

WebSphere Adapters



WebSphere Adapter for PeopleSoft Enterprise Version 6.0 User Guide

Version 6.0

Note

Before using this information, be sure to read the general information in "Notices" on page 55.

7April2006

This edition of this document applies to Adapter for PeopleSoft Enterprise (Product number 5724L81), Version 6.0, and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

© Copyright International Business Machines Corporation 2005, 2006. All rights reserved.

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

Contents

WebSphere Adapter for PeopleSoft Enterprise, Version 6.0 User Guide	1
Product overview	1
Audience	1
IBM WebSphere Adapters	1
Task road map: IBM WebSphere Adapter for PeopleSoft Enterprise	3
Enterprise service discovery	4
How the WebSphere Adapter for PeopleSoft Enterprise works	4
Locale and globalization support	6
Learning about business objects	8
Business object structure	8
Attribute properties	8
Application-specific information	9
Attribute application-specific information	10
Supported verbs	11
Supported operations	11
Installing the adapter	14
Adapter environment	14
Installed file structure	14
Creating a custom event project in PeopleTools	15
Deploying the adapter	17
Creating a project	17
Adding external dependencies	18
Creating a business integration module	21
Generating reference bindings	27
Creating an authentication alias	27
Exporting the EAR file	28
Deploying the adapter project	28
Configuring the adapter on the server	28
Troubleshooting	29
Contacting IBM Software Support	29
Enabling logging	31
Enabling tracing	32
Enabling the Common Event Infrastructure (CEI)	32
Troubleshooting tips	33
Using the sample applications	34
Structure of the application package	35
Business object structure	37
Deploying and configuring for scenario 1	37
Deploying and configuring for scenario 2	41
References	42
PeopleCode for custom event project	43
Connection properties	47
Resource adapter properties	48
J2C connection factory properties	50
Activation Specification properties	51
EDT configuration information	53
Notices	55
Programming interface information	57
Trademarks and service marks	57

WebSphere Adapter for PeopleSoft Enterprise, Version 6.0 User Guide

The IBM® WebSphere® Adapter for PeopleSoft Enterprise facilitates the exchange of business objects between PeopleSoft systems and J2EE-based programming models.

Product overview

The IBM WebSphere Adapter for PeopleSoft Enterprise facilitates the exchange of business objects between PeopleSoft Enterprise applications and J2EE based application components.

Audience

The information in this topic defines the users of the WebSphere Adapter products and details the skills they require.

The audience for the adapter user guide includes data and application integrators who are responsible for assembling application components into a complete solution and preparing this solution for testing and deployment. These users require the following general skills:

- A good understanding of the business solution and the business environment
- Knowledge of application and solution components, to enable their efficient collaboration at run-time
- A detailed understanding of databases, data access issues, transactional models and connections across heterogeneous relational databases, queues, and web services
- Familiarity with integration tools

The application integrator is also responsible for detailed testing activities and needs these additional skills:

- Creating required scripts, tools, and templates for testing and deployment
- Creating integration workspaces and integrating systems & subsystems
- Resolving interdependencies between entities such as Enterprise Java™ Beans (EJBs), workflows, and web pages
- Validating the application or solution

The data integrator is also responsible for enabling access to a range of data sources for the application developers. The required skills include:

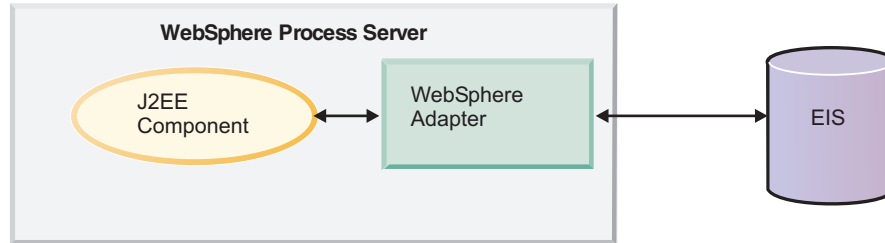
- Installing and configuring integration capabilities or point-to-point gateways
- Writing procedures to use database access logic efficiently
- Building data models for external data access tools
- Implementing security measures

IBM WebSphere Adapters

An IBM WebSphere Adapter implements the Java 2 Enterprise Edition (J2EE) Connector architecture (JCA), version 1.5. Also known as resource adapters or JCA

adapters, WebSphere Adapters enable managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components supported by WebSphere Process Server.

A WebSphere Adapter



The IBM^(R) WebSphere^(R) Adapter portfolio is a new generation of adapters based on the Java 2 Platform, Enterprise Edition (J2EE) standard. JCA is a standard architecture for integrating J2EE applications with enterprise information systems. Each of these systems provides native APIs for identifying a function to call, specifying its input data, and processing its output data. The goal of the JCA is to provide an independent API for coding these functions, to facilitate data sharing, and to integrate J2EE applications with existing and other EISs. The JCA standard accomplishes this by defining a series of contracts that govern interactions between an EIS and J2EE components within an application server.

Fully compliant with the JCA standard, WebSphere Adapters have been developed to run on WebSphere Process Server. A WebSphere Adapter does the following:

- Integrates with WebSphere Process Server.
- Connects an application running on WebSphere Process Server with an EIS.
- Enables data exchange between the application and the EIS.

Each WebSphere Adapter is made up of the following:

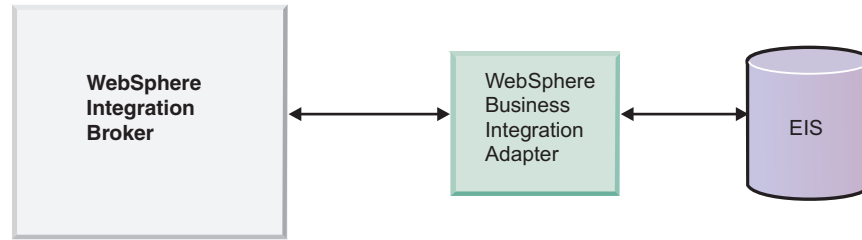
- An implementation of the (J2EE) Connector Architecture (JCA), version 1.5 that supports WebSphere Process Server
- An enterprise metadata discovery component— you use this component with the enterprise service discovery wizard to introspect the EIS— to generate business objects and other service component architecture (SCA) artifacts that are compiled in a standard enterprise application archive (EAR) file.

WebSphere Adapters use service data objects (SDO) for representing data objects.

WebSphere Adapters and WebSphere Business Integration Adapters

Unlike WebSphere Adapters, WebSphere Business Integration Adapters are not JCA-compliant.

A WebSphere Business Integration Adapter



As shown in the figure, WebSphere Business Integration Adapters are distributed. They reside outside of the application server. The server, or integration broker, communicates with this type of adapter through a Java Messaging Service (JMS) transport layer.

Other differences between WebSphere Adapters and WebSphere Business Integration Adapters include the following:

- **Connection management** WebSphere Adapters rely on standard JCA contracts to manage life-cycle tasks such as stopping, starting; WebSphere Business Integration Adapters rely on the WebSphere Adapter Framework to manage connectivity.
- **Event notification** Known as inbound event notification for WebSphere Adapters.
- **Request processing** Known as outbound support in WebSphere Adapters.
- **Object definition** With WebSphere Adapters, you use an enterprise metadata discovery component to probe an EIS and develop business objects and other useful artifacts. This enterprise metadata discovery component is part of the WebSphere Adapter. WebSphere Business Integration Adapters use a separate Object Discovery Agent (ODA) to probe an EIS and generate business object definition schemas.

Task road map: IBM WebSphere Adapter for PeopleSoft Enterprise

Task road map provides an overview of all tasks required for the installation and usage of the adapter.

Task	Description
Learning about Business Objects	Describes business objects for the WebSphere Adapter for PeopleSoft EIS.
Installing the adapter	Describes how to install the WebSphere Adapter for PeopleSoft EIS.
Deploying the adapter	Describes how to deploy the WebSphere Adapter for PeopleSoft EIS for inbound operations.
Configuring the adapter on the server	Describes how to configure the properties for the WebSphere Adapter for PeopleSoft EIS.
Troubleshooting the adapter	Describes how to troubleshoot the WebSphere Adapter for PeopleSoft EIS.
Using the sample applications	Describes how to use the sample applications for the WebSphere Adapter for PeopleSoft EIS.

Enterprise service discovery

The enterprise service discovery wizard allows you to generate business objects for enterprise information system (EIS) or database entities.

The enterprise service discovery wizard provides a blue print for business objects. It allows you to browse the metadata information of an EIS or database, enables selection of the artifacts of interest, and generates deployable service objects and descriptions. By selecting meta-object nodes from the metadata tree structure, you can generate business objects for EIS or database entities. The metadata is transformed into service data objects consisting business graphs and business objects.

The enterprise service discovery wizard allows you to perform the following actions:

- Generate business objects
- Set application-specific information on the business objects
- Set application-specific information on properties
- Provide service descriptions for inbound and outbound events
- Provide connection descriptions for inbound and outbound events

How the WebSphere Adapter for PeopleSoft Enterprise works

The Adapter for PeopleSoft Enterprise connects to two layers of PeopleTools API classes in order to access PeopleSoft applications running on application servers.

The adapter accesses the API classes in the following order:

1. The adapter accesses the primary 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 Java API and uses it to communicate with the EIS. Once the adapter connects to this layer of the PeopleSoft architecture, the following entities are exposed and available to the adapter:
 - All business objects in the component interface definition
 - PeopleCode methods associated with the underlying components
 - Records, except searches and menu-specific processing options

Outbound and inbound events

Describes the outbound processing logic used by the adapter to retrieve, update, or create a PeopleSoft component interface and populate it with business data, and states the custom project requirement for inbound processing.

To perform an outbound operation on an inbound business object the adapter follows these steps:

1. It extracts metadata from the InteractionSpec that identifies the appropriate component interface to access.
2. It extracts properties from the InteractionSpec that determine the outbound operation to perform.
3. Within the component interface, the adapter sets the keys from values specified in the business objects.
4. The adapter instantiates an existing component interface to retrieve, update, or create a component interface, and then it populates that component interface with data from the hierarchy of business objects.

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

The adapter can support inbound events as well. If your environment requires inbound support, you must first use PeopleTools Application Designer to create a custom event project.

Related tasks

“Creating a custom event project in PeopleTools” on page 15

Describes steps for creating a custom event notification project, which is needed in your environment uses inbound processing.

Asynchronous event delivery

Describes how the adapter uses staging and polling to support inbound processing.

The adapter supports inbound processing or event notification by delivering events along with data from the PeopleSoft Enterprise Information System (EIS) to end point applications. An event is represented by a row in the event store, which contains information that the adapter uses to detect the event and instantiate the PeopleSoft component interfaces. Events in the EIS must exist in one of five event state values, as described in the following table.

Event status values in PeopleSoft EIS

Event value in the EIS event table	Description
-1	There was an error processing the event.
0	Event is ready for polling.
1	Polling successful; information about the event has been retrieved from the EIS.
3	Polling has been initiated, and the retrieval of event information is in progress.
99	Indicates a future-dated event. If the resource adapter property PollFutureEvents is set to true , the adapter checks to see if the future date has arrived. If so, the event status is changed to 0 to indicate that the event is ready for polling.

Note: If you want events deleted from the event table after they have been successfully polled and their status set to 1, add the following PeopleCode to the SavePostChange method for the IBM_EVENT_BC.GBL component:

```

/* IBM_EVENT_STATUS check and deletion of record*/
/* Verify that the status is 1 before deleting*/
/* - i.e. the event poll was successful*/
If IBM_EVENT_TBL.IBM_EVENT_STATUS = "1" Then
    &CUREVENTREC = GetRecord(Record.IBM_EVENT_TBL);
    /* delete the record from the event table */
    &CUREVENTREC.delete();
End-If;

```

The adapter maintains a staging table to enable assured event delivery. All event records that are being processed are maintained in the staging table. As the adapter delivers events to an end point application, the record in the staging table is removed. If the event could not be delivered to the end point application, it remains in the staging table. The adapter performs recovery for such events in each poll cycle when the application is started. The following table shows the relationship between the status of an event in the event store, and the existence of event records in the staging table.

