Oracle® Application Server Integration

Adapter for PeopleSoft 8 User's Guide

10*g* (9.0.4)

Part No. B10298-01

September 2003

ORACLE®

Oracle Application Server Integration Adapter for PeopleSoft 8 User's Guide, 10*g* (9.0.4)

Part No. B10298-01

Primary Author: Krista Gervais

Contributing Authors: Seetha Rao, Clement Pellerin, Philip Bridger

Contributors: Arvind Jain, Bo Stern, Maneesh Joshi, Seshu Adunuthula

Contents

3 Defining an Interaction

4 Component Interface Interactions

Index

Send Us Your Comments

Oracle Application Server Integration Adapter for PeopleSoft 8 User's Guide, 10*g* (9.0.4)

Part No. B10298-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: 650-506-7407 Attn: Oracle Application Server Documentation Manager
- Postal service:
  Oracle Corporation
  Oracle Application Server Documentation
  500 Oracle Parkway, M/S 2op3
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This guide explains how to use Oracle Application Server ProcessConnect and the Oracle Application Server Integration Adapter for PeopleSoft 8 to access PeopleSoft Component Interfaces. In this guide you will learn how to define a delivery channel for PeopleSoft and add an interaction to generate native events, which are XML instances defined by XSD (XML payload defined by an XML Schema Definition instance). In this guide you will also find a chapter describing the datatype mapping between PeopleSoft and XSD.

> **See Also:** *Oracle Application Server ProcessConnect User's Guide* for more information on using native events with Oracle Application Server ProcessConnect business processes

This preface contains these topics:

Intended Audience

Oracle Application Server Integration Adapter for PeopleSoft 8 User's Guide is intended for anyone who performs the following tasks:

- Creates delivery channels and interactions with a PeopleSoft system
- Maintains applications

To use this document, you need some knowledge of PeopleSoft Component Interfaces.

Organization

This document contains:

Chapter 1, "Introduction to Oracle Application Server Integration Adapter for PeopleSoft 8"

This chapter describes the Oracle Application Server Integration Adapter for PeopleSoft 8 and the hardware and software requirements.

This chapter also provides instructions for adding a custom Component Interface to a PeopleSoft server.

Chapter 2, "Defining a Delivery Channel"

This chapter provides instructions for using Oracle Application Server ProcessConnect to define a delivery channel for a PeopleSoft system.

Chapter 3, "Defining an Interaction"

This chapter provides instructions for using Oracle Application Server ProcessConnect to add a PeopleSoft Component Interface interaction.

Chapter 4, "Component Interface Interactions"

This chapter provides information on the PeopleSoft Component Interface interactions used by the Oracle Application Server Integration Adapter for PeopleSoft 8.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Application Server ProcessConnect User's Guide* in the Oracle Application Server Documentation Library
- *Oracle Application Server Installation Guide*

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://otn.oracle.com/membership`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://otn.oracle.com/docs`

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.

Convention	Meaning	Example
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and interactions, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to True. Connect as oe user. The JRepUtil class implements these interactions.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the *parallel_clause*. Run U*old_release*.SQL where *old_release* refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (*digits* [, *precision*])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE | DISABLE}
|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]
...	Horizontal ellipsis points indicate either: ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code	CREATE TABLE ... AS *subquery*; SELECT *col1*, *col2*, ... , *coln* FROM employees;
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V$DATAFILE; NAME ---------- /fsl/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;

Convention	Meaning	Example

`lowercase`	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		`sqlplus hr/hr`
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	`c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32`
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`

Convention	Meaning	Example
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"` `C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)`
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. For Windows NT, the default location was `C:\orant`. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora`*`nn`*, where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle9i Database Getting Started for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

1

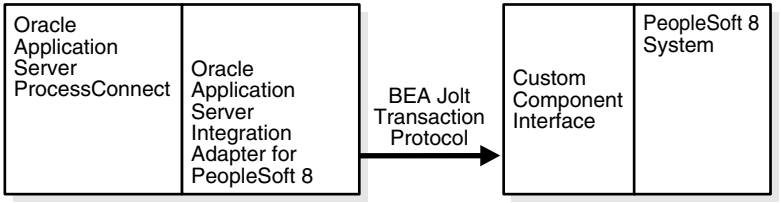
Introduction to Oracle Application Server Integration Adapter for PeopleSoft 8

Oracle Application Server ProcessConnect connects to a PeopleSoft system through the Oracle Application Server Integration Adapter for PeopleSoft 8. The Oracle Application Server Integration Adapter for PeopleSoft 8 provides connectivity and executes interactions on a PeopleSoft system. This chapter discusses the following topics:

Architecture: Oracle Application Server Integration Adapter for PeopleSoft 8

The Oracle Application Server Integration Adapter for PeopleSoft 8 is a JCA-based component that plugs in to Oracle Application Server ProcessConnect. Using Oracle Application Server ProcessConnect, you can access Component Interfaces. The Oracle Application Server Integration Adapter for PeopleSoft 8 communicates with the PeopleSoft system using the PeopleSoft `psjoa` classes which connects to the PeopleSoft system through the BEA System, Jolt Transaction Protocol. As shown in Figure 1–1, the Oracle Application Server Integration Adapter for PeopleSoft 8 runs on the machine running Oracle Application Server ProcessConnect.

Figure 1–1 Architecture



To use the Oracle Application Server Integration Adapter for PeopleSoft 8, a modification to the PeopleSoft environment is required. A custom component must be imported into the PeopleSoft system.

> **See Also:** "Importing the Custom Component Interface" on page 1-4

PeopleSoft Component Interfaces

The Oracle Application Server Integration Adapter for PeopleSoft 8 supports outbound interactions. The interactions have both request and reply records. Each record has one record element associated with it. The following naming convention is used for the records:

- Request records

  ```
  AEProtocolName_interactionGroupName_interactionName_Request
  ```

- Reply records

  ```
  AEProtocolName_interactionGroupName_interactionName_Reply
  ```

See Also: Chapter 3, "Defining an Interaction"

PeopleSoft Component Interfaces contain tables and data formats for specific tasks. A Component Interface declares the set of interactions that a Component Interface supports, but it does not implement the behavior. The Oracle Application Server Integration Adapter for PeopleSoft 8 provides standard interactions for Component Interfaces:

- Create
- CreateEx
- DeleteOnly
- Find
- Get
- Update
- UpdateEx

See Also: Chapter 4, "Component Interface Interactions"

Required Software

The Oracle Application Server Integration Adapter for PeopleSoft 8 connects to the PeopleSoft Component Interface, Version 8.17.02.

The following PeopleSoft 8 jar file must be accessible:

- `psjoa.jar`

Before using Component Interfaces, you must import a custom Component Interface into the PeopleSoft environment.

See Also: "Importing the Custom Component Interface" on page 1-4

Supported Platforms

Oracle Application Server Integration Adapter for PeopleSoft 8 is supported on:

- Solaris 8 (2.8)
- HP-UX 11.0

Postinstallation

Verify you are using PeopleSoft, Version 8.17.02.

