PeopleSoft®

# PeopleSoft Enterprise Components for Financials, Enterprise Service Automation and Supply Chain Management 8.8 PeopleBook

**December 2003**

PeopleSoft Enterprise Components for Financials, Enterprise Service Automation and Supply Chain Management 8.8 PeopleBook
SKU FSCM88ECP-B 1203


## Open Source Disclosure

PeopleSoft takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation. The following open source software may be used in PeopleSoft products and the following disclaimers are provided.

Apache Software Foundation

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OpenSSL

Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SSLeay

Copyright (c) 1995-1998 Eric Young. All rights reserved.

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Loki Library

Copyright (c) 2001 by Andrei Alexandrescu. This code accompanies the book:

Alexandrescu, Andrei. "Modern C++ Design: Generic Programming and Design Patterns Applied". Copyright (c) 2001. Addison-Wesley. Permission to use, copy, modify, distribute and sell this software for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

# Contents

# Part 3
# Common Objects and Components

# Part 4
# Portal Utilities

# Part 5
# Enterprise Integration

# About These PeopleBooks

PeopleBooks provide you with the information that you need to implement and use PeopleSoft applications.

This preface discusses:

- PeopleSoft application prerequisites.
- PeopleSoft application fundamentals.
- Related documentation.
- Typographical elements and visual cues.
- Comments and suggestions.
- Common elements in PeopleBooks.

---

**Note.** PeopleBooks document only page elements that require additional explanation. If a page element is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common elements for the section, chapter, PeopleBook, or product line. Elements that are common to all PeopleSoft applications are defined in this preface.

---

## PeopleSoft Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use PeopleSoft applications.

See *Using PeopleSoft Applications.*

You might also want to complete at least one PeopleSoft introductory training course.

You should be familiar with navigating the system and adding, updating, and deleting information by using PeopleSoft windows, menus, and pages. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your PeopleSoft applications most effectively.

## PeopleSoft Application Fundamentals

Each application PeopleBook provides implementation and processing information for your PeopleSoft database. However, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals PeopleBook. Each PeopleSoft product line has its own version of this documentation.

The application fundamentals PeopleBook consists of important topics that apply to many or all PeopleSoft applications across a product line. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of this central PeopleBook. It is the starting point for fundamentals, such as setting up control tables and administering security.

# Related Documentation

This section discusses how to:

- Obtain documentation updates.
- Order printed documentation.

## Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on the PeopleSoft Customer Connection web site. Through the Documentation section of PeopleSoft Customer Connection, you can download files to add to your PeopleBook Library. You'll find a variety of useful and timely materials, including updates to the full PeopleSoft documentation that is delivered on your PeopleBooks CD-ROM.

**Important!** Before you upgrade, you must check PeopleSoft Customer Connection for updates to the upgrade instructions. PeopleSoft continually posts updates as the upgrade process is refined.

### See Also

PeopleSoft Customer Connection web site, http://www.peoplesoft.com/corp/en/login.asp

## Ordering Printed Documentation

You can order printed, bound volumes of the complete PeopleSoft documentation that is delivered on your PeopleBooks CD-ROM. PeopleSoft makes printed documentation available for each major release shortly after the software is shipped. Customers and partners can order printed PeopleSoft documentation by using any of these methods:

- Web
- Telephone
- Email

### Web

From the Documentation section of the PeopleSoft Customer Connection web site, access the PeopleSoft Press web site under the Ordering PeopleBooks topic. The PeopleSoft Press web site is a joint venture between PeopleSoft and Consolidated Publications Incorporated (CPI), the book print vendor. Use a credit card, money order, cashier's check, or purchase order to place your order.

### Telephone

Contact CPI at 800 888 3559.

### Email

Send email to CPI at psoftpress@cc.larwood.com.

### See Also

PeopleSoft Customer Connection web site, http://www.peoplesoft.com/corp/en/login.asp

# Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.

## Typographical Conventions

The following table contains the typographical conventions that are used in PeopleBooks:

| Typographical Convention or Visual Cue | Description |
|---|---|
| **Bold** | Indicates PeopleCode function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call. |
| *Italics* | Indicates field values, emphasis, and PeopleSoft or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply.<br>We also use italics when we refer to words as words or letters as letters, as in the following: Enter the number 0, not the letter *O*. |
| KEY+KEY | Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press W. |
| `Monospace font` | Indicates a PeopleCode program or other code example. |
| " " (quotation marks) | Indicate chapter titles in cross-references and words that are used differently from their intended meanings. |

| Typographical Convention or Visual Cue | Description |
|---|---|
| . . . (ellipses) | Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax. |
| { } (curly braces) | Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe ( \| ). |
| [ ] (square brackets) | Indicate optional items in PeopleCode syntax. |
| & (ampersand) | When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object.<br><br>Ampersands also precede all PeopleCode variables. |
| (ISO) | Information that applies to a specific country, to the U.S. federal government, or to the education and government market, is preceded by a three-letter code in parentheses.<br><br>The code for the U.S. federal government is USF; the code for education and government is E&G, and the country codes from the International Standards Organization are used for specific countries. Here is an example:<br><br>(GER) If you're administering German employees, German law requires you to indicate special nationality and citizenship information for German workers using nationality codes established by the German DEUEV Directive. |
| Cross-references | PeopleBooks provide cross-references either below the heading "See Also" or on a separate line preceded by the word *See*. Cross-references lead to other documentation that is pertinent to the immediately preceding documentation. |

# Visual Cues

PeopleBooks contain the following visual cues.

## Notes

Notes indicate information that you should pay particular attention to as you work with the PeopleSoft system.

---

**Note.** Example of a note.

---

A note that is preceded by *Important!* is crucial and includes information that concerns what you must do for the system to function properly.

**Important!** Example of an important note.

### Warnings

Warnings indicate crucial configuration considerations. Pay close attention to warning messages.

***Warning!*** Example of a warning.

## Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about PeopleBooks and other PeopleSoft reference and training materials. Please send your suggestions to:

PeopleSoft Product Documentation Manager PeopleSoft, Inc. 4460 Hacienda Drive Pleasanton, CA 94588

Or send email comments to doc@peoplesoft.com.

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions.

## Common Elements in These PeopleBooks

| | |
|---|---|
| **As of Date** | The last date for which a report or process includes data. |
| **Business Unit** | An ID that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization. |
| **Description** | Enter up to 30 characters of text. |
| **Effective Date** | The date on which a table row becomes effective; the date that an action begins. For example, to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row. |
| **Once, Always,** and **Don't Run** | Select Once to run the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run.<br>Select Always to run the request every time the batch process runs.<br>Select Don't Run to ignore the request when the batch process runs. |
| **Report Manager** | Click to access the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list). |

| | |
|---|---|
| **Process Monitor** | Click to access the Process List page, where you can view the status of submitted process requests. |
| **Run** | Click to access the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format. |
| **Request ID** | An ID that represents a set of selection criteria for a report or process. |
| **User ID** | An ID that represents the person who generates a transaction. |
| **SetID** | An ID that represents a set of control table information, or TableSets. TableSets enable you to share control table information and processing options among business units. The goal is to minimize redundant data and system maintenance tasks. When you assign a setID to a record group in a business unit, you indicate that all of the tables in the record group are shared between that business unit and any other business unit that also assigns that setID to that record group. For example, you can define a group of common job codes that are shared between several business units. Each business unit that shares the job codes is assigned the same setID for that record group. |
| **Short Description** | Enter up to 15 characters of text. |

## See Also

*Using PeopleSoft Applications*

*PeopleSoft Process Scheduler*

# Enterprise Components Preface

This preface provides a general overview of the contents in Enterprise Components.

## Enterprise Components

Enteprise Components includes a variety of features that are generally available as common objects or components to all product lines delivered by PeopleSoft. This PeopleBook has five parts:

- *Getting Started With Enterprise Components.* Includes getting started sections describing each of the product areas.
- *PeopleSoft Directory Interface.* Describes how to enhance the integration between PeopleSoft and your LDAP directory service by providing a single, centralized user profile for PeopleSoft and non-PeopleSoft applications. The flexible directory structure and schema leverage the LDAP security model to control access to PeopleSoft applications.
- *Common Objects and Components.* Features generally available as common objects or components to all product lines—Interactive Reports (for Business Analytic Modeler), Market Rates, Datasets and Credit Card Interface.
- *Portal Utilities.* Describes navigation pages, navigation collections, maintenance of the portal registry.
- *Enterprise Integration.* Describes features related to integration from one PeopleSoft product to another, and integration between PeopleSoft and third party products.

# PART 1

# Getting Started With Enterprise Components

**Chapter 1**
**Getting Started With Enterprise Components**

# CHAPTER 1

# Getting Started With Enterprise Components

Enterprise Components comprises many features which have been subdivided into the following:

- PeopleSoft Directory Interface
- Common Objects and Components
- Portal Utilities
- Enterprise Integration

## Getting Started With PeopleSoft Directory Interface

Setting up to use the PeopleSoft Directory Interface requires you access components in PeopleSoft PeopleTools to define and activate nodes and transactions, create authentication and user profile maps, and so on. You also access components in Enterprise Components to configure and load the directory.

See Chapter 2, "Using PeopleSoft Directory Interface," page 9.

### Implementation Guidelines

Consider these guidelines for best results.

- *LDAP Searches*. Some LDAP searches may generate LDAP referrals to other servers participating in your directory. You must be able to ping by hostname all servers in the directory from the application server. If any server is unreachable by hostname from the application server, you can add a line for the server to the hosts. Your directory information tree must have user entries at the leaf level. This is required when an entry needs to be moved from one branch to another. The entry needs to be at the leaf level so that the system can read user attributes, one of which is the password. file on the application server.
- *Entry Limit*. In the directory, configure the entry limit value to be larger than the number of rows that you expect will be returned. The default value is usually not sufficient.
- *Directory Tree*. Your directory information tree must have user entries at the leaf level. This is required when an entry needs to be moved from one branch to another. The entry needs to be at the leaf level so that the system can read user attributes, one of which is the password.
- The following items apply to implementations that use Microsoft Active Directory:
  - The registry key HKLM\System\CurrentControlSet\Servies\NTDS\Paramteres\Schema Update Allowed must be present and set to a nonzero DWORD on the Active Directory FSMO Role Owner.
  - When creating structural object classes in Microsoft Active Directory, you need to specify containment. PsftJobs can be children of the following classes of objects only: builtinDomain, organizationalUnit, and domainDNS.

- You must add the server names in the Directory Setup component as they appear on the DNSHost Name attribute on the server entries under the CN=Sites entry.

# Getting Started With Common Objects and Components

Enterprise Components provides objects and components that can be used by various PeopleSoft applications. For set up or implementation information of the following components, refer to each chapter.

See Chapter 4, “Defining and Storing PeopleSoft Market Rates ,” Setting Up and Maintaining Market Rates, page 48.

See Chapter 5, “Using Datasets,” page 59.

See Chapter 6, “Using Interactive Reports,” page 65.

See Chapter 7, “Setting Up the Credit Card Interface,” page 73.

# Getting Started With Portal Utilities

PeopleSoft Enterprise Component portal utilities include components and processes that support the creation and maintenance of Navigation Collections and their publications.

The following sections cover system- and registry-level default options that you may choose to set up as a part of your implementation of Navigation Collections.

See Chapter 9, “Setting Up Navigation Collection Options,” Defining Navigation Collection System Options, page 107.

See Chapter 9, “Setting Up Navigation Collection Options,” Defining Navigation Collection Registry Options, page 111.

The following sections cover display options that you may choose to set up as a part of your implementation of navigation pages:

See Chapter 8, “Working with Navigation Pages,” Enabling Standard Navigation Page Display, page 102.

See Chapter 8, “Working with Navigation Pages,” Enabling Navigation Page Display Caching, page 103.

# Getting Started With Enterprise Integration

PeopleSoft enterprise integration technology includes various features and utilities. Which ones you use will depend on your PeopleSoft application.

If you are using Enterprise Integration Points, the following sections may be helpful:

See Appendix A, “EIP Naming Standards,” page 239.

See Appendix B, “PeopleSoft Design Patterns,” page 245.

See Chapter 14, “Activating Messaging EIPs,” Setting Up PeopleSoft Messaging EIPs, page 163.

See Chapter 14, “Activating Messaging EIPs,” Setting Up Related Languages, page 167.

See Chapter 16, “Using the Error Handling Utility,” Setting Up and Maintaining Message Errors, page 198.

# PART 2

# PeopleSoft Directory Interface

# CHAPTER 2

# Using PeopleSoft Directory Interface

This chapter provides an overview of PeopleSoft Directory Interface and discusses how to:

- Configure the directory.
- Load the directory schema.
- Set up mappings.
- Load PeopleSoft data into your directory.

## Understanding PeopleSoft Directory Interface

PeopleSoft Directory Interface utilizes Lightweight Directory Access Protocol (LDAP) directory services to authenticate users of PeopleSoft applications.

PeopleSoft Directory Interface provides additional mappings and integration points, such as application messages, that enable PeopleSoft data and LDAP data to stay synchronized. Most directory data, such as user ID, name, and email address, is also maintained in your PeopleSoft database. When you use PeopleSoft Directory Interface, you make selected PeopleSoft data available to the directory, and you maintain the data in PeopleSoft.

When information changes in the PeopleSoft database, PeopleSoft Directory Interface captures that updated information and automatically updates the equivalent information in the directory server, or it writes the updates to a file for you to apply at another time.

### Understanding Data Mapping

PeopleSoft information is stored in tables according to a relational model. The information in your LDAP directory is stored in trees according to a hierarchical model. You use PeopleSoft Directory Interface to map selected PeopleSoft data to corresponding data in the directory service. When PeopleSoft Directory Interface receives user data from the PeopleSoft database, it can map the data objects to the corresponding objects in the directory.

For PeopleSoft Directory Interface to map PeopleSoft information to your directory, it needs information about the directory hierarchical structure, or *directory information tree.*

Entries are made up of a *distinguished name* and *attribute/value pairs.* The distinguished name identifies an entry's position in the tree, and the attributes hold the data that make up the entry.

Available attributes for an object class entry are specified in the directory schema. You must load the schema into the directory interface before you can map PeopleSoft data to the directory.

PeopleSoft Directory Interface mapping tables map LDAP attributes to PeopleSoft application messages. Each application message contains selected information about a PeopleSoft record and its fields.

**Note.** Refer to PeopleSoft application documentation for information about specific messages delivered by PeopleSoft applications.

### Understanding Data Synchronization

Once you have loaded PeopleSoft data into your LDAP directory, you can synchronize the data. You use either:

- PeopleSoft Business Interlinks

  PeopleSoft Business Interlinks updates the data in real time, so that your directory information is always synchronized with PeopleSoft.

- LDAP Data Interchange Format (LDIF) files.

  You can load LDIF files as needed or defined by your system.

**Note.** The application server needs to be configured for receiving messages.

### Delivered Business Interlinks

PeopleSoft delivers the following business interlinks with PeopleSoft Directory Interface:

| | |
|---|---|
| **EO_DS_ADD** | Adds a new entry to the directory by creating a distinguished name and its corresponding attributes. |
| **EO_DS_BIND** | Authenticates the information exchanged between the database and the directory. |
| **EO_DS_DEL** | Deletes an entry from the directory. |
| **EO_DS_MODDN** | Renames a directory entry. Changes its distinguished name by renaming the actual entry or changing its position in the directory entry. |
| **EO_DS_MODIFY** | Changes the attributes of an entry. |
| **EO_DS_SEARCH** | Searches for directory entries and their corresponding attributes. |

# Overview of Using PeopleSoft Directory Interface

The section briefly describes the steps needed to use PeopleSoft Directory Interface.

## Setting Up in PeopleSoft Application Designer and Integration Broker

Perform the following steps in PeopleSoft PeopleTools Application Designer and Integration Broker.

### What To Do in PeopleSoft Application Designer

Access PeopleSoft Application Designer.

- Create authentication and user profile maps as needed. If you are going to authenticate users with the directory server, a PeopleSoft user profile is required—that is, a row in the PSOPRDEFN table where PeopleSoft user information is stored. In this context, you "cache" LDAP user information inside your PeopleSoft system. (Properties you specify in the Mandatory and Optional Properties pages of the Mappings component are the columns in PSOPRDEFN that the system populates with values from your directory server.) PeopleSoft applications use this cache of user information, not your directory server. Whenever a transaction requires user information, the application refers to the local PSOPRDEFN table instead of querying the directory server.
- Add Signon PeopleCode. Directory authentication requires Signon PeopleCode be enabled and configured with proper permissions. After a user signs onto the system and the Signon PeopleCode executes, PeopleSoft creates a row for the user in the user definition table by retrieving the LDAP information and creating a local cache. Signon PeopleCode maintains this row automatically and any changes made in the directory server are reproduced in the local cache. Using the Mappings component, set up mappings. To keep the data synchronized, you must map PeopleSoft data to the equivalent directory objects. Then, PeopleSoft Directory Interface associates the fields in the message to the attributes in the directory and updates the selected directory attributes with the field data from the message.
- Activate the DSCHNL channel. (Open the message channel and select Run.)

See *PeopleSoft PeopleTools 8.44 PeopleBook: Application Designer*

### What To Do in PeopleSoft Integration Broker

Access PeopleSoft Integration Broker.

- Activate a relevant node. This node should be the default local node.
- Define and activate transactions. This step depends on the application; for example, in an HR implementation messages, such as Dept, Location, Person and Job, are defined in addition to core messages such as DSMINPUT.

See *PeopleSoft PeopleTools 8.44 PeopleBook: Integration Broker*

## Using the Directory Configuration Component

Access Directory Configuration component from the browser menu.

- Using the Directory Configuration component, configure the directory. Enter appropriate connection information such as the server name (DNS or IP address) and the listening port number, the user DN and associated password.
- Using the Schema Management page, select names of Object Classes and Attribute Types and then cache the schema.
- To keep the data synchronized, you must map PeopleSoft data to the equivalent directory objects. Set up mappings using the Mappings component. Once this is complete, PeopleSoft Directory Interface associates the fields in the message to the attributes in the directory and updates the selected directory attributes with the field data from the message.
- Using the Membership Rules component, create rules and memberships, if desired.
- Load data in the directory.
- Set directory search criteria. Enter search parameters to query the directory and view the results.

# Common Elements Used in This Chapter

| | |
|---|---|
| **Directory ID** | Unique identifier for the directory. |
| **Description** | A brief description of the directory. |
| **Directory Product** | Specify the directory product from the drop down list. |
| **Default Connect DN** (default connect distinguished name) | Displays the connect distinguished name associated with the directory ID that you selected. Use this ID to connect to the directory server. |
| **Password** | Password to access the directory. |
| **LDAP Server** | The server name where the directory resides. |
| **Port** | The LDAP server port associated with the LDAP server you select. |
| **SSL Port** | The secure socket layer port. |

# Defining and Configuring the Directory

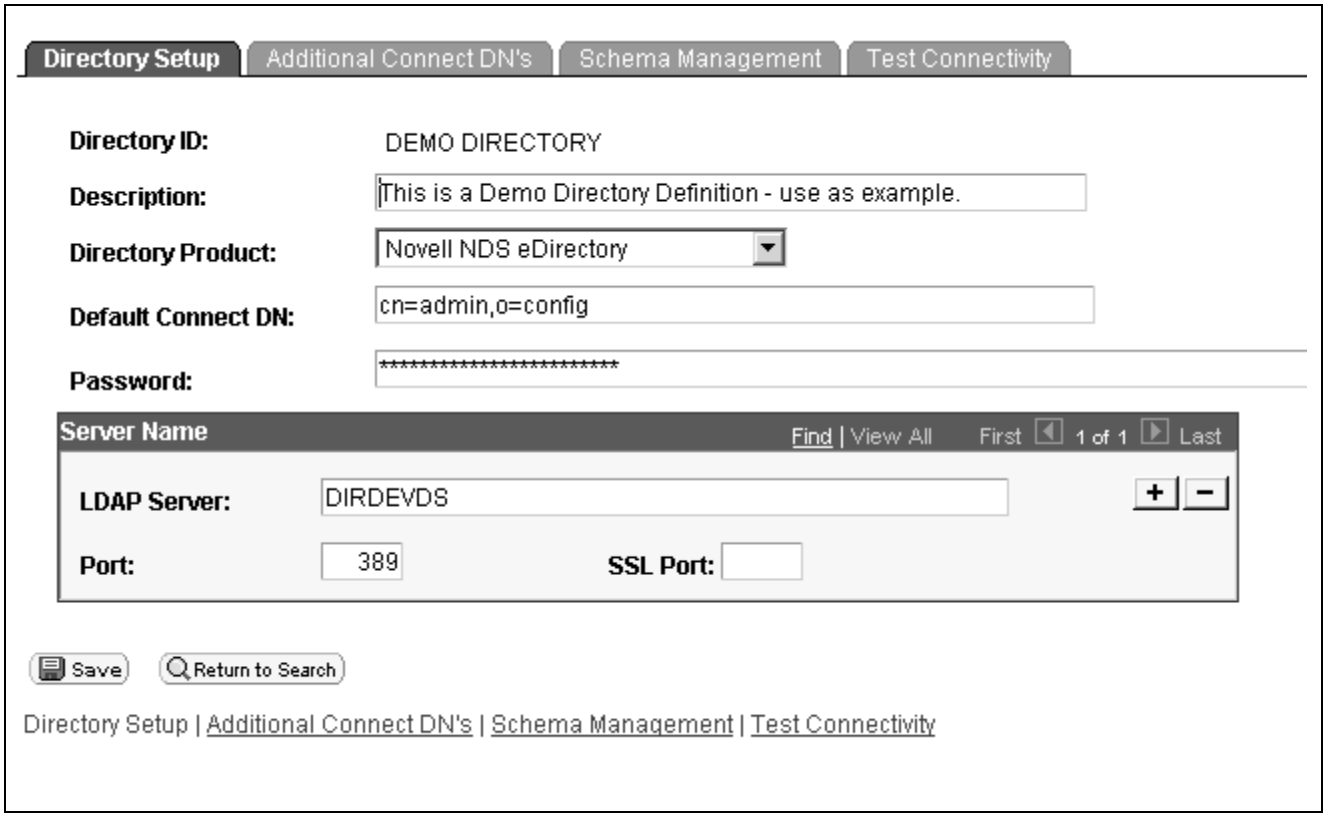
Use the Configure Directory component to define and configure the directory connection.

## Pages Used to Define and Configure the Directory

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Directory Setup | DSDIRSETUP | Enterprise Components, Directory Interface, Definitions, Directory Configurations | Enter values to configure the directory. |
| Additional Connect DNs | DSSERVERID | Enterprise Components, Directory Interface, Definitions, Directory Configurations, Additional Connect DNs | Add values for additional connect DNs. |
| Schema Management | DSEXTINSTALL | Enterprise Components, Directory Interface, Definitions, Directory Configurations, Schema Management | Manage schema, apply schema PeopleSoft schema extensions. |
| Test Connectivity | DSSRCHRSLT | Enterprise Components, Directory Interface, Definitions, Directory Configurations, Test Connectivity | Test the directory connectivity. |
| Cache Schema | DSSCHEMACACHE | Enterprise Components, Directory Interface, Definitions, Schema Caching | Cache the schema. |
| Delete Directory | DSPURGEDIRID | Enterprise Components, Directory Interface, Definitions, Directory Configurations, Directory Deletions | Delete the directory configuration. |

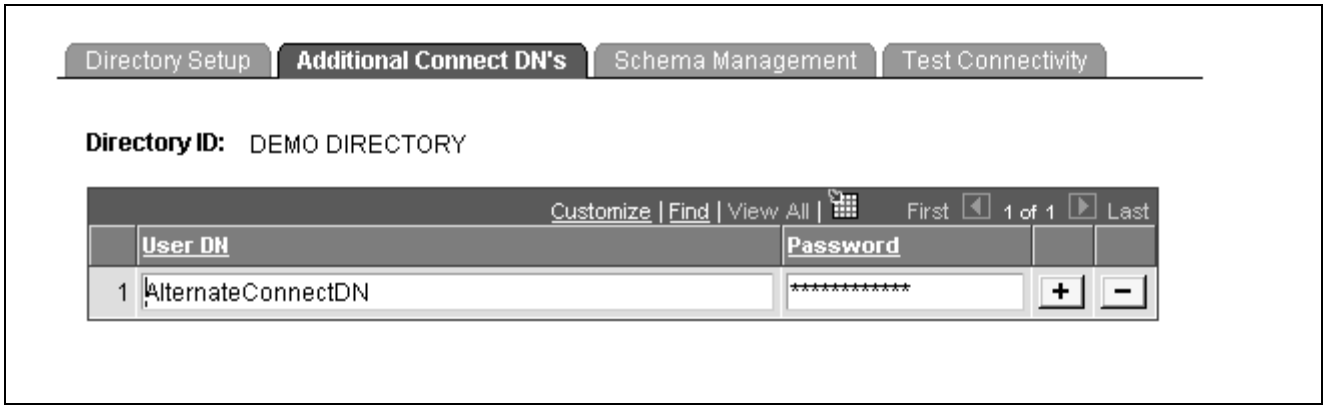
## Configuring the Directory Connection

Access the Directory Setup page.



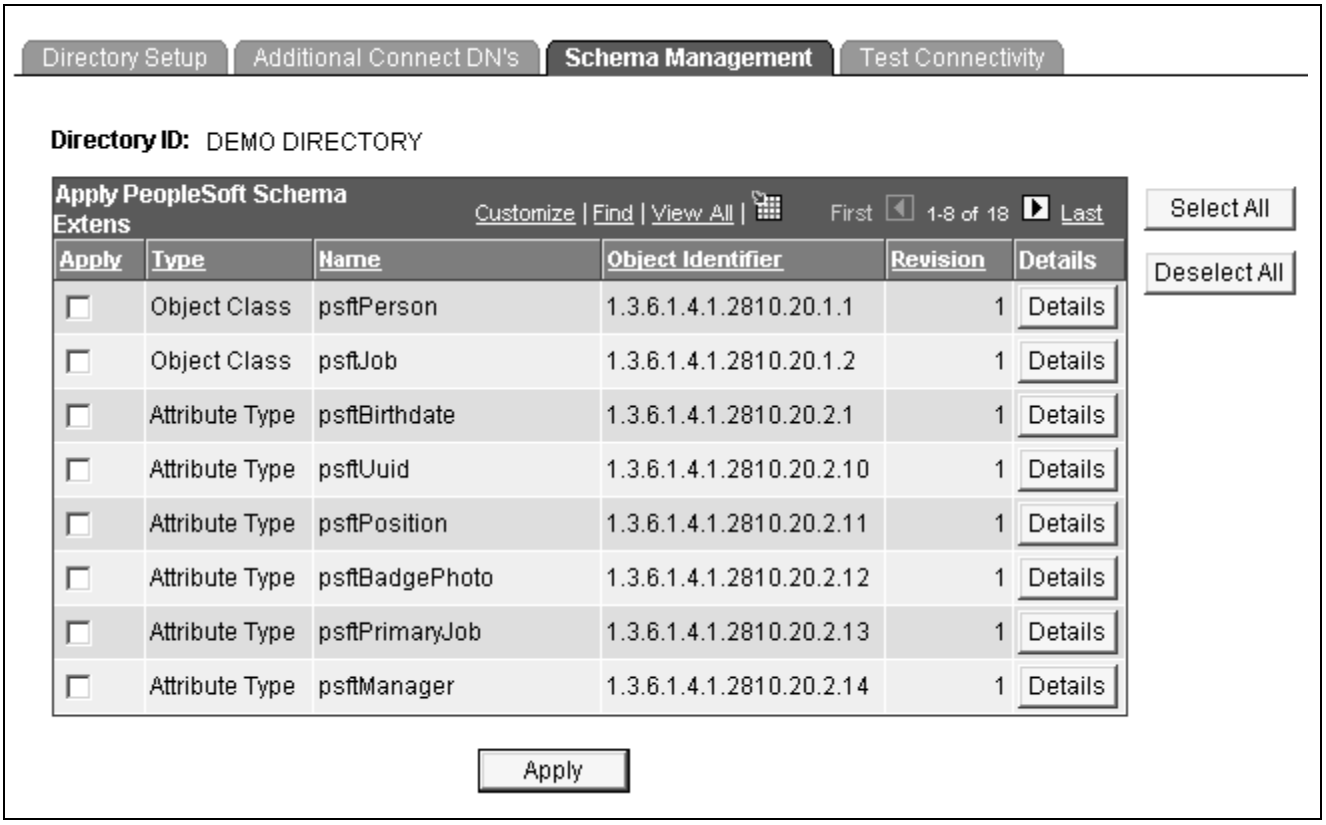

Directory Setup

Access the Additional Connect DNs page and add more Connect DNs and password, if needed.

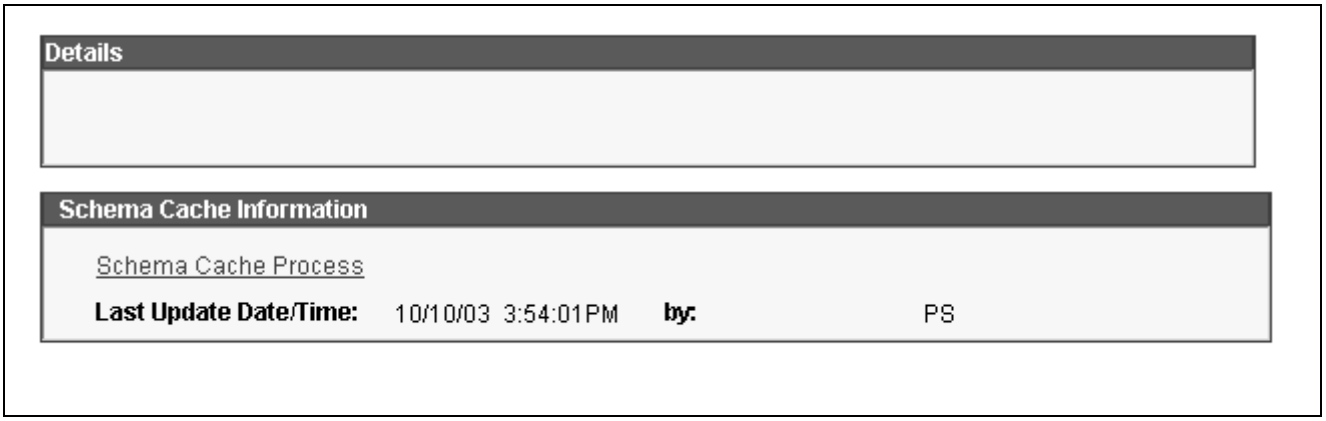



Additional Connect DNs

Access the Schema Management page.




Schema Management page

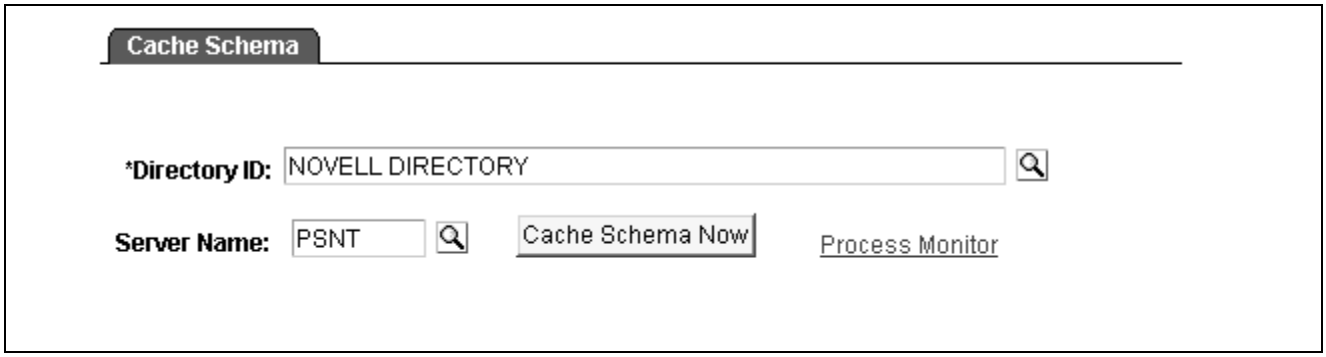



Schema Management page (cont.)

Activate the checkboxes of those object classes or attribute types that you want applied to the cache schema.

## Caching the Schema

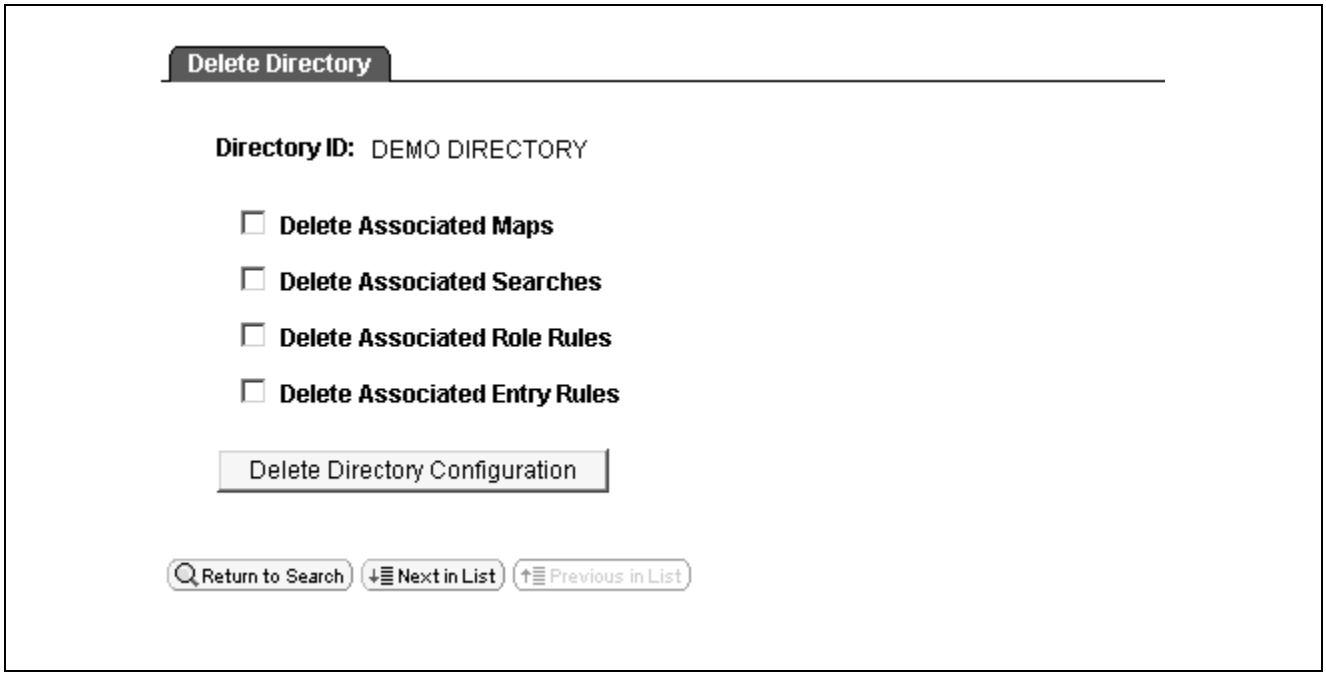
Access the Schema Caching page. Enter the Directory ID and Server Name of the schema to be cached and click Cache Schema Now.




Cache Schema page

## Deleting the Directory Configuration

Access the Delete Directory page. Select the checkboxes for the desired directory configuration deletions.




Delete Directory page

# (Optional) Setting Up Directory Authentication

For information on setting up authentication servers, user profile maps, and role membership rules, refer to the following documentation.

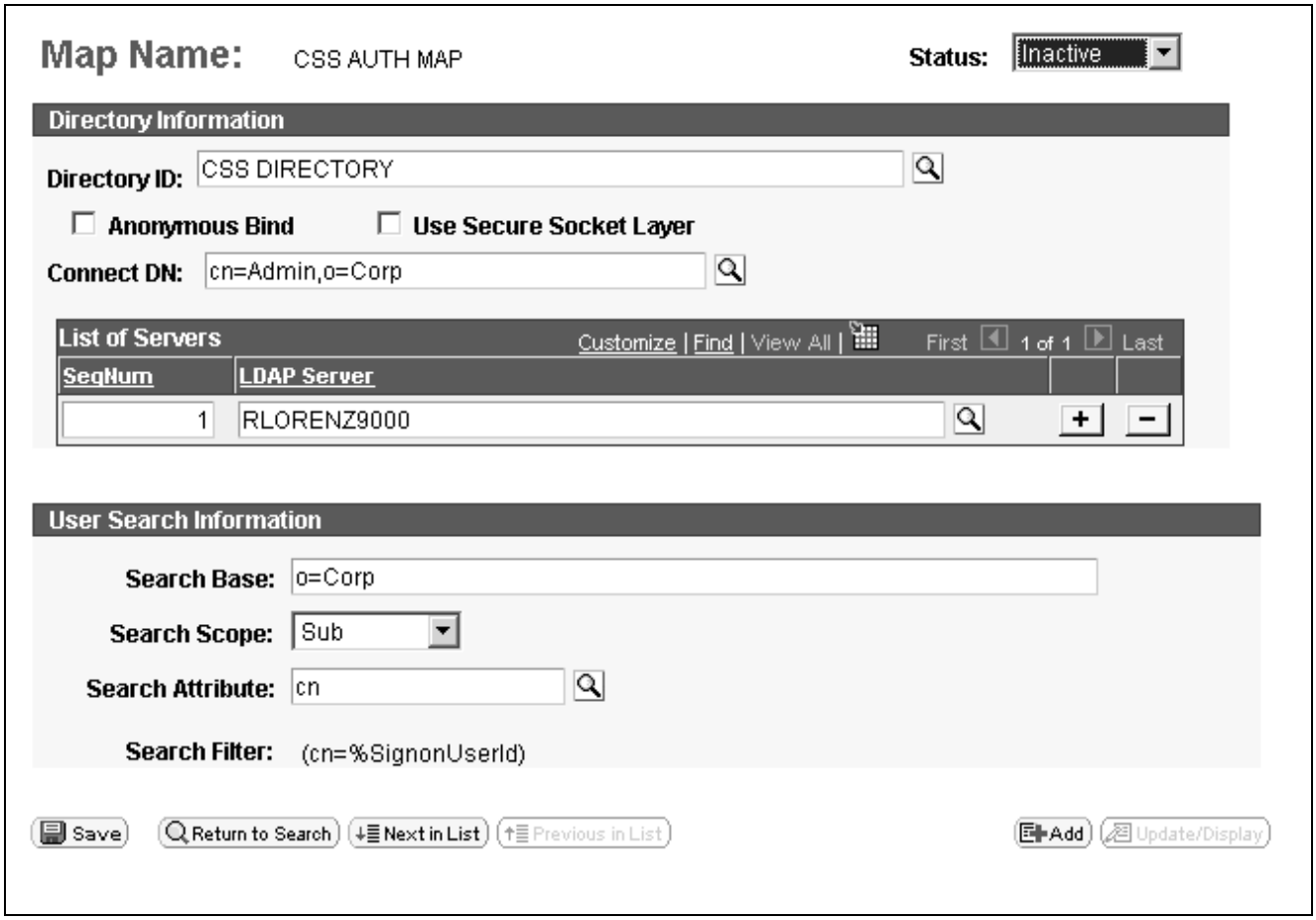
See *PeopleSoft PeopleTools 8.44 PeopleBook: PeopleTools Security*

## Pages Used to Set Up Directory Authentication

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Authentication (Map Name) | DSSECMAPMAIN | Enterprise Components, Directory Interface, Mappings, Authentication | Create a mapping for the directory that the system relies on for authenticating users. |
| Mandatory User Properties | DSUSRPRFLMANMAP | Enterprise Components, Directory Interface, Mappings, User Profiles, Mandatory User Properties | Specify the attributes required for sign-in. You can have the system retrieve these mandatory values from the directory server, or you can enter default values. |
| Optional User Properties | DSUSRPRFLOPTMAP | Enterprise Components, Directory Interface, Mappings, User Profiles, Optional User Properties | Specify optional user properties to store in and retrieve from the directory. You can specify general, permission list, and workflow attributes. All these attributes appear in the User Profile component. |

## Using Map Authentication

Access the Mappings Authentication page.



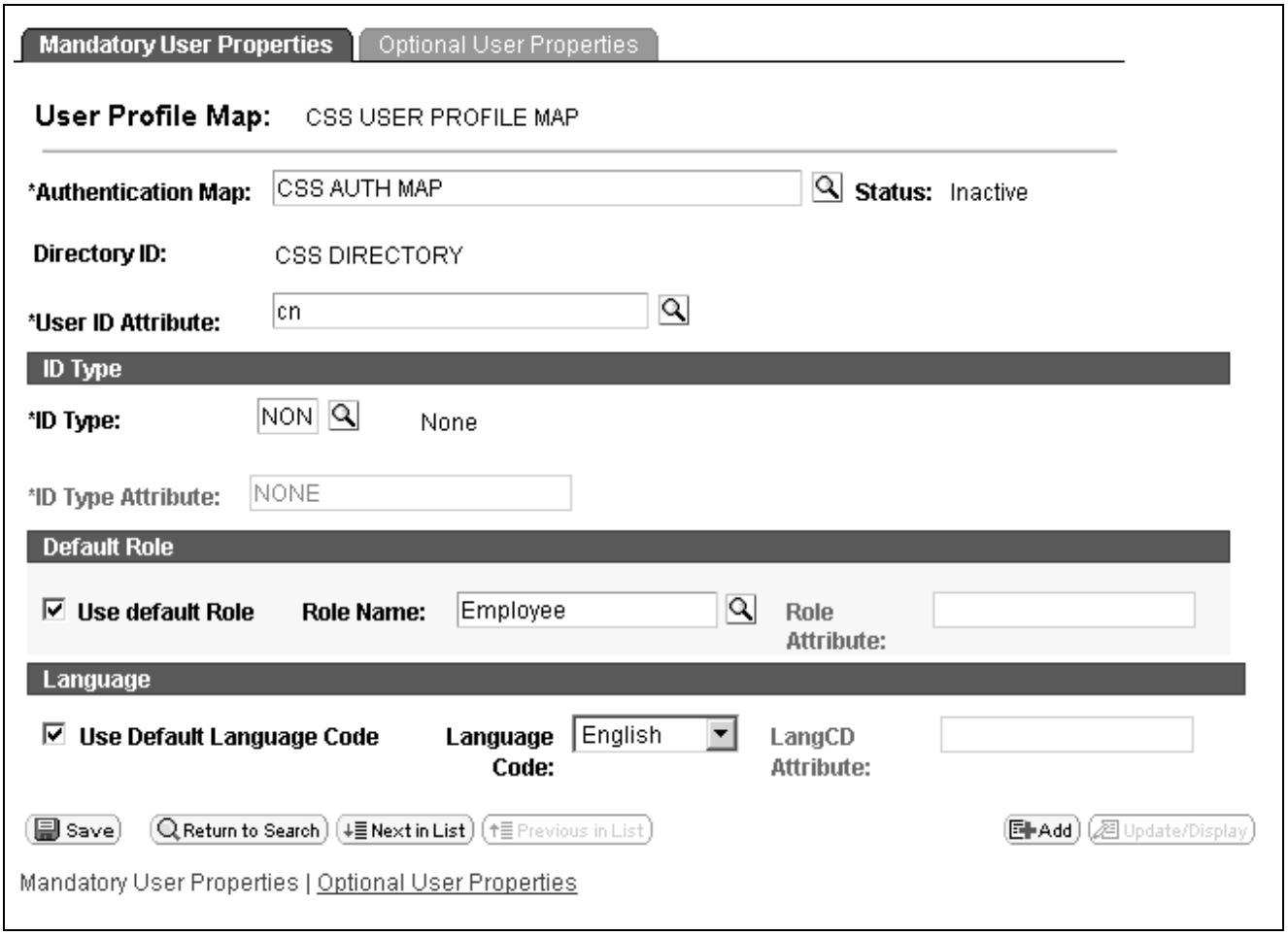

Map Authentication page

**Anonymous Bind** If directory data required for authentication and user profile maintenance is visible to an anonymous connection you may select this checkbox.

**Use Secure Socket Layer** Select this if you are using SSL between PeopleSoft and the directory server.

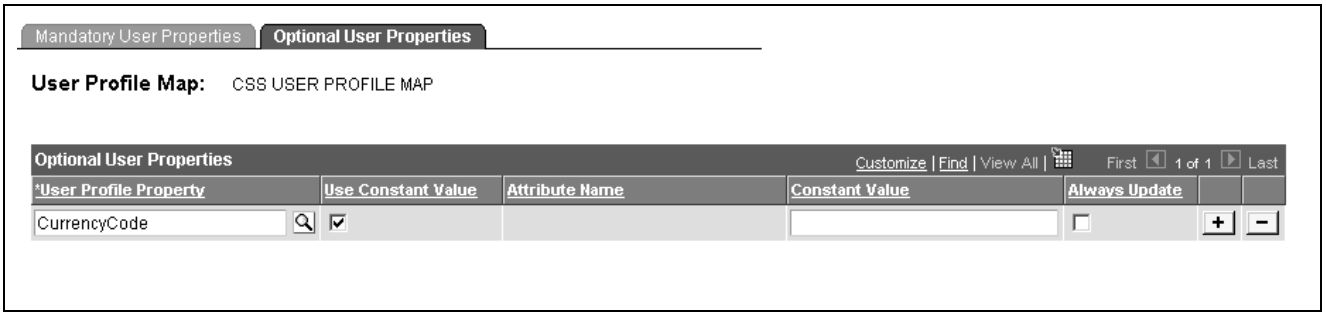
## Viewing User Properties

Access the Mandatory or Optional User Properties Page.




Mandatory User Properties page

Select the Authentication Map and checkboxes and fields as needed.




Optional User Properties page

# Setting Up Mappings

This section provides an overview of mapping and discusses how to:

- Create mappings.
- Modify a distinguished name.

- Specify distinguished name details.
- Translate or perform functions with database values.
- Map PeopleSoft data to directory object class attributes.
- Locate delivered messages.

## Understanding Mappings

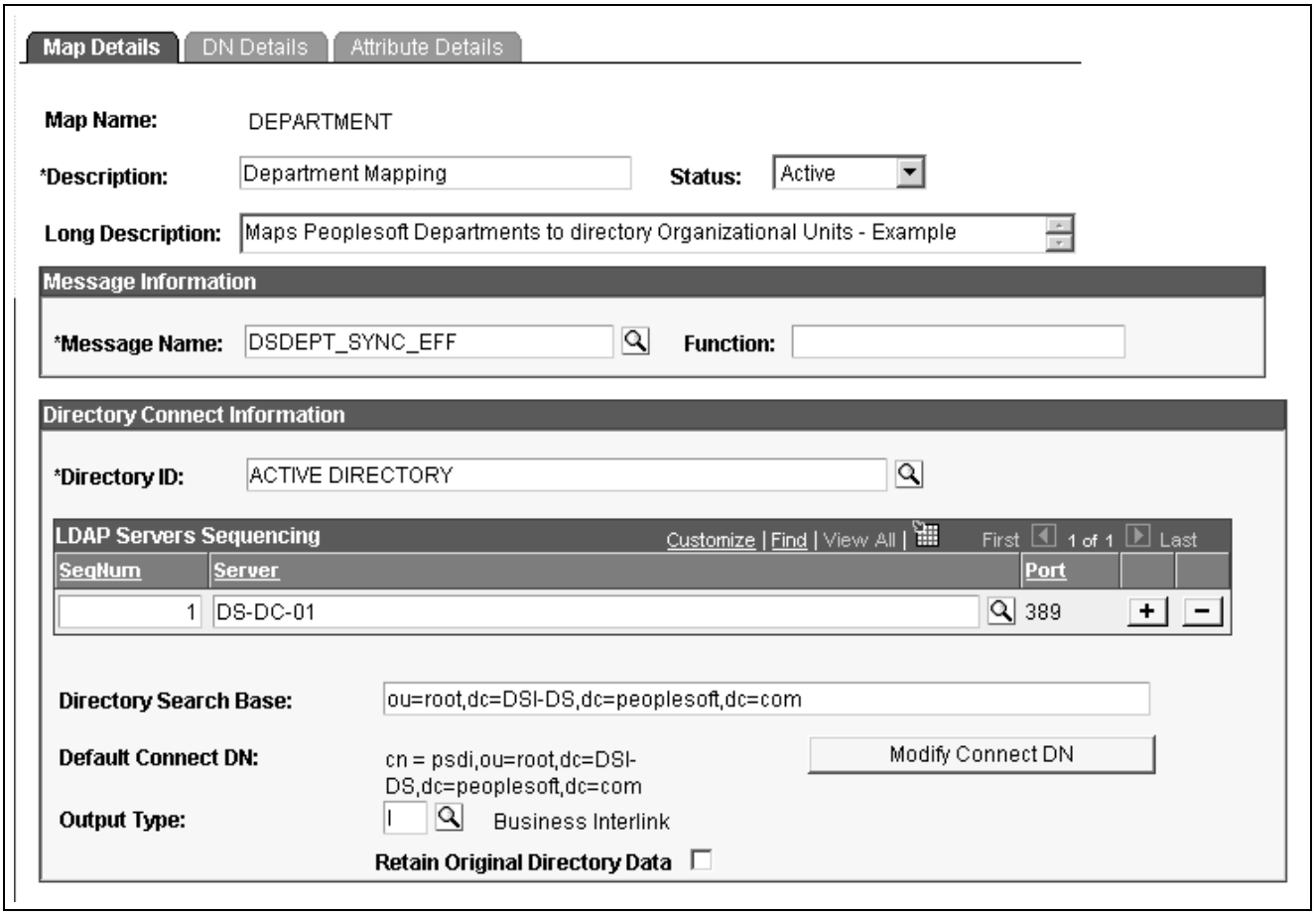
You map PeopleSoft data to the equivalent directory objects to keep the data synchronized. PeopleSoft Directory Interface receives PeopleSoft data from application messages published whenever there's a business event associated with the messages identified in the Directory Mapping component. Each message contains information about records and the most recent data for the record fields. Using the mapping information that you set up, PeopleSoft Directory Interface associates the fields in the message to the attributes in the directory and then updates the selected directory attributes with the field data from the message.

## Pages Used to Set Up Mappings

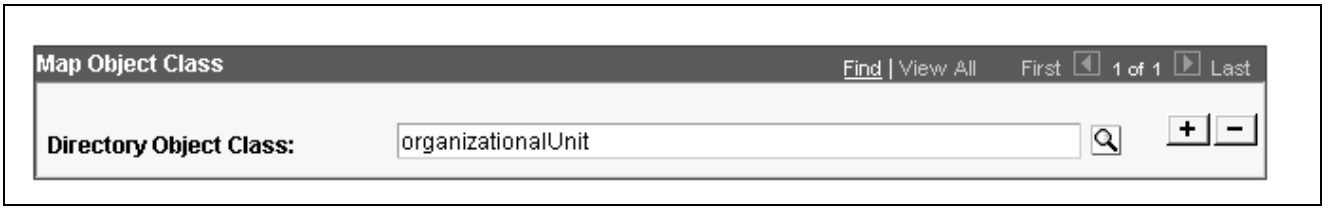
| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Map Details | EO_DSMAP | Enterprise Components, Directory Interface, Mappings, Directory, Map Details | Set up a mapping and enter the data relationship details between PeopleSoft data and directory data. |
| Directory Interface | EO_DSUSERDN | Enterprise Components, Directory Interface, Mappings, Directory, Map Details, Modify Connect DN | Allows you to modify the Connect DN. |
| DN Details | EO_DSDN | Enterprise Components, Directory Interface, Mappings, Directory, DN Details | Set up the relationship between the data contained in the message that you selected on the Message Details page and the directory entry's distinguished name. |
| Attribute Details | EO_DSATTRIB | Enterprise Components, Directory Interface, Mappings, Directory, Attribute Details | Set up the relationship between the data in the application message that you selected on the Message Details page and the directory object class attributes. |

## Creating Mappings

Access the Map Details page.




Map Details page




Map Details page (cont.)

**Status** Select the appropriate status:

- *Active:* The map is active and ready to be used.
- *Inactive:* The map is not ready to be used.
- *Remote:*Not used at this time, and may appear unavailable.

## Message Information

**Message Name** Select the message to associate with this mapping. The message contains the PeopleSoft records and fields that have the data that you want to associate with the attributes that make up the directory entry that you select in the Directory Connect Information group box. For example, if you select the output − DEPTID object class, select the department (DSDEPT_SYNC) application message because it contains the fields relevant to the department object class.

**Function** Enter the name of the PeopleCode function that you want to run using this message as an input parameter. The function can use any of the fields contained in the message to produce an output value for one or more of the fields that you map. This enables you to use a field in a function without mapping to it directly. For example, if you want the employee ID value sent to the directory to be a value combining the employee ID and the salary code, enter a function that produces that value. You then only have to map to the EmplID field to insert the derived employee ID in the directory.

### Directory Connect Information

**SeqNum** (directory sequence number) Indicate the order in which the server should be used when the system processes this mapping. If the first server is unavailable, the system attempts to access the other servers in sequence until it finds an available one. If you are using multiple servers, this enables you to distribute the load across servers.

**Directory Search Base** Enter a directory search base. The search base is the entry in the directory information tree from which the system begins a search relating to this mapping. For example, if, on the Attribute Details page, you select to have a field value updated indirectly, PeopleSoft Directory Interface searches for and updates all instances of that field in entries from that point in the information tree down.

**Modify Connect DN** (modify connect distinguished name) Click to access the Map Details page.

**Output Type** Select the method that the system should use to send the mapped data to the directory data. Select *I* to output data to the directory directly through a business interlink. Select *F* to output data to an LDAP Data Interchange Format (LDIF) file to be manually updated in the directory.

Use the same output type for all your mappings to keep data consistent in the directory.

**Retain Original Directory Data** When you move data in your directory using the directory interface, the directory interface copies the data to the new location and then deletes the old version. Select this check box to preserve the original data. You can select this check box at a later date provided that you do it before the data move.

**Note.** Select this check box if your directory contains binary data. Move the binary data with your directory administrative tool.

### Map Object Class

**Directory Object Class** Select one or more directory object classes. The object classes that you select determine the attributes you can map to PeopleSoft data.

## Modifying the Distinguished Name

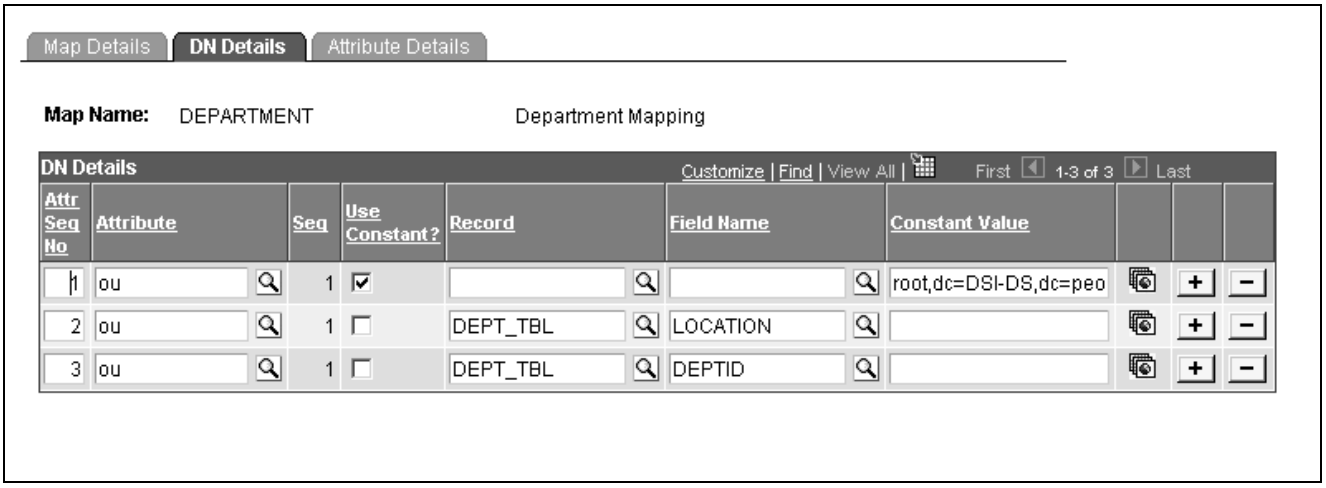
Click Modify Connect DN.