Note: The event distribution or staging table has to be in an XA compliant database. Properties that allow that configuration are defined on the activation specification.

Event states for recovery

EventStore state	Staging table	Meaning
Event marked as new/ready-to-be-pollled in the EIS.	No record of the event is in the staging table.	This is a new event. The adapter will process it normally.
Event marked as new/ready-to-be-pollled in the EIS.	A record exists in the staging table that refers to this event.	A record exists in the staging table that refers to this event.
Event marked as in-progress in the EIS.	A record exists in the staging table that refers to this event.	A record exists in the staging table that refers to this event.
Event marked as in-progress in the EIS.	No record of the event is in the staging table.	The event was successfully sent. The adapter will delete it the next time it recovers.

Event delivery is processed following these steps:

- An event is detected in the PeopleSoft EIS.
- A record corresponding to the event is created in the staging table.
- The event is marked as in progress in the EIS.
- For all operations except delete: The adapter retrieves the event from the EIS, publishes it to the endpoint, and simultaneously deletes the record from the staging table.
- For a delete operation: The adapter sets the key fields, publishes it to the end point, and simultaneously deletes the record from the staging table.
- When the event has been published as determined by data in the staging table, the adapter deletes the original event from the EIS.

These steps are repeated for each event.

Event delivery can be either unordered or ordered.

- For unordered delivery, the event manager spawns multiple delivery threads, one per event.
- For ordered delivery, the event manager creates one thread that delivers each event in sequence.

Locale and globalization support

This adapter has been globalized so that it can support single- and double-byte character sets and deliver message text in the specified language.

This adapter supports the processing of bidirectional script data for Arabic and Hebrew languages. To use the bidirectional capacity, you must configure the bidirectional properties. In this user guide, the term *bidirectional properties* refers to the properties that control invocation of bidirectional support.

If your enterprise information system (EIS) uses a bidirectional format that differs from the Windows® standard format, all properties with bidirectional support are transformed from the Windows standard format to the bidirectional format of the target EIS. The adapter also transforms such data from the EIS into Windows standard format before passing it to WebSphere Process Server.

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). Most components in the WebSphere Business Integration system are written in Java. Therefore, when data is transferred between most WebSphere Business Integration system components, there is no need for character conversion.

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

WebSphere Process Server bidirectional language format

WebSphere Process Server uses the bidirectional language format of ILYNN (implicit, left-to-right, on, off, nominal), which is also the Windows bidirectional language format. All other bidirectional language formats must be converted prior to being introduced to WebSphere Process Server.

Five attributes must be set for the proper bidirectional language format. The attributes and settings are listed in the table titled “Bidirectional attributes.”

Bidirectional attributes

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

The adapter is responsible for transforming data into a Logical-Left-to-Right format before sending the data into WebSphere Process Server components.

Note: The locale setting of the user interface (browser) defines the bidirectional language display and edit format. WebSphere Process Server user interfaces must convert locale-specific formats to the WebSphere Process Server default format.

Bidirectional property levels

You can set bidirectional properties at several different levels. For more details on these properties and how to set them using the enterprise service discovery wizard, refer to the sections on creating the adapter project and configuring the adapter.

Editing bidirectional properties

You can edit the bidirectional properties for business objects and business object attributes using annotations in the Business Object Editor in WebSphere Integration Developer. The annotations are stored in the business object (the *.xsd file). For more information, refer to the Business Object Editor documentation on the WebSphere Integration Developer website at <http://www.ibm.com/software/integration/wid>.

You can also edit certain bidirectional properties once they have been defined by using the assembly editor in WebSphere Integration Developer. For more information on using bidirectional properties at run time, refer to the general technical paper and the adapter technical paper regarding bidirectional support. For more information on the assembly editor, refer to the assembly editor documentation on the WebSphere Integration Developer website at <http://www.ibm.com/software/integration/wid>.

Learning about business objects

A business object represents a conceptual unit of business data, and the information in this section describes the metadata, naming conventions, structure, attribute properties, supported verbs, and application-specific information of business objects used by WebSphere Adapter for PeopleSoft Enterprise.

Business object structure

A business object is a set of attributes that represent a business entity, an action on the data (such as a create or update operation), and instructions for processing the data. Business objects must have certain structures.

The adapter supports and processes hierarchical business objects. The top-level business object that is contained within the business graph must have a one-to-one correspondence with the component interface created through PeopleTools. The created component is the top-level business object, and collections that occur within it are children of it.

Attribute properties

Describes the attributes (parts) of a business object used by WebSphere Adapter for PeopleSoft Enterprise.

The attribute properties table describes the attribute properties for the WebSphere Adapter for PeopleSoft Enterprise.

Attribute properties

Attribute property	Description
Name	Indicates the name of the Business Object attribute.
Type	Indicates the 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: CHAR maps to attribute type String NUMBER maps to attribute type BigDecimal LONG maps to attribute type Long SIGN maps to attribute type BigDecimal DATE maps to attribute type Date TIME maps to attribute type Time DTTM maps to attribute type DateTime
Key and foreign key	PeopleSoft architecture does not use the foreign key attribute. Child business objects have their own keys that have the primary key application-specific information. They also inherit keys from their parent business object.
Cardinality	Single cardinality for simple attributes; multiple cardinality for container attributes.
Special	None

Application-specific information

Describes properties that represent information that is specific to PeopleSoft Enterprise Application (EIS) systems, and used in PeopleSoft business objects.

Business object application-specific information.

The following table provides information about business object application-specific information.

Property name	Description
ObjectName	Name of the corresponding object in the PeopleSoft system; in other words, the component interface name as defined in the PeopleTools. Set only for the top-level business object.
InsAtOldestEffDtPos	When this is 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 application-specific property, do not set <code>InsAtCurrentEffDtPos</code> .

Property name	Description
InstAtCurrentEffDtPos	When this is 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 application-specific property, do not InsAtOldestEffDtPos.
EnableCrossReferencing	If the PeopleSoft system will automatically generate key values, set this property to true. The adapter will then copy the created values back from the PeopleSoft system to the business object.
GetCurrentItem	Set this property to true only for effective-dated components, that is, when only current effective-dated items need to be retrieved.

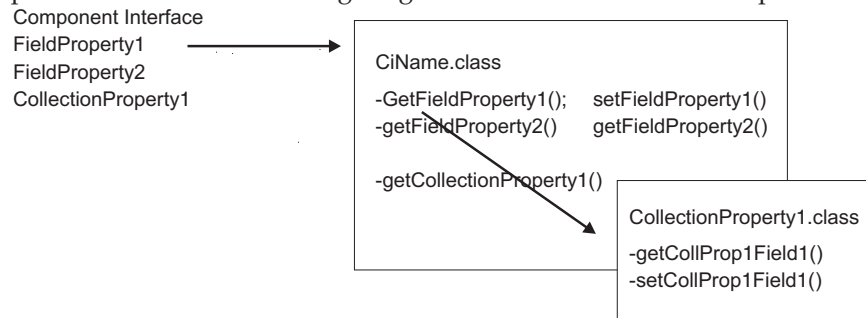
Attribute application-specific information

Describes the metadata available to attribute application-specific information.

PeopleSoft component interfaces include collections and fields. A collection is essentially a group of more field and collection properties. For each field property, there are two getter and setter methods generated as part of the API. The method names are:

- `get<Property Name>()`
- `set<Property Name>()`

where `<Property Name>` is the name of the property. A collection class is generated for each collection property and a `get<Collection Name>` is also generated in the parent's class. The following diagram describes the relationship.



So that a business object reflects the structure illustrated in the diagram, each attribute that corresponds to a Field Property on a component interface is considered a simple attribute. The following table shows the metadata available to each attribute.

Attribute application-specific information	Description
Getter	Getter method for that attribute.
Setter	Setter method for that attribute.
PrimaryKey	GetKey/CreateKey attribute in the component interface or a key in the child collection.

Attribute application-specific information	Description
FindKey	FindKey attribute in the component interface.
EffectiveDate	EffectiveDate attribute in the component interface.
EffectiveSequence	EffectiveSequence attribute in the component interface.
SoftDeleteValue	Set the value for this property with the value that will be used to update the corresponding value in the component interface when a top-level delete operation occurs. For example, if the property 'status' has its SoftDeleteValue set to inactive in the business object, set this value so that the same property in the component interface will be similarly updated by a top-level delete operation.

Supported verbs

WebSphere Adapter for PeopleSoft Enterprise supports verbs that allow data objects to be created, updated, and deleted.

Supported verbs

The following verbs are supported in the business graph.

- Create
- Update
- UpdateWithDelete
- Delete

Supported operations

The adapter performs inbound and outbound operations on business objects. The supported operations are listed here.

Inbound operations

The adapter provides only after-image support for inbound operations. When the adapter receives a business object for updating, it assumes that the business object represents the desired state of the data after the update. These are the supported operations:

- Create
- Update
- Delete

Outbound operations

The adapter supports the following outbound operations:

- Create
- Retrieve
- Update
- Delete
- Exists
- RetrieveAll
- Apply changes

Outbound operations

The outbound operations that the adapter supports includes create, retrieve, update, and delete. In addition, the adapter supports some variants of these four operations that are used in special circumstances.

Create operation:

Describes adapter processing for create operations

When a create operation is used, the adapter gets the 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 operation:

Describes adapter processing logic when a retrieve operation is used.

During a retrieve operation, the adapter gets the 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.

Update operation:

Describes adapter processing for update operations, represented as snapshot updates. A snapshot update involves comparison of child business objects in the enterprise information system, and then action against the child objects depending on the results of the comparison.

Snapshot updates are preformed by retrieving an object from the EIS and comparing it to the target business object. It processes the objects as follows:

- When the comparison reveals extra child objects in the EIS, the children are deleted.
- When the comparison reveals missing children in the EIS, the children are created.
- When the comparison reveals child objects that have been updated in the EIS, they are updated.

An update operation with the verb set to UpdateWithDelete can be used to provide better performance. The UpdateWithDelete verb requires a ChangeSummary that must include business object-level creations, updates, and deletions. This enables the adapter to perform the operation without the overhead of having to retrieve the existing entities from the EIS and do comparisons, because the ChangeSummary indicates what needs to be done. If the ChangeSummary is empty, the adapter does not take any action on the request.

Delete operation:

When application-specific information is set for a certain value, the adapter supports soft-delete operations, but only against a top level object within a PeopleSoft Component Interface.

The adapter supports soft deletes against the component interface. This means that it updates only those attributes that have the SoftDeleteValue application-specific information set. It updates the corresponding property in the enterprise information system (EIS) with the value that has been set for SoftDeleteValue application-specific information. For example, if SoftDeleteValue is set as inactive, the corresponding EIS property will be updated with a status of inactive. Soft deletes are supported only against a top-level object. If no attribute has SoftDeleteValue application-specific information set, the adapter generates an exception message on a Delete operation.

Exist operation:

The exist operation is used to determine whether or not an object exists. Perform an exist operation when it is necessary to check for an object but not process it.

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 the enterprise information system.

RetrieveAll operation:

The RetrieveAll operation returns a container object that contains any number of business graphs. The business graphs represent the objects.

During a RetrieveAll operation, the adapter uses application-specific values to determine the find keys in the component interface. Attribute values are set on the find keys in the component interface, and the adapter fetches all the objects that satisfy the search criteria. Once the objects are retrieved, the adapter creates a container business object instance. The adapter then adds the business graphs representing the records retrieved from the enterprise information system to the container business object.

The following limitations apply to a RetrieveAll operation:

- The number of records that the adapter can return is limited to the value set for the MaxRecords property in the InteractionSpec instance.
- The PeopleSoft enterprise information system limits the number of records that can be returned to 300, even if more records satisfy the condition.