Copy the following jar file from your PeopleSoft installation and place it in *ORACLE_install*/ip/adapters/lib:

- psjoa.jar

Verify you have imported a custom Component Interface into the PeopleSoft environment.

The files in *ORACLE_install*/ip/adapters/config/PeopleSoft are provided to create a custom component interface on the PeopleSoft system. The custom component interface must be created on the PeopleSoft system before the Oracle Application Server Integration Adapter for PeopleSoft 8 can be used. Refer to the detailed instructions in *ORACLE_install*/ip/adapters/config/PeopleSoft/readme.html.

> **See Also:** "Importing the Custom Component Interface" on page 1-4

Importing the Custom Component Interface

You must modify the PeopleSoft environment to use Oracle Application Server ProcessConnect with the PeopleSoft system. The custom component, GET_CI_INFO, must be imported into PeopleSoft. After importing the custom Component Interface, you can browse Component Interfaces within PeopleSoft. This task is only required on the initial setup of the PeopleSoft system to interact with Oracle Application Server ProcessConnect.

The following instructions explain how to import the custom Component Interface.

Creating the Component Interface

The following steps provide a brief overview of how to use a PeopleSoft application to create the Component Interface.

Using PeopleSoft's Application Designer:

1. Select **Start > Programs > PeopleSoft > Application Designer**.
2. Select a 3-tier connection type and click **OK**.

   For example, select **Application Server** from the list.

3. Select **File > New** in the Application Designer.
4. Select **Component Interface** and click **OK**.
5. Click **Select** in the Select dialog.

   A list of Component Interfaces is displayed.
6. Select any simple component, for example, `SIMPLECALENDAR`, and click **Select**.

   The custom interactions to be installed do not use any properties of the Component Interface.
7. Select **File > Save As**.
8. Type `GET_CI_INFO` in the **Save Name As** field and click **OK**.
9. Right-click any interaction of your new Component Interface.
10. Select **View PeopleCode**.

    A text editor window is displayed.
11. Access the *`ORACLE_HOME`*`/ip/adapters/config/PeopleSoft` directory and copy the contents of `get_ci_info.pc` into the text editor window.
12. Select **File > Save** to save the new Component Interface, `GET_CI_INFO`.

See Also: PeopleSoft online help for complete instructions

Setting Security

After importing the custom `GET_CI_INFO` Component Interface into PeopleSoft, set the security settings for the `GetCINamespace`, `GetDetails`, and `GetCollections` interactions for Oracle Application Server ProcessConnect.

Using PeopleSoft's Application Designer:

1. Select **Go > PeopleTools > Maintain Security**.
2. Select **Use > Permission Lists > Component Interface > Update/Display**.
3. Type an applicable permission list item in the Permission List text box, for example, `ALLPNLS`, and click **OK**.

   You can also click **OK** to retrieve a list of possible permission lists if you do not know which items are available.
4. In the list of Component Interfaces, find the following line:

   Click On This Row To Enter Data

5. Select this row and type `GET_CI_INFO` to add a new Component Interface in the Maintain Security - Use - Permission Lists dialog box.
6. Select **Edit** from the menu.

   A dialog box containing the permissions for this Component Interface is displayed.
7. Click **Full Access (All)** to set full access to all interactions.
8. Click **OK** to exit the Designer.

The PeopleSoft server is set. You can now use Oracle Application Server ProcessConnect and select PeopleSoft Component Interfaces interactions.

2

Defining a Delivery Channel

This chapter describes how to use Oracle Application Server ProcessConnect to define a delivery channel to connect to a PeopleSoft Component Interface system.

This chapter discusses the following topics:

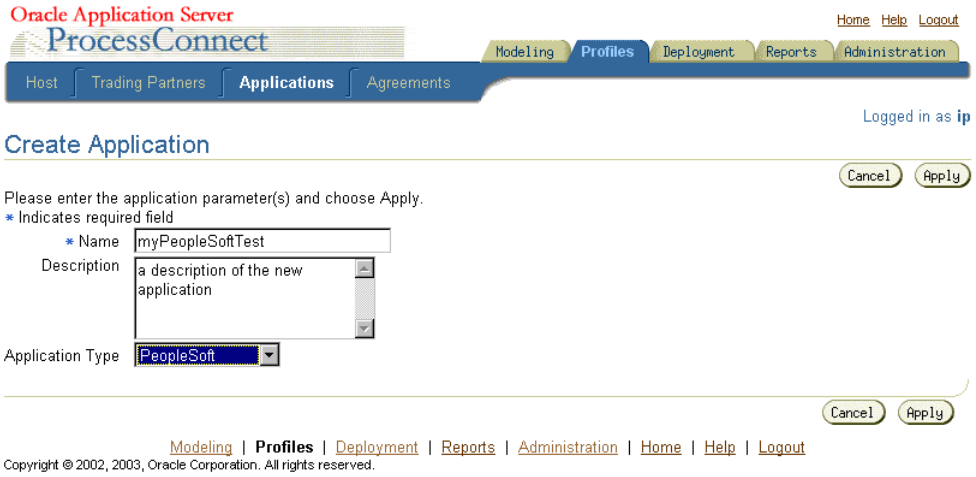
- Adding and Configuring a Delivery Channel
- Troubleshooting PeopleSoft Settings

Adding and Configuring a Delivery Channel

Part of the application definition includes adding a delivery channel for the adapter. Setting up the delivery channel in Oracle Application Server ProcessConnect requires information which is specific to the adapter.

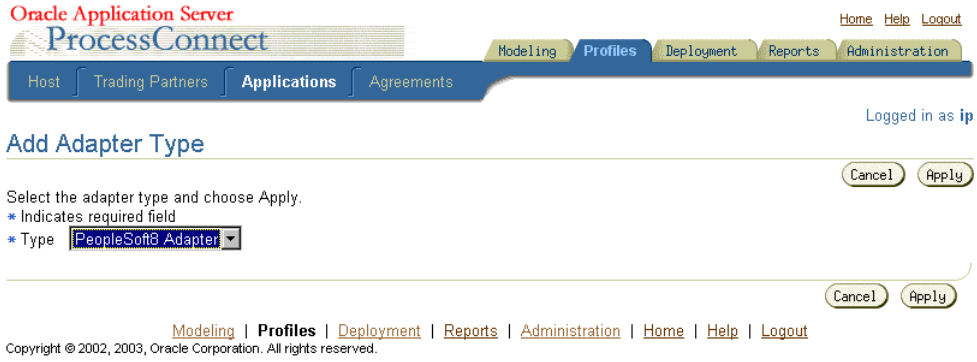
> **See Also:** *Oracle Application Server ProcessConnect User's Guide* for details about adding an application delivery channel in Oracle Application Server ProcessConnect

1. Select the **Profiles** > **Applications** tabs.
2. Click **Create**.



3. Type an application name in the **Name** field.
4. Select the **PeopleSoft** application type from the **Application Type** box and click **Apply**.
5. Click **Add** in the **Adapter Types** section to add the Oracle Application Server Integration Adapter for PeopleSoft 8.