**Use Default (Admin) DN** (use default [administrative] distinguished name[yes/no])
Select to use the default connect distinguished name value that you set up in PeopleTools.

**User DN**(user distinguished name)
Displays the alternative IDs that you can use to connect to the specified directory ID. You can use a user ID (and password) other then the default one listed on the Directory Setup page in PeopleTools. Because the default user ID is most likely an administrative ID, this enables you to set up a more secure user ID for the scope of the mapping.

## Specifying Distinguished Name Details

Access the DN Details page.




DN Details page

Associate the data contained in the message that you selected on the Map Details page with the entry's distinguished name.

### DN Details

**Attr Seq No** (attribute sequence number)
The system assigns an attribute sequence number to the attributes. Some directory attribute values consist of multiple values. The attribute sequence number distinguishes between the different attribute values and indicates to PeopleSoft Directory Interface the order in which the PeopleSoft and constant values should be assigned to the attribute.

**Attribute**
Select the directory attributes associated with the mapping's distinguished name. For example, for the Department entry, map the o − Corporation first, the l − location second, and then the ou − Department attribute.

**Seq** (sequence)
Enter the sequence number of the directory attribute. The directory builds the entry's distinguished name using the attributes in sequential order.

**Use Constant** and **Constant Value**
Select to use the constant value that you enter in the Constant Value field to populate this attribute instead of a PeopleSoft field value.

**Record** and **Field Name**
Select the name of the record that contains the PeopleSoft field and the PeopleSoft field containing the value to assign to this attribute.

Click to access the Translate Table page and translate database values or instruct the system to perform functions with database values.

**Note.** Use this page when constructing distinguished names across active directory multiple domains.

### Example Entry

An entry's distinguished name is built by applying the attributes in a sequential order. The order for the department entry example would be constructed using the data in the following table:

| Sequence Number | Directory Attribute | Attribute Sequence Number | Use Constant Value | Record (Table) Name | Field Name | Constant Value |
|---|---|---|---|---|---|---|
| 1 | o | 1 | Yes | | | Corp |
| 2 | l | 1 | No | DEPT_TBL | LOCATION | |
| 3 | ou | 1 | No | DEPT_TBL | DEPTID | |

## Translating or Performing Functions With Database Values

Access the Translate Table page.

**Translate Value** Select to replace the database value with the Distinguished Name field value for the selected attribute.

**PeopleCode Function** Select to use the selected database object value as a parameter in a PeopleCode function. The system uses the resulting value as the attribute's distinguished name.

**Don't Transform value** Select to instruct the system to keep the database value as is. This option is the default value for this field.

**Database Value** Enter the database value that you want the system to replace. For example, every time the database value *Vancouver* appears in the Location attribute, the system replaces it with the distinguished name*Van.*

This field is available only when you select Translate Value as the transformation option.

**Distinguished Name** Enter the distinguished name value to replace the database value with.

This field is available only when you select Translate Value as the transformation option.

**PeopleCode Function Name** Enter the PeopleCode function that the system should use to calculate the distinguished name for the selected attribute.

This field is available only when you select PeopleCode Function as the transformation option.

## Setting Up PeopleCode Attribute-Level Functions

When the mapping function accesses the values in the selected field, the field value is passed into a PeopleCode function as a parameter and the output is assigned to the attribute in the directory.

Before you can enter a function on this page in the PeopleCode Function Name field, you must set up the function in the FUNCLIB_DS_PC.DSDYNFUNC FieldFormula.

To create a function:

1. Open the FUNCLIB_DS_PC.DSDYNFUNC FieldFormula.
2. Add a section in DSDynamicAttrFunc.
3. In the evaluate statement, add the following section for each function that you want to add (*FuncX* is equal to your function name):

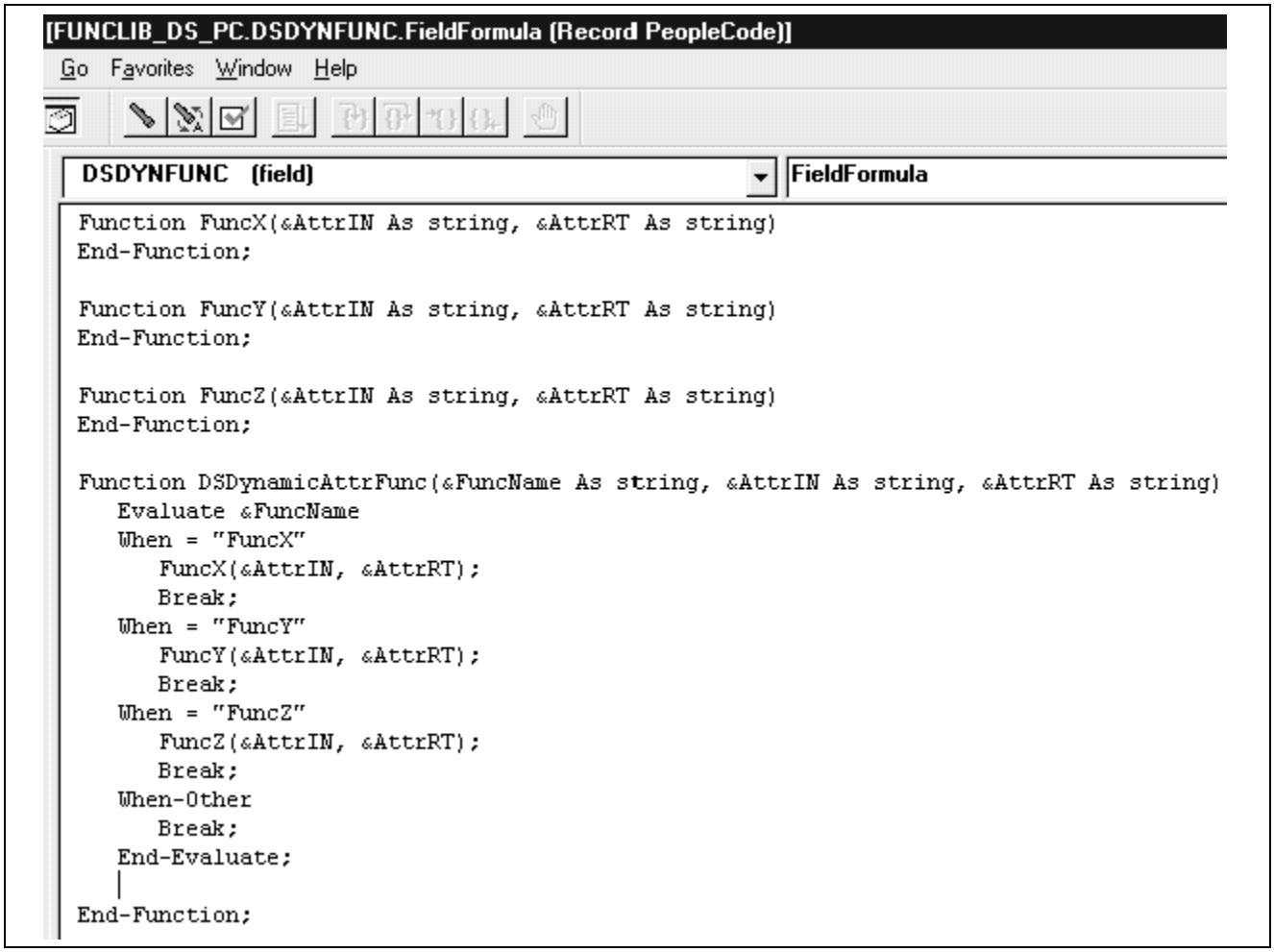
```
When = 'FuncX'
   FuncX(&AttrIn, &AttrRT);
   Break;
```

4. Define a DSDynamicAttrFunc PeopleCode function.

   The parameter list must contain two parameters, an attribute type string input and an attribute type string output.

## PeopleCode Function Example

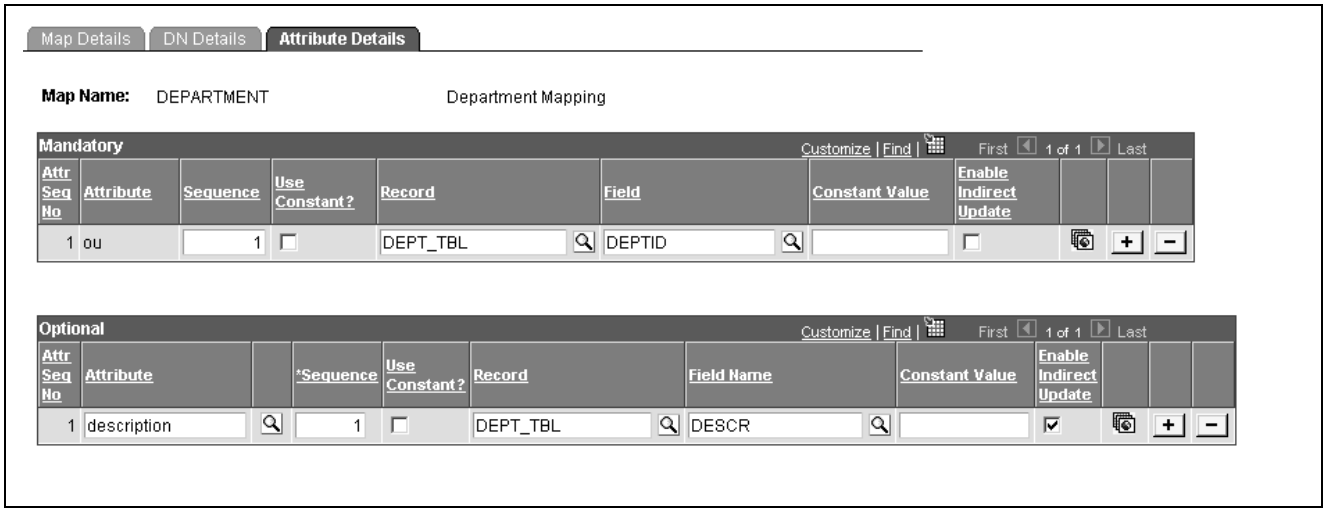
The following example displays the setup for functions FuncX, FuncY, and FuncZ.




Setup functions on the FUNCLIB_DS_PC.DSDYNFUNC FieldFormula

## Mapping Data to Directory Object Class Attributes

Access the Attribute Details page.




Attribute Details page

On the Attribute Details page, associate the fields contained in the message that you selected on the Map Details page with the attributes that provide more detail about an entry. Some attributes are mandatory (an object class's mandatory attributes are defined in the directory schema) and must be mapped to either a constant value or record or field. For the department example, you would map PeopleSoft records and fields to the mandatory attributes (such as DeptID), and you could add additional attributes that would give you more information about the object class, such as description.

**Note.** The system does not update related-display field values unless the source field is also mapped. If the source field is not mapped, the audit process still indicates and enables you to update any discrepancies. For example, when you map to an employee's job code, the directory entry also includes the job code description. If you change the job code description on the Job Code component, the system updates the related-display description field on the employee's Job Data page, but it does not update to the directory, because it is not included in the mapping.

***Warning!*** The fields that you map to mandatory attributes must contain data or the mapping will fail. You can guarantee that there will be data in the fields by mapping mandatory attributes to required fields.

**Attr Seq No** (attribute sequence number) — Displays the attribute sequence number assigned to this attribute.

**Attribute** — In the Mandatory scroll area, the system displays the mandatory attributes for this object class.

In the Optional scroll area, select optional attributes.

**Seq** (sequence number) — Enter a sequence number for this attribute. Some directory attribute values are made up of multiple values. The attribute sequence number distinguishes between the different attribute values and indicates to PeopleSoft Directory Interface the order in which the PeopleSoft and constant values should be assigned to the attribute.

**Ind Upd** (indirect update) — Select if the field that you selected is used as an attribute in the directory outside of this mapping and you want it to be updated when this field is updated. The system only updates attributes in entries at lower levels on the directory information tree than this entry.

## Locating Delivered Messages

Your PeopleSoft application that supports the PeopleSoft Directory Interface delivers a set of messages to be used share information with your directory service.

For information on these messages and how they work in conjunction with the PeopleSoft Directory Interface, see your PeopleSoft application documentation.

# (Optional) Setting Up Entry Membership Rules

Entry membership rules enable you to modify a directory entry, such as a group, based on criteria stored in the PeopleSoft database. This section discusses how to:

- Create entry definitions.
- Specify entry membership rules.

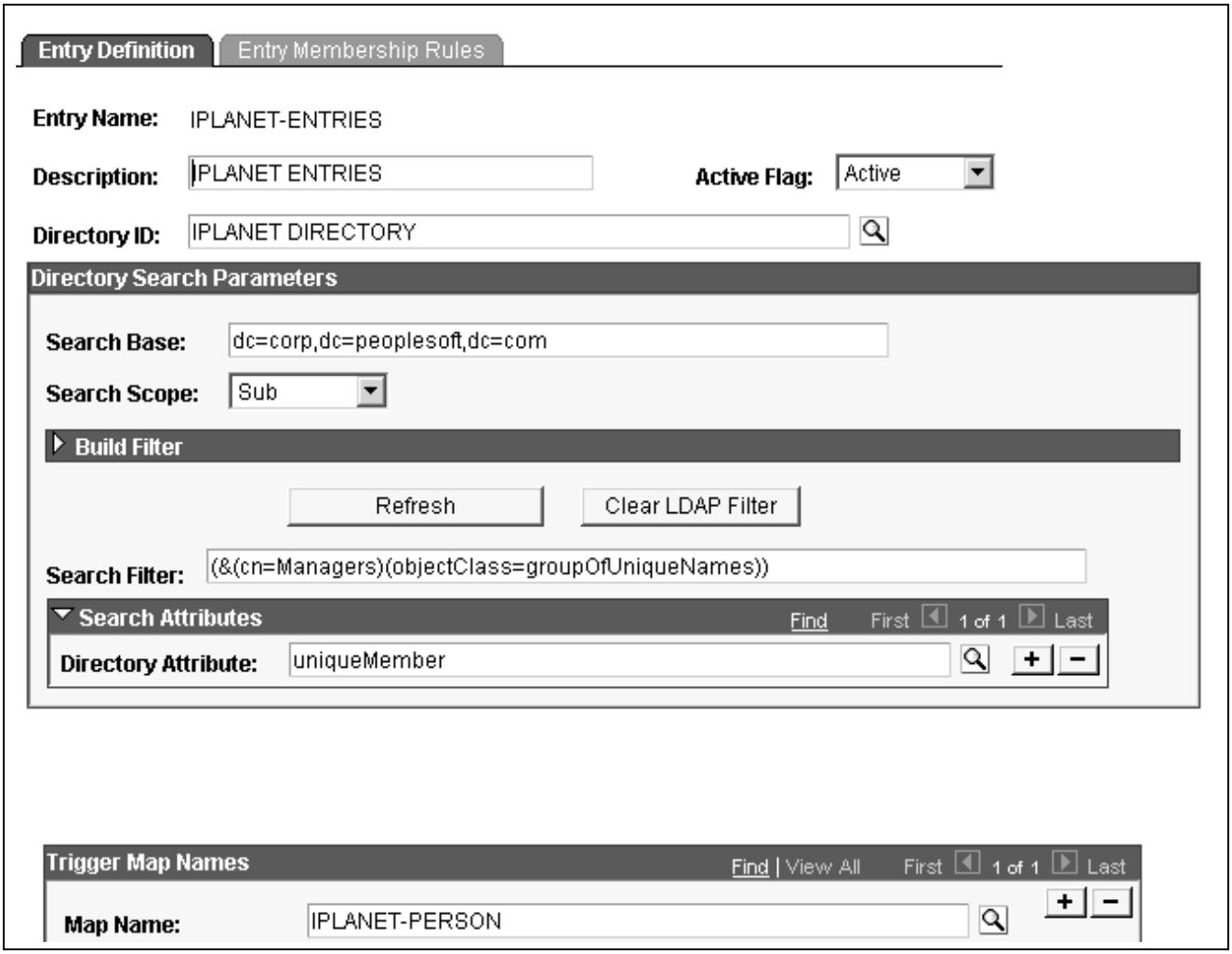
This feature provides a method to match any type of directory entry to rules that are meaningful in PeopleSoft. You can use membership rules to create any type of logical grouping in the directory. The groupings are not restricted to security purposes.

## Pages Used to Set Up Entry Membership Rules

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Entry Definition | EO_DSCONTAINERDEFN | Enterprise Components, Directory Interface, Membership Rules, Entries, Entry Definition | Create a directory entry definition. |
| Entry Membership Rules | EO_DSSECRULES | Enterprise Components, Directory Interface, Membership Rules, Entries, Entry Membership Rules | Establish entry membership rules. |

## Creating Entry Definitions

Access the Entry Definition page.




Entry Definition page

**Entry Name** Displays the entry name that you entered on the search page. The system uses this value for the entry name throughout the application, so it must be the name of an existing entry in the external directory. PeopleSoft assumes that the name is unique in the directory.

**Active Flag** Select to enable rules. Rules that aren't active do not run.

## Directory Search Parameters

**Search Base** Enter the distinguished name of the base under which this entry will be located in the directory. The application performs an LDAP search to retrieve the distinguished name of the entry using this field as the base.

**Search Scope** Select from:

*Base:* The query searches only the value in the Search Base field.

*One:* The query searches only the entries one level down from the value in the Search Base field.

*Sub:* The query searches the value in the Search Base field and all entries beneath it.

### Build Filter

**( )**(parentheses) Select the check boxes below the parentheses to group expressions. You can group more than one line together using the check box on the left for the first line and the check box on the right for the last line.

**Attribute** Enter the name of the attribute that will store the members of the entry in the external directory. It is typically set to *member,* but the attribute name could be anything that you choose.

**Operation** Assign an operator to your rule such as <, <=, <>, =, >, or >=.

**Value** Assign a value to the to the attribute in your rule.

**And/Or** To add another line to your rule, select *AND* or *OR* depending on your rule logic. Select *END* to signify the end of the search. Select *NONE* if you are not using this kind of filter.

**Refresh** After you make changes using the Build Filter options, click this button to update the Search Filter edit box to reflect the changes.

**Clear LDAP Filter** Click this button to delete all values from the Search Filter edit box and the Build Filter selections.

**Search Filter** Displays the filter that the system applies to the search for the distinguished name of the defined entry. Typically displays the directory object class of the entry in the form “objectclass = GroupOfUniqueNames”, for example. This indicates what type of entry to search. To retrieve the correct entry distinguished names, the system adds the name of the entry to the search filter at runtime. The name retrieved by the LDAP search using this filter is tied to the rules defined in the Entry Membership Rules page. When these rules run, the employee that the system is currently processing is either added to or deleted from the distinguished name retrieved by the search.

### Search Attributes

**Directory Attribute** Select the attribute of the entry being defined that will contain all the members of this entry. This attribute must be valid for the current entry in the directory. The employees that satisfy the entry membership rules of this entry are added under this entry as a new value of this attribute. Because of this, there will be as many attribute values as there are employees satisfying the entry membership rules. If this field is left blank, the application uses *member* as a default attribute name.

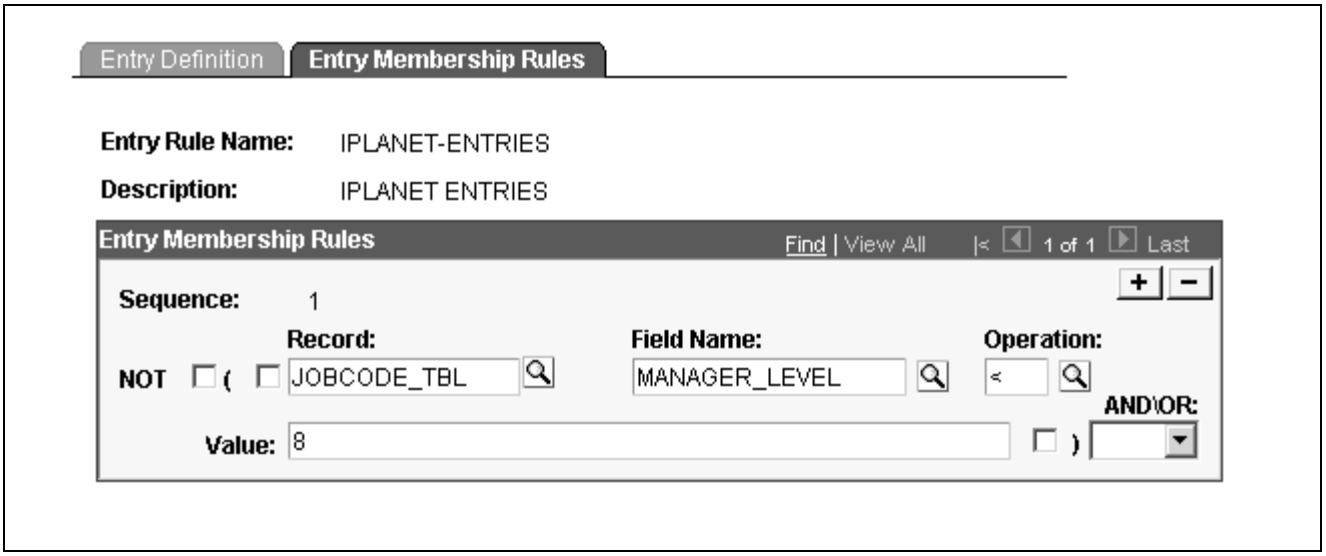
### Trigger Message Names

**Map Names** Select the names of the maps to associate with the entry definition. Besides being a security feature, this also improves performance at runtime, because only applicable rules are evaluated.

**Note.** Run the directory audit if an entry rule has changed or if you want to initialize the directory entries.

## Specifying Entry Membership Rules

Access the Entry Membership Rules page.




Entry Membership Rules page

### Entry Membership Rules

**Sequence** — Displays the sequence of a rule within a rule set. The sequence becomes significant when you enter more than one rule.

**NOT** — Select to negate the rule that you enter. This is similar to using the symbol *!* to reverse the truth value of an operand.

**( )**(parentheses) — Select the check boxes to add parentheses around your rule. You can group more than one line together using the check box on the left for the first line and the check box on the right for the last line.

**Record** and **Field Name** — Enter the name of the PeopleSoft record and field containing the information to be tested.

**Operation** — Enter the appropriate operator, such as < , <= , <> , = , > , or >=.

**Value** — Enter the value that the employee's data needs to be tested on. This can be any value of the same type as the field used in the rule, such as String, number, date, and so on.

**AND/OR** — To add another line to your rule, select *AND* or *OR* depending on your rule logic. Select *END* to signify the end of the search. Select *NONE* if you are not using this kind of filter.

The entry rules are logical expressions that can be either true or false. They are composed of filters on database objects associated by logical operators. Rules have the following form:

```
[NOT] [ ( ]  Record . Field   operatorConstant [ ) ] [AND/OR]
```

The symbols between square brackets are optional. The operator can be <, <=, <>, =, >, or >=. A rule set is composed of single rules joined by AND or OR Boolean operators if necessary. The following example shows a series of single rules joined to make one compound rule.

( JOB.LOCATION = ‘KC004’ AND [1]

JOB.COMPRATE > 15000 ) OR [2]

NOT JOB.DEPTID = ‘GBIY004’[3]

**Note.** There are no limits to the number of rules used within a rule set.

# Loading Data into the Directory

This section provides an overview of loading the directory and discusses how to load the directory with PeopleSoft data.

## Understanding Directory Load Behavior

Use the Directory Load process when there is no existing data in the directory. The process overwrites any data in the directory.

If you have data in your directory, use the Directory Audit process instead of the Directory Load process. The audit process compares the PeopleSoft data to your existing directory data and enables you to review and resolve any possible conflicts.

**Note.** For HRMS customers only, there is an alternative process named DSMAPINPUT FullSync that you can use in place of the Directory Load process. This new process does not replace the Directory Load process; it is provided as an alternative to load the data if performance becomes an issue.

See *PeopleSoft HRMS PeopleBooks 8.8: Application Fundamentals*

## Pages Used to Load PeopleSoft Data into the Directory

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Directory Load | EO_RUNCTL_DS_LOAD | Enterprise Components, Directory Interface, Load Directory | Run the Directory Load process. |

## Loading the Directory with PeopleSoft Data

Access the Directory Load page.


Directory Load page

**LDIF File** Select to have the process send the data to an LDIF file for you to load in the directory.

**Direct Update** Select to have the process directly update the directory.

**Run** Click to run the process using PeopleSoft Process Scheduler.

# CHAPTER 3

# Reviewing Directory Data and Generating Reports

This chapter discusses how to:

- Review LDAP directory data.
- View PeopleSoft Directory Interface reports.

## Reviewing LDAP Directory Data

This section discusses how to:

- Run a directory audit.
- Run a directory search.

### Pages Used to Review Directory Data

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Directory Audit | EO_RUNCTL_DS_AUDIT | Enterprise Components, Directory Interface, Transaction Audit History, Run Directory Audit | Run the Directory Audit process. |
| Directory Search | EO_DSSRCHDIRECTORY | Enterprise Components, Directory Interface, Transaction Audit History, Search Directory | Define search parameters to query the directory. The page saves the search parameters for future use. |

### Running a Directory Audit

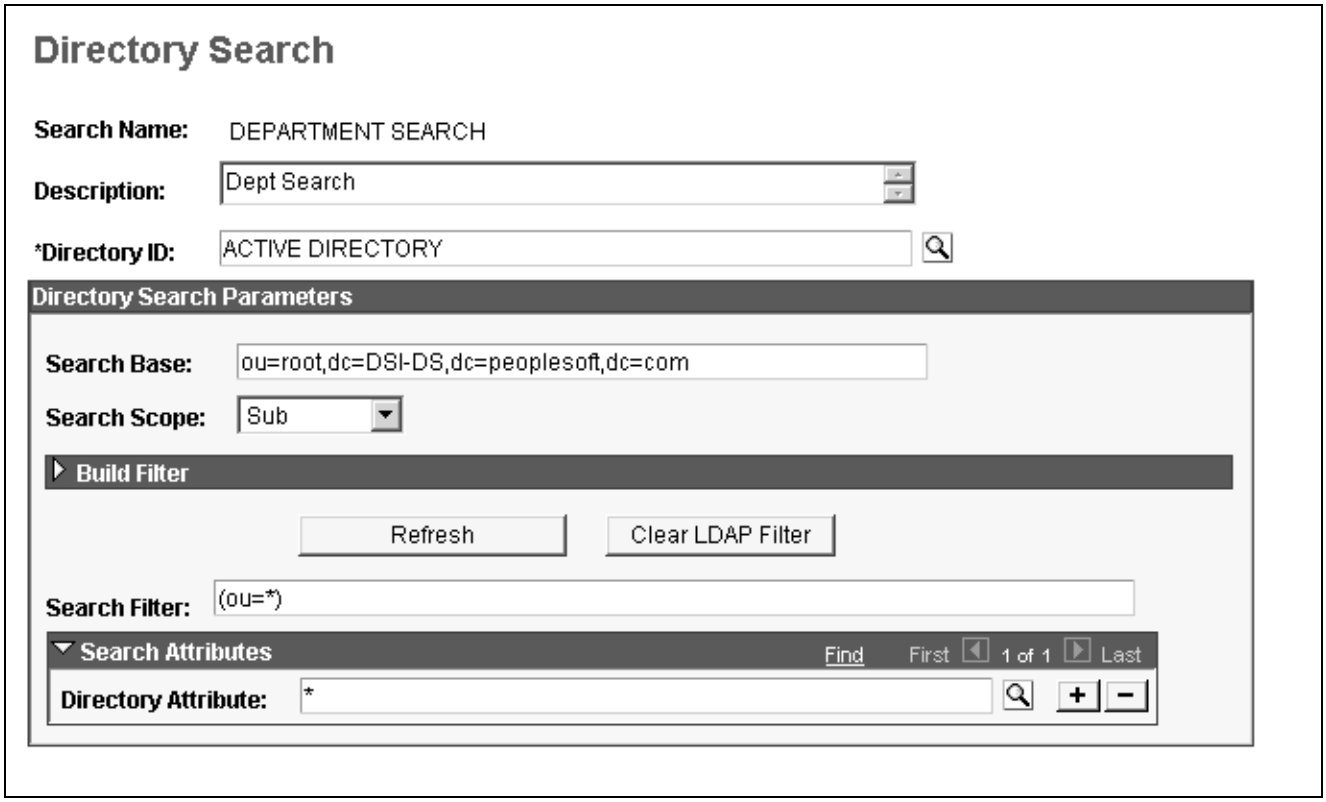
Access the Directory Audit page.

Use the Directory Audit process to ensure that your directory database has the same data as your PeopleSoft database. The Directory Audit process compares the data in the directory to the data in the PeopleSoft database identified in the selected map and creates an LDAP Data Interchange Format (LDIF) file containing any discrepancies using PeopleSoft as the authority. You can then use the LDIF file to update the directory.

**Map Name** Select the name of the map that the audit should be run against.

### Running a Directory Search

Access the Directory Search page.




Directory Audit Search page

Use the Directory Search page to define search parameters to query the directory and view the results. Search results are displayed on the Search Results page as they appear in the directory.

**Search Name** Enter the search name. The system saves the search parameters that you enter on this page and stores them under this name for future use.

## Directory Search Parameters

**Search Base** Select the directory entry that is the search base for this search. The search base is the entry in the directory information tree at which the search begins querying.

**Search Scope** Select from:

*Base:* The query searches only the value in the Search Base field.

*One:* The query searches only the entries one level down from the value in the Search Base field.

*Sub:* The query searches the value in the Search Base field and all entries beneath it.

## Build Filter

Use the fields in the Construct Filter group box to create an attribute-specific filter. For example, if you want data on a single person, enter the attribute name *Person*, the operation =, and enter the person's name in the Value field. You can construct multiple filters.

**( )**(parentheses) Use parentheses to separate filter statements.

**Attribute Name** Enter the name of the attribute whose data you want to filter.

**Operation** Select an operator to determine how to filter the data.

**Value** Select a value to compare against when filtering.

**And/Or** Select from:

- *And:*A search must meet the criteria of multiple statements.
- *END:* This value marks the end of a query. This value applies parentheses appropriately.
- *NoOp:*No operator is used.
- *OR:*A search must meet the criteria of one of multiple statements.

**Search Filter** You can narrow the search (for example, instruct the system to search for all attributes but one) by entering a search filter. Enter the search filter using standard LDAP protocol.

You can specify one or more attributes to search for. To search all attributes, enter an asterisk ( * ).

For more information about the LDAP protocol, see your directory documentation.

### Search Attributes

**Directory Attribute** Select the attributes to search for.

Leave this field blank to search all attributes.

# Viewing PeopleSoft Directory Interface Reports

This section discusses how to:

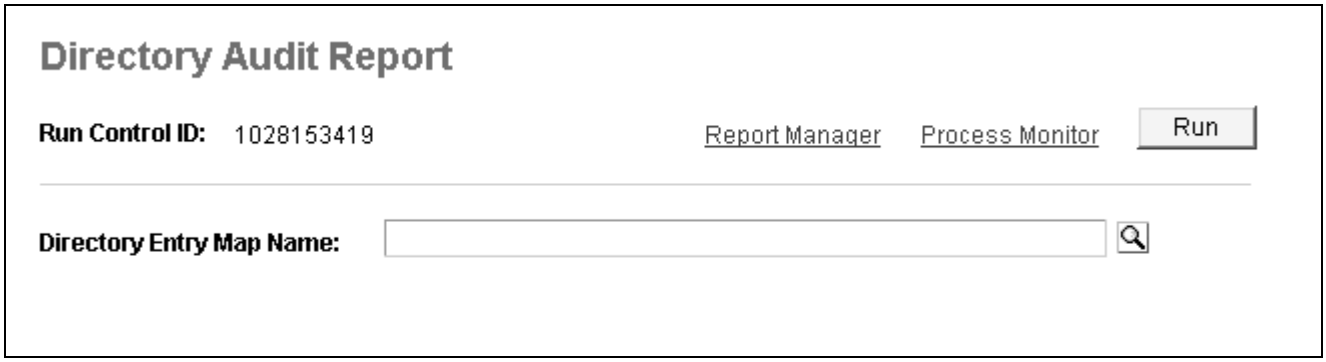
- View the directory audit report.
- View the business interlink status report.

## Pages Used to View PeopleSoft Directory Interface Reports

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Directory Audit Report | EO_RUN_DS_AUD_RPT | Enterprise Components, Directory Interface, Audit Report | Generate the Directory Audit report. |
| Directory BI Status Report | EO_RUN_DS_BI_RPT | Enterprise Components, Directory Interface, Business Interlink Status Rpt | Generate the Directory BI Status report. |

## Viewing the Directory Audit Report

Access the Dir Audit Report page.




Dir Audit Report page

The Directory Audit SQR report (EO_DS001) locates and reports discrepancies between the PeopleSoft database and your directory. Before you can run the Directory Audit report, run the Directory Audit process. The Directory Audit process populates a comparison record containing the data that differs between the PeopleSoft database and the directory and creates an LDIF file with this data that can be used to update the directory. The Directory Audit report is based on this record, so you can verify what will be updated in the directory if you apply the LDIF file.

The report generates the following errors:

- 1: The distinguished name is not found in the directory.
- 2: The distinguished name is not found in PeopleSoft.
- 3: The attribute is in PeopleSoft but not in the directory.
- 4: The attribute is in the directory but not in PeopleSoft.
- 5: The value is in PeopleSoft but not in the directory.
- 6 : The value is in the directory but not in PeopleSoft.

## Viewing the Business Interlink Status Report

Access the Directory BI Status Report page.


Directory BI Status Report page

If you selected an output type of Business Interlinks when setting up maps to associate PeopleSoft fields to directory attributes, the system uses PeopleSoft Business Interlinks to modify the directory. If errors are produced as a result of the interlinks, the system writes the errors to an error record. The Business Interlink Status SQR report (EO_DS002) retrieves and presents the data contained in this error record.

| | |
|---|---|
| **Delete History Error Rows for MAP** | Select to delete historical error rows for this map after reporting them. The PS_EO_BILOAD_ERR record retains error data for this map until you run the report with this check box selected. |
| **Run** | Click Run to run the report using PeopleSoft Process Scheduler. |

## See Also

*PeopleSoft PeopleTools 8.44 PeopleBook: Process Scheduler*

# PART 3

# Common Objects and Components

# CHAPTER 4

# Defining and Storing PeopleSoft Market Rates

This chapter provides an overview of the PeopleSoft approach to market rates and discusses how to:

- Set up and maintain currency quotations.
- Define currency quotations.
- Calculate currency rates.

## Understanding the PeopleSoft Approach to Market Rates

PeopleSoft offers a core set of objects (fields, tables, work records, pages, and PeopleCode functions); and a recommended set of standard techniques and formulas, to support a common approach to defining and storing market rates and for converting currency throughout PeopleSoft applications. This section discusses:

- Market rate types, data, indices and definitions.
- Currency quotation methods.
- Calculated reciprocal rates.
- Quotation units.
- Triangulation.
- Tolerance checking.

### Market Rate Types

The Market Rate Type table (RT_TYPE_TBL) is an edit table that stores data about market rate types: that is, subcategories within a market rate index. For example, some common types of exchange rates are official rate, spot rate, and free market rate.

### Market Rate Data

The Market Rate Data table (RT_RATE_TBL) is the common repository for all types of exchange rates, including exchange rates, interest rates, and market indexes.

For market rate indexes such as interest rates, FROM_CUR and TO_CUR may have the same value.

During online maintenance of market rates, you don't view or change RATE_MULT and RATE_DIV directly, but instead access the visual rate, which is calculated by the page logic based on RATE_MULT, RATE_DIV, and the quotation method defined for the currency pair. The visual rate is stored temporarily on a page work record.

When maintaining triangulated rates, you can access a primary visual rate (which is typically, but not necessarily, the cross rate); or you can access all three visual rates via a secondary detail page.

## Market Rate Indices

Market rate indices, defined on the Market Rate Index table (RT_INDEX_TBL), provide the highest level of organization for market rates in the PeopleSoft system. These indices define general categories of market rates, such as exchange rates, interest rates, and commodity exchange rates.

You can define one index as the default exchange rate index on the Market Rate Index page. The logic in this page ensures that only one exchange rate index can be defined as the default exchange rate index.

---

**Note.** The Market Rates Index page does not ensure that a default index has been defined, but it is obvious if no default is defined, because the Market Rate Default view does not return any data.

---

## Market Rate Definitions

Tolerances for market rates are defined on the Market Rate Definition table. This stores data that defines, for each market rate, the maximum degree of variance that is accepted when a user maintains or overrides a market rate. The Error Type setting determines how the system responds when the maximum variance is exceeded. The following screen shot shows the record definition for the Market Rate Definition table:

For interest rates, the Market Rate Definition table also stores an interest basis. Possible interest basis values are:

- 30/360
- 30E/360
- Actual/360
- Actual/365
- Actual/Actual

These values are maintained in the Translate table for the INT_BASIS field.

### Market Rate Definition Default View

The Market Rate Definition Default view (RT_DEF_DFLT_VW) selects rows from the Market Rate Definition table that have a term of zero and index defined as the default exchange rate index.

## Currency Quotation Methods

A currency quotation method, defined for an exchange rate, stores data that determines how the application interprets a visual rate entered by a user (or multiple visual rates, in the case of triangulated exchange rates) into the RATE_MULT and RATE_DIV values stored on the Market Rate Data table. Conversely, it also determines how the stored RATE_MULT and RATE_DIV values are interpreted into the visual rate displayed to the user. The quotation method can be direct or indirect, and it can be non-triangulated or a triangulated conversion via a third reference currency. The currency quotation method also determines the quotation units of the From currency.

It is not necessary to define a currency quotation method for every exchange rate. If, during maintenance of market rates, no quotation method is found for an exchange rate, the page logic assumes these defaults:

- The exchange rate is direct.

- The quotation units are equal to 1.
- The exchange rate is not triangulated.

---

**Note.** This use of default values supports backward compatibility with previous exchange rate data, including calculated reciprocal rates, if your implementation requires them.

---

## Calculated Reciprocal Rates

Calculated reciprocal rates enable an accounting system to use the direct quotation method for all currency conversions by calculating a reciprocal rate to use in the conversion formula when the From and To currencies are reversed. For example, the calculated reciprocal of 1 USD equals 5.8 FRF would be 1 FRF equals 0.172414 USD: in each case, you can multiply the From currency by the rate to get the equivalent amount in the To currency.

Use of calculated reciprocal rates introduces rounding errors and presents a user interface that is inconsistent with common business practice. PeopleSoft support for indirect and direct quotation methods allows applications to eliminate use of calculated reciprocal rates by using a single rate by which you either divide or multiply, depending on whether the conversion method is direct or indirect.

## Quotation Units

PeopleSoft supports quotation units, sometimes called scaling factors, which are often used to preserve more decimal precision. For example, the exchange rate between Swiss francs (CHF) and Deutsche marks (DEM) may be stated as 100 CHF = 119.335 DEM instead of 1 CHF = 1.19335 DEM.

The use of quote units is common business practice in several European countries.

## Using Triangulation

Triangulation is the process by which a conversion between two currencies takes place via a third reference currency. This process may be used in hyperinflationary environments, where all conversions to the local currency are done via a stronger, more stable currency such as USD. Also, this process may be used when a country is undergoing a currency revaluation.

To support triangulation, PeopleSoft provides a means to define, for a currency pair, that it triangulates via a fixed reference currency. The actual conversion process is done in a two-step procedure, where the From currency amount is first converted to the reference currency and then to the destination currency, using the appropriate exchange rates. Supporting triangulation also affects the user interface, as there are now two, and possibly three, exchange rates that are relevant to the conversion. When viewing a triangulated conversion at the detail level, users access three visual rates:

- A rate for converting the From currency to the reference currency.
- A rate for converting the reference currency to the To currency.
- A *cross rate* indicating the rate that would be required to convert the From currency directly into the To currency.

The cross rate in a triangulated conversion is not typically maintained directly. The system enables you to maintain those non-triangulated rates that are components of the triangulated rate, then run a process to generate the triangulated exchange rate. (You can, however, override the cross rate, which causes one of the other exchange rate values to be recalculated to synchronize it with the overridden cross rate.)

For example, suppose an implementation was using triangulation to convert from USD to FRF. You would directly maintain the visual rate from the USD to euros (1.25 in the example table) and rate from euros to FRF (6.8 in the example table). You could then an application engine program process to derive the triangulated rate for converting from USD to FRF. The results are shown in the following table:

| Currency Pair | Quote Method | Quote Units | Primary Visual Rate | RATE_MULT | RATE_DIV |
|---|---|---|---|---|---|
| USD to Euro | Indirect | 1 | 1.25 | 1 | 1.25 |
| Euro to FRF | Direct | 1 | 6.8 | 6.8 | 1 |
| USD to FRF | Dir/Trng/Euro | 1 | 5.44 | 6.8 | 1.25 |

For performing the actual conversion, applications interpret the visual rates into RATE_MULT and RATE_DIV values based on the quotation method for the exchange, then use the RATE_MULT and RATE_DIV values stored in the Market Rates Data table in the currency conversion formula, either by accessing the values directly, or by calling the ConvertCurrency PeopleCode function.

---

**Note.** For information on how a specific application supports maintenance of triangulated exchange rates, see the documentation for that application.

---

## Tolerance Checking

It is common for applications to support tolerance checking (against user-specified tolerances) in all places where exchange rates can be entered or overridden. With the introduction of indirect quotation methods and quote units, tolerance checking is even more critical to ensure data entry accuracy.

## Conversion Factor Fields and the Visual Rate

Support for both direct and indirect currency quotations creates a potential for complex currency conversion formulas in applications. To avoid excess conditional logic in the conversion formula, PeopleSoft provides two fields to store the conversion factor, RATE_MULT and RATE_DIV. The rate that you enter is called the *visual rate.* This visual rate is generally stored in either RATE_MULT or RATE_DIV, based on the quote method. The quote units are stored in whichever field does not contain the visual rate. As a result, the formula for currency conversion remains consistent:

(from-currency / rate-div) × rate-mult = to-currency

This formula is also used for currency conversion in PeopleCode programs for online processing, as well as in SQR and COBOL processes.

The following table shows a few basic examples of how visual rates are transformed into RATE_MULT and RATE_DIV, according to the quote method and quote units for the currency pair:

| Currency Pair | Quote Method | Quote Units | Primary Visual Rate | RATE_MULT | RATE_DIV |
|---|---|---|---|---|---|
| USD to GBP | Indirect | 1 | 1.6 | 1 | 1.6 |
| GBP to USD | Direct | 1 | 1.6 | 1.6 | 1 |
| DEM to CHF | Indirect | 100 | 119.335 | 100 | 119.335 |
| CHF to DEM | Direct | 100 | 119.335 | 119.335 | 100 |
| USD to Euro | Indirect | 1 | 1.25 | 1 | 1.25 |
| Euro to FRF | Direct | 1 | 6.8 | 6.8 | 1 |
| USD to FRF | Dir/Trng/Euro | 1 | 5.44 | 6.8 | 1.25 |
| FRF to Euro | Indirect | 1 | 6.8 | 1 | 6.8 |
| Euro to USD | Direct | 1 | 1.25 | 1.25 | 1 |
| FRF to USD | Indir/Trng/Euro | 1 | 5.44 | 1.25 | 6.8 |

In all cases, the visual rate for a currency pair remains the same, regardless of the direction. This is consistent with business standards. For a direct quoted rate, you multiply by the visual rate; therefore the visual rate goes into RATE_MULT and 1 (or the quote units) goes into RATE_DIV. For an indirect quoted rate, you divide by the visual rate; therefore the visual rate goes into RATE_DIV and 1 (or the quote units) goes into RATE_MULT.

The following examples show indirect quotation, direct quotation with quote units, and triangulation:

100 USD to GBP (indirect) = (100 USD / 1.6) × 1 = 62.50 GBP

1000 CHF to DEM (direct w/ units) = (1000 CHF / 100) × 119.335 = 1193.35 DEM

100 USD to FRF (triangulate) = (100 USD / 1.25) × 6.8 = 544 FRF

## Application-Specific Requirements for Currency Conversion

Each application that shows a visual rate on a page must have an application-specific work record to hold the visual rate and the PeopleCode associated with it; this can be an existing work record. The suggested name for the field is VISUAL_RATE. The work record should also have a field to store the original rate for purposes of tolerance checking.

The application also typically provides an application-specific table to store RATE_MULT and RATE_DIV values that are stored on the database.

Application-specific PeopleCode needs to format work record fields and call the common functions in various circumstances, such as RowInit or FieldChange on the currency or visual rate.

### See Also

*PeopleSoft PeopleTools 8.44 PeopleBook: Global Technology*

*PeopleSoft PeopleTools 8.44 PeopleBook: PeopleCode Reference*

# Setting Up and Maintaining Market Rates

This section discusses how to:

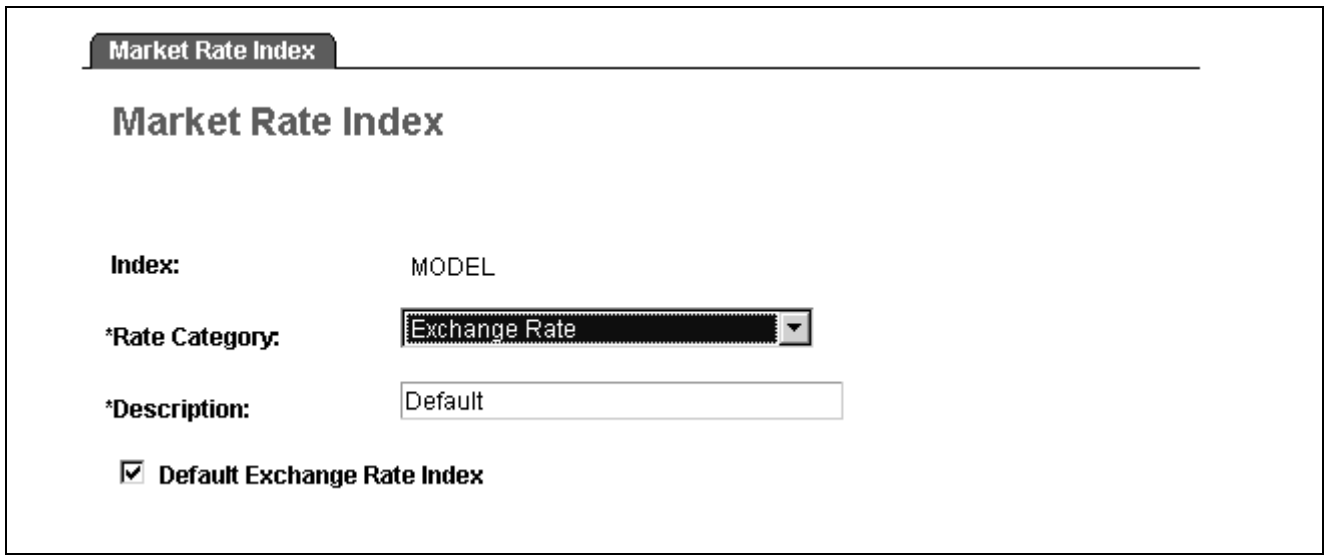
- Define market rate indices.
- Define market rates.
- Set rate variance limits.
- Specify exchange rate details.

## Pages Used to Set Up and Maintain Market Rates

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Market Rate Index | RT_INDEX_TBL | Depends on application. | Map indices for market rates into broad categories, such as exchange rates and interest rates.<br>Market rate indices are an application's highest level of organization for market rates. |
| Market Rates | RT_RATE_TBL | Depends on application. | View and maintain market rates. The fields available on the page vary depending on the rate category. |
| Rate Definition | RT_RATE_DEF_SEC | Depends on application. | Set limits for rate variances between rates. |
| Exchange Rate Detail | EXCH_RT_DTL | Click the Exchange Rate Detail button on the Market Rates page. | View detailed information about an exchange rate. |
| Currency Code Table | CURRENCY_CD_TBL | Depends on application. | Displays currency codes. |

## Defining Market Rate Indices

Access the Market Rate Index page.




Market Rate Index page

**Index** Displays the key term for the highest level of organization for market rates in the application.

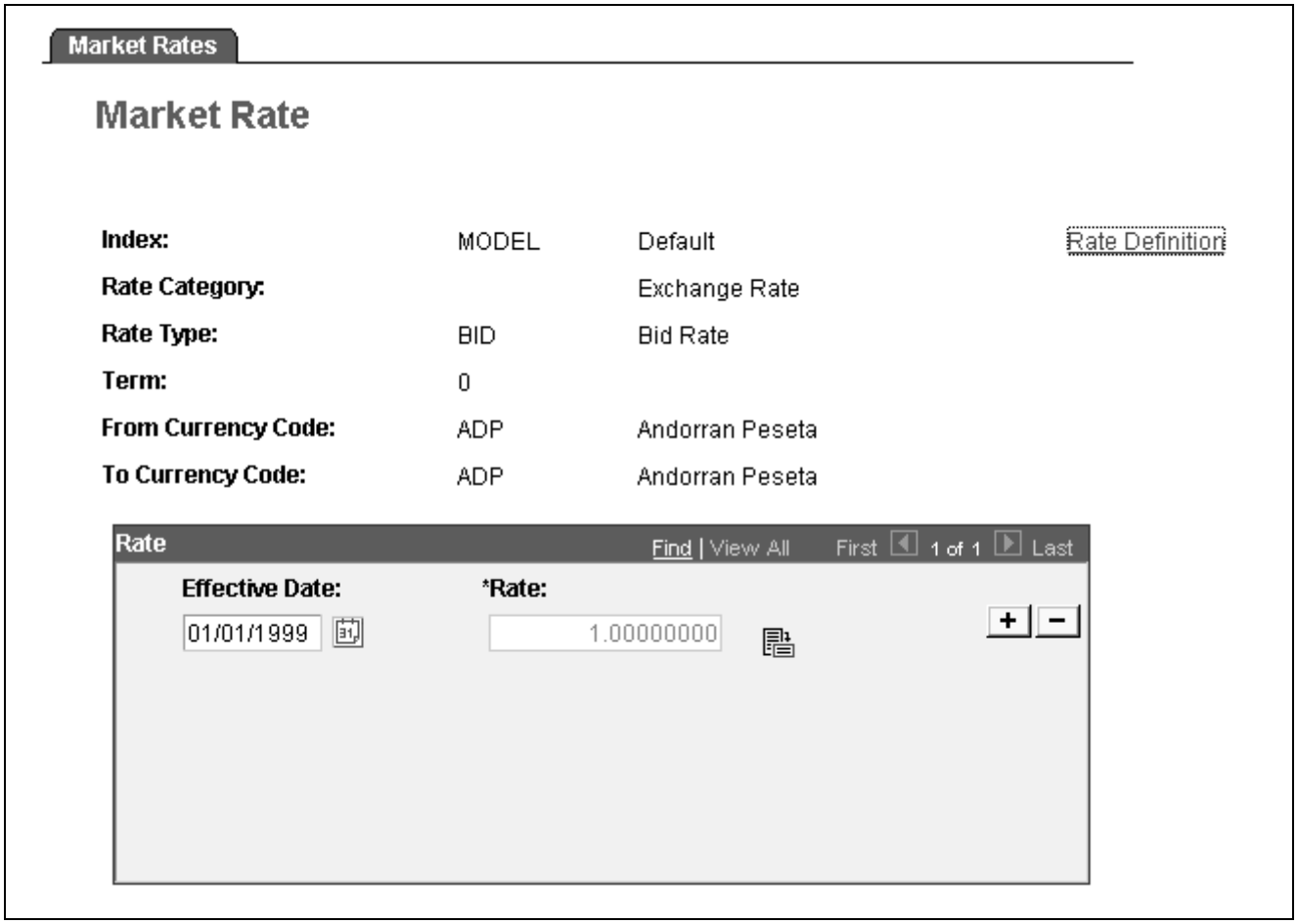
**Rate Category** Select a general category for the market rate index, such as *Exchange Rate*, *Commodity Price*, or *Interest Rate.*

**Default Exchange Rate Index** Select to indicate that the current market rate index is the default exchange rate index. This field is available only if:

- The Rate Category field is set to *Exchange Rate*..
- No other index is currently defined as the default exchange rate index.

## Defining Market Rates

Access the Market Rates page.




Market Rates page

This page is not editable if all the following are true:

- The rate is triangulated.
- The primary visual rate is the cross rate.
- The Allow Override check box is cleared for the exchange rate's quotation method on the Quotation Method page.

The Rate scroll area displays the visual rate, or in the case of triangulated exchange rates, the primary visual rate, which is typically the cross rate, but which can also be one of the other component rates of the triangle. Click the button next to the Rate field to access Exchange Rate Detail page and view all three visual rates of a triangulated exchange rate.

