BEA WebLogic Adapter for PeopleSoft® 8

User Guide

Version 8.1.0
Document Revised: July 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

1. Introducing the BEA WebLogic Adapter for PeopleSoft 8

2. Using PeopleSoft 8 Component Interfaces

3. Using PeopleSoft 8 Application Messaging

4. Generating Schemas for PeopleSoft 8 Component Interfaces

5. Defining Application Views for PeopleSoft 8

Index

About This Document

This document describes how to install, configure, and use the BEA WebLogic Adapter for PeopleSoft 8. This document is organized as follows:

- Chapter 1, "Introducing the BEA WebLogic Adapter for PeopleSoft 8,"describes the adapter, how it relates to both PeopleSoft 8 business objects and WebLogic Integration.
- Chapter 2, "Using PeopleSoft 8 Component Interfaces," describes how to create component interfaces and generate component interface APIs used when you generate request and response schemas for services.
- Chapter 3, "Using PeopleSoft 8 Application Messaging," describes how to create messages and generate message API used when you create event schemas.
- Chapter 4, "Generating Schemas for PeopleSoft 8 Component Interfaces," describes how to generate schemas for your PeopleSoft 8 component interfaces and messages using the BEA Application Explorer.
- Chapter 5, "Defining Application Views for PeopleSoft 8," describes application views and how to use them to configure events and services.

Who Should Read This Documentation

This document is intended for the following members of an integration team:

- Integration Specialists—Lead the integration design effort. Integration specialists have expertise in defining the business and technical requirements of integration projects, and in designing integration solutions that implement specific features of WebLogic Integration.

The skills of integration specialists include business and technical analysis, architecture design, project management, and WebLogic Integration product knowledge.

- Technical Analysts—Provide expertise in an organization's information technology infrastructure, including telecommunications, operating systems, applications, data repositories, future technologies, and IT organizations. The skills of technical analysts include technical analysis, application design, and information systems knowledge.
- Enterprise Information System (EIS) Specialists—Provide domain expertise in the systems that are being integrated using WebLogic adapters. The skills of EIS specialists include technical analysis and application integration design.
- System Administrators—Provide in-depth technical and operational knowledge about databases and applications deployed in an organization. The skills of system administrators include capacity and load analysis, performance analysis and tuning, deployment topologies, and support planning.

We assume that users have the following skills:

- PeopleSoft 8 general product knowledge, including Application Designer and PeopleTools
- PeopleSoft Application Messaging
- PeopleSoft Component Interface
- Business application knowledge in a specific application area
- Knowledge of the PeopleSoft 8 processes and data model in the required application area
- General knowledge of client-server concepts

Additional Information

To learn more about the software components associated with the adapter, see the following BEA documents:

- *BEA WebLogic Adapter for PeopleSoft 8 Release Notes*

  http://edocs.bea.com/wladapters/docs81/pdf/relnotes.pdf

- *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide*

  http://edocs.bea.com/wladapters/docs81/pdf/install.pdf

- *BEA Application Explorer Installation and Configuration Guide*

  http://edocs.bea.com/wladapters/bae/docs81/pdf/install.pdf

- *Introduction to the BEA WebLogic Adapters*

  http://edocs.bea.com/wladapters/docs81/pdf/intro.pdf

- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation

  http://dev2dev.bea.com/products/product.jsp?highlight=wla

- Application Integration documentation

  http://edocs.bea.com/wli/docs81/aiover/index.html

  http://edocs.bea.com/wli/docs81/aiuser/index.html

- BEA WebLogic Integration documentation

  http://edocs.bea.com/wli/docs81/index.html

- BEA WebLogic Platform documentation

  http://edocs.bea.com/platform/docs81/index.html

You may also find it helpful to refer to the following PeopleSoft 8 documentation topics:

- PeopleSoft Component Interface
- PeopleSoft Internet Architecture Administration
- PeopleSoft Application Messaging

How to Use This Document

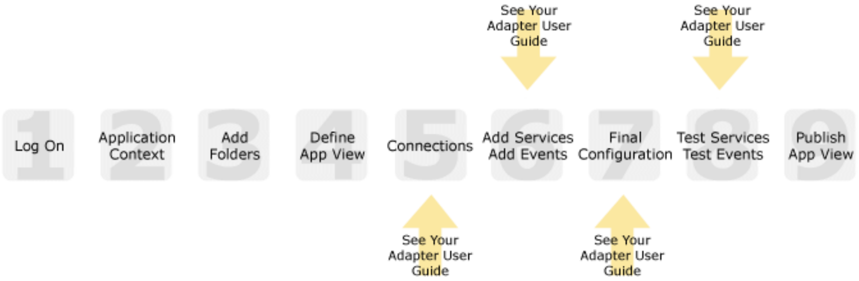
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your Peoplesoft system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 1 Information Interlock with *Using the Application Integration Design Console*


See Your Adapter User Guide
See Your Adapter User Guide
1 Log On
2 Application Context
3 Add Folders
4 Define App View
5 Connections
6 Add Services Add Events
7 Final Configuration
8 Test Services Test Events
9 Publish App View
See Your Adapter User Guide
See Your Adapter User Guide

Contact Us!

Your feedback on the BEA WebLogic Adapter for PeopleSoft 8 documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for PeopleSoft 8 documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for PeopleSoft 8 and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for PeopleSoft 8, or if you have problems using the BEA WebLogic Adapter for PeopleSoft 8, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: buildobjclient [-v] [-o name] [-f *file-list*]... [-l *file-list*]...
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: • That an argument can be repeated several times in a command line • That the statement omits additional optional arguments • That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: buildobjclient [-v] [-o name] [-f *file-list*]... [-l *file-list*]...
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Introducing the BEA WebLogic Adapter for PeopleSoft 8

This section introduces the BEA WebLogic Adapter for PeopleSoft 8 and describes how the adapter enables integration with PeopleSoft 8 business objects and WebLogic Integration.

It includes the following topics:

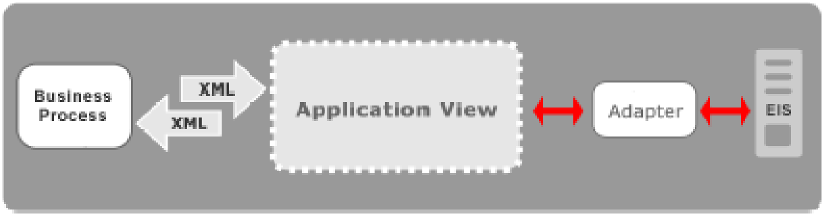
- About Adapters and BEA WebLogic Integration
- Key Components of Integration Solutions
- About the BEA WebLogic Adapter for PeopleSoft 8
- Getting Started With the Adapter for PeopleSoft 8

About Adapters and BEA WebLogic Integration

The BEA application integration solution uses adapters and application views to help you integrate applications in your enterprise. At its most fundamental level, an *adapter* is software that connects an enterprise information system (EIS) and an integration server. This bi-directional connection consists of *services*—interactions that originate in the adapter (and may require a response from the EIS)—and *events*, interactions that originate in the EIS.

Most EIS systems make selected information and functions available to other applications by way of specialized integration APIs. An adapter connects to the EIS through its integration API, or through database or system calls, and exposes the available functions from the EIS. However, rather than exposing the intricacies of APIs to users, WebLogic Integration incorporates *applications views*—business-oriented interfaces that provide a layer of abstraction between an adapter and the EIS capabilities the adapter exposes.

Figure 1-1 Application View in an Integration Solution



Application views contain definitions for the services and events used by business processes to communicate with an EIS. They also contain connection information and XML schema that define inputs and outputs for services and events. After an adapter is deployed, you can use its Web-based interface to define as many applications views as you need, and other WebLogic Integration components and applications can use that adapter to access data on the EIS.

To learn more about the role of adapters in application integration architecture, see "Key Components of Integration Solutions" on page 1-3.

To learn more about adapters in general, see the *Introduction to the BEA WebLogic Adapters* at the following URL:

`http://edocs.bea.com/wladapters/docs81/index.html`

Key Components of Integration Solutions

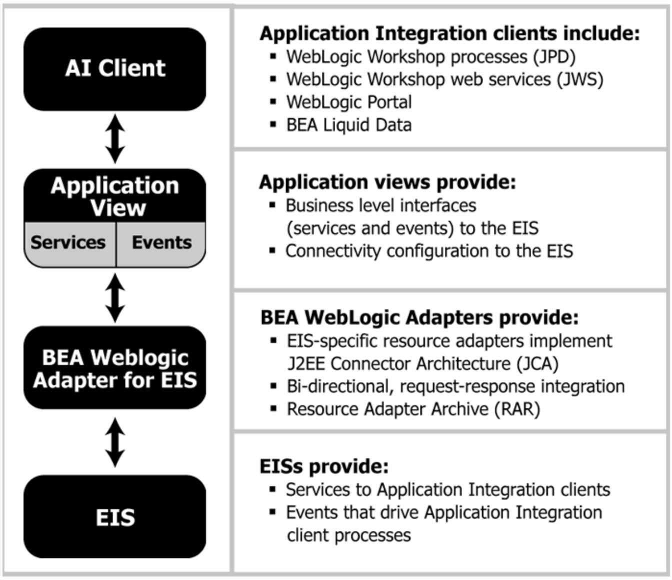
This section describes some of the key concepts you need to be familiar with before you work with an adapter.

- Basic WebLogic Integration Architecture
- Enterprise Information Systems
- Resource Adapters
- Application Views
- Service Clients and Event Consumers
- EIS Metadata, Schemas, and Repositories

Basic WebLogic Integration Architecture

Adapters are used in conjunction with the Application Integration component of BEA WebLogic Integration. This component provides a systematic, standards-based architecture for hosting business-oriented interfaces to enterprise applications.

Figure 1-2 Adapters in the Application Integration Architecture


AI Client
Application View
Services
Events
BEA Weblogic Adapter for EIS
EIS
Application Integration clients include:
▪ WebLogic Workshop processes (JPD)
▪ WebLogic Workshop web services (JWS)
▪ WebLogic Portal
▪ BEA Liquid Data
Application views provide:
▪ Business level interfaces (services and events) to the EIS
▪ Connectivity configuration to the EIS
BEA WebLogic Adapters provide:
▪ EIS-specific resource adapters implement J2EE Connector Architecture (JCA)
▪ Bi-directional, request-response integration
▪ Resource Adapter Archive (RAR)
EISs provide:
▪ Services to Application Integration clients
▪ Events that drive Application Integration client processes

For general information about Application Integration, see the following documents:

- *Introducing Application Integration* at the following URL:

  http://edocs.bea.com/wli/docs81/aiover/index.html

- *Using the Application Integration Design Console* at the following URL:

  http://edocs.bea.com/wli/docs81/aiuser/index.html

Enterprise Information Systems

An *enterprise information system* (EIS) is software that provides the information infrastructure for an enterprise. An EIS offers a set of services to its clients, which are made available to clients via local and/or remote interfaces. An integration solution involves integration with one or more EISs.

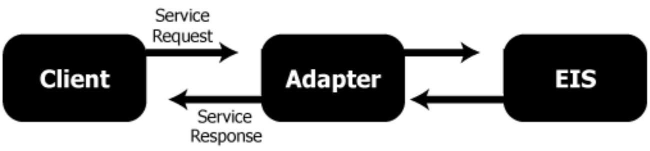
Resource Adapters

A *resource adapter* (or simply *adapter*) is a BEA software component that acts as a connector between an EIS and a J2EE application server (such as BEA WebLogic Server). Each adapter provides bi-directional, request-response integration with a specific application or technology.

Adapters handle two general types of operations:

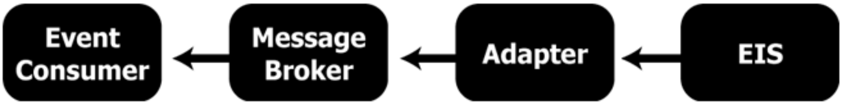
- *Services* are request / response communications with the EIS. Client applications submit service requests to the EIS via the adapter, and the adapter returns the For example, a business process might invoke a PeopleSoft 8 Component Interface or execute a SELECT statement on a database. EIS response back to the client. Responses are either synchronous or asynchronous.

Figure 1-3 Service Invocations



- *Events* are asynchronous, one-way messages received from an EIS. For example, the adapter can receive a PeopleSoft 8 business event or a message from an MQSeries system. The adapter routes the EIS message to the appropriate software component via the WebLogic Integration Message Broker and the Application Integration JMS infrastructure.

Figure 1-4 Event Notifications



In effect, a service is a *request for some work to be done* and an event is a *notification that some work has been done*.

For more information about the specific services and events supported by Adapter for PeopleSoft 8, see "About the BEA WebLogic Adapter for PeopleSoft 8" on page 1-9.

To learn more about the WebLogic Integration Message Broker and the Application Integration JMS infrastructure, see *Introducing Application Integration* at the following URL:

`http://edocs.bea.com/wli/docs81/aiover/index.html`

Application Views

An *application view* is a business oriented interface to objects and operations within an EIS. Application views include the information needed to communicate with the EIS as well as configurations for services and events. Application views define:

To learn more about using application views in business processes, see the WebLogic Workshop documentation at the following URL:

`http://edocs.bea.com/workshop/docs81/doc/en/core/index.html`

You typically define an application view for a specific business process. Therefore, you might have multiple application views defined for a single adapter, each designed to meet a specific requirement.

An application view defines:

- **Communication with the EIS**, including connection settings, login credentials, and so on.
- **Service invocations**, including the information that the EIS requires for the request, as well as any service request and response schemas associated with the service.
- **Event notifications**, including the information that the EIS publishes and the event schemas for inbound messages.

You create application views in either of two ways:

- Using application views—For detailed information about application views, see "Understanding Application Views" in *Introducing Application Integration* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiover/2intfra.html`

- Writing custom code. For more information, see "Using Application Views by Writing Custom Code" in *Using the Application Integration Design Console* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiuser/4usrcust.html`

An application view for Adapter for PeopleSoft 8 provides these features:

- Standards-based data representation. All events, requests, and responses are represented as standards-based XML.

- Abstraction from the details of the EIS. Application views offer a level of abstraction from the details of the underlying EIS, freeing the developers to concentrate on the business processes and data and not on the configuration and details of that system.

To learn more about application views, see Chapter 5, "Defining Application Views for PeopleSoft 8."

Service Clients and Event Consumers

In an integration solution, there are clients that invoke services and consumers for event notifications.

Service Clients

A variety of clients can invoke services on an EIS via an application view. They include BEA WebLogic Workshop business processes, web services, and portals; queries and BEA Liquid Data; and custom Java applications.

For more information, see the following topics in the BEA WebLogic Workshop Help System:

- "Building Integration Applications"

- "Building Web Services"

- "Building Portal Applications"

at the following URL:

`http://edocs.bea.com/workshop/docs81/doc/en/core/index.html`

In addition, see "Using Applications With Business Processes" in *Application Integration Design Console* at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/3usruse.html

Event Consumers

Adapters deliver events using the WebLogic Integration Message Broker, which provides business processes with a channels-based publish and subscribe communication mechanism. Consumers can include BEA WebLogic Workshop business processes, web services, and portals, as well as custom Java applications.

For more information, see the following topics in the BEA WebLogic Workshop Help System:

- "Message Broker Subscription Control" in "Message Broker Controls"
- "Building Integration Applications"

at the following URL:

http://edocs.bea.com/workshop/docs81/doc/en/core/index.html

In addition, see "Receiving Events" in "Using Applications With Business Processes" in *Application Integration Design Console* at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/3usruse.html

EIS Metadata, Schemas, and Repositories

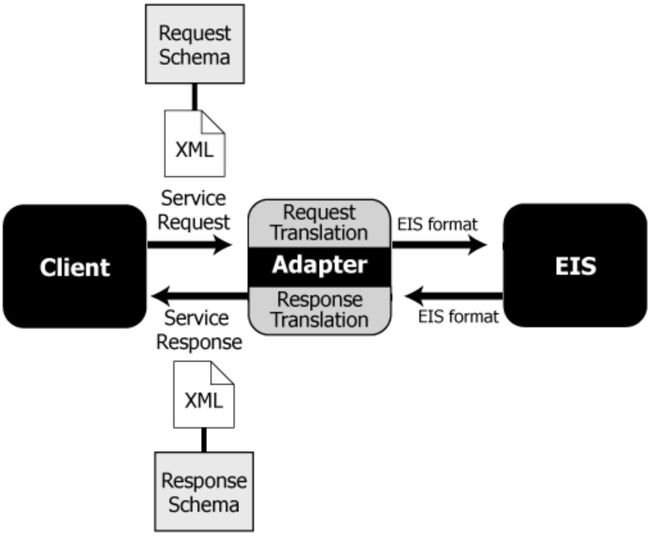
Each EIS uses its own interface to handle service requests and event notifications. Some interfaces are API-based, while others use database or system calls. For example, SAP provides a BAPI interface that defines the parameters and syntax for BAPI requests and responses. For each EIS, the EIS interface defines the *metadata* that applications can use to integrate with the EIS. The EIS publishes data and expects requests in the format dictated by its interface rules and metadata.