   The Add Adapter Type page appears.



6. Select **PeopleSoft Adapter** in the **Type** box on the Add Adapter Type page and click **Apply**.

   The Adapter Type Details: PeopleSoft Adapter page is displayed.

7. Click **Create** in the **Delivery Channels** section of the Adapter Type Details page.

8. Add the following information to create a delivery channel.

   In Table 2–1 an asterisk (*) indicates a required field.

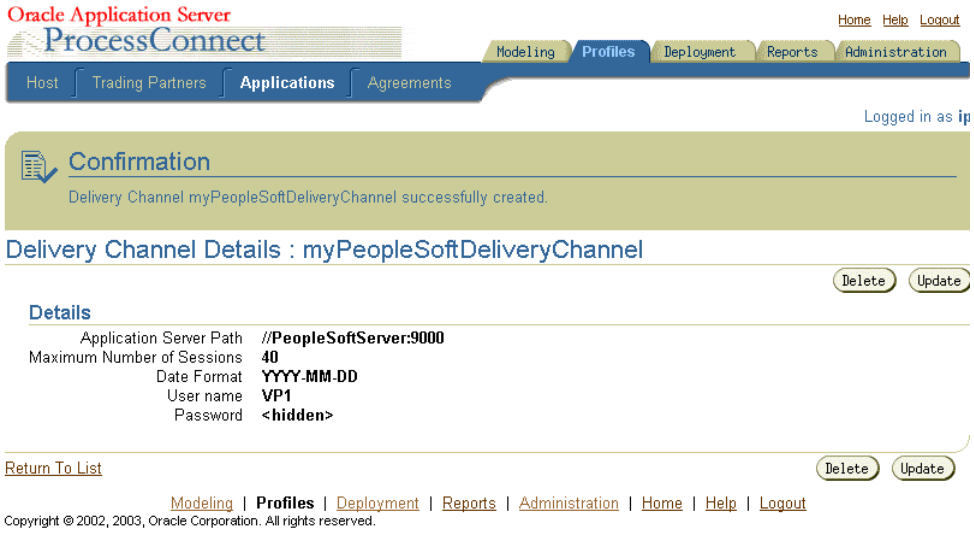
Table 2–1 PeopleSoft Specific Parameters

Field	Description
Name*	Enter a name for the delivery channel.
Application Server Path*	Enter a string representing the machine and port on which the PeopleSoft Application Server is running and listening. The syntax is `//machine_name:port`. Ask your PeopleSoft Administrator for the *port* value.
Maximum Number of Sessions	Enter the maximum number of sessions. The default value is 40 sessions. If you use a smaller value you may experience a degradation in runtime performance. The opposite is also true; a bigger value may exceed the ability of the server and can result in runtime errors.
Date Format	Date format when the value is not used as a `key`. Each `Y` represents a year digit (up to 4). Each `M` represents a month digit (up to 2). Each `D` represents a day digit (up to 2). Non-alphanumeric characters must appear literally. The default format is `YYYY-MM-DD`.
User Name*	Enter a PeopleSoft user name to log on to the PeopleSoft system.
Password*	Enter the password of the specified user. You must have developer rights to the areas of the PeopleSoft system you want to access.

See Also: "Troubleshooting PeopleSoft Settings" on page 2-6 for connection error information

9. Click **Apply** after entering your parameters to connect to a PeopleSoft system.

   The delivery channel Confirmation screen appears. You can modify any parameters and click **Update** in the confirmation page to change your parameters or click **Delete** to remove the channel.



10. Click **Return to List** to return to the Adapter Details page.

The delivery channel is added for the adapter and you can now add interactions.

See Also: Chapter 3, "Defining an Interaction"

Troubleshooting PeopleSoft Settings

Error ID	Possible Cause / Error Description	Possible Correction
E-JNI0004	No `psjoa.jar`. A Java exception occurred.	Verify location of the PeopleSoft `psjoa.jar` file. Refer to "Postinstallation" on page 1-4.
E-PSFT0030	No `psjoa.jar`. Failed to instantiate Component Interface Beans.	Verify location of the PeopleSoft `psjoa.jar` file. Refer to "Postinstallation" on page 1-4.
E-PSFT0019	Wrong server name. Connection to PeopleSoft Application Server failed.	Verify PeopleSoft host and user parameters. Refer to "Application Server Path*" on page 2-4.

Error ID	Possible Cause / Error Description	Possible Correction
E-PSFT0024	Wrong User Name and Password. Connection failed. Error Message: JavaClient is an Invalid User name, or you typed the wrong password.	PeopleSoft user name and password are required and are case-sensitive. Make sure you are entering the information in the correct upper and lower cases. Refer to "User Name*" on page 2-4 and "Password*" on page 2-4

3

Defining an Interaction

This chapter describes how to configure Oracle Application Server ProcessConnect to access Component Interfaces in a PeopleSoft system.

This chapter discusses the following topic:

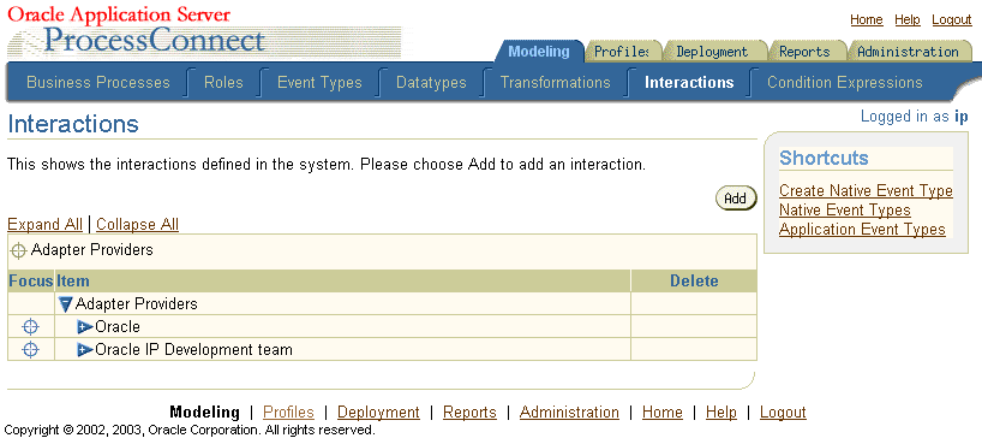
- Adding an Interaction

Adding an Interaction

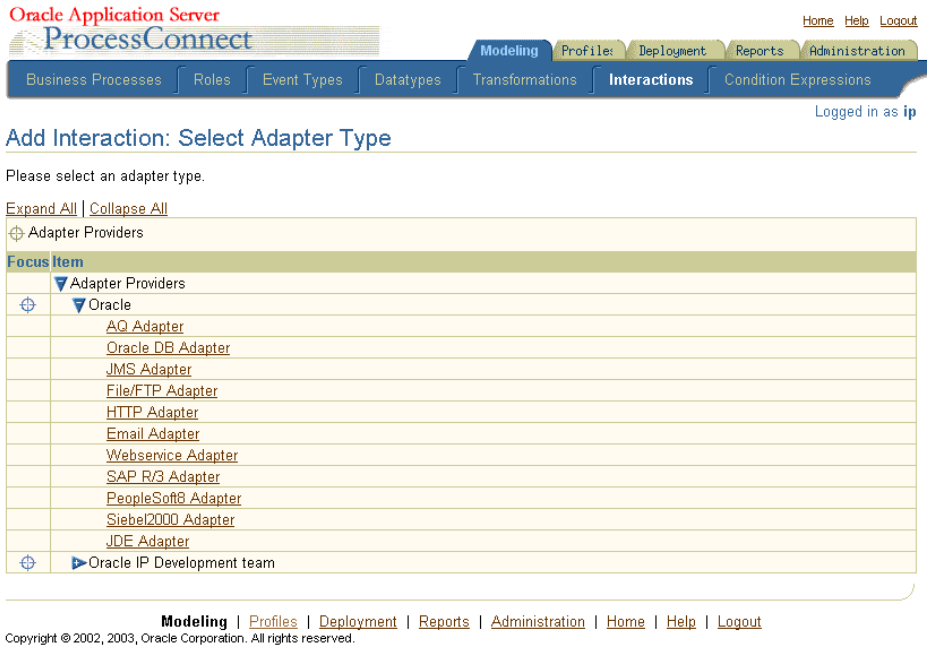
After defining a delivery channel for a PeopleSoft system, you can add interactions. Follow these instructions to add a Component Interface as an interaction in Oracle Application Server ProcessConnect.

> **See Also:** *Oracle Application Server ProcessConnect User's Guide* for details about interactions in Oracle Application Server ProcessConnect

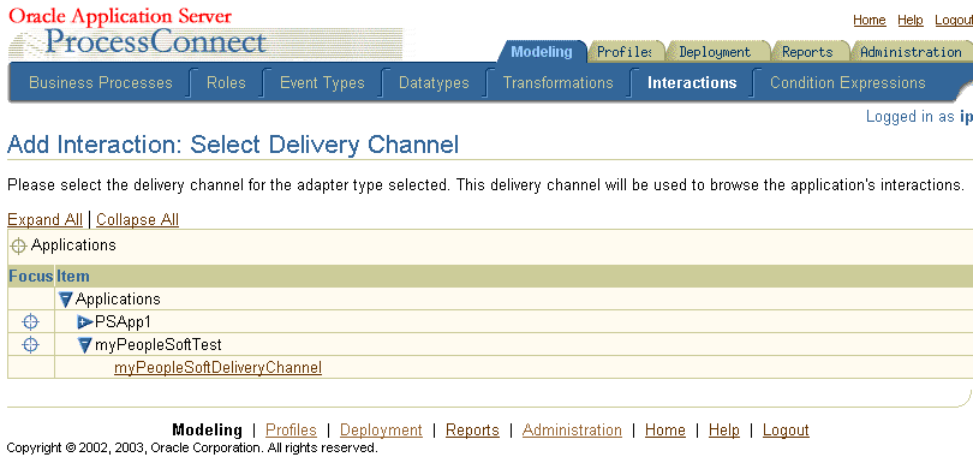
1. Select **Modeling > Interactions**.



2. Click **Add** to add an interaction.
3. Expand **Oracle**.

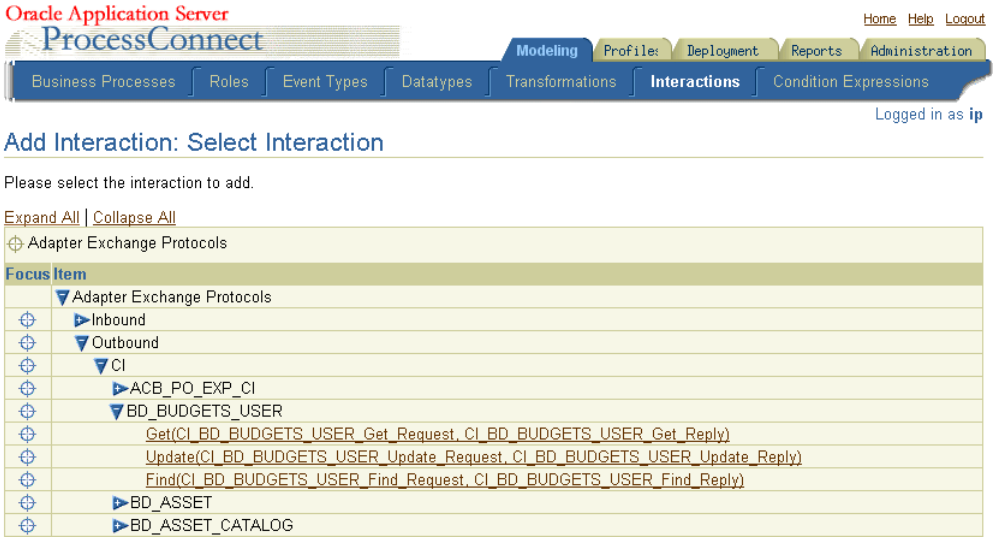


4. Select **PeopleSoft8 Adapter**.
5. Select a delivery channel.



6. Click the **Outbound > CI** nodes.

   The Add Interaction: Select Interaction page displays the Component Interfaces that you can access.

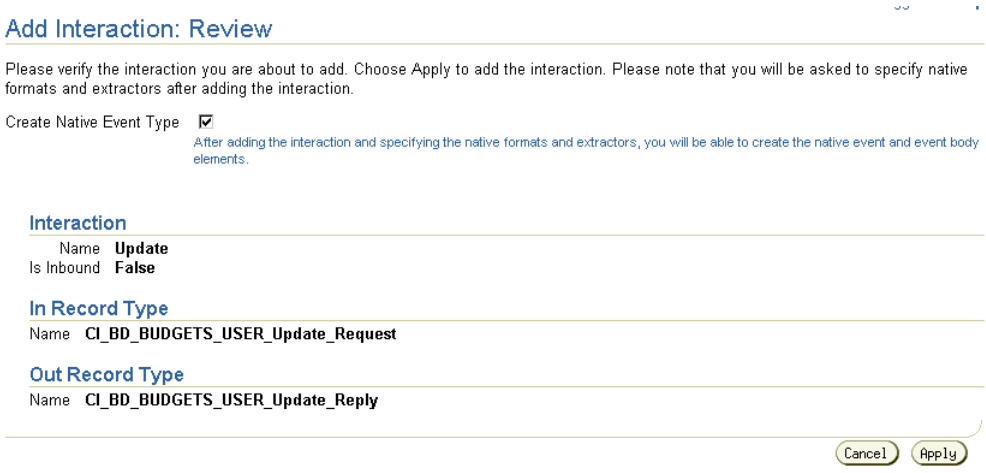


   When you open the **CI** folder you can view all of the available Component Interfaces in your PeopleSoft system. A **Component Interface** declares the set of interactions that a Component Interface supports, but it does not implement the behavior.