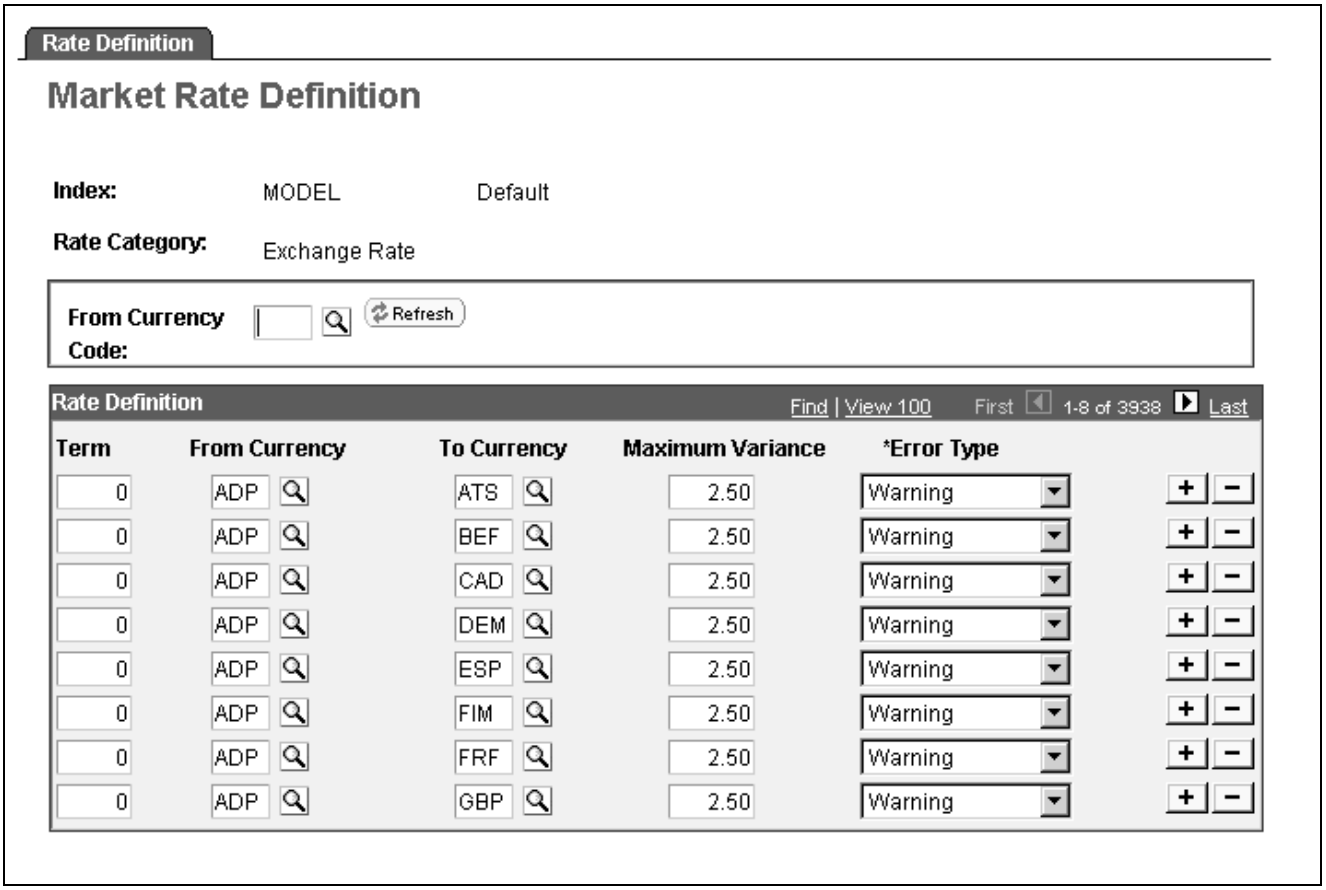
If a quotation method has been defined for the currency pair, and if the Auto-reciprocate check box for the quotation method is selected, then creating or maintaining a rate for a currency pair automatically creates or updates the rate for the reciprocal currency pair. For example, if you change the USD-to-GBP rate, the GBP-to-USD rate is automatically updated. You can only auto-reciprocate currency pairs for which quotation methods are defined.

If a rate definition does not already exist for the currency pair, one is automatically created with the default values of 2.5 percent maximum variance and Warning message processing.

**Note.** The results of changing the rate definition do not take effect until you save, close, then reopen the Market Rates page.

## Setting Rate Variance Limits

Access the Rate Definition page.



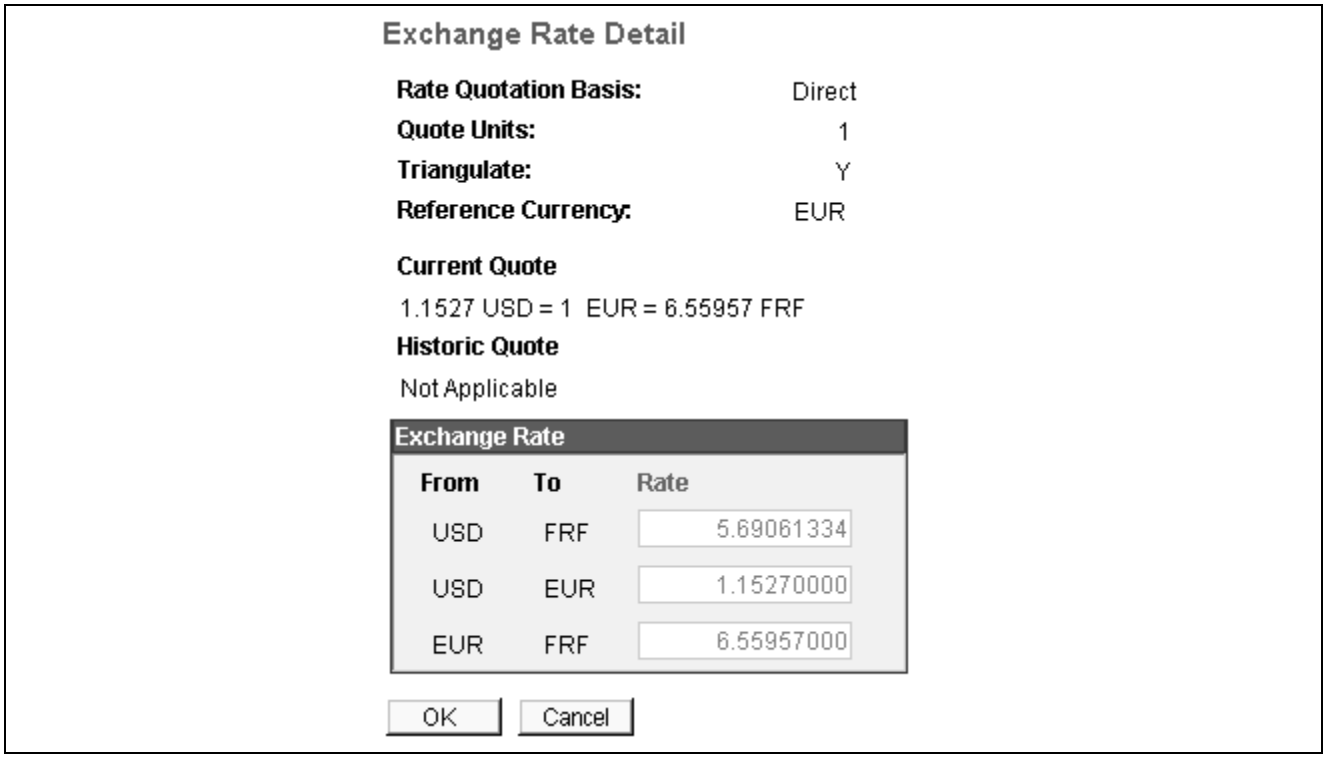

Rate Definition page

**Maximum Variance** Enter the percentage of variance allowed to maintain the market rate. If the change exceeds the tolerance, an error results. The default value is *2.5%*.

**Error Type** Select the type of error that results when maximum variance is exceeded during data entry. If set to *Warning*, the system alerts the user but permits the change. If set to *Stop*, the system alerts the user and does not permit the change. If set to *None*, no tolerance checking is done. The default value is *Warning*.

## Specifying Exchange Rate Details

Access the Exchange Rate Detail page.




Exchange Rate Detail page

The primary record for this page is the Exchange Rate Work record; however the data is derived principally from the Quotation Method table. For triangulated rates, you can update all three components of the triangulated rate.

When viewing a historic exchange rate, the rate is shown according to the current quotation method, which is not necessarily the quotation method originally used by the historic rate. This makes it convenient to compare how the exchange rate has changed over time, using a consistent quotation method, even if the quotation method has actually changed. When the system determines that an exchange rate is inconsistent with the current quotation method defined for that exchange rate, it displays a historic quote in the Historic Quote field.

For example, if you are viewing a historical rate where FRF were converted to USD directly using a calculated reciprocal rate of 1 FRF equals 0.1470588 USD, and the current quotation method for this currency pair is indirect, the conversion function recalculates the visual rate based on indirect quotation, that is 6.8000001 FRF = 1 USD.

**Rate Quotation Basis** Displays the quotation basis for the exchange rate as defined in the Quotation Method page. If no quotation method is defined the quotation basis is *Direct.*

**Quote Units** Displays the quote units for the exchange rate as defined in the Currency Quotation Method page. If no quotation method is defined the quote units is *1.*

**Triangulated** Displays the Triangulate setting for the exchange rate as defined in the Currency Quotation Method page. If no quotation method is defined, the Triangulate setting is *N.*

**Reference Currency** For triangulated exchange rates only. Displays the reference currency used in the triangulated exchange.

**Current Quote** Displays the current exchange rate used to convert the From currency to the To currency.

A direct, non-triangulated rate shows quote units (or 1) on the left side of the equals sign and the visual rate on the right, for example:

1 USD = 1.40000000 CAD

An indirect, non-triangulated rate displays the visual rate on the left side of the equal sign and quote units (or 1) on the right, for example:

1.400000000 CAD = 1 USD

A triangulated rate displays two component rates of the triangle: the rate for converting the From currency to the reference currency, and the rate for converting the reference currency to the To currency, for example:

1.25 USD = 1 EUR = 6.8 FRF

**Historic Quote** Displays a quote based on the quotation method originally used by a historic exchange rate, if the page logic determines that the exchange rate, as stored in the database, is inconsistent with the current quotation method. The field also displays a quote if the historic quote method was non-triangulated and the current quote method is triangulated. If the system does not determine that the exchange rate is inconsistent with the current quotation method, the field displays *Not Applicable.*

**Note.** A historic quote is also displayed if you override a cross rate and bypass triangulation, because the exchange rate being used is inconsistent with the current quotation method.

**Exchange Rate** Displays a single visual rate for non-triangulated exchange rates or all three component visual rates for triangulated exchange rates. The cross rate for triangulated exchange rates is editable only if the Allow Override check box is selected in the exchange rate's quotation method definition.

# Defining Currency Quotations

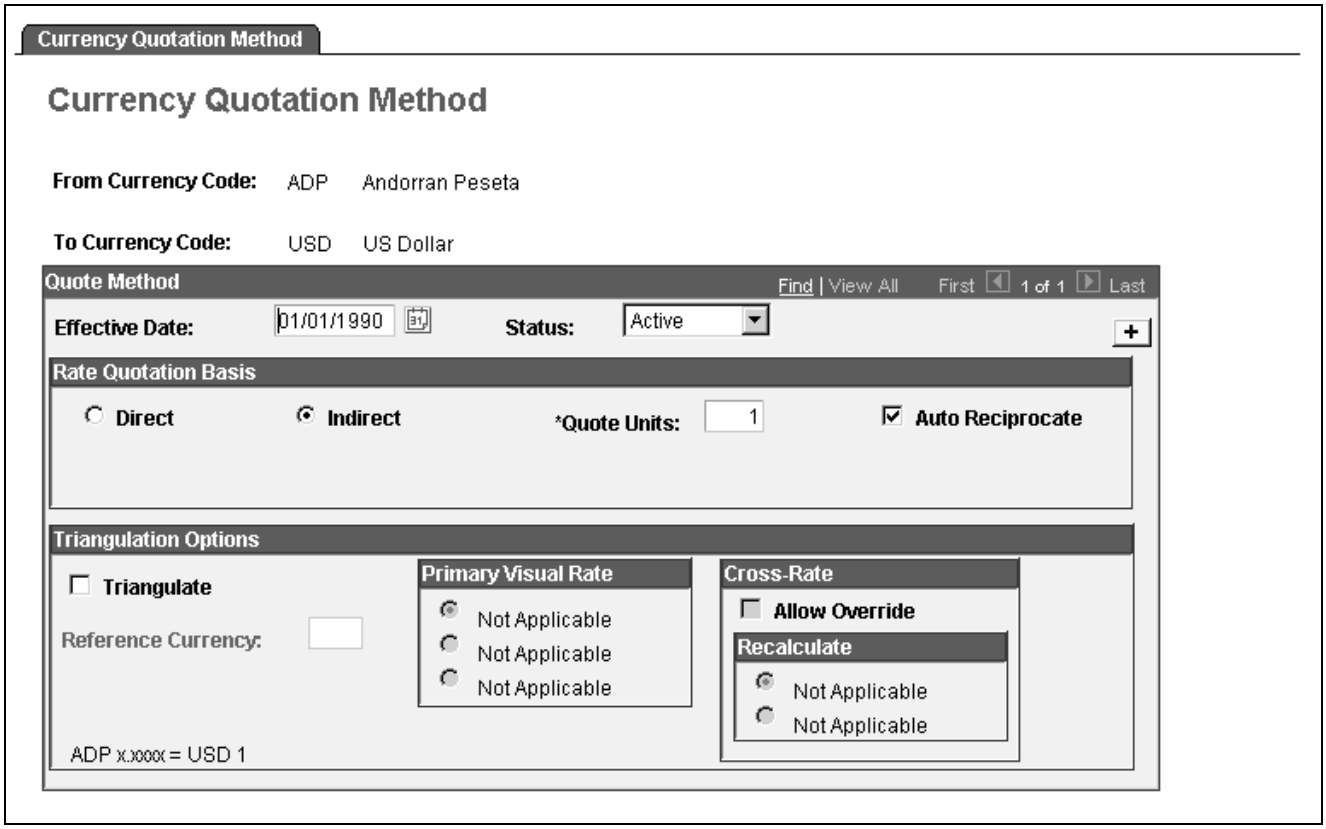
The currency quotation method you define for an exchange rate determines how the visual rate is interpreted into the RATE_MULT and RATE_DIV values stored on the database and, conversely, how the visual rate is derived from RATE_MULT and RATE_DIV.

## Page Used to Define Currency Quotations

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Currency Quotation Method | CURR_QUOTE_PNL | Depends on application. | Set up and maintain currency quotation methods for exchange rates. |

## Specifying Currency Quotation Methods

Access the Currency Quotation Method page.




Currency Quotation Method page

**Note.** You can view the currency quotation method for an exchange rate on the Exchange Rate Detail page while working on the Market Rates page.

**Direct** and **Indirect**

Select whether the rates for this currency pair are quoted direct or indirect.

For example, with USD to FRF,

- A direct quote means that 1 USD equals x.xxxx FRF
- An indirect quote means that x.xxxx USD equals 1 FRF

Even currency pairs that triangulate have to be classified as either direct or indirect, for use in displaying the calculated cross rate. The default value is *Direct.*

**Quote Units**

Enter a scaling factor to the exchange rate, as is common business practice for some currencies. This field can have any value, but usually is a power of 10 (10, 100, 1000, and so forth). The default value is *1*.

**Auto Reciprocate**

Select to automatically create or update the rate for the reciprocal currency pair whenever an exchange rate is added or updated.

For example, when the user enters a new USD to GBP rate, the GBP to USD rate is automatically updated. You can only autoreciprocate currency pairs for which quotation methods have been established. Selected by default.

The Auto Reciprocate check box pertains to the maintenance of the exchange rates on the Market Rates page. The quote method itself always autoreciprocates.

**Triangulate** Select to triangulate conversions between this currency pair using a reference currency.

**Reference Currency** Enter the reference currency for a triangulated conversion.

**Primary Visual Rate** With triangulated currency pairs, there are three exchange rates to consider:

- The rate between the From currency and the reference currency.
- The rate between the reference currency and the To currency.
- The calculated cross rate between the From currency and the To currency.

Select which of these three rates to be the primary visual rate, which is displayed on the primary pages and reports. In online applications, other components of the rate can be viewed and modified on the Exchange Rate Detail page.

By default the rate between the From and To currencies (the cross rate) is the primary visual rate.

**Allow Override** With triangulated currency pairs, select to enable users to override the cross rates. Clear by default.

**Recalculate** If Allow Override check box is selected, select to indicate which of the two other rates should be recalculated to bring the triangle back into balance again. By default, the first radio button in the group is selected.

# Calculating Currency Rates

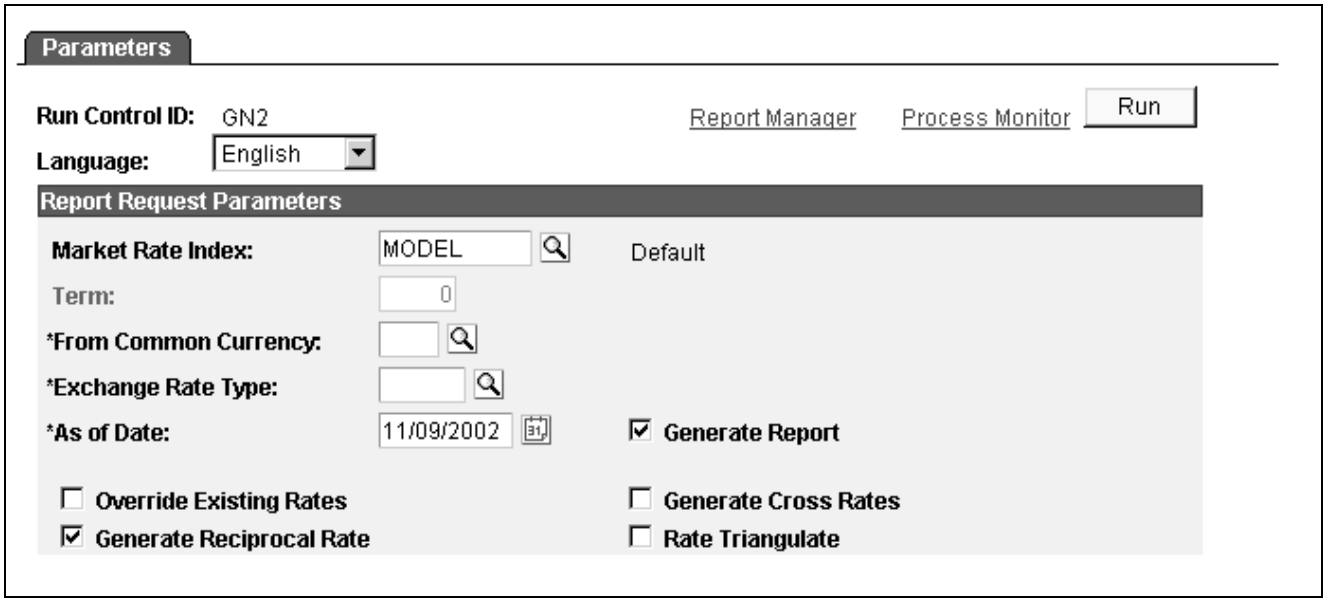
When calculating currency in PeopleSoft applications, all rates are processed and checked against the definitions established for each currency pair on the Currency Quotation page.

## Page Used to Calculate Currency Rates

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Parameters | RUN_EO9030 | Depends on application. | Set up reciprocal and cross rates. |

## Calculating Rates

Access the Parameters page.




Parameters page

**Term** Choose the term, which is expressed in days. For non-Treasury applications, use the default index, *MODEL*, and a term of *0*.

**As of Date** Select the effective date of the newly created exchange rates (the output of the process). The as of date also determines the rates used as the basis for the calculations (the input to the process). The report uses the most current currency quotation method for the currency pair as the input to the process. If the as of date is the current effective rate on the specified date, it can affect triangulation: a USD to FRF triangulated rate effective April 1, 2000, might be composed of the EUR to USD rate also effective April 1, 2000, and the fixed EUR to FRF rate effective January 1, 1999.

**Generate Report** Select to generate a report that displays the exchange rates and the reciprocal and cross-rate calculations.

**Override Existing Rates** Select to override rates for this exchange rate type currently residing on the Exchange Rate table (regardless of the as of date).

**Generate Reciprocal** Select to automatically calculate reciprocal rates.

**Generate Cross Rates** Select to automatically generate cross rates.

For example, for cross-currency rates between USD, FRF, and DEM, enter *USD to FRF = 5* and *USD to DEM = 1.43;* the system automatically generates FRF to DEM = 1.43/5 = 0.286. If you generate cross rates, select a From and To currency. The To Curr field is useful if you need a particular rate. You can enter a wild card of *%* in either field or both fields to indicate from all or to all currencies. Press F7 to add more currency types for cross rate generation.

**Rate Triangulate** Select to convert two currencies through a third currency.

### Reading the Generate Reciprocal Field

PeopleSoft does not directly manipulate the exchange rates; rather, the system uses numerator and denominator values such that:

From currency × RATE_MULT/RATE_DIV = To currency

Suppose you want a reciprocal rate between USD and FRF You enter USD to FRF = 5; the system automatically generates the reciprocal rate FRF to USD = 1/5, or 0.2. The conversion formula in this case is:

$$xFRF \frac{(RT\_MULT)\ 1}{(RT\_DIV)\ 5} = yUSD$$

# CHAPTER 5

# Using Datasets

This chapter provides an overview of datasets and discusses how to:

- Define dataset rules.
- Define user roles.
- Define mobile data distribution.

## Understanding Datasets

Datasets enable role-based filtering and distribution of data. You can limit the range and quantity of data displayed for a user by associating dataset rules with user roles. The result of dataset rules is a set of data appropriate to the user's roles.

You can also limit the range and quantity of data passed to a mobile device by defining data distribution rules based on datasets. Data distribution rules define the selection of data downloaded to a mobile device. The data set may differ depending on the mobile device.

**Note.** If you are using PeopleCode to control data distribution, consider using datasets instead.

## Defining Dataset Rules

This section provides an overview of dataset rules and discusses how to create dataset rules.

### Understanding Dataset Rules

Dataset rules define datasets for use in conjunction with each user role's security rules. Defining dataset rules creates Structured Query Language (SQL) statements that select the dataset displayed for each rule.

Generally, define rules, users, and roles in this order:

1. Define dataset rules to determine the data to be displayed.
2. Assign the dataset rules to user roles according to role security and data requirements.

   Each user role can have multiple dataset rules. You can use existing user roles or create new roles.
3. Assign user roles to individual user IDs.

   Each user can have multiple roles.

## Page Used to Define Dataset Rules

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Dataset Rules | EOEC_DATASET | Enterprise Components, Component Configurations, Datasets, Dataset Rules | Define the rules that make up a dataset. |

## Creating Dataset Rules

Access the Dataset Rules page.

**Status** Select *Active* or *Inactive*.

**Search Record Name** Select the name of the search record for this rule. You can create a view specifically for use in the rule.

**AND or OR** For second and subsequent rule conditions, specify *AND* or *OR*, or leave blank if the rule statements are nested.

**…((** and **))…** If the AND or OR field is left blank, specify the nesting level for this condition. Be sure to match opening and closing parentheses.

**Field Name** Select the field name on which this rule operates.

**Operator** Specify the operation with which to compare the specified field value. Select from standard conditional operators.

**Field Value** Specify the value of the specified field against which to compare.

**Test SQL** Click to test the validity of the rule conditions. The result is returned below the button.

**Show SQL** Click to view the SQL statement generated by the rule.

# Defining User Roles

Set up user roles by associating dataset rules with user roles.

## Page Used to Define User Roles

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Dataset Roles | EOEC_MP_ROLE | Enterprise Components, Component Configurations, Datasets, Dataset Roles | Define the dataset rules that apply to user roles. |

## Defining Dataset Roles

Access the Dataset Roles page.

**Dataset Name** Select the dataset with which the component rule is associated.

| | |
|---|---|
| **Rule** | Select the component rule. |
| **Laptop** and **PDA** | Select to display the resulting data on a laptop computer or PDA. |

**Note.** If neither Laptop nor PDA is selected, no data from this rule is displayed.

# Defining Mobile Data Distribution

Use datasets to define the data distributed to mobile devices running PeopleTools Mobile Agent. This section provides an overview of mobile data distribution and discusses how to:

- Define mobile data distribution.
- Use mobile user rules.

## Understanding Mobile Data Distribution

Mobile devices can have limited processing power, storage capacity, and display space. You can limit the range and quantity of data passed to the mobile device by associating dataset rules with synchronizable component interfaces. Mobile data distribution rules define the selection of data from network servers for download to a mobile device. The result of data distribution rules is a set of data appropriate to the user's roles. The set of data may differ depending on the mobile device.

Data distribution for mobile applications implements security and filters the data downloaded to the mobile device. You define data distribution for mobile devices based on datasets by selecting a rule assigned to the user's role.

### Implementing Mobile Data Distribution

To filter data defined by dataset rules to mobile devices, developers must include the function DistributeDataByRules() in the synchronizable component interface's OnSelect PeopleCode method.

For example:

```
Declare Function DistributeDataByRules PeopleCode
FUNCLIB_ECMOBIL.EOEC_ONSELECT FieldFormula;
DistributeDataByRules();
```

### See Also

*PeopleSoft PeopleTools 8.44 PeopleBook: Mobile Agent*

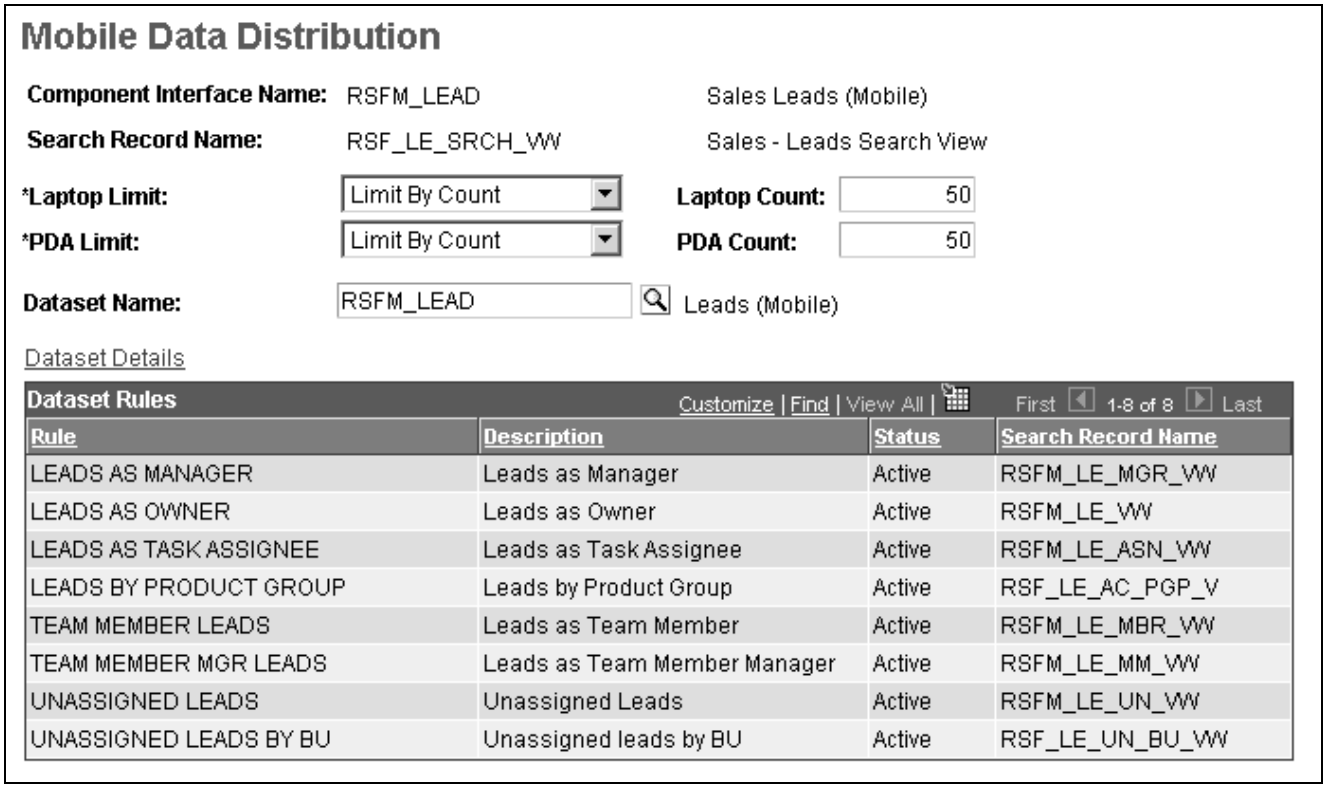
*PeopleSoft PeopleTools 8.44 PeopleBook: PeopleCode Reference*

## Pages Used to Define Mobile Data Distribution

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Mobile Data Distribution | EOEC_COMPONENT | Enterprise Components, Component Configurations, Mobile, Mobile Data Distribution | Define data distribution rules for mobile devices based on datasets. |
| Mobile User Rules | EOEC_MP_USERRULE | Enterprise Components, Component Configurations, Mobile, Mobile User Rules | Preview the effect of mobile data distribution. |

## Defining Mobile Data Distribution

Access the Mobile Data Distribution page.




Mobile Data Distribution page

**Note.** Component interfaces selected for mobile data distribution must be synchronizable; only synchronizable component interfaces are available from the prompt.

**Laptop Limit** Select the limiting factor for data instances to be downloaded to a laptop computer during synchronization. Select from:

- *All Data:* All data matching the rule's conditions is downloaded.
- *Limit By Count:* Only data matching the rule's conditions up to the specified count is downloaded. Referenced data is also included.

- *Referenced Data Only:* Only data referenced by the component interface is downloaded.

**PDA Limit** Select the limiting factor for data instances to be downloaded to a PDA during synchronization. Select from:

- *All Data:* All data matching the rule's conditions is downloaded.
- *Limit By Count:* Only data matching the rule's conditions up to the specified count is downloaded. Referenced data is also included.
- *Referenced Data Only:* Only data referenced by the component interface is downloaded.

**Laptop Count** and **PDA Count** If you select *Limit By Count* in the Laptop Limit field or the PDA Limit field, specify the maximum number of data instances to be downloaded.

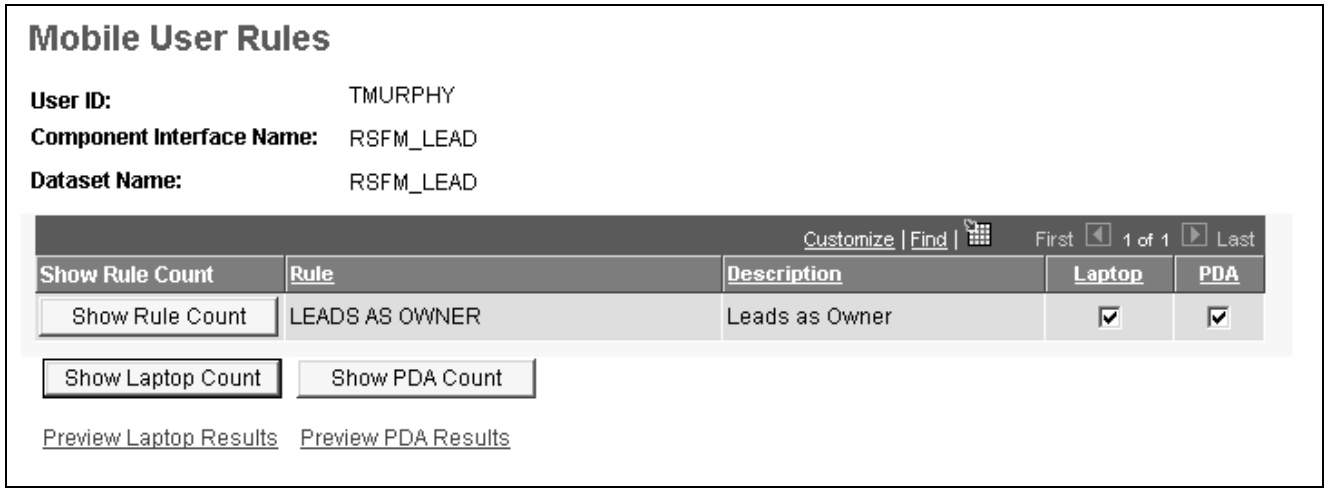
**Dataset Name** Select the dataset to apply to this distribution rule.

**Dataset Details** Click to view the selected dataset definition.

The Dataset Rules grid lists rules for the specified dataset.

## Using Mobile User Rules

Access the Mobile User Rules page.




Mobile User Rules page

A mobile user can specify whether a selected rule returns data to a selected mobile device. The user must be signed in with a user ID, not as an administrator, to define user rules.

**Show Rule Count** Click to view how many results the distribution rule returns.

**Show Laptop Count** and **Show PDA Count** Click to view how many results the distribution rule returns to a laptop computer or PDA.

**Laptop** and **PDA** Select to display the results of this rule on a laptop computer or on a PDA, or both.

**Note.** If neither Laptop nor PDA is selected, no data from this rule is displayed.

| | |
|---|---|
| **Preview Laptop Results** and **Preview PDA Results** | Click to preview the data that will be downloaded to a laptop or PDA as a result of this rule. |

# CHAPTER 6

# Using Interactive Reports

This chapter provides an overview of PeopleSoft Business Analysis Modeler Interactive Reports and discusses configuration of the Interactive Report environment.

**Note.** Each PeopleSoft application that delivers an Interactive Report provides documentation for viewing that specific report.

## Understanding Interactive Reports

PeopleSoft Business Analysis Modeler is a calculation and analysis tool used by various PeopleSoft applications. Interactive Reports use PeopleSoft Business Analysis Modeler to calculate and present multi-dimensional data to the user.

Interactive Reports include the following:

- Pages to define and configure Interactive Reports.
- PeopleSoft queries to select the appropriate data for the report.
- The PeopleSoft Business Analysis Modeler model file containing the data structure and calculation rules.
- The PeopleSoft Business Analysis Modeler PageView file containing the report layout.
- A page to capture the capture query prompt values and launch the report.

## Common Elements Used in This Chapter

| | |
|---|---|
| **Report ID** | Unique Interactive Report identifier assigned by the report developer. |
| **Description** | Describes the associated report, server, and database connection, depending on which page this field appears. |
| **Last Update Date/Time** | Lists the user and date the page was most recently updated. |

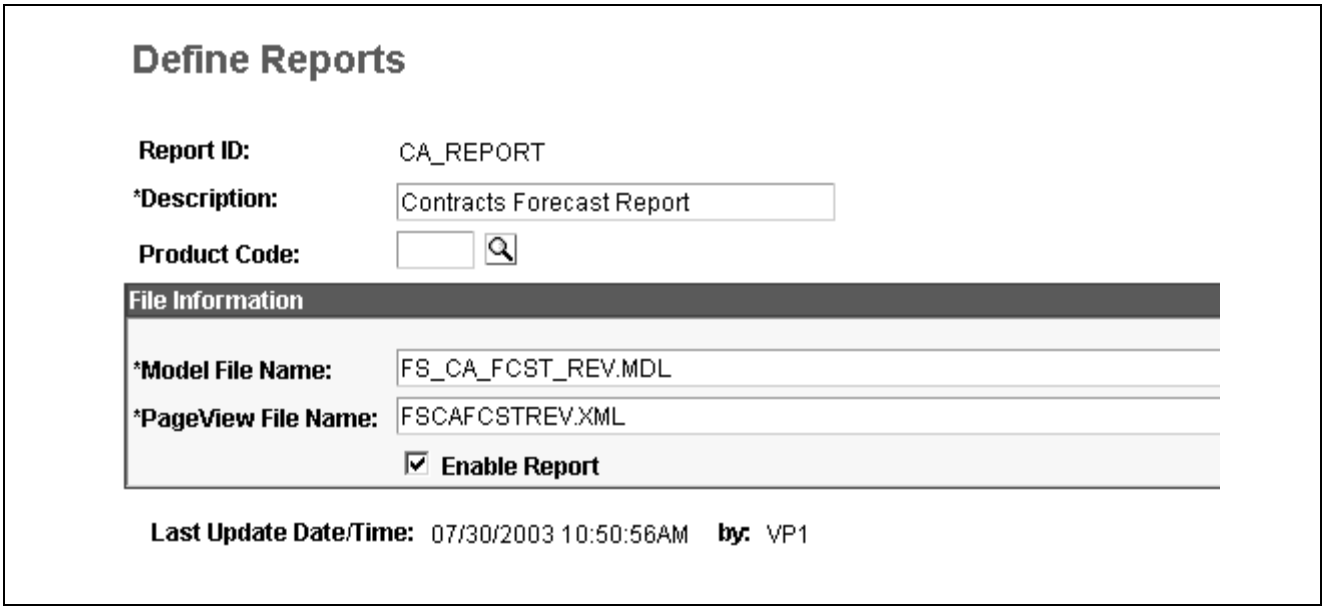
# Configuring Interactive Reports

## Pages Used to Configure Interactive Reports

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Define Reports | EOBF_REPORT | Enterprise Components, Interactive Report Definitions, Reports | Defines the report file names. |
| Define Servers | EOBF_SERVER | Enterprise Components, Interactive Report Definitions, Servers | Defines web server and file paths. |
| Define ODBC Connections | EOBF_ODBC | Enterprise Components, Interactive Report Definitions, ODBC Connections | Defines the ODBC connection string to the PeopleSoft database. |
| Environment Configuration | EOBF_CFG_REPORT | Enterprise Components, Interactive Reports Definition, Environment Configuraton | Defines the PeopleSoft Business Analysis Modeler environment for a specific report. |
| Map Queries | EOBF_QRY_NAME | Enterprise Components, Interactive Reports Definition, Queries | Maps specific PeopleSoft queries to import maps in the PeopleSoft Business Analysis Modeler model. Import map entries on this page have been defined by the PeopleSoft application delivering the report. |
| Base Language Query Prompt | EOBF_QRY_PRMPT | Enterprise Components, Interactive Reports Definition, Queries, View Query Prompts | Specifies the record and field containing the base language query prompt values when the report is launched. |
| Related Language Query Prompt | EOBF_QRY_PRMPT | Enterprise Components, Interactive Reports Definition, Queries, View Query Prompts | Specifies the record and field containing the related language query prompt values when the report is launched. |
| Validate Environment | EOBF_ENV_UTILITY | Enterprise Components, Interactive Reports Definition, Validate Environment | Validates the connection of the architectural components at installation. |

## Defining Reports

Access theDefine Reports page.




Define Interactive Reports

**Product Code** (Optional) Code of the PeopleSoft licensed product associated with the report.

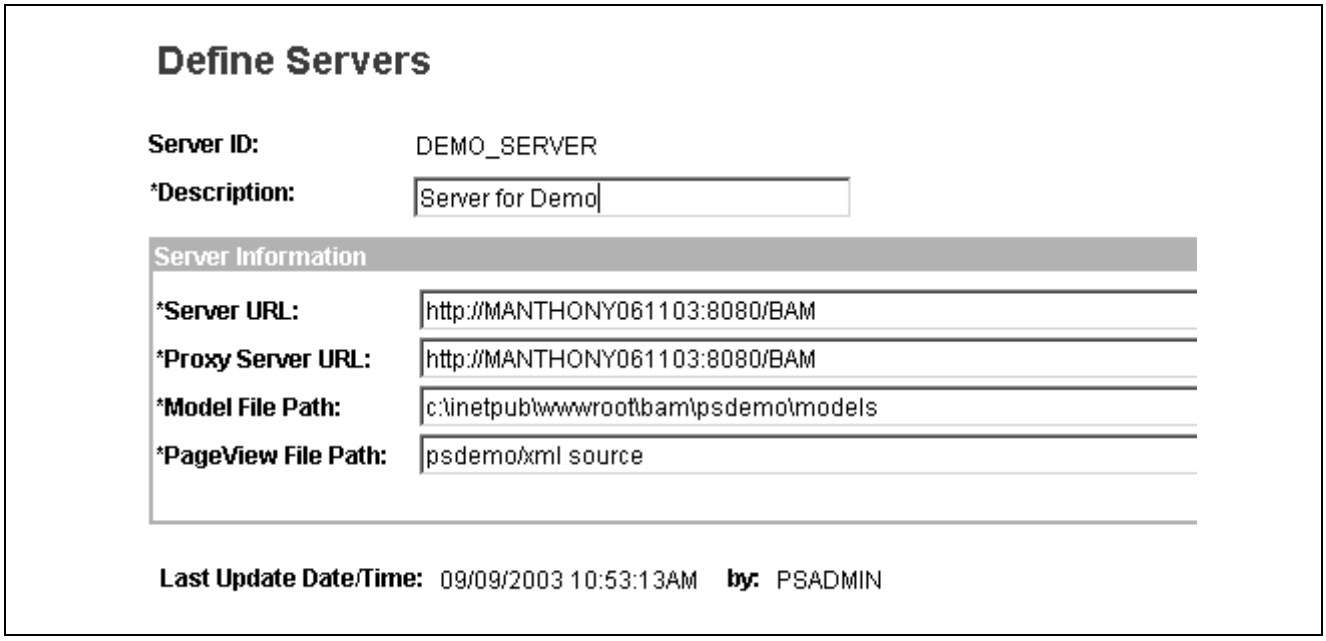
**Model File Name** File name of the PeopleSoft Business Analysis Modeler model file delivered for the report.

**Page View Name** File name of the PeopleSoft Business Analysis Modeler PageView file delivered for the report.

**Enable Report** Select this checkbox to allow valid users access to the report.

## Defining Servers

Access the Define Servers page.




Define Servers

**Server ID** Unique identifier of the PeopleSoft Business Analysis Modeler Analytic Web Server used for Interactive Reports.

**Server URL** and **Proxy Server URL** Location of the PeopleSoft Business Analysis Modeler Analytic Web Server and proxy server. Include the port specification if its value is other than 80. If there is no proxy server, enter the Server URLProxy Server URL field.

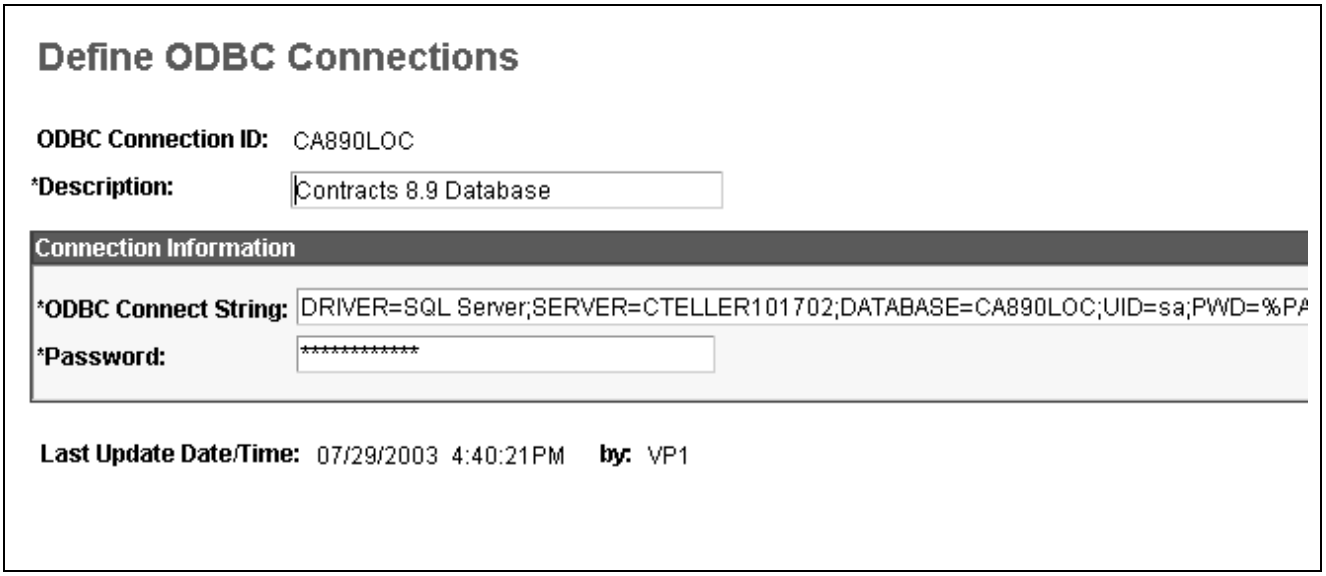
**Model File Path** Path of the .MDL (model) file shown on the Define Reports page. These files are usually on the PeopleSoft Business Analysis Modeler Analytic Web Server in the c:\inetpub\wwwroot\bam*PeopleSoft product*\models directory.

**PageView File Path** Path of the .XML file shown on the Define Reports page. These files are usually on the PeopleSoft Business Analysis Modeler Analytic Web Server in the c:\inetpub\wwwroot\bam*PeopleSoft product*\xml source directory.

**Note.** The PageView File Path is a relative path under c:\inetpub\wwwroot\BAM \ with forward slashes indicating subdirectories.

## Defining ODBC Connections

Access the Define ODBC Connections page.




Defining ODBC Connections

**ODBC Connection ID** Unique identifier of the ODBC Connect String.

**ODBC Connect String** Used for queries to the PeopleSoft database. The exact connect string syntax depends on the database platform and is listed in the following table.

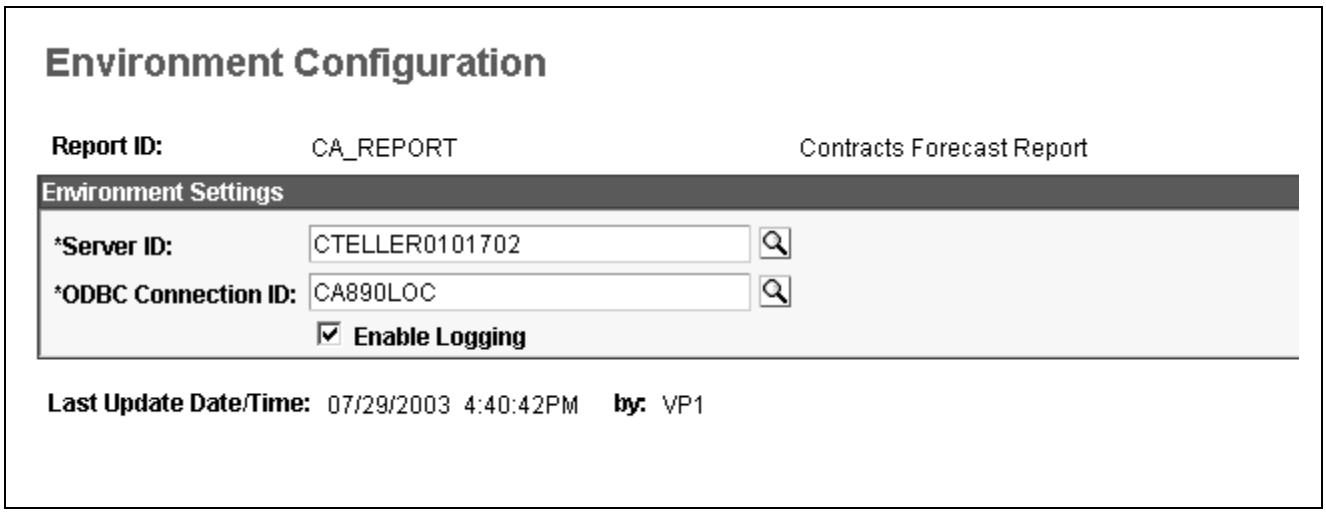
**Password** and **Confirm Password** Enter a password to connect to the database listed in the ODBC Connect String. Re-enter the password in the Confirm Password field. The user ID listed must have a minimum of *Select* privileges on the database.

The following table lists the syntax to use for the various database platforms.

| Database Platform | Connection String Syntax |
|---|---|
| Microsoft SQL Server | DSN=*Data Source*;SERVER=*Server Name*;DATABASE=*Database Name*;UID=*UserID*;PWD=%PASSWORD%; |
| Oracle | DSN=*Data Source*;UID=*UserID*;PWD=%PASSWORD%; DBTYPE=ORACLE; |
| DB2 Unix | DSN=*Data Source*;UID=*UserID*;PWD=%PASSWORD%; SCHEMA=*Schema Owner*; |
| DB2 OS390 | DSN=*Data Source*;UID=*UserID*;PWD=%PASSWORD%;SCHEMA=*Schema Owner*;SQLID=*Database Name*; |
| Sybase | DSN=*Data Source*;UID=*UserID*;PWD=%PASSWORD%;DBTYPE=SYBASE; |
| Informix | DSN=*Data Source*;UID=*UserID*;PWD=%PASSWORD%; |

## Configuring the Environment

Access the Environment Configuration page.




Environment Configuration

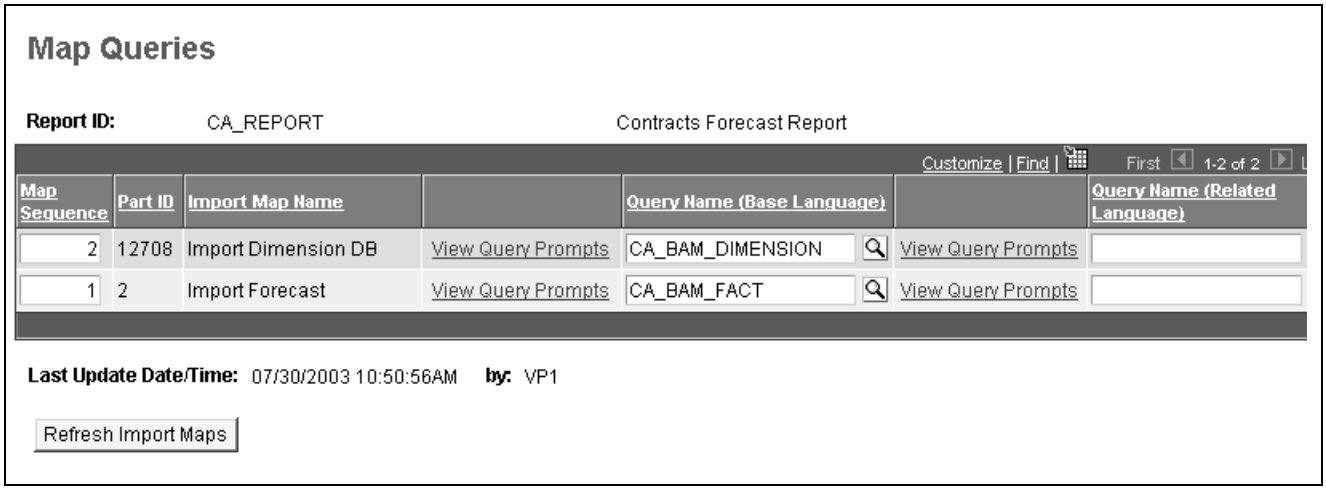
Select the Server ID and ODBC Connection ID to be used with this report.

**Enable Logging** Select this checkbox to have all PeopleSoft Business Analysis Modeler Web Server commands logged. The default location for this log isc:\inetpub\wwwroot\bam\logyyyymmdd.txt. You can modify the location of the log during installation.

See *PeopleSoft Business Analysis Modeler 8.8 Installation Guide* .

## Mapping Queries

Access the Map Queries page.




Map Queries

**Map Sequence** Defines the order in which the queries are executed. Order can be important in the validation of data integrity.

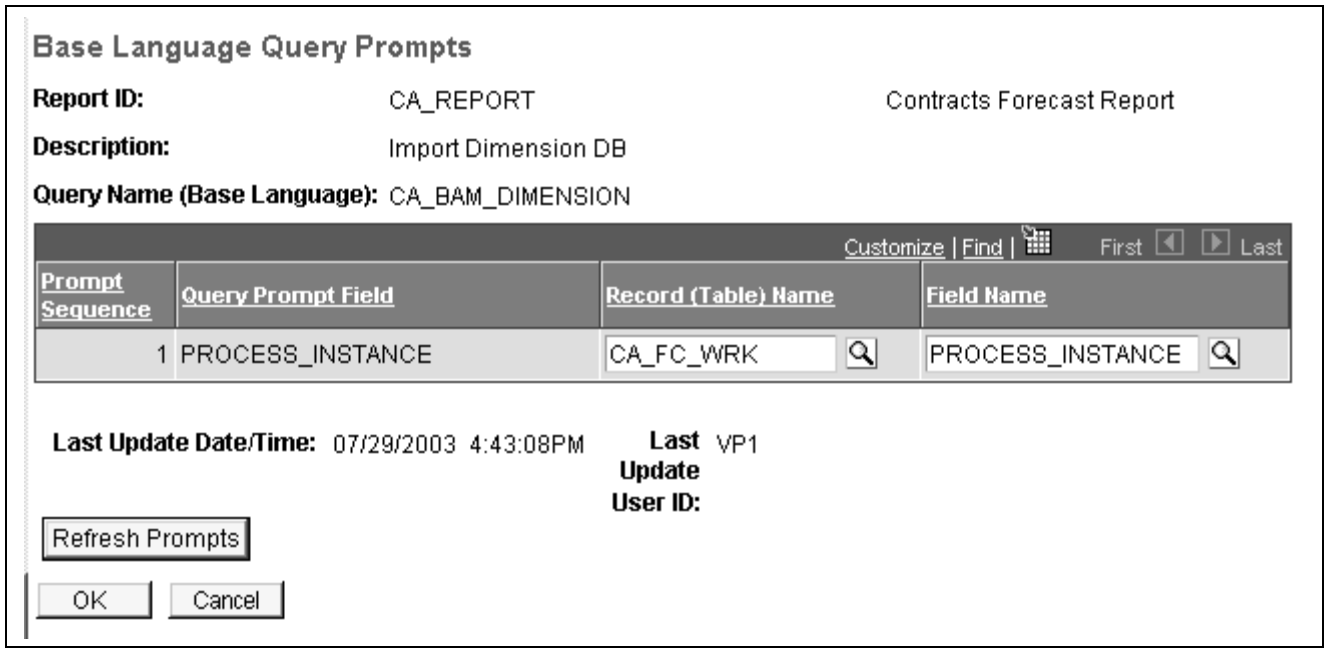
**Part ID** Used to identify PeopleSoft Business Analysis Modeler Import Maps. This value is retrieved from the .MDL file.

**Import Map Name** Import Map name in the related .MDL file.

| | |
|---|---|
| **View Query Prompts** | If you select this link, a Query Prompt page appears listing prompts for the query. |
| **Query Name (Base Language)** | Identifies the PeopleSoft query used for the selection of data to the Import Map. |
| **Query Name (Related Language) (Optional)** | If the report is used in a multi-lingual environment, identifies the PeopleSoft query which retrieves translated descriptions used in the Interactive Report. |
| **Refresh Import Maps** | Clicking this retrieves a new list of import maps from the .MDL file and deletes current entries on the page. |

## Viewing Query Prompts

Access the Base Language Query Prompt or Related Language Query Prompt secondary page.

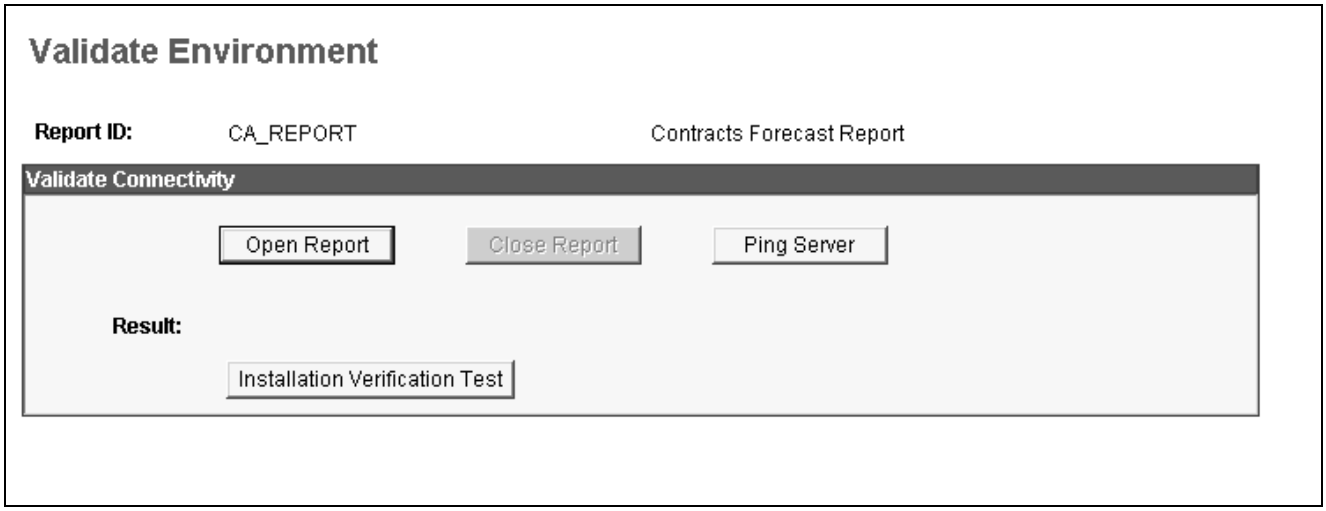



Base Language Query Prompt

| | |
|---|---|
| **Prompt Sequence** | Displays the order of the prompt in the query definition. |
| **Query Prompt** | Displays the prompt field in the query definition. |
| **Record (Table) Name** and **Field Name** | Specifies the record name and field name containing the prompt value. |
| **Refresh Prompts** | Clicking this retrieves a new list of prompts and deletes current entries in the page. |

## Validating the Environment

Access the Validate Environment page.