Schemas

At run-time, the EIS and the adapter exchange service requests, service responses, and events via XML documents. The adapter handles the data translation between XML documents and the EIS format, using *schemas* that map the data between XML and the EIS format: Depending on the adapter, you either create the schema using the BEA Application Explorer, or have it automatically generated by the adapter when you add services and events to an application view.

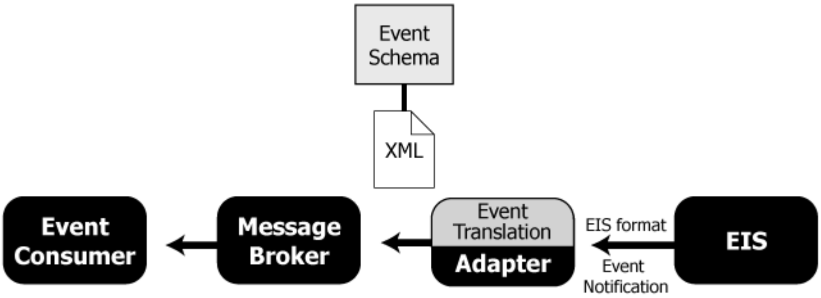
- For service requests, the request arrives at the adapter in the form of an XML document. The adapter uses the *request schema* associated with the service to translate the request to the format that the EIS expects. Similarly, when the adapter receives the response back from the EIS, it uses the *response schema* associated with the service to translate the response to an XML document that the requesting application handles.

Figure 1-5 Adapter Translation for Service Invocations



- For event notifications, the message arrives at the adapter in the format that the EIS uses to publish the event. The adapter uses the *event schema* associated with the event to translate the response to an XML document that the subscribed application handles.

Figure 1-6 Adapter Translation for Event Notifications



To learn more about schemas, see Chapter 4, "Generating Schemas for PeopleSoft 8 Component Interfaces."

Repositories

Once you have created the necessary schemas, you save them in a file-based *repository*, along with a manifest file that associates the schemas with events and services. When you configure application views in the Application View Console, you specify the location of the repository so that the application view can find the schemas as needed. For more information, see Chapter 5, "Defining Application Views for PeopleSoft 8."

About the BEA WebLogic Adapter for PeopleSoft 8

The BEA WebLogic Adapter for PeopleSoft 8 connects to your PeopleSoft system so that you can easily use your PeopleSoft data and functions within your business processes. The adapter provides scalable, reliable, and secure access to your PeopleSoft system.

This section includes the following topics:

- Supported PeopleSoft Operations for Application Integration
- Supported Services
- Supported Events
- Benefits of the Adapter for PeopleSoft 8

Supported PeopleSoft Operations for Application Integration

The Adapter for PeopleSoft 8 supports synchronous and asynchronous, bi-directional message interactions with the PeopleSoft Component Interface and PeopleSoft Application Messaging Manager facilities.

It provides integration with the following PeopleSoft operations:

- Access to the PeopleSoft Component Interface framework using APIs for the component interfaces that correspond to supported PeopleSoft business objects
- Access to PeopleSoft Application Messaging features using PeopleSoft XML

About the PeopleSoft Component Interface

A PeopleSoft Component Interface is a container used to:

- Distribute PeopleSoft application data among PeopleSoft logical systems
- Exchange PeopleSoft application data with non-PeopleSoft systems

The Component Interface is based on an existing business process within the PeopleSoft system, such as a purchase order entry. This business process can be one delivered by PeopleSoft or developed by a user. The component interface also inherits its methods (Add, Update, and so on) and business logic from the underlying business process.

PeopleSoft delivers a number of generic component interfaces—called Enterprise Integration Points (EIPs)—with each PeopleSoft application. In addition, PeopleSoft 8 users can modify existing EIPs or develop their own custom component interfaces.

About PeopleSoft Application Messaging

PeopleSoft Application Messaging facilitates integration to PeopleSoft 8 using PeopleSoft XML. The BEA WebLogic Adapter for PeopleSoft 8 uses a PeopleSoft handler that must be configured within the PeopleSoft application gateway using TCP/IP transport services.

Supported Services

The Adapter for PeopleSoft 8 supports a service that allows you to access and integrate PeopleSoft business objects using component interfaces and the Java API.

Supported Events

The Adapter for PeopleSoft 8 supports two types of events, one for each type of PeopleSoft 8 Transport: TCP/IP and HTTP. In each case, the adapter picks up a PeopleSoft XML file and passes it to an event variable within a business process.

These are the events supported by Adapter for PeopleSoft 8.

- TCP/IP event, in which PeopleSoft sends an XML file that represents a PeopleSoft business event via TCP/IP to WebLogic Integration.
- HTTP event, in which PeopleSoft sends an XML file that represents a PeopleSoft business event via HTTP to WebLogic Integration.

Benefits of the Adapter for PeopleSoft 8

The combination of the adapter and WebLogic Integration supplies everything you need to integrate your workflows and enterprise applications with your PeopleSoft 8 system.

- Integration can be achieved without custom coding.
- Business processes can be started by events generated by PeopleSoft 8.
- Business processes can request and receive data from your PeopleSoft 8 system using services.
- Adapter events and services are standards-based. The adapter services and events provide extensions to the *J2EE Connector Architecture* (JCA) version 1.0 from Sun Microsystems, Inc. For more information, see the Sun JCA page at the following URL:

  `http://java.sun.com/j2ee/connector/`
- The adapter and WebLogic Integration solution is scalable. The BEA WebLogic Platform provides clustering, load balancing, and resource pooling for a scalable solution. For more information about scalability, see the following URL:

  `http://e-docs.bea.com/wls/docs81/cluster/index.html`
- The adapter and WebLogic Integration solution benefits from the fault-tolerant features of the BEA WebLogic Platform. For more information about high availability, see the following URL:

  `http://edocs.bea.com/wli/docs81/deploy/index.html`
- The adapter and WebLogic Integration solution is secure, using the security features of the BEA WebLogic Platform and the security of your PeopleSoft 8 system. For more information about security, see the following URL:

  `http://edocs.bea.com/wls/docs81/secintro/index.html`

Getting Started With the Adapter for PeopleSoft 8

This section gives an overview of how to get started using the BEA WebLogic Adapter for PeopleSoft 8 within the context of an application integration solution. Integration with PeopleSoft 8 involves the following tasks:

- Step 1: Design the Application Integration Solution
- Step 2: Determine the Required PeopleSoft Component Interfaces and Application Messages
- Step 3: Generate Schemas for PeopleSoft Component Interfaces and Application Messages
- Step 4: Define Application Views and Configure Services and Events
- Step 5: Integrate with Other BEA Software Components

- Step 6: Deploy the Solution to the Production Environment

Step 1: Design the Application Integration Solution

The first step is to design an application integration solution, which includes (but is not limited to) such tasks as:

- Defining the overall scope of application integration.
- Determining the business process(es) to integrate.
- Determining which WebLogic Platform components will be involved in the integration, such as web services or workflows designed in WebLogic Workshop, portals created in WebLogic Portal, and so on.
- Determining which external systems and technologies will be involved in the integration, such as PeopleSoft 8 systems and other EISs.
- Determining which BEA WebLogic Adapters for WebLogic Integration will be required, such as the BEA WebLogic Adapter for PeopleSoft 8. An application integration solution can involve multiple adapters.

This step involves the expertise of business analysts, system integrators, and EIS specialists (including PeopleSoft 8 specialists). Note that an application integration solution can be part of a larger integration solution.

To learn more about designing an application integration solution, see *Designing WebLogic Integration Solutions* at the following URL:

`http://edocs.bea.com/wli/docs81/design/index.html`

Step 2: Determine the Required PeopleSoft Component Interfaces and Application Messages

Within the larger context of an application integration project, you must determine which PeopleSoft 8 component interfaces are required for adapter services and which application messages are required to generate events. Before you set up a service or event in an application view, you must have the appropriate component interface or application message set up in PeopleSoft 8.

Factors to consider include (but are not limited to):

- Type of PeopleSoft 8 component interfaces and application messages required for services and events

- PeopleSoft 8 business events involved in WLI business processes
- Logins required to access the PeopleSoft system and access the required business events
- Whether operations are, from the adapter point of view:
  - services, which notify the PeopleSoft system via a PeopleSoft component interface, and, in addition, whether such services should be processed synchronously or asynchronously
  - events, which are notifications from the PeopleSoft 8 system that trigger workflows

This step involves the expertise of PeopleSoft 8 specialists, including analysts and administrators.

Step 3: Generate Schemas for PeopleSoft Component Interfaces and Application Messages

After identifying and creating the PeopleSoft component interfaces and application messages required for the application integration solution, you must generate the XML schemas that will be used to exchange data with one or more PeopleSoft 8 systems:

- Services require two XML schemas: one for the request and another for the response.
- Events require a single XML schema to handle the data sent by the PeopleSoft system.

You use the BEA Application Explorer tool to generate schemas for PeopleSoft 8 component interfaces and application messages. To learn more about schemas, see Chapter 4, "Generating Schemas for PeopleSoft 8 Component Interfaces."

Step 4: Define Application Views and Configure Services and Events

After you create the schemas for your PeopleSoft component interfaces and application messages, you create an application view that provides an XML-based interface between WebLogic Server and a particular PeopleSoft 8 system within your enterprise. If you are accessing multiple PeopleSoft 8 systems, you define a separate application view for each PeopleSoft 8 system you want to access. To provide different levels of security access (such as "guest" and "administrator"), define a separate application view for each security level.

Once you define an application view, you can configure events and services in that application view that employ the XML schemas that you created in "Step 3: Generate Schemas for PeopleSoft Component Interfaces and Application Messages" on page 1-13. To learn more about generating schemas, see Chapter 4, “Generating Schemas for PeopleSoft 8 Component Interfaces.”

To learn more about defining application views, see Chapter 5, “Defining Application Views for PeopleSoft 8” in conjunction with *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 5: Integrate with Other BEA Software Components

Once you have configured and published one or more application views for PeopleSoft 8 integration, you can integrate these application views into other BEA software components, such as workflows or web services created in BEA WebLogic Workshop, or portals built with BEA WebLogic Portal.

For more information, see *Using the Application Integration Design Console*, particularly Chapter 3, “Using Application Views with Application Workflows,” at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 6: Deploy the Solution to the Production Environment

After you have designed, built, and tested your application integration solution, you can deploy it into a production environment. The following list describes some of the tasks involved in deploying an application integration:

- Design the deployment.
- Deploy the required components of the BEA WebLogic Platform.
- Install and deploy the BEA WebLogic Adapter for PeopleSoft 8 as described in *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide*
- Deploy your application views and schemas for PeopleSoft 8 integration.
- Verify business processes in the production environment.
- Monitor and tune the deployment.

To learn more about deploying your application integration solution, see *Deploying WebLogic Integration Solutions* at the following URL:

`http://edocs.bea.com/wls/docs81/deployment/index.html`

CHAPTER 2

Using PeopleSoft 8 Component Interfaces

Before you can create service schemas and services, you must first create PeopleSoft component interfaces based on the PeopleSoft business processes that you want to access. This section describes how to create and test PeopleSoft component interfaces for use with the BEA WebLogic Adapter for PeopleSoft 8.

It includes the following topics:

- About Creating Component Interfaces
- Creating a Component Interface
- Configuring Component Interface Security
- Testing a Component Interface
- Generating Component Interface APIs

About Creating Component Interfaces

PeopleSoft uses component interfaces to provide access to the underlying business logic and data associated with a component interface.

Before the BEA WebLogic Adapter for PeopleSoft 8 can use a component interface, you must create the interface within PeopleSoft and compile it outside of the PeopleSoft environment. You make the resulting files available to the WebLogic environment in the `CLASSPATH` variable in the `startWebLogic.cmd` file.

When you create component interfaces, you use the PeopleSoft Application Designer to select the source component and add or edit methods and properties as needed. You also set security for the component interface and test it. For more information about using the PeopleSoft Application Designer, see your PeopleBooks documentation.

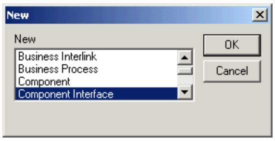
Creating a Component Interface

When you create a component interface, you select the source component and then add or modify methods and properties.

To create a component interface:

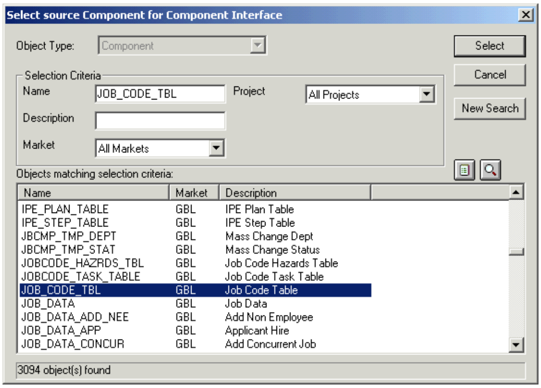
1. Launch the PeopleSoft Application Designer.
2. Choose File → New.

   The New dialog box appears.

3. Select Component Interface and click OK.

   The Select Source Component for Component Interface window appears, displaying the available components.


Select source Component for Component Interface

Object Type: Component

Selection Criteria

Name: JOB_CODE_TBL

Project: All Projects

Description:

Market: All Markets

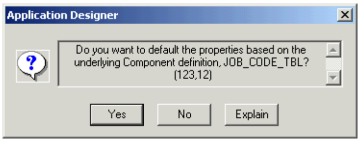
Objects matching selection criteria:

Name	Market	Description
IPE_PLAN_TABLE	GBL	IPE Plan Table
IPE_STEP_TABLE	GBL	IPE Step Table
JBCMP_TMP_DEPT	GBL	Mass Change Dept
JBCMP_TMP_STAT	GBL	Mass Change Status
JOBCODE_HAZRDS_TBL	GBL	Job Code Hazards Table
JOBCODE_TASK_TABLE	GBL	Job Code Task Table
JOB_CODE_TBL	GBL	Job Code Table
JOB_DATA	GBL	Job Data
JOB_DATA_ADD_NEE	GBL	Add Non Employee
JOB_DATA_APP	GBL	Applicant Hire
JOB_DATA_CONCUR	GBL	Add Concurrent Job

3094 object(s) found

4. Select the component to use as a basis for the component interface and click Select.

   The Application Designer dialog box appears.


Application Designer

Do you want to default the properties based on the underlying Component definition, JOB_CODE_TBL? (123,12)

5. Choose whether to display the underlying properties for the component interface.

   – Click Yes to start creating the component interface with the properties of the underlying component interface displayed.

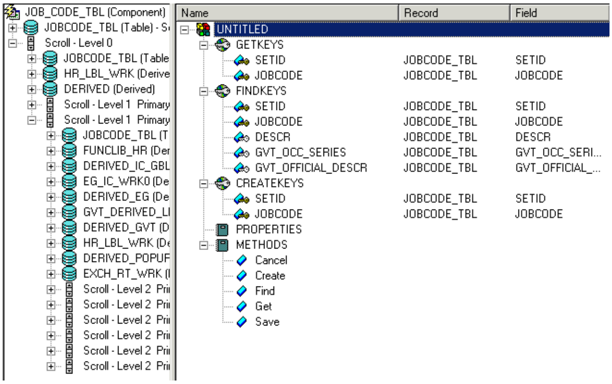
   – Click No to create the component interface without displaying properties.

   This option allows you to select components manually by dragging them from the left pane in the Application Designer to the right pane. You can right-click both panes to select various functions to perform depending on which pane is active.

   Note: If the component interface is large, you may want to expose the component properties manually.

   For a complete list of functions, see your PeopleBooks documentation.

   The Application Designer displays the properties and methods in your component interface.



JOB_CODE_TBL (Component)
JOBCODE_TBL (Table) - S
Scroll - Level 0
JOBCODE_TBL (Table
HR_LBL_WRK (Derive
DERIVED (Derived)
Scroll - Level 1 Primary
Scroll - Level 1 Primary
JOBCODE_TBL (T
FUNCLIB_HR (Der
DERIVED_IC_GBL
EG_IC_WRK0 (De
DERIVED_EG (De
GVT_DERIVED_LI
DERIVED_GVT (D
HR_LBL_WRK (D
DERIVED_POPUF
EXCH_RT_WRK (I
Scroll - Level 2 Pri
Scroll - Level 2 Pri
Scroll - Level 2 Pri
Scroll - Level 2 Pri
Scroll - Level 2 Pri
Scroll - Level 2 Pri

Name	Record	Field
UNTITLED		
GETKEYS		
SETID	JOBCODE_TBL	SETID
JOBCODE	JOBCODE_TBL	JOBCODE
FINDKEYS		
SETID	JOBCODE_TBL	SETID
JOBCODE	JOBCODE_TBL	JOBCODE
DESCR	JOBCODE_TBL	DESCR
GVT_OCC_SERIES	JOBCODE_TBL	GVT_OCC_SERI...
GVT_OFFICIAL_DESCR	JOBCODE_TBL	GVT_OFFICIAL_...
CREATEKEYS		
SETID	JOBCODE_TBL	SETID
JOBCODE	JOBCODE_TBL	JOBCODE
PROPERTIES		
METHODS		
Cancel		
Create		
Find		
Get		
Save		

Creating User-Defined Methods

The standard component interface methods are Create, Find, Get, and Save. The ones that are available depend on the capabilities of the underlying component. For example, if the underlying component does not contain Add capabilities, the Create method is not available. You can add methods, called user-defined methods, after you save the component interface.