When performing a RetrieveAll operation, the adapter uses wild-card searching, so the results are not exact. For example, if the FindKey attribute value is set as "G", the adapter retrieves all objects having the corresponding attribute starting with "G". If none of the FindKey attributes have any values set, the adapter retrieves all the records for the component subject to the maximum limitations.

ApplyChanges operation:

During this operation, the adapter processes any business object requiring a create, update, or delete operation.

This operation offers processing efficiency and it also simplifies data mapping between business objects. If the top-level verb exists in the business object, then the business object is processed as an after-image. If no top-level verb exists in the business object, then the ChangeSummary is processed.

When the ApplyChanges operation is used for create or delete transactions, the processing is the same as for those operations except that the ChangeSummary contains information about the object and the operation, and the verb is not set.

When the ApplyChanges operation is used for an update transaction, the adapter performs delta update processing if the verb is not set in the business object. The adapter inspects the ChangeSummary to identify the update operation for each business object in the input hierarchy, and then it performs those operations on the objects. For the top-level object, the operation must be update, but for child objects, the operation can be either create, update, or delete.

Therefore, the ApplyChanges operation offers processing efficiency because it can appropriately process an input business object to the adapter that only contains child objects within the hierarchy that require action. The update operation that is processed as a snapshot update, by comparison, requires that the entire business object hierarchy be input to the adapter.

Installing the adapter

Topics in this section provide information about the hardware and software requirements for installing the adapter, as well as a listing of files that are installed with the adapter.

For information on installing, see the *WebSphere Adapters Installation Guide*.

Adapter environment

The information in this topic provides hardware, software, and other requirements for WebSphere Adapter for PeopleSoft Enterprise.

Hardware and software requirements

For hardware and software requirements for this adapter, see IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: Hardware and Software Requirements. Select your adapter from the list of WebSphere adapters.

Adapter-specific information

If your environment requires inbound event support, you must either create a custom event project in PeopleTools, or use the sample event project IBM_EVENT_V600 that is provided in the samples folder with the adapter.

Related tasks

“Creating a custom event project in PeopleTools” on page 15

Describes steps for creating a custom event notification project, which is needed in your environment uses inbound processing.

Installed file structure

After you install the adapter, you can view the installed files and directories that have the installation directory as their root.

If the installation directory for the adapter is c:\WebSphereBI, then the CWYES_PeopleSoft.rar file has the following absolute path: c:\WebSphereBI.\adapter\PeopleSoft\deploy\CWYES_PeopleSoft.rar.

Note the following:

- The adapter resource archive (RAR) file contains both the adapter and the enterprise service discovery tool files.
- UNIX[®] and Windows operating systems share the same installed directory and file structure, with the only difference being the directory path designation (forward slash, /, UNIX, backslash, \, for Windows).

The following table lists the installed files for the WebSphere Adapter for PeopleSoft Enterprise. Directories and files are grouped into categories.

Installed directories and files

Directory and file category	Directories and files
RAR files	/adapter/PeopleSoft/deploy/CWYES_PeopleSoft.rar
Sample files	/adapter/PeopleSoft/samples/IBM_EVENT_V600/IBM_EVENT_V600.ini /adapter/PeopleSoft/samples/IBM_EVENT_V600/IBM_EVENT_V600.XML /adapter/PeopleSoft/samples/WBI_CUSTOMER/WBI_CUSTOMER.ini /adapter/PeopleSoft/samples/WBI_CUSTOMER/WBI_CUSTOMER.XML /adapter/PeopleSoft/samples/Apps/sampleinboundApp.ear /adapter/PeopleSoft/samples/Apps/sampleoutboundApp.ear
Notices file	/adapter/PeopleSoft/notices.txt
ISA plug-in zip file	/adapter/PeopleSoft/ISAPlugIn/ com.ibm.esupport.client.SS6FE6_RAPeopleSoft.zip
IBM Tivoli [®] License Manager (ITLM) file	/adapter/PeopleSoft/5724L81E060000.sys
Log Messages zip, and tar files	/adapter/PeopleSoft/messages/CWYES_PeopleSoft_messages.zip /adapter/PeopleSoft/messages/CWYES_PeopleSoft_messages.tar /adapter/PeopleSoft/messages/CWYBS_AdapterFoundation_messages.zip /adapter/PeopleSoft/messages/CWYBS_AdapterFoundation_messages.tar

Creating a custom event project in PeopleTools

Describes steps for creating a custom event notification project, which is needed in your environment uses inbound processing.

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 in the samples folder with the adapter. You can modify and use and the sample project, or you can create your own project using PeopleTools. If you create your own project, make sure that you complete the steps below.

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 will be retrieved from IBM_FETCH_ID record. This will be 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 will be used for the component's retrieval from the EIS.
IBM_EVENT_STATUS	Will be set to 0 or to 99. If the event is ready to be polled, status will be set to 0 and the IBMPublishEvent function will be called. If the event is to be polled at a future date, status will be set to 99 and the IBMPublishFutureDatedEvent function will be called.
IBM_OBJECT_VERB	Verb that will be set on the business object graph containing the retrieved business object.
IBM_EVENT_DTTM	Date on which the event is created. For a future dated event, this is the effective date.
IBM_NEXT_EVENT_ID	Field that will have the latest event ID under the record IBM_FETCH_ID. It will be incremented for each event added to the IBM_EVENT_TBL, and it will populate the IBM_EVENT_ID field within that table.

Custom project field names and descriptions

3. Create a record named IBM_EVENT_TBL and add to it all the fields you have just created.
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 default 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 classpath. For complete information about building a PeopleTools project and testing the PeopleSoft Component Interface, refer to PeopleSoft documentation.

Related concepts

“Outbound and inbound events” on page 4

Describes the outbound processing logic used by the adapter to retrieve, update, or create a PeopleSoft component interface and populate it with business data, and states the custom project requirement for inbound processing.

“Adapter environment” on page 14

The information in this topic provides hardware, software, and other requirements for WebSphere Adapter for PeopleSoft Enterprise.

Related reference

“PeopleCode for custom event project” on page 43

If you create a custom event project for inbound support, the following PeopleCode must be added to the project.

Deploying the adapter

Steps required to deploy the adapter on WebSphere Process Server.

You must install these products before you can install and deploy the adapter:

- WebSphere Integration Developer, Version 6.0 (WebSphere Integration Developer)
- WebSphere Adapter for PeopleSoft enterprise information system (EIS), installed on the same workstation as WebSphere Integration Developer
- IBM WebSphere Process Server for Multiplatforms, Version 6.0

For WebSphere Process Server information, see IBM WebSphere Process Server for Multiplatforms, Version 6.0.

In principle, deploying the adapter is the same as deploying any other component on WebSphere Process Server.

Related tasks

“Deploying and configuring for scenario 2” on page 41

Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts.

Creating a project

Creates a project in your workspace using WebSphere Integration Developer.

The first task of deploying the adapter is create a project for the adapter. Do this by importing the adapter .RAR file into WebSphere Integration Developer. Complete the following steps using WebSphere Integration Developer. For more information, see the documentation at IBM WebSphere Integration Developer.

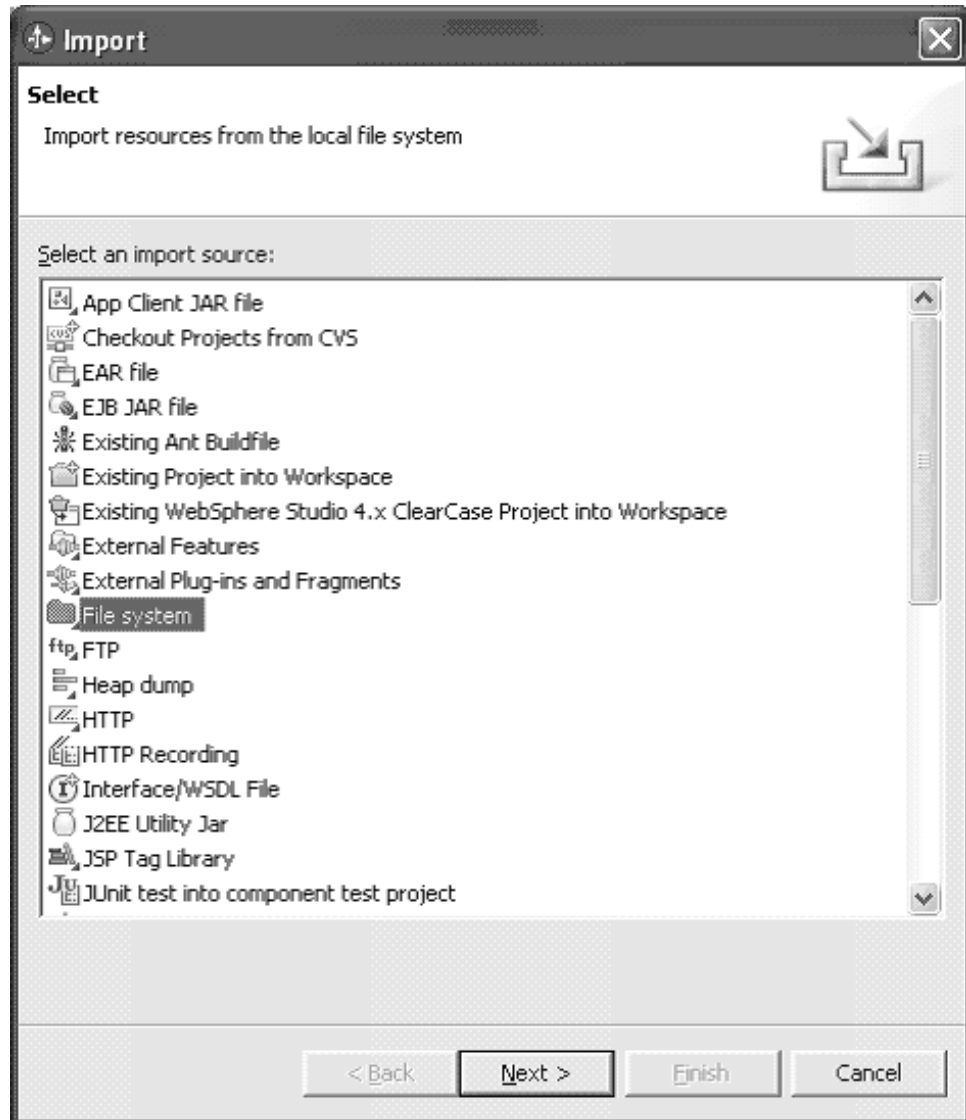
1. Launch the WebSphere Integration Developer.
2. Switch to the J2EE perspective.
3. Right-click **Connector Projects**, and select **File > Import** from the menu.
4. Select the location from where you will import the .RAR file (the same location where you copied your adapter file during installation), and specify a project name.
5. Clear the **Add module to an EAR project** check box.
6. Click **Finish** to import the RAR file. This creates a new J2EE Connector project in the workspace.

Adding external dependencies

External software dependencies are files that are specific to the enterprise information system, and required in order to deploy the adapter. Add them to the project by using WebSphere Integration Developer.