Validate Environment

**Open Report** and **Close Report** Clicking Open Report opens the .MDL file and performs a checkout on that model. Clicking Close Report removes the checkout created. The Result field displays a message indicating success or failure for both operations.

**Ping Server** Clicking this sends a command to ping the PeopleSoft Business Analysis Modeler Analytic Web Server. The Result field displays a message indicating success or failure.

# CHAPTER 7

# Setting Up the Credit Card Interface

This chapter provides an overview of credit card processing and discusses how to:

- Set up the credit card interface.
- Configure the Java interlink plug-in.
- Configure the PeopleSoft Application Designer plug-in.
- Set transaction inputs and outputs.

---

**Note.** PeopleSoft applications use PeopleSoft Business Interlink architecture to interface with credit card processing vendors. To configure interfaces with unsupported vendors, create an interlink plug-in.

---

## Understanding Credit Card Processing

This section discusses:

- Credit card processing options.
- CyberSource integration.
- XML-based integration.
- Manual processing.

### Credit Card Processing Options

Use the credit card processing interface to integrate with credit card processing vendors.

You have the following options for credit card processing:

- Use CyberSource (a third party) for credit card authorization and payment.

  If you use CyberSource (or another third-party application that conforms to CyberSource's API, such as TouchNet), agents processing credit cards can use the Authorize Credit Card to submit the transaction to CyberSource for authorization, billing, or credit. You can choose which types of transactions to permit.

  Integration with other vendors requires customization.

- Use an XML-compliant credit card processing vendor.

  If you use an XML-compliant credit card processing vendor, you need to ensure that the XML message delivered is transformed to the format that the vendor is expecting. PeopleSoft delivers an EOEC CCI SYNC message that generates the XML.

  Agents can then process credit cards using the Authorize Credit Card or similar page to submit the transaction to the vendor for authorization, billing, or credit. You can choose which types of transactions to permit.

• Manual credit card processing.

If you do not use CyberSource or XML-based credit card processing, agents can use the Authorize Credit Card or similar page to capture information for use with your own solution for processing credit card payments.

### CyberSource Integration

The Credit Card Authorize Bill And Credit EIP enables interaction between PeopleSoft Support and CyberSource. The EIP includes the EOEC_CCI_TRANSACTION business interlink. Activate the business interlink by entering a CyberSource merchant ID on the CyberSource Setup page.

PeopleSoft does not ship CyberSource code or programs. Contact CyberSource to contract for its services and to obtain a CyberSource merchant ID and supporting files. When contacting CyberSource, identify yourself as a PeopleSoft customer.

### XML-Based Integration

If you use an XML-based interface, you must use PeopleSoft Integration Broker node setup to indicate where the processor is located.

In configuring the node, specify that the node type is *External*, and the routing type is *Explicit.* Also specify the authentication option that you arranged with the credit card processor.

### Manual Processing

If you decide not to use CyberSource or an XML-based credit card processing system, then the Authorize Credit Card or similar page captures information for use with your own solution for processing credit card payments.

To support manual processing, you must build your own process to read credit card data from the PeopleSoft tables and process the credit card transaction.

## Setting Up the Credit Card Interface

This section discusses how to:

• Define connection parameters.

• Define accepted credit card types.

• Test the credit card interface.

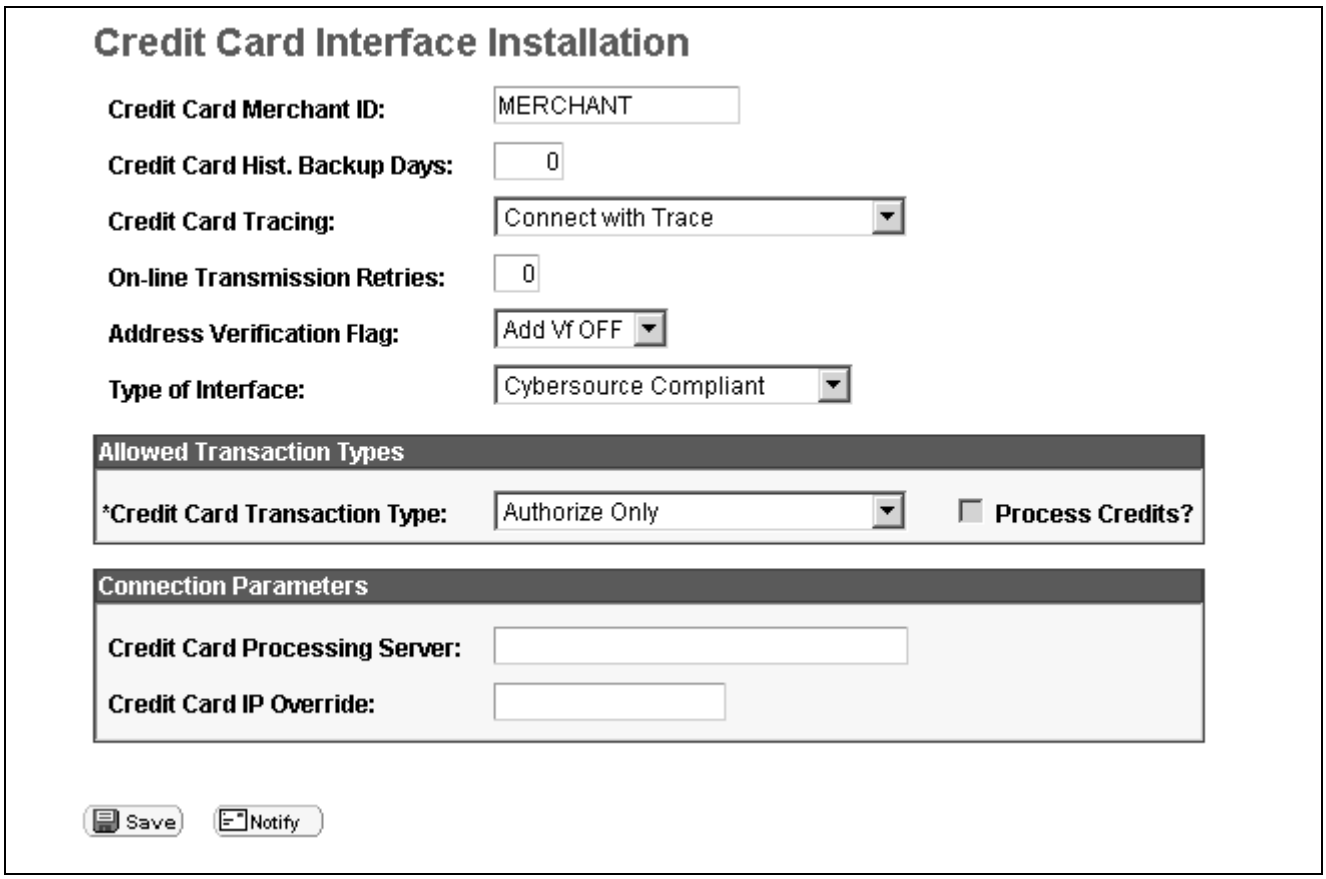
• Test the credit card transaction.

## Pages Used in Setting Up the Credit Card Interface

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Credit Card Interface Installation | EOEC_CCI_INSTAL | Enterprise Utilities, Component Configurations, Credit Card Interface, Setup | Set up connection parameters for credit card processing calls to CyberSource or another third-party credit card authorization and payment vendor that conforms to CyberSource.<br>Before you set up credit card processing options, establish your merchant account with CyberSource (or another CyberSource-compatible vendor). |
| Credit Card Setup | EOEC_CCI_CARDTYPE | Enterprise Utilities, Component Configurations, Credit Card Interface, Credit Card Types | Define the types of credit cards you accept for credit card processing. |
| Credit Card Entry/Display Test | EOEC_CCI_TEST | Enterprise Utilities, Component Configurations, Credit Card Interface, Test Credit Card Interface | Enter test credit card information, which you can then submit to verify that your credit card processing is functioning properly. |

## Defining Connection Parameters

Access the Credit Card Interface Installation page.




Credit Card Interface Installation page

This page is required for both CyberSource (or other third-party service that conforms to CyberSource) and XML-compliant processing systems. Verify connection requirements with your vendor.

**Credit Card Merchant ID** Enter the merchant ID supplied by CyberSource or your other vendor. You must have a merchant ID in order for the Submit button to appear on the Authorize Credit Card or similar page.

**Credit Card Hist. Backup Days** (credit card history backup days) If you create a process that archives history records, specify the number of days that you retain credit card authorization history records. This field does not currently affect retention of credit card authorization history.

**Credit Card Tracing** Select whether to keep a log of credit card transactions. If you trace your activity, the system creates the trace file CreditCardTrace.txt and places it in the PeopleTools APPSERV directory in the subdirectory with the same name as your database (X:\PT820\APPSERV\mydatabase\CreditCardTrace.txt).

The trace file contains a list of the parameters passed for each credit card transaction. If the file already exists, new transactions are added to the end of the file, providing a running log.

To reconfigure the Credit Card Authorize Bill and Credit EIP to give the trace file a different name or location, modify the EOEC_CCI_TRANSACTION business interlink using PeopleSoft Application Designer.

Select from the following options:

*Connect with Trace:* Connect to CyberSource or your other vendor and create a trace file.

*No Connect with Trace:* Create a trace file only. Do not attempt to connect to CyberSource or your other vendor. Use this option to test or troubleshoot credit card transaction data.

*Production:* Send data directly to CyberSource or your other vendor without creating a trace file. This is the most efficient method of transmitting your data; however, you have no record of the passed parameters.

*Vendor Trace (CyberSource only):* Invoke a troubleshooting utility provided by CyberSource to aid in diagnosing connection problems. The utility creates the troubleshooting file ICSAPILOG.TXT.

See CyberSource documentation.

**On-line Transmission Retries** Enter a value between *0* and *9* to specify how many times the system should reattempt transmissions in the event of transmission failure.

**Address Verification Flag** Credit card transmissions can fail authorization if the address you send doesn't exactly match the billing address for the credit card. Select from:

*Add Vf ON* (address verification on): Transactions fail when the address you send does not match the credit card billing address. This is the default value.

*Add Vf OFF* (address verification off): Transactions do not fail when the address you send does not match the billing address on the credit card.

**Type of Interface** Select *CyberSource Compliant* for CyberSource or another vendor that complies with CyberSource. Select *XML Compliant* for an XML-compliant vendor.

## Allowed Transaction Types

**Credit Card Transaction Type** Choose the types of transactions your agents are allowed to submit. Disallowed transaction types are not available on the Authorize Credit Card page. Select from:

*Authorize Only:* CyberSource or your other vendor verifies that the card is valid for the charge (the customer has enough credit to pay for the order, the card is not stolen, and so on). The vendor does not bill the credit card.

*Bill Only:* CyberSource or the other vendor bills the card without first verifying that the card is valid for the charge. Select this option if you have preauthorized the transaction and you want to submit the transaction for billing only.

*Authorize and Bill:* CyberSource or your other vendor performs both authorization and billing during the same transaction. The vendor charges the customer's credit card on receiving authorization.

*Credit Only:* CyberSource or your other vendor credits the customer's credit card.

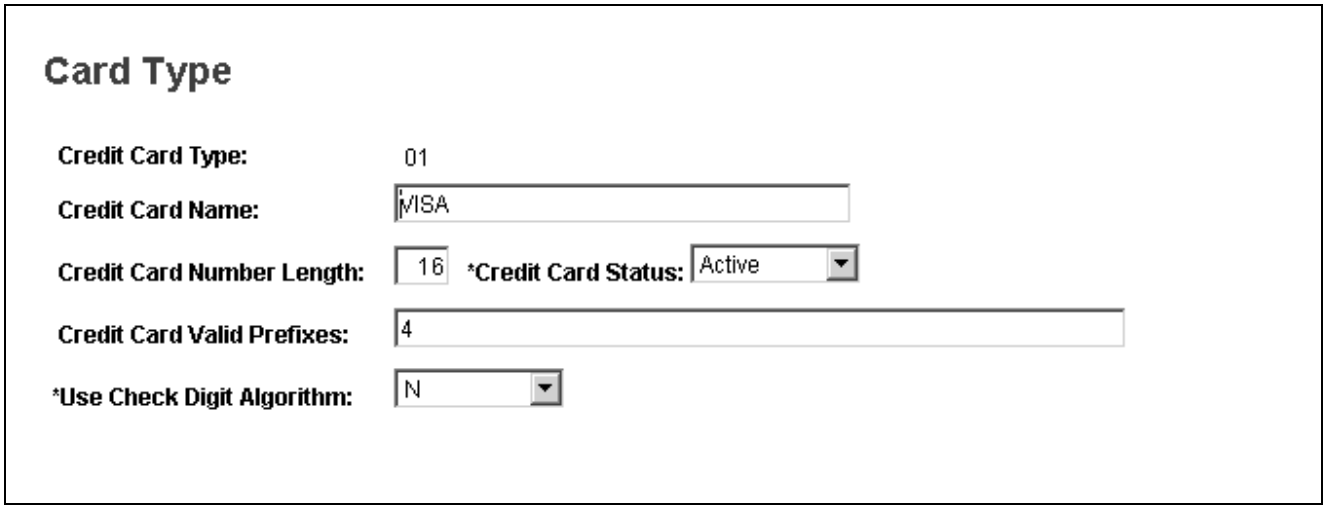
**Process Credits** Select to permit agents to submit credit transactions as well as billing transactions. Available only if you selected either *Authorize and Bill* or *Bill Only* in the Credit Card Transaction Types field.

### Connection Parameters

CyberSource (and other compatible third-party vendors) provide you with information to connect with their systems. Enter that information so that PeopleSoft can make the connection when you submit a transaction for authorization.

## Defining Accepted Credit Card Types

Access the Card Type page.




Card Type page

This page is required for all types of credit card processing.

PeopleSoft delivers data for most popular credit card types; you can modify the existing definitions or add to them, but you do not have to create new definitions if your organization uses what is supplied.

**Credit Card Type** The following credit card types are delivered: *01* for Visa, *02* for MasterCard, *03* for Diners Club/Carte Blanche, *04* for American Express, and *05* for Discover.

**Credit Card Name** Enter a credit card name such as Visa or MasterCard. The name should match the credit card type so that you can identify the card without memorizing the credit card type codes.

**Credit Card Number Length** Enter the card's standard credit card number length. Before transmitting a request to CyberSource or another vendor, the system validates the credit card number length against this number.

**Credit Card Status** Select *Active* if you accept this type of credit card. Select *Inactive* if you don't accept this type of credit card. Inactive credit card types do not appear on the Authorize Credit Card page.

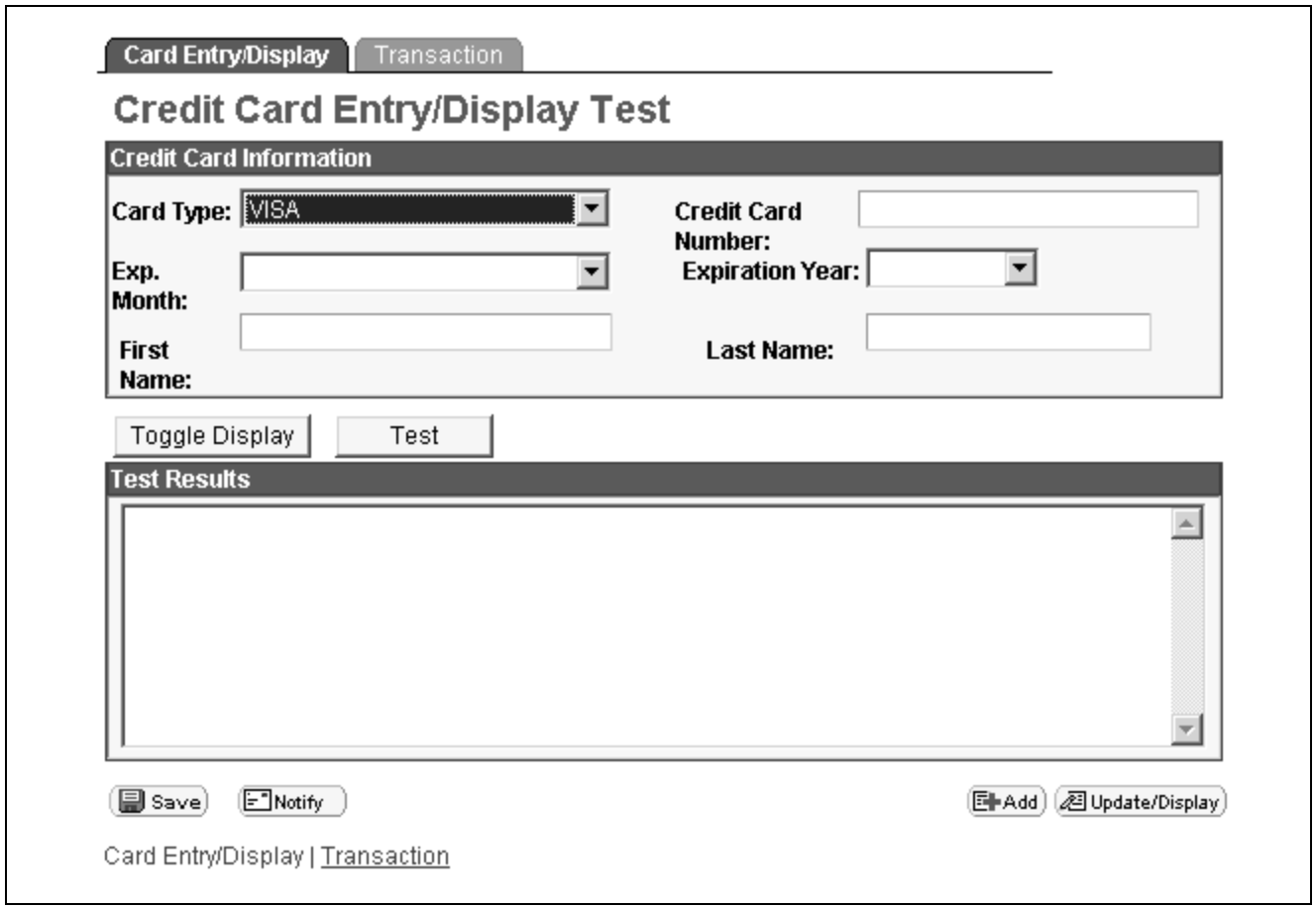
**Credit Card Valid Prefixes** Enter all valid prefixes for this type of credit card. Enter multiple prefixes in comma-separated format with no spaces in between. (The system removes

any characters other than numbers and commas when you move to another field.) Before transmitting a request to your vendor, the system validates that the credit card number starts with a valid prefix.

**Use Check Digit Algorithm** Select *Y* (yes) to use the MOD 10 Digit Check algorithm to validate credit card numbers before transmitting requests to your vendor. The MOD 10 Digit Check algorithm verifies whether card numbers you enter into the system are legitimate.

## Testing the Credit Card Interface

Access the Credit Card Entry/Display Test page.




Credit Card Entry/Display Test page

Use this page to test credit card processing and verify that it is working properly.

**Card Type** Select a card type to test. Only the allowed types are shown.

**Credit Card Number** Enter the credit card number that you're testing. The number you enter should match the card type you specified in the Card Type field.

**Exp. Month** (expiration month), **Expiration Year, First Name** , and **Last Name** Enter the information for the card that you're testing.

**Toggle Display** Click to switch between display-only and editable modes.

**Test** Click to begin the test.

**Test Results** The results of the test appear here. If the card number is valid, the message VALID CARD NUMBER appears; if the card number is not valid, a message explaining why appears (for example, the card number is incorrect, the card is expired, and so on.)

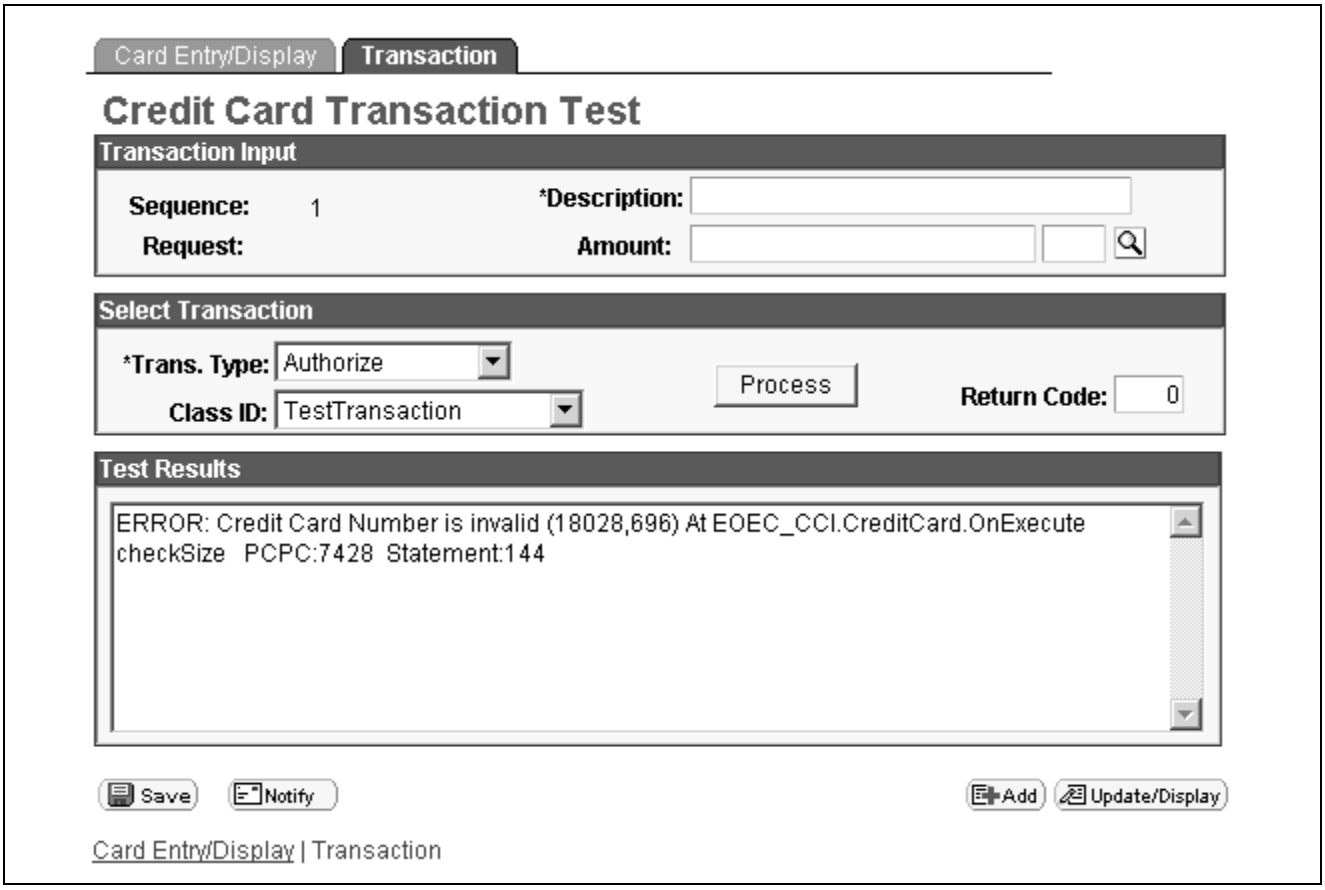
### Credit Card Sample Data

You can use the following credit card sample data in test transactions:

| Credit Card Type | Credit Card Number |
|---|---|
| Visa | 4111111111111111 |
| MasterCard | 5555555555554444 |
| American Express | 378282246310005 |
| Discover | 6011111111111117 |

## Testing Credit Card Transactions

Access the Credit Card Transaction Test page.




Credit Card Transaction Test page

Use this page to test transactions.

**Sequence** and **Request** — Displays a number that distinguishes this transaction from other transactions. It is similar to a run control or job number.

**Amount** — Enter an amount and select the look up icon to set the currency.

**Trans. Type** (transaction type) — Choose a transaction type to test:

*Authorize Only:* Verify that the card is valid for the charge.

*Bill Only:* Bill the card without first verifying that the card is valid for the charge.

*Authorize and Bill:* Perform both authorization and billing during the same transaction.

*Credit Only:* Credit the customer's credit card.

**Class ID** — Select either *Test Transaction* (the default) or select the type of interface that has been specified in the installation options. *ProcessBrokerTransaction* corresponds to an XML-compliant interface, while *Interlink Transaction* corresponds to a CyberSource compliant interface.

**Process** — Click to process the transaction.

**Return Code** Enter a return code to test whether proper error messages and results are returned.

**Test Results** The results of the test appear here. If the card number is valid, the message VALID CARD NUMBER appears; if the card number is not valid, a message explaining why appears (for example, the card number is incorrect, the card is expired, and so on.)

## Credit Card Transaction Request Message

This process connects to the credit card processor and processes messages regarding the success of the connection. If there is an error in the process or in the passed transaction parameters, the process returns one of the following error codes in the Test Results area.

| Return_status | Description |
|---|---|
| -3 | Error Opening Trace File |
| -4 | Vendor Error − ICS_INIT failed |
| -5 | Unsupported Service |
| -6 | Credit card number is too long |
| -7 | Phone number is too long |
| -8 | State field length is invalid |
| -9 | Zip Code field is too long |
| -10 | Amount must be greater than zero |
| -11 | Vendor Error − ICS_SEND failed |
| -12 | Decryption Failed |
| -15 | Request ID is required |
| -16 | Currency is required |
| -17 | Phone is required |
| -18 | Email ID is required |

| Return_status | Description |
|---|---|
| -19 | Zip Code is required |
| -20 | City is required |
| -21 | Country code is required |
| -23 | Address 1 is required |
| -99 | Trace Run Only |

## Credit Card Transaction Response Message

The credit card processor sends a return code and status message to the requesting PeopleSoft application.

The format is:

The Return_status is a three-digit numeric code with these possible values:

- 0 indicating a successful transaction.
- Any negative number indicating an error.
- 100-199 meaning declined by vendor but OK to display to online users.
- 200-299 meaning declined by vendor and NOT OK to display to online users.

The Return_status_msg is a string with these possible values:

- PeopleSoft Error - trapped prior to transmission.
- Approve - transaction approved.
- Decline − transaction declined.
- Connection Error.
- Unrecoverable Error.

## Credit Card Interface Messages

The messages listed in the table generate XML that is used to send data back and forth between the credit card interface and the third-party XML-compliant vendor’s API.

These messages are:

| Message Name | Description |
|---|---|
| EOEC_CCI_SYNC | Synchronous transaction request that the credit card interface sends to the third-party vendor. The request can be for an authorization, bill, or credit transaction. |
| EOEC_CCI_RESPONSE | The response to the request the credit card interface receives from the third-party vendor. |

# Configuring the Java Interlink Plug-In

PeopleSoft delivers a Java interface to your third-party credit card authorization and payment application.

This section provides an overview of the Java interlink plug-in and discusses how to:

- Set up the CyberSource API and Java plug-in on Microsoft Windows NT systems.
- Set up the CyberSource API and Java plug-in on UNIX systems.

## Understanding the Java Interlink Plug-In

Using PeopleSoft Business Interlink technology, the Java interface supports four kind of transactions:

- Authorize-only.
- Bill-only.
- Authorize-and-bill.
- Credit.

Use the plug-in to interface with the CyberSource credit card processing services product. The business interlink Java plug-in (delivered with your PeopleTools installation) uses the CyberSource Java API, version 3.5.0.0. Before you configure your system, obtain the CyberSource Java API. You must obtain the Java version of the API even if you are already using the C++ plug-in and API.

See the CyberSource website, www.cybersource.com

Download the zipped API files and use the instructions in this section to configure the environment.

## Setting Up the CyberSource API and Java Plug-in on Microsoft Windows NT Systems

This section discusses how to:

- Set up the business interlink environment.
- Set up the CyberSource API.
- Set up the plug-in.

### Setting Up the Business Interlink Environment

To set up the business interlink environment:

1. Ensure that two entries in the PATH environment variable point to the Java Run-Time Environment (JRE) bin.

   Do not confuse the JRE bin with the Java Development Kit (JDK) bin. The two entries in the PATH environment variable should point to the*JRE home*\jre\bin and *JRE home directory*\jre\bin\classic.

2. Create an entry in the CLASSPATH environment variable.

   Point the entry in the CLASSPATH environment variable to the psinterlinks.jar file. The psinterlinks.jar file is located in *PS Tools*\class.

   You must include the name of the .jar file in the CLASSPATH entry; for example, c:\pstools\class\psinterlinks.jar.

### Setting Up the CyberSource API

To set up the CyberSource API:

1. Install the CyberSource API using CyberSource instructions.
2. Follow the CyberSource installation directions for making CLASSPATH entries.
3. Update the CyberSource properties file for your CyberSource environment.

   Update the file *CyberSource Home*\cdkjava12-3500\properties\ICSClient.props to point to the key and certificate files located in the keys directory. Use a forward slash—not a back slash—to separate the directories, as follows:

   - For the private key file: myPrivateKey=C:/opt/ics/keys/*filename.pvt*
   - For the file with the location of your certificate: myCert=C:/opt/ics/keys/*filename.crt*
   - For the file with the location of the CyberSource certificate: serverCert=C:/opt /ics/keys/CyberSource_SJC_US.crt

4. Run the Java class ICSAuthTest.class.

   The Java class ICSAuthTest.class is located in the directory where you installed CyberSource. You must run this to ensure the API is set up properly. To run the ICSAuthTest.class:

   a. Edit the ICSAuthTest.java code to change the expiration to a future date.

   b. Open the Windows command prompt and change drives and directories to point to the test directory.

   c. Type *Javac ICSAuthTest.java* to recompile the edited file.

   d. Type *Java ICSAuthTest* to run the test.

   e. Verify that the Java class ICSAuthTest.class ran successfully.

### Setting Up the Plug-in

To set up the plug-in:

1. Move the pscreditcard.jar file to the desired directory.

   The pscreditcard.jar file is installed in *PS Tools*\class. You can move this .jar file to another directory for more stringent security access. The .jar file operates with execute permission only.

2. Update the CLASSPATH variable to point to the pscreditcard.jar file.

   Include the file name in the CLASSPATH entry.

3. Open PeopleSoft Application Designer, and select File, New, Business Interlink.

4. Select pscreditcard.dll.

5. Verify that the text that appears in the Version field in the new Business Interlink dialog box begins with *plugin.*

   If the text reads *Failed to initialize JVM*, your PATH environment variable is not pointing to the JRE directories. Ensure that your PATH points to both the JRE bin and classic directories; for example: *JRE Home*\jre\bin and *JRE Home*\jre\bin\classic.

6. If you are using three-tier architecture while accessing this plug-in, update the psappsrv.env file.

   To update the psappsrv.env file:

   a. Locate the psappsrv.env file in your application server domain folder (*PS Tools\appserv\your domain*\ psappsrv.env), and open it in a text editor.

   b. Append the path to your JRE \bin\classic directory to the end of the PATH variable in psappsrv.env.

   c. Ensure that there is a semicolon between path entries.

# Setting Up the CyberSource API and the Java Plug-in on UNIX Systems

This section discusses how to:

- Set up the business interlink environment.
- Set up the CyberSource API.
- Set up the plug-in.
- Configure the application server.
- Set up the Microsoft Windows client plug-in.

## Setting Up the Business Interlink Environment

To set up the business interlink environment:

1. Ensure that pspkgs.sh was run as part of the installation process.

2. Verify that the file psinterlinks.jar file exists in *PS Tools*/class.

   If psinterlinks.jar does not exist, pspkgs.sh may have been run incorrectly.

3. Verify that the Java Virtual Machine is installed under the Tools home directory.

   If the Java Virtual Machine is not installed, pspkgs.sh may have run incorrectly.

4. Locate psconfig.sh, and open it in a text editor.

5. Ensure that there is an entry in the PATH environment variable in psconfig.sh pointing to the JRE bin.

   Do not confuse the JRE bin with the JDK bin. The entry in the PATH environment variable should point as follows: PATH=$PS_JRE/bin:$PATH;export PATH

6. Create an entry in the PS_CLASSPATH environment variable in psconfig.sh.

   Point the entry to the psinterlinks.jar file located in *PS Tools*/class. You must include the name of the .jar file in the PS_CLASSPATH entry; for example, /pstools/class/psinterlinks.jar.

### Setting Up the CyberSource API

Complete these steps for application servers.

To set up the CyberSource API:

1. Install the CyberSource API using the CyberSource documentation.
2. Update CLASSPATH and PS_CLASSPATH in psconfig.sh to point to the API .jar file.

   The API .jar file is cdkjava3500.jar.
3. Update CLASSPATH in psconfig.sh to point to the test directory under the API's cdkjava118-3500 directory.
4. Save your changes to psconfig.sh.
5. Update the CyberSource properties file.

   Update the file *CyberSource Home*/cdkjava118-3500/properties/ICSClient.props to point to the key and certificate files located in the keys directory. Be sure to use a forward slash—not a back slash—to separate the directories, as follows:

   - For the private key file: myPrivateKey=C:/opt/ics/keys/*filename.pvt*
   - For the file with the location of your certificate: myCert=C:/opt/ics/keys/*filename.crt*
   - For the file with the location of the CyberSource certificate: serverCert=C:/opt /ics/keys/CyberSource_SJC_US.crt
6. Run psconfig.sh to set the properties.
7. Run the Java class ICSAuthTest.class.

   The Java class ICSAuthTest.class is located in the directory where you installed CyberSource. You must run this class to ensure that the API is set up properly.

   Before you run the test, edit the ICSAuthTest.java code to change the expiration to a future date and then recompile ICSAuthTest.java by typing *Javac ICSAuthTest.java* at the prompt.

---

**Note.** You do not need to make a system-level CLASSPATH entry in addition to the CLASSPATH entry in psconfig.sh. If the Java class ICSAuthTest.class runs successfully, you can remove the CLASSPATH references to the test directory and the cdkjava3500.jar file.

---

### Setting Up the Plug-in

Complete these steps for application servers and PeopleSoft Process Scheduler servers.

To set up the plug-in:

1. Confirm that the pscreditcard.jar file is located in *PS Tools*/appserv/classes.

   You can move this .jar file to another directory to tighten security and limit access. If you move this .jar file to a different location, ensure that you reference the new location in the following steps.
2. Update PS_CLASSPATH in psconfig.sh to point to the pscreditcard.jar file in the classes directory.

Include the file name in the entry.

3. Confirm that the libpscreditcard file is in the *PS Tools*/bin/InterfaceDrivers directory.

**Note.** File extensions may be different for your UNIX platform. Use the same naming convention as the libpsjavaproxy file for the libpscreditcard file.

4. Stop and restart the application server.

   Stop the services; do not stop the machine.

### Configuring the Application Server

To configure the application server:

1. Copy the ICSClient.props file from the CyberSource properties directory to the application server domain directory.
2. Open the psappsrv.env text file that is located in your application server (not the web server) domain folder.
3. Append the path to your JRE/bin/classic directory to the end of the PATH variable.

   Ensure that there is a semicolon between path entries. You can find the JRE path in your PATH system variable.
4. Shut down and restart your application server.

### Setting Up the Microsoft Windows Client Plug-in

To set up the Microsoft Windows client plug-in, verify that the pscreditcard.xml file is in the client/InterfaceDrivers directory.

# Configuring the PeopleSoft Application Designer Plug-in

This section discusses how to:

- Configure access to account information.
- Configure proxy server support.
- Set up tracing.
- Check parameter check logic.
- Set the decryption option.

**Note.** All credit card transactions are supported through one business interlink transaction.

## Configuring Access to Account Information

The following configuration parameters are needed to complete any transaction:

- Merchant ID
- Private key file

- Certificate file
- Server ID
- Server certificate
- Server URL

The plug-in offers the following three configuration options:

- Maintain the six configuration parameters in a properties file.

  Use this option if you are building a new credit card processing application.

- Pass the merchant ID and server URL as input parameters, and specify the path of the key and certificate files.

  Use this option if you have an existing application that has used the C++ plug-in, but which you are now porting to UNIX.

- Use this plug-in just as you use the C++ version.

  The plug-in gets the certificate and key files from c:\opt\ics\keys. Use this option if you have been using the C++ plug-in and want to continue to operate on a Microsoft Windows platform.

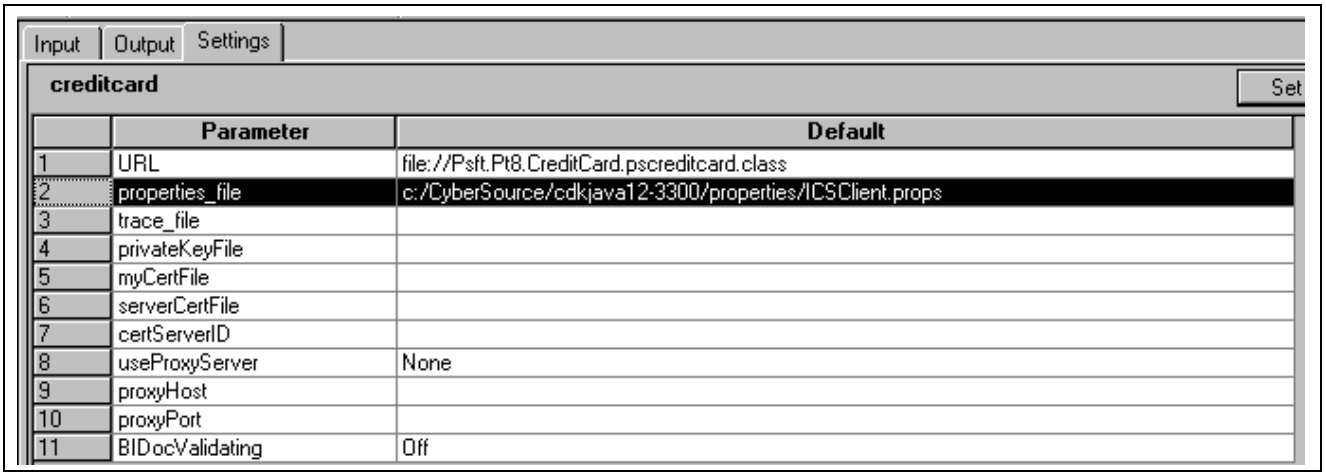
---

**Note.** The following configuration options demonstrate default input parameter values. After setting up a transaction, set the input parameter values as default values to test them in the Business Interlink Tester. Typically, PeopleCode sets these input parameter values at runtime. In PeopleSoft Application Designer, select File, Open, Business Interlink to open a business interlink transaction. Define the configuration parameters for your credit card business interlink definition on the Settings and Input pages.

---

## Maintaining Configuration Parameters in a Properties File

The following page shows settings for maintaining parameters in a properties file:




Settings page for keeping configuration parameters in a properties file

To maintain configuration parameters in a properties file:

1. Update the CyberSource properties file.

   Update the file (*CyberSource Home*/cdkjava12-3500/properties/ICSClient.props) to point to the key and certificate files located in the keys directory. Use a forward slash to separate directories, as follows:

   - For the private key file: myPrivateKey=C:/opt/ics/keys/*filename.pvt*

- For the file with the location of your certificate: myCert=C:/opt/ics/keys/*filename.crt*
- For the file with the location of the CyberSource certificate: serverCert=C:/opt /ics/keys/CyberSource_SJC_US.crt

2. Set the value of the properties_file configuration parameter to the path of the ICSClient.props file.

   In PeopleSoft Application Designer, select the Settings tab to set the value. Use a forward slash to separate directories.

3. Leave the privateKeyFile, myCertFile, serverCertFile, and certServerID fields blank.

### Passing the Merchant ID and Server URL as Input Parameters and Specifying the Path of the Key and Certificate Files

To pass the merchant ID and server URL as input parameters and to specify the path of the key and certificate files:

1. Delete the value of the properties_file config parameter on the Settings page.

   Enter the paths for the key and certificate files in their appropriate configuration parameters. Use a forward slash to separate directories. The default value for the certificate server ID is *CyberSource_SJC_US.*

2. Include the merchant and server_host input parameters in your transaction.

   Do not include the text *http://* in the server URL string.

### Using the Plug-in as You Use the C++ Version

To use the Java plug-in as you use the C++ version:

1. Delete the values of the properties_file, certificate, key, and server ID field parameters on the Settings page of the business interlink definition.
2. Use the merchant and server_host field input parameters in your transaction.

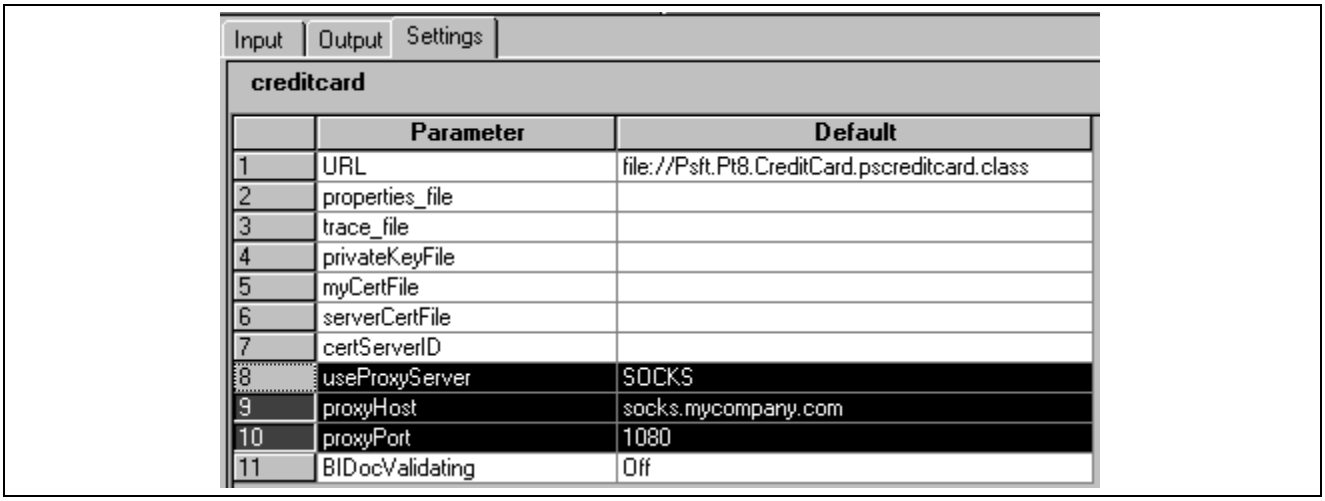
   Do not include the text *http://* in the server URL string.

## Configuring Proxy Server Support

The Java plug-in supports the HTTP proxy without a username and password, and it supports the SOCKS proxy protocols.

See CyberSource ICS2 developers guide.

The following illustration shows the settings for configuring proxy support:




Settings page for implementing proxy server support

To implement proxy server support:

1. Select the proxy type from the useProxyServer configuration parameter.
2. Enter the values of the host and port in the proxyHost and proxyPort configuration parameters.

# Setting Up Tracing

The plug-in and the CyberSource API support the use of a trace file. To keep a log of the transactions, use the plug-in's trace file and reserve the API's trace file for debugging. The plug-in trace file:

- Uses the input names as they appear in PeopleSoft Application Designer, not CyberSource API property names.
- Prints only the last four digits of the credit card number.
- Prints output values and any error messages.

## Activating Tracing

There are two methods for activating tracing in the Java plug-in:

- Leave the trace_file blank and use the trace input parameter to toggle tracing.

  The trace file writes to c:\temp\crcard.txt.

- Enter a path and file name in the trace_file configuration parameter, and use the trace input parameter to enable the tracing (trace=*Y*) or disable the tracing (trace =*N).*

  Use this method for new credit card definitions, for porting existing credit card definitions to UNIX, or for writing the trace file to a directory other than c:\temp. Use forward slashes when specifying directory paths.

## Activating Trace Run-Only Mode

To see the input values that are sent to the CyberSource API and to confirm that the plug-in can instantiate the CyberSource interface without sending the transaction to CyberSource, pass a value of *T* with the trace input parameter. If a trace file has been specified in the trace_file configuration parameter, the system uses that trace file. Otherwise, the system uses the c:\temp\crcard.txt trace file.

## Setting Parameter Check Logic

If the paramchk parameter is set to *Y,* the plug-in runs in trace run-only mode and performs additional checks in the input parameters to ensure that the parameters do not exceed maximum allowed field lengths. The system sends the transaction information to the CyberSource API.

## Setting the Decryption Option

Credit card numbers can be sent encrypted to the plug-in. Do not change this option unless you also change the PeopleCode to pass the credit card number as plain text.

# Setting Transaction Inputs and Outputs

If you create a new business interlink definition, you need to set inputs and outputs.

This section provides an overview of required parameters and discusses how to:

- Use authorize-and-bill, authorize-only, and bill-only transactions.
- Identify input fields that are not used or not supported.

## Understanding Required Parameters

The service and merchant_ref input parameters are required in all credit card transactions. The method that you use to configure the plug-in determines which fields are required.

### Service Input Field

The system uses the service parameter to determine which of the four credit card transactions it should conduct through business interlink transactions. The service parameter values perform the following functions:

| Value | Function |
|---|---|
| 1 | Authorize Only |
| 2 | Authorize and Bill |
| 3 | Bill Only |
| 4 | Credit |

### Merchant_ref Input Field

The merchant_ref field is a merchant-generated order reference or tracking number, which contains information such as an order number. Use the merchant_ref_number field as your own tracking number for the order, to keep track of requests sent to CyberSource relating to the same order.

## Using Authorize-and-Bill Transactions

The authorize-and-bill transaction (service=*2)* performs both authorize and bill actions and returns a bill amount for verification of the amount billed.

## Using Authorize-Only and Bill-Only Transactions

The authorize-only transaction (service=*1)* authorizes the amount passed in and returns a request ID to be used later in a bill transaction.

The bill-only transaction (service=*3)* requires the use of a request ID obtained from an authorize-only transaction.

Required inputs for bill-only are: Service, Merchant_ref, Amount, Currency, and Rqstid.

Outputs are: return_status, Return_status_msg, Ret_msg1, Ret_msg2, Ret_authcd, Ret_authdttm, Rqstid_out, Bill_amount, and Trans_ref_no.

---

**Note.** PeopleSoft Support uses all transaction types; PeopleSoft Order Capture uses just the authorize-only transaction.

---

## Address Verification Service

CyberSource returns the decline code *DAVSNO* if the billing address does not match the billing address for the credit card. If you set the value of the avs_ignore field to *yes* and include it in your business interlink definition, the system allows the authorization transaction to proceed despite the billing address discrepancy. If the value of the avs_ignore field is anything other than *yes*, the address verification proceeds as usual.

The address verification service returns a value that is stored in the ret_avs field. The values are:

| Value | Description |
|---|---|
| A | The street number matches, but the 5-digit ZIP code and 9-digit ZIP code do not match. |
| E | Address verification data is invalid. |
| N | The street number, 5-digit ZIP code, and 9-digit ZIP code do not match. |
| R | System unavailable. |
| S | Issuer does not offer address verification. |
| U | Address information unavailable. This code occurs if a customer tries to use an international card, or if the address verification in United States banks is not functioning properly. |

| Value | Description |
|---|---|
| W | The street number does not match, but the 5-digit ZIP code and 9-digit ZIP code match. |
| X | Exact match. The street number, 5-digit ZIP code, and 9-digit ZIP code match. |
| Y | The street number and 5-digit ZIP code match. |
| Z | The 5-digit ZIP code matches. |

## Fraud Screen Service

Both the authorize-only and the authorize-and-bill transactions support the fraud screen transaction.

See CyberSource documentation.

**Fraud_screen** Determines if the fraud screen transaction request should be sent with the current transaction.

**Score_criteria_modified** Determines if fraud screen score tuning parameters were altered for this transaction. If this field is set to *yes*, all of the following score tuning parameter fields must be included in the transaction:

Required inputs: Service, Merchant_ref, Fname, Lname, Addr1, City, Country, State, Zip, Email, Phone, Expmo, Expyr, Ccnum, Amount, Currency.

Output: Return_status, Return_status_msg, Ret_msg1, Ret_msg2, Ret_authcd Ret_authdttm, Rqstid_out, Bill_amount (auth/bill trans).

## Credit Transactions

Use one of the following methods to perform a credit transaction:

- Use a request ID.

  This option enables you to use the value of a request ID returned from a previous request for bill transactions. This value is matched with a previous bill for the same order. If this field is present and there is no matching bill, the transaction fails. If the bill request ID is not present, the credit transaction requires the customer billing information.

  Required inputs are: Service, Merchant_ref, Amount, Currency, and Rqstid.

  Outputs are: Return_status, Return_status_msg, Ret_msg1, Ret_msg2, Ret_authcd, Ret_authdttm, Rqstid_out, Credit_amount, and Trans_ref_no.

- Use billing information.

  Required inputs are: Service, Merchant_ref, Fname, Lname, Addr1, City, Country, State, Zip, Email, Phone, Expmo, Expyr, Ccnum, Amount, and Currency.

Output: Return_status, Return_status_msg, Ret_msg1,Ret_msg2, Ret_authcd, Ret_authdttm, Rqstid_out, and Credit_amount.

## Identifying Input Fields That Are Not Used or Not Supported

The following input fields are not used or supported as of 11/06/00:

| Input Field | Comment |
|---|---|
| Cctype | Not used by CyberSource. |
| Ip_address | Proxy server support is included. The Java CDK supports use of both HTTP and SOCKS proxy |
| Extra1 | N/A |
| Extra2 | N/A |

# PART 4

# Portal Utilities

# CHAPTER 8

# Working with Navigation Pages

This chapter provides an overview of navigation pages and discusses how to:

- Enable the display of standard navigation pages.
- Enable navigation page display caching

## Describing Navigation Pages

In addition to menu navigation, PeopleSoft applications include navigation pages, which serve as alternatives to menu navigation. These navigation pages provide a user-friendly navigation tool in the form of task-driven pages that provide intuitive access to pages needed to complete your business processes. These navigation pages can be configured to incorporate the use of icons to further increase the user's ability to intuitively navigate their tasks.

There are two types of navigation pages:

- Standard navigation pages.
- Custom navigation pages.

Navigation pages share these common elements:

**Breadcrumbs**

Your portal administrator may configure your navigation pages to display breadcrumbs. These breadcrumbs display across the top of a navigation page and are generated according to the navigation path used to access the current content displayed. You can select the breadcrumbs to navigate through the path you used to access the current content.

See Chapter 9, "Setting Up Navigation Collection Options," page 105.

**Edit <folder name> Folder** or **Edit <Navigation Collection name> Collection**

The Edit <folder name> Folder link displays if you have accessed a standard navigation page and you have permissions defined for your user ID or role that enable you to access the Structure and Content component.

Select the Edit <folder name> Folder link to access the Folder Administration page, where you can edit aspects of the folder that was used to generate the selected standard navigation page.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Defining Folders

The Edit <Navigation Collection name> Collection link displays if you have accessed a custom navigation page and you have permissions defined for your user ID or role that enable you to edit Navigation Collections.

Select the Edit <Navigation Collection name> Collection link to access the Navigation Collections component, where you can edit the definition that was used to build the selected custom navigation page.

See Chapter 10, "Using Navigation Collections," Creating and Maintaining Navigation Collections, page 125.

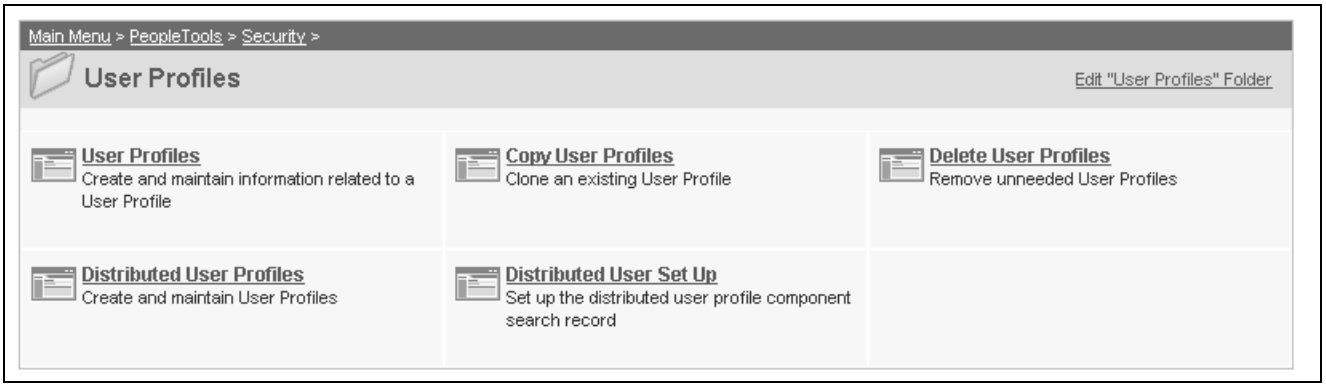
## Describing Standard Navigation Pages

You access standard navigation pages by clicking a folder in the menu navigation. Elements embedded in the folder, including content references and other folders, are arranged in a graphical format on a standard navigation page displayed to the right of the menu navigation.

For example, clicking the Security folder in the menu navigation displays the Security standard navigation page as shown here:


Security standard navigation page

When you click a folder link on a standard navigation page, you access another standard navigation page that reflects the contents of the selected folder. For example, when you click the User Profiles folder, the User Profiles standard navigation page displays as shown here:




User Profiles standard navigation page

When you click a content reference link on a standard navigation page, you access the actual page to which the content reference points.

Standard navigation pages are available for every folder that appears in your menu navigation that has also been configured to display a standard navigation page. You enable the display of standard navigation pages at the registry level on the General Settings page. You can then disable the display of standard navigation pages for specific folders in the registry on the Folder Administration page.

See Chapter 8, "Working with Navigation Pages," Enabling Standard Navigation Page Display, page 102.

These pages are dynamically generated based on folder and content reference sequence numbers defined in your portal registry, as well as settings defined by your portal administrator in the Navigation Collection System Options and Registry Options components.

See Chapter 10, "Using Navigation Collections," Describing Folder and Link Sequence on Navigation Pages, page 121.