To create a user-defined method:

1. Right-click anywhere in the component interface view and select View.

2. Select View Peoplecode from the shortcut menu, or select an object in the component interface and choose View → Application Designer → View Peoplecode.

   The Peoplecode contains implementations of the component's methods.

3. Add a new method to the Peoplecode.

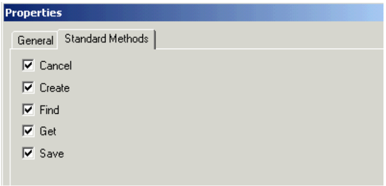
   For more information, see your PeopleBooks documentation.

Viewing or Changing Available Methods

You can view the available methods for a component interface and change them if necessary.

To view or change available methods:

1. Open the Component Interface Properties dialog box and click the Standard Methods tab.



☑ Cancel
☑ Create
☑ Find
☑ Get
☑ Save

2. Select the desired methods and click OK.
3. View the Peoplecode and modify the methods as necessary.

   For more information, see your PeopleBooks documentation.

About Properties

You can add properties from the records in the component view or delete any property in the component interface that you do not want to expose. You can also rename a property by selecting it and typing a new name. However, if you rename a property, it can be referenced in the component interface only by the new name, not by the underlying component name.

Configuring Component Interface Security

You must set up security for the component interface before you can begin testing. The way in which you set it up depends on your PeopleSoft version. This section contains these topics:

- Configuring Component Interface Security for PeopleSoft Version 8.1x
- Configuring Component Interface Security for PeopleSoft Version 8.4 or Higher

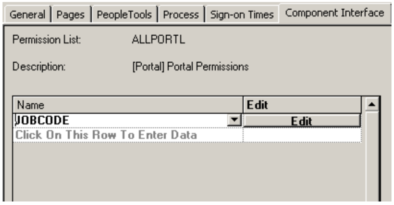
Configuring Component Interface Security for PeopleSoft Version 8.1x

This section describes how to configure component interface security for PeopleSoft Version 8.1x.

To configure component interface security for PeopleSoft version 8.1x:

1. Choose Go → PeopleTools → Maintain Security → Use → Permission Lists → Component Interface → Update/Display.
2. Choose the relevant Permission list.

Before Security can be set, the Permission List(s) to be used must already be identified. For more information on Permission Lists, see the PeopleBooks documentation.



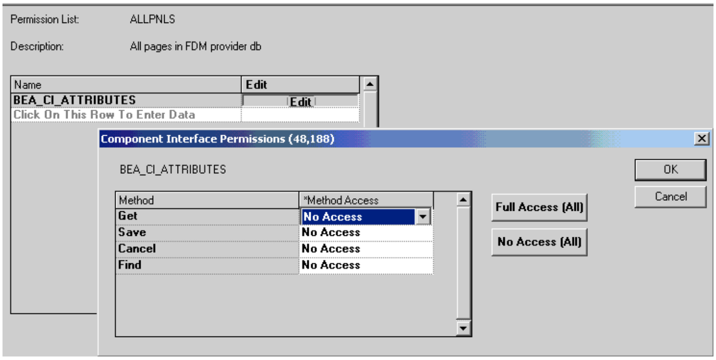
Permission List: ALLPORTL

Description: [Portal] Portal Permissions

Name	Edit
JOBCODE	Edit
Click On This Row To Enter Data	

3. Insert the new Component Interface that you created and click Edit.

   The Component Interface displays all available methods, including user-defined methods. This enables you to specify whether this particular Permission List should have Full or Partial Access.



Permission List: ALLPNLS

Description: All pages in FDM provider db

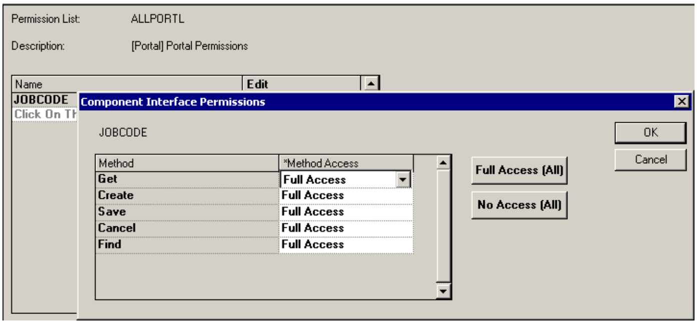
Name	Edit
BEA_CI_ATTRIBUTES	Edit
Click On This Row To Enter Data	

BEA_CI_ATTRIBUTES

Method	*Method Access
Get	No Access
Save	No Access
Cancel	No Access
Find	No Access

4. Select the desired access for this Permission List and click OK.

   In the following example, the ALLPORTL Permission List has Full Access for all methods.



Permission List: ALLPORTL

Description: [Portal] Portal Permissions

Name	Edit
JOBCODE	
Click On Th	

JOBCODE

Method	*Method Access
Get	Full Access
Create	Full Access
Save	Full Access
Cancel	Full Access
Find	Full Access

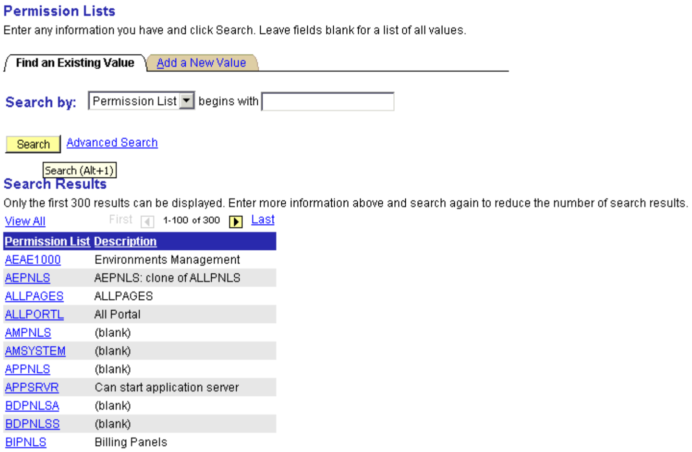
Configuring Component Interface Security for PeopleSoft Version 8.4 or Higher

This section describes how to configure component interface security for PeopleSoft Version 8.4 or higher.

To configure component interface security for PeopleSoft version 8.4 or higher:

1. Click PeopleTools → Security → User Profiles → Permissions & Roles → Permission Lists.
2. Click the Search button.

   The Permissions List search window opens.



Permission Lists

Enter any information you have and click Search. Leave fields blank for a list of all values.

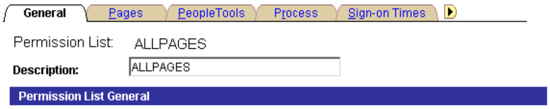
Search Results

Only the first 300 results can be displayed. Enter more information above and search again to reduce the number of search results.

Permission List	Description
AEAE1000	Environments Management
AEPNLS	AEPNLS: clone of ALLPNLS
ALLPAGES	ALLPAGES
ALLPORTL	All Portal
AMPNLS	(blank)
AMSYSTEM	(blank)
APPNLS	(blank)
APPSRVR	Can start application server
BDPNLSA	(blank)
BDPNLSS	(blank)
BIPNLS	Billing Panels

3. Select the relevant Permission List.

   The Permissions List window with several tabs opens.

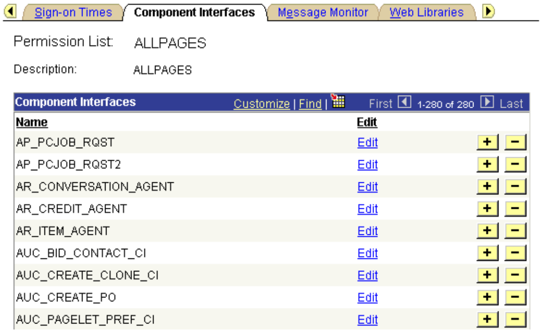


Permission List: ALLPAGES

Description: ALLPAGES

Permission List General

4. Click the arrow to the right of the Sign-on Times tab to display additional tabs. Click the Component Interfaces tab.



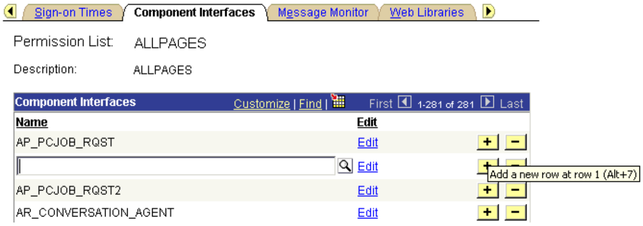
Permission List: ALLPAGES

Description: ALLPAGES

Component Interfaces — Customize | Find | First 1-280 of 280 Last

Name	Edit	
AP_PCJOB_RQST	Edit	+ −
AP_PCJOB_RQST2	Edit	+ −
AR_CONVERSATION_AGENT	Edit	+ −
AR_CREDIT_AGENT	Edit	+ −
AR_ITEM_AGENT	Edit	+ −
AUC_BID_CONTACT_CI	Edit	+ −
AUC_CREATE_CLONE_CI	Edit	+ −
AUC_CREATE_PO	Edit	+ −
AUC_PAGELET_PREF_CI	Edit	+ −

5. Click the + button to add a new row to the Component Interfaces list.



Permission List: ALLPAGES

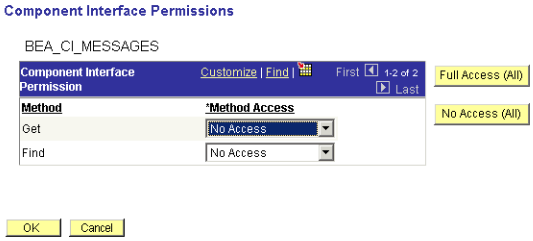
Description: ALLPAGES

Component Interfaces — Customize | Find | First 1-281 of 281 Last

Name	Edit	
AP_PCJOB_RQST	Edit	+ −
	Edit	+ −
AP_PCJOB_RQST2	Edit	+ −
AR_CONVERSATION_AGENT	Edit	+ −

6. Enter the component interface name and click Edit.

   In this example, a component interface used by the adapter opens.



Component Interface Permissions

BEA_CI_MESSAGES

Component Interface Permission — Customize | Find | First 1-2 of 2 Last

Method	*Method Access
Get	No Access
Find	No Access

7. Select Full Access for the Get and Find methods. Click OK.

8. After you have configured security for your component interface, scroll to the bottom of the Component Interface Permissions window and click Save.

Testing a Component Interface

The Adapter for PeopleSoft 8 uses PeopleSoft metadata and component interfaces, which enables it to accommodate new or modified component interfaces. The adapter makes no assumptions about component interfaces except that they are logical and valid. Therefore, you should test each component interface before you use it as a source for the adapter.

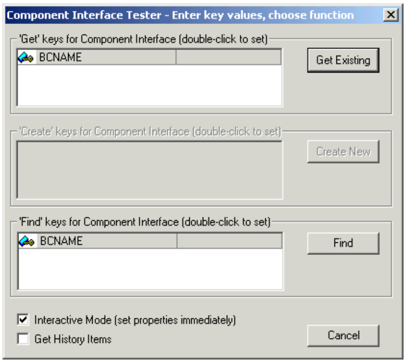
Note: If you make changes to the underlying data or if the component interface is made invalid as a result of a PeopleSoft upgrade, you must repair the invalid component interface before the adapter uses it.

While testing a component interface, you can use these options in the Component Interface Tester dialog box:

- Find displays all possible entries for the underlying component.
- Get Existing displays the exposed properties for the key you enter.
- Create New lets you select, create, and update the properties of the underlying component.

To test a component interface:

1. In the PeopleSoft Application Designer, choose Tools → Test Component Interface. Then, click the minimized Component Interface Tester dialog box to bring it to the foreground.



Component Interface Tester - Enter key values, choose function

'Get' keys for Component Interface (double-click to set)

BCNAME

'Create' keys for Component Interface (double-click to set)

'Find' keys for Component Interface (double-click to set)

BCNAME

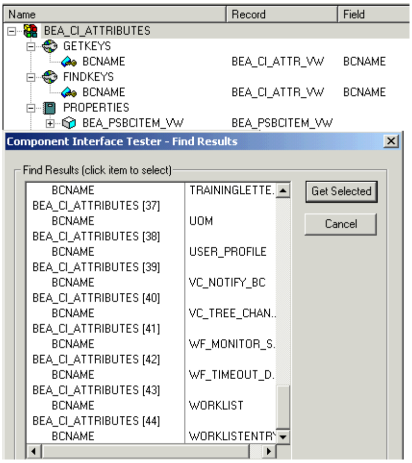
☑ Interactive Mode (set properties immediately)

☐ Get History Items

In this example, the Create New option is grayed out because the Add method is not applicable to this component.

2. Click Find.

The Component Interface Tester displays the results.



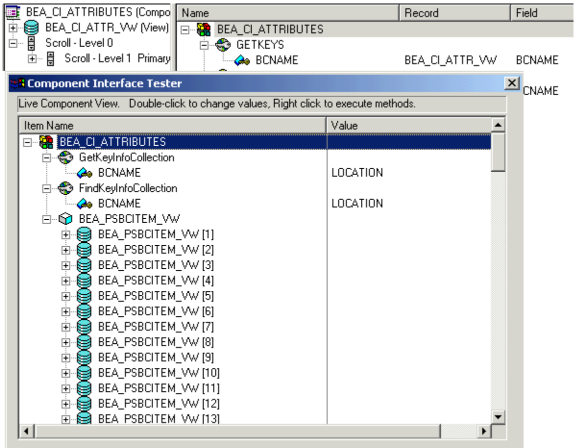
Name	Record	Field
BEA_CI_ATTRIBUTES		
GETKEYS		
BCNAME	BEA_CI_ATTR_VW	BCNAME
FINDKEYS		
BCNAME	BEA_CI_ATTR_VW	BCNAME
PROPERTIES		
BEA_PSBCITEM_VW	BEA_PSBCITEM_VW	

Component Interface Tester - Find Results

Find Results (click item to select)

BCNAME	TRAININGLETTE...
BEA_CI_ATTRIBUTES [37]	
BCNAME	UOM
BEA_CI_ATTRIBUTES [38]	
BCNAME	USER_PROFILE
BEA_CI_ATTRIBUTES [39]	
BCNAME	VC_NOTIFY_BC
BEA_CI_ATTRIBUTES [40]	
BCNAME	VC_TREE_CHAN...
BEA_CI_ATTRIBUTES [41]	
BCNAME	WF_MONITOR_S...
BEA_CI_ATTRIBUTES [42]	
BCNAME	WF_TIMEOUT_D...
BEA_CI_ATTRIBUTES [43]	
BCNAME	WORKLIST
BEA_CI_ATTRIBUTES [44]	
BCNAME	WORKLISTENTR

3. To display data for a key, select it in the left pane and click the Get Selected button.



BEA_CI_ATTRIBUTES (Compo
BEA_CI_ATTR_VW (View)
Scroll - Level 0
Scroll - Level 1 Primary

Name	Record	Field
BEA_CI_ATTRIBUTES		
GETKEYS		
BCNAME	BEA_CI_ATTR_VW	BCNAME
		CNAME

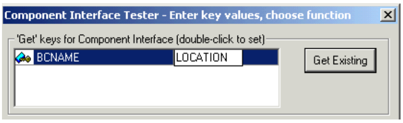
Component Interface Tester

Live Component View. Double-click to change values, Right click to execute methods.

Item Name	Value
BEA_CI_ATTRIBUTES	
GetKeyInfoCollection	
BCNAME	LOCATION
FindKeyInfoCollection	
BCNAME	LOCATION
BEA_PSBCITEM_VW	
BEA_PSBCITEM_VW [1]	
BEA_PSBCITEM_VW [2]	
BEA_PSBCITEM_VW [3]	
BEA_PSBCITEM_VW [4]	
BEA_PSBCITEM_VW [5]	
BEA_PSBCITEM_VW [6]	
BEA_PSBCITEM_VW [7]	
BEA_PSBCITEM_VW [8]	
BEA_PSBCITEM_VW [9]	
BEA_PSBCITEM_VW [10]	
BEA_PSBCITEM_VW [11]	
BEA_PSBCITEM_VW [12]	
BEA_PSBCITEM_VW [13]	

If Read-Only access is not specified in the Permission List, you can change the values in the field.