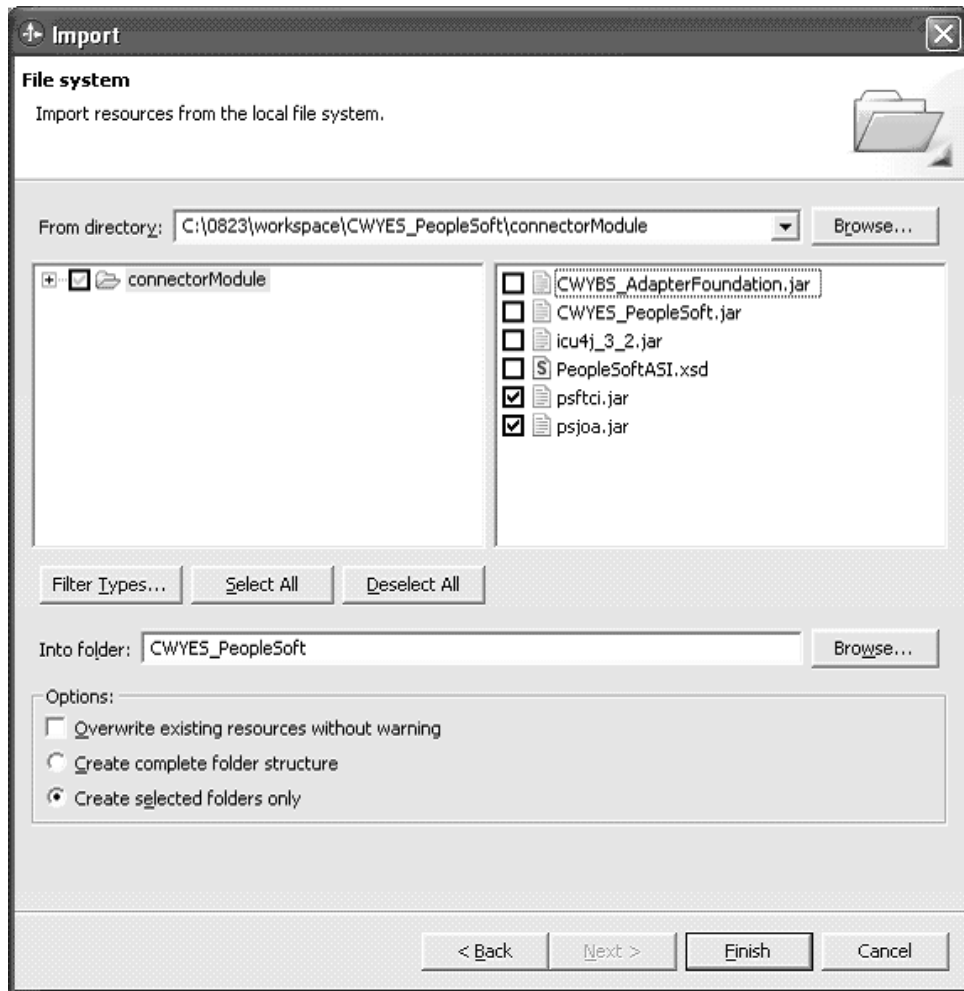
After you create the adapter application project, you must add the required external dependencies into the project. The psjoa.jar file is required, as well as a JAR file that contains the component interface API classes. To add the files, first copy them into a destination connectorModule folder, such as WID\workspace\CWYES_PeopleSoft\connectorModule. Follow these steps to copy them to the destination:

1. Click **File > Import**.

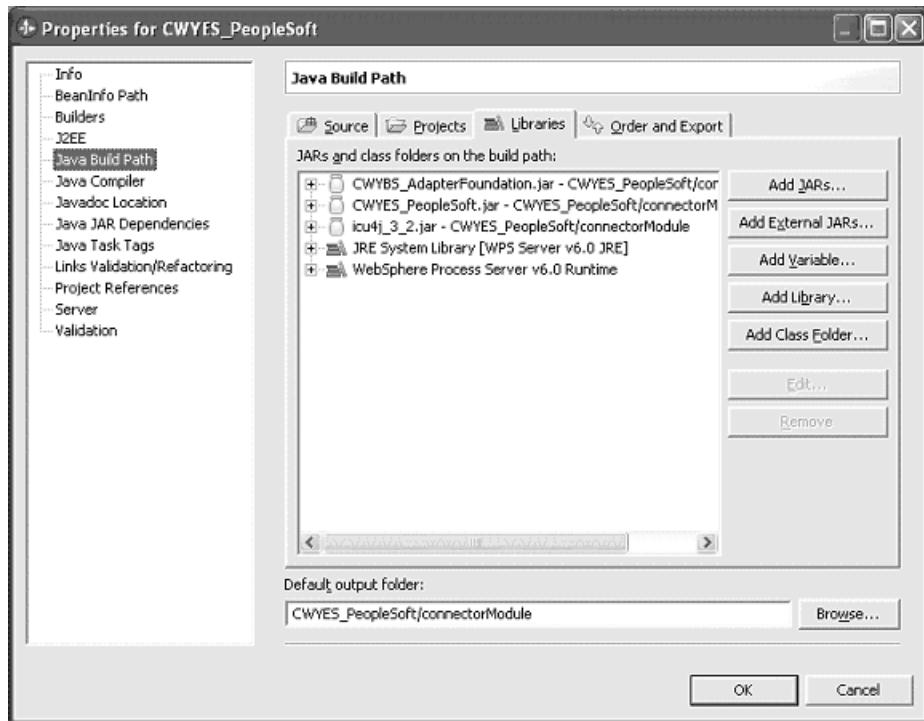


2. Select **File System**, and then click **Next**.
3. On the Import File System window, click the **Browse** button at the top of the window to browse to the location of the JAR files.
4. Select the JAR files on the right pane.
5. Click the **Browse** button in the middle of the window, and select the CWYES_PeopleSoft/connectorModule to copy the files into the

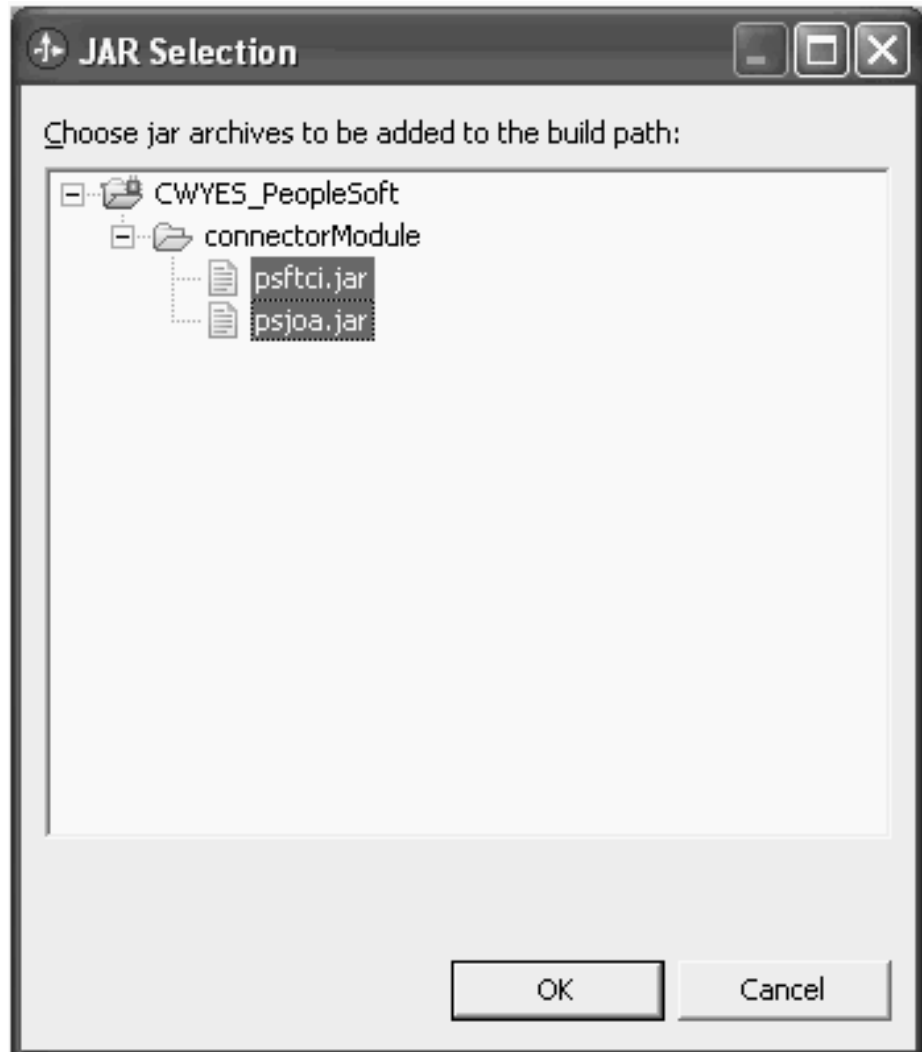
connectorModule folder of the connector project.



6. Click **Finish**.
7. Add the two files you have just copied to the connectorModule folder to the project as dependencies:
 - a. Under the **Connector Projects** folder, right click and select **Properties**.
 - b. From the Properties window, select **Java Build Path** in the left pane.
 - c. Click the **Libraries** tab.



- d. Select **Add JARS**.
- e. From the JAR Selection window, expand the nodes and select the psjoa.jar file and the component interface JAR file.



f. Click **OK** .

Creating a business integration module

Create a business integration module and configure service descriptions using the enterprise service discovery wizard.

You complete the configuration process using the enterprise service discovery wizard in WebSphere Integration Developer. As you perform the following steps, you enter all the information necessary to configure the adapter for the first time. The output from the enterprise service discovery wizard is saved to a business integration module, which contains the business objects, the import file (which describes outbound processing, as defined by the Managed Connection Factory and Interaction Specification), the export file (which describes inbound event processing, as defined by the Activation Specification), and the Web Services Description Language (WSDL) file.

Note: During deployment, if you specify J2C activation specification properties when you initially configure the service, those property settings will remain in place (in other words, you cannot update the properties later, after you install the application, through the WebSphere Process Server administrative console). If, for any reason, you want to set the J2C activation specification

properties after installing the application via the administrative console, then you should refrain from setting them during deployment. Note that J2C connection factory properties can be set during deployment and then updated via the administrative console after you install the application.

1. Switch to the Business Integration Perspective in WebSphere Integration Developer.
2. Right-click the frame of the Business Integration perspective window and select **New > Enterprise Service Discovery** from the pop-up menu. If Enterprise Service Discovery is not an option under New, expand **Others > Business Integration** to locate it.
3. When prompted to select a resource adapter, select **IBM WebSphere Adapter for PeopleSoft Enterprise** and then click **Next**.
4. Specify connection configuration properties on the Configure Settings for Discovery Agent window. For a full list and descriptions of these properties, see "Connection properties" under "References" in this documentation. Properties marked with an asterisk are mandatory. When finished, click **Next**.