   Oracle Application Server ProcessConnect exposes standard interactions, **Create**, **CreateEx**, **Get**, **Find**, **Update**, and **UpdateEx**. User-defined interactions can also be exposed in the same manner as the standard interactions. Expanding the Component Interface displays the interactions. An interaction is the function of Component Interface that is capable of performing an operation.

   Expanding the Component Interface displays a long list of Component Interfaces. Each Component Interface is similar to data than a real object with behavior. Therefore, there is a set of methods for each Component Interface; however, not every Component interface has all supported methods.

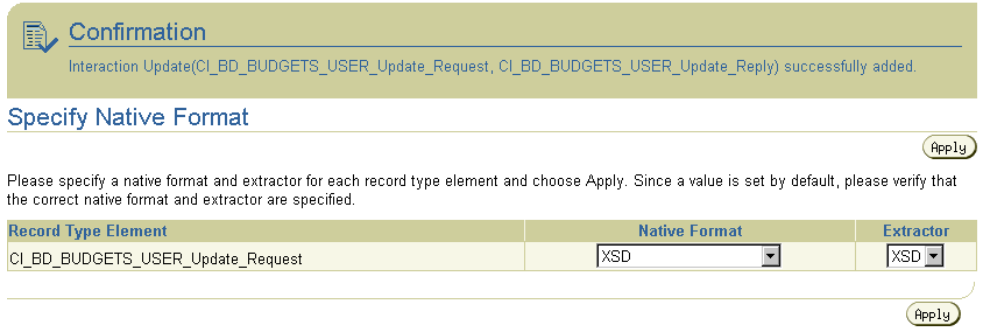
7. Select a group and an interaction. For this discussion, expand **BD_BUDGETS_USER** and select the **UpdateEx** interaction.



The Add Interaction: Review page displays the details. For an Outbound selection there is an InRecord Type and an OutRecord Type.

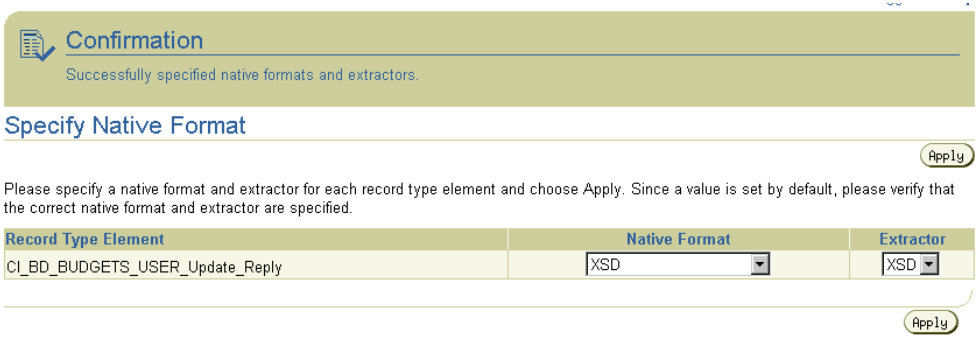
8. Click **Apply**.

   The Confirmation screen appears, allowing you to specify the Native Format of the request.



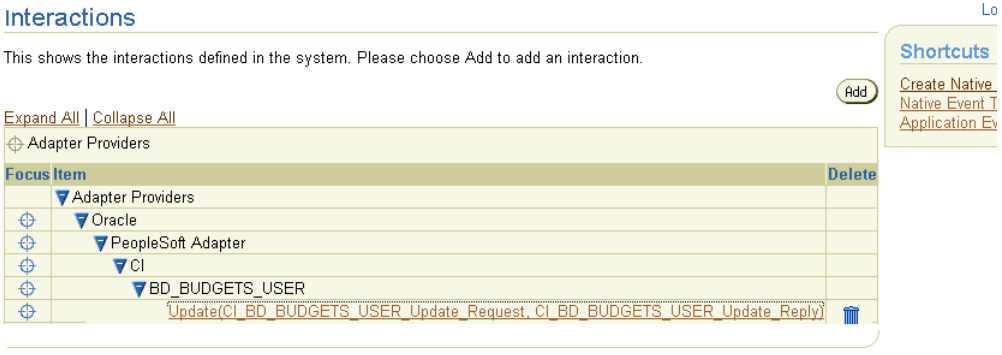
9. Verify that the selection for the Native Format and the Extractor of the request is **XSD** and click **Apply**.

   The Confirmation screen appears, allowing you to specify the Native Format of the reply.

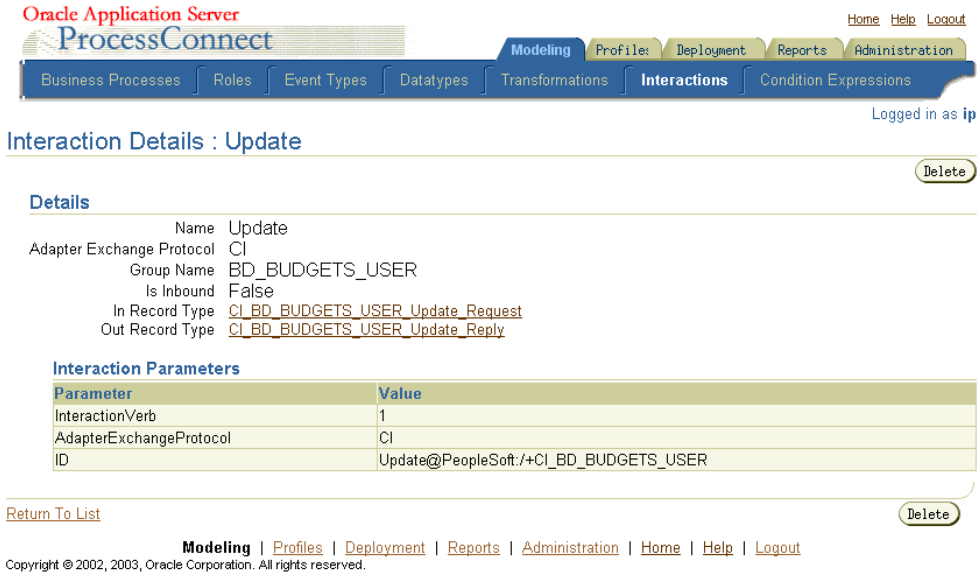


10. Verify that the selection for the Native Format and the Extractor of the reply is **XSD** and click **Apply**.
11. The interaction continues into the Create Native Event Type wizard. You can continue using the instructions in the *Oracle Application Server ProcessConnect User's Guide* for this wizard.

On completion, the new interaction appears in the Interactions list.



You can click the interaction link for a complete view of the interaction details.