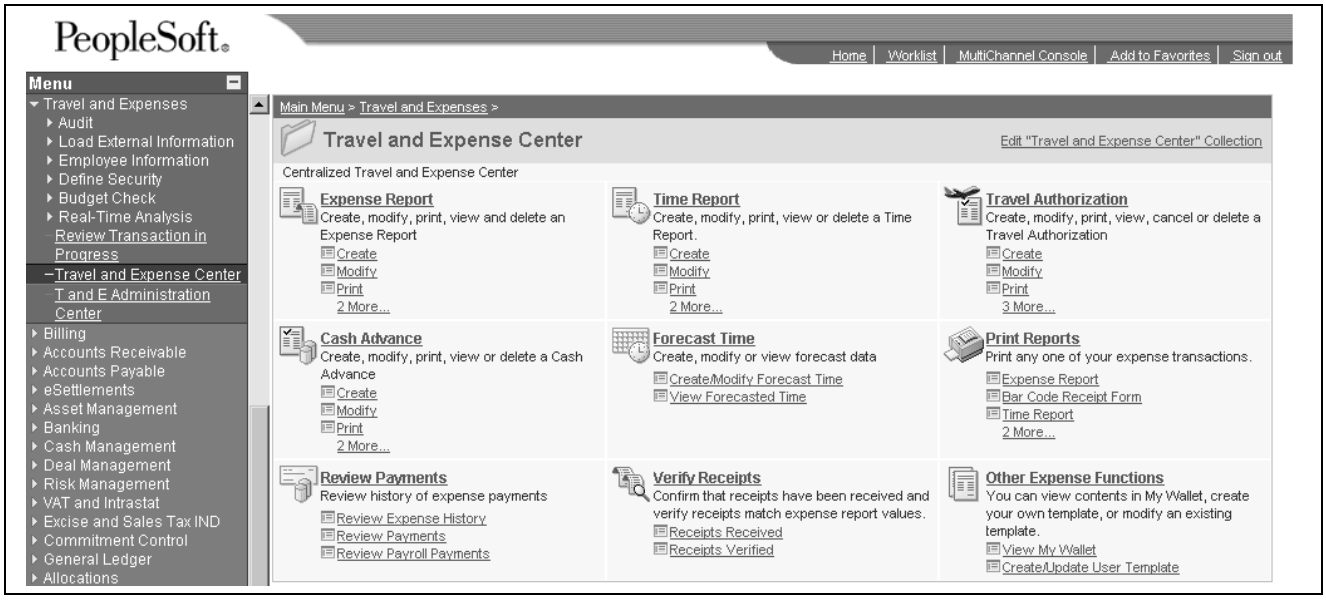
See Chapter 9, "Setting Up Navigation Collection Options," page 105.

## Describing Custom Navigation Pages

You access custom navigation pages by way of links in the menu navigation. These custom navigation pages have been designed to facilitate your navigation through and completion of tasks and business processes in your PeopleSoft applications.

While standard navigation pages automatically display only the contents of a selected menu folder, custom navigation pages can be designed to display any content references or folders available in the menu navigation, regardless of their location.

When you click a menu navigation link to a custom navigation page, the page displays to the right of the menu navigation. For example, selecting the Travel and Expense Center link in the menu navigation displays the Travel and Expense Center custom navigation page as shown here:




Travel and Expense Center custom navigation page

When you click a folder link on a custom navigation page, you access another navigation page that reflects the contents of the selected folder.

When you click a content reference link on a custom navigation page, you access the actual page to which the content reference points.

Custom navigation pages are delivered to address commonly used tasks and business processes in your PeopleSoft application. Authorized users can edit delivered custom navigation pages, as well as create new ones tailored to suit the business process needs of your organization.

See Chapter 9, “Setting Up Navigation Collection Options,” page 105.

# Enabling Standard Navigation Page Display

In this section, we discuss how to enable the display of standard navigation pages when you click a menu navigation folder.

To enable the display of standard navigation pages:

1. If you are enabling the display of standard navigation pages for the registry in the portal you are logged into, access the General Settings page. If you are enabling the display of standard navigation pages for the registry in a portal you are not logged into, access the Portal Definitions - Edit Portal page.
2. Select the Is Folder Navigation Enabled option.
3. Select the delivered Base Navigation Page content reference (EOPP_SCFNAV_BASEPAGE_SCR) in the Folder Navigation Object Name field. This Base Navigation Page content reference is designed to display standard navigation pages as they are documented in this chapter.
4. You can disable the display of a standard navigation page for an individual folder in the selected registry by selecting the Is Folder Navigation Disabled option on the Folder Administration page.

5. You can designate a content reference for an individual folder that overrides the content reference defined at the registry level by selecting a value in the Folder Navigation Object Name field on the Folder Administration page. The content reference you designate as the folder navigation object is launched when the folder label is selected.

### See Also

*PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Managing General Portal Settings

*PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Administering Portal Definitions

# Enabling Navigation Page Display Caching

You may choose to enable the caching of displayed navigation pages. Enabling caching improves performance because the next time the same navigation page is accessed from any computer using the same web server, the navigation page is retrieved from the web server cache memory. To clear the web server cache memory, stop and restart the web server.

Methods for enabling caching are slightly different for each of the following methods you can use to access navigation pages.

## To enable caching for a custom navigation page accessed by clicking a content reference link:

See Chapter 8, “Working with Navigation Pages,” Describing Custom Navigation Pages, page 101.

1. Select the Enable Caching option on the Publish Collection page when you publish the custom navigation page for which you want caching enabled.

   See Chapter 10, “Using Navigation Collections,” Publishing Navigation Collections, page 144.

2. Upon publication, the Cache Enabled Value field value you entered on the System Options page is added to the published custom navigation page registry structure definition.

   See Chapter 9, “Setting Up Navigation Collection Options,” page 105.

## To enable caching for folder or breadcrumb on a navigation page:

1. Select the Enable Drill Down Cache option on the System Options page.
2. Enter a value in the Cache Enabled Value field (*role,max-age=30,* for example) on the System Options page.

## To enable caching for a standard navigation page accessed by clicking a menu navigation folder:

See Chapter 8, “Working with Navigation Pages,” Describing Standard Navigation Pages, page 100.

1. In the Structure and Content component, edit the Base Navigation Page content reference (EOPP_SCFNAV_BASEPAGE_SCR) on the Content Ref Administration page. The Base Navigation Page content reference is located in the Common Portal - Hidden folder.

   See *PeopleTools Internet Technology,* "Administering Portals," Administering Content References

2. In the Content Reference Attributes group box, enter the *PSCACHECONTROL* attribute name and an attribute value, such as *role,max-age=30*.

# CHAPTER 9

# Setting Up Navigation Collection Options

This chapter provides an overview of Navigation Collection options and discusses how set up these options.

## Describing the Navigation Collection Option Default Hierarchy

The following table provides a listing of available Navigation Collection options along with the levels at which they can be set and overridden. Options defined at the system level provide default values, with options defined at the registry, source, and Navigation Collection levels being capable of overriding each previous level.

We deliver a set of system-level defaults. These values can be changed on the System Options page or overridden at successive levels in the default hierarchy. However, these default values cannot be deleted as system-level default values are required.

| Options | System (default) | Registry (overrides system default) | Source Reference (overrides system and registry values) | Collection Reference (overrides system, registry, and source reference values) |
|---|---|---|---|---|
| Registry object prefix | X | | | |
| Show breadcrumbs | X | X | | |
| Show main menu breadcrumb | X | X | | |
| Registry node name | X | X | | |
| Large (parent) image | X | X | X<br>(with attribute name EPPSC_IMAGE) | X |
| Small (child) image | X | X | | |
| Style sheet | X | X | | X |
| Show images | X | X | | X |
| Maximum child links | X | X | | X |

| Options | System (default) | Registry (overrides system default) | Source Reference (overrides system and registry values) | Collection Reference (overrides system, registry, and source reference values) |
|---|---|---|---|---|
| Maximum columns | X | X | | |
| Owner ID | X | | | X |

System options are set on the System Options page.

See Chapter 9, “Setting Up Navigation Collection Options,” Defining Navigation Collection System Options, page 107.

Registry options are set on the Registry Options page.

See Chapter 9, “Setting Up Navigation Collection Options,” Defining Navigation Collection Registry Options, page 111.

Source reference options are set in the Structure and Content component.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals"

Navigation Collection reference options are set on the Maintain Collections page.

See Chapter 10, “Using Navigation Collections,” Creating and Maintaining Navigation Collections, page 125.

The default option to display standard navigation pages when you click a menu folder is set at the registry level on the General Setting page. You can override this default on the source reference.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Managing General Portal Settings

# Defining Navigation Collection Options

In this section, we discuss how to:

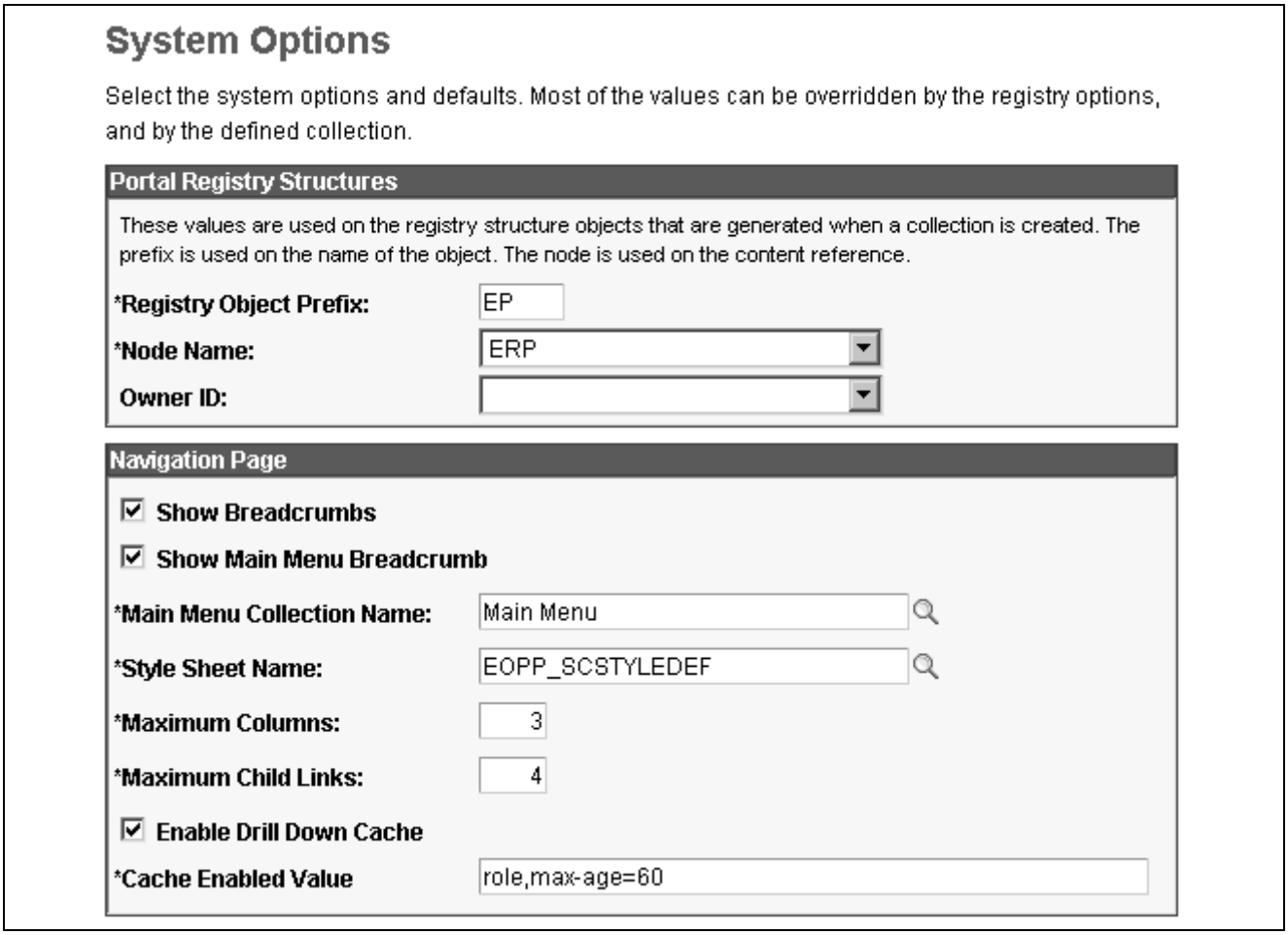
- Define Navigation Collection system options.
- Define Navigation Collection registry options.

## Pages Used to Define Navigation Collection Options

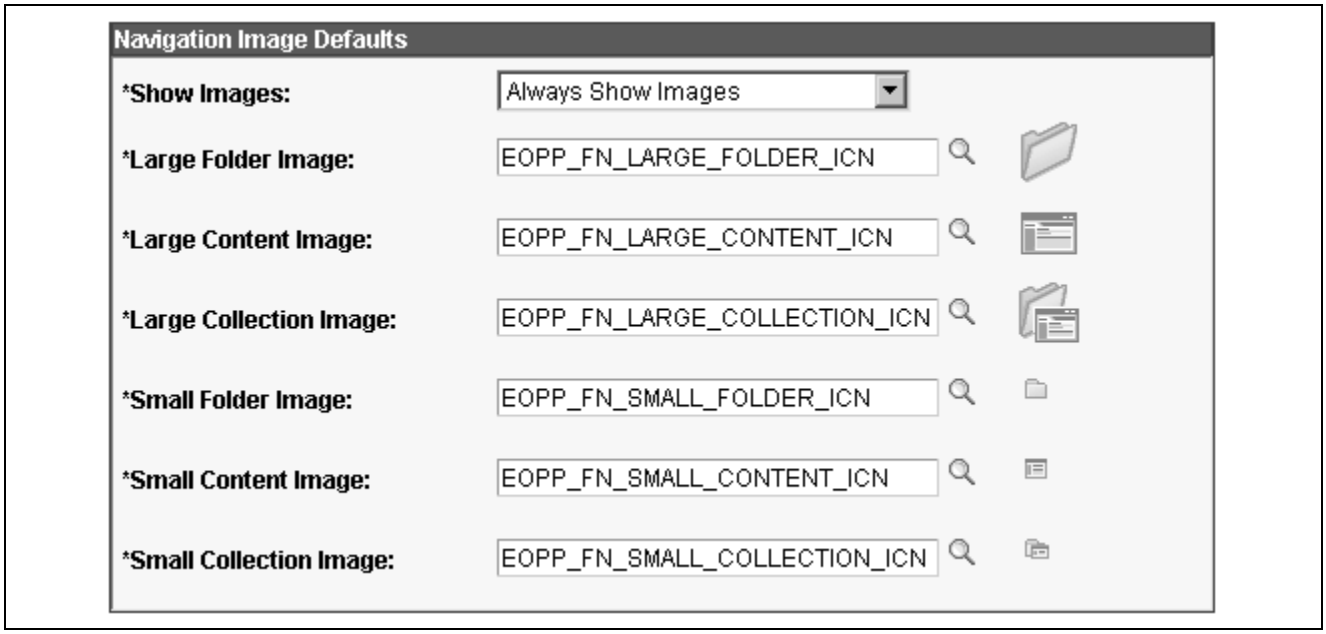
| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| System Options | EOPP_OPTIONS | Enterprise Components, Portal Utilities, System Options | Define system-level default values for Navigation Collection options. These defaults may be overridable at the registry, source reference, and Navigation Collection reference levels. |
| Registry Options | EOPP_SITE_OPT | Enterprise Components, Portal Utilities, Registry Options | Define registry-level values for Navigation Collection options. These values may override system-level defaults, as well as be overridable at the source reference and Navigation Collection reference levels. |

## Defining Navigation Collection System Options

Access the System Options page.




System Options page (1 of 2)




System Options page (2 of 2)

We deliver a set of system-level defaults. These values can be changed on this page or overridden at successive levels in the default hierarchy. However, these default values cannot be deleted as system-level default values are required.

See Chapter 9, "Setting Up Navigation Collection Options," Describing the Navigation Collection Option Default Hierarchy, page 105.

## Portal Registry Structures

**Registry Object Prefix** Enter the prefix you want the system to use for the generated registry structure object names (content and folder references) created by Navigation Collection definitions, as well as for the published Navigation Collection pages and pagelets.

For example, the object name of a published navigation pagelet will take on this format: `<prefix>_SC_PGT_<Navigation Collection label>`. The object name of a published navigation page will take on this format: `<prefix>_SC_SP_<Navigation Collection label>`.

**Node Name** You should not change the default value delivered with your system. This default value corresponds to the content provider node for your database. For example, the following databases are delivered with the following default node names:

PeopleSoft Financials/SCM: *ERP*

PeopleSoft CRM: *CRM*

PeopleSoft HCM: *HRMS*

PeopleSoft Enterprise Portal: *Local Node*

**Owner ID**

Use this value to easily identify Navigation Collections created by your organization when searching for Navigation Collections through PeopleSoft Application Designer or SQL queries on the portal registry table.

Do not select a delivered PeopleSoft owner ID. You can define unique owner IDs for your organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer .

## Navigation Page

**Show Breadcrumbs**

Select to indicate that you want breadcrumbs to display on Navigation Collection page publications and standard navigation pages.

These breadcrumbs reflect the path taken by the user as he navigates through left-navigation menu folders, Navigation Collection page folders, and standard navigation page folders. Displaying these breadcrumbs facilitates user navigation, as the left-navigation menu is not updated to reflect the user's navigation through Navigation Collection pages and standard navigation pages.

The first breadcrumb displayed is the Main Menu collection name link that points to the delivered EOPP_PORTAL_ROOT Navigation Collection, which is based on the PORTAL_ROOT_OBJECT folder. Alternatively, you can select a different Navigation Collection to be used as the first (main menu) breadcrumb using the Main Menu Collection Name field, or you can choose not to display a first breadcrumb.

The next set of breadcrumbs display dependent on the left-navigation menu folders the user opened to access the Navigation Collection page.

Subsequent breadcrumbs display dependent on the folders the user accesses on the Navigation Collection page.

**Show Main Menu Breadcrumb**

If you have selected the Show Breadcrumbs option, select this option to include the Main Menu collection name as the first entry in the breadcrumb path displayed on the Navigation Collection page publication.

If you clear this option, the first entry in the breadcrumb path is dependent on the first left-navigation folder the user opened to access the Navigation Collection page. The Main Menu link does not display as the first link in the breadcrumbs.

**Main Menu Collection Name**

This field displays when you have the Show Main Menu Breadcrumb option selected. The default value is set to *Main Menu,* which designates that the first breadcrumb displayed accesses the Main Menu page. You may choose to replace the first breadcrumb Main Menu link with a link that accesses an alternate Navigation Collection you have created.

The first breadcrumb Main Menu link to the Main Menu page provides access to the entire navigation for your database, filtered by security. You may want to provide a first breadcrumb link that provides access to a Navigation Collection that you have configured to contain only selected folders and content references.

**Style Sheet Name** Select the cascading style sheet (CSS) you want to use to generate Navigation Collection publications. Cascading style sheets are defined in PeopleSoft Application Designer.

This setting applies to both navigation pages and navigation pagelets.

See *PeopleTools 8.44 PeopleBook: PeopleSoft Application Designer,* "Creating Style Sheet Definitions"

**Maximum Columns** Enter the maximum number of columns you want to be displayed on a Navigation Collection publication page.

Navigation Collection pagelets display one or two columns depending on the user's settings on the Personalize Layout page. Narrow pagelets display contents in one column. Wide pagelets display contents in three columns.

This setting applies to only navigation pages.

See *Using PeopleSoft Applications PeopleBook*, "Setting User Preferences," Personalizing Your Homepage

**Maximum Child Links** Enter the maximum number of links you want to be displayed beneath each folder on the Navigation Collection publication. If the number of links in the folder exceeds this number, an X More... link appears beneath the folder that accesses a page from which you can access all links stored in the Navigation Collection folder.

For example, if you set the Maximum Child Links field value to *4* and the Navigation Collection folder contains five links, three links display on the Navigation Collection page along with the X More... link.

When you select the X More... link, a page containing all five links in the Navigation Collection folder displays.

This setting applies to both navigation pages and navigation pagelets.

**Enable Drill Down Cache** Select to enable caching for navigation pages accessed by selecting a folder on a navigation page. You must also enter a value in the Cache Enabled Value field on this page.

See Chapter 8, "Working with Navigation Pages," Enabling Navigation Page Display Caching, page 103.

**Cache Enabled Value** Enter the value that you want to assign to the PSCACHECONTROL attribute in the registry structure definition when enabling caching for applicable navigation page access methods. For example, you can enter a value such as *role,max-age=30*.

See Chapter 8, "Working with Navigation Pages," Enabling Navigation Page Display Caching, page 103.

## Navigation Image Defaults

**Show Images** *Always Show Images.* Select if you always want images to display on Navigation Collection publication pages and pagelets.

*Never Show Images.* Select if you never want images to display on Navigation Collection publications.

*Only Show Defined Images.* Select if you only want specifically defined images to display on Navigation Collection publications. If you select this option, only images defined as a part of the Navigation Collection or defined on the source reference are displayed on the Navigation Collection's publications. System- and registry-level default images are not displayed.

**Large Folder Image** Select the default image that you want to display for top-level folders on a Navigation Collection publication.

**Large Content Image** Select the default image that you want to display for top-level content references on a Navigation Collection publication.

**Large Collection Image** Select the default image that you want to display for top-level Navigation Collection references on a Navigation Collection publication.

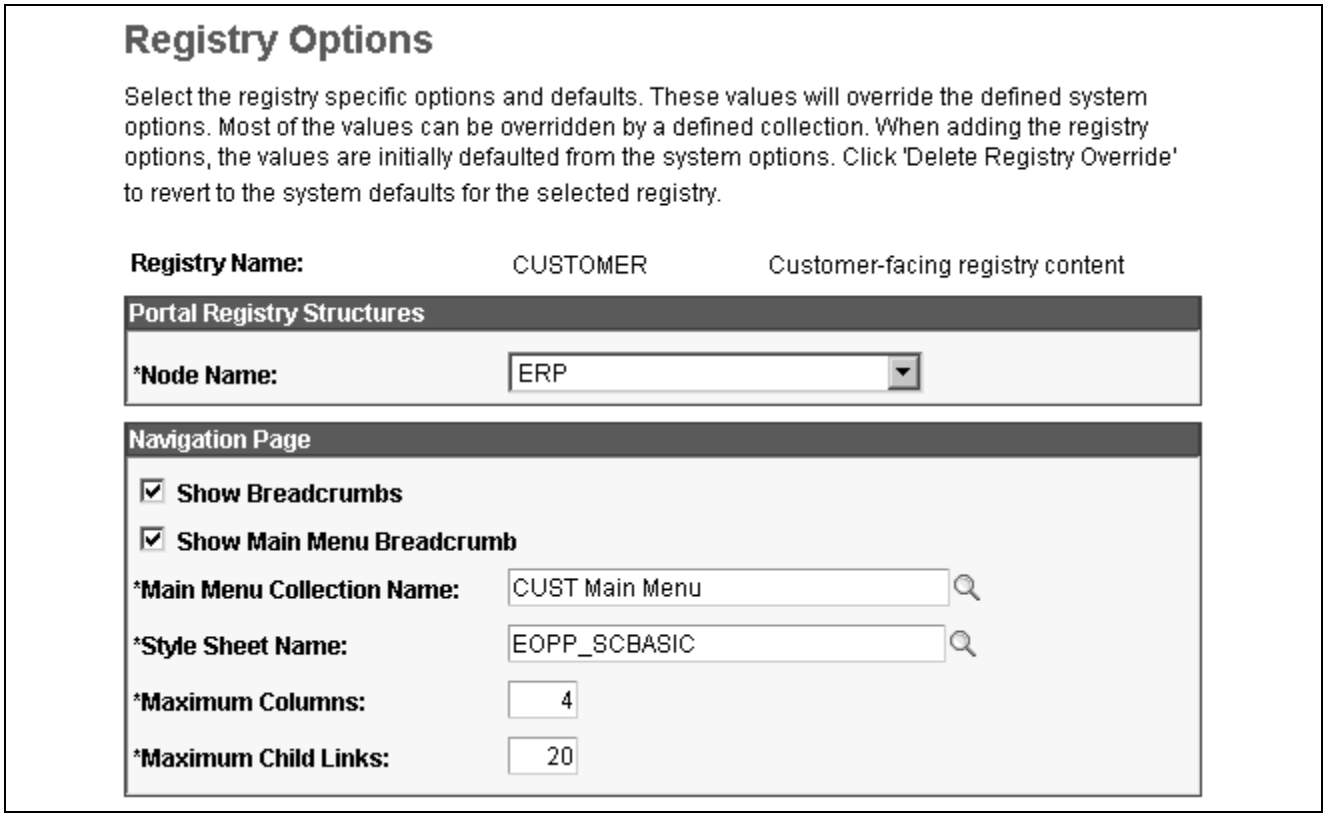
**Small Folder Image** Select the default image that you want to display for a folder within a folder on a Navigation Collection publication.

**Small Content Image** Select the default image that you want to display for a content reference within a folder on a Navigation Collection publication.

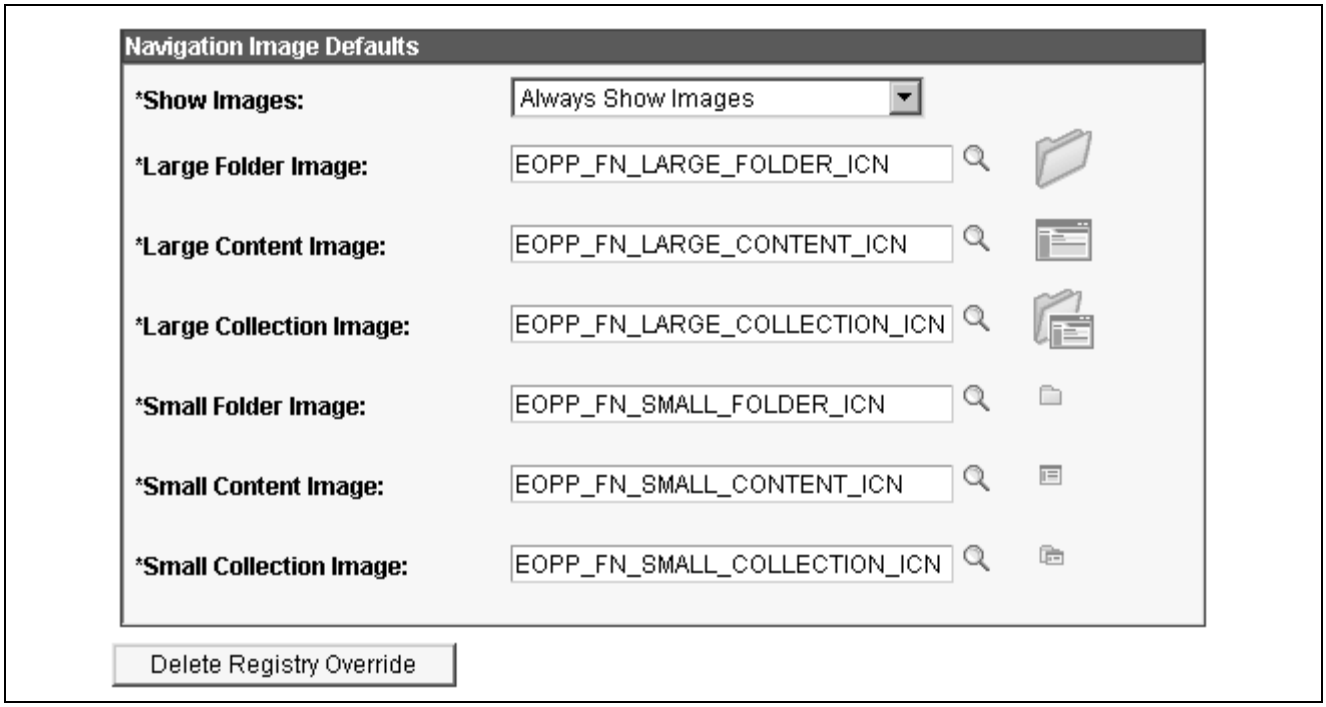
**Small Collection Image** Select the default image that you want to display for a Navigation Collection within a folder on a Navigation Collection publication.

## Defining Navigation Collection Registry Options

Access the Registry Options page.




Registry Options page (1 of 2)




Registry Options page (2 of 2)

All options you set on this page override defaults defined at the system level on the System Options page.

See Chapter 9, "Setting Up Navigation Collection Options," Describing the Navigation Collection Option Default Hierarchy, page 105.

**Registry Name** Select the portal registry for which you want to define Navigation Collection options that override system-level defaults.

### Portal Registry Structures

**Node Name** You should not change the default value delivered with your system. This default value corresponds to the content provider node for your database. For example, the following databases are delivered with the following default node names:

PeopleSoft Financials/SCM: *ERP*

PeopleSoft CRM: *CRM*

PeopleSoft HCM: *HRMS*

PeopleSoft Enterprise Portal: *Local Node*

### Navigation Page

**Show Breadcrumbs** Select to indicate that you want breadcrumbs to display on Navigation Collection page publications and standard navigation pages in the selected portal registry.

These breadcrumbs reflect the path taken by the user as he navigates through left-navigation menu folders, Navigation Collection page folders, and standard navigation page folders. Displaying these breadcrumbs facilitates user navigation, as the left-navigation menu is not updated to reflect the user's navigation through Navigation Collection pages and standard navigation pages.

The first breadcrumb displayed is the Main Menu collection name link that points to the delivered EOPP_PORTAL_ROOT Navigation Collection, which is based on the PORTAL_ROOT_OBJECT folder. Alternatively, you can select a different Navigation Collection to be used as the first (main menu) breadcrumb using the Main Menu Collection Name field, or you can choose not to display a first breadcrumb.

The next set of breadcrumbs display dependent on the left-navigation menu folders the user opened to access the Navigation Collection page.

Subsequent breadcrumbs display dependent on the folders the user accesses on the Navigation Collection page.

**Show Main Menu Breadcrumb** If you have selected the Show Breadcrumbs option, select this option to include the Main Menu collection name as the first entry in the breadcrumb path displayed on a Navigation Collection page publication accessed using the selected portal registry. If you clear this option, the first entry in the breadcrumb path is dependent on the first left-navigation folder the user opened to access the Navigation Collection page. The Main Menu link does not display as the first link in the breadcrumbs.

**Main Menu Collection Name** This field displays when you have the Show Main Menu Breadcrumb option selected. The default value is set to *Main Menu*, which designates that the first breadcrumb displayed for the selected portal registry accesses the Main Menu

pagelet. You may choose to replace the first breadcrumb Main Menu link with a link that accesses an alternate Navigation Collection you have created.

The first breadcrumb Main Menu link to the Main Menu pagelet provides access to the entire navigation for your database, filtered by security. You may want to provide a first breadcrumb link that provides access to a Navigation Collection that you have configured to contain only selected folders and content references.

**Style Sheet Name** Select the CSS you want to use to generate Navigation Collection publications accessed using the selected portal registry. Cascading style sheets are defined in PeopleSoft Application Designer. This setting applies to both navigation pages and navigation pagelets.

See *PeopleTools 8.44 PeopleBook: PeopleSoft Application Designer,* "Creating Style Sheet Definitions"

**Maximum Columns** Enter the maximum number of columns you want to be displayed on a Navigation Collection publication accessed using the selected registry.

Navigation Collection pagelets display one or two columns depending on the user's settings on the Personalize Layout page. Narrow pagelets display contents in one column. Wide pagelets display contents in three columns.

This setting applies to only navigation pages.

See *Using PeopleSoft Applications PeopleBook,* "Setting User Preferences," Personalizing Your Homepage

**Maximum Child Links** Enter the maximum number of links you want to be displayed beneath each folder on a Navigation Collection publication accessed using the selected portal registry. If the number of links in the folder exceeds this number, an X More... link appears beneath the folder that accesses a page from which you can access all links stored in the Navigation Collection folder.

For example, if you set the Maximum Child Links field value to *4* and the Navigation Collection folder contains five links, three links display on the Navigation Collection page along with the X More... link.

When you select the X More... link, a page containing all five links in the Navigation Collection folder displays.

This setting applies to both navigation pages and navigation pagelets.

## Navigation Image Defaults

**Show Images** *Always Show Images.* Select if you always want images to display on Navigation Collection pages and pagelets accessed using the selected portal registry.

*Never Show Images.* Select if you never want images to display on Navigation Collection publications accessed using the selected portal registry.

*Only Show Defined Images.* Select if you only want specifically defined images to display on Navigation Collection publications accessed using the selected portal registry. If you select this option, only images defined

as a part of the Navigation Collection or defined on the source reference are displayed on the Navigation Collection's publications. System and registry level default images are not displayed.

**Large Folder Image** Select the image that you want to display for top-level folders on a Navigation Collection publication accessed using the selected portal registry.

**Large Content Image** Select the image that you want to display for top-level content references on a Navigation Collection publication accessed using the selected portal registry.

**Large Collection Image** Select the image that you want to display for top-level Navigation Collection references on a Navigation Collection publication accessed using the selected portal registry.

**Small Folder Image** Select the image that you want to display for a folder within a folder on a Navigation Collection publication accessed using the selected portal registry.

**Small Content Image** Select the image that you want to display for a content reference within a folder on a Navigation Collection publication accessed using the selected portal registry.

**Small Collection Image** Select the image that you want to display for a Navigation Collection within a folder on a Navigation Collection publication accessed using the selected portal registry.

**Delete Registry Override** Click to revert the portal registry's Navigation Collection options to the system-level defaults defined on the System Options page.

# CHAPTER 10

# Using Navigation Collections

This chapter provides an overview of Navigation Collections and discusses how to:

- Create and maintain Navigation Collections.
- Publish Navigation Collections.

## Common Elements Used in This Chapter

Click the spellcheck button to check the spelling of text in the associated field.

See *Using PeopleSoft Applications PeopleBook*, "Working with Pages," Using Spell Check

## Describing Navigation Collections

Navigation Collections provide you with a flexible tool for building alternative taxonomies of the contents stored in your portal registry. These alternative taxonomies, or groupings of links to portal content, can then be deployed to different users or groups of users, creating navigation that specifically addresses your users' business needs.

To help understand Navigation Collections, consider the following analogy. An operating system contains a repository called a file system where all information about that operating system and its user files are stored. This file system is a rigid structure that contains information in different segments of the structure. Each piece of information is uniquely tagged and identified. The PeopleTools portal registry parallels this operating system's file system.

In a Microsoft Windows operating system, there are a number of ways to access a piece of information in the file system. For example, when you use Windows, you can use the File Explorer to navigate through a visual representation of the rigid file system. File Explorer enables you to perform different tasks with a file such as open the file, delete the file, view the properties of the file, and so forth.

You can also access this same file using a shortcut, which is a pointer to the actual file in the file system. This shortcut is stored on the file system and provides a method of file access that is an alternative to the rigid file system. As with the file itself, you can perform a variety of tasks related to the shortcut, such as create a shortcut, delete a shortcut, open a shortcut, and so forth.

The links in Navigation Collections provide an alternate method of accessing content in the portal registry. A Navigation Collection is a published compilation of these links. A link in a Navigation Collection is a pointer to a content reference or folder reference) in the portal registry; however, it does not store the contents of the content reference or folder reference. Navigation Collection links store only the reference attributes needed to construct a URL at runtime to redirect output to the original content reference or folder reference. These attributes include the target portal, target folder reference, and target content reference.

Once created, a Navigation Collection can be published as a navigation page or navigation pagelet for use on a user home page.

---

**Note.** The option to publish or access a Navigation Collection as a navigation pagelet is available if you have purchased PeopleSoft Enterprise Portal or a Portal Pack.

---

When you define and save a Navigation Collection, the system automatically stores it in a hidden folder under Portal Objects in the portal registry.

You can choose to publish Navigation Collections as public content. If you do not choose to provide public access to a Navigation Collection, you can run the Sync Collection Security (synchronize collection security) process to apply content reference and folder security defined in the portal registry to the published navigation page or pagelet content reference. You can also allow manual maintenance the security access to published pages and pagelets by selecting the *Do Not Allow Security Sync* option for the publication on the Publish Collection page.

See Chapter 10, “Using Navigation Collections,” Publishing Navigation Collections, page 144.

Navigation Collections allow you to distribute the creation and maintenance of collections to subject matter experts in your organization, such as functional administrators and business analysts. The subject matter experts creating Navigation Collections should understand the content their user communities need to access on a frequent basis. They should also know where to find links to this content, whether they are accessed by way of portal navigation or external URLs.

Navigation Collections can contain the following elements:

- Links to existing content references.
- Links to existing non-hidden folders in the menu.

  The Navigation Collection will contain all contents of the folder and its subfolders.
- Custom folders that can be used to group combinations of the elements listed above.

### See Also

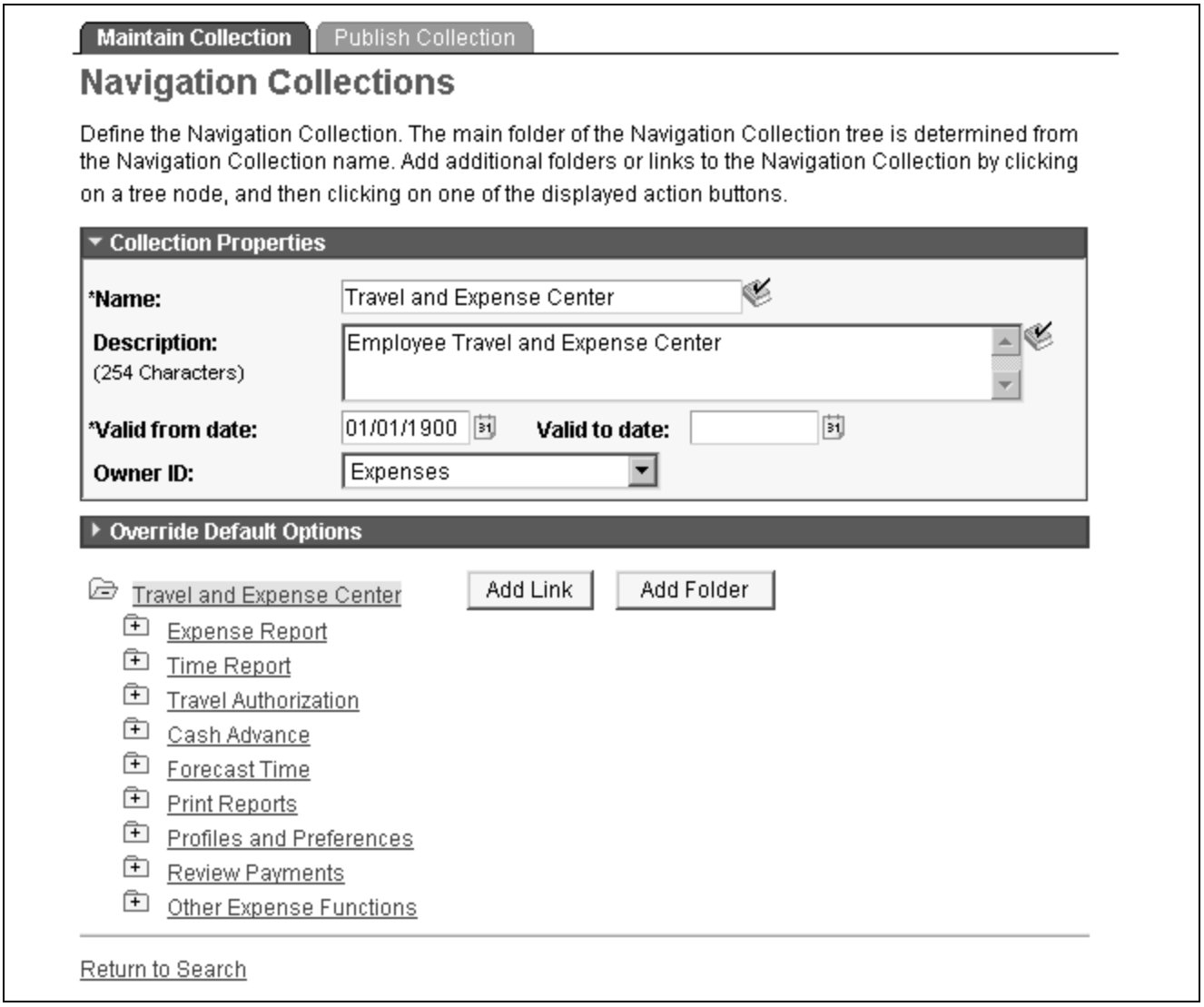
Chapter 11, “Running Portal Utility Processes,” Synchronizing Collection Security, page 149

# Describing Navigation Collection Publications

In this section, we provide overviews of the Navigation Collections published as navigation pages and navigation pagelets.

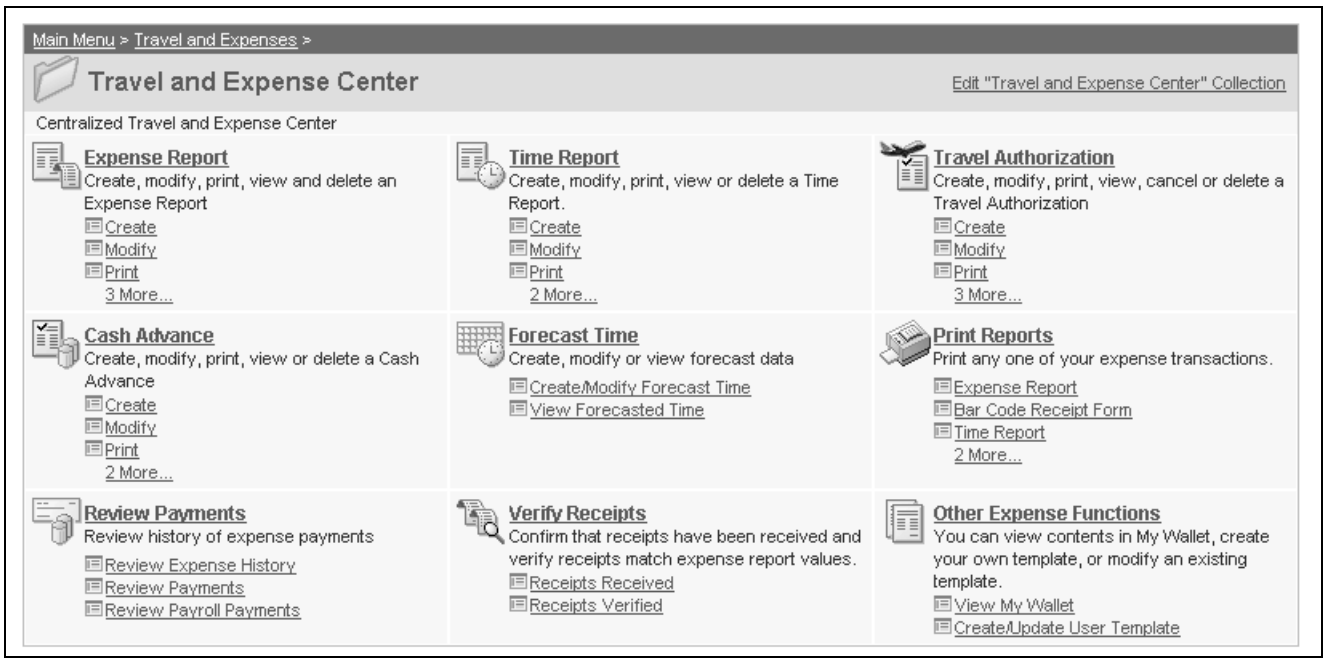
**Note.** The option to publish or access a Navigation Collection as a navigation pagelet is available if you have purchased PeopleSoft Enterprise Portal or a Portal Pack.

The sample Navigation Collection publications provided in this section were created using the Navigation Collection definition shown in the following example.



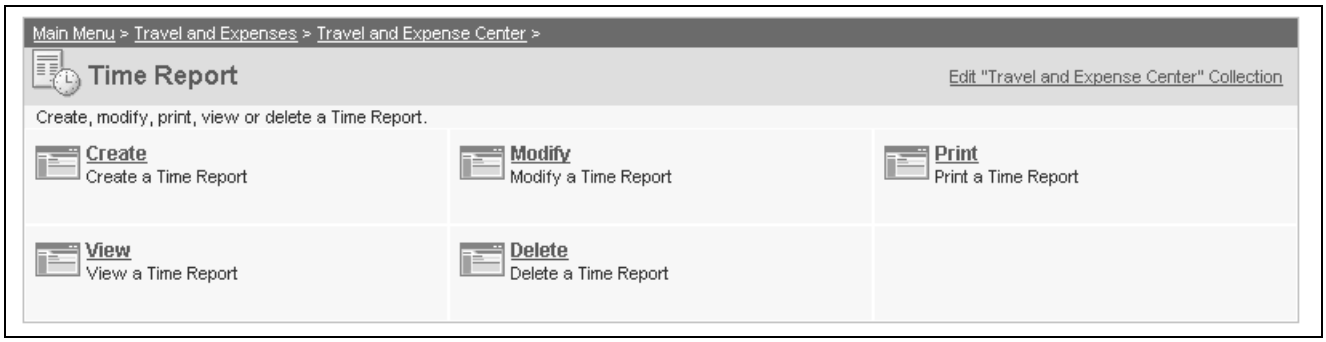

Maintain Collection page

The following example illustrates the appearance of a Navigation Collection published as a navigation page.




Travel and Expense Center navigation page

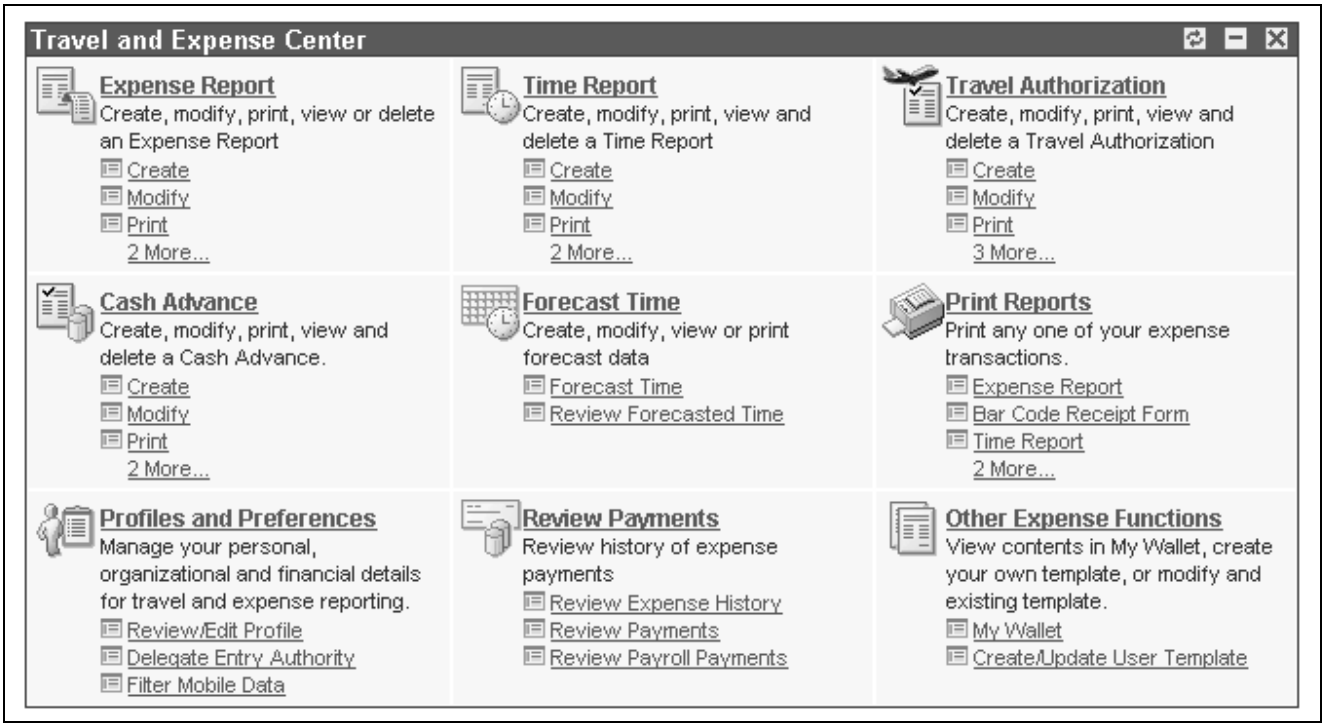
Top-level folders in the Navigation Collection may be transformed into bold and linked section headings that are used to organize the folders and content references they contain, as shown in these examples. However, the appearance of your published Navigation Collections is dependent on its assigned style sheet. The assigned style sheet must use style classes contained in the EOPP_SCSTYLEDEF style class definition. Selecting the top-level folder link accesses a page containing any folders and content references nested in the parent folder, as shown in the following example.




Time Report navigation page

Select the Edit <Navigation Collection name> Collection link on the navigation page to access the Navigation Collection definition on the Maintain Collection page. This link displays only for users associated with the EOPP2050 permission list.

The following example illustrates the appearance of a Navigation Collection published as a navigation pagelet.




Travel and Expense Center navigation pagelet

**Note.** The option to publish or access a Navigation Collection as a navigation pagelet is available if you have purchased PeopleSoft Enterprise Portal or a Portal Pack.

Folders and content references work in much the same way as they do for the navigation page publication.

### See Also

Chapter 10, "Using Navigation Collections," Publishing Navigation Collections, page 144

Chapter 10, "Using Navigation Collections," Creating and Maintaining Navigation Collections, page 125

# Describing Folder and Link Sequence on Navigation Pages

This section discusses the factors that determine the sequence in which folders and links display on the following types of navigation pages:

- Standard navigation pages
- Custom navigation pages

### Standard Navigation Pages

These pages are accessed by selecting folders in the menu navigation and are automatically generated according to the contents of the selected menu navigation folder.

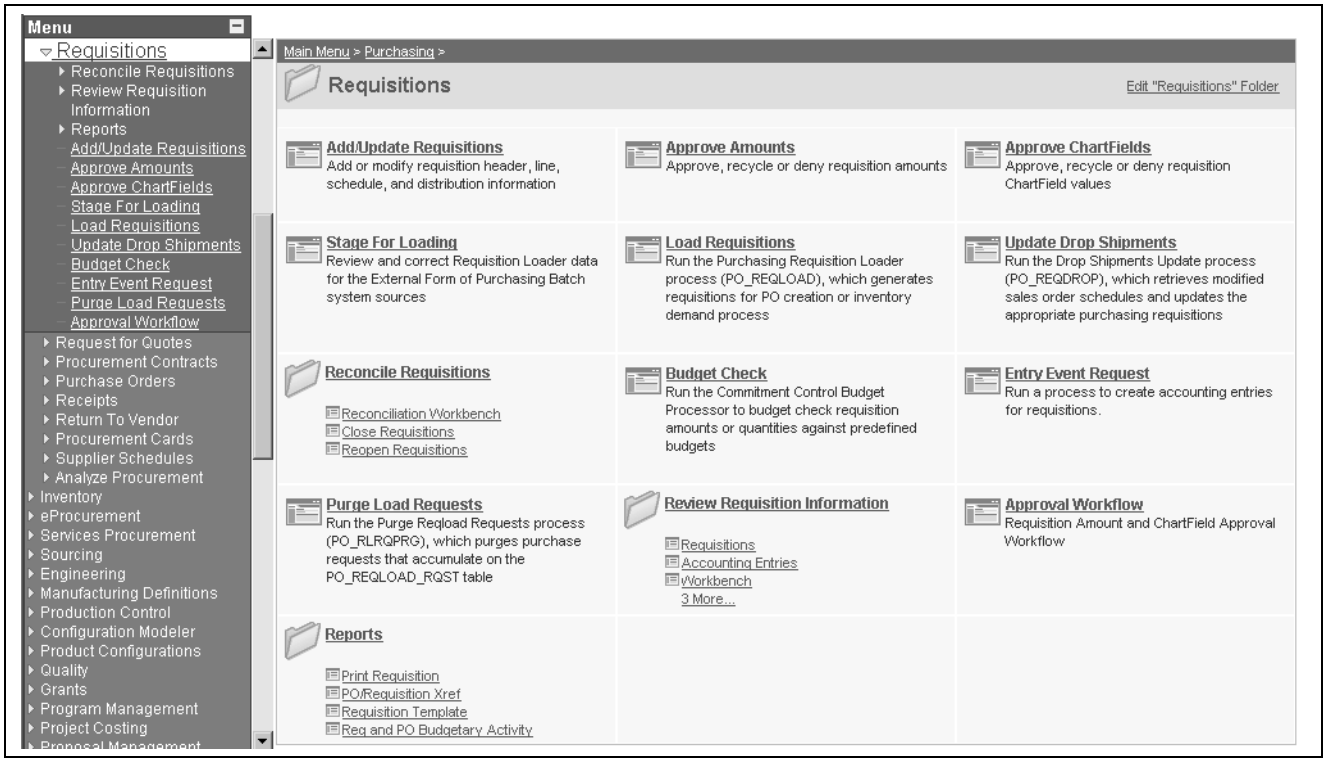
See Chapter 8, "Working with Navigation Pages," Describing Standard Navigation Pages, page 100.

The sequence in which folders and links appear on standard navigation pages is defined in the Structure and Content component.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Administering Folders

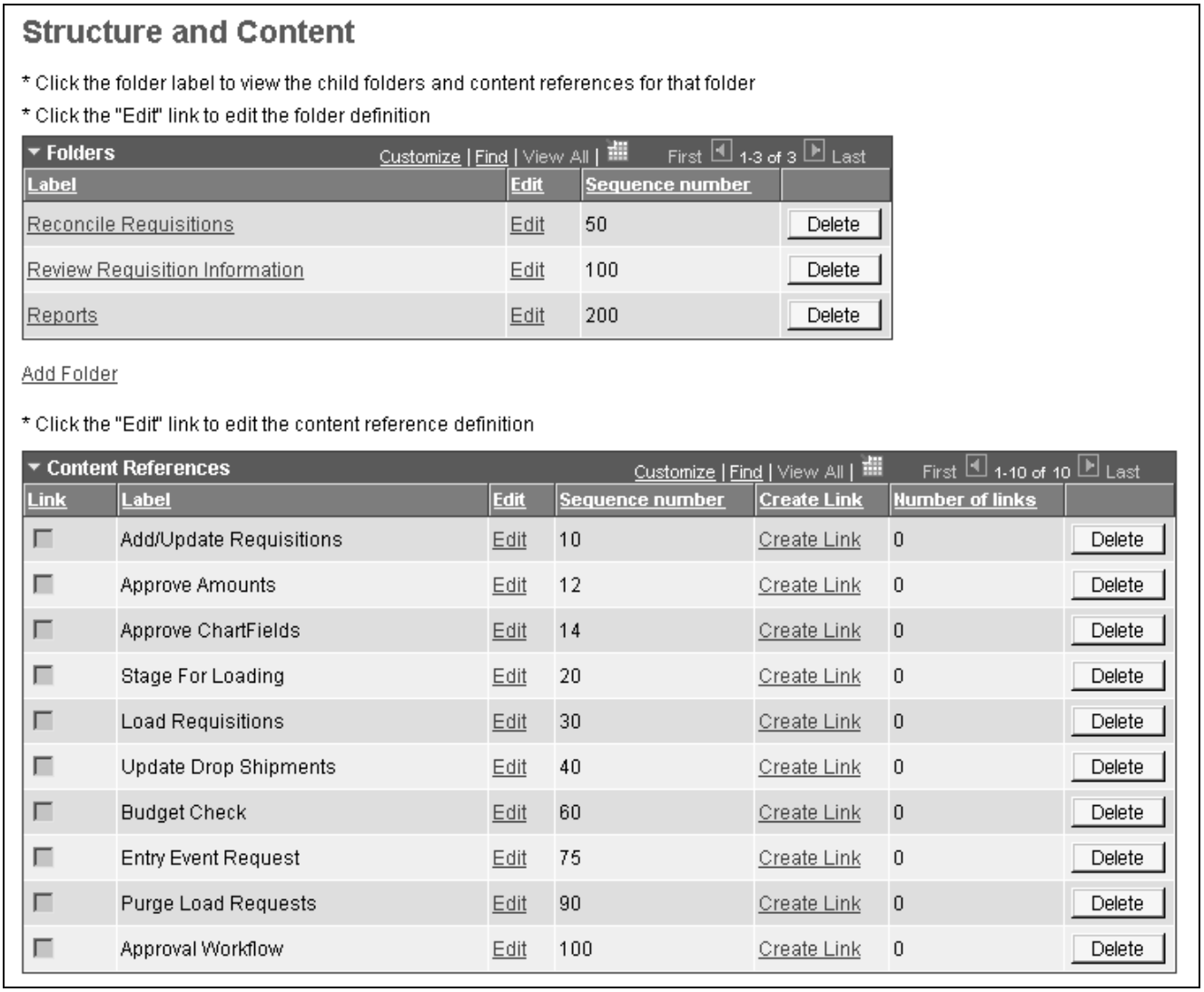
See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Administering Content References

Although the contents of a standard navigation page correspond to that of the menu navigation folder selected to display it, the sequence of elements on the page may not be the same as the sequence in which they are displayed in the menu navigation. For example, notice the differences in the sequence of elements in the Requisitions menu navigation folder and the sequence of elements on the Requisitions standard navigation page:




Comparing the Requisitions menu navigation folder to the Requisitions standard navigation page

As dictated by menu navigation functionality, folder elements are displayed before content reference links. The order of elements within these groupings is determined by the sequence numbers listed in the Structure and Content component. The following example provides the sequence numbers defined for the folders and content references in the Requisitions menu folder. You will notice that the sequence numbers correspond to the order of folders, and the order of content reference links in the Requisitions menu navigation folder.