Enterprise Service Discovery

Configure Settings for Discovery Agent

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

Connection Configuration

Miscellaneous

Prefix: PSOFT

Component Interface Jar:* C:\resourceadapter\PSFTCI.jar **Browse...**

Configure Additional BO Properties

User Credentials

Username: * MyUserid

Password: * *****

Machine Credentials

Hostname: * PeopleSoftServer

Port Number: * 9000

BiDi Properties

BiDi Transformation

BiDi OrderingSchema: Implicit

BiDi Direction: LTR

BiDi SymmetricSwapping

BiDi Shaping: Initial

BiDi NumericShaping: Nominal

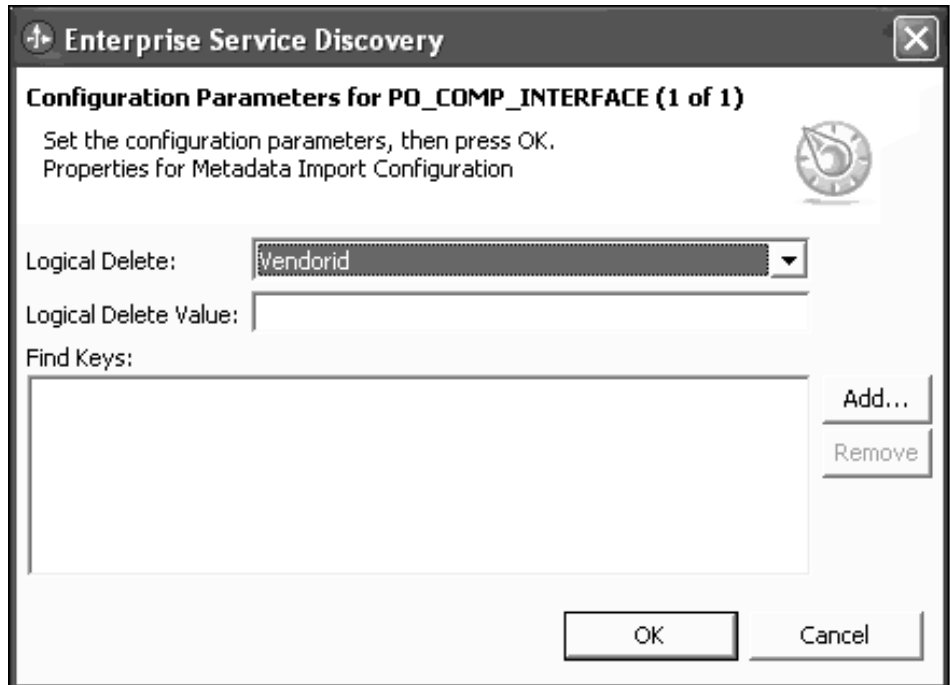
Show Advanced >>

5. In the Find and Discover Enterprise Services window, click **Run Query**.

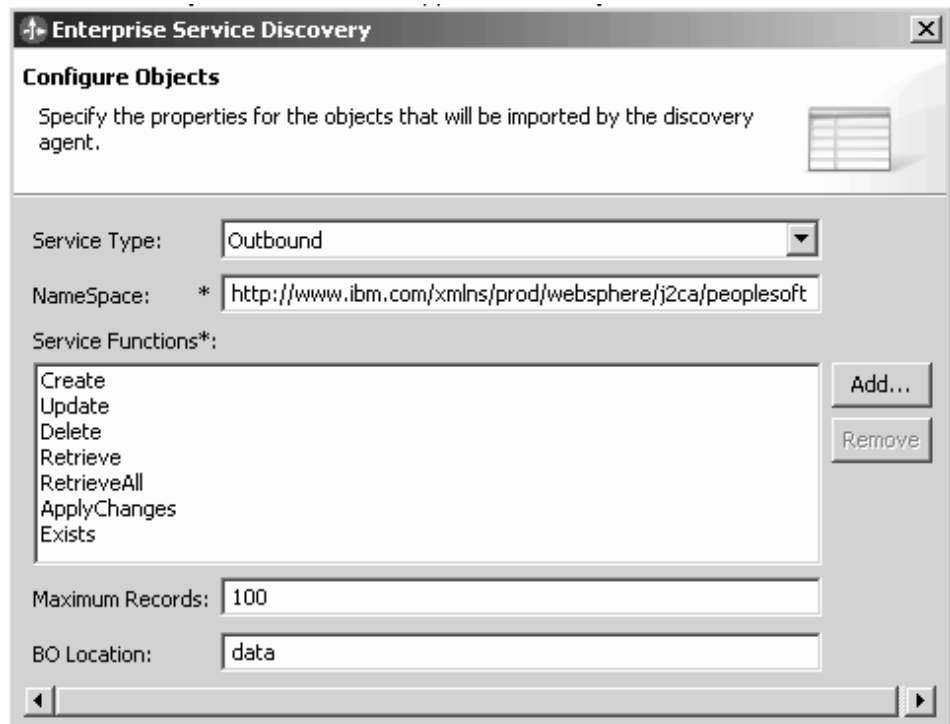


The business objects at the server are returned and placed in the Objects discovered by query pane. Importing and selecting these objects determines the artifacts that will be represented in the component. Select the objects you want and click **Add** to build the collection of objects.

6. If you selected the **Configure Additional BO Properties** check box on the Configure Settings for Discovery Agent window, the Configuration Parameters window appears, where you can do the following:



- a. In the **Logical Delete** field, select one attribute from the interface component to identify it as a SoftDelete key. All attributes of the interface component are available except child collection attributes.
 - b. In the **Logical Delete Value** field, select a value for the attribute you have identified for use as a SoftDelete key. The choice you make here is used to update records in the component when a delete operation is performed. For example, if you selected status in the logical delete field, and then you select closed in the logical delete value field, records in the component will have their status value updated to closed when the adapter performs a delete operation.
 - c. In the **Find Keys** field, select attributes from the interface component to use as FindKeys. All attributes of the interface component are available except child collection attributes. You can select multiple attributes.
7. In the Configure Objects window, specify properties for the objects that will be imported by the discovery agent. Mandatory properties are marked with an asterisk.



If you specify an Outbound **Service Type**, an outbound service description will be generated. If you specify an Inbound Service Type, an inbound service description will be generated. The **Service Functions** you select determine the operations on the EIS server your component will invoke.

8. In the **Generate Artifacts** window, specify values for properties for the project that will be added to the workspace. Mandatory Resource Adapter, Managed Connection factory, and Activation specification properties are marked with an asterisk. Make sure to select **Use discovered connection properties**, and the enter a value for the **J2C Authentication Data Entry** field. See [Creating an authentication alias](#). Click **New** to create a new business integration module.

Enterprise Service Discovery

Generate Artifacts

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

Properties for Interface

Module:

Namespace:

Use Default Namespace

Folder:

Name: *

Description:

Deploy connector with module

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

Use connection properties specified on server

Use discovered connection properties

J2C Authentication Data Entry:

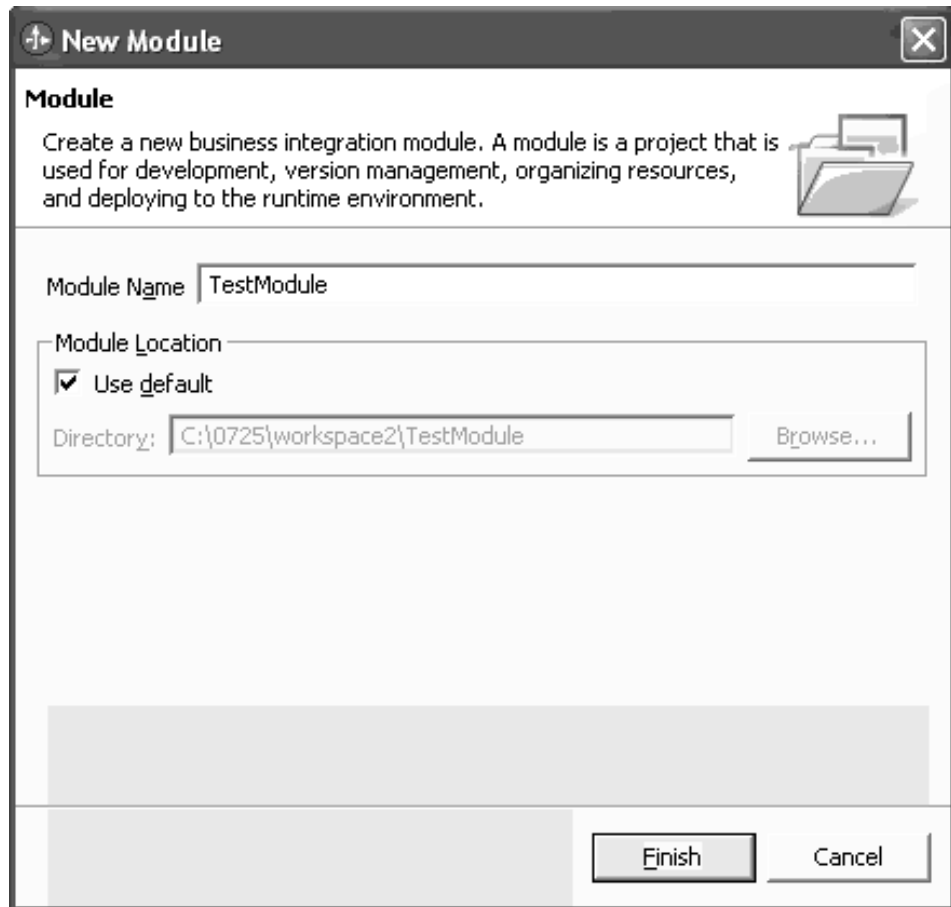
Managed Connection Factory Properties

User Credentials

Username:

Password:

9. In the window that opens, specify the module name where the SCA artifacts (business objects, their properties, import file, export file, and WSDL) will be saved.



10. Click **Finish**.

Related reference

“Connection properties” on page 47

Connection properties are used by the enterprise service discovery wizard for connecting to the PeopleSoft enterprise information system.

Generating reference bindings

Creates a reference to the adapter that is used by other WebSphere Business Integration components to access the adapter.

Reference bindings are used to link the adapter to other server processes. For complete information about reference bindings, see the documentation for IBM WebSphere Integration Developer, Version 6.0.

Creating an authentication alias

Describes steps to create an authentication alias.

Before you install the application, you must create an authentication alias for use with your PeopleSoft enterprise information system instance. After an authentication alias has been created, other PeopleSoft EIS project modules can use it as well.

1. In the WebSphere Administrative console, click **Security > Global security**.
2. On the right side, under Authentication, click **JAAS Configuration > J2C Authentication data**.

3. Click **New**.
4. In the **Alias** field, enter an alias name.
5. In the **User ID** and **Password** fields, enter a user name and password for connecting to the EIS.
6. Click **OK**.
7. Click **Save**.

Exporting the EAR file

Before you can run the project, you must export it to an enterprise archive (EAR) file through WebSphere Integration Developer.

1. In the Business Integration perspective window of the WebSphere Integration Developer tool, right click the module and select **Export** from the pop-up menu. The Export window appears.
2. Select EAR file from the Export - Select window. The Export - EAR Export window appears.
3. In the EAR Export window, select the EAR project you want to export and the destination directory, and then click **Finish**.

Deploying the adapter project

This is the last step in the deployment process. When you Install and deploy the application project module, the adapter, which is embedded within the project module, runs as part of the installed application.

Installing the application project module is the last step of the deployment process.

1. In the WebSphere Administrative console, click **Applications > Install New Applications**.
2. Under **Path to the new application**, specify the path of the enterprise archive (EAR) file, then click **Next**.
3. Continue to click **Next** through the steps.
4. Click **Finish** to complete the installation and deployment process.

Configuring the adapter on the server

By using the WebSphere administrative console, you can configure resource adapter properties, as well as the properties for J2C connection factory and J2C activation specification.

The adapter has three categories of properties that are configured using the WebSphere administrative console:

- Resource adapter properties
- J2C connection factory properties (which correspond to the ManagedConnectionFactory interface)
- J2C activation specification properties (which correspond to the ActivationSpec interface)

To configure properties using the WebSphere Process Server Administrative Console, follow these steps:

1. Start the WebSphere administrative console.
2. Under Resources, select **Resource Adapters**.
3. Under Resource Adapters, select **Adapter for PeopleSoft Enterprise**. The General Properties page appears.

4. Under Additional Properties, select one of the following:
 - Custom properties, to configure default resource adapter properties.
 - J2C connection factories, to configure the ManagedConnectionSpec properties, which are used to configure a target EIS (enterprise information system) instance.
 - J2C Activation specifications, to configure message end point properties.
5. Do one of the following:
 - If you selected Custom properties, the Custom properties page appears. Select the name of the default resource adapter property you wish to configure and set the value as desired.
 - If you selected J2C connection factories, select the name of the J2C connection factory that you want to configure, then, select Connection pool properties, Advanced connection factory properties, or Custom properties, depending on which J2C connection factory properties you want to configure. Connection pool and Advanced connection factory properties are properties you configure if you are developing your own adapter.
 - If you selected J2C Activation specifications, select the name of the J2C activation specification that you want to configure. Then select the name of the message end point property you want to configure and set the value as desired.

Troubleshooting

Multiple resources are available to help you troubleshoot and resolve problems. These include setting trace levels to capture information about adapter processing, reviewing log files to track processing events, reviewing tips to resolve some commonly encountered issues, and contacting IBM Software Support.