4. To use the Get option, enter an existing Key by clicking the Get Existing button.



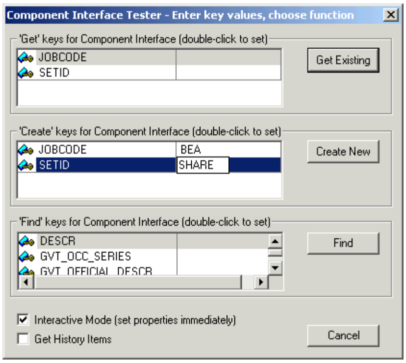
Component Interface Tester - Enter key values, choose function

'Get' keys for Component Interface (double-click to set)

BCNAME	LOCATION

This returns the exposed properties for the key entered. You can change values if Update access has been specified.

5. To use the Create option, Enter the relevant Create keys to create, update, and select the properties of the underlying component.



Component Interface Tester - Enter key values, choose function

'Get' keys for Component Interface (double-click to set)

JOBCODE	
SETID	

'Create' keys for Component Interface (double-click to set)

JOBCODE	BEA
SETID	SHARE

'Find' keys for Component Interface (double-click to set)

DESCR	
GVT_OCC_SERIES	
GVT_OFFICIAL_DESCR	

☑ Interactive Mode (set properties immediately)

☐ Get History Items

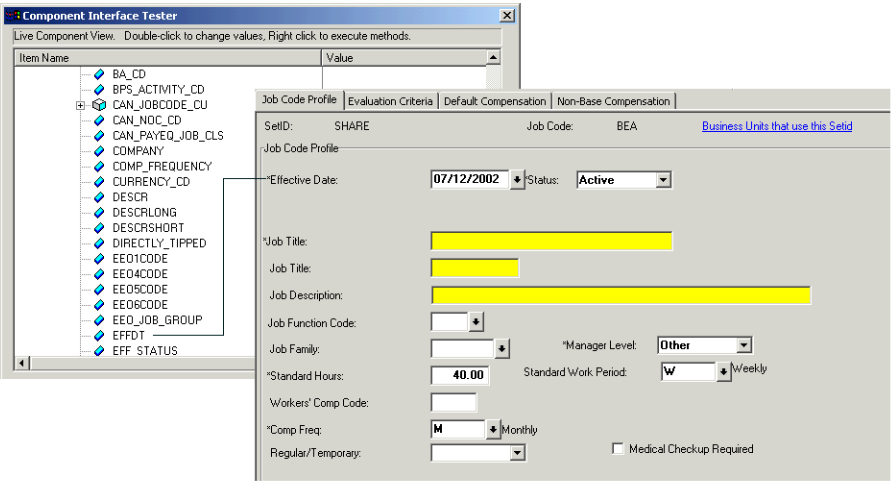
6. Enter valid values in Create keys and expand the Table name with default data in place. The JOBCODE Data Display window opens.

7. Make the required changes. Any changes are validated against the underlying Business Logic for the component.

8. When you finish making changes, right-click the top item in the pane and choose Save.

   You can now use the keys you created with the Get Method for viewing data.

   You can view the data you added in the PeopleSoft Component. Note that the Effective Date is one of the default values.



Component Interface Tester

Live Component View. Double-click to change values, Right click to execute methods.

Item Name	Value
BA_CD	
BPS_ACTIVITY_CD	
CAN_JOBCODE_CU	
CAN_NOC_CD	
CAN_PAYEQ_JOB_CLS	
COMPANY	
COMP_FREQUENCY	
CURRENCY_CD	
DESCR	
DESCRLONG	
DESCRSHORT	
DIRECTLY_TIPPED	
EEO1CODE	
EEO4CODE	
EEO5CODE	
EEO6CODE	
EEO_JOB_GROUP	
EFFDT	
EFF_STATUS	

SetID: SHARE Job Code: BEA

Job Code Profile

*Effective Date: 07/12/2002 *Status: Active

*Job Title:

Job Title:

Job Description:

Job Function Code:

Job Family: *Manager Level: Other

*Standard Hours: 40.00 Standard Work Period: W Weekly

Workers' Comp Code:

*Comp Freq: M Monthly

Regular/Temporary: ☐ Medical Checkup Required

Generating Component Interface APIs

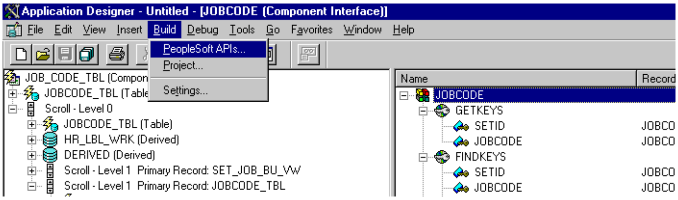
Regardless of whether you are using a PeopleSoft supplied Enterprise Integration Point (EIP) or a custom-developed Component Interface, a PeopleSoft API must be created to enable communications with the PeopleSoft application. This API is simply a collection of Java class files which reside on the client machine and intermediate between the client application layer and PeopleSoft.

Building the PeopleSoft API Java Programs

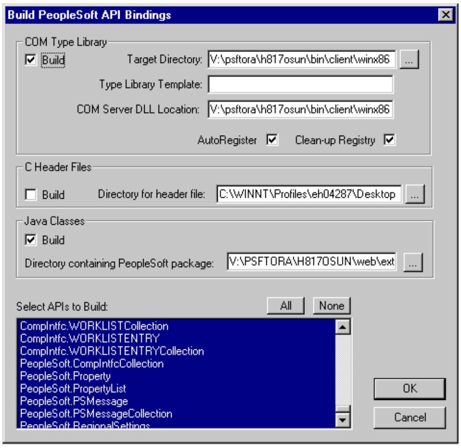
To build the PeopleSoft API Java programs:

1. From the PeopleSoft Application Designer, open any Component Interface.

2. Click the right pane, then select Build → PeopleSoft APIs.



The Build PeopleSoft API Binding dialog box appears.


Build PeopleSoft API Bindings

COM Type Library

☑ Build Target Directory: V:\psftora\h817osun\bin\client\winx86

Type Library Template:

COM Server DLL Location: V:\psftora\h817osun\bin\client\winx86

AutoRegister ☑ Clean-up Registry ☑

C Header Files

☐ Build Directory for header file: C:\WINNT\Profiles\eh04287\Desktop

Java Classes

☑ Build

Directory containing PeopleSoft package: V:\PSFTORA\H817OSUN\web\ext

Select APIs to Build:

CompIntfc.WORKLISTCollection
CompIntfc.WORKLISTENTRY
CompIntfc.WORKLISTENTRYCollection
PeopleSoft.CompIntfcCollection
PeopleSoft.Property
PeopleSoft.PropertyList
PeopleSoft.PSMessage
PeopleSoft.PSMessageCollection
PeopleSoft.RegionalSettings

3. Since you are creating Java files, turn off the COM Type Library Build setting.

4. Select a directory on your local machine where the Java files will be placed.

   For example: `c:\psoft_components`.

5. Select the APIs to build.

   You can choose the default of ALL APIs (which can get quite large) or you may select individual APIs.

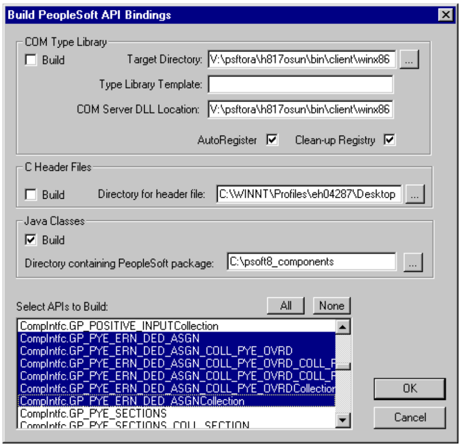
6. To build all files, click OK.

   PeopleSoft generates the files. This will take a few minutes. Once the process is complete, a message appears in the output window. You are now ready to compile the Java files, as described in "Compiling the PeopleSoft API Java Programs" on page 2-14.

7. To create APIs for a specific Component Interface or Component Interfaces, click None. This clears the selected APIs.

8. Select the APIs appropriate for your Component Interface.

   The API names all begin with the name of your Component Interface. There may be fewer than five or more than 50 APIs to build for a particular Component Interface. The following figure illustrates the `GP_PYE_ERN_DED_ASGN` Component Interface from the HR 8.1 application.



Build PeopleSoft API Bindings

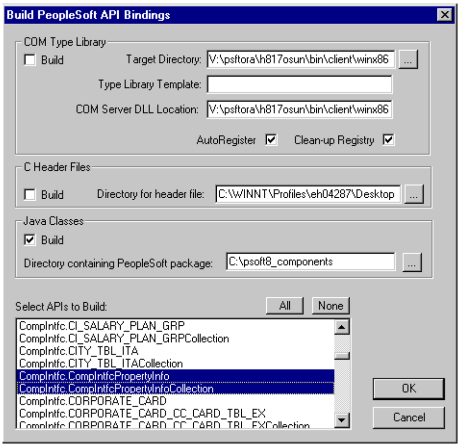
COM Type Library
☐ Build
Target Directory: V:\psftora\h817osun\bin\client\winx86
Type Library Template:
COM Server DLL Location: V:\psftora\h817osun\bin\client\winx86
AutoRegister ☑ Clean-up Registry ☑

C Header Files
☐ Build
Directory for header file: C:\WINNT\Profiles\eh04287\Desktop

Java Classes
☑ Build
Directory containing PeopleSoft package: C:\psoft8_components

Select APIs to Build:
CompIntfc.GP_POSITIVE_INPUTCollection
CompIntfc.GP_PYE_ERN_DED_ASGN
CompIntfc.GP_PYE_ERN_DED_ASGN_COLL_PYE_OVRD
CompIntfc.GP_PYE_ERN_DED_ASGN_COLL_PYE_OVRD_COLL_F
CompIntfc.GP_PYE_ERN_DED_ASGN_COLL_PYE_OVRD_COLL_F
CompIntfc.GP_PYE_ERN_DED_ASGN_COLL_PYE_OVRDCollectio
CompIntfc.GP_PYE_ERN_DED_ASGNCollection
CompIntfc.GP_PYE_SECTIONS
CompIntfc.GP_PYE_SECTIONS_COLL_SECTION

9. In addition to the APIs for your chosen Component Interface, you must also generate the API files for the generic Component Interface properties. You may select these items in the same step as the Component Interface build or they may be done separately.



Build PeopleSoft API Bindings

COM Type Library

☐ Build Target Directory: V:\psftora\h817osun\bin\client\winx86

Type Library Template:

COM Server DLL Location: V:\psftora\h817osun\bin\client\winx86

AutoRegister ☑ Clean-up Registry ☑

C Header Files

☐ Build Directory for header file: C:\WINNT\Profiles\eh04287\Desktop

Java Classes

☑ Build

Directory containing PeopleSoft package: C:\psoft8_components

Select APIs to Build:

CompIntfc.CI_SALARY_PLAN_GRP
CompIntfc.CI_SALARY_PLAN_GRPCollection
CompIntfc.CITY_TBL_ITA
CompIntfc.CITY_TBL_ITACollection
CompIntfc.CompIntfcPropertyInfo
CompIntfc.CompIntfcPropertyInfoCollection
CompIntfc.CORPORATE_CARD
CompIntfc.CORPORATE_CARD_CC_CARD_TBL_EX
CompIntfc.CORPORATE_CARD_CC_CARD_TBL_EXCollection

10. Click OK.

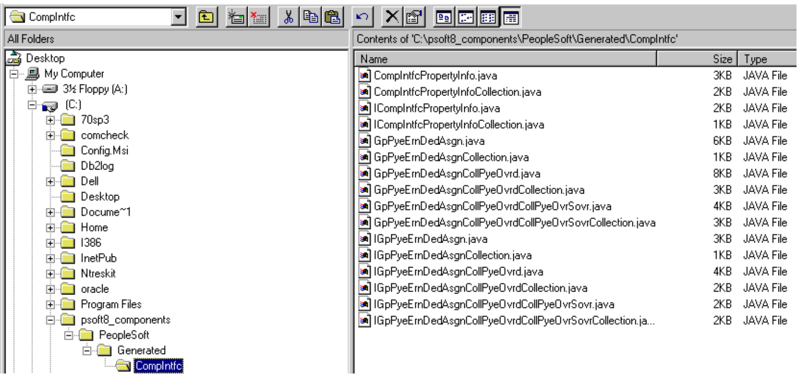
    PeopleSoft generates the files. This will take a few minutes. Once the process is complete, a message appears in the output window. You are now ready to compile the Java files, as described in "Compiling the PeopleSoft API Java Programs" on page 2-14.

Compiling the PeopleSoft API Java Programs

PeopleSoft places the Java programs to be compiled in a directory structure

psoft8_components`\PeopleSoft\generated\CompIntfc`;

here, *psoft8_components* is the directory specified during the build process. If you chose to generate all APIs, a second directory, *psoft8_components*`\PeopleSoft\generated\PeopleSoft`, is also created. You do not need to access it.



Contents of 'C:\psoft8_components\PeopleSoft\Generated\CompIntfc'

Name	Size	Type
CompIntfcPropertyInfo.java	3KB	JAVA File
CompIntfcPropertyInfoCollection.java	2KB	JAVA File
ICompIntfcPropertyInfo.java	2KB	JAVA File
ICompIntfcPropertyInfoCollection.java	1KB	JAVA File
GpPyeErnDedAsgn.java	6KB	JAVA File
GpPyeErnDedAsgnCollection.java	1KB	JAVA File
GpPyeErnDedAsgnCollPyeOvrd.java	8KB	JAVA File
GpPyeErnDedAsgnCollPyeOvrdCollection.java	3KB	JAVA File
GpPyeErnDedAsgnCollPyeOvrdCollPyeOvrSovr.java	4KB	JAVA File
GpPyeErnDedAsgnCollPyeOvrdCollPyeOvrSovrCollection.java	3KB	JAVA File
IGpPyeErnDedAsgn.java	3KB	JAVA File
IGpPyeErnDedAsgnCollection.java	1KB	JAVA File
IGpPyeErnDedAsgnCollPyeOvrd.java	4KB	JAVA File
IGpPyeErnDedAsgnCollPyeOvrdCollection.java	2KB	JAVA File
IGpPyeErnDedAsgnCollPyeOvrdCollPyeOvrSovr.java	2KB	JAVA File
IGpPyeErnDedAsgnCollPyeOvrdCollPyeOvrSovrCollection.ja...	2KB	JAVA File

There are two Java programs for every API file that you selected when you built the Java programs, as described in "Building the PeopleSoft API Java Programs" on page 2-12.

Note: Before you can compile the Java programs, you need the PeopleSoft Java Object Adapter, psjoa.jar located on your PeopleSoft Application Server in the PS_HOME\Web\psjoa directory.

To compile the PeopleSoft API Java programs:

1. If you have not already done so, copy psjoa.jar from the PeopleSoft Application Server to your local machine, and ensure that it is in the Java class path when you compile.

2. Compile the Java programs.

   When you compile, you must remember to respect the \PeopleSoft\generated\CompIntfc pathing.

   The following sample compile code would be placed in the psoft8_components directory; the code assumes that psjoa.jar is also placed in psoft8_components.

   ```
   @echo off
   set JAVA_HOME=<my-java-home>
   set PATH=%JAVA_HOME%\bin;%PATH%
   set CLASSPATH=%JAVA_HOME%\lib\tools.jar;psjoa.jar;%CLASSPATH%
   javac -classpath %CLASSPATH% .\PeopleSoft\Generated\CompIntfc\*.java
   ```

   This code places the class files back in the same directory with the Java files, but you may choose a different location depending on your site requirements.

3. Regardless of where you place the class file, the CLASSPATH variable in the startWebLogic.cmd file must be set to point to the directory directly above \PeopleSoft\generated\CompIntfc. For example, if you were to keep the class files in the psoft8_components directory, then psoft8_components would be in the class path.

   Alternatively, you may compress the class files into a jar file and place the jar file in your class path.

   The following code, if placed in the psoft8_components directory, will create a jar file:

   ```
   @echo off
   set JAVA_HOME=my-java-home
   set PATH=%JAVA_HOME%\bin;%PATH%
   set CLASSPATH=%JAVA_HOME%\lib\tools.jar;%CLASSPATH%
   jar cvf my-jar-file.jar  .\PeopleSoft\Generated\CompIntfc\*.class
   ```

Using PeopleSoft 8 Application Messaging

When you create events, you use PeopleSoft message channels to pass PeopleSoft XML to the WebLogic Server. The Adapter for PeopleSoft 8 uses a PeopleSoft 8 handler that you configure within the PeopleSoft application gateway using TCP/IP transport services. This section describes how to use and create of PeopleSoft message channels.

It includes the following topics:

- About PeopleSoft Application Messaging
- Configuring PeopleSoft for Application Messaging
- Setting up a Message Node for PeopleSoft Version 8.4 or Higher

Note: To learn how to configure WebLogic Server for the handler, see the *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide*.

About PeopleSoft Application Messaging

The Adapter for PeopleSoft 8 supports the integration of PeopleSoft applications and third party systems by publishing business events as XML-formatted messages. To subscribe to data, the Adapter for PeopleSoft 8 can accept and process the XML messages that PeopleSoft 8 posts via a custom Java subscription handler (or plug-in) to the PeopleSoft Application Messaging Gateway Servlet.