Structure and Content component

Unlike the grouped elements in the menu navigation folder, you will notice that on the Requisitions standard navigation page, folder elements and content reference links intermingle. However, their overall sequence is also determined by the sequence numbers listed in the Structure and Content component.

For example, the Reconcile Requisitions folder (sequence 50) appears first in the Requisitions menu navigation folder because, as discussed earlier, folders are displayed first in menu navigation, and it has the lowest sequence number of the folders in its group.

However, on the Requisitions standard navigation page, the Reconcile Requisitions folder (sequence 50) does not appear until after the Add/Update Requisitions content reference (sequence 10), Approval Amounts content reference (sequence 12)...and Update Drop Shipments content reference (sequence 40). This is because folder and link sequence on standard navigation pages is determined strictly by sequence numbers in the Structure and Content component, and not on the type of element being displayed.

If a folder element and content reference link have the same sequence number, the folder is displayed before the content reference on the standard navigation page. For example, see the Review Requisition Information folder (sequence 100) and Approval Workflow content reference (sequence 100).

### Custom Navigation Pages

These pages are accessed by clicking content reference links in the menu navigation and are built using the Navigation Collection component.

See Chapter 8, "Working with Navigation Pages," Describing Custom Navigation Pages, page 101.

The sequence in which folders and links appear on custom navigation pages is defined by the Placement Properties set in the Navigation Collection definition for the custom navigation page.

See Chapter 10, "Using Navigation Collections," Editing a Folder, page 138.

See Chapter 10, "Using Navigation Collections," Editing a Link, page 142.

Once the folder and link sequences for a custom navigation page are registered, their sequence numbers are propagated to the registry and are available in the Structure and Content component. However, you should not modify custom navigation pages using the Structure and Content component. You should only modify sequences using the Placement Properties in the Navigation Collection definition for the customer navigation page.

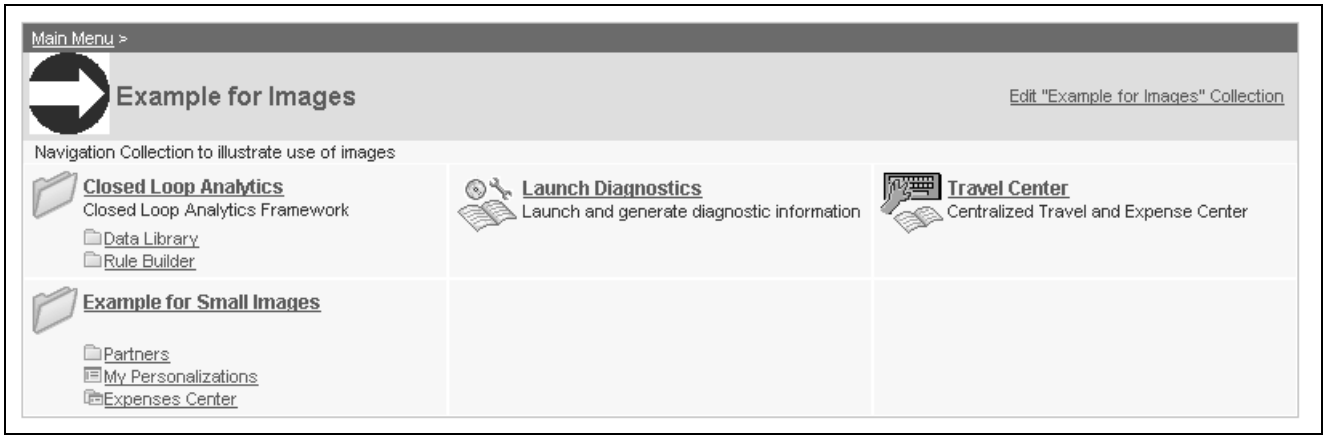
When you save the Navigation Collection, these modifications to the Placement Properties sequences will be updated to the registry.

## Describing Navigation Collection Images

In this section, we discuss how the images you define for your Navigation Collections are published and displayed on Navigation Collection publications. You can associate images with Navigation Collections themselves, as well as with Navigation Collection folders and content references. You can define these images at the system, registry, source, and individual Navigation Collection levels.

See Chapter 9, "Setting Up Navigation Collection Options," Describing the Navigation Collection Option Default Hierarchy, page 105.

The following example Navigation Collection page publication displays the different ways in which images can display.




Example for Images navigation page

For our example Navigation Collection above, let's say that the following images have been defined for the different image types.

This image was specifically defined as the collection image for this specific Navigation Collection.

This image was been defined as the default large folder image on the System Options page, and was not overridden at the source reference, registry, or Navigation Collection level. Large folder images display for folders at the top level of a Navigation Collection publication.

This image was specifically defined as the content image in the Navigation Collection definition. Large content images display for content references at the top level of a Navigation Collection publication.

This image represents a Navigation Collection that was added to this Navigation Collection. This image was selected as the collection image in the referenced Navigation Collection definition, and was not overridden in this Navigation Collection definition. Large collection images display for Navigation Collections at the top level of a Navigation Collection publication.

This image has been defined as the small folder image. Small folder images display for folders nested within top-level folders on a Navigation Collection publication.

This image has been defined as the small content image. Small content images display for content references nested within top-level folders of a Navigation Collection publication.

This image has been defined as the small collection image. Small collection images display for Navigation Collections nested within top-level folders of a Navigation Collection publication.

While large (parent) images can be set at the system or registry default levels, and overridden with unique images defined at the source reference or Navigation Collection levels, small (child) images can be defined only at the system and registry default levels. This means that the small images displayed on Navigation Collection publications will always be the default small image defined at the system or registry level.

If the Navigation Collection content, folder, or Navigation Collection reference for which a small image is displayed has a unique image defined at the source reference or Navigation Collection level, this unique large image does not display until the element is accessed as a top- or parent-level element on a Navigation Collection page.

For example, while the Expenses Center is displayed with its default small image, when you access the Example for Small Images navigation page, the Expenses Center displays with its unique large image.

# Creating and Maintaining Navigation Collections

In this section, we discuss how to:

- Search for a Navigation Collection.
- Create or edit a Navigation Collection.
- Add a Navigation Collection folder.
- Select a source folder.
- Select Navigation Collection images.
- Edit a Navigation Collection folder.
- Add a Navigation Collection link.
- Select a source link.
- Edit a Navigation Collection link.

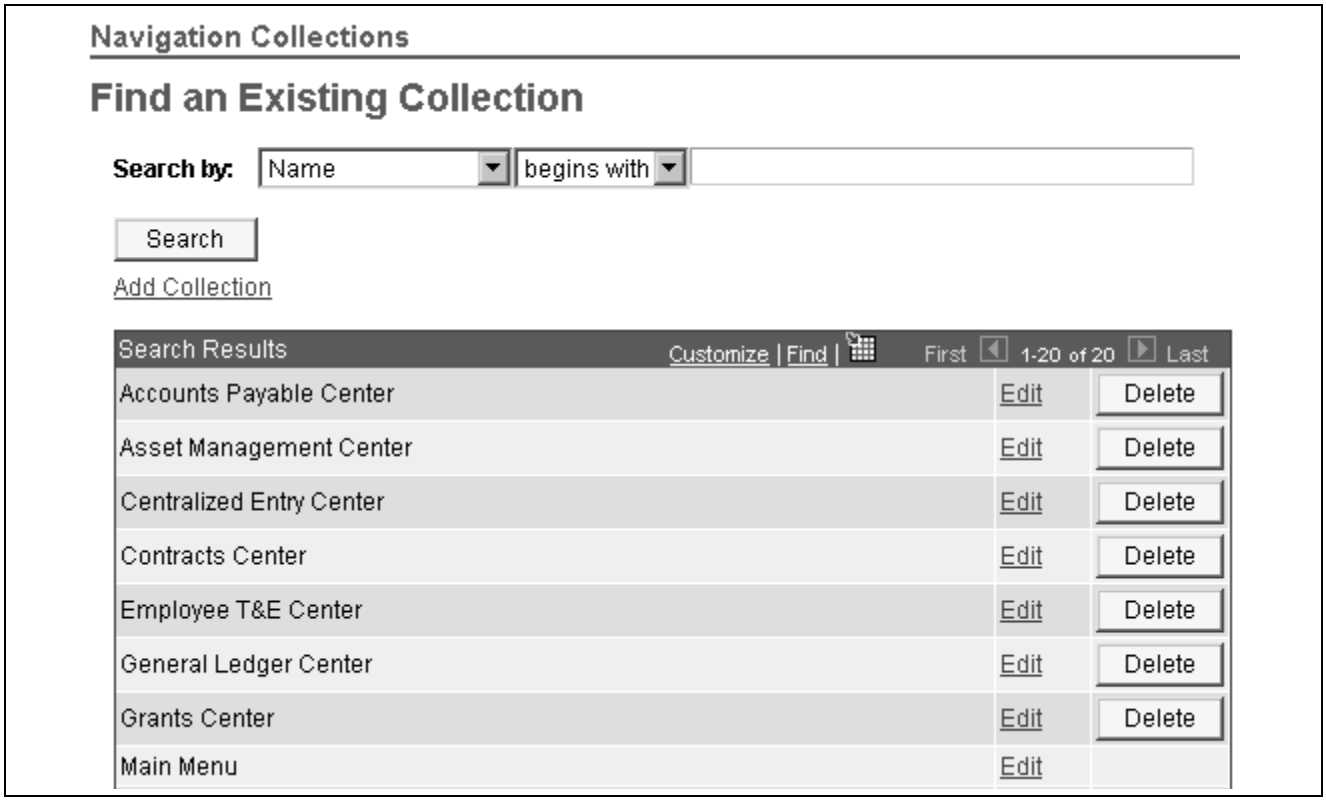
## Pages Used to Create and Maintain Navigation Collections

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Find an Existing Collection | EOPP_SCSRCH | • Enterprise Components, Portal Utilities, Navigation Collections<br>• Select the Return to Search link on the Maintain Collections page. | Search for an existing Navigation Collection that you want to edit or delete. Access pages you can use to create a new Navigation Collection. |
| Maintain Collection | EOPP_SCMAINTCOLL | • Select the Edit link for an existing Navigation Collection on the Find an Existing Collection page.<br>• Select the Add Collection link on the Find an Existing Collection page.<br>• Select the Edit <Navigation Collection name> Collection link on a custom navigation page. | Create and maintain Navigation Collections. |
| Add Folder | EOPP_SCAE_FOLDER | Click the Add Folder button on the Maintain Collections page. | Add a Navigation Collection folder reference. |
| Select Source Folder | EOPP_SCBROWSEFLDR | • Click the Browse Portal Registry button on the Add Folder page.<br>• Click the Browse Portal Registry button on the Edit Folder page. | View a graphical display of a selected portal registry and select the content reference you want to add to your Navigation Collection. |

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Browse Image Catalog | EOPP_SCIMGSRCH | • Click the Browse Image Catalog button on the Add Folder page.<br>• Click the Browse Image Catalog button on the Edit Folder page.<br>• Click the Browse Image Catalog button on the Add Link page.<br>• Click the Browse Image Catalog button on the Edit Link page. | Search for and select an image you want to associate with the selected folder or content reference. This image displays on the published navigation page or pagelet. |
| Edit Folder | EOPP_SCAE_FOLDER | Click the Edit Folder button on the Maintain Collections page. | Edit an existing Navigation Collection folder reference. |
| Add Link | EOPP_SCAE_SHORTCUT | Click the Add Link button on the Maintain Collections page. | Add a content reference link. |
| Select Source Link | EOPP_SCBROWSEREG | • Click the Browse Portal Registry button on the Add Link page.<br>• Click the Browse Portal Registry button on the Edit Link page.<br>• Click the Move button on the Publish Collections page. | When accessed from the Add Link page and Edit Link page, view a graphical display of a selected portal registry and select the folder you want to reference in your Navigation Collection.<br>When accessed from the Publish Collections page, view a graphical display of the portal registry defined for the navigation page publication and select the folder in which you want the navigation page to reside. |
| Edit Link | EOPP_SCAE_SHORTCUT | Click the Edit Link button on the Maintain Collections page. | Edit an existing content reference link. |

## Searching for a Navigation Collection

Access the Find an Existing Collection page.




Find an Existing Collection page

**Search by**

*Description.* Select to conduct your search based on Navigation Collection description text.

*Name.* Select to conduct your search based on Navigation Collection name text.

Use the *begins with* and *contains* options to limit your search results.

**Add Collection**

Select to access the Maintain Collection page, where you can define a new Navigation Collection.

## Search Results

**Edit**

Select to access the Maintain Collection page, where you can edit an existing Navigation Collection.

**Delete**

Click to delete the existing Navigation Collection. You are prompted with the Navigation Collections - Delete Confirmation page, where you can confirm or cancel the deletion of the Navigation Collection.

---

**Note.** The Delete button does not display for the Main Menu Navigation Collection. While you can edit this Navigation Collection, you should not delete it as the Main Menu Navigation Collection is used by the system as a default page to display when it encounters certain errors. Displaying the Main Menu rather than an error page enables you to continue to navigate through your system.

---

# Creating or Editing a Navigation Collection

Access the Maintain Collection page.


Maintain Collection page

---

**Note.** If you modify a Navigation Collection that has already been published, you do not need to republish the collection using the Publish Collection page. Saving your edits on the Maintain Collection page is sufficient because Navigation Collection publications are dynamically generated for display based on their saved Navigation Collection definitions.

---

## Collection Properties

**Name** Enter a name for your Navigation Collection. This text displays as the label on the published navigation page or pagelet.

**Description** Enter a description of your Navigation Collection. This text can display in several locations.

If the Navigation Collection is published as a navigation page, the description text displays as hover text for the menu item link used to access the page. The description text also displays on the navigation page.

If the Navigation Collection is published as a navigation pagelet, the description text displays on the navigation pagelet.

**Valid from date/Valid to date** Enter a range of dates during which you want this Navigation Collection to be available. This information is copied to the portal registry when you register the Navigation Collection using the options on the Publish Collection page.

We recommend that you leave the Valid to date field clear, if you anticipate that the Navigation Collection should remain active indefinitely.

**Owner ID** Use this value to easily identify Navigation Collections created by your organization when searching for Navigation Collections through PeopleSoft Application Designer or SQL queries on the portal registry table.

Do not select a delivered PeopleSoft owner ID. You can define unique owner IDs for your organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer .

If a default value has been set at the system or registry level, it displays, but can be overridden.

See Chapter 9, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 105.

## Override Default Options

The Override Default Options group box enables you to enter values that override any Navigation Collection defaults defined at the system, registry, and source reference levels and which apply to only the Navigation Collection you are creating. Leave fields clear if you don’t want to override any default values.

See Chapter 9, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 105.

**Style Sheet Name** Select the cascading style sheet (CSS) you want to use to generate publications for this Navigation Collection. Cascading style sheets are defined in PeopleSoft Application Designer.

See *PeopleTools 8.44 PeopleBook: PeopleSoft Application Designer,* "Creating Style Sheet Definitions"

**Maximum Child Links** Enter the maximum number of child links you want to display on publications for this Navigation Collection.

**Show Images** *Always Show Images.* Select if you always want images to display on publications for this Navigation Collection.

*Never Show Images.* Select if you never want images to display on publications for this Navigation Collection.

*Only Show Defined Images.* Select if you only want specifically defined images to display on publications for this Navigation Collection. If you select this option, only images defined as a part of this Navigation Collection or the source reference are displayed on the publications for this Navigation Collection. System- and registry-level default images are not displayed.

**Note.** If you select the Only Show Defined Images option, small images do not display on your Navigation Collection publications as small images can only be defined as system- and registry-level defaults.

**Collection Image** Select the large image you want to display on publications for this Navigation Collection.

The Navigation Collection you define is displayed as a tree hierarchy of nodes at the bottom of the page.

The Name value you entered for the Navigation Collection displays as a root node link at the top of the hierarchy. Each node below the Navigation Collection name is either a folder or content reference link you defined using the following options and their associated pages.

A Navigation Collection may be composed of a series of individual content reference links, a series of folders containing content reference links, or a combination of individual content reference links and folders.

**Add Link** Click to access the Add Link page. This option displays when you select the Navigation Collection root node folder or user-defined folders in the Navigation Collection hierarchy.

If you use this option at the root node level of the hierarchy, the content reference link you create appears directly off the root node, at the same level in the Navigation Collection structure as a folder.

**Edit Link** Click to access the Edit Link page. This option displays when you select a content reference link in the Navigation Collection hierarchy.

**Delete Link** Click to access the Navigation Collections - Delete Confirmation page, where you are prompted to confirm or cancel the deletion of the selected content reference link. This option displays when you select a content reference link in the Navigation Collection hierarchy.

**Add Folder** Click to access the Add Folder page. This option displays when you select the Navigation Collection root node folder or user-defined folders in the Navigation Collection hierarchy.

If you use this option at the root node level of the hierarchy, the folder you create appears directly off the root node. Using this option while in any folder in the hierarchy creates a folder within the selected folder.

**Edit Folder** Click to access the Edit Folder page. This option displays when you select any folder other than the root node folder.

**Delete Folder** Click to access the Navigation Collections - Delete Confirmation page, where you are prompted to confirm or cancel the deletion of the selected folder. This option displays when you select any folder other than

the root note folder in the Navigation Collection hierarchy. This only deletes the folder from the Navigation Collection.

Designates a folder whose content is user-defined . These folders are added on the Add Folder page using the *User Defined Folder* folder type.

Click to expand the node and access options that enable you to edit or delete the folder.

Alternatively, you may choose to select only the associated folder name link, which displays options that enable you to edit or delete the folder, but does not expand folder contents.

Designates a folder whose content is menu-based. These folders are added on the Add Folder page using the *Menu Folder* folder type.

You cannot view folder contents because a menu-based folder automatically inherits the content and folder references registered in the selected menu folder. This content cannot be edited from within Navigation Collections.

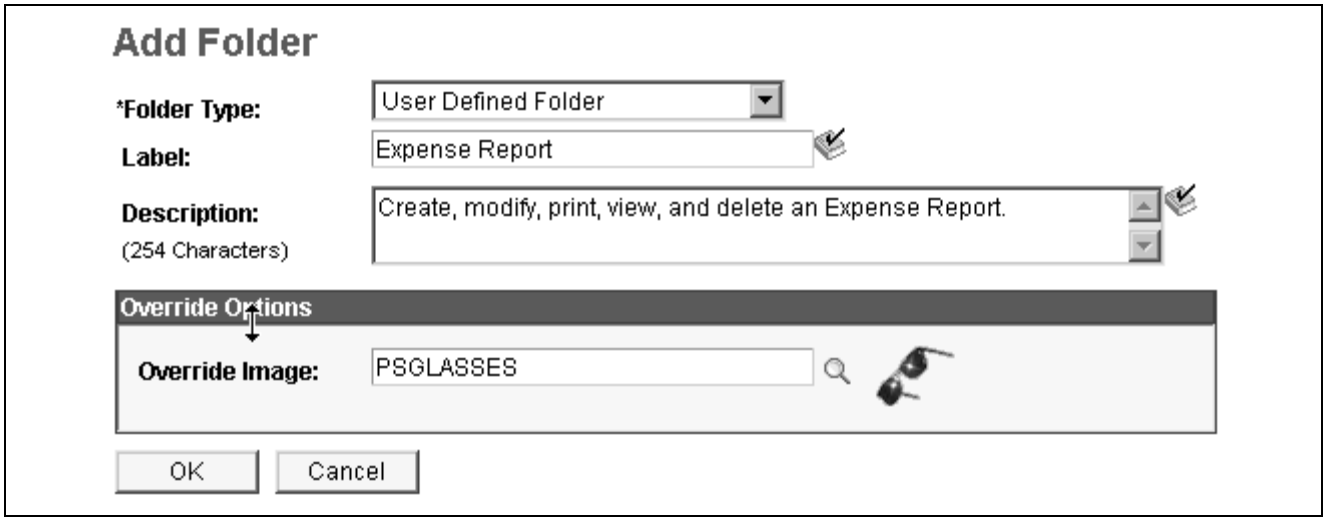
Select the associated folder name link to access options that enable you to edit specific aspects of the folder or delete the folder from the Navigation Collection.

Designates a content reference link in the hierarchy.

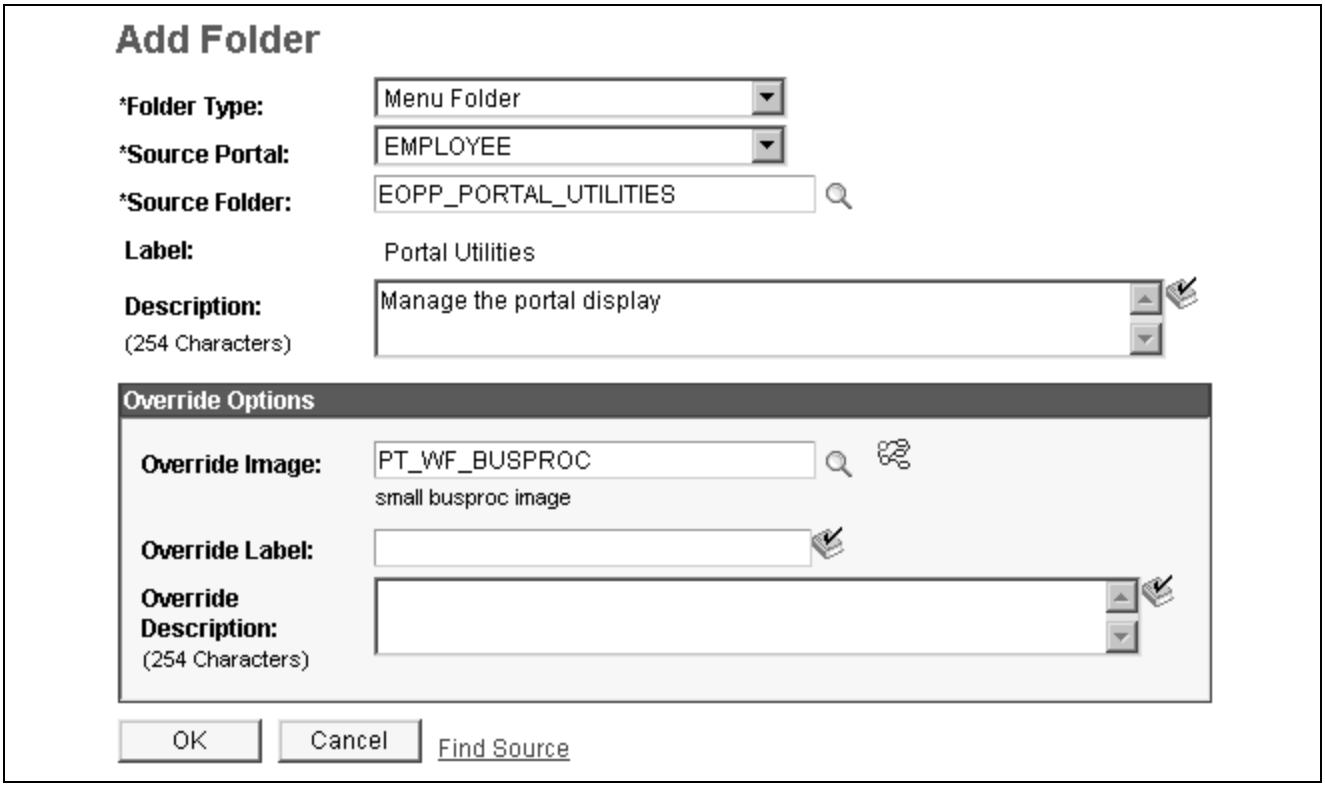
Click to access options that enable you to edit or delete the content reference from the Navigation Collection.

## Adding a Folder

Access the Add Folder page.




Add Folder page (user-defined folder)




Add Folder page (menu folder)

**Folder Type**

*Menu Folder.* Creates a reference to an existing folder in the portal registry. The reference automatically inherits all of the content and folder references registered in the selected menu folder.

*User Defined Folder.* Creates a user-defined folder, enabling you to specify the folder label, description, and valid from and to dates for the folder.

**Source Portal**

This field displays when the Folder Type value is set to *Menu Folder.*

Select the portal registry from which you want to select a folder to reference. The current portal is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases when a Navigation Collection will be moved between portals, and the folder reference you are defining should automatically refer to the current portal.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals"

**Source Folder**

This field displays when the Folder Type value is set to *Menu Folder.*

Select a folder to reference. Click the Browse Portal Registry button to access the Select Source Folder page, where you can view a graphical display of the selected portal registry and select the folder you want to reference.

**Label**

If you are adding a menu-based folder, the label on the source populates this field, but can be overridden in the Override Label field.

This text appears as link text for the folder on the Maintain Collection page, as well as on the Navigation Collection publication.

**Description** If you are adding a menu-based folder, the description on the source populates this field, but can be overridden in the Override Description field.

This text displays as hover text for a folder link, as well as descriptive text beneath the folder link on the Navigation Collection publication.

## Override Options

**Override Image** Select the image you want to display as the icon for the folder reference. If a default folder image has been defined at the system or registry level, this image will override the default. This image displays as the large image for the folder on the Navigation Collection publication.

When defined for folders nested within folders, the image displays on the Navigation Collection page that is accessed to display the contents of the parent folder.

If you override the image, changes to the image on the source reference are not reflected on the Navigation Collection publication. If you want source reference changes to be reflected on the Navigation Collection publication, do not override this image.

See Chapter 9, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 105.

See Chapter 10, “Using Navigation Collections,” Describing Navigation Collection Images, page 124.

**Override Label** If you are adding a menu-folder, you can enter a label to override label text in the Label field.

If you override label text, changes to label text on the source reference are not reflected on the Navigation Collection publication. If you want source reference changes to be reflected on the Navigation Collection publication, do not enter override text in this field.

**Override Description** If you are adding a menu-based folder, you can enter a description to override description text in the Description field.

If you override description text, changes to description text on the source reference are not reflected on the Navigation Collection publication. If you want source reference changes to be reflected on the Navigation Collection publication, do not enter override text in this field.

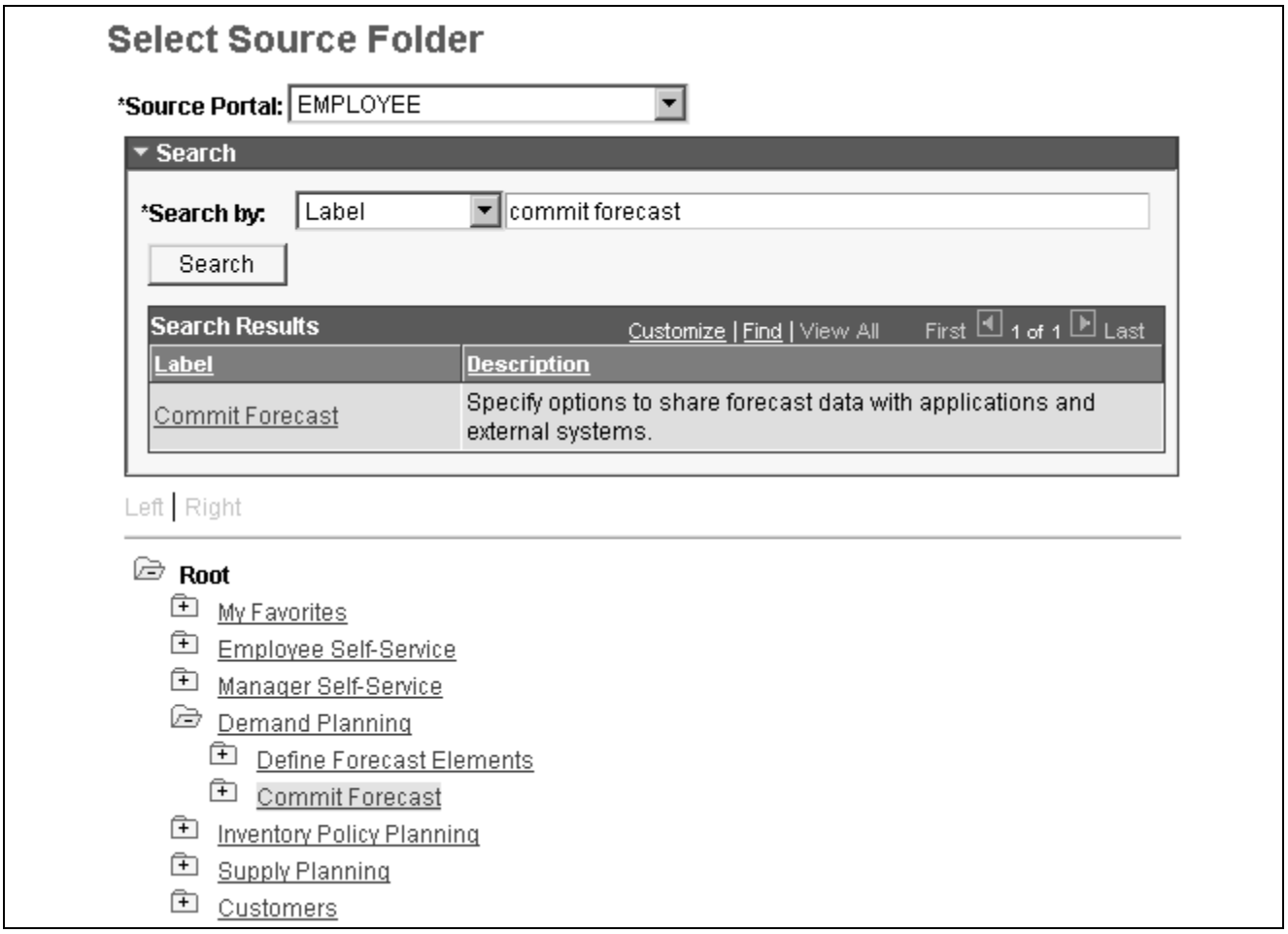
**Find Source** This link displays for users associated with the PTPT1300 permission list or Portal Administrator role when the Folder Type value is set to *Menu Folder.*

Select to access the Folder Administration page, where you can view details about the selected menu folder. You may want to access this page to add a source reference image, change a source reference label, or check on source reference security.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Administering Folders

## Selecting a Source Folder

Access the Select Source Folder page.




Select Source Folder page

**Source Portal** Select the portal registry from which you want to select a folder to reference. The portal designated on your originating page is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases when a Navigation Collection will be moved between portals, and the folder reference you are defining should automatically refer to the current portal.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals"

### Search

**Search by** *Description.* Select to conduct your search based on folder description text. This search is not case sensitive. You can perform partial match

searches. For example, doing a label search on "source" may not return any "source" hits, but may return hits on "resource."

*Label.* Select to conduct your search based on folder label text.

Alternatively, you may perform a search by drilling down into the tree structure displayed at the bottom of the page.

### Search Results

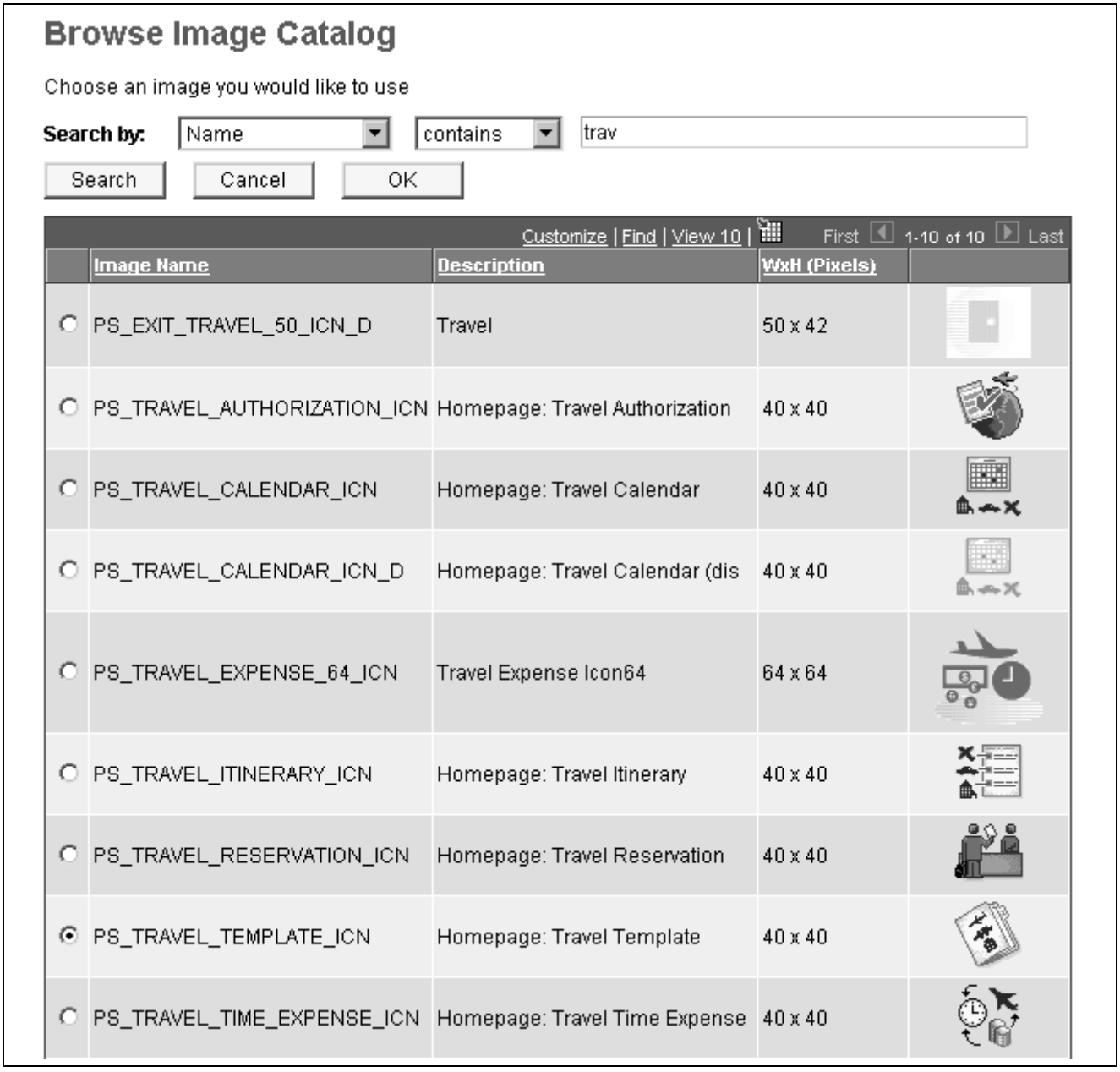
**Label** Select the linked label text to locate and select the referenced folder in the portal registry hierarchy displayed at the bottom of the page.

**Description** Displays the text description of the folder.

Once you have located your desired folder, click the folder link in the hierarchy to select the folder and return to your originating page.

## Selecting Navigation Collection Images

Access the Browse Image Catalog page.




Browse Image Catalog page

**Search by**

*Name.* Select to conduct your search based on the image file name.

*Description.* Select to conduct your search based on image description text.

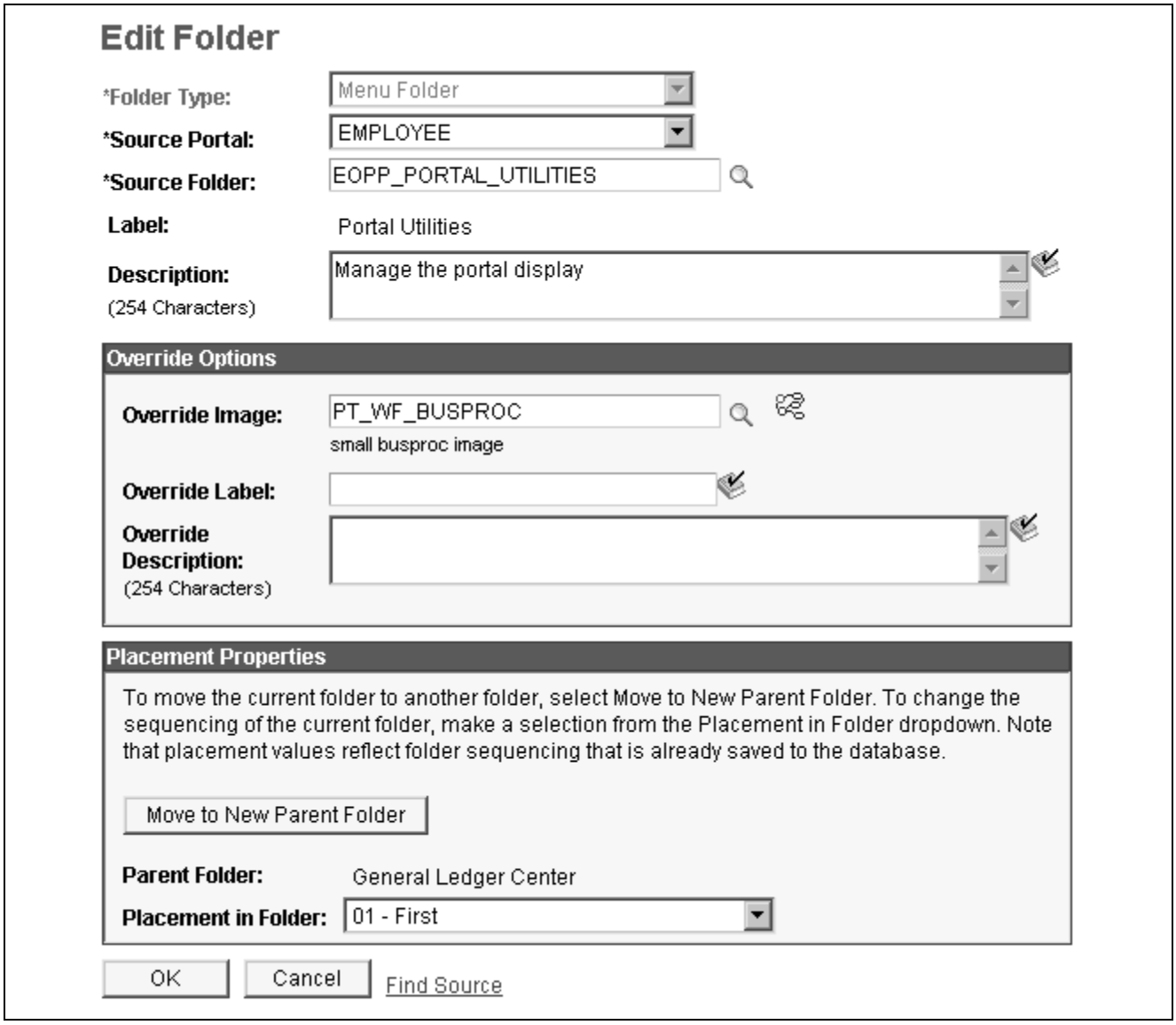
You can further refine your search by selecting the adjacent *begins with* or *contains* option.

This search is not case sensitive. You can perform partial match searches. For example, doing a search on "source" using the *Description* and *contains* options may not return any "source" hits, but may return hits on "resource."

Once you have located the desired image, select the radio button for the image and click OK.

# Editing a Folder

Access the Edit Folder page.




Edit Folder page

With the exception of the Placement Properties group box, which is discussed below, all other options on this page are available on and documented as a part of the Add Folder page.

See Chapter 10, "Using Navigation Collections," Adding a Folder, page 132.

## Placement Properties

**Move to New Parent Folder** Click to move the folder you are editing to a folder other than the folder designated in the Parent Folder field.

You can only change the placement of a folder reference after you have initially saved the Navigation Collection.

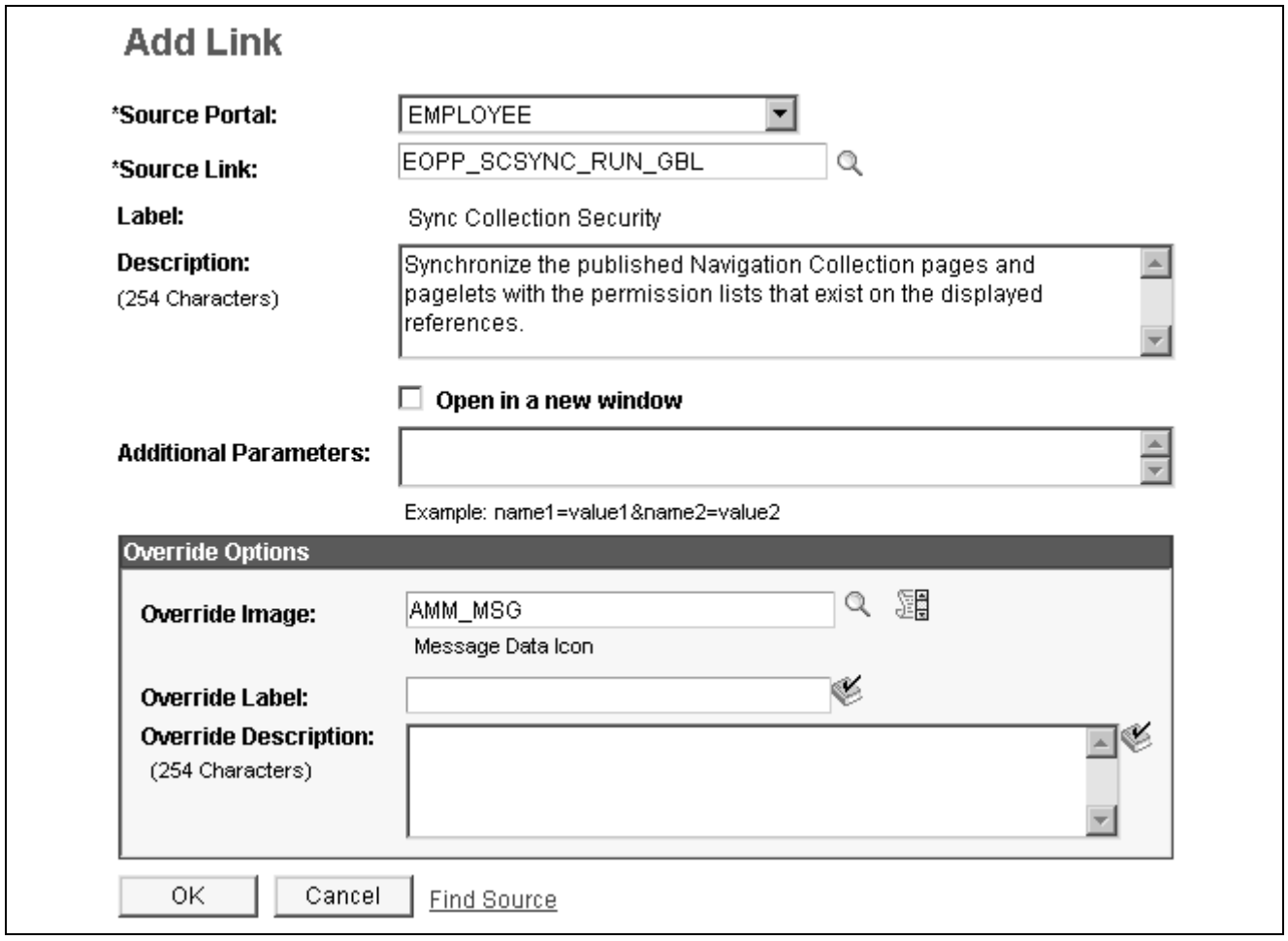
**Parent Folder** Displays the parent folder of the folder you are editing.

**Placement in Folder** Select a value in the drop-down list box to change the placement of the folder within the parent folder.

Available placement values reflect saved folder sequencing.

## Adding a Link

Access the Add Link page.




Add Link page

**Source Portal** Select the portal registry from which you want to select a content reference. The current portal is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases when a Navigation Collection will be moved between portals, and the content reference you are defining should automatically refer to the current portal.

See *PeopleTools 8.44 PeopleBook: Internet Technology*, "Administering Portals"

**Source Link** Select a content reference to add to the Navigation Collection. Click the Browse Portal Registry button to access the Select Source Link page, where you can view a graphical display of the selected portal registry and select the content reference you want to add.

You can use this option to add a published navigation page to the Navigation Collection you are creating.

**Label** The existing menu content reference label text populates this field, but can be overridden in the Override Label field. This text appears as link text for the content reference on the Maintain Collection page, as well as on the Navigation Collection publication.

**Description** The existing description text populates this field, but can be overridden in the Override Description field. This description text displays as hover text for the content reference link.

**Open in a new window** Select if you want the page accessed by the content reference to open in a new browser window.

**Additional Parameters** Define additional query parameters that are appended to the uniform resource locator (URL) used to access the content reference. Defining these query parameters here make it unnecessary to register the same content reference multiple times with different parameters.

For example, you can use these additional parameters to point users to a specific row of data on an application business transaction page. The query string parameter can be the search record field name = field value.

These query parameters are stored as attributes with the content reference and are applied to the URL at runtime.

## Override Options

**Override Image** Select the image you want to display as the large image for the content reference. If a default content image has been defined at the system or registry level, this image will override the default. This large image displays for the content reference on the Navigation Collection publication.

See Chapter 9, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 105.

**Override Label** Enter a label to override label text in the Label field.

If you override label text, changes to label text on the source reference are not reflected on the Navigation Collection publication. If you want source reference changes to be reflected on the Navigation Collection publication, do not enter override text in this field.

**Override Description** Enter a description to override description text in the Description field.

If you override description text, changes to description text on the source reference are not reflected on the Navigation Collection publication. If you want source reference changes to be reflected on the Navigation Collection publication, do not enter override text in this field.

**Find Source**

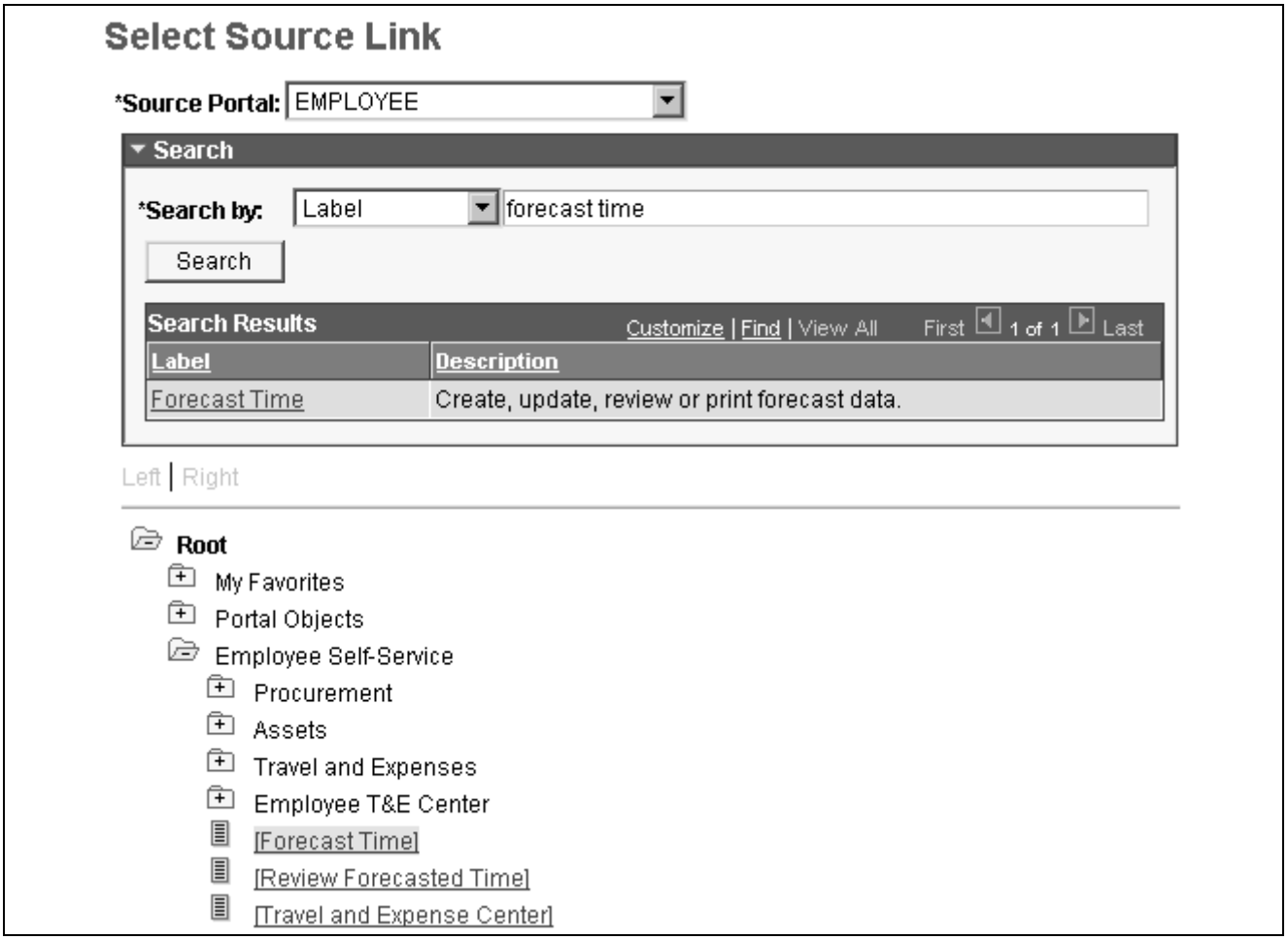
This link displays for users associated with the PTPT1300 permission list or Portal Administrator Role once a Source Link value has been selected.

Select to access the Content Ref Administration page, where you can view content reference details, such as security, image attributes, valid to and from dates, and content provider for the selected source content reference.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Administering Content References

## Selecting a Source Link

Access the Select Source Link page.




Select Source Link page

**Source Portal**

Select the portal registry from which you want to select a content reference. The portal designated on your originating page is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases when a Navigation Collection will be moved between portals, and the content reference you are defining should automatically refer to the current portal.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals"

### Search

**Search by**

*Description.* Select to conduct your search based on content reference description text. This search is not case sensitive. You can perform partial match searches. For example, doing a label search on "source" may not return any "source" hits, but may return hits on "resource."

*Label.* Select to conduct your search based on content reference label text.

Alternatively, you may perform a search by drilling down into the tree structure displayed at the bottom of the page.

### Search Results

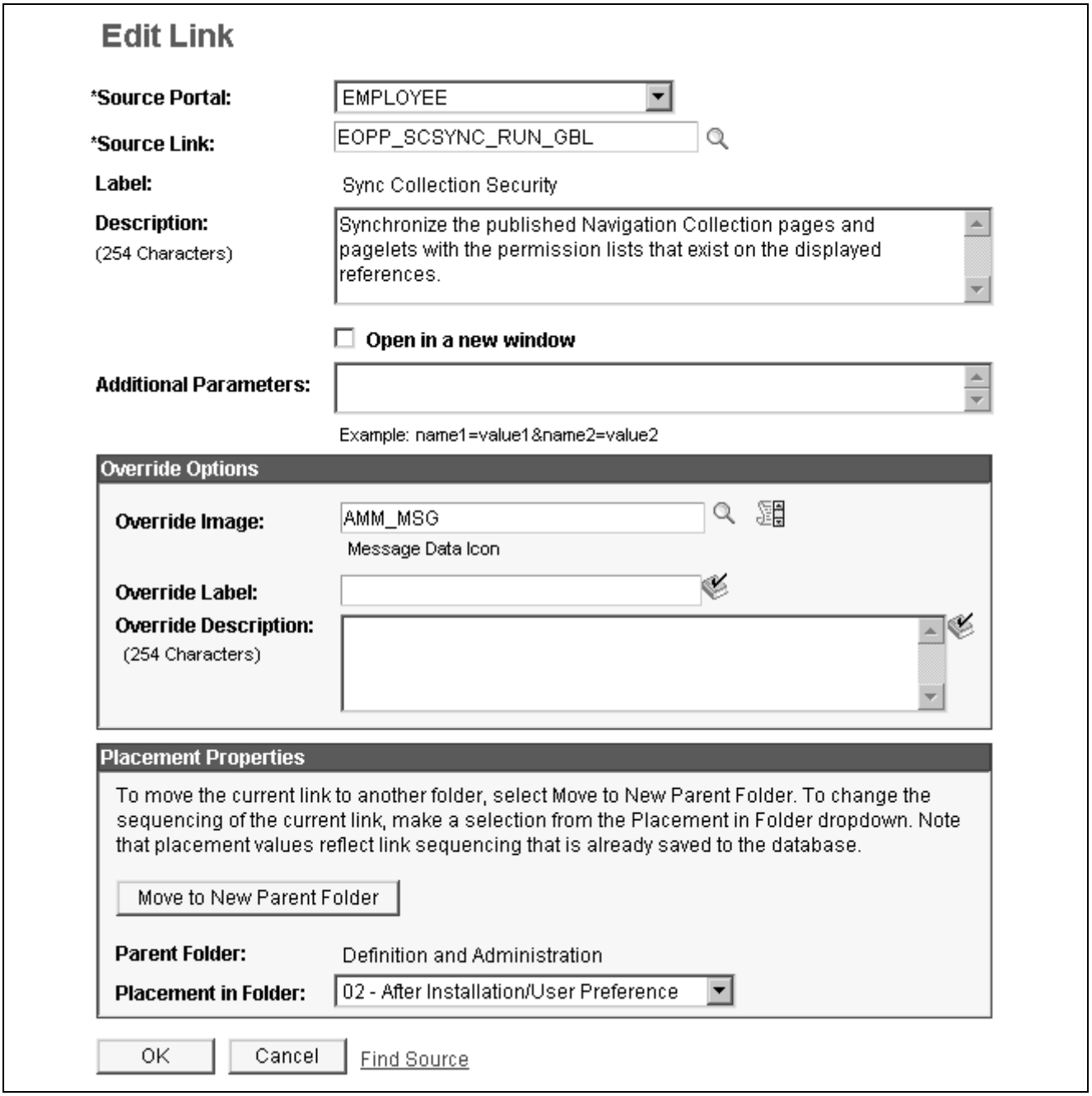
**Label** Select the linked label text to locate and select the content reference in the portal registry hierarchy displayed at the bottom of the page.

**Description** Displays the text description of the content reference.

Once you have located your desired content reference, click the folder link in the hierarchy to select the folder and return to your originating page.

## Editing a Link

Access the Edit Link page.




Edit Link Page

With the exception of the Placement Properties group box, which is discussed below, all other options on this page are available on and documented as a part of the Add Link page.

See Chapter 10, "Using Navigation Collections," Adding a Link, page 139.

## Placement Properties

**Move to New Parent Folder** Click to move the content reference you are editing to a folder other than the folder designated in the Parent Folder field.

**Parent Folder** Displays the parent folder of the content reference you are editing.

**Placement in Folder** Select a value in the drop-down list box to change the placement of the content reference within its selected Navigation Collection folder.