Oracle Application Server
ProcessConnect
Home Help Logout
Modeling Profiles Deployment Reports Administration
Business Processes Roles Event Types Datatypes Transformations Interactions Condition Expressions
Logged in as ip
Interaction Details : Update
Delete
Details
Name Update
Adapter Exchange Protocol CI
Group Name BD_BUDGETS_USER
Is Inbound False
In Record Type CI_BD_BUDGETS_USER_Update_Request
Out Record Type CI_BD_BUDGETS_USER_Update_Reply
Interaction Parameters
Parameter Value
InteractionVerb 1
AdapterExchangeProtocol CI
ID Update@PeopleSoft:/+CI_BD_BUDGETS_USER
Return To List
Delete
Modeling | Profiles | Deployment | Reports | Administration | Home | Help | Logout
Copyright © 2002, 2003, Oracle Corporation. All rights reserved.

4

Component Interface Interactions

This chapter provides information about the Component Interface standard and user-defined interactions for Oracle Application Server Integration Adapter for PeopleSoft 8. This chapter discusses the following topics:

Component Interface Standard Interactions

Oracle Application Server ProcessConnect provides the following standard interactions for Component Interfaces:

- Create
- CreateEx
- DeleteOnly
- Find
- Get
- Update
- UpdateEx

In the following discussions, a single *record* is correlated to multiple *keys*.

Create

Use the `Create` interaction to create a new record using a set of unique keys and specified properties.

Syntax

```
Create (key1, key2, ... keyn, properties)
```

where:

- *key* (in/out parameter)—The individual key parameters (*key1*, *key2*, .. *keyn*) must be supplied. This set of keys must not exist in the server database, that is, they must be unique. The keys correspond to the set of `Create Keys` as defined for the particular Component Interface.
- *properties* (structure)—Contains a complete structure of the Component Interface properties, which is inserted into the record created with the specified keys.

Description

You can call `Create()` without a set of explicit keys; however, the `Create` interaction returns a set of keys. This behavior is supported with PeopleCode, which is triggered on the server. For example, to create a Purchase Order, the client may not know what the next available PO number is. By specifying `NEXT` as the PO number key, the call triggers PeopleCode, which determines the next available PO

number. This information must be returned to the calling client, using the in/out key parameters.

The Oracle Application Server ProcessConnect `Create()` interaction is available if the PeopleSoft's `Create` and `Save` interactions in the Component Interface are enabled on the PeopleSoft server.

CreateEx

You use the **CreateEx** interaction to create a new record using a set of unique keys and specified properties.

Syntax

```
CreateEx(key1, key2, ..., keyn, interactiveMode, properties)
```

where

- `key` (in/out parameter). The individual key parameters (`key1`, `key2`, .. `keyn`) must be supplied. This set of keys must not exist in the server database, that is, they must be unique. The keys correspond to the set of `Create Keys` as defined for the particular Component Interface.
- The `properties` parameter is a structure that contains all the properties of the Component Interface. When the `CreateEx` interaction is called, these properties are inserted into the record created with the specified key(s).
- The `interactiveMode` flag is used for error handling. When accessing properties in a Component Interface, the Oracle Application Server Integration Adapter for PeopleSoft 8 uses PeopleSoft-provided APIs that read and write individual fields in the Component Interface. However, these changes are not propagated to the PeopleSoft server one at a time. Instead, the `psjoa.jar` (with which the Oracle Application Server Integration Adapter for PeopleSoft 8 interacts) packages all the changes and sends the changes to the server in one package. If any of the individual updates fail, a generic error is returned, which does not pinpoint the actual error. With the interactive mode set to True, every field update is sent to the server individually. This has a substantial impact on performance, but it does provide specific error information if the update fails (for example, invalid value used for setting a field).

  The `interactiveMode` provides maximum performance and provides error reporting at the field update level. To use this feature properly, it is recommended that normal calls be made with the `interactiveMode` set to FALSE. There should be no impact on performance. If an error is returned, the

same call can be re-tried with the interactiveMode flag set to True. When the call fails, the server returns a more precise error message.

Description

In some situations, it is common practice to call CreateEx() without a set of explicit keys, but the CreateEx interaction returns them. This behavior is supported with PeopleCode that gets triggered on the server. For example, to create a Purchase Order, the client may not know what the next available PO number is. By specifying NEXT as the PO number key, the call triggers PeopleCode which determines the next available PO number. This information must be returned to the calling client, using the in/out key parameters.

The Oracle Application Server ProcessConnect CreateEx() interaction is available if the PeopleSoft's Create and Save interactions in the Component Interface are enabled on the PeopleSoft server.

DeleteOnly

The DeleteOnly method allows you to delete items in a collection.

Syntax

```
DeleteOnly(key1, key2, ..., keyn, correctionMode, interactiveMode, properties)
```

where

- All the key parameters must be supplied. This set of keys must exist in the server database, else an error occurs. The keys correspond to the set of Get Keys as defined for the particular Component Interface.
- The correctionMode is a Boolean flag, when set to True, allows deletion of past effective-dated items in a collection. Specifically, it allows the deletion of items that have EFFDT prior to the current effective date. Without this flag set to True, any modification to these items result in an error returned from PeopleSoft server.

> **Note:** The correctionMode argument is only exposed for those Component Interfaces that contain effective-dated items. Otherwise it is not shown as part of the argument.

Note: Setting the `correctionMode` to True should be avoided in a production environment. This is also the recommendation from PeopleSoft when using the `correctionMode`—events that have already occurred (as determined by the past EFFDT key) should not be modified. This allows for the creation of an audit trail. The `correctionMode` flag in DeleteOnly allows this safety mechanism to be bypassed. The recommended practice is for past events to be deactivated by setting a field in the item.

- The `interactiveMode` flag is used for error handling. When accessing properties in a Component Interface, the Oracle Application Server Integration Adapter for PeopleSoft 8 uses PeopleSoft-provided APIs that read and write individual fields in the Component Interface. However, these changes are not propagated to the PeopleSoft server one at a time. Instead, the `psjoa.jar` (with which the Oracle Application Server Integration Adapter for PeopleSoft 8 interacts) packages all the changes and sends the changes to the server in one package. If any of the individual updates fail, a generic error is returned, which does not pinpoint the actual error. With the interactive mode set to True, every field update is sent to the server individually. This has a substantial impact on performance, but it does provide specific error information if the update fails (for example, invalid value used for setting a field).

  The `interactiveMode` provides maximum performance and provides error reporting at the field update level. To use this feature properly, it is recommended that normal calls be made with the `interactiveMode` set to FALSE. There should be no impact on performance. If an error is returned, the same call can be re-tried with the `interactiveMode` flag set to True. When the call fails, the server returns a more precise error message.

- `properties` contain a subset of the structure that exists on the server. All items that are *leaves* are deleted.

Description

The `properties` have the same datatype as the CreateEx or UpdateEx methods of this Component Interface; however, only the key values are important. The non`key` values are ignored. The `key` values must match those on the server, otherwise an exception is raised.

