 78
 - managed connection factory 71, 76
 - resource adapter 69, 75

D

- data operations 94
- debugging
 - self-help resources 90
- deployment
 - environments 59
 - options 11
 - to production environment 64
 - to test environment 59
- developerWorks 132
- developerWorks resources, WebSphere Adapters 131

E

- EAR file
 - exporting 66
 - installing on server 66
- education, WebSphere Adapters 131
- embedded adapter
 - activation specification properties, setting 73
 - considerations for using 13
 - description 11
 - managed connection factory properties, setting 71
 - resource adapter properties, setting 69
- embedded deployment 60
- enableHASupport property 15
- event store 4
- exporting module as EAR file 66
- external dependencies, adding 34, 59, 60
- External service connection properties 101, 114
- external service wizard
 - authentication in 10

F

- faults
 - description 89
- FFDC (first-failure data capture) 88
- files
 - SystemOut.log log file 87
 - trace.log trace file 87
- first-failure data capture (FFDC) 88

H

- hardware and software requirements 2
- hardware requirements 2
- high-availability environment
 - deploying in 14
 - description 14
 - inbound processes 15
 - outbound processes 15

I

- IBM WebSphere Adapter Toolkit 132
- implementation, Java 61
- importing into project 34
- inbound configuration properties 112

- installing EAR file 66
- interaction specification properties
 - changing 57
- interaction specification property 112

J

- JAR file, adding external 34, 59, 60
- Java 2 security 11
- Java implementation 61

L

- Log Analyzer 86
- Log and Trace Analyzer, support for 85
- log and trace files 85
- log files
 - changing file name 87
 - disabling 86
 - enabling 86
 - level of detail 86
 - location 88
- logging
 - configuring properties with administrative console 86

M

- managed (J2C) connection factory properties
 - setting in administrative console 71, 76
- matrix, compatibility 2
- messages, adapter 131
- metadata 93
 - business-object-level 93
 - property-level 94
- migration 19
 - WebSphere InterChange Server migration wizard 22
- migration considerations 16
- migration overview
 - WebSphere InterChange Server applications 20
- monitoring performance 80

O

- outbound configuration properties 100

P

- package files for adapters 87
- Performance Monitoring Infrastructure (PMI)
 - configuring 80
 - description 80
 - viewing performance statistics 82
- performance statistics 82
- PMI (Performance Monitoring Infrastructure)
 - configuring 80
 - description 80
 - viewing performance statistics 82
- polling 4
- problem determination
 - self-help resources 90
- project interchange (PI) file
 - updating without migrating 18
- properties
 - activation specification 73, 78
 - list of 120

- properties (*continued*)
 - configuration properties
 - inbound 112
 - outbound 100
 - inbound configuration 112
 - managed (J2C) connection factory 71, 76
 - outbound configuration 100
 - resource adapter 69, 75
- property-level metadata 94

R

- RAR (resource adapter archive) file
 - description 64
 - installing on server 64
- RAR (resource adapter) file
 - importing 33
- Redbooks, WebSphere Adapters 131
- related information 131
- related products, information 131
- requirements, hardware and software 2
- resource adapter (RAR) file
 - importing 33
- resource adapter archive (RAR) file
 - description 64
 - installing on server 64
- resource adapter properties
 - details 104, 116
 - setting in administrative console 69, 75
- Retry limit property 127
- road map for configuring the module 29
- road map for migrating
 - WebSphere InterChange Server applications 20
- runtime environment
 - authentication in 11
 - deploying EAR file to 64

S

- samples 27
- security
 - disguising sensitive data 9
- Security 9
- security, Java 2 11
- self-help resources 90
- sensitive data, disguising 9
- software dependencies, adding external 34, 59, 60
- software requirements 2
- stand-alone adapter
 - activation specification properties, setting 78
 - considerations for using 14
 - description 11
 - managed connection factory properties, setting 76
 - resource adapter properties, setting 75
- starting adapter applications 79
- stopping adapter applications 79
- support
 - overview 85
 - self-help resources 90
 - technical 132
- supported data operations 94
- SystemOut.log file 87

T

- target component 60
- Technical overview 2
- technical support 132
- technotes 2, 90, 132
- technotes, WebSphere Adapters 131
- test environment
 - adding module to 62
 - deploying to 59, 62
 - testing modules 63
- trace files
 - changing file name 87
 - disabling 86
 - enabling 86
 - level of detail 86
 - location 88
- trace.log file 87
- tracing
 - configuring properties with administrative console 86
- troubleshooting
 - overview 85
 - self-help resources 90
- tutorials 27

W

- WebSphere Adapters, version 6.0, information 132
- WebSphere Adapters, version 6.0.2.x, information 132
- WebSphere Application Server information 132
- WebSphere business integration adapters 19
- WebSphere Business Integration Adapters information 132
- WebSphere Business Process Management, version 6.1.x, information 132
- WebSphere Enterprise Service Bus information 132
- WebSphere Extended Deployment 14
- WebSphere Integration Developer
 - information 132
 - test environment 59
- WebSphere Process Server
 - information 132
- WebSphere Process Server or WebSphere Enterprise Service Bus
 - deploying to 64
- wiring components 60



Printed in USA