Available placement values reflect saved content reference sequencing.

# Publishing Navigation Collections

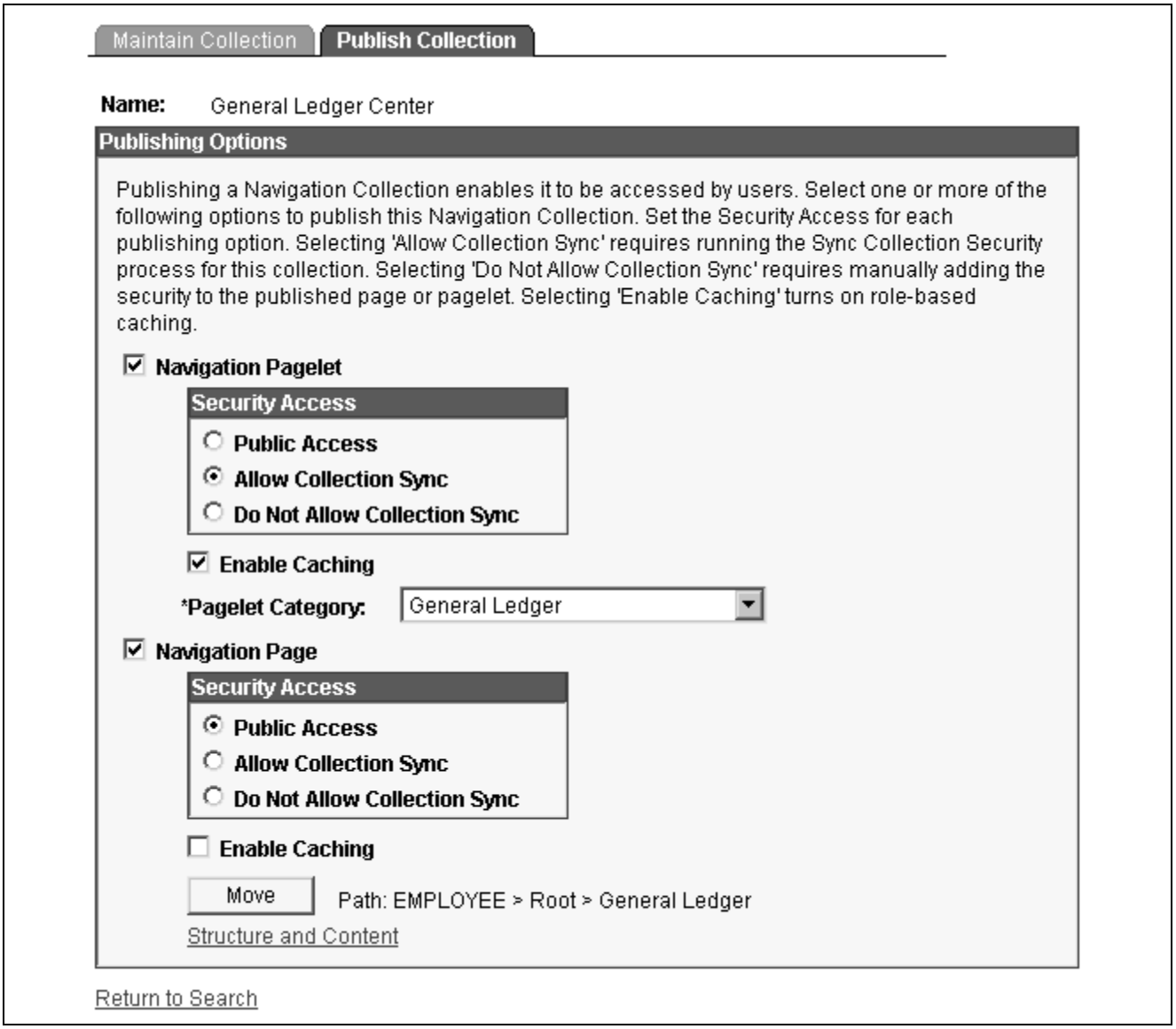
In this section, we discuss how to publish Navigation Collections.

## Pages Used to Publish Navigation Collections

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Publish Collection | EOPP_SCPUBLISHCOLL | Enterprise Components, Portal Utilities, Navigation Collections, Publish Collection | Publish the selected Navigation Collection to a navigation pagelet and/or page.<br><br>**Note.** The option to publish or access a Navigation Collection as a navigation pagelet is available if you have purchased PeopleSoft Enterprise Portal or a Portal Pack. |
| Select Folder | EOPP_SCBROWSEFLDR | Click the Move button on the Publish Collection page. | Select the parent folder in which you want the navigation page to display in the left-navigation menu. |

## Publishing a Navigation Collection

Access the Publish Collection page.




Publish Collection page

**Name** Displays the name of the Navigation Collection defined on the Maintain Collection page.

## Publishing Options

**Navigation Pagelet** Select to publish the Navigation Collection as a navigation pagelet, which can be added to a user's homepage tab.

**Note.** The option to publish or access a Navigation Collection as a navigation pagelet is available if you have purchased PeopleSoft Enterprise Portal or a Portal Pack.

**Security Access** *Public Access.* Select to designate that you want all users to be able to access the pagelet.

*Allow Collection Sync* (allow collection synchronization). Select to designate that you want the pagelet to be eligible for processing by the Sync Collection Security process.

*Do Not Allow Collection Sync* (do not allow collection synchronization). Select to designate that you do not want the pagelet security to be updated by the Sync Collection Security process. Select this option to ensure that any manually specified security you define for the pagelet cannot be unintentionally overridden by a run of the process.

See Chapter 11, “Running Portal Utility Processes,” Synchronizing Collection Security, page 149.

**Enable Caching**

Select if you want the this published pagelet to have role-based caching enabled. Selecting this option adds portal caching attributes to the published navigation pagelet.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Using Portal Caching Features

**Pagelet Category**

Select the category in which you want the navigation pagelet to appear for selection on the Personalize Content page.

Available pagelet categories are defined in the Structure and Content component. Select the Portal Objects folder link, then select the Pagelets folder link. Select the Add Folder link to add a pagelet category folder.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Managing Portal Objects

**Structure and Content**

Select to access the Content Ref Administration page, where you can access details about the content reference created for the navigation pagelet publication. For example, you can check on the results of the Sync Collection Security (synchronize collection security) Application Engine process (EOPP_SCSYNC) and change content reference label text.

See Chapter 11, “Running Portal Utility Processes,” Synchronizing Collection Security, page 149.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Administering Content References

**Navigation Page**

Select to publish the Navigation Collection as a navigation page, which can be accessed from the menu.

**Security Access**

*Public Access.* Select to designate that you want all users to be able to access the page

*Allow Collection Sync* (allow collection synchronization). Select to designate that you want the page to be eligible for processing by the Sync Collection Security process.

*Do Not Allow Collection Sync* (do not allow collection synchronization). Select to designate that you do not want the page security to be updated by the Sync Collection Security process. Select this option to ensure

that any manually specified security you define for the page cannot be unintentionally overridden by a run of the process.

See Chapter 11, “Running Portal Utility Processes,” Synchronizing Collection Security, page 149.

**Enable Caching** Select if you want the this published page to have role-based caching enabled. Selecting this option adds portal caching attributes to the published navigation page.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Using Portal Caching Features"

**Move** Click to access the Select Source Folder page, where you can select the parent folder in which you want the navigation page to display in the left-navigation menu.

See Chapter 10, “Using Navigation Collections,” Selecting a Source Folder, page 135.

**Structure and Content** Select to access the Content Ref Administration page, where you can access details about the content reference created for the navigation page publication. For example, you can check on the results of the Sync Collection Security process and change navigation page label text.

See Chapter 11, “Running Portal Utility Processes,” Synchronizing Collection Security, page 149.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals," Administering Content References

# CHAPTER 11

# Running Portal Utility Processes

This chapter provides overviews of portal utility processes and discusses how to:

- Synchronize Navigation Collection security.
- Delete empty portal registry folders.
- Clean portal projects.

## Synchronizing Collection Security

In this section, we discuss how to synchronize security permissions for your Navigation Collections using the Sync Collection Security (synchronize collection security) Application Engine process (EOPP_SCSYNC).

### Overview of the Sync Collection Security Process

The Sync Collection Security process updates security for published Navigation Collection pages and pagelets by adding permission lists and/or roles defined for content references and folders in the Navigation Collection to the published navigation page or pagelet permissions.

The Sync Collection Security process runs for Navigation Collection publications according to the Security Access options set for the publication on the Publish Collection page. The process runs for Navigation Collection publications that were published with the *Allow Collection Sync* option selected. The process is not run for Navigation Collection publications that were published with the *Do Not Allow Collection Sync* option selected. The process removes extraneous security objects for Navigation Collections that were published with the *Public Access* option selected.

This process should be run after security updates, and when you publish or change a navigation page or pagelet that was published with the *Allow Collection Sync* option selected.

#### See Also

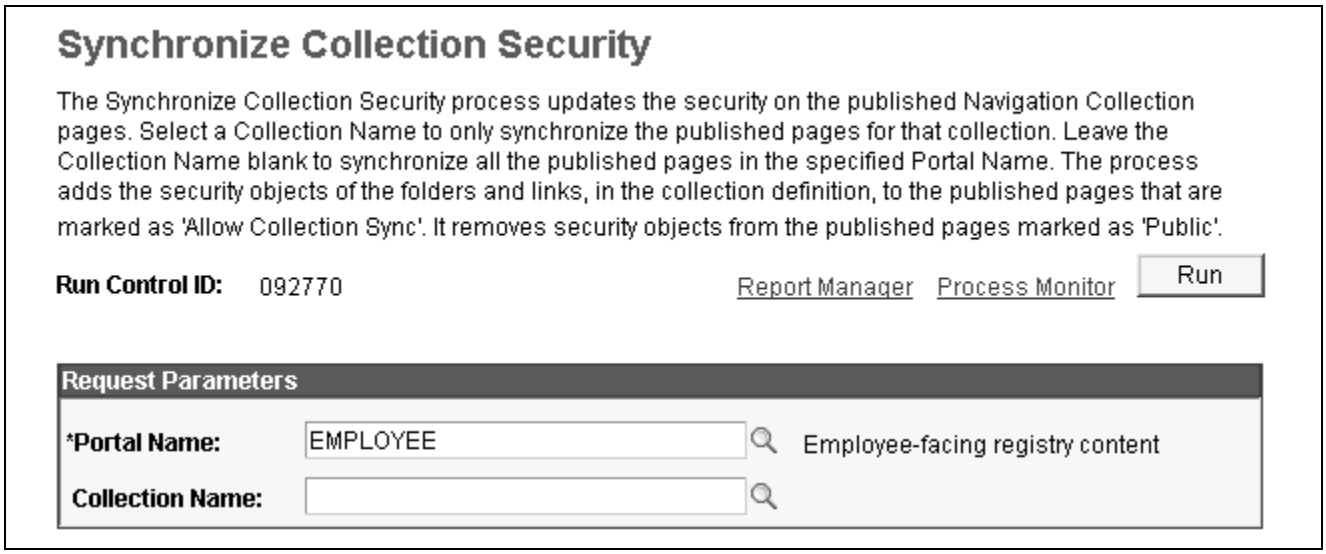
Chapter 10, “Using Navigation Collections,” Publishing Navigation Collections, page 144

### Page Used to Synchronize Collection Security

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Synchronize Collection Security | EOPP_SCSYNC_RUN | Enterprise Components, Portal Utilities, Sync Collection Security | Set run control parameters to run the Sync Collection Security process. |

## Running the Sync Collection Security Process

Access the Synchronize Collection Security page.




Synchronize Collection Security page

### Request Parameters

**Portal Name** Select the name of the portal for which you want to synchronize security for published Navigation Collections.

**Collection Name** Select the name of the Navigation Collection for which you want to synchronize security. Only Navigation Collections created for the selected portal are available for selection.

Do not select a Collection Name field value if you want to synchronize security for all non-public Navigation Collection publications that exist in the selected portal.

**Note.** Navigation Collection pages or pagelets that were published with the *Do Not Allow Security Sync* option selected on the Publish Collection page are not eligible for processing.

### See Also

Chapter 10, “Using Navigation Collections,” Publishing Navigation Collections, page 144

# Deleting Empty Portal Registry Folders

In this section, we discuss how to delete empty portal registry folders using the Delete Empty Folders Application Engine process (EOPP_FD_DEL).

## Overview of the Delete Empty Folders Process

The Delete Empty Folders process deletes portal registry folder references that do not contain child folders or content. The process does not delete empty Navigation Collection folder placeholders, nor the My Favorites folder. The user ID running the process must have been assigned the security role of Portal Administrator.

This process is only needed when you are deleting content references during an upgrade or implementation, and you are removing out-of-date navigation and replacing it with the current delivered navigation. This process should be run after copying a "delete" portal project.

## Page Used to Delete Empty Portal Registry Folders

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Delete Empty Folders | EOPP_FD_RUN | Enterprise Components, Portal Utilities, Delete Empty Folders | Set run control parameters to run the Delete Empty Folders process. |

# Cleaning Portal Projects

In this section, we discuss how to clean portal projects using the Clean Portal Project Application Engine process (EOPP_CPPROJ).

## Overview of the Clean Portal Project Process

The Clean Portal Project process removes unnecessary and potentially harmful data from a portal project definition before you copy it to the PeopleSoft Enterprise Portal. This process is run as a part of an upgrade or implementation when you want to load navigation from a content provider into the PeopleSoft Enterprise Portal.

The Clean Portal Project process removes common objects from the content provider portal project that, because they are common, also exist in the PeopleSoft Enterprise Portal. Run this process on the portal project before copying it to the PeopleSoft Enterprise Portal so that you don’t override these common objects configured for use with the PeopleSoft Enterprise Portal with the common objects that were configured in the content provider.

The Clean Portal Project process ensures that you only copy the navigation objects that you feel are necessary, such as registry structures, templates, permissions, and so forth.

***Warning!*** Navigation Collections cannot be rendered in the PeopleSoft Enterprise Portal if the content provider registry is hosted by a non-local node.

Because Navigation Collections actively access the registry, a published Navigation Collection page or pagelet cannot be rendered for a registry that is hosted by a non-local node. If you want to render a content provider Navigation Collection page or pagelet in the PeopleSoft Enterprise Portal, the content provider’s registry must be hosted by a local node in the content provider database.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Administering Portals"

To clean a portal project:

1. Create a project definition in PeopleSoft Application Designer.

Insert all permission lists and a portal registry definition. When inserting the portal registry, be sure to select *Portal Registry Structure* as the related definition.

a. Open PeopleSoft Application Designer.

b. Select File, New, Project.

c. Select Insert Definitions Into Project.

d. Set the Definition Type to *Portal Registry Definitions.* Click Enter.

e. Enter the portal name for the navigation you want to include in the portal project (Employee, Supplier, or Customer, for example).

f. Select the *Portal Registry Structures* value in the Related Definitions group box.

g. Click Insert.

h. Set the Definition Type to *Permission Lists.* Click Enter.

i. Click Select All.

j. Click Insert.

k. Select File, Save Project.

l. Enter a project name.

m. Click OK.

2. Run the Clean Portal Project process on the project. Use the run control page to select the navigation options you want to include in the process.
3. Delete the two-tier database cache using PeopleSoft Configuration Manager.
4. Export the clean project to file using PeopleSoft Application Designer.

### See Also

*PeopleTools 8.44 PeopleBook: Data Administration Tools*, "PeopleSoft Configuration Manager"

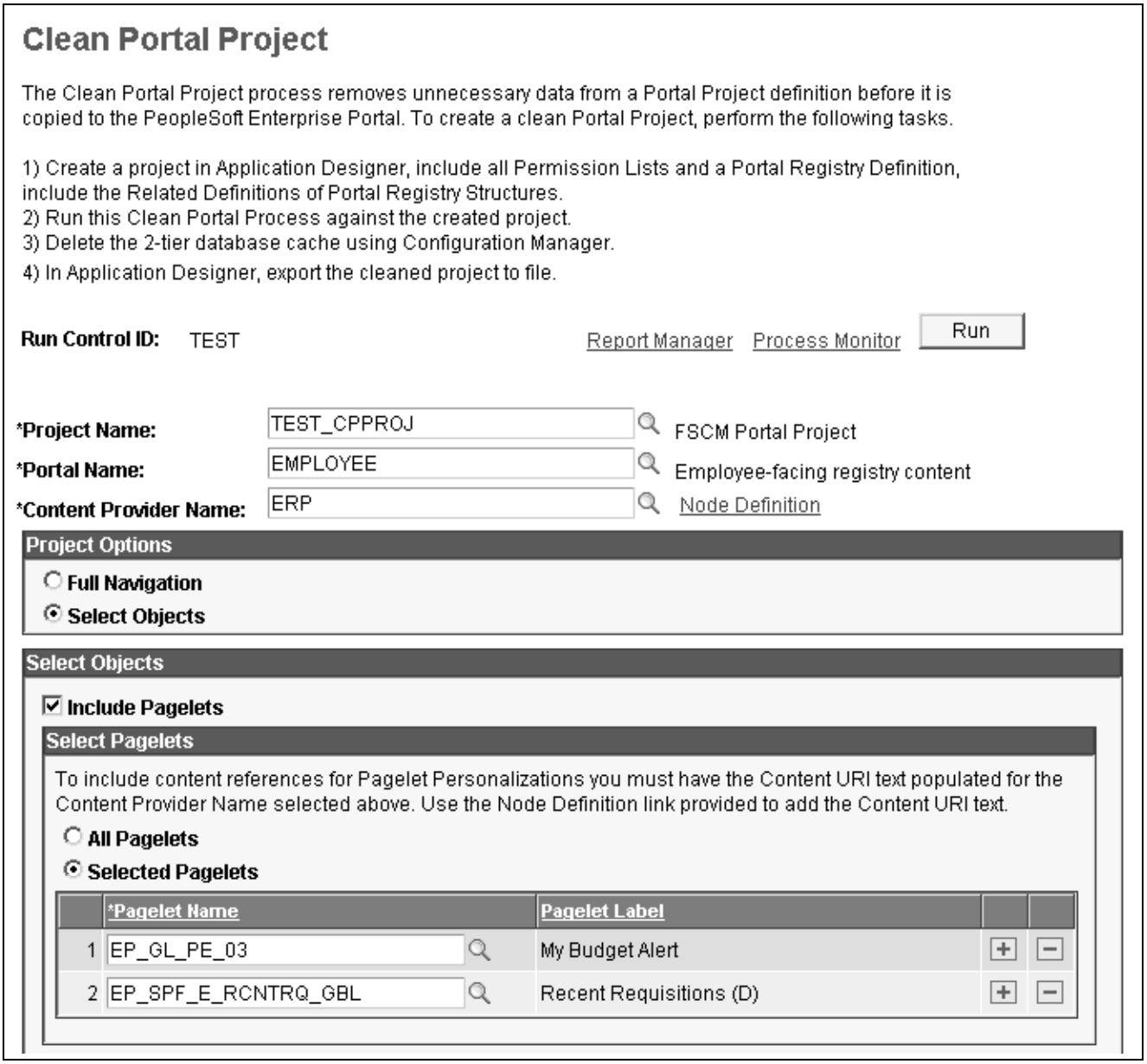
*PeopleTools 8.44 PeopleBook: PeopleSoft Application Designer*

## Page Used to Clean Portal Projects

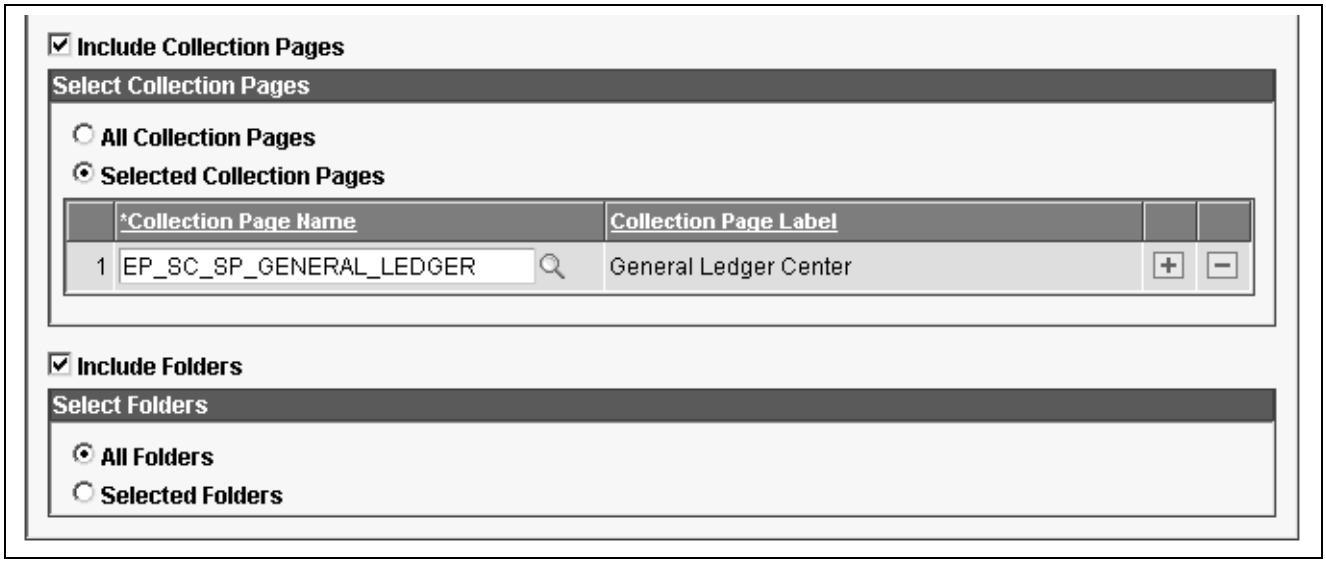
| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Clean Portal Project | EOPP_CPPROJ_RUN | Enterprise Components, Portal Utilities, Clean Portal Project | Set run control parameters for the Clean Portal Project process. |

## Running the Clean Portal Project Process

Access the Clean Portal Project page.




Clean Portal Project page (1 of 2)




Clean Portal Project page (2 of 2)

**Project Name** Select the name of the portal project you want to clean for export into the PeopleSoft Enterprise Portal. Projects available for selection are those in the database that contain at least one registry structure.

**Portal Name** Select the name of the portal that contains the registry structures you want to copy. Available portal names are derived from the portal project you selected, as well from as the project data that contains the portal name of the registry structures.

**Content Provider Name** Select the content provider node name of the registry structure you want to include in your portal project. Available node names are derived from the portal project you selected.

**Node Definition** Select to access the Node Definitions - Portal Content page. If you have selected the Include Pagelets option on this page, you must have a URL value in the node definition to enable the Clean Portal Project process to correctly include personalization pages for the pagelets you copy over to the PeopleSoft Enterprise Portal. Access the Node Definitions - Portal Content page to enter the content URI text for the content provider node name selected on this page.

When you select this link, you will be prompted to save or cancel any changes you have made in the component, and you will then be transferred to the Node Definitions - Portal Content page.

See *PeopleTools 8.44 PeopleBook: Internet Technology,* "Configuring Advanced Environment Features," Defining Portal Nodes

## Project Options

**Full Navigation** Select to run the Clean Portal Project process on all objects in the portal project.

**Select Objects** Select to run the Clean Portal Project process on select objects in the portal project. When you select this option, the Select Objects

group box displays, enabling you to select the type of navigation you want to include in your portal project.

Rather than copying over and having to maintain the full content provider navigation structure, you can select this option to display other options on the page that enable you to select only pagelets, navigation pages, or folders for copying over to the PeopleSoft Enterprise Portal.

## Select Objects

**Include Pagelets** Select to include pagelets (including Navigation Collection pagelets) in the portal project in the run of the Clean Portal Project process. When you select this option, the Select Pagelets group box displays, enabling you to specify which pagelets you want to include in the run of the process.

The actual published Navigation Collection pagelet content references are included for copying to the PeopleSoft Enterprise Portal. The Navigation Collection definitions are not included.

Pagelet personalization pages will be included in the portal project as long as the content provider node name contains the correct URL. Select the Node Definition link to access the Node Definitions - Portal Content page, where you can enter this URL.

**Include Collection Pages** Select to include Navigation Collection pages in the portal project in the run of the Clean Portal Project process. When you select this option, the Select Collection Pages group box displays, enabling you to specify which Navigation Collection pages you want to include in the run of the process. The actual published Navigation Collection pages are included for copying to the PeopleSoft Enterprise Portal. The Navigation Collection definitions are not included.

**Include Folders** Select to include folders in the portal project in the run of the Clean Portal Project process. When you select this option, the Select Folders group box displays, enabling you to specify which folders you want to include in the run of the process.

Selecting a folder for inclusion in the process also includes all of its children and parents for copying into the PeopleSoft Enterprise Portal.

## Select Pagelets

**All Pagelets** Select to include all pagelets in the portal project in the run of the Clean Portal Project process.

**Selected Pagelets** Select to include only selected pagelets in the portal project in the run of the Clean Portal Project process. Selecting this option displays the Pagelet Name field, enabling you to select specific pagelets that you want to include in the run of the process.

## Select Collection Pages

**All Collection Pages** Select to include all published Navigation Collection pages in the portal project in the run of the Clean Portal Project process.

**Selected Collection Pages** Select to include only selected published Navigation Collection pages in the portal project in the run of the Clean Portal Project process. Selecting this option displays the Collection Page Name field, enabling you to select specific published Navigation Collection pages that you want to include in the run of the process.

## Select Folders

**All Folders** Select to include all folders in the portal project in the run of the Clean Portal Project process.

**Selected Folders** Select to include only selected folders in the portal project in the run of the Clean Portal Project process. Selecting this option displays the Folder Name field, enabling you to select specific folders that you want to include in the run of the process.

# PART 5

# Enterprise Integration

# CHAPTER 12

# Understanding Enterprise Integration

This chapter gives an overview of PeopleSoft's Enterprise Integration and discusses:

- PeopleSoft Messaging.
- PeopleSoft Business Interlinks.
- PeopleSoft Component Interfaces.
- File layouts.
- PeopleSoft Integration Broker.

## Understanding PeopleSoft Messaging

PeopleSoft Messaging provides the ability to synchronize data from one application or system with another. A message defines the records and fields to be published or subscribed to. At runtime, XML is built to represent the message structure and application data, though you can build, publish, and subscribe to messages without knowledge of XML.

## Understanding PeopleSoft Business Interlinks

PeopleSoft Business Interlinks enables you to perform component-based, real-time integration from PeopleSoft to external systems. PeopleSoft Business Interlinks creates synchronous transactions that enable PeopleSoft applications to pass data to and receive data from the external system in real time. You can use PeopleSoft Business Interlinks to integrate PeopleSoft with third-party systems, with another PeopleSoft application, or with systems on the internet.

## Understanding PeopleSoft Component Interfaces

A component interface is a set of application programming interfaces (APIs) that you can use to access and modify PeopleSoft database information using a program instead of the PeopleSoft client. PeopleSoft Component Interfaces expose a PeopleSoft component (a set of pages grouped for a business purpose) for synchronous access from another application (PeopleCode, Java, C/C++, or Component Object Model [COM]). A PeopleCode program or an external program (Java, C/C++, or COM) can view, enter, manipulate, and access PeopleSoft component data, business logic, and functionality.

# Understanding File Layouts

A file layout is a definition (or mapping) of a file to be processed. It identifies where fields are located in file data. Once you create a file layout, you can write PeopleCode programs that ultimately use file layout, either to read data from or write data to a file.

In addition to manipulating transaction data, you can use file layouts and flat files to move data between your PeopleSoft database and external systems (data interchange).

# Understanding PeopleSoft Integration Broker

PeopleSoft Integration Broker facilitates synchronous and asynchronous messaging among internal systems and trading partners, while managing message structure, message format, and transport disparities.

PeopleSoft Integration Broker comprises two high-level subsystems: the integration engine and the integration gateway. The integration engine runs on the PeopleSoft application server. It is tied closely to PeopleSoft applications and produces or consumes messages for these applications.

The integration gateway manages the receipt and delivery of messages passed among systems through PeopleSoft Integration Broker. It provides support for the leading TCP/IP protocols used in the marketplace today, and more importantly, provides extensible interfaces for the development of new connectors for communication with legacy and internet-based systems.

# CHAPTER 13

# Understanding Enterprise Integration Points

This chapter discusses enterprise integration points (EIPs).

## Overview of EIPs

Provided by a PeopleSoft application, an EIP is an interface that is used to communicate with other PeopleSoft or external applications. An EIP provides integration details for PeopleSoft applications, including which technologies are involved, technology details, integration broker information for messaging, and how different EIPs are related.

The EIP consists of data rules for the applications that it supports. The EIPs that are delivered with PeopleSoft provide generic functionality so that they can be adapted for use with as many programs as possible.

An EIP can be implemented by using different technologies available in PeopleTools, such as messaging, component interfaces, business interlinks, XML links, and electronic data interchange (EDI).

EIPs can be associated with or used by application groups. An application group is a logical grouping of applications that use an EIP in the same business manner.

Other than this grouping facility, an application group and an application mean the same thing. In the rest of the documentation, the words *application group* and *application* are used interchangeably unless clearly specified.

Every EIP is owned by at least one application, but can be used in multiple applications. Therefore, if an application sends an EIP, and another application can receive the same EIP, the two systems should be interoperable, assuming the data structure and the rules of the EIP are implemented the same in both places.

However, sometimes two applications might use the same EIP but implement it in different ways. For example, one application that uses the Customer EIP may need to transform the data before it can be sent to an external system, which has another data structure for its customer information.

An EIP can be a part of multiple application groups. For example, the Department Table EIP may be used by a number of application groups, including PeopleSoft Human Resources, CRM, and General Ledger.

# CHAPTER 14

# Activating Messaging EIPs

This chapter discusses how to:

- Set up PeopleSoft Messaging EIPs.
- Set up related languages.
- Examine related-language messaging scenarios.

## Setting Up PeopleSoft Messaging EIPs

This section discusses how to:

- Activate messages for publication.
- Set up publication rules.
- (Optional) Map nodes to a chunking rule.
- (Optional) Assign business units to a chunk rule.
- (Optional) Assign setIDs to a chunking rule.
- Specify OnRoute PeopleCode.

**Note.** For more information and technical details about these EIPs, and for information on how and with what applications to use them, consult the relevant application PeopleBook.

## Pages Used to Set Up PeopleSoft Messaging EIPs

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Message Properties | EOEI_MSGDFN_PG | Enterprise Components, EI Repository, Message Properties | Activate messages. |
| Batch Publish Rules | EO_MSGPUBATCH | Enterprise Components, Integration Definitions, Batch Publish Rules | Set up publication rules.<br>You must activate a publication rule for the publication messages that you create to follow. This rule includes instructions on message chunking, if necessary. |
| Add Nodes to Chunk Rule | EO_ADNODECHUNK_PNL | Enterprise Components, Map Chunking Rules, Node to ChunkRule | Map PeopleSoft message nodes to chunking rules. |

## Activating Messages for Publication

PeopleSoft delivers messages with a default status of *Inactive*. You must activate each message before attempting to publish or subscribe to messages. You can activate messages by using either the enterprise integration repository or PeopleSoft Application Designer.

### Activating Messages Using PeopleSoft Application Designer

To activate a message for publication by using PeopleSoft Application Designer:

1. Open PeopleSoft Application Designer.
2. Select File, Open.

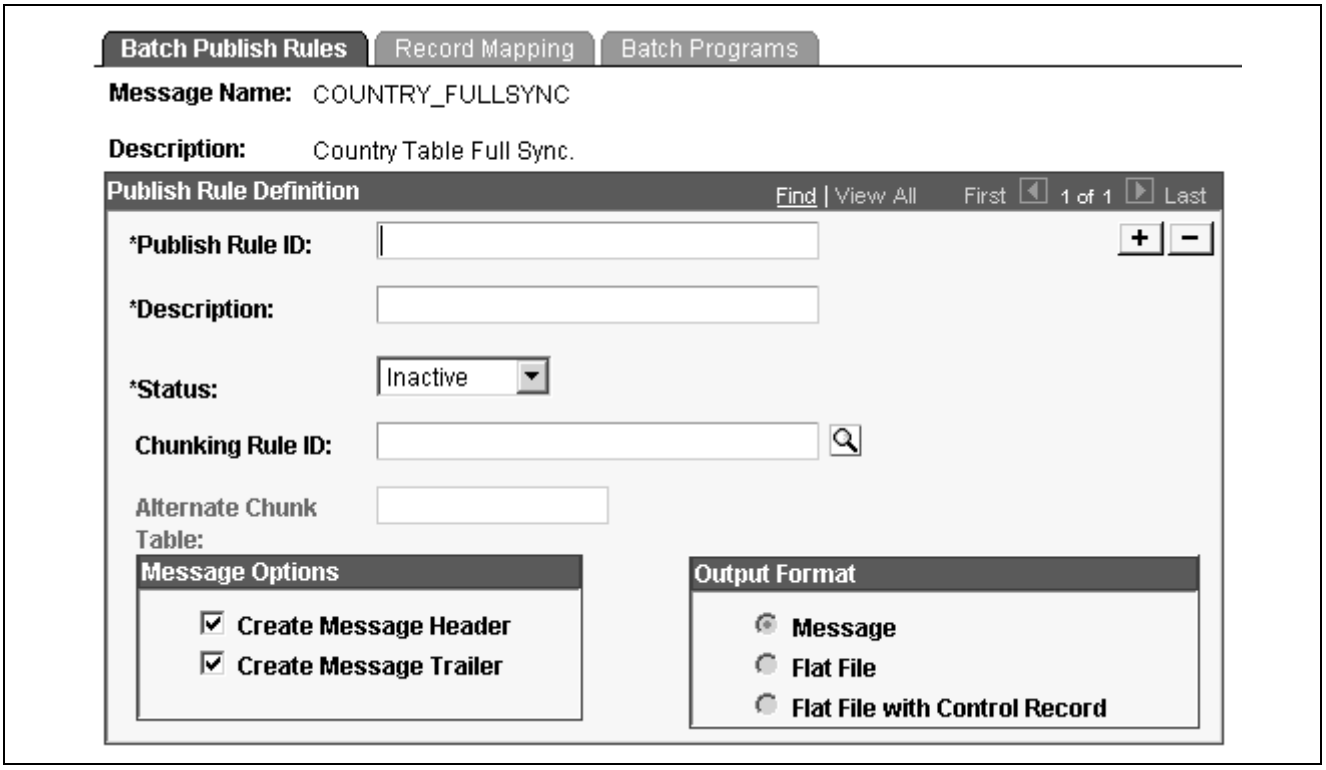
   The Open Object dialog box appears.
3. In the Definition menu, select *Message*.
4. Enter the message name in the Name field.
5. Click Open.

   The message that you specified opens.
6. Select File, Definition Properties.

   The Message Properties dialog box appears.
7. In the Message Properties dialog box, select the Use tab.
8. Select the Active check box to activate the message.
9. Click OK.
10. Select File, Save to save the message definition.

## Setting Up Publication Rules

Access the Batch Publish Rules page.




Batch Publish Rules page

If the data that you're transmitting does not fit in a single message, or if you want to send different parts of the message to different target systems, set up the rules to chunk the message and associate it with the publish rule. The business unit and setID chunking rules are standard in PeopleSoft applications, but you can configure chunking rules.

**Publish Rule ID** — Select the name of the message for which you're setting up rules.

**Status** — Select *Active* to activate this publish rule definition for this message. Select *Inactive* to prevent this rule from applying to this message.

**Chunking Rule ID** and **Alternate Chunk Table** — Enter the unique chunking rule name that is set up when you created the chunking rule. The message that you publish is routed based on this field. If necessary, enter an additional field in the Alternate Chunk Rule ID field by which to chunk the message.

## Message Options

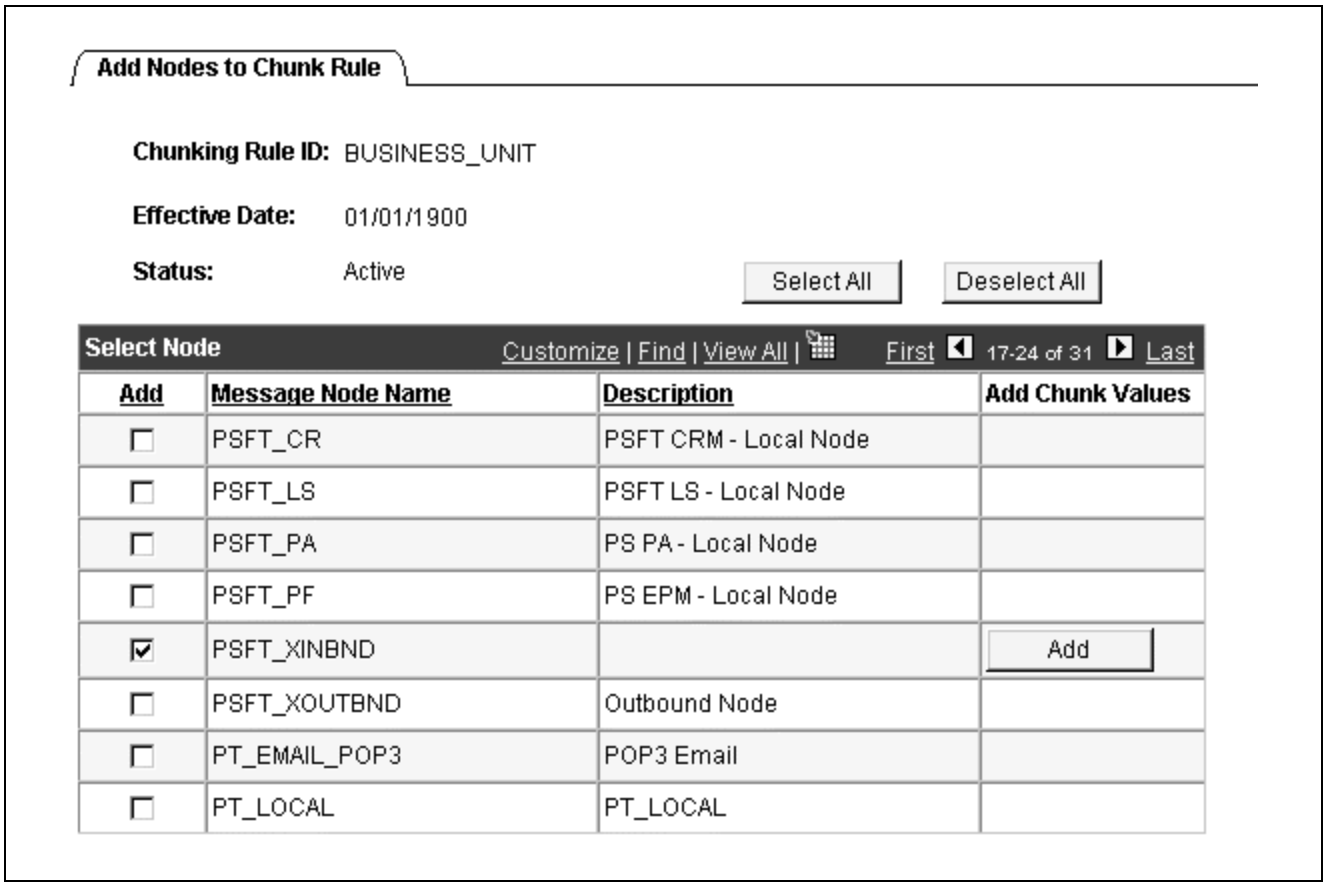
Many PeopleSoft systems rely on a message header and message trailer to trigger subscription PeopleCode to discard old table data and insert the new incoming data. As a general rule, all FullSync messages should use a header and trailer. Sync messages don't need headers and trailers.

## Output Format

The Application Engine program used to chunk messages can create either an XML message that flows through messaging architecture or a flat file that is generated in PeopleSoft Process Scheduler and not published elsewhere. Always select Message as the format when you send data to PeopleSoft systems.

## Mapping Nodes to a Chunking Rule

Access the Add Nodes to Chunk Rule page.




Add Nodes to Chunk Rule page

To map nodes to a chunk rule:

1. In the Add column, select the check box next to the nodes that you defined earlier.

   After you select a node, use the Add button in the Add Chunk Values column to open the Quick Map page for the message you defined earlier.

2. Click Save.

## Assigning Business Units or SetIDs to a Chunking Rule

See Chapter 15, “Assigning Publishing Rules ,” Assigning Business Units or SetIDs to a Chunking Rule, page 188.

## Specifying OnRoute PeopleCode

Specify the PeopleCode program in the OnRoute event that you use to route the message chunk to the correct subscriber node.

To set up OnRoute PeopleCode:

1. Open PeopleSoft Application Designer.

2. Select File, Open.
3. Select *Message* in the Definition menu.
4. Enter the message name in the Name field.
5. Click Open.

   The system opens the message definition.
6. Select View, View PeopleCode.
7. Select *OnRouteSend* from the PeopleCode Event drop-down list box.

   The system displays the PeopleCode editor. The following is an example of an OnRoute Send PeopleCode program:

```
Declare Function GetNodes PeopleCode FUNCLIB_INEIP_PUBLISH_ROUTE_PC FieldForumula;
GetNodes (" ");
```

8. Click Save.

# Setting Up Related Languages

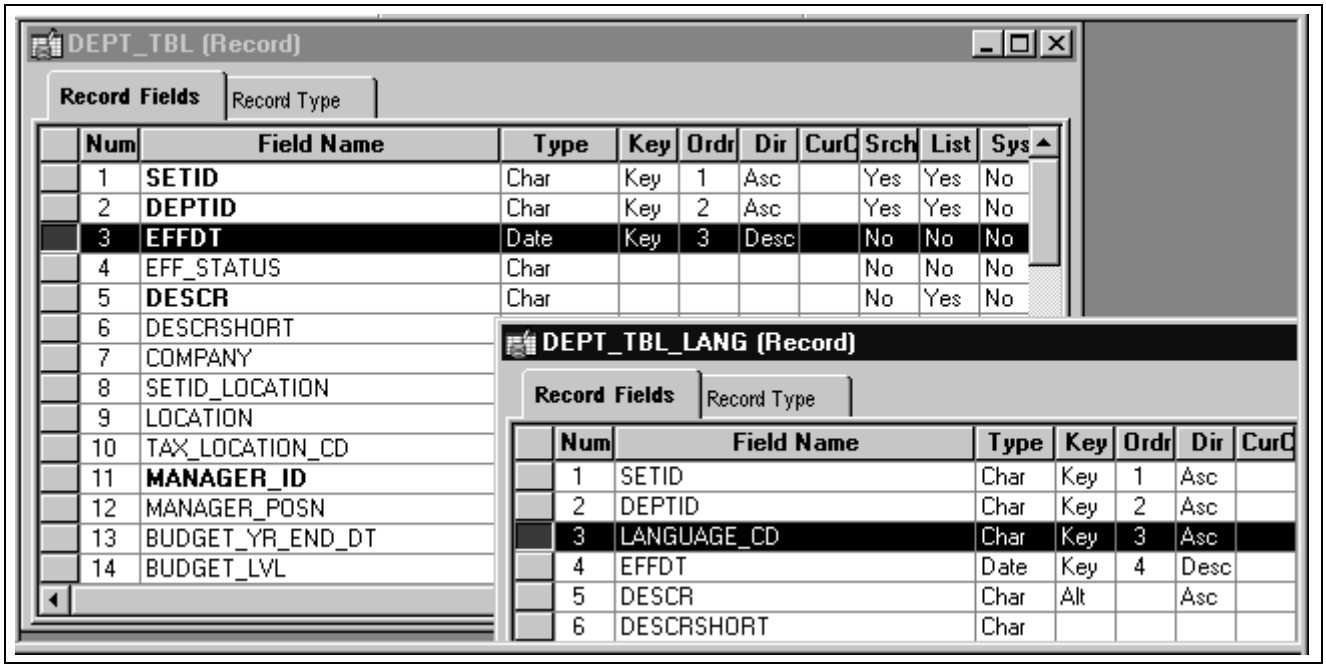
This section provides an overview of related language tables and related language guidelines for PeopleSoft Messaging and discusses how to:

- Interpret the component processor behavior.
- Publish a message from a component.
- Publish a message from batch programs.
- Subscribe to data in a PeopleSoft multilingual environment.
- Subscribe to data in a non-multilingual environment.

## Understanding Related Language Tables

There are several possible scenarios that you can use to familiarize yourself when setting up related languages for a message.

A department table, for example, must publish information in German as well as English. In the following screen shot, the base application tables and related-language tables have a parent-child relationship. The related-language table has the same name as the parent table, but is suffixed with _LANG, in accordance with PeopleSoft naming conventions.




Related-language record definitions

Consider the following:

- The parent table, DEPT_TBL, has the related-language child table, DEPT_TBL_LANG.
- The child table has the same fields as the parent table, plus an additional field of LANGUAGE_CD.
- The child table's attributes are all of the translatable textual fields of the parent record.

## Understanding Related Language Guidelines for Messaging

When publishing a full message, generate messages that contain the contents of an entire table by first generating a message in the base language of the system that contains the full table contents. Then generate messages for each of the related languages that are supported by the publishing system. Each message should contain the full message structure for that message object (levels 0, 1, 2, and so on). The language-specific messages should contain the translatable field values for that language and include the base language fields that are not translatable.

When subscribing to a full message, specify the language code only at level 0 of the message. This captures and sets the user's preferred language to level 0 of the PeopleSoft Common Application Message Attributes (PSCAMA) message header. All data within the message must be in the same language. Follow these steps:

1. Delete the base language tables and related-language tables.
2. Replace these tables with data from the messages as appropriate.
3. Place only those related-language field values that are supported by the subscribing system into the related-language tables.
4. Add the related-language table entry only if the base language table entry already exists.

When publishing an incremental message, the PeopleSoft system generates base messages in the user's preferred language by using the user ID's language code. Putting the user's preferred language code in the message header PSCAMA record defines the message language for the subscribing system.

When subscribing to an incremental message using PeopleSoft Component Interfaces, use a simple PeopleCode program that performs a SetLanguage (messagelanguage) call to a component interface with the message definition. This enables the subscribing system to process the data in the appropriate related or base language for that system.

When subscribing to an incremental message using PeopleCode only, the PeopleCode program must simulate what the component processor does. The PeopleSoft system provides a generic Subscribe_IncrReplication PeopleCode function that provides basic language-related ability for incremental message subscriptions.

---

**Note.** All PeopleSoft subscription processes that are associated with textual information work as if the content is related-language enabled; thus the processes provide support for customer related-language extensions and future PeopleSoft enhancements.

For PeopleSoft-to-PeopleSoft system integration, you do not need to specify the language-sensitive data on either system.

---

All of the PeopleCode functions that are needed for related language processing of incremental and full messages are in the FUNCLIB_EOEIP record. The record contains two functions:

- Subscribe_IncrReplication has related-language processing for an incremental message subscription process.
- Subscribe_FullReplication has related-language processing for a batch subscribe process.

## Interpreting Component Processor Behavior

When you open a component, the component processor:

1. Compares the user's preferred language against the base language for the database.
2. Uses the record information from the base application table (DEPARTMENT_TBL).

   If a record in the base application table exists for the user's preferred language, the fields on the related-language table (DEPARTMENT_LANG) overlay the record information. For example, a German user sees German descriptions even if the base language for the database is English.

When you change the user's preferred language and save the component:

1. The component processor writes all the data for related-language fields back to the related-language table.
2. The component processor writes the rest of the data back to the base application table.
3. The German user's entries for the DESCR and DESCRSHORT fields are saved back to the DEPARTMENT_LANG table with its key values and the LANGUAGE_CD field in German.
4. The data that was entered by the German user in the key fields, as well as MANAGER_NAME and ACCOUNTING_OWNER fields, are saved on the parent record DEPARTMENT_TBL.

## Publishing a Message from a Component

The PeopleSoft system employs the user's preferred language to determine the language of a message that is published from a component. The default for LANGUAGE_CD is set to the preferred language code (OPERATOR.LANGUAGE_CD). The standard for incremental changes is to publish only the data that has changed, in the language to which it was changed. Changing the preferred language to translate data generates new messages appropriately.

# Publishing a Message from Batch Programs

Application Engine and Structured Query Report incremental message programs should use the base language of the system. These programs perform their accesses and updates on the base tables only, even if related-language tables are supported for those business objects. Related-language tables are featured in batch program processing only in generating warnings or errors that use the message catalog.

When a batch application program runs, the processing is done in the base language of the system, and messages are generated in only the base language.

# Subscribing to Data in a PeopleSoft Multilingual Environment

The subscription process sets the language for processing the message to the language needed by the subscriber system. For example, if the subscribing system's base language is French, and the PeopleSoft system is sending German data, the subscription process must store the German data in a related-language table and the nontranslatable data in the French base application tables. Neither system's base language matters; only the base language of the subscribing system and the actual message language are used.

# Subscribing to Data in a Non-Multilingual Environment

When handling subscribing systems that do not support multiple languages, you can subscribe to data in these ways:

- Setting specific message publish routing PeopleCode.
- Sending messages to appropriate nodes.
- Permitting the subscription to process itself.
- Subscribing to data that is specific to an external system.

## Setting Message Publish Routing PeopleCode

Set message publish routing PeopleCode to send only messages in a particular language code to a subscribing node.

The subscribing node does not need to check the language in which the message was generated; any message that it receives is in its language is automatically used to update the subscribing system's database. To implement this option:

- Determine the languages in which messages are published.
- Determine the message language that a subscribing node receives.
- Add PeopleCode routing logic on the publish side to check the language code of the first occurrence in the message record and return a list of subscribing nodes that should receive the message for that language.

The publish routing PeopleCode guarantees that the message is sent to the correct subscribing nodes by using the message language code.

## Sending Messages to All Appropriate Nodes

Send messages to all appropriate nodes regardless of the language and have the subscription routing PeopleCode filter out messages in different languages.

PeopleCode on the message subscription routing checks the language code of the first occurrence in the message and controls whether the node should receive the message. To implement this option:

- Ensure that messages in all languages are sent to all appropriate nodes.
- Add the PeopleCode to compare the message language against a hard-coded language value for the subscribing system.

The advantage of putting the logic within the subscription routing PeopleCode is that every message is checked for a language value match.

### Permitting the Subscription to Process Itself

You can permit the subscription process itself, rather than the routing PeopleCode for the message and channel, to determine whether the message should be processed for the subscribing system.

The subscription process checks the language code of the PSCAMA record for the first instance of the message against a hard-coded value for the subscribing system. If the language code does not match, the message is ignored. If the message language code does match, it's considered a base language message, and it replaces all data on the subscribing system according to the audit action flags on the message records.

**Note.** Generic subscription processes should not filter messages based on language code, to avoid data integrity issues.

### Subscribing to Data That Is Specific to an External System

Subscribe to data that is specific to an external system for language code, business unit, or setID requirements that are specific to an external system.

Use the chunking rule and the routing control tables that PeopleSoft supplies to select a portion of the data and send it to a specific node.

Use the PeopleSoft-supplied Publish Header (PublishHdr) component to enter the partitioning views and fields for a message. This chunks the message so that all contents within a single message contain the same partitioning value (such as business unit, setID, or application-specific fields).

Set up the routing control for the message so the message is sent only to the appropriate nodes. The PeopleSoft system supplies a business unit routing control (BU Routing Control) component and a setID routing control component that enable applications to specify for each message which nodes should receive the partitioned message data.

After you set up the chunking rule and routing rules, both the full data publish and batch publish programs partition the data according to the appropriate value and route it accordingly. You can now publish and subscribe to the message.

## Examining Related-Language Messaging Scenarios

Actual related-language messaging scenarios include publishing a non-base language message and switching a preferred language.

This section discusses how to:

- Publish a non-base language message to a PeopleSoft subscribing system with a different base language and no prior data.
- Switch the preferred language to Japanese and update the same employee's name and address.

## Publishing a Non-Base Language Message to a Subscribing System With a Different Base Language and No Prior Data

In the following example, the publishing system base language is English, the subscriber base language is Japanese, and the user's preferred language is German.

To publish a non-base language message:

1. An online user adds a new level 0 key on a page.

   The data is stored in both the base language table (English) and the related-language table (German).
2. The system publishes a message in the user's preferred language (German).
3. The data is inserted into the subscribing system.

   The data is inserted into both the base language table and related-language table (German) because it is added for the first time; data cannot reside in a related-language table without corresponding data in the base table.

## Switching the Preferred Language to Japanese and Updating the Same Employee's Name and Address

To switch a preferred language:

1. A user switches the preferred language from English to Japanese.

   The user updates the same record as in the previous scenario. Data that is not language-sensitive is updated in the base table (English). Language-sensitive data is inserted into the related-language table (Japanese).
2. A system publishes a message in the user's preferred language (Japanese).
3. Data is inserted into the subscribing system.

All data goes to the base table (Japanese), because the message was sent in the same language as the subscribing system's base language.

# CHAPTER 15

# Assigning Publishing Rules

This chapter provides an overview of message publishing rules and discusses how to:

- Assign full table publishing rules.
- Assign batch publishing rules.
- Set up message chunking.

## Understanding Publishing Rules

This section discusses:

- The Publish utility.
- Terms used when defining rules.

### The Publish Utility

The Publish utility automates the process of copying the contents of an entire table into a remote database or legacy system. Use the utility to synchronize data from an existing system when a new PeopleSoft system is installed. The data is chunked based on the MaxMessageSize parameter.

Control the size, number, and frequency of these data messages by using a series of data publishing rules and:

- Whether to create header and trailer records.
- Where the data comes from.
- Whether to chunk the message.
- Which related languages to publish.

The publishing rules include pages for full table publishing rules, batch publishing rules, record mapping, languages, and batch programs.

The MaxMessageSize (maximum message size) field in the PeopleSoft Option (PSOPTIONS) table limits the size of the message. Before processing each level zero record, the Publish utility compares the size of the message against the value in the MaxMessageSize field. When the message size exceeds the value in the MaxMessageSize field, the message publishes and a new message starts. You can also specify message chunking in the publish rules for the message, which enables the message to publish when the value of a chunk field changes.

## Terms Used When Defining Publish Rules

When defining publish rules, you need to know the following terms:

**Alternate Chunking Table** This secondary chunk table is a separate view of an existing chunk table but with one or more field names customized. It enables you to reuse an existing chunk table. For example, if the record EO_BUSUNT_EOC has BUSINESS_UNIT as the chunking field, you can create a view of this table that has BUSINESS_UNIT_IN as the chunking field.

**Batch Publish** This term describes jobs or processes that run independently from their initiating process. A batch process can also run at one or more predetermined times in the future from the initiating request. A batch process is appropriate for publishing incremental changes to data in a batch environment or when processing large volumes.

**Chunking** Chunking refers to the automatic breaking of a message into several smaller messages based on values in fields in the level zero record. Chunking on business unit means that all transactions within the message are for the same business unit value.

**Chunking Fields** Chunking fields are key fields in the level zero record that are used to break the message into parts.

**Chunking Rule** A chunking rule points to the chunking table. Multiple chunking rules can point to the same chunking table.

**Chunking Table** A chunking table is a derived or Structured Query Language (SQL) table that contains the fields by which the message is chunked. SQL chunk tables define the valid values of the chunking fields and the nodes to which the message is published.

**Message Header** The header is data that precedes a message and triggers the subscription process to initialize tables before receiving the data messages.

**Message Trailer** The trailer is data that follows a message and triggers the subscription process to indicate that all data messages have been received.