The following demonstrates the use of the `key` values. If a collection contains the items:

```
item0
item1
item2
item3
```

you can delete `item1` and `item3` by providing the keys of `item1` and `item3` in the `properties`:

```
item1
item3
```

After the call, the server has the remaining items in the collection:

```
item0
item2
```

The second example, shows the items containing other collections:

```
item0
    item0a
item1
    item1a
    item1b
    item1c
item2
    item2a
    item2b
```

You can delete `item1b` and all of `item2` by giving the keys to `item1b` and `item2`:

```
item1
    item1b
item2
```

By providing an empty subcollection for `item2`, you turn it into a leaf and that entire subbranch is deleted. After the call, the server has the remaining items:

```
item0
    item0a
item1
    item1a
    item1c
```

Find

Use the Find interaction to return a list of keys that satisfy the supplied partial search keys. If the Component Interface has only one instance without a key, then the Find() interaction is not generated.

See Also: "Get" on page 4-7

Syntax

```
Find (partialKey, keyList)
```

where:

- *partialKey* (structure)—The individual keys are optional.
- *keyList* (output parameter)—A list of keys that matches the partialKey.

The keys correspond to the set of Find Keys as defined for the particular Component Interface.

Description

When specifying *partialKey*, you can use the same wildcard search available from the PeopleSoft internal Find() interaction. For example, the partial ACCOUNT key of 11 returns all ACCOUNT keys that start with 11, whereas %40 returns all ACCOUNT keys that contain 40 anywhere within the key. The partial key _4_4 returns all ACCOUNT keys with the character 4 in the second and fourth positions.

The Oracle Application Server ProcessConnect Find() interaction is available if the PeopleSoft Find interaction in the Component Interface is enabled and Get keys are available.

Note: With the current implementation of the PeopleSoft Server, if more than 300 items match the search criteria, the call fails. This is a restriction of the PeopleSoft server.

Get

Use the Get interaction to retrieve properties based on the input key parameters (*key1, key2, ... keyn*). The output parameter is a structure containing the properties of the record that matches the key parameters. If the Component Interface has only one instance without a key, then the Get interaction does not contain any key parameters.

See Also: "Find" on page 4-7

Syntax

```
Get (key1, key2, ... keyn, properties)
Get (key1, key2, ... keyn, getHistoryItems, properties)
```

where:

- All the *key* parameters must be supplied. This set of keys must exist in the server database; otherwise an error occurs. The keys correspond to the set of `Get Keys` as defined for the particular Component Interface.
- *properties* (structure)—Contains a complete structure of the Component Interface properties, which is returned upon completion of the call.
- *getHistoryItems*—a Boolean value. If the properties of the Component Interface contain effective dated items below level 0 (that is, a field with a name of EFFDT) the Boolean parameter, `getHistoryItems`, is used to control the range of effective dated items returned.
  - True—all effective dated items are returned as a sequence (which could be embedded in any level). These include all past effective dated items, the current effective dated item, as well as all future effective dated items
  - False—Only the current and all future effective dated items are returned. If subsequent calls to update on the same instance are made, then getHistoryItems should be set to `False`.

Description

If the Component Interface does not have a key, as in the case where only one instance exists, then the `Get()` interaction has the form:

```
Get(properties)
```

See Also: PeopleSoft documentation for more information on effective dated items

The Oracle Application Server ProcessConnect `Get()` interaction is available if the PeopleSoft `Get` interaction in the Component Interface is enabled.

See Also: "UpdateEx" on page 4-10

Update

Use the Update interaction to update properties based on the input key parameters (*key1*, *key2*, … *keyn*).

Syntax

```
Update (key1, key2, ... keyn, properties)
```

where:

- All the *key* parameters must be supplied. This set of keys must exist in the server database; otherwise an error occurs. The keys correspond to the set of Get Keys as defined for the particular Component Interface.
- *properties* (structure)—Contains a complete structure of the Component Interface properties, which replaces the existing properties in the database.

Description

When calling this interaction, the properties of the record corresponding to the keys are replaced by the properties input parameter. All collections within the original record are deleted and replaced by those in the input parameter, properties. The sizes of these collections do not have to match, as the procedure within Update is to delete all existing collection items and then insert the given ones.

If the properties of the Component Interface contain effective dated items below level 0 (that is, a key field with a name of EFFDT) an additional parameter is required, getHistoryItems. This parameter is of type Boolean. If it is set to True then all effective dated items are returned as a sequence (which could be embedded in any level). These include all past effective dated items, the current effective dated item, as well as all future effective dated items. If the getHistoryItems parameter is set to False, only the current and all future effective dated items are returned. If subsequent calls to update on the same instance are to be made, then getHistoryItems should be set to False.

If the Component Interface does not have any keys, as in the case where only one instance can exist, then the Update() interaction has the form:

```
Update(properties)
```

The Oracle Application Server ProcessConnect Update() interaction is available if the PeopleSoft Get and Save interactions in the Component Interface are enabled in the PeopleSoft application.

UpdateEx

You use the **UpdateEx** interaction to update properties based on the input key parameters (`key1`, `key2`, … `keyn`). Using UpdateEx, it is not possible to delete items in a collection. A separate interaction, **DeleteOnly**, facilitates deletion.

Syntax

```
UpdateEx (key1, key2, ... keyn, correctionMode, interactiveMode, properties)
```

where

- All the `key` parameters must be supplied. This set of keys must exist in the server database, else an error occurs. The keys correspond to the set of Get `Keys` as defined for the particular Component Interface.
- The `correctionMode` is a Boolean flag, when set to True, allows modifications to Component Interfaces with effective-dated items either by updating the field values, or by inserting new items into a collection. Specifically, it allows modification to items that have EFFDT prior to the current effective date. Without this flag set to True, any modification to these items result in an error returned from PeopleSoft server.

  Note: The `correctionMode` argument is only exposed for those Component Interfaces that contain effective-dated items. Otherwise it is not shown as part of the argument.

  Note: Setting the `correctionMode` to True should be avoided in a production environment. This is also the recommendation from PeopleSoft when using the `correctionMode`—events that have already occurred (as determined by the past EFFDT key) should not be modified. This allows for the creation of an audit trail. The `correctionMode` flag in UpdateEx allows this safety mechanism to be bypassed. The recommended practice is for past events to be deactivated by setting a field in the item, and then adding (instead of deleting) the updated item.

- The `interactiveMode` flag is used for error handling. When accessing properties in a Component Interface, the Oracle Application Server Integration Adapter for PeopleSoft 8 uses PeopleSoft-provided APIs that read and write

individual fields in the Component Interface. However, these changes are not propagated to the PeopleSoft server one at a time. Instead, the `psjoa.jar` (with which the Oracle Application Server Integration Adapter for PeopleSoft 8 interacts) packages all the changes and sends the changes to the server in one package. If any of the individual updates fail, a generic error is returned, which does not pinpoint the actual error. With the interactive mode set to True, every field update is sent to the server individually. This has a substantial impact on performance, but it does provide specific error information if the update fails (for example, invalid value used for setting a field).

The `interactiveMode` provides maximum performance and provides error reporting at the field update level. To use this feature properly, it is recommended that normal calls be made with the `interactiveMode` set to FALSE. There should be no impact on performance. If an error is returned, the same call can be re-tried with the `interactiveMode` flag set to True. When the call fails, the server returns a more precise error message.