About PeopleSoft Handlers

PeopleSoft uses handlers to integrate PeopleSoft XML with the outside world. PeopleSoft provides a set of common interfaces that write PeopleSoft XML as a result of a business event, such as an addition to a table or a change to some piece of data. Two common PeopleSoft handlers are the SimpleFileHandler and the MQSeriesHandler. Depending on the handler specified in the PeopleSoft Message Channel definition, PeopleSoft posts an XML document to a file or to an MQSeries Queue. WebLogic Integration can handle events associated with files and MQSeries queues.

Adapter for PeopleSoft 8 supports events using two types of handlers:

- TCP/IP communications
- HTTP communications

About the BEA TCP/IP Handler

The Adapter for PeopleSoft 8 enables the posting of PeopleSoft XML directly to the WebLogic Integration environment by using a special handler. This handler is a plug-in class file that is installed within the PeopleSoft Gateway Web application server. This handler is the BEA PeopleSoft TCP/IP handler. WebLogic Integration can receive PeopleSoft event XML from the Application Messaging Gateway servlet through TCP.

To learn more about the installation and configuration of the handler software, see *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide*.

Configuring PeopleSoft for Application Messaging

The PeopleSoft XML output is triggered within PeopleSoft as a result of a business event. The PeopleSoft environment must be configured correctly for application messaging.

Note: This internal PeopleSoft configuration is a task for a person with knowledge of PeopleSoft. The installation of the software and the BEA TCP/IP handler is described in the *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide*.

The WebLogic Integration environment can receive a PeopleSoft XML document as an event using standard PeopleSoft handlers such as file or MQSeries, or it can use the specialized BEA TCP/IP handler.

Regardless of the handler you use, you must:

- Establish a Message Node to publish to the appropriate handler. See Setting up a Message Node for PeopleSoft 8.1 or Setting up a Message Node for PeopleSoft Version 8.4 or Higher

- Select or create a message channel with routing rules that point to the handler. See Selecting a Message Channel or your PeopleBooks documentation for information about creating a message channel.

- Select or create a message that defines the business event (such as an Add or Update transaction). See Creating a Message or Configuring an Existing Message.

- Test the message. See Setting the Message Channel.

For information about installing the BEA TCP/IP Handler and configuring the handler on the PeopleSoft Gateway Server, see the *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide.*

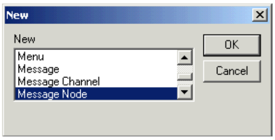
Setting up a Message Node for PeopleSoft 8.1

A message node publishes messages to the handler you specify, in this case, the BEA TCP/IP handler. When you create a message node, it's a good idea to work in a new PeopleSoft project.

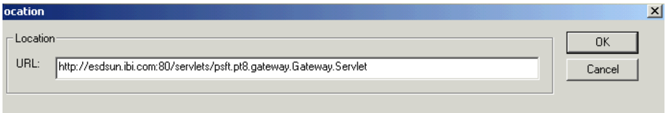
To set up a message node for PeopleSoft 8.1:

1. Choose File → New.

2. In the New dialog box appears, select MessageNode and click OK.



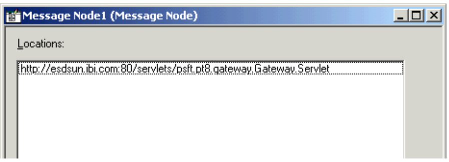
The Location dialog box appears.



URL: http://esdsun.ibi.com:80/servlets/psft.pt8.gateway.Gateway.Servlet

3. Enter the URL of the PeopleSoft Application Gateway (handler directory) and click OK.

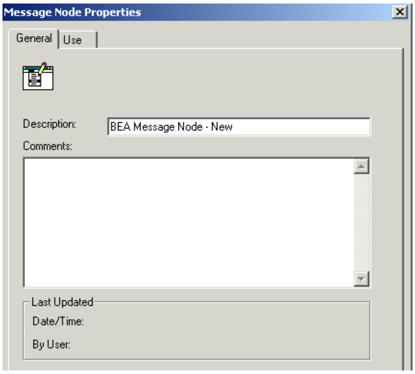
   The Message Node Window opens.



Locations:
http://esdsun.ibi.com:80/servlets/psft.pt8.gateway.Gateway.Servlet

4. Choose File → Object Properties.

   The Message Node Properties dialog box opens.



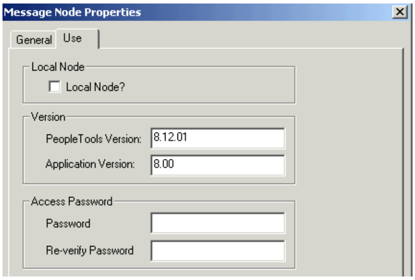
Description: BEA Message Node - New

Comments:

Date/Time:

By User:

5. Click the Use tab.



□ Local Node?

PeopleTools Version: 8.12.01

Application Version: 8.00

Password

Re-verify Password

6. Enter the PeopleTools Version, for example, 8.12.01, and click OK.

7. Save the message node and provide a name, such as BEA_NODE.

Note: It's a good idea to insert the message node into your project.

Selecting a Message Channel

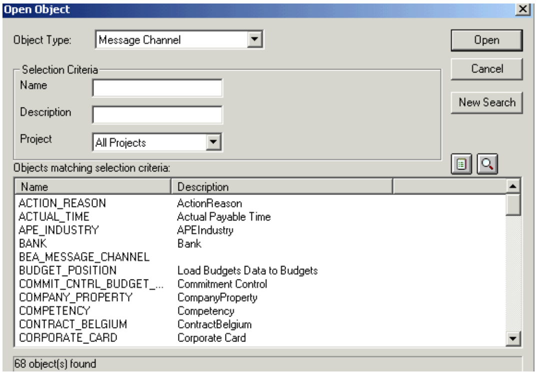
After you create a message node, the next step is to establish a message channel by selecting an existing message channel provided by PeopleSoft or creating a new one. This example uses the ENTERPRISE_SETUP message channel.

To learn how to create a new message channel, see your PeopleBooks documentation.

To select a message channel:

1. Choose File → Open.

   The Open Object dialog box appears.



Object Type: Message Channel

Selection Criteria

Name:

Description:

Project: All Projects

Objects matching selection criteria:

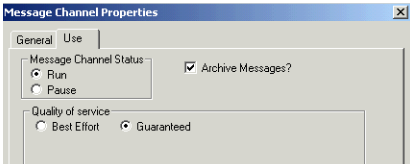
Name	Description
ACTION_REASON	ActionReason
ACTUAL_TIME	Actual Payable Time
APE_INDUSTRY	APEIndustry
BANK	Bank
BEA_MESSAGE_CHANNEL	
BUDGET_POSITION	Load Budgets Data to Budgets
COMMIT_CNTRL_BUDGET_...	Commitment Control
COMPANY_PROPERTY	CompanyProperty
COMPETENCY	Competency
CONTRACT_BELGIUM	ContractBelgium
CORPORATE_CARD	Corporate Card

68 object(s) found

2. Select Message Channel as the Object Type. For example, select ENTERPRISE_SETUP in the Objects list and click Open.

   The message channel, ENTERPRISE_SETUP, opens with the Messages tab active.

3. Open the Message Channel Properties dialog box.



Message Channel Status

● Run

○ Pause

☑ Archive Messages?

Quality of service

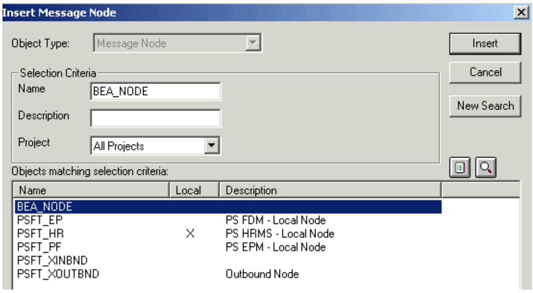
○ Best Effort ● Guaranteed

Note: Message archiving and quality of service are site dependent.

4. Click the Use tab, set Message Channel Status to Run, and click OK.

5. Select the Routing Rules tab (not illustrated) and select Insert Message Node.

The Insert Message Node dialog box appears.



Object Type: Message Node

Name: BEA_NODE

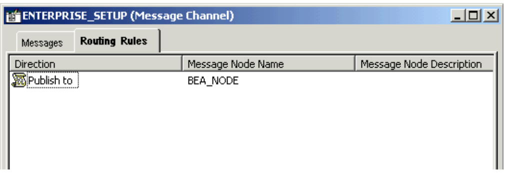
Description:

Project: All Projects

Name	Local	Description
BEA_NODE		
PSFT_EP		PS FDM - Local Node
PSFT_HR	X	PS HRMS - Local Node
PSFT_PF		PS EPM - Local Node
PSFT_XINBND		
PSFT_XOUTBND		Outbound Node

6. Select the message node, BEA_NODE you created previously, and click Insert.

   The ENTERPRISE_SETUP window opens.



Direction	Message Node Name	Message Node Description
Publish to	BEA_NODE	

7. Right-click BEA_NODE and choose Routing Direction → Publish To.

8. Save the message channel.

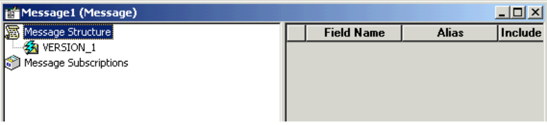
Note: It’s a good idea to insert the message channel into the project.

Creating a Message

You create a new message in the PeopleSoft Application Designer.

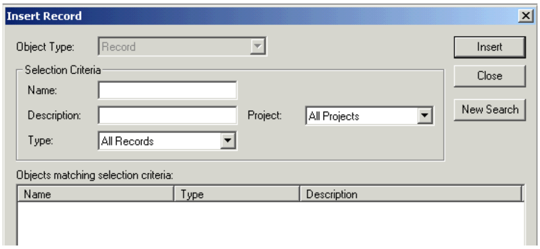
To create a new message

1. Choose File → New → Message.



2. In the Message window, right-click the version number and select Insert Child Record from the shortcut menu.

   The Insert Record dialog box opens.



Object Type: Record

Name:

Description:

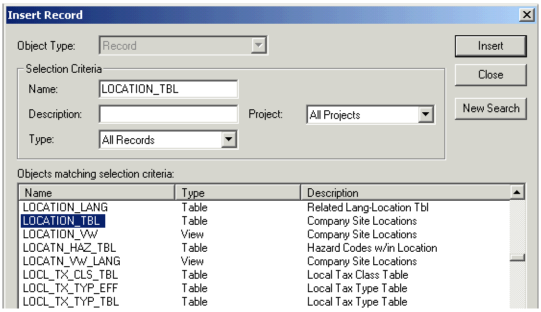
Project: All Projects

Type: All Records

Objects matching selection criteria:

Name	Type	Description

3. Click Insert to obtain a list of records or type the name of a records. For example, click LOCATION_TBL.



Object Type: Record

Name: LOCATION_TBL

Description:

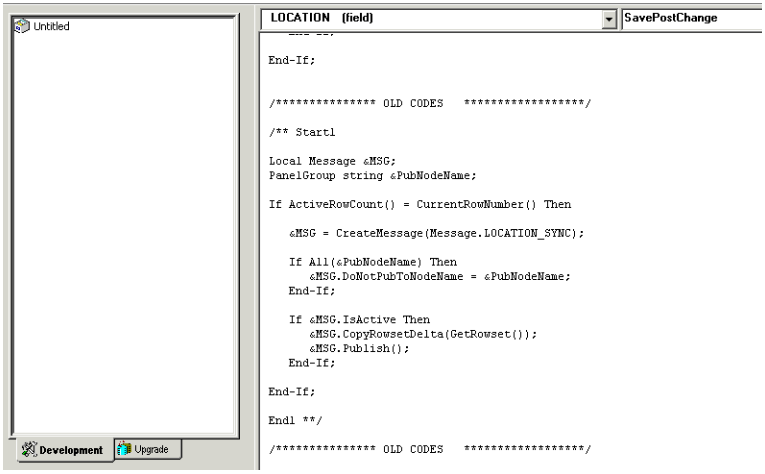
Project: All Projects

Type: All Records

Objects matching selection criteria:

Name	Type	Description
LOCATION_LANG	Table	Related Lang-Location Tbl
LOCATION_TBL	Table	Company Site Locations
LOCATION_VW	View	Company Site Locations
LOCATN_HAZ_TBL	Table	Hazard Codes w/in Location
LOCATN_VW_LANG	View	Company Site Locations
LOCL_TX_CLS_TBL	Table	Local Tax Class Table
LOCL_TX_TYP_EFF	Table	Local Tax Type Table
LOCL_TX_TYP_TBL	Table	Local Tax Type Table

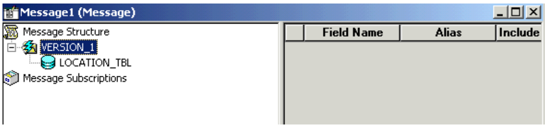
The SavePostChange PeopleCode of the LOCATION_TBL record displays the following code in the right pane. This is what triggers the message being published. You must add similar code to any tables that publishes data online.



```
End-If;


/*************** OLD CODES   ******************/

/** Start1

Local Message &MSG;
PanelGroup string &PubNodeName;

If ActiveRowCount() = CurrentRowNumber() Then

   &MSG = CreateMessage(Message.LOCATION_SYNC);

   If All(&PubNodeName) Then
      &MSG.DoNotPubToNodeName = &PubNodeName;
   End-If;

   If &MSG.IsActive Then
      &MSG.CopyRowsetDelta(GetRowset());
      &MSG.Publish();
   End-If;

End-If;

End1 **/

/*************** OLD CODES   ******************/
```

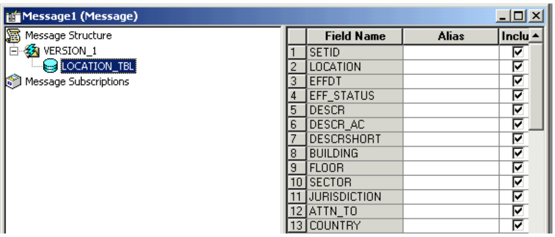
4. Click Insert in the Insert Record dialog box.

The Message window opens.



	Field Name	Alias	Include

5. Double-click the table name to display the table fields.

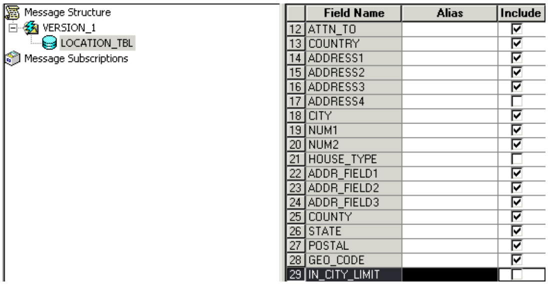


	Field Name	Alias	Inclu
1	SETID		☑
2	LOCATION		☑
3	EFFDT		☑
4	EFF_STATUS		☑
5	DESCR		☑
6	DESCR_AC		☑
7	DESCRSHORT		☑
8	BUILDING		☑
9	FLOOR		☑
10	SECTOR		☑
11	JURISDICTION		☑
12	ATTN_TO		☑
13	COUNTRY		☑

You may not require every field for the message.

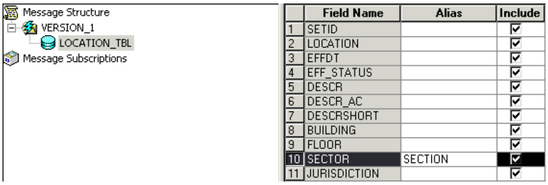
6. Click the appropriate check box in the Include column to select or remove fields in the table.

   In the example below, Address 4, House Type, and In City Limit fields have been excluded from the message.



	Field Name	Alias	Include
12	ATTN_TO		☑
13	COUNTRY		☑
14	ADDRESS1		☑
15	ADDRESS2		☑
16	ADDRESS3		☑
17	ADDRESS4		□
18	CITY		☑
19	NUM1		☑
20	NUM2		☑
21	HOUSE_TYPE		□
22	ADDR_FIELD1		☑
23	ADDR_FIELD2		☑
24	ADDR_FIELD3		☑
25	COUNTY		☑
26	STATE		☑
27	POSTAL		☑
28	GEO_CODE		☑
29	IN_CITY_LIMIT		□

7. To change a field's output name, enter the new name in the Alias field. For example, in the following window, the Sector field has been renamed to Section.



	Field Name	Alias	Include
1	SETID		☑
2	LOCATION		☑
3	EFFDT		☑
4	EFF_STATUS		☑
5	DESCR		☑
6	DESCR_AC		☑
7	DESCRSHORT		☑
8	BUILDING		☑
9	FLOOR		☑
10	SECTOR	SECTION	☑
11	JURISDICTION		☑

8. To create a new version of the message, right-click the version and choose Insert Version from the shortcut menu.

Setting the Message Channel

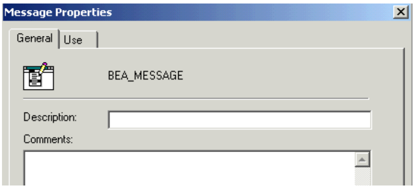
After you are satisfied that your message is set up correctly, it's a good idea to check that it is going to the right channel. To do so, you must set the Message Channel.