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus[®], and Rational[®] products, as well as DB2[®] and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage[®] in one of the following ways:
 - **Online:** Go to the Passport Advantage Web page and click How to Enroll.
 - **By phone:** For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region.
- For IBM eServer[™] software products (including, but not limited to, DB2 and WebSphere products that run in zSeries[®], pSeries[®], and iSeries[™] environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to

the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

- Determine the business impact of your problem.
 - Describe your problem and gather background information.
 - Submit your problem to IBM Software Support.
1. Determine the business impact of your problem. When you report a problem to IBM, you will be asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

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

2. Describe your problem and gather background information. When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:
 - What software versions were you running when the problem occurred?
 - Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
 - Can the problem be recreated? If so, what steps led to the failure?
 - Have any changes been made to the system? (For example, hardware, operating system, networking software, and so on.)
 - Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.
3. Submit your problem to IBM Software Support. You can submit your problem in one of two ways:
 - **Online:** Go to the Submit and track problems page on the IBM Software Support site. Enter your information into the appropriate problem submission tool.
 - **By phone:** For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support will create an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail.

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

Enabling logging

The WebSphere Adapter for PeopleSoft Enterprise maintains a log file that you can view to determine the status of event processing. All events and errors that relate to the adapter are tracked by the log file, along with the date, time, and event for each log entry. Since the adapter logs an error message when it encounters an error or warning condition, the log file is a good source to start troubleshooting problems.

Logging for the adapter is enabled through the WebSphere Process Server Administrative Console. Follow the steps below to enable logging.

1. Start the WebSphere Process Server administrative console.
2. From the administrative console, select **Troubleshooting** → **Logs and Trace**.
3. Click **Component** to specify a log detail level for individual components or click **Groups** to specify a log detail for a predefined group of components.
4. Select the logging level that you need. The "Logging levels" table describes the different logging levels that can be set through the WebSphere Process Server administrative console.

Note: To view log events that are below the Detail Level, you must enable the Diagnostic Trace Service. Log events that are at Detail Level or above can be viewed in the SystemOut log, the IBM Service log (when enabled), or the Diagnostic Trace Service (when enabled).

Logging levels

Level	Indicator	Description
Audit	A	Significant event affecting server state or resources
Config	C	Configuration change or status.
Detail	D	General information detailing subtask progress.
Fatal	F	Task cannot continue. Component cannot function.
Info	I	General information outlining overall task progress
Severe	E	Task cannot continue. Component can still function. This also includes conditions that indicate an impending fatal error - i.e. reporting on situations that strongly suggest that resources are on the verge of being depleted.
Warning	W	Potential error or impending error. This also includes conditions that indicate a progressive failure - for example, the potential leaking of resources.

5. Click **Apply** to save your changes.

Enabling tracing

Tracing determines what level of errors or warnings are captured in the adapter log file. You can trace messages regarding adapter processing by defining a tracing level.

The trace levels can be configured in the WebSphere Process Server administrative console. Follow the steps below to enable and set tracing levels.

1. Start the WebSphere Process Server administrative console.
2. From the administrative console, select **Troubleshooting** → **Logs and Trace**.
3. Select the Tracing level that you need. The "Tracing levels" table describes the different tracing levels that can be set through the WebSphere Process Server administrative console.

Tracing levels

Level	Indicator	Description
Fine	1	General trace. Includes broad actions being taken by adapter such as establishing a connection to the EIS, converting an event in the EIS to a business object (only key values), processing a business object (only key values).
Finer	2	Detailed trace that provides more granular information on the logic being performed by the adapter including the various API calls being made to the EIS and any parameters or return values.
Finest	3	This is the most detailed level and should include method entry / exit / return values. Complete business object dumps should be included. At this level, all detail needed to debug problems should be provided.

4. Click **Apply** to save your changes.

Enabling the Common Event Infrastructure (CEI)

This topic describes how to enable the Common Event Infrastructure (CEI) for the adapter.

You must publish the IBM WebSphere Adapters Event Definitions file to the CEI catalog before you can set these event definitions. For instruction on how to do this, refer to the CEI documentation found on the WebSphere Process Server web site at <http://www.ibm.com/software/integration/wps>.

1. Start the WebSphere administrative console.
2. Go to **Troubleshooting** → **Log and Trace** and select <your server name>.

3. There are many options for the General Properties. Select **Change Log Detail Level**, and then select **com.ibm.j2ca.*** for JCA components. Under this section there is a subcomponent for each adapter type:
 - com.ibm.j2ca.flatfile.* (WebSphere Adapter for Flat Files)
 - com.ibm.j2ca.jdbc.* (WebSphere Adapter for JDBC)
 - com.ibm.j2ca.peoplesoft.* (WebSphere Adapter for PeopleSoft)
 - com.ibm.j2ca.sap.* (WebSphere Adapter for SAP)
 - com.ibm.j2ca.siebel.* (WebSphere Adapter for Siebel)
4. Select the component that matches your adapter. Each adapter component has two subcomponents, one for logging and one for CEI. They are:
 - *subcomponent name.log.adapter id*
 - *subcomponent name.cei.adapter id*

For example, com.ibm.j2ca.siebel.cei.<AdapterID1>. For each instance of a deployed adapter, the system will show a separate ID.
5. Select the CEI adapter ID that you want to enable.
6. From the drop-down menu, you can choose from the following:
 - off - turn CEI off
 - fine- turn CEI on with Event Content set to Empty
 - finer- turn CEI on with Event Content set to Digest
 - finest- turn CEI on with Event Content set to Full
 - all - same as finest

For information on what each Event Content level means (Empty, Digest and Full), and for more information on using the Common Base Event model and the Common Event Infrastructure, refer to the documentation on the WebSphere Process Server web site at <http://www.ibm.com/software/integration/wps>

Troubleshooting tips

Describes some settings and procedures to check if you encounter problems running the adapter.

If you are unable to connect to the PeopleSoft EIS using the enterprise metadata discovery component or the adapter, make sure of the following:

- The psjoa and component interface jar files have been added to classpath.
- Valid values have been entered for Username, Password, serverName, and PortNumber.
- The PeopleSoft application server is running.

If you encounter problems when using the enterprise metadata discovery component, make sure of the following:

- You have selected the **Use Discovered Connection Properties** option within the enterprise service discovery wizard.
- You have specified a valid language value for the configured PeopleSoft instance.
- You have specified the authentication alias that will be used during deployment.

If you encounter problems when creating the EAR file, try the following:

- Make sure there are no validation errors in WebSphere Integration Developer

- Check the deployment descriptors for EJB projects on the module for entries of JNDI names for Managed Connection Factory or ActivationSpec. If there are errors, try rebuilding the workspace by selecting **Project > Clean option**, or try closing and opening WebSphere Integration Developer.

If you encounter problems at run time, make sure you have not created multiple service descriptions within the same Business Integration module that contain the same business object names and targetNameSpace definitions. If you need to use more than one service description, use a common set of business objects for both; do not save each description to a different location using the BO Location field in the enterprise metadata discovery component. If the services are saved to different business objects, there is no mechanism at run time to define which business object definitions are for which service descriptions.

Using the sample applications

The adapter provides a sample application that illustrates how you deploy an application package and how the adapter processes business objects. The application presents two scenarios, one for each audience of the adapter. The adapter audience consists of two users: the application integrator and the data integrator.

For the sample application, the two scenarios presented are as follows:

Sample application scenarios

Scenario	Description	Audience
Scenario 1	<ul style="list-style-type: none"> • Provides the already-generated artifacts and illustrates how the adapter processes business objects. Using the enterprise service discovery wizard to generate artifacts is not required in this scenario. • Targeted at an audience that is responsible for assembling application components into a solution and preparing this solution for testing and deployment. 	Application integrator

Scenario	Description	Audience
Scenario 2	<ul style="list-style-type: none"> • Illustrates how you use the enterprise service discovery wizard to discover PeopleSoft component interfaces and develop the business objects that the adapter processes. • Targeted at an audience with the same responsibilities as the application integrator, but is further responsible for enabling access to a range of data sources for the application developers. 	Data integrator

The sample illustrates the following scenarios using the provided customer project:

- Outbound processing allowing create, update, retrieve, retrieval, exists, and delete operations
- Inbound processing allowing create, update, and delete operations

Structure of the application package

The sample application files are installed when you install the adapter. They are packaged in an archive file that is installed in the Sample folder.

The sample application package includes all the required artifacts, so for Scenario 1, you do not need to use the enterprise service discovery wizard to obtain these. A minimal set of files is used for Scenario 2. When you run Scenario 2, the enterprise service discovery wizard generates these artifacts and configures the adapter.

The files are found in the following folder: \adapter\PeopleSoft\samples.

A PeopleTools project folder, WBI_Customer contains the following two files:

- WBI_Customer.xml
- WBI_Customer.ini

These files are used to create the required PeopleSoft EIS components. Copy these two files onto the PeopleSoft EIS file system. Then, do the following in PeopleTools:

- Import these two files through the Application designer.
- Build the project so that the tables are created in the database.
- Grant access permissions for the imported component interface.
- Grant access permissions for the PeopleTools Java APIs generated for the component interface.
- Repeat the steps above to import the event components IBM_EVENT_V600.xml and IBM_EVENT_V600.ini. These files are found in the IBM_EVENT_V600 folder.
- Compile the Java files generated, and create a jar file containing the classes generated. Name the file psftci.jar.

Note: For detailed information on completing the above steps, refer to PeopleSoft documentation.

Two .EAR files, sampleoutboundApp.EAR and sampleinboundApp.EAR are also provided within the adapter/PeopleSoft/samples/Apps folder. Copy them to a location of your choice.

The sampleoutboundApp.EAR contains the following files:

- Manifest.mf
- application.xml
- deployment.xml
- CWYES_PeopleSoft.rar
- sampleoutboundWeb.war
- sampleoutboundEJB.jar
- sampleoutbound.jar
- sampleoutboundEJBClient.jar

The sampleoutbound.jar file contains the business object (xsd), import, and wsdl files:

- Manifest.mf
- WbiCustomerCGB.xsd
- WbiCustomerCi.xsd
- Wbiaddress.xsd
- Wbiphone.xsd
- WbiCustomerCiContainer.xsd
- .runtime
- sampleoutbound.perfs
- PeopleSoftOutboundInterface.import
- PeopleSoftOutboundInterface.wsdl
- sca.module
- sca.modulex

The sampleinboundApp.EAR contains the following files:

- Manifest.mf
- application.xml
- deployment.xml
- CWYES_PeopleSoft.rar
- sampleinboundWeb.war
- sampleinboundEJB.jar
- sampleinbound.jar
- sampleinboundEJBClient.jar

The sampleinbound.jar file contains the business object (xsd), export, wsdl, and the listener class (Component1Impl) files:

- Manifest.mf
- WbiCustomerCGB.xsd
- WbiCustomerCi.xsd
- Wbiaddress.xsd

- Wbiphone.xsd
- WbiCustomerCiContainer.xsd
- .runtime
- Component1.component
- PeopleSoftInboundInterface.import
- PeopleSoftInboundInterface.wsdl
- Component1.Impl.class
- Component1.Impl.java
- sca.module
- sca.modulex

After you generate files for Scenario 2, compare your results against these lists to confirm successful generation.

Business object structure

Describes the structure of the business objects used in the sample applications.

Business objects are provided and used to demonstrate both outbound and inbound operations.