Description

When calling this interaction, the properties of the record corresponding to the keys are replaced by the input parameter `properties`. All collections within the original record are deleted and replaced by those in the input parameter, `properties`. The sizes of these collections do not have to match, as the procedure within `UpdateEx` is to delete all existing collection items and then insert the given ones.

If the properties of the Component Interface contain effective dated items, then the `properties` parameter must contain all future effective dated items, as the original list is replaced. This provides the mechanism for adding and deleting future effective dated items. However, if the properties also contain past effective dated items, an error is returned, as past effective dated items cannot be modified. If the current effective dated item is also included, it is ignored. This permits the client to call `Get()` with the `getHistoryItems` parameter set to False, and then modify any future effective dated items or add new future effective dated items, and then passing the structure as parameter for the `UpdateEx()` interaction.

If the Component Interface does not have any key, as in the case where only one instance can exist, then the `UpdateEx()` interaction has the form:

```
UpdateEx(correctionMode, interactiveMode, properties)
```

The Oracle Application Server ProcessConnect `UpdateEx()` interaction is available if the PeopleSoft `Get` and `Save` interactions in the Component Interface are enabled in the PeopleSoft application.

Component Interface User-Defined Interactions

Oracle Application Server ProcessConnect supports user-defined interactions in Component Interfaces. The signatures are of the form:

```
myRet=myMethod(parameter1, parameter2, ...)
```

where:

- *parameter1*,*parameter2*—Input parameters
- *myRet*—The return value

The parameters can only be input parameters to the interaction. Only one value can be returned from the interaction as the return parameter.

Note: The Component Interface that contains user-defined interactions must have the PeopleSoft `Find` and `Get` interactions enabled.

Basic Datatypes

Table 4–1 describes the list of basic types in PeopleSoft and how they map to Oracle Application Server ProcessConnect types. The `xsd` prefix stands for the namespace `http://www.w3.org/2001/XMLSchema`.

Table 4–1 Basic Datatypes

PeopleSoft	Oracle Application Server ProcessConnect
Char(n)	xsd:string
Date	xsd:date
DateTime	xsd:dateTime
Long(n)	xsd:string
Nbr(n, 0) ; n = 1 to 2	xsd:byte
Nbr(n, 0) ; n = 3 to 4	xsd:short
Nbr(n, 0) ; n = 5 to 9	xsd:int
Nbr(n, 0) ; n = 10 to 14	xsd:double
Nbr(n, 0) ; n > 14	xsd:string

Table 4–1 (Cont.) Basic Datatypes

PeopleSoft	Oracle Application Server ProcessConnect
Nbr(n, d); n = 1 to 14; d > 0	xsd:double
Nbr(n, d); n > 14; d > 0	xsd:string
Sign (n, 0) ; n = 1 to 2	xsd:byte
Sign (n, 0) ; n = 3 to 4	xsd:short
Sign (n, 0) ; n = 5 to 9	xsd:int
Sign (n, 0) ; n = 10 to 14	xsd:double
Sign (n, 0) ; n > 14	xsd:string
Sign (n, d); n = 1 to 14; d > 0	xsd:double
Sign (n, d); n > 14; d > 0	xsd:string
Time	xsd:time

Basic Type Special Considerations

The following provides further explanations for datatypes that require special attention.

- Strings
- Numeric Types
- Date/Time Types

Strings

Although these types can contain strings of any length, you may still have length limitations at runtime. A cast failure occurs at runtime if a client program is sending a string to PeopleSoft as a key or a property, and if that string is longer than *n* (refer to the table). This is not true if *n* is zero. In that case, PeopleSoft accepts strings of any length, up to the maximum as defined by the back-end database.

Datatype	Description
`Char(`*n*`)`	Any string of *n* characters. *n* is between 1 and 254.

Datatype	Description
`Long(n)`	Any string of *n* characters. *n* is between 0 and 64000. `Long(0)` is an unbounded string.

Numeric Types

The `Nbr` and `Sign` datatypes in PeopleSoft map to datatype `xsd:byte`, `xsd:short`, `xsd:int`, `xsd:double`, or `xsd:string` based on the values of *n* and *d*, which specify the number of integral and fractional digits respectively. Due to a current limitation in the PeopleSoft API, output values cannot have a precision that exceeds a float; for example, the return value of the client call can only keep a maximum precision of 7 digits regardless of the client type.

If the PeopleSoft type is `Nbr`, verify that the client inputs positive values. The precision may be lost if it exceeds the specified *n* and *d*. For example, 123.456 loses two digits when converted to a `Nbr(3,1)`.

Date/Time Types

In PeopleSoft, the `Date` type contains only the date information. Time properties may be defined with different levels of precision:

- hour:minutes (Time default format)
- hour:minutes:seconds (Time Scnds format)
- hour:minutes:seconds:milliseconds (Time McroS format)

The Oracle Application Server Integration Adapter for PeopleSoft 8 translates all `Date` types to `xsd:date`, all `Time` types to `xsd:time` and all `DateTime` types to `xsd:dateTime`.

Because the `Time` part in `Date` values always allow you to specify up to milliseconds, you may lose precision if you are not aware of the actual precision of the field in the PeopleSoft database.

`Date`, `DateTime` and `Time` types appearing in user-defined interactions are mapped as strings, due to a limitation in PeopleSoft.

Effective Date Properties

PeopleSoft provides the ability to schedule and keep track of planned items by using a special property called `Effective Date` (abbreviated `EFFDT`). Such items

are either in effect or merely planned, depending on whether their date is before or after PeopleSoft's current date.

If the properties of a Component Interface contain such effective dated items (that is, a field with a name of `EFFDT`), the adapter makes it possible for callers to retrieve the complete set of values or only those values not yet effective—those that can still be changed.

For Components Interfaces with properties that include an effective date, the adapter provides an additional parameter, called `getHistoryItems`, to the `Get` operations. This parameter is of type Boolean and if it is set to True then all effective dated items are returned. These include all past effective dated items, the current effective dated item, as well as all future effective dated items.

If the `getHistoryItems` parameter is set to False only the current and all future effective dated items are returned. Choose False if your intention is to add or change to these items (because past items cannot be changed).

It is also possible to have multiple Effective Dated items having the same Effective Date. In this situation, an additional property, `Effective Sequence` (EFFSEQ), must also be provided. The values of the EFFSEQ must be unique to differentiate items with the same Effective Date.

The `correctionMode` argument in both the UpdateEx and DeleteOnly interactions control whether past effective dated items can be modified. If it is set to True, all items can be modified. Otherwise, modifying past effective dated item generates an exception.

When calling the `Update` interaction on a Component Interface that has Effective Dated items, you must take care not to include any Effective Dates of a value earlier than PeopleSoft's current Effective Date, or the call fails with an exception. However, the current Effective Dated item can be included as it is bypassed when setting properties. If Effective Sequence exists, then all current Effective Dated items with matching Effective Sequences in the server are skipped when setting properties.

Index