To set the message channel:

1. Open the Message Properties dialog box.

   For example, you can right-click the middle pane and select Message Properties from the shortcut menu.

   The Message Properties dialog box opens.



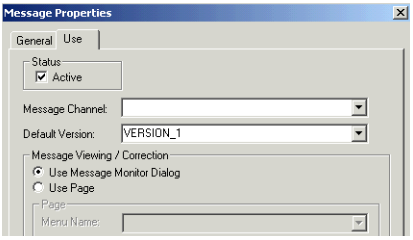
BEA_MESSAGE

Description:

Comments:

2. Enter a description and comments.

3. Click the Use tab.



Status

☑ Active

Message Channel:

Default Version: VERSION_1

Message Viewing / Correction

● Use Message Monitor Dialog

○ Use Page

Page

Menu Name:

4. Set Status to Active.

5. Select a message channel in the Message Channel list or enter a Message Channel name.

6. Click the Use Message Monitor Dialog button to display messages in the Application Message Monitor.

7. Enter the message details and click OK.

8. Choose File → Save As, enter a new file name, and click OK.

   You can now select the new message channel or a previously created one.

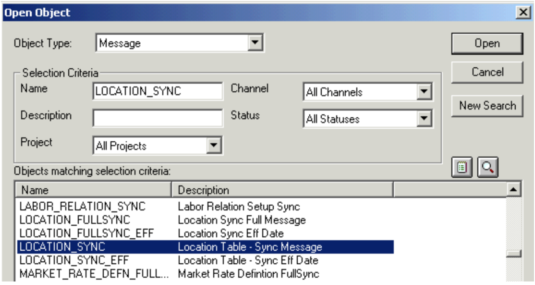
Configuring an Existing Message

Rather than creating a new message, you can use an existing message provided by PeopleSoft. For more information, consult your PeopleSoft administrator.

To configure an existing message:

1. Choose File → Open → Message.

   The Open Object dialog box appears.



Object Type: Message

Selection Criteria

Name: LOCATION_SYNC

Channel: All Channels

Description:

Status: All Statuses

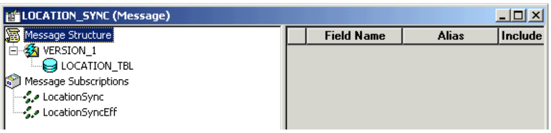
Project: All Projects

Objects matching selection criteria:

Name	Description
LABOR_RELATION_SYNC	Labor Relation Setup Sync
LOCATION_FULLSYNC	Location Sync Full Message
LOCATION_FULLSYNC_EFF	Location Sync Eff Date
LOCATION_SYNC	Location Table - Sync Message
LOCATION_SYNC_EFF	Location Table - Sync Eff Date
MARKET_RATE_DEFN_FULL...	Market Rate Defintion FullSync

2. Double-click the message you want to use. For example, LOCATION_SYNC.

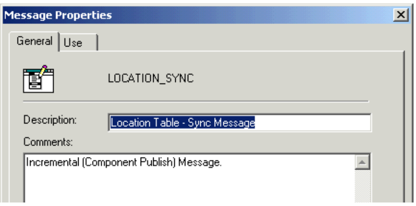
   The message window opens. In this case, the LOCATION_SYNC window opens.



- Message Structure
  - VERSION_1
    - LOCATION_TBL
- Message Subscriptions
  - LocationSync
  - LocationSyncEff

Field Name	Alias	Include

3. Display the Message Properties dialog box.



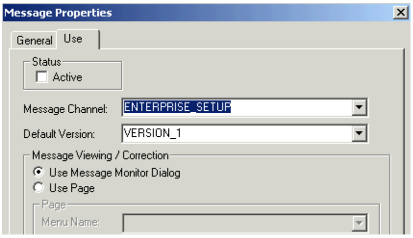
LOCATION_SYNC

Description: Location Table - Sync Message

Comments:

Incremental (Component Publish) Message.

4. Click the Use tab in the Message Properties dialog box.



Status

☐ Active

Message Channel: ENTERPRISE_SETUP

Default Version: VERSION_1

Message Viewing / Correction

● Use Message Monitor Dialog

○ Use Page

Page

Menu Name:

5. Set Status to Active.

6. Make sure correct message channel appears in the Message Channel list, for example, ENTERPRISE_SETUP and click OK.

7. Save the message.

Note: It's a good idea to insert the message into your project. See your PeopleBooks documentation for more information.

You are now ready to test the PeopleSoft event.

You can test the Message Channel using the PeopleSoft SimpleFileHandler. Testing outside of the WebLogic Server facilitates problem identification. As illustrated in the previous example, if you modify a location within PeopleSoft, an XML document representing the event is sent to a file which indicates that the message channel has been correctly configured.

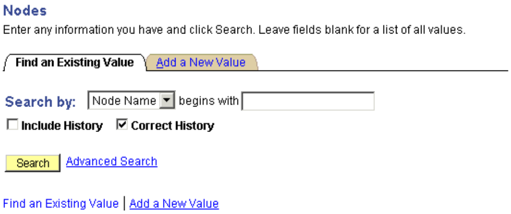
Setting up a Message Node for PeopleSoft Version 8.4 or Higher

A message node publishes messages to the handler you specify, in this case, the BEA TCP/IP handler. When you create a message node, it's a good idea to work in a new PeopleSoft project.

To create a new node in PeopleSoft version 8.4 or higher:

1. Click PeopleTools → Integration Broker → Node Definitions.

   The Node Definitions page appears.



Nodes

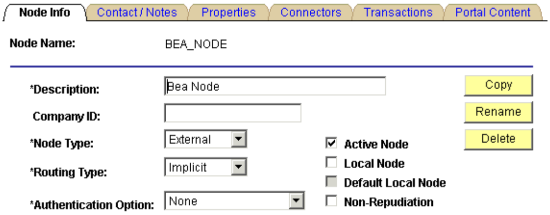
Enter any information you have and click Search. Leave fields blank for a list of all values.

Search by: Node Name begins with

☐ Include History ☑ Correct History

2. Click the Add a New Value tab.

   The Node Info tab appears.



Node Name: BEA_NODE

*Description: Bea Node

Company ID:

*Node Type: External

*Routing Type: Implicit

*Authentication Option: None

☑ Active Node

□ Local Node

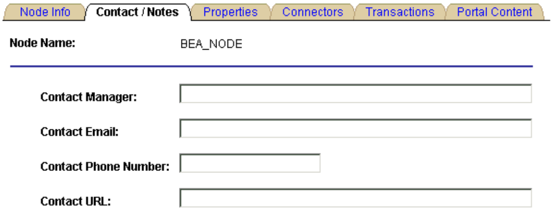
□ Default Local Node

□ Non-Repudiation

   a. Enter a description.

   b. Set Node Type to External and Routing Type to Implicit.

3. If you wish to enter contact information for a remote node, click the Contact / Notes tab and enter contact information.



Node Name: BEA_NODE

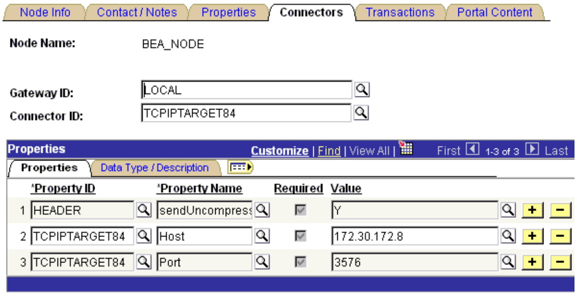
Contact Manager:

Contact Email:

Contact Phone Number:

Contact URL:

4. Click the Connectors tab.



Node Name: BEA_NODE

Gateway ID: LOCAL

Connector ID: TCPIPTARGET84

	*Property ID	*Property Name	Required	Value
1	HEADER	sendUncompress	☑	Y
2	TCPIPTARGET84	Host	☑	172.30.172.8
3	TCPIPTARGET84	Port	☑	3576

   The connectors listed are registered with the selected gateway. This field specifies the target connector appropriate to the communication method preferred by the current node.

5. Select the local Gateway ID.

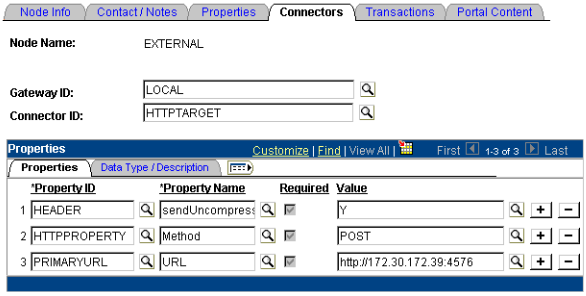
6. For TCP/IP, do the following:

   a. Select TCPIPTARGET84 as the Connection ID.

b. Enter the host name or IP address of the WebLogic Integration Server in the Property ID field of the host property name.

c. Enter the TCP port for the application view.

7. For HTTP, do the following:



Node Name: EXTERNAL

Gateway ID: LOCAL

Connector ID: HTTPTARGET

Properties

	*Property ID	*Property Name	Required	Value
1	HEADER	sendUncompress	☑	Y
2	HTTPPROPERTY	Method	☑	POST
3	PRIMARYURL	URL	☑	http://172.30.172.39:4576

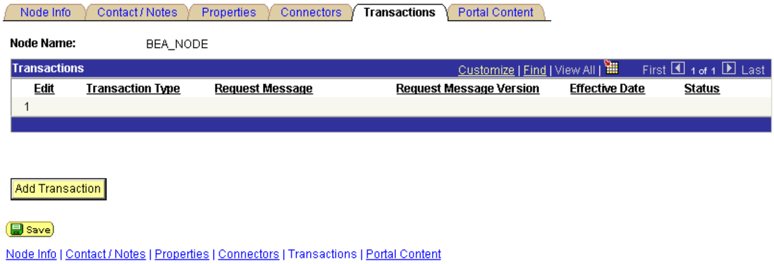
a. Select HTTPTARGET as the Connection ID.

b. Change the HEADER/sendUncompressed parameter to Y.

c. Change the HTTPPROPERTY/Method to POST.

d. Set the PRIMARYURL/URL to the URL and port of your WebLogic Integration Server. The port is the port entered when the application view was created.

8. Click the Transactions tab and click Save.


Node Info
Contact / Notes
Properties
Connectors
Transactions
Portal Content
Customize | Find | View All |
First 1 of 1 Last
Add Transaction
Save
Node Info | Contact / Notes | Properties | Connectors | Transactions | Portal Content

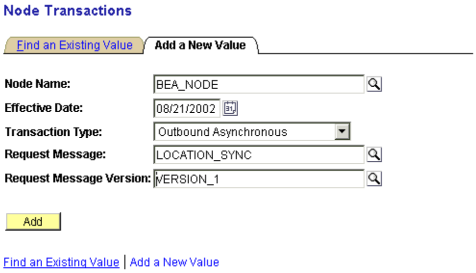
Node Name: BEA_NODE

Transactions

	Edit	Transaction Type	Request Message	Request Message Version	Effective Date	Status
1						

9. Click Add Transaction.

The Node Transactions page appears.



Node Transactions

Find an Existing Value | Add a New Value

Node Name:	BEA_NODE
Effective Date:	08/21/2002
Transaction Type:	Outbound Asynchronous
Request Message:	LOCATION_SYNC
Request Message Version:	VERSION_1

Add

Find an Existing Value | Add a New Value

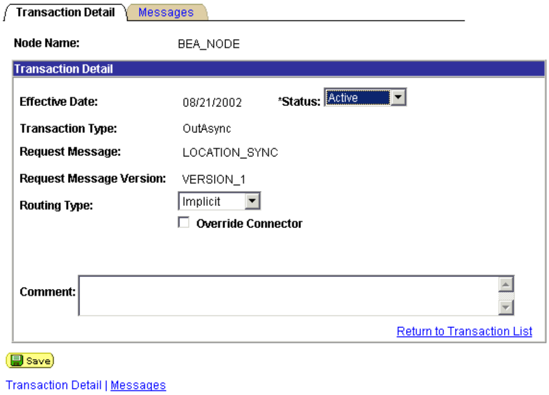
a. Select a transaction type from the drop-down list.

Note: Currently, only the Outbound Asynchronous transaction type is supported.

b. Enter a value for the Request Message field and a value for the Request Message Version field.

c. Click Add.

The Transaction Detail tab appears.



Transaction Detail | Messages

Node Name: BEA_NODE

Transaction Detail

Effective Date:	08/21/2002	*Status:	Active
Transaction Type:	OutAsync		
Request Message:	LOCATION_SYNC		
Request Message Version:	VERSION_1		
Routing Type:	Implicit		

□ Override Connector

Comment:

Return to Transaction List

Save

Transaction Detail | Messages

10. Set Routing Type to Implicit.

11. Repeat the add transaction process for every message you are using. Click Save.

12. Return to the Node Info tab and click Save.

Generating Schemas for PeopleSoft 8 Component Interfaces

The Adapter for PeopleSoft 8 uses XML documents to communicate with your PeopleSoft system's component interfaces and Java API for services, and messages for events. The format of these XML documents is determined by the component interfaces in your PeopleSoft system. After you create component interfaces and generate their Java APIs, you can use them as input to the BEA Application Explorer to generate the schemas that the adapter will use.

This section explains how to use the BEA Application Explorer to generate schemas. It contains the following topics:

- Before You Begin
- About the BEA Application Explorer
- Starting the BEA Application Explorer
- Setting the Session Path
- Managing PeopleSoft Connections
- Managing Schemas

Before You Begin

Before you begin to generate schema for the Adapter for PeopleSoft 8, you must:

- Download and install the BEA Application Explorer software. To learn more, see the *BEA Application Explorer Installation and Configuration Guide* at the following URL:

  `http://edocs.bea.com/wladapters/docs81/index.html`

- Obtain the information necessary to connect to your PeopleSoft 8 system. Contact your PeopleSoft 8 administrator for this information.

- For services, create component interfaces and generate the component interface APIs. To learn more, see "Using PeopleSoft 8 Component Interfaces" on page 2-1.

- For events, create messages and generate the message APIs. To learn more, see "Using PeopleSoft 8 Component Interfaces" on page 2-1.

About the BEA Application Explorer

The BEA Application Explorer uses intelligence about PeopleSoft 8 combined with metadata provided by PeopleSoft component interfaces and messages to generate the schemas required to build application view services and events.

This section contains the following topics:

- About the Process for Defining Schemas

- Types of Schemas Generated by the BEA Application Explorer

About the Process for Defining Schemas

The process for defining XML schemas includes the following steps:

1. Starting the BEA Application Explorer.

2. Setting the Session Path.

   The BEA Application Explorer uses this path to create the directory for the schemas.

3. Creating a New Connection or Using an Existing Connection.

4. Creating Schemas for Services and Creating Schemas for Events.

Types of Schemas Generated by the BEA Application Explorer

Each service or event the Adapter for PeopleSoft 8 uses must be defined by a schema. The BEA Application Explorer generates XML schemas for:

- Service Requests
- Service Responses
- Events

Service Requests

Service requests are requests for action that your application makes to your PeopleSoft 8 system. Requests are defined by request schema. As part of the definition, the request schema defines the input parameters required by the PeopleSoft 8 system. The PeopleSoft 8 system responds to the request with a service response.

Service Responses

Service responses are the way the PeopleSoft 8 system responds to a service request. A service response schema defines this service response. Service requests always have corresponding responses.

Events

Events are generated by the PeopleSoft 8 system as a result of activity on that system. You can use these events to trigger an action in your application. For example, the PeopleSoft 8 system may generate an event when customer information is updated. If your application must do something when this happens, your application is a consumer of this event. Events are defined by event schema.

Starting the BEA Application Explorer

You use the BEA Application Explorer to generate service request schemas, service response schemas, and event schemas. The schemas you create are published in the WebLogic Integration repository. Before you can create these schemas, however, you must create PeopleSoft component interfaces. If you have not already done so, see your PeopleBooks for instructions.

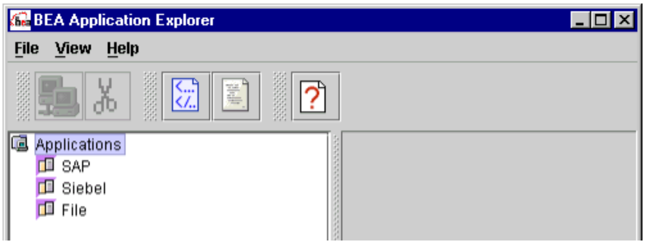
You must supply the BEA Application Explorer with the location of the component interface API classes for the component interfaces you are using.

To start the BEA Application Explorer:

1. Open the BEA Application Explorer.

   - In Windows, choose Windows Start→Programs→BEA Application Explorer.
   - On UNIX, run the startup script `beabse.sh` or the Java command `java com.ibi.common.ui.StartPanel`.