The following business objects are used for outbound operations:

- WBICustomerCibG: This is the business graph
- WBICustomerCi: A complex object under the business graph
- WBIAddress: A complex object under WBICustomerCi
- WBIPhone: A complex object under WBIAddress
- WBICustomerCiContainer: This is used for RetrieveAll operations

The following business objects are used for inbound operations:

- WBICustomerCibG: This is the business graph
- WBICustomerCi: A complex object under the business graph
- WBIAddress: A complex object under WBICustomerCi
- WBIPhone: A complex object under WBIAddress

Deploying and configuring for scenario 1

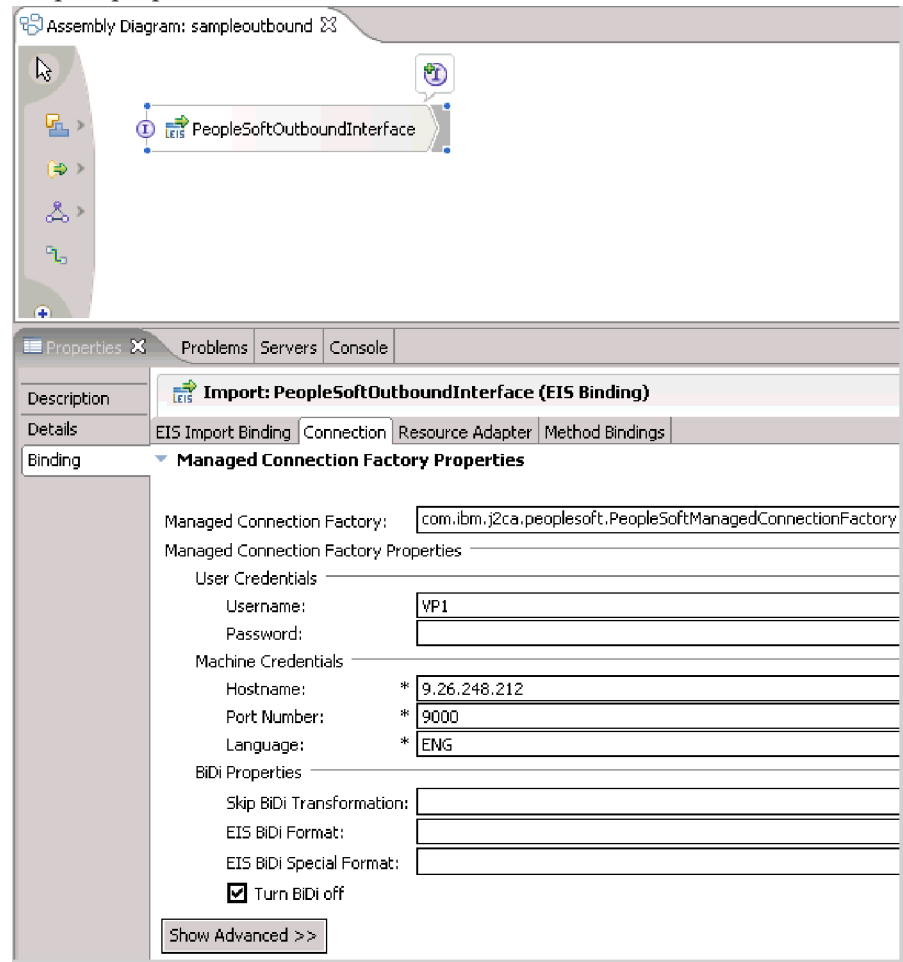
Scenario 1 sample application provides a configured instance of the adapter and all the necessary SCA artifacts, so you are not required to deploy the package and configure the adapter. Simply import the enterprise archive (EAR) file to your WebSphere Integration Developer workspace using WebSphere Integration Developer.

Follow these steps to deploy and configure the adapter for scenario 1:

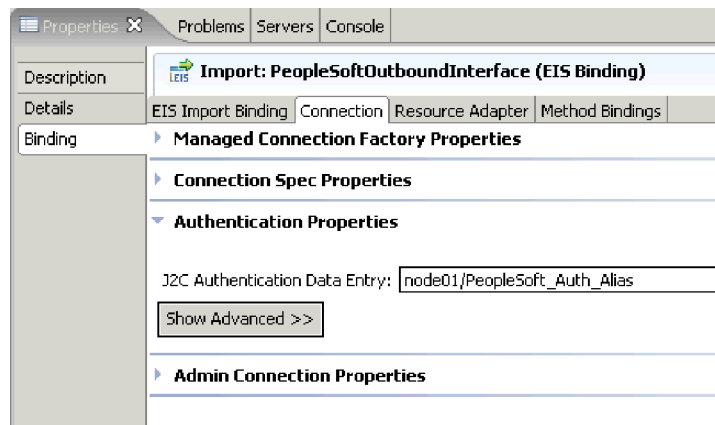
1. Uncompress the file SampleOutboundApp.ear to a directory of your choice. The CWYES_PeopleSoft.rar file and various JAR files are now listed in the directory.
2. In WebSphere Integration Developer, create a module and assign it the same name as the EAR file without the letters "App." For example, if the EAR file is SampleOutboundApp.ear, then the module name in this case should be SampleOutbound.

3. If the connector project is not already defined in your WebSphere Integration Developer workspace, import the latest CWYES_PeopleSoftAdapter.rar file in WebSphere Integration Developer. Importing the RAR file creates a connector project for the adapter.
4. Establish a dependency between the connector project and the module project by doing the following:
 - a. On the Project tab, right-click on the module.
 - b. Select **Properties > Java Build Path**.
 - c. Right-click on the module and select **Open Dependency Editor**.
 - d. Select CWYES_PeopleSoft in the J2EE selection box.
 - e. Select the check boxes for **On Build Path** and **Deploy with Module**.
5. Update the Java build path so that it includes the PeopleSoft dependency jars psjoa.jar and psftci.jar
6. Use any zip utility to extract the contents of the module JAR file into the module you created in Step 2. For example, extract sampleoutbound.jar into the module sampleoutbound.
7. Refresh the module.
8. Open the Business Integration perspective and edit the service description using the assembly editor. Do the following:
 - a. For outbound only: Select "sampleoutbound" within the project "sampleoutbound." For inbound only: Select "sampleinbound" within the project "sampleinbound."
 - b. For outbound: Set the Managed Connection factory and Resource adapter properties. For inbound: Set the Activation Specification and Resource

adapter properties.

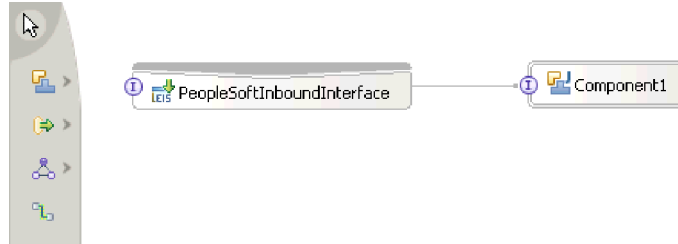


- c. Set the J2C authentication alias you have created. If you have not created one, do so through the WebSphere Process Server administrative console.



9. For inbound operations: You must create the listener MDB class and then bind it in the "sampleinbound" assembly diagram. Follow these steps to do so:
 - a. Open the sampleinbound assembly diagram under the sampleinbound project.
 - b. Select the component without implementation task.
 - c. Click the assembly diagram. "Component1" will be created.

- d. Link "PeopleSoftInboundInterface" to "Component1" you have just created.



- e. Right-click "Component1", and then select **Generate Implementation > Java**.
- f. Add Javacode as needed to the following methods that support outbound operations:

```
emitCreateAfterImageWbiCustomerCi
emitUpdateAfterImageWbiCustomerCi
emitDeleteAfterImageWbiCustomerCi
```

The following example shows a system.out added to methods in order to confirm entry to the listener.

```
public void emitCreateAfterImageWbiCustomerCi (
    DataObject emitCreateAfterImageWbiCustomerCiInput) {
    //TODO Needs to be implemented. Added system.out to confirm it comes in here
    System.out.println("emitCreateAfterImageWbiCustomerCi successful");
}
```

10. Go to the Projects tab and select the check box next to the Connector project you have created.
11. Start WebSphere Process Server.
12. After WebSphere Process Server has started, export an EAR out of the project and install it through the administrative console.
13. Check through the administrative console to ensure that the application has successfully started. Stop and restart the server if not.
14. For inbound operations only: Once the application starts, check the following to ensure success: for inbound operations, testing of the sample app ends at this point.
 - a. Navigate to WPS_Folder>\bi_v6\profiles\default\logs and examine the adapter trace file, PeopleSoftInboundTrace.txt to make sure it has the line 'emit<operation>AfterImageWbiCustomerCi successful, where <operation> can be Create, Update or Delete.
 - b. Examine the IBM_EVENT_TBL record to ensure that it contains events.
 - c. Examine the data in the PeopleSoft EIS by checking event status through the IBM_EVENT_CI. If the transaction was successful, the status will be set to 1.

These steps conclude testing the sample application for inbound operations.

15. For outbound operations only: use WebSphere Integration Developer test client to test the adapter by doing the following:
 - a. Ensure you are in the Business Integration Perspective. Right-click **sampleoutbound project > Test > Test Module**.
 - b. Select the operation as needed. Naming conventions correspond to the operation functions. For example, createWbiCustomerCi is for creating the component hierarchy in the PeopleSoft EIS.
 - c. Set values as needed, and set the verb for create, update, or delete operations.

- d. Click continue.
- e. Check the returned object to test the success of the operation. If retrieval is successful, you will be able to see the object in the test module. If retrieval is not successful, you will see the exception in the test module.



► General Properties

▼ Detailed Properties

Module: [sampleoutbound](#)

Component: [PeopleSoftOutboundInterface](#)

Interface: [PeopleSoftOutboundInterface](#)

Operation: [retrieveWbiCustomerCi](#)

Return parameters:

Name	Type	Value
[-] retrieveWbiCustomerCiOu...	WbiCustomerCIBG	
verb	VerbType	<null>
[-] WbiCustomerCi	WbiCustomerCi	
Customerid	BigInteger	7003
Customerfirstname	String	A
Customerlastname	String	S
Customerdob	String	
[-] Wbiaddress	Wbiaddress []	
[-] Wbiaddress[0]	Wbiaddress	
Addressid	BigInteger	4000
Addrline1	String	XTZ
Addrline2	String	
Customercity	String	
Customerstate	String	
Customerzip	String	
Customercoun...	String	
Wbiphone	Wbiphone []	<null>
InteractiveMode	Boolean	false
GetHistoryItems	Boolean	false
EditHistoryItems	Boolean	false
GetDummyRows	Boolean	false

These steps conclude testing the sample application for outbound operations.

Deploying and configuring for scenario 2

Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts.

Before you begin to deploy and configure, import the CWYES_PeopleSoft.RAR file into the project.

For this task, you must run the enterprise service discovery wizard within the IBM WebSphere Integration Developer to set adapter configuration properties. To change the property values later, use the WebSphere Process Server administrative console.

Note: Note: During deployment, if you specify J2C activation specification properties when you initially configure the service, those property settings will remain in place (in other words, you cannot update the properties later, after you install the application, via the WebSphere Process Server administrative console). If, for any reason, you want to set the J2C activation specification properties after installing the application via the administrative

console, then you should refrain from setting them during deployment. Note that J2C connection factory properties can be set during deployment and then updated via the administrative console after you install the application.

1. In the Business Integration Perspective of WebSphere Integration Developer, right-click the frame and from the pop-up menu, select **New** → **Enterprise Service Discovery**.
2. Click **Next** in the following screen.
3. In the Import Configurations window, select **IBM WebSphere Adapter for PeopleSoft Enterprise**, and then click **Next**.
4. In the Discovery Agent Initialize Properties window, select the component interface psftci.far. Set properties marked with an asterisk (*) and click the **Next** button.
5. In the Find and Discover Enterprise Services window, click the **Run Query** button.
6. Under Objects discovered by query, select WBI_CUSTOMER_CI and click the **Add Selected** button. Then click **Next**.
7. For outbound processing, do not change the **Service Type**; accept the default value Outbound. For inbound processing, change the **Service Type** to **Inbound**. Enter data in the **BO Location** field, and then click **Next**. (Remove **Apply Changes** if you plan to test through the test client.)
8. Click the **New** button to create a new project, and then specify TestOutbound in the **Module Name** field, and then click **Next**. (For inbound processing, specify TestInbound in the **Module Name** field.)
9. Click **Finish**.
10. In the Saving Properties window, enter Sample in the **Folder** field, and then click **Specify Connection Properties**. Set properties marked with an asterisk (*) and specify IBM_EVENT_CI for the **Component Interface** field. Make sure to specify a language, and to select **Use discovered connection properties**. Click the **Finish** button. These steps should produce a new project, either TestOutbound or TestInbound, that appears in the Business Integration window.
11. Export the project to the .EAR file by selecting the project and then clicking **File, Export**.
12. Deploy the .EAR file.

The adapter application package, including all its SCA artifacts, is now configured and deployed. You can now run the sample application.

Related tasks

“Deploying the adapter” on page 17

Steps required to deploy the adapter on WebSphere Process Server.

Related reference

“Connection properties” on page 47

Connection properties are used by the enterprise service discovery wizard for connecting to the PeopleSoft enterprise information system.

References

This section provides useful reference information.

PeopleCode for custom event project

If you create a custom event project for inbound support, the following PeopleCode must be added 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 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 = &BO;
&IBMREC.Insert();
End-Function;
Function IBMPublishEvent(&BO, &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;

```

Related tasks

“Creating a custom event project in PeopleTools” on page 15
 Describes steps for creating a custom event notification project, which is needed in your environment uses inbound processing.

Connection properties

Connection properties are used by the enterprise service discovery wizard for connecting to the PeopleSoft enterprise information system.

The table titled Metadata discovery connection properties describes the properties used by the enterprise service discovery wizard to connect to the PeopleSoft EIS.

Metadata discovery connection properties

Property	Description
Prefix	The prefix to be added to the name of the business object.
Component Interface Jar	The path and name of the component Interface jar files.
Username	User name to login to the EIS.
Password	Password for the corresponding user name.
Hostname	Name or IP Address of the machine hosting the EIS.
Port	Jolt port number.

Bidirectional connection properties

When you initially configure the adapter, if you want to enable bidirectional language support for the enterprise service discovery wizard, you must configure a set of properties. These properties determine the bi-di format of field names that display throughout the enterprise service discovery wizard and the bi-di format of the values you enter in these fields. The properties, which are used by the enterprise service discovery wizard during adapter configuration to communicate

with the PeopleSoft EIS, appear in the first window of the wizard (the Configure Settings for Discovery Agent window) under the heading BiDi properties. Note that the only fields in this first window that are affected by the bi-di format defined are User Name and Password. Other fields in this first window do not have bi-di support. The following table describes the BiDi properties that appear in the first window of the enterprise service discovery wizard. The following table lists the bidirectional properties that are used during connection configuration.

Bidirectional connection properties

Property	Type	Description
BiDiTransformation	Boolean	Turns bidirectional support on or off. The default value is false, which means that bi-di support is turned off.
BiDOrderingSchema	String	Determines the type of text schema used; either implicit (logical), or visual. The default value is implicit.
BiDiDirection	String	Determines the text direction used. Possible values are LTR (left to right), RTL (right to left), ContextualLTR (contextual left to right), and ContextualRTL (contextual right to left.) The default value is LTR.
BiDiSymmetricSwapping	Boolean	Determines whether systemic swapping is turned on or off. The default value is True, which means that systemic swapping is turned on.
BiDi Shaping	String	Determines the bi-di format used by the enterprise service discovery wizard while it communicates with the PeopleSoft EIS. Possible values are Nominal, National, Contextual The default value is Nominal.
BiDiNumericShaping	String	Determines the bi-di format used by the enterprise service discovery wizard while it communicates with the PeopleSoft EIS. Possible values are Nominal, National, Contextual The default value is Nominal.

Related tasks

“Creating a business integration module” on page 21

Create a business integration module and configure service descriptions using the enterprise service discovery wizard.

“Deploying and configuring for scenario 2” on page 41

Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts.

Resource adapter properties

Configure the resource adapter properties by using the WebSphere Process Server administrative console.

Resource adapter properties

Property	Description
BONamespace	This is a required string value. It represents the namespace for the business object definitions used by the adapter. A default value is provided but you may change it during enterprise service discovery. This property supports globalized values.
EventKeyDelimiter	The delimiter for the Object key name-value pair in the event table. This property does not support globalized values.
PollFutureEvents	Enables polling of events that are future effective-dated only on or after the set date. This property does not support globalized values.
PingCompInterface	Used to check if the session is corrupt or not. An existing component interface name must be specified in order for the adapter to check for its existence. If the component interface does not exist or if the operation times out, the adapter returns an error. This property does not support globalized values.
LogFileName	This is a required string value. It represents the full path of the log file.
LogNumberOfFiles	An integer value, this property represents the number of log files to use. When a log file reaches its maximum size it will start using another log file. If no values is specified it will be set to 1.
LogMaxFileSize	Optional, integer value. Size of the log files in kilobytes. If no value is specified the file will have no maximum size.
TraceFileName	Required, string value. This property represents the full path to the trace file.
TraceNumberOfFiles	Optional, integer value. The number of trace files to use. When a trace file reaches its maximum size it will start using another trace file. If no values is specified it will be set to 1.
TraceFileSizeMax	Optional, integer value. Size of the trace files in kilobytes. If no values is specified the file will have no maximum size.
Timeout	Any integer greater than 1. Amount of time in milliseconds to wait for a response from the EIS.
EISEncoding	A string value used if adapter needs to convert unicode to/from native encoding like in case of file handling etc.. This should specify the native encoding to use.

Property	Description
BiDiEIS	The bidirectional (bi-di) form used by the EIS for its business data. The adapter normalizes the EIS bi-di data to the application server's logical left-to-right bi-di format for inbound communication and from the application server's format back to EIS bi-di format for outbound communication.
BiDiMetadata	Defines the bi-di format for all application-specific information contained within a business object exchanged between the adapter and the EIS.
BiDiSkip	Determines whether or not bi-di support is invoked by the adapter for transactions with the EIS.
BiDiSpecialFormat	Signifies a category of values that are subject to special treatment during invocation of bi-di transformation to ensure accurate transformation of the category. Categories are predefined. For example: FTP URLs and email addresses.
BiDiEISOrderingSchema	A string value that determines the type of text schema used; either implicit (logical), or visual. The default value is implicit.
BiDiTurnBiDiOff	Turns off bi-di support support. This property takes precedence over the BiDiSkip property, and it allows users who do not require bidirectional script data support to turn it off.
BiDiEISDirection	A string value that determines the text direction used. Possible values are LTR (left to right), RTL (right to left), ContextualLTR (contextual left to right), and ContextualRTL (contextual right to left.) The default value is LTR.
BiDiEISSymmetricSwapping	A boolean value, this property determines whether systemic swapping is turned on or off. The default value is True, which means that systemic swapping is turned on.
BiDiTransformation	A boolean value that turns bi-di support on or off. The default value is false, which means that bi-di support is turned off.

J2C connection factory properties

Configure the J2C connection factory (ManagedConnectionFactory) properties by using the WebSphere Process Server administrative console.

J2C connection factory properties

Property	Description
User name	The user name for logging into the PeopleSoft enterprise information system (EIS). This property does not support globalized values. This property is enabled for bidirectional script languages.
Password	The password for the user. This property does not support globalized values. This property is enabled for bidirectional script languages.
Host name	Name or IP Address of the machine hosting the PeopleSoft EIS. This property does not support globalized values.
Port	Jolt port number. This property does not support globalized values.
Language	Language to be used. This property does not support globalized values.

Activation Specification properties

Configure the inbound J2C activation specification properties by using the WebSphere Process Server administrative console.

J2C activation specification properties

Property	Description
User name	The user name for logging into the PeopleSoft EIS. This property does not support globalized values.
Password	The password for the user. This property does not support globalized values.
Host name	Name or IP Address of the machine hosting the PeopleSoft EIS. This property does not support globalized values.
Port	Jolt port number. This property does not support globalized values.
Language	Language to be used. This property does not support globalized values.
EventCIName	The name of the component interface that the adapter will interact with during inbound operations. This is a required value. Default Value: IBM_EVENT_CI.

Property	Description
PollPeriod	This is considered the rate (in milliseconds) at which the EIS event store is polled for new inbound events. This property is required, and valid values are any integer equal to or greater than 0. If set to "0", the adapter will not wait between cycles. The poll cycle is established at a fixed-rate, meaning that if an execution of the poll cycle is delayed for any reason, such as a prior poll cycle taking longer than expected to complete, the next cycle will occur immediately to catch up. Default: 500
PollQuantity	This property is required and must be set to any positive integer greater than 0. It determines the number of events to deliver to each endpoint per poll cycle. Default: 1
EDTDriverName	This string value is optional. If used, it represents the XA database driver to use to connect to the Event Distribution Table for inbound events. Example: com.ibm.db2j.jdbc.DB2jXADataSource If there is no value present, the Event Manager will be unable to perform recovery so delivery is not assured.
EDTDatabaseName	String value, not required. The name of the database for the event distribution table.
EDTTableName	String value, not required. The name of the table that represents the event distribution table.
EDTUserName	String value, not required. The username required to connect to the database for the event distribution table.
EDTPassword	String value, not required. The password required to connect to the database for the event distribution table.
EDTSchemaName	String value, not required. The schema name for the event distribution table.
EDTPortNumber	Integer value, not required. The port number for the event distribution table's database.
EDTServerName	String value, not required. The server name for the event distribution table's database.
EDTURL	String value, not required. The URL for the event distribution table's database.
DeliveryType	String value, not required. If used, set to ORDERED or UNORDERED. This determines the order in which the events will be published. One at a time (ORDERED), or all at once (UNORDERED). The default value is ORDERED.
AutoCreateEDT	Boolean value, not required. If used, this flag indicates whether the adapter should create the EDT table automatically if it doesn't already exist. Default: true

EDT configuration information

Provides sample event data table (EDT) configuration information for various databases.

For configuration of the EDT, a particular XA provider may use any combination of the EDT properties in addition to the EDTDriverName. If a property must be set for an XA DataSource's configuration, and it is not present in the activation specification, a configuration file may be used. This file is called *EDTExtendedProperties.properties*. Place it where the adapter is installed. The property file uses a standard Java properties file format.

The sample configuration values that follow omit these properties: EDTUserName, EDTUserPassword, and EDTTableName because they are always required. Also, EDTSchema is omitted as it is only required for a non-default schema. Values shown in brackets are not literal, and you should replace them with appropriate values for your environment during configuration.

Sample values for EDT properties on DB2 databases

Property	Sample value
EDTDriverName	COM.ibm.db2.jdbc.DB2XADataSource
EDTDatabaseName	[an existing database]

Sample values for EDT properties on Cloudscape™ databases

Property	Sample value
EDTDriverName	com.ibm.db2j.jdbc.DB2jXADataSource
EDTDatabaseName	[any name; database need not exist]

Sample values for EDT properties on Informix® databases

Property	Sample value
EDTDriverName	com.informix.jdbc.IfxXADataSource
EDTServerName	[The name of the Informix instance on the server]
EDTPortNumber	[port number]
EDTDatabaseName	[The database name]
EDTExtendedProperties.properties	Include this file to complete the onfiguration
ifxIFXHOST	[The physical name of the database server]

Sample values for EDT properties on Oracle databases

Property	Sample value
EDTDriverName	oracle.jdbc.xa.client.OracleXADatasource
EDTURL	jdbc:oracle:thin:@myServer:1521:myDatabase

Sample values for EDT properties on MS-SQL Server

Property	Sample value
EDTDriverName	com.microsoft.jdbcx.sqlserver.SQLServerDataSource
EDTServerName	[name of server hosting SQLServer]
EDTDatabaseName	[name of the existing database]
EDTPortNumber	[port]

Sample values for EDT properties on Sybase databases

Property	Sample value
EDTDriverName	com.sybase.jdbc2.jdbc.SybXADataSource
EDTServerName	[name of server hosting Sybase]
EDTDatabaseName	[name of the existing database]
EDTPortNumber	[port]

Notices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

i5/OS
IBM
the IBM logo
AIX
AIX 5L
CICS
CrossWorlds
DB2
DB2 Universal Database
Domino
HelpNow
IMS
Informix
iSeries
Lotus
Lotus Notes
MQIntegrator
MQSeries
MVS
Notes
OS/400
Passport Advantage
pSeries
Redbooks
SupportPac
WebSphere
z/OS

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

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

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



WebSphere Adapters, Version 6.0



Printed in USA