   The BEA Application Explorer window appears.



Setting the Session Path

The session path determines the directory where the BEA Application Explorer places your generated XML schemas and connection information. Your schemas are stored here:

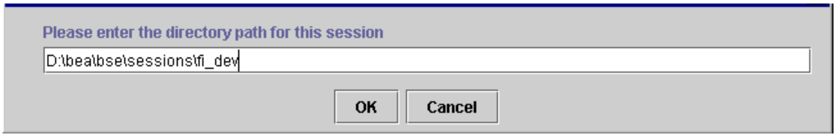
- On Windows: `*session_path*\peoplesoft\*connection_name*\schemas`
- On UNIX: `*session_path*/peoplesoft/*connection_name*/schemas`

Here, *`connection_name`* is the value you specify when you select a connection. To learn more about selecting a connection, see "Managing PeopleSoft Connections."

To set the session path:

1. From the File menu, choose Session.

   The Enter Session Path window appears, displaying a default path.



Please enter the directory path for this session

D:\bea\bse\sessions\fi_dev

2. Do one of the following:

   - To accept the default session path, click OK.
   - To specify a different path, enter the path and click OK.

     Specifying a different path allows you to group your schema according to project, or other logical group.

Managing PeopleSoft Connections

The BEA Application Explorer must connect to your PeopleSoft system before you can generate schemas. Therefore, you must first define a connect to your PeopleSoft system.

This section includes the following topics:

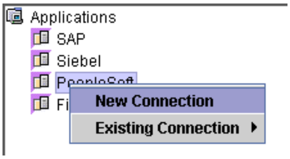
- Creating a New Connection
- Using an Existing Connection
- Disconnecting from PeopleSoft
- Removing Connections

Creating a New Connection

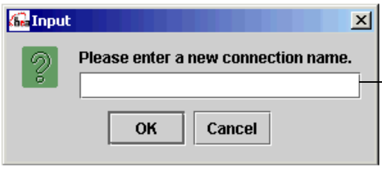
When you are create a new connection, you need to provide the name and port of the PeopleSoft server as well as your PeopleSoft user name and password.

To create a new connection:

1. In the left pane of the BEA Application Explorer window, under Applications right-click PeopleSoft 8 → New Connection.



The BEA Application Explorer prompts you for a connection name.

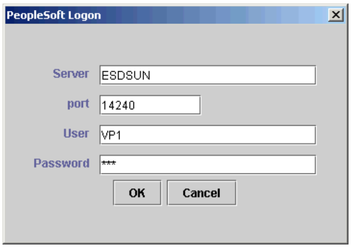


Please enter a new connection name.

Enter a name for this connection.

2. Enter a name for this connection and click OK.

The PeopleSoft Logon dialog box appears.



PeopleSoft Logon

Server ESDSUN

port 14240

User VP1

Password ***

3. Enter the information that allows PeopleSoft Client applications to connect to PeopleSoft.

   - PeopleSoft Application Server name
   - PeopleSoft Application Server Port
   - PeopleSoft User name
   - PeopleSoft Password

   For more information, see your PeopleSofts documentation or consult your PeopleSoft System Administrator.

4. Click OK.

   It may take several minutes to loading the internal cached file. However, this speeds the process for subsequently displaying and creating schemas for other component interfaces. Once connected, the BEA Application Explorer displays a list of component interfaces.

   The connection appears under the PeopleSoft node in the BEA Application Explorer window. You can now view component interfaces and messages in your PeopleSoft system.

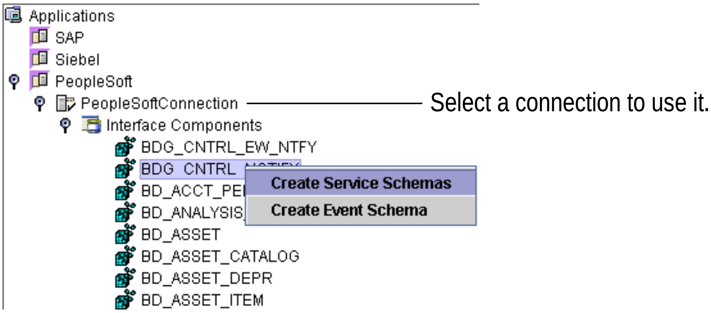
Using an Existing Connection

You can use an existing connection rather than creating a new one.

To use an existing PeopleSoft connection:

1. In the left pane of the BEA Application Explorer window, under Applications right-click PeopleSoft 8 → Existing Connection → *your connection*.

   The connection appears below the PeopleSoft 8 node.



Select a connection to use it.

2. If the connection parameters do not correspond to your system, edit them in the PeopleSoft Logon Window.

3. Click OK.

Disconnecting from PeopleSoft

The BEA Application Explorer allows you to disconnect from PeopleSoft.

To disconnect from PeopleSoft:

- In the left pane of the BEA Application Explorer, right-click on the connection. Choose Disconnect.

This disconnects from PeopleSoft, and the connection icon change to indicate that it not currently connected. To re-establish the connection, right-click on the connection and choose Connect.

Removing Connections

The BEA Application Explorer allows you to remove connections when you no longer need them.

To remove a connection:

- In the left pane of the BEA Application Explorer, right-click on the connection. Choose Remove.

Managing Schemas

You need to create a schema for each service and event your application uses. You use the BEA Application Explorer to create these schemas.

This section explains:

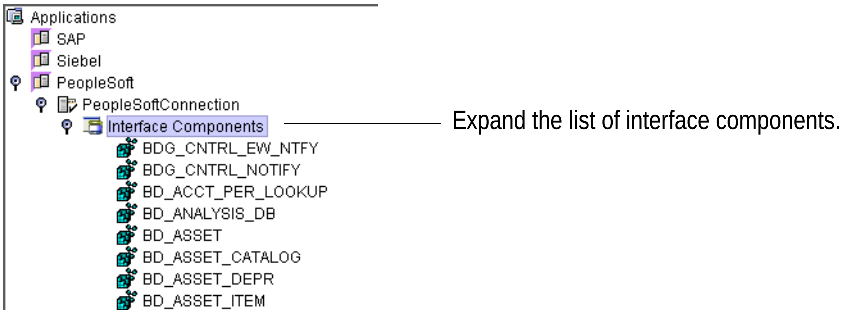
- Creating Schemas for Services
- Creating Schemas for Events
- Removing Schemas

Creating Schemas for Services

Services require two schemas, one for the request and one for the response. Services always have these two schema, even is the response is not used by your application.

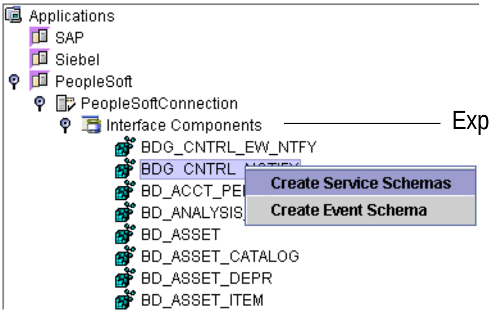
To create a schema for a service:

1. Start BEA Application Explorer. To learn more, see "Starting the BEA Application Explorer" on page 4-3.
2. Set the session path. This determines where the BEA Application Explorer places your schemas. To learn more, see "Setting the Session Path" on page 4-4.
3. Select or create a connection to PeopleSoft. To learn more, see "Managing PeopleSoft Connections" on page 4-5.
4. Expand the tree under Applications → PeopleSoft → *connection name* → Component Interfaces to see the items for which you may create a schema. If you cannot expand the tree beneath PeopleSoft, you have not set a connection for PeopleSoft.



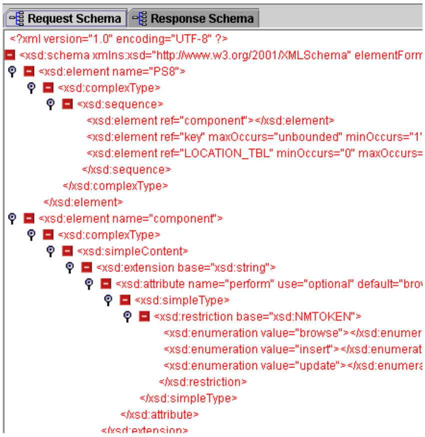
Expand the list of interface components.

5. Select a component and right-click to generate service request and response schemas.



Expand the list of interface components.

The BEA Application Explorer displays tabs that show the request and response schemas.



```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementForm
  <xsd:element name="PS8">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="component"></xsd:element>
        <xsd:element ref="key" maxOccurs="unbounded" minOccurs="1'
        <xsd:element ref="LOCATION_TBL" minOccurs="0" maxOccurs=
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="component">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:string">
          <xsd:attribute name="perform" use="optional" default="bro
            <xsd:simpleType>
              <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="browse"></xsd:enumer
                <xsd:enumeration value="insert"></xsd:enumerat
                <xsd:enumeration value="update"></xsd:enumera
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:attribute>
        </xsd:extension>
```

The BEA Application Explorer creates a directory structure within the working directory you identified earlier. In this example, the working directory is `C:\BEA\BEASCHEMAS`.

Within this directory, the BEA Application Explorer creates a folder called `Peoplesoft` as well as subfolders to hold the schemas for each configured PeopleSoft connection. In this example, the schemas have been created in the folder called `PeopleSoftConnection`, and the BEA Application Explorer adds the following items to the folder `C:\BEA\BEASCHEMAS\PeopleSoft\PeopleSoftConnection`:

– `manifest.xml`

– *`service_name`*`.xsd`

– *`service_name`*`_response.xsd`

You have successfully created service request and response schemas for this component interface.

Creating Schemas for Events

Events only require one schema. There is no response expected when PeopleSoft generates an event.

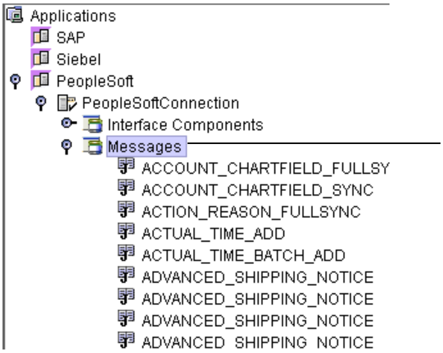
To create a schema for an event:

1. Start BEA Application Explorer. To learn more, see “Starting the BEA Application Explorer” on page 4-3.

2. Set the session path. This determines where the BEA Application Explorer places your schemas. To learn more, see “Setting the Session Path” on page 4-4.

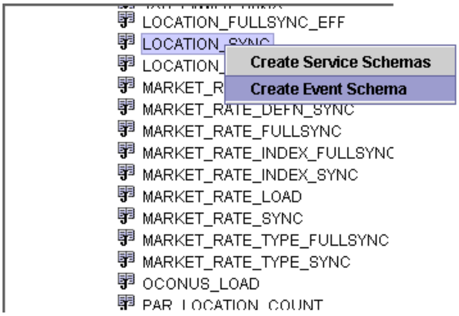
3. Select or create a connection to your EIS. To learn more, see “Managing PeopleSoft Connections” on page 4-5.

4. Expand the tree under Applications → PeopleSoft → *connection name* → Component Interfaces to see the items for which you may create a schema. If you cannot expand the tree beneath PeopleSoft, you have not set a connection for PeopleSoft.



Expand the list of messages.

5. Right-click a message and choose Create Event Schemas.



The BEA Application Explorer displays tabs that show the event schema.

If it has not already done so, the BEA Application Explorer creates a directory structure within the working directory you identified earlier. In this example, the working directory is `C:\BEA\BEASCHEMAS`.

Within this directory, the BEA Application Explorer creates a folder called `Peoplesoft` as well as subfolders to hold the schemas for each configured PeopleSoft connection. In this example, the schemas have been created in the folder called `PeopleSoftConnection`, and the BEA Application Explorer adds the following items to the folder `C:\BEA\BEASCHEMAS\PeopleSoft\PeopleSoftConnection`:

- `manifest.xml`

- *`event_name`*`.xsd`

You have successfully created an event schema for this message.

Removing Schemas

To remove a schema:

1. Right-click on a component interface or message for which there is at least one schema.

   If there is an event schema defined for this message, the menu has a Remove Event Schemas option.

   If there are service schemas defined for this component interface, the menu has a Remove Event Schema option.

2. Choose the appropriate option.

Next Steps

After you have defined schemas for your events and services, the next step is to create an application view. An application view makes the services and events available to applications. To learn more about application views, see "Defining Application Views for PeopleSoft 8" on page 5-1.

CHAPTER 5

Defining Application Views for PeopleSoft 8

An application view is a business-oriented interface to objects and operations within an EIS.

This section presents the following topics:

- How to Use This Document
- Before You Begin
- About Application Views
- About Defining Application Views
- Defining Service Connection Parameters
- Setting Service Properties
- Setting Event Properties
- Defining Event Connection Parameters
- Testing Services
- Testing Events Using a Service
- Testing Events Manually

How to Use This Document

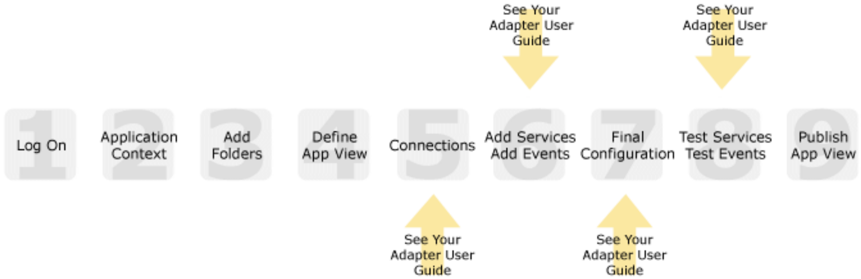
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your Siebel system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 5-1 Information Interlock with ***Using the Application Integration Design Console***


See Your Adapter User Guide
See Your Adapter User Guide
1 Log On
2 Application Context
3 Add Folders
4 Define App View
5 Connections
6 Add Services Add Events
7 Final Configuration
8 Test Services Test Events
9 Publish App View
See Your Adapter User Guide
See Your Adapter User Guide

Before You Begin

Before you define an application view, make sure you have:

- Installed and deployed the adapter according to the instructions in *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide*.
- Determined which business processes need to be supported by the application view. The required business processes determine the types of services and events you include in your application views. Therefore, you must gather information about the application's business requirements from the business analyst. Once you determine the necessary business

processes, you can define and test the appropriate services and events. For more information, see "Getting Started With the Adapter for PeopleSoft 8" on page 1-11.

- Gathered the connection information for your PeopleSoft 8 system. To learn more about the connection information needed by the BEA Application Explorer for your PeopleSoft 8 system, see "Before You Begin" on page 4-2.

About Application Views

An application view defines:

- Connection information for the EIS, including login information, connection settings, and so on.
- Service invocations, including the information the EIS requires for this request, as well as the request and response schemas associated with the service.
- Event notifications, including the information the EIS publishes and the event schema for inbound messages.

Typically, an application view is configured for a single business purpose and contains only the services and events required for that purpose. An EIS might have multiple application views, each defined for a different purpose.

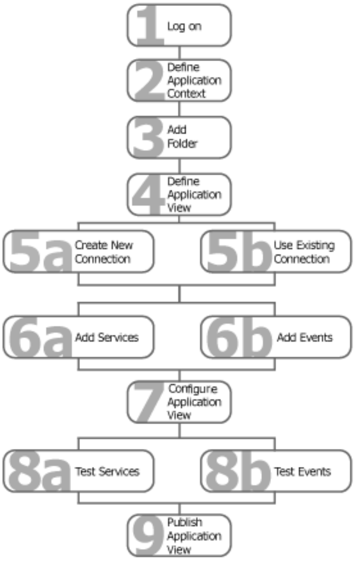
About Defining Application Views

Defining an application view is a multi-step process described in *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The information you enter depends on the requirements of your business process and your EIS system configuration. Figure 5-2 summarizes the procedure for defining and configuring an application view.

Figure 5-2 Process for Defining and Configuring an Application View


1 Log on
2 Define Application Context
3 Add Folder
4 Define Application View
5a Create New Connection
5b Use Existing Connection
6a Add Services
6b Add Events
7 Configure Application View
8a Test Services
8b Test Events
9 Publish Application View

To define an application view:

1. Log onto the WebLogic Integration Application View Console.
2. Define the application context by selecting an existing application or specifying a new application name and root directory.

   This application will be using the events and services you define in your application view. The application view works within the context of this application.

3. Add folders as required to help you organize application views.
4. Define a new application view for your adapter.
5. Add a new connection service or select an existing one.

   If you are adding a new connection service, see "Defining Service Connection Parameters" on page 5-5 for details about PeopleSoft 8 requirements.

6. Add the events and services for this application view.

   See the following sections for details about PeopleSoft 8 requirements:

   – "Setting Service Properties" on page 5-6

   - “Setting Event Properties” on page 5-8

7. Perform final configuration tasks.

   If you are adding an event connection, see “Defining Event Connection Parameters” on page 5-11 for details about PeopleSoft 8 requirements.

8. Test all services and events to make sure they can properly interact with the target PeopleSoft 8 system.

   See the following sections for details about PeopleSoft 8 requirements:

   - “Testing Services” on page 5-13
   - “Testing Events Using a Service” on page 5-14
   - “Testing Events Manually” on page 5-15

9. Publish the application view to the target WebLogic Workshop application.

   This is the application you specified in step 2. Publishing the application view allows workflow developers within the target application to interact with the newly published application view using an Application View control.

Defining Service Connection Parameters

This information applies to “Step 5A, Create a New Browsing Connection” in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

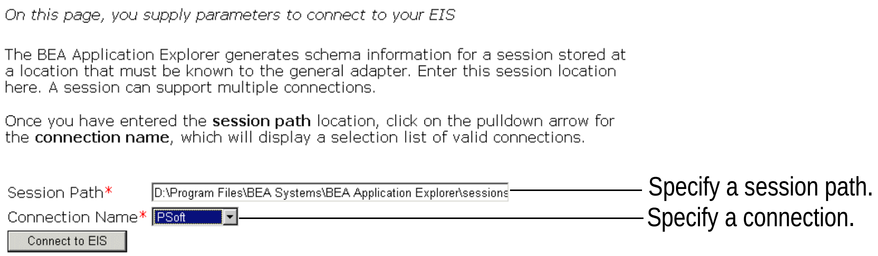
The Select Browsing Connection page allows you to choose the type of connection factory to associate with the application view. You can select a connection factory within an existing instance of the adapter or create a connection factory within a new adapter instance.

After you enter a connection name and description, you use the Configure Connection Parameters page to specify connection parameters for a connection factory.

To create a new browsing connection:

1. In the Create New Browsing Connections page, enter a connection name and description as described in *Using the Application Integration Design Console*.

   The Configure Connection Parameters page appears to allow you to configure the newly created connection factory within the new adapter instance.



On this page, you supply parameters to connect to your EIS

The BEA Application Explorer generates schema information for a session stored at a location that must be known to the general adapter. Enter this session location here. A session can support multiple connections.

Once you have entered the **session path** location, click on the pulldown arrow for the **connection name**, which will display a selection list of valid connections.

Session Path* D:\Program Files\BEA Systems\BEA Application Explorer\sessions

Connection Name* PSoft

Connect to EIS

Specify a session path.

Specify a connection.

Note: A red asterisk (*) indicates that a field is required.

2. Specify a session path and connection name.

   This information enables the application view to interact with the target PeopleSoft system. You need enter this information only once per application view. The session path is the location of with working directory you established earlier. The connection name is the name of the connection you used for creating schemas.

3. Click Connect to EIS.

   You return to the Create New Browsing Connections, where you can specify connection pool parameters and logging levels. For more information, see *Using the Application Integration Design Console* at the following URL:

   http://edocs.bea.com/wli/docs81/aiuser/index.html

Setting Service Properties

This information applies to “Step 6A, Add a Service to an Application View” in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Adapter for PeopleSoft 8 uses services to make requests of the PeopleSoft system. A service consists of both a request and a response.

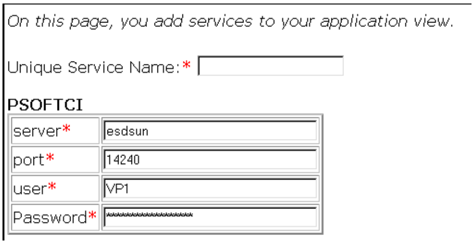
To set service properties:

1. Enter a unique service name that describes the function the service performs.

   Valid characters for the service name include a-z, A-Z, 0-9, and _ (underscore).

2. Select File from the Select list.

The Add Services page displays the fields required for a service.



On this page, you add services to your application view.

Unique Service Name:*

PSOFTCI

server*	esdsun
port*	14240
user*	VP1
Password*	****************

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 5-1 PeopleSoft Service Parameters

Parameter	Description
server	The name of the PeopleSoft server
port	The port on the PeopleSoft server
user	The PeopleSoft administrator user name
password	The password of the administrative user

4. See "Common Service and Event Settings" on page 5-7 for information about selecting a schema and configuring logging and tracing.

Common Service and Event Settings

1 2 3 4 5 6 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

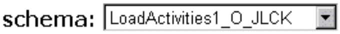
http://edocs.bea.com/wli/docs81/aiuser/index.html

You select a schema and select logging options the same way for all services.

To set common service settings:

1. In the Schema list, select the schema you want to use with this service.

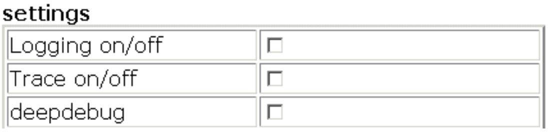
   For more information, see Chapter 4, "Generating Schemas for PeopleSoft 8 Component Interfaces."



schema: LoadActivities1_O_JLCK

2. Configure logging and tracing for this service, as follows:

   Logging captures information from your adapter and writes it in a log file. Tracing displays runtime information in the console. You set the type and amount of information you wish to capture as part of the final configuration tasks. This is described in detail in *Using the Application Integration Design Console*.

   settings

Logging on/off	□
Trace on/off	□
deepdebug	□

   a. Select the Logging on/off check box to enable logging for this service. Logging information is written to a log file (`BEA_FILE_1_0.log`) that appears in the directory from which the application was started.

   b. Select the Trace on/off check box to enable tracing for this service. Trace information appears in the runtime console.

   c. Select the deepdebug check box to enable additional trace information for deeper troubleshooting.

3. Click Add to add the service.

   For more information about the next step, see *Using the Application Integration Design Console* at the following URL:

   `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Setting Event Properties

This information applies to “Step 6B, Add an Event to an Application View” in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The Adapter for PeopleSoft 8 supports the following events:

- TCP/IP Event
- HTTP Event

TCP/IP Event

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

The TCP/IP event is one way PeopleSoft sends an XML file that represents the PeopleSoft event—via Application Messaging—to WebLogic Integration. The PeopleSoft XML document is passed to an event variable that is set in a business process.

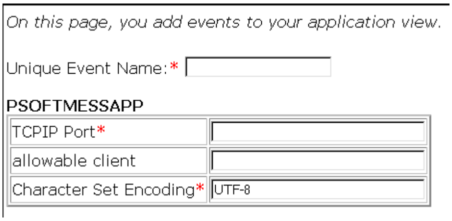
To set TCP/IP event properties:

1. Enter a unique event name that describes the function the event performs.

   Valid characters for the event name include a-z, A-Z, 0-9, and _ (underscore).

2. Select TCP/IP from the Select list.

   The Add Events page displays the fields required for this event type.

   On this page, you add events to your application view.

   Unique Event Name:*

   PSOFTMESSAPP

TCPIP Port*	
allowable client	
Character Set Encoding*	UTF-8

   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 5-2 PeopleSoft TCP/IP Event Parameters

Parameter	Description
TCP/IP Port	The name of an unused TCP/IP port of your choice to be used for listening for events that originate from PeopleSoft. You must use this port when you configure the PeopleSoft gateway. See *BEA WebLogic Adapter for PeopleSoft 8 Installation and Configuration Guide*.
Allowable client	The IP address of the PeopleSoft machine that is sending the event to the adapter. This optional field allows the adapter to restrict the IP address of the sending PeopleSoft system.

Table 5-2 PeopleSoft TCP/IP Event Parameters (Continued)

Parameter	Description
Character Set Encoding	The character set encoding for inbound documents. For example, UTF-8.

4. See "Common Service and Event Settings" on page 5-7 for information about selecting a schema and configuring logging and tracing.

HTTP Event

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The HTTP event waits for the arrival of a PeopleSoft XML message on the URL and specified port. This URL and port must correspond to the URL and port specified when you configured the HTTP Target Connector on the node in the Integration Broker. See "Setting up a Message Node for PeopleSoft Version 8.4 or Higher" on page 3-11. HTTP is available on the Integration Broker starting with PeopleSoft 8.4.

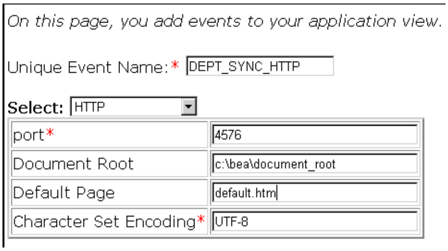
To set HTTP event properties:

1. Enter a unique event name that describes the function the event performs.

   Valid characters for the event name include a-z, A-Z, 0-9, and _ (underscore).

2. Select HTTP from the Select list.

   The Add Events page displays the fields required for this event type.



On this page, you add events to your application view.

Unique Event Name:* DEPT_SYNC_HTTP

Select: HTTP

port*	4576
Document Root	c:\bea\document_root
Default Page	default.htm
Character Set Encoding*	UTF-8

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 5-3 PeopleSoft HTTP Event Parameters

Parameter	Description
Port	The name of port specified in the primary URL supplied when the HTTP node is configured. See "Setting up a Message Node for PeopleSoft Version 8.4 or Higher" on page 3-11.
Document Root	Base directory from which all HTTP pages will be served.
Default Page	A page that must be present if no page is identified in the incoming HTTP(s) request. The name and type of the file are arbitrary.
Character Set Encoding	The character set encoding for inbound documents. For example, UTF-8.

4. See "Common Service and Event Settings" on page 5-7 for information about selecting a schema and configuring logging and tracing.

Defining Event Connection Parameters

This information applies to "Step 7, Perform Final Configuration Tasks" in *Using the Application Integration Design Console*, at the following URL:

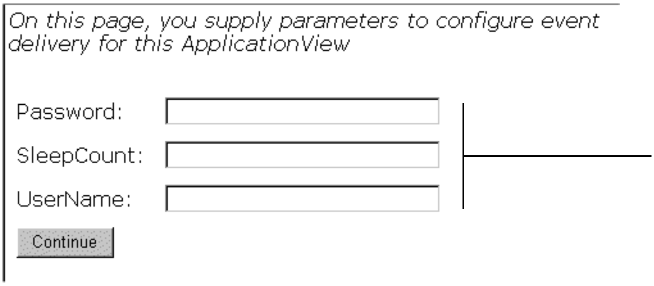
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Once you have finished adding services and events and have saved your application view, you must perform some final configuration tasks, including configuring event delivery connections, before testing the services and events. You perform these configuration tasks from the Final Configuration and Testing page.

To define event connection parameters:

1. In Connections area on the Application View Administration page, click Select/Edit.
2. In the Event Connection area, click Event to edit the default event connection.

   The Configure Event Delivery Parameters page appears.


On this page, you supply parameters to configure event delivery for this ApplicationView

Password:

SleepCount:

UserName:

Enter connection information for your system.

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 5-4 Event Connection Parameters

Parameter	Description
username	Your WebLogic Server Administration Console user name, defined in the `startWebLogic` script
password	The password for your WebLogic Server Administration Console user name
SleepCount	The number of seconds the adapter will wait between polling for events

The event delivery parameters you enter on this page enable connection to your PeopleSoft 8 system and are used when generating events. The parameters are specific to the associated adapter and are defined in the `wli-ra.xml` file within the base adapter.

4. Click Save to save your event delivery parameter settings. Click Continue to return to the Edit Event Adapter page, and then click Back to return to the Final Configuration and Testing page.

   The Edit Event Adapter page allows you to define event parameters and configure the information that will be logged for the connection factory. Select one of the following settings for the log:

   – Log errors and audit messages

   – Log warnings, errors, and audit messages

   – Log informational, warning, error, and audit messages

   – Log all messages

The table that follows describes the type of information that each logging message contains.

Table 5-5 Logging message categories

This type of message	Contains
Audit	Extremely important information related to the business processing performed by an adapter.
Error	Information about an error that has occurred in the adapter, which may affect system stability.
Warning	Information about a suspicious situation that has occurred. Although this is not an error, it could have an impact on adapter operation.
Information	Information about normal adapter operations.

Testing Services

1 2 3 4 5 6 7 **8** 9

This information applies to "Step 8A, Test an Application View's Services" in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

The purpose of testing an application view service is to evaluate whether that service interacts properly with the target PeopleSoft 8 system. When you test a service, you supply any inputs required to start the service. For the Adapter for PeopleSoft 8, the input is in the form of a valid XML string based on the schema you created for your PeopleSoft 8 application. You can create the XML file by running XML Spy, which is installed with WebLogic Integration, against your schema files.

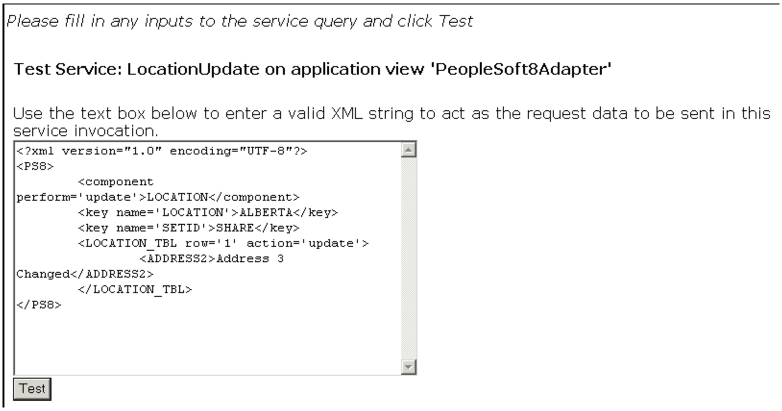
Note: Before you test an application view, it must contain at least one event or service. Also, you must place the application view in test mode. To place an application view in test mode, click the Test button at the bottom of the Application View Administration page.

To test a service:

1. In the Application View Administration page, click the Test link beside the service to be tested.

   The Test Services page appears.

2. In the Test Service window, copy the appropriate XML strings from the PeopleSoft XML file for your account.



Please fill in any inputs to the service query and click Test

Test Service: LocationUpdate on application view 'PeopleSoft8Adapter'

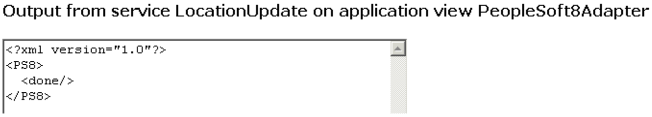
Use the text box below to enter a valid XML string to act as the request data to be sent in this service invocation.

```
<?xml version="1.0" encoding="UTF-8"?>
<PS8>
        <component
perform='update'>LOCATION</component>
        <key name='LOCATION'>ALBERTA</key>
        <key name='SETID'>SHARE</key>
        <LOCATION_TBL row='1' action='update'>
                <ADDRESS2>Address 3
Changed</ADDRESS2>
        </LOCATION_TBL>
</PS8>
```

In this case, the XML string is updating a row names Address 3 in the Alberta location.

3. Click Test.

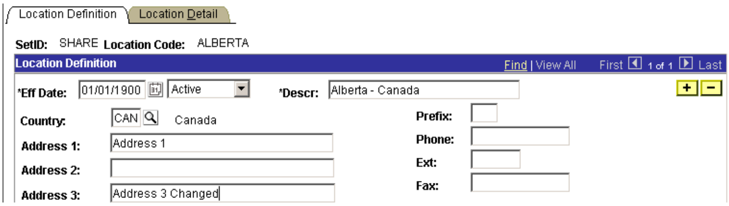
   The results appear in the Test Results window.



Output from service LocationUpdate on application view PeopleSoft8Adapter

```
<?xml version="1.0"?>
<PS8>
  <done/>
</PS8>
```

You can also test the results by checking in your PeopleSoft system.



SetID: SHARE Location Code: ALBERTA

Location Definition

*Eff Date:	01/01/1900 Active	*Descr:	Alberta - Canada
Country:	CAN Canada	Prefix:	
Address 1:	Address 1	Phone:	
Address 2:		Ext:	
Address 3:	Address 3 Changed	Fax:	

Testing Events Using a Service

This information applies to “Step 8B, Test an Application View’s Events” in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by PeopleSoft 8. When you test an event, you can trigger the event using a service or manually.

Note: Before you test an application view, it must contain at least one event or service. Also, you must place the application view in test mode. To place an application view in test mode, click the Test button at the bottom of the Application View Administration page.

To test an event:

1. In the Application View Administration page, click the Test link beside the service to be tested.

   The Test Events page appears.

2. Click Service and select a service that triggers the event you are testing.

3. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

4. Click Test and enter the XML string needed to trigger the service.

   The service is executed.

   – If the test succeeds, the Test Result page appears, showing the event document, the service input document, and the service output document.

   – If the test fails, the Test Result page displays only a Timed Out message.

Testing Events Manually

This information applies to “Step 8B, Test an Application View’s Events” in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Note: Before you test an application view, it must contain at least one event or service. Also, you must place the application view in test mode. To place an application view in test mode, click the Test button at the bottom of the Application View Administration page.

To test an event manually:

1. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

2. Click Test. The test waits for an event to trigger it.

3. Using the triggering PeopleSoft 8 application, perform an action that executes the service that, in turn, tests the application view event.

   – If the test succeeds, the Test Result page displays the event document from the application.

   – If the test fails or the event does not occur within the allotted time, the Test Result page displays a Timed Out message.

Index