**Full Publish** The full publish process seeds, or initially populates or repopulates, a copy of an entire table into a remote database or legacy system. The entire contents of the table are published to all systems that require a copy of the table.

# Assigning Full Table Publishing Rules

All PeopleSoft applications use common, centralized tables and pages to define how to publish full table messages. The Publish utility uses full table publish rules to process the data.

This section discusses how to:

- Associate a rule to a message and characterize the rule.
- Map a message record to another record.

• Specify languages in which to publish messages.

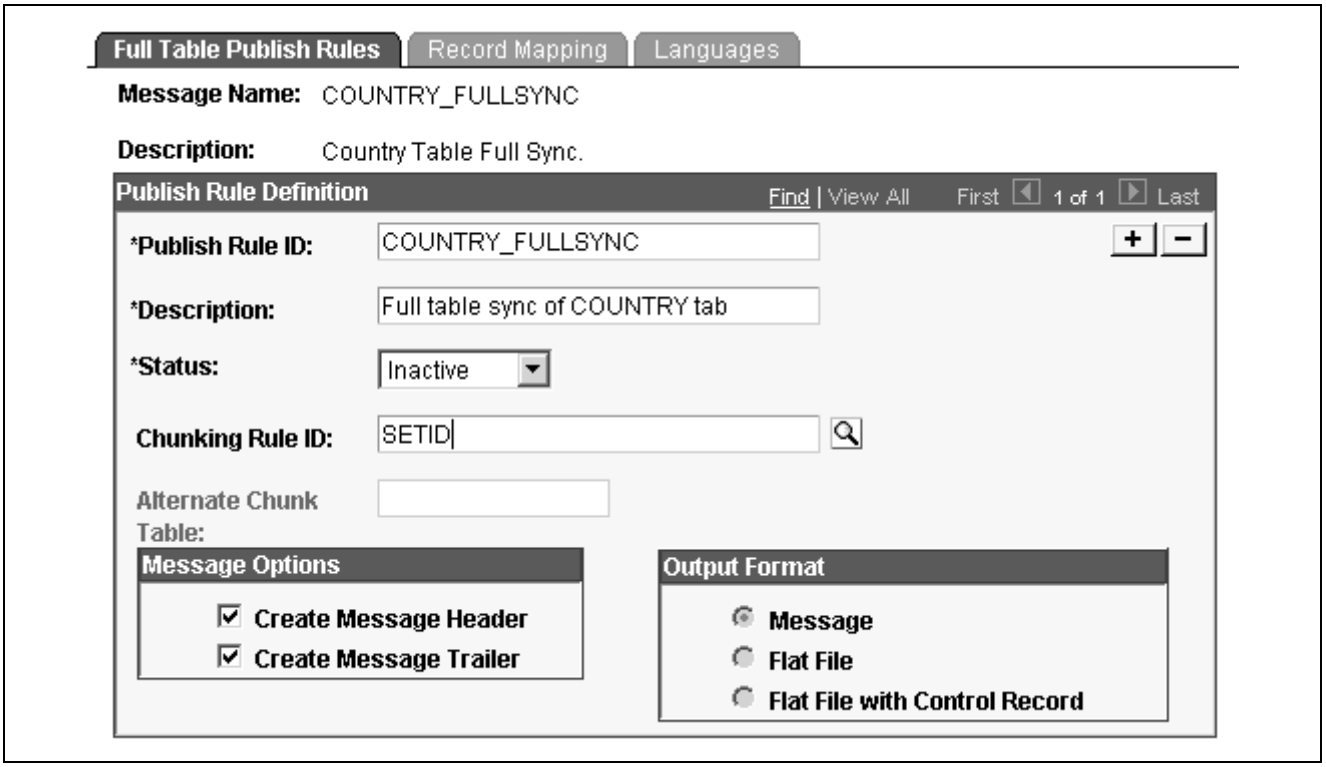
**Note.** You can create multiple publish rules for the same message. The Publish utility treats each publish rule as a separate publishing cycle.

## Pages Used to Assign Full Table Publishing Rules

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Full Table Publish Rules | EO_MSGPUBFULL | Enterprise Components, Integration Definitions, Full Data Publish Rules | Associate a rule to a message and characterize the rule. |
| Record Mapping | EO_MSGRECMAP | Enterprise Components, Integration Definitions, Full Data Publish Rules, Record Mapping | Map a message record to another record. |
| Languages | EO_MSGLANGUAGE2 | Enterprise Components, Integration Definitions, Full Data Publish Rules, Languages | Specify languages in which to publish a message. |

## Associating a Rule to a Message

Access the Full Table Publish Rules page.




Full Table Publish Rules page

**Publish Rule ID** Enter an ID for the user-defined rule to associate with this message. Assign a logical name to make the rule easy to find.

| | |
|---|---|
| **Status** | Select *Active* to activate the publish definition. |
| **Create Message Header** and **Create Message Trailer** | Ensure that the subscribing process does not need the header or trailer process before you clear these check boxes for a Batch Publish message. Header messages trigger special logic (in a PeopleCode program) on the PeopleSoft full message subscription that deletes the existing application records. Also, some applications use the trailer message to indicate that all data messages have been received and to initiate the validation process. The documentation for the individual message should note whether headers and trailers are supported. |
| **Create Delay Records** | This check box appears only if the message name ends with *FULLSYNC_EFF* (such as MESSAGENAME_FULLSYNC_EFF). Select this check box to write all future-dated rows to the delay table. Also select this check box in conjunction with record mapping views that publish only the current effective-dated rows. |

### Output Format

These options indicate the output of the Publish utility. Valid output formats are:

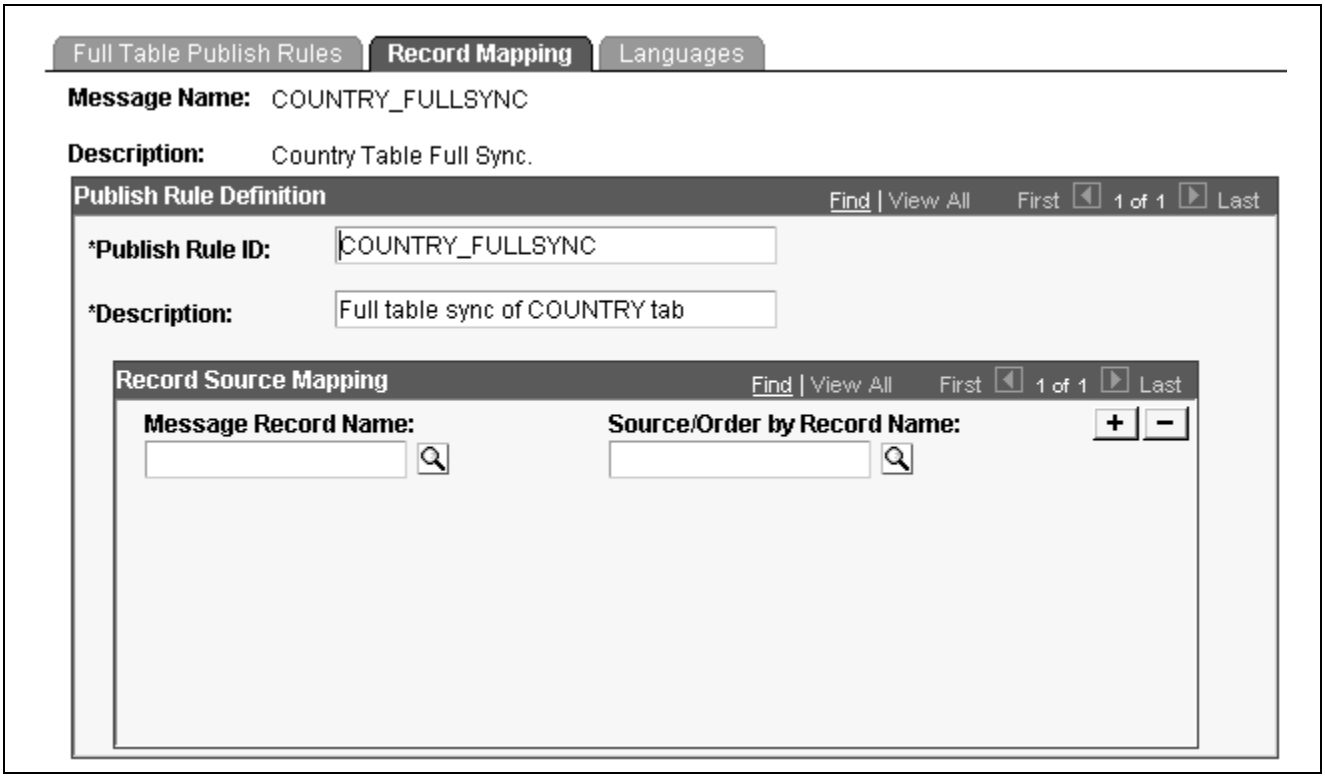
- Message (default value).
- Flat file.
- Flat file with control record.

Select the flat file option when you set up publish rule definitions for the message. If there is a file layout object with the same name as the message object, you can modify the output format field on the Publish Rule Definition page. You can create multiple publish rules with different output formats for the same message. If you select flat file output, the header and trailer messages aren't created, and a single output flat file is created even if you already specified the chunking rule. The file layout definition must have a record structure identical to that of the message; if the AUDIT_ACTN field does not exist in the record, you must add it to the file layout record definition.

The option becomes modifiable if there is a file layout object with the same name as the message definition. The directory location of the flat file is determined by the value in the OUTPUT parameter in PeopleSoft Configuration Manager. The flat file name is *messageName_SequenceNumber*.out, where *messageName* is the message name and *SequenceNumber* is the next available sequential number for files with that message name.

## Mapping a Message Record to Another Record

Access the Record Mapping page.




Record Mapping page

Specify the source data for a record in a message from:

- A staging table.
- A temporary table.
- A view.

Regardless of which source table that you use, ensure that the source table field names are identical to the field names in the target message record. Key fields must also adhere to the parent and child relationship. (Keys of a parent record must exist in the child record, in the same sequence.) The Publish utility uses the Source/Order By Record Name field to select rows for publishing. You use key matching to find all child rows of a parent.

The chunk field should be the primary key field in all Source/Order By records. If the chunk field is not a key field of the level zero record, join the chunk field to all records in which the chunk field does not exist by using views in PeopleSoft Application Designer.

Three reasons to use the record mapping feature in a publish rule are:

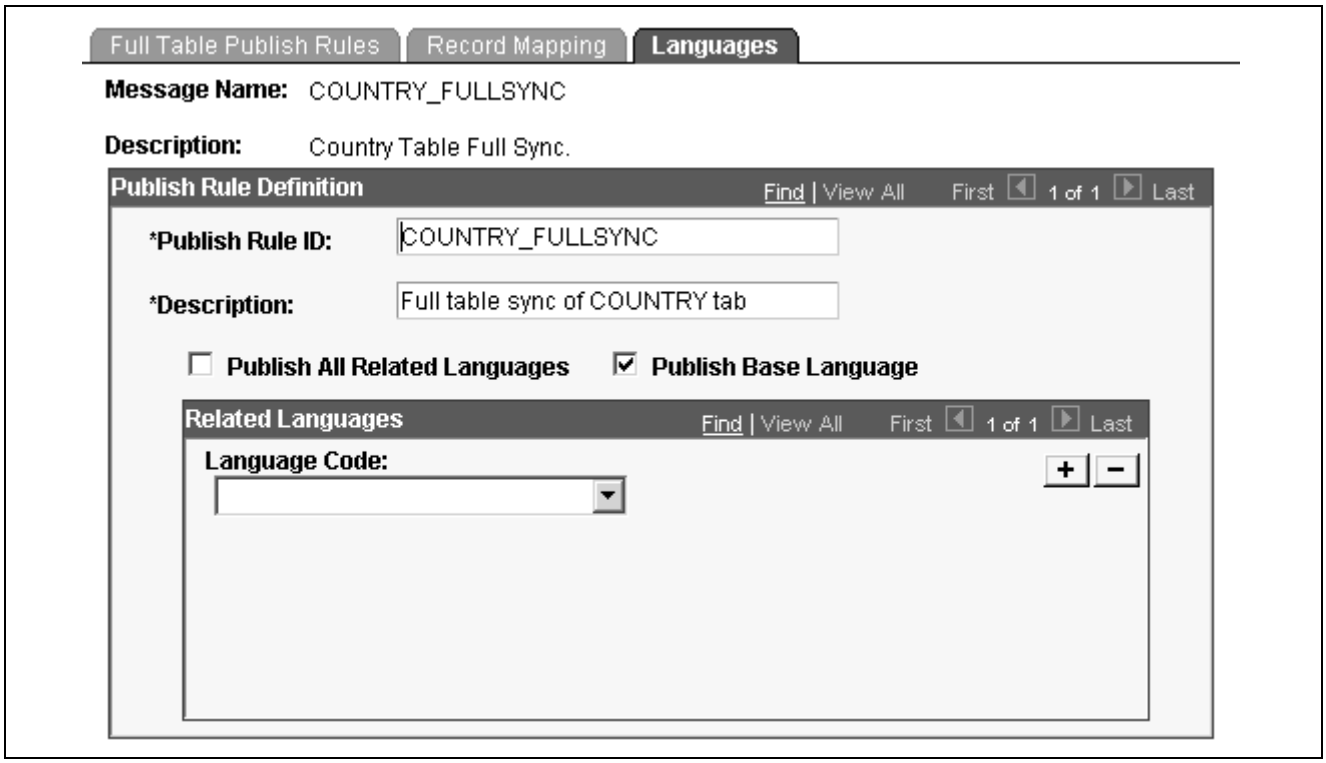
- The published data comes from staging tables or temporary tables.

  Using staging tables is the only way to publish rows that have been deleted from application master tables.
- The published data needs an order sequence that differs from that of the records in the message.
- You must create a view that selects only current effective-dated rows in situations where the application table is effective-dated and the subscribing database cannot process future-dated transactions.

Enter only those message records with a different source or ordering record. If the message record name and the source or ordering record name are identical, do not insert a row for that record on the Record Mapping page.

# Specifying Languages in Which to Publish Messages

Access the Languages page.




Languages page

**Publish All Related Languages** Select to indicate whether to publish the message in all of the related languages. If selected, the scroll area that you use to enter individual related languages is unavailable. This check box is clear by default.

**Publish Base Language** Select to publish the message in the base language. This check box is selected by default.

**Language Code** Select the related language in which to publish the message.

The Publish utility creates the following related language messages:

- One message in the base language of the publishing system.
- One message for each language in the related language tables for the base tables.

The subscribing system receives the messages in the order in which they are published. For example, if the base language is English, with French and then German as related language tables, the Publish utility creates the messages in this order:

1. (Optional) Header message.
2. English message 1.
3. French message 1.
4. German message 1.

5. English message 2.
6. French message 2.
7. German message 2.
8. (Optional) Trailer message.

# Assigning Batch Publishing Rules

All applications can use common, centralized tables and pages to define how to publish incremental messages from an application program. The Publish utility uses batch publish rules to process the data from the application program.

This section discusses how to:

- Associate a rule to a message and characterize the rule.
- Map a message record to another record.
- Assign an application program to a publishing rule.

**Note.** You can link application programs to multiple publishing rules for the same message or different messages. The Publish utility treats each publishing rule as a separate publishing cycle.

## Pages Used to Assign Batch Publishing Rules

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Batch Publish Rules | EO_MSGPUBATCH | Enterprise Components, Integration Definitions, Batch Publish Rules | Associate a rule to a message and characterize the rule. |
| Record Mapping | EO_MSGRECMAP | Enterprise Components, Integration Definitions, Batch Publish Rules, Record Mapping | Map a message record to another record. |
| Batch Programs | EO_MSGBATPGM | Enterprise Components, Integration Definitions, Batch Publish Rules, Batch Programs | Assign an application program (PROCESS_NAME) to the publish rule. |

## Associating a Rule to a Message

Access the Batch Publish Rules page.


Batch Publish Rules page

**Publish Rule ID** Enter the rule ID to associate with the publish rule that you define.

**Status** Select *Active*if the publish definition is valid.

**Chunking Rule ID** Associate a chunking method with the publish rule, if needed.

## Mapping a Message Record to Another Record

Access the Record Mapping page.

**Message Record Name** Enter the name of the record in the message that you want to map to another record.

**Source/Order By Record Name** Enter the record name that the Publish utility uses to select data.

This page specifies the source data for a record in a message. It works in the same manner and accomplishes the same purpose as the Record Mapping page for a full table publish.

## Assigning an Application Program to a Publishing Rule

Access the Batch Programs page.

**Process Name** Enter the name of the COBOL, Structured Query Report (SQR), or Application Engine program that is marking the records to be published.

The Publish utility initially receives the process name from the batch parameter record that is created by the application program. The program then

retrieves and processes each publish rule for the application process name. The process name can be any 12-character string, as long as it matches what the application program inserts into the batch parameter record.

If you select a flat file format, the Publish utility does not create a header or trailer message, and the utility ignores any chunking rules. Instead, the utility creates a single flat file.

# Setting Up Message Chunking

This section provides an overview of message chunking and discusses how to:

- Identify when to use chunking.
- Select chunking fields.
- Create chunking rules.
- Define the chunking rule description.
- Maintain chunking data for business units.
- Maintain chunking data for setIDs.
- Add nodes to existing chunking rules.
- Assign business units or setIDs to a chunking rule.
- Assign chunking rules to a business unit.
- Assign chunking rules to a setID.
- Create custom chunking tables.

## Pages Used to Set Up Message Chunking

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Chunking Rule | EO_CHUNKRULE | Enterprise Components, Map Chunking Rules, Define Chunking Rule | Define a chunking rule description. |
| BusUnit Mapping (business unit mapping) | EO_CHUNKBU | Enterprise Components, Map Chunking Rules, Business Units | Maintain chunking data for business units. |
| SetId Mapping | EO_CHUNKSETID | Enterprise Components, Map Chunking Rules, SetId | Maintain chunking data for setIDs. |
| Add Nodes to Chunk Rule | EO_ADNODECHUNK_PNL | Enterprise Components, Map Chunking Rules, Node to ChunkRule | Add nodes to existing chunking rules. |
| Quick Map | EO_ADDBUNODE_PNL | Enterprise Components, Map Chunking Rules, Node to ChunkRule<br>Add nodes to the chunking rule. | Assign business units to a chunking rule. |
| Map Business Unit | EO_ADDNODEBU_PN | Enterprise Components, Map Chunking Rules, ChunkRule/Node to Bu | Assign chunking rules to a business unit. |
| Map Set IDs | EO_ADDNODESID_PNL | Enterprise Components, Map Chunking Rules, ChunkRule/Node to Setid | Assign chunking rules to a setID. |

## Understanding Message Chunking

If you publish to multiple nodes, you might want the messages to be routed based on a specific field. Chunking rules direct the message. You can use the same chunking fields for breaking a large message into smaller messages, as well as for associating messages with a node (based on those same fields). You set up this kind of relationship between nodes and the fields used to break the message apart (break fields) by using chunking message pages.

You can, depending on some XML content-based logic, use message chunking to route and deliver groups of transactions to different third-party nodes.

For example, consider purchase orders. If you run the full batch publish, each third-party node receives an XML message containing all purchase orders that are dispatched, regardless of whether any purchase orders are intended for that particular customer. With chunking, however, you can set up a chunking rule to chunk a message by customer ID. This creates an XML message for each customer that contains only purchase orders intended for that particular customer.

---

**Note.** Do not confuse message chunking with channel partitioning. You use channel partitioning to partition a channel by a level zero key field. If a field exists on level zero of the record in the message by which you can uniquely distinguish and group transactions to be processed in parallel, partitioning the message by this field increases performance. Without partitioning, a PeopleSoft subscribing system must process each incoming message in the order in which the message is received according to the Publication ID (PUBID) field.

---

## Identifying When to Use Chunking

Use chunking when:

- The message data is large, and the subscriber is consistently interested only in part of the data.
- Subscribers can more efficiently process the message data by chunking messages.
- You have trading-partner-specific content and legally do not want data to be shared among vendors.

You can chunk messages by:

- Locations and inventory shipments by business unit.
- Customers by setID.
- Employees by department or company.
- Sales order acknowledgements by setID and customer ID.
- Purchase orders and purchase order changes by setID and vendor ID.

## Selecting Chunking Fields

To maximize performance and prevent application developers from maintaining complicated views of the data, create staging or temporary tables that contain the chunking fields as the highest order key fields.

Chunking fields can affect performance and alter the options that are available to the Publish utility. The Publish utility creates SQL for each table that is defined in the message object. Tables that are defined in the message can be mapped to an alternative source table or viewed on the Record Mapping page under Publish Rule Definition page.

The source table (or view) serves two purposes:

- It enables the data that must be published to come from a source other than the table that is defined in the message.
- It enables the data to be ordered so that the Publish utility can process rows in the correct sequence.

Because the SQL is run only once and includes a subquery against the values in the chunking table, you must define all chunking fields in every table that is used to retrieve data for the message. The SQL order-by clause is set according to the key fields that are defined in the table. The result is that chunking fields must be defined as key fields for the Publish utility to work.

### Example of Generated SQL for Chunking

This is an example of SQL that is generated for chunking:

```
Select * from PS_INV_ITEMS A
where EXISTS (Select 'Y' from PS_EO_SETID_EOC B
where B.CHUNK_RULE_ID = 'SETID'
and B.EFFDT = '20000201'
and A.SETID = B.SETID)
 order by A.SETID, INV_ITEM_ID, EFFDT
```

The field that you select to chunk on determines the view table that you must create:

| Chunking Field Attribute | Corresponding View Table |
|---|---|
| Chunking field is a key field in level zero table. | By rule, the chunking fields are also key fields in the child tables. The key fields of a parent table must be key fields in the child table and in the same order. If the chunking fields are not the highest order key fields, create a view that consists of all fields in the source table, with the chunking fields as the highest order key fields, followed by the remaining key fields from the source table. Then map this view table to the source table on the Record Mapping page under Publish Rule Definition page. |
| Chunking field is not a key field. | Create a view of the source data that consists of all fields in the source table, with the chunking fields as the highest order key fields, followed by the rest of the key fields from the source table. Then map this view table to the source table on the Record Mapping page under Publish Rule Definition page. |
| Chunking field is not in a source table. | Create a view table that joins the source table to an existing table that contains the chunking fields. This view must consist of all fields in the source table and the chunking fields from the joined table. The chunking fields are the highest order key fields, followed by the rest of the key fields from the source table. Then map the view table to the source table on the Record Mapping page under Publish Rule Definition page. |

**Note.** When running a batch publish rule, the Publish utility runs cleanup logic, which either updates fields or deletes rows in the source tables. If the source table is a view that contains a join, then the option to delete published rows fails.

## Example

The following sample SQL code creates a view table that joins the PS_OMEC_CP_OPT_DET source table to an existing PS_OMEC_HDR_OUT table that contains the chunking fields. The B.SETID_CUSTOMER and the B.CUST_ID chunking fields are the highest order key fields from the joined table (PS_OMEC_HDR_OUT), followed by the rest of the key fields from the source table (PS_OMEC_CP_OPT_DET).

```
SELECT B.SETID_CUSTOMER
, B.CUST_ID
, A.BUSINESS_UNIT
, A.ORDER_NO
, A.ORDER_INT_LINE_NO
, A.CP_MODE
, A.CP_COMP_SEQ
, A.OPTION_NAME
, A.OPTION_VALUE
, A.OPTION_DESC
, A.VAR_TYPE
```

```
, A.VAR_LENGTH
, A.VAR_DECIMAL
, A.PROCESS_INSTANCE
, A.AUDIT_ACTN
, A.IN_PROCESS_FLG
FROM PS_OMEC_CP_OPT_DET A
, PS_OMEC_HDR_OUT B
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
AND A.ORDER_NO = B.ORDER_NO
```

## Creating Chunking Rules

The chunking rule consists of four tables:

| Level | Table | Description |
|---|---|---|
| Level 0 | EO_CHUNCKRULE | A system table delivered with live data. |
| Level 1 | EO_CHUNKEFFDT | A system table delivered with live data. When a chunking rule is saved, a row is added to this table with the effective date (EFFDT) field automatically populated from the current date and the effective status set to*Active*. |
| Level 2 | EO_CHUNKNODE | This is not a system table and is delivered empty. |
| Level 3 | *NAME_EOC* | A user-defined chunking table. |

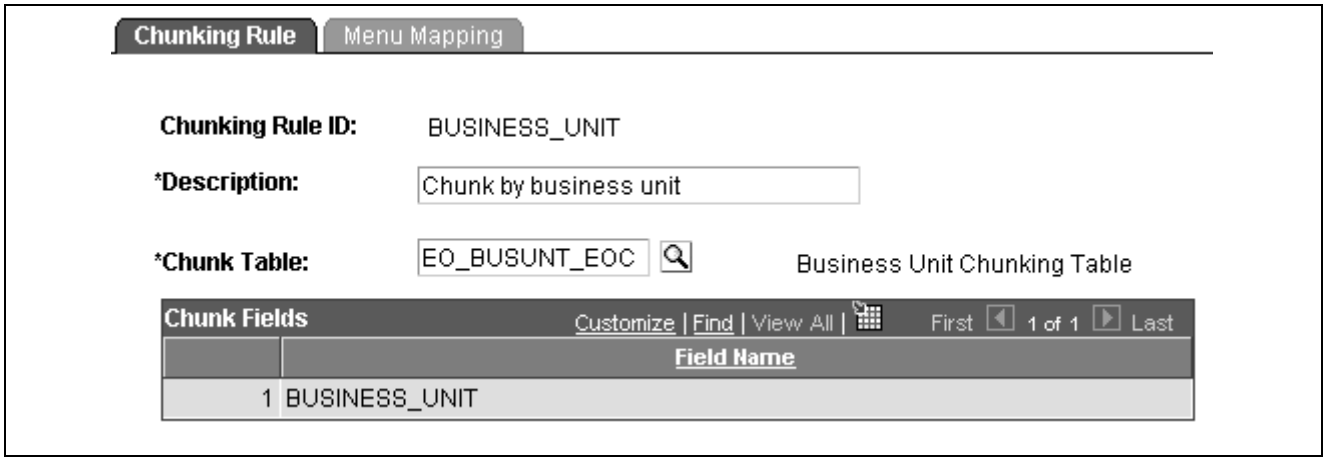
All user-defined chunking table names must end in *_EOC*. PeopleSoft provides two standard tables: EO_BUSUNIT_EOC for business unit values and EO_SETID_EOC for setID values. The different types of user-defined chunking tables are:

| Table Type | Description |
|---|---|
| Derived Table | Contains only the chunking fields. Can be used by the Publish utility to chunk the message whenever the value of the chunking field changes. In derived tables, there is no relationship between the value of the chunking fields and message node names that are used to route the message. OnRoute PeopleCode needs hard-coded routing logic or additional tables to route the message to the appropriate nodes. |

| Table Type | Description |
|---|---|
| SQL Tables | Contains the fields:<br>• CHUNK_RULE_ID<br>• EFFDT<br>• MSGNODENAME<br>• Chunking fields<br>Limits the published data to the values of the chunking fields in the chunking table and contains the message node name that is used to route the message. |
| Alternate Chunking Tables | Enables reuse of existing chunking tables. Must end in *_EOV*. |

## Defining the Chunking Rule Description

Access the Chunking Rule page.



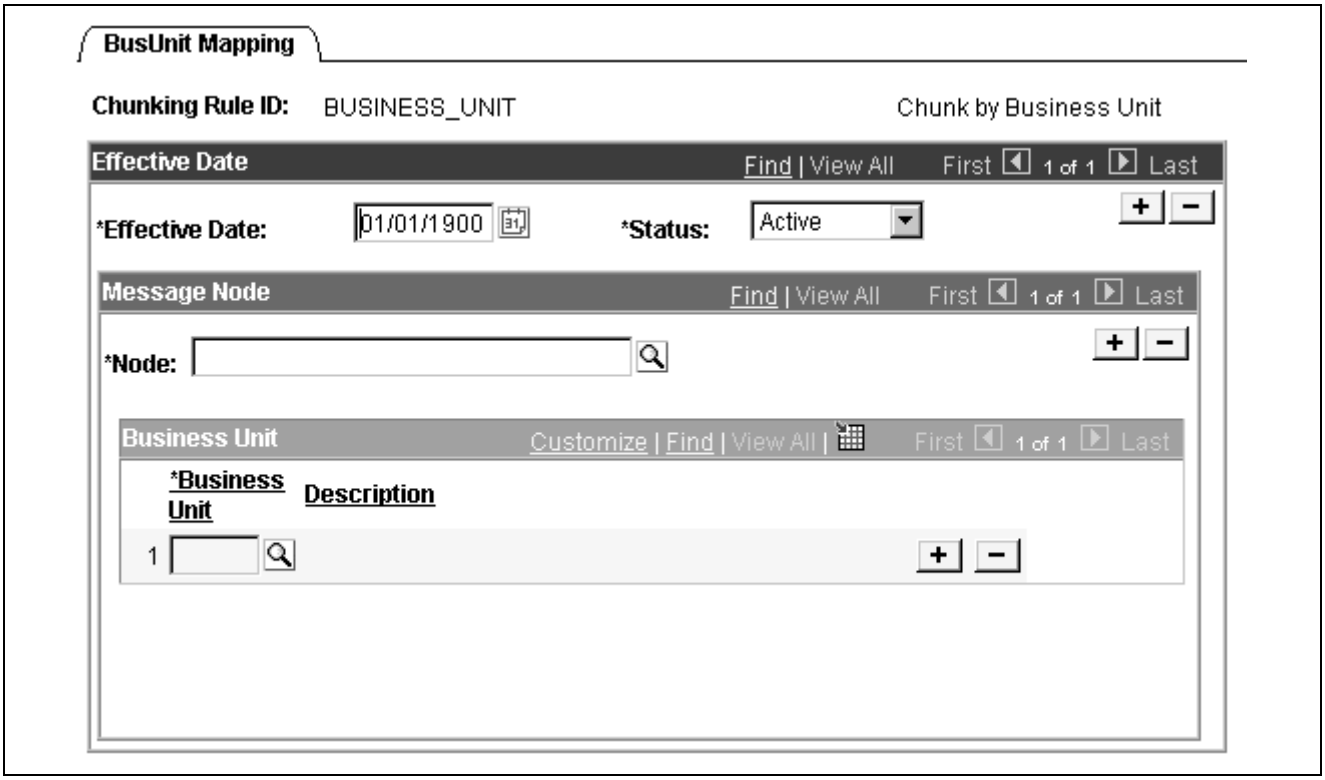

Chunking Rule page

Chunk field are the chunking fields that are defined in the Chunk Tablefield.

The chunking fields appear in this scroll area to verify that the correct chunking table was entered. PeopleSoft provides chunking rules for business unit and setID for all application databases. Adding a new chunking rule inserts a row into the EO_CHUNKEFFDT table, with a default effective date (EFFDT) of the current date.

## Maintaining Chunking Data for Business Units

Access the BusUnit Mapping page.




BusUnit Mapping page

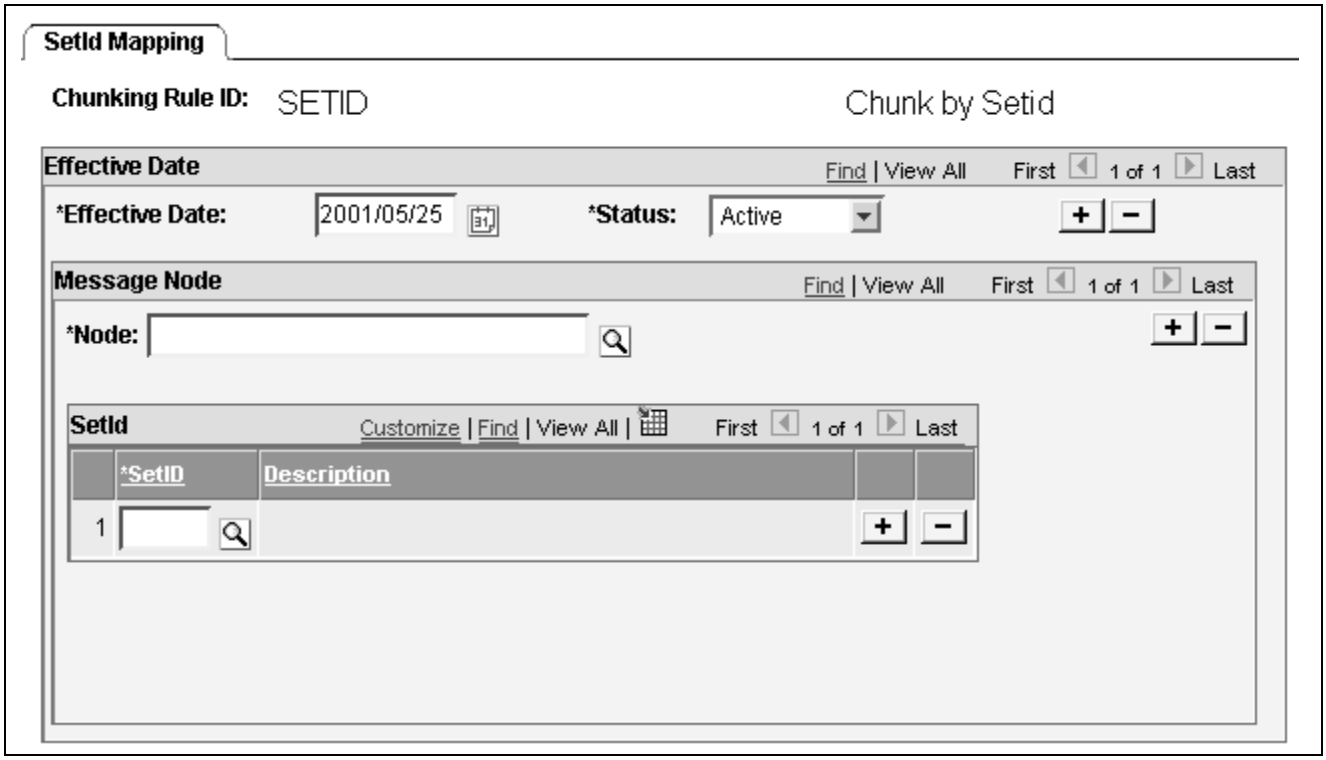
All four levels of the chunking rule tables appear.

PeopleSoft provides chunking tables for business unit and setID that are maintained by a series of components (such as components that are created for maintaining the business unit chunking table).

You can use each component to update the underlying relationship between the business unit and the subscribing nodes. You can maintain the data either by business unit or by node, individually or as a group, to reduce the amount of entry work.

## Maintaining Chunking Data for SetIDs

Access the SetId Mapping page.




SetId Mapping page

All four levels of the chunking rule tables appear.

PeopleSoft provides chunking tables for business unit and setID that are maintained by a series of components (such as components that are created for maintaining the business unit chunking table).

You can use each component to update the underlying relationship between the business unit and the subscribing nodes. You can maintain the data either by business unit or by node, individually or in a group, to reduce the amount of entry work.

## Adding Nodes to Existing Chunking Rules

Access the Add Nodes to Chunk Rule page.

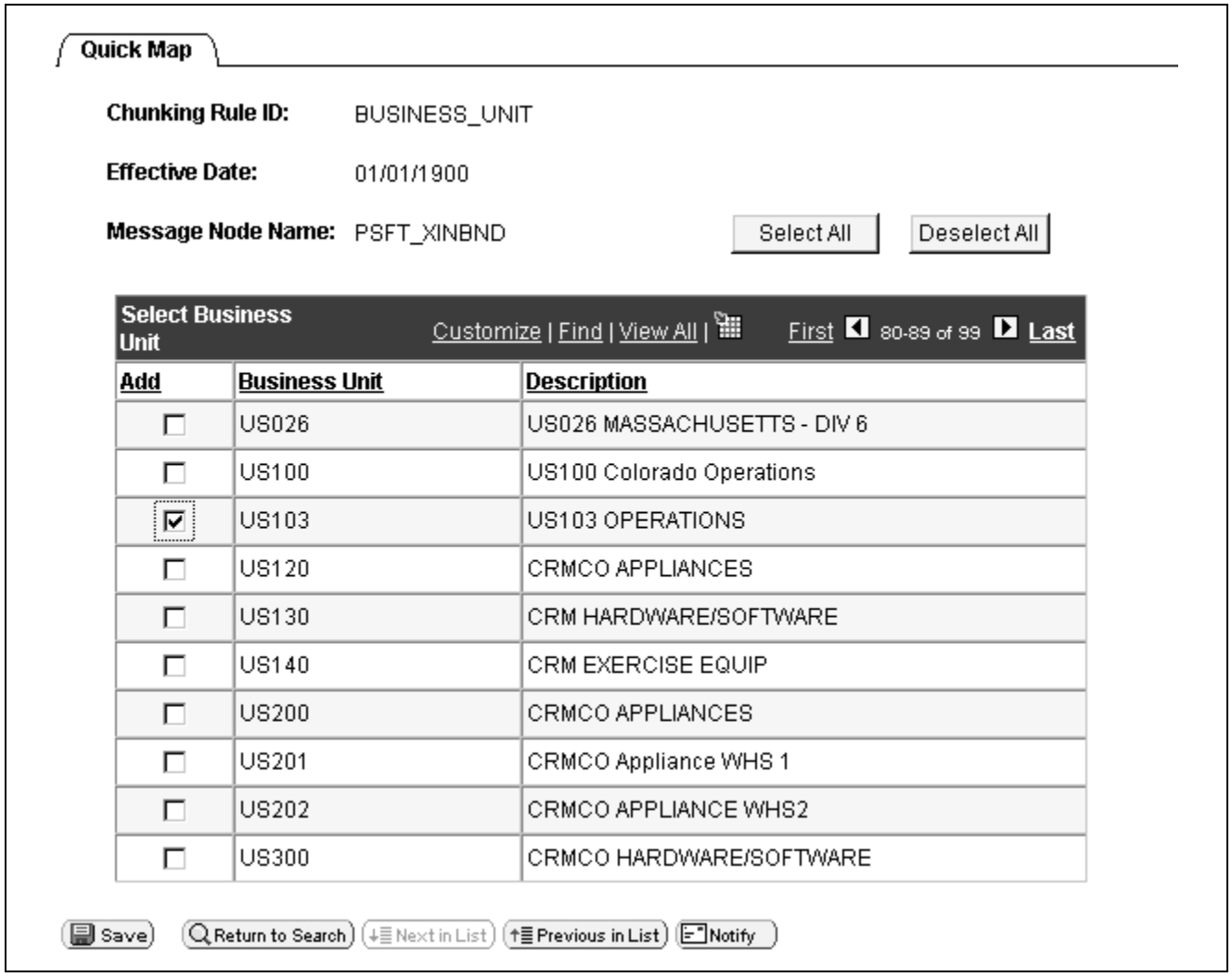
To add nodes to an existing chunking rule:

1. Select the check box in the Add column of the nodes that you defined earlier.
2. Click the Save button to display the Add Chunk Values column.
3. Click the Add button in the Add Chunk Values column for the nodes that you want to add.

   The Quick Map page appears.

## Assigning Business Units or SetIDs to a Chunking Rule

Access the Quick Map page.




Quick Map page

If you previously accessed business unit chunking rules, you can add business units to a chunking rule. If you previously accessed setIDs, you can add setIDs to a chunking rule.

---

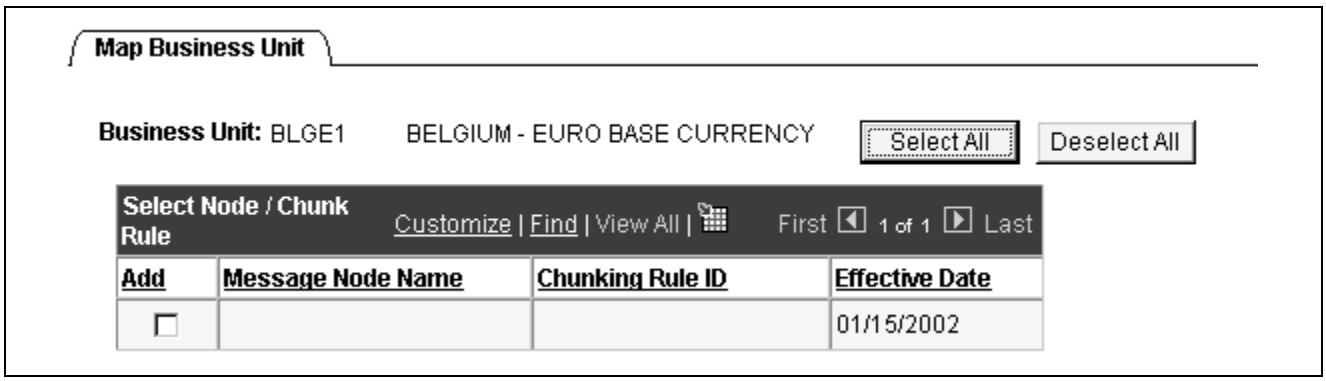
**Note.** You cannot access the Quick Map page without first using either the BusUnit Mapping page or the SetId Mapping page to add an effective-dated node to the chunking rule ID.

---

**Select All** and **Deselect All** Click to add or remove all business units that are assigned to the node. Add check boxes are selected for business units that are assigned to the node.

## Assigning Chunking Rules to a Business Unit

Access the Map Business Unit page.

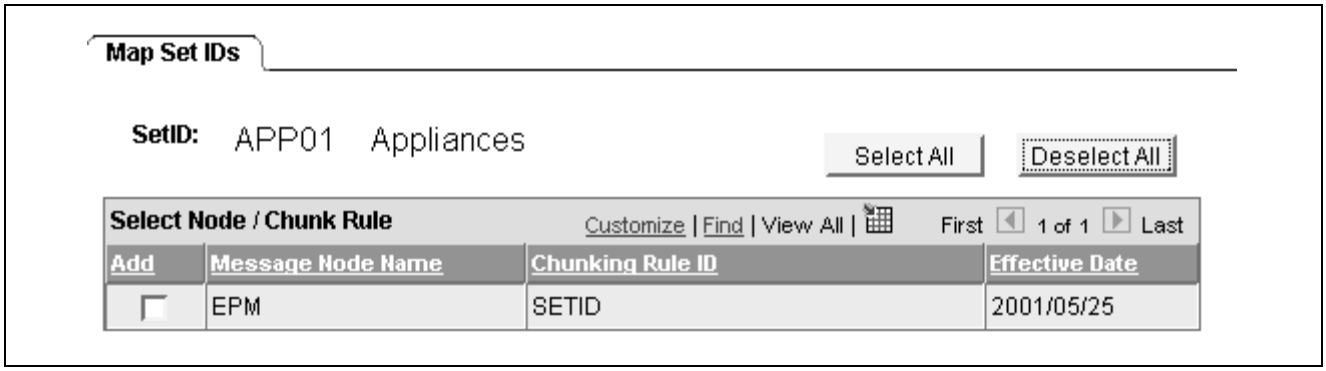



Map Business Unit page

**Select All** and **Deselect All** Click to add or remove all message nodes that are assigned to the business unit. Add check boxes are selected for message nodes that are assigned to the business units.

## Assigning Chunking Rules to a SetID

Access the Map Set IDs page.




Map Set IDs page

**Select All** and **Deselect All** Click to add or remove all message nodes that are assigned to the setID. Add check boxes are selected for message nodes that are assigned to the setIDs.

# Creating Custom Chunking Tables

This section discusses how to:

- Create a custom chunking table.
- Create a view for the component search record.
- Create maintenance pages.
- Create a component.
- Create routing PeopleCode.

## Creating a Custom Chunking Table

To create a custom chunking table:

1. Select File, Open in PeopleSoft Application Designer.
2. Select *Record* in the Definition drop-down menu.
3. Open the EO_BUSUNT_EOC record.
4. Save the record as *YOUR_TABLE*_EOC.
5. Remove the BUSINESS_UNIT field.
6. Insert the custom chunking fields at the bottom of the record.
7. Select File, Save.
8. Build the SQL table.

## Creating a View for the Component Search Record

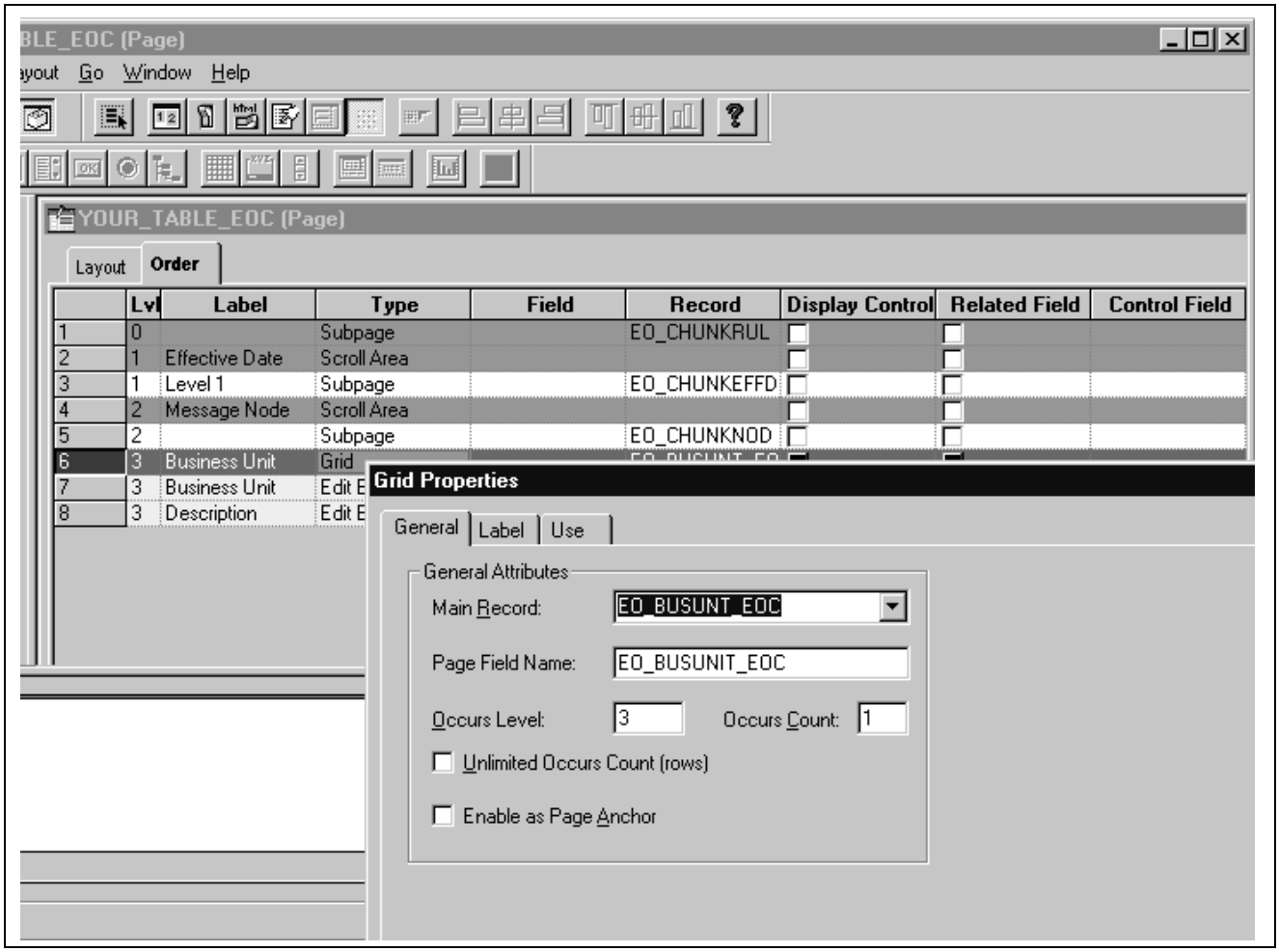
To create a view for the component search record:

1. Select File, Open in PeopleSoft Application Designer.
2. Select *Record* in the Definition drop-down menu.
3. Open the EO_CHUNKBU_VW record.
4. Save the record as *YOUR_TABLE*_VW.
5. Select the Record Type tab.
6. Open the SQL editor.
7. Modify the Where clause.

   Change `WHERE RECNAME_CHUNK=EO_BUSUNT_EOC` to `WHERE RECNAME_CHUNK=YOUR_TABLE_EOC.`
8. Select File, Save.
9. Build the SQL view.

## Creating Maintenance Pages

You can also create maintenance pages.




Grid Properties dialog box

To create maintenance pages:

1. Select File, Open in PeopleSoft Application Designer.
2. Select *Page* in the Definition drop-down menu.
3. Open the EO_CHUNKBU page.
4. Save the page as *YOUR_PAGE*.
5. Select the Order tab.
6. In the Type column, double-click Grid to open the Grid Properties dialog box.
7. Change the value in the Main Record and Page Field Name fields to *YOUR_TABLE_EOC*.
8. Click OK.
9. Delete the business unit and description columns.
10. Add chunking fields from YOUR_TABLE_EOC.
11. Select File, Save.

## Creating a Component

To create a component:

1. Select File, New in PeopleSoft Application Designer.
2. Select *Component* in the New Definition box.
3. Select Insert, Page Into Component.
4. Enter *YOUR_PAGE*.
5. Click Close.
6. Select File, Definition Properties
7. Select the Use tab to edit component properties:
    a. In the Search record field, enter *YOUR_TABLE_VW*.
    b. Select the Update/Display, Update/Display All,and Correction check boxes.
8. Click OK.
9. Select File, Save As, and save the page group as *YOUR_COMPONENT*.
10. Add YOUR_PAGE_GROUP to YOUR_MENU that is used by your application.

## Creating Routing PeopleCode

OnRouteSend (and OnRouteReceive) are PeopleCode events that are tied to the message for routing, based on the message contents. If you want the contents of the message (such as a message chunking field value) to determine which subscribing nodes should receive the message, OnRouteReceive PeopleCode must contain the logic to examine the message and return a list of subscribing nodes.

PeopleCode functions provided by common components, GetNodes and RtnNodes, work with any message and chunking rule. For a given message, these nodes select the chunking rule for the publish rule that is assigned to the message.

The functions do the following:

- Build SQL based on the chunking fields as defined in the chunking table.
- Extract chunking field values from the message.
- Run the associated SQL.
- Compare the array of nodes returned to the application server against the list of nodes for the message channel.
- Create a publish contract for nodes in both arrays.

  You can override the publish rule from the message, specifying an optional parameter when calling the functions.
- Return an array of nodes that is based on the nodes that are assigned to the message channel if the publish rule is invalid or does not contain a chunking rule.

  Returning an array of nodes enables the functions to work regardless of whether chunking is set up for the publish rule.

To route any message that uses chunking, use generic PeopleCode functions.

These functions are called from routing PeopleCode:

| | |
|---|---|
| **GetNodes** | Returns an array of nodes to the application server. |
| **RtnNodes** | Returns an array of nodes to the calling PeopleCode. |

These functions are internal functions:

| | |
|---|---|
| **FndNodes** | Builds an array of nodes for the message. |
| **FndChannelNodes** | Builds an array of nodes for the channel. Used when there is no chunking rule for a publish rule or when the publish rule is not found. |
| **GetPubRule** | Selects the chunking rule for the publish rule. |
| **GetChunkInfo** | Selects the chunk table for the chunking rule. |
| **BuildSQL** | Builds SQL to select nodes from the chunking table for specific chunking field values from the message. |
| **GetValue** | Gets the chunking field values from the message. |

This is an additional function:

| | |
|---|---|
| **HasNodes** | Determines whether a chunking field is mapped to any nodes for a particular chunking rule. |

The following code example shows the logic that you can add to SavePostChange PeopleCode for the Customer_General component to verify if the setID can publish the message by calling the HasNodes() function:

```
Declare Function HasNodes PeopleCode FUNCLIB_EOEIP.PUBLISH_ROUTE_PC FieldFormula;
Local Message &MSG;
Local Rowset &RS0;
Local string &PublishRule;
&MSG = CreateMessage(MESSAGE.CUSTOMER_MSG);
/* Check if message is active */
If &MSG.IsActive Then
   &RS0 = GetLevel0();
   &PublishRule = "CUSTOMER_SYNC";
   /* Call function passing publish rule and rowset, which returns true if this⇒
 setID can publish the message */
   If (HasNodes(&PublishRule, &RSO)) Then
      &RS0 = GetLevel0();
      &MSG.CopyRowsetDelta(&RS0);
      &MSG.Publish();
   End-If;
End-If;
```

The following code example shows the PeopleCode that you can add to the OnRouteSend event for the message channel that is associated with any message where chunking is assigned to a publish rule:

```
Declare Function GETNODES PeopleCode FUNCLIB_EOEIP.PUBLISH_ROUTE_PC FieldFormula;
Local string &PublishRule;
/* Set PublishRule if you want to override the value in the message PSCAMA.  */
```

```
/* &PublishRule = "CUSTOMER_SYNC";                                          */
/* Call function that looks at setID of first transaction in the message and⇒
 returns a list of subscribing nodes to route the message */
GETNODES(&PublishRule);
```

# CHAPTER 16

# Using the Error Handling Utility

This chapter provides an overview of the Error Handling utility and discusses how to:

- Set up and maintain message errors.
- Correct message errors.

**Note.** If you use the Error Handling utility, you should be familiar with messaging technology.

## Understanding the Error Handling Utility

The Error Handling utility is a PeopleTools application that you use to view and correct messages that are received by the subscriber. You can also use this utility to correct data that is stored in staging tables.

### Error Management Process

PeopleSoft applications that receive flat file data from other systems through batch processes have built-in facilities to validate and correct data prior to updating the main application tables. Likewise, before updating core PeopleSoft application tables, the subscription process detects data errors in the messages that it receives. These error messages are stored in either message queues or staging tables.

In some cases, however, errors can't be sent back to the third party for correction (such as when data is in a flat file). In these cases, you must provide error processing on incoming data so that messages that contain information on business objects, such as items and vendors, can be corrected and reprocessed in the PeopleSoft system.

In many enterprise integration point (EIP) implementations, especially those that involve huge transactions and complex data validations, subscription codes are written to run simple incoming data validations. Upon a successful outcome, the system performs these steps:

1. Writes incoming data to staging tables.
2. Runs more stringent data validation processes, usually in batch processes.

As a result of the first subscription process, entries are written to a special staging table (EO_EIP_CTL). The EO_EIP_CTL table is keyed by a single key, EIP_CTL_ID, and has messaging keys. The EO_EIP_CTL table maintains links between the source message and the staged data and ensures that the data maintenance program can identify the staging tables.

To find the particular message that you want to view:

1. Select a message by using standard selection criteria, including data maintenance programs, business unit or setID, data status, and staging table or message queue.
2. Click a detail button to access the application page related to the edited message.

3. Make your edits.
4. Save the message.

---

**Note.** Messages that are selected from a staging table are saved to the staging table. Before they're saved permanently (to the application table), the corrected messages that were selected from the message queue are assigned a status of *Changed* and saved to the same queue. The messages then undergo the subscription process data validation again, after which they're saved to an application table.

---

A data maintenance program corrects errors in data that enters the PeopleSoft system.

---

**Note.** Subscription error management does not apply to real-time data that enters the PeopleSoft system through PeopleSoft Business Interlinks.

---

Error management consists of the following sequential activities:

1. The PeopleSoft system receives XML messages from a third-party source.
2. Subscription processes validate the data and send the data to:
   - PeopleSoft application tables (if no errors were found).
   - Message queue (if errors were found).
   - PeopleSoft staging tables (if subscription logic requires).
3. The Error Handling utility interacts with:
   - Message queue.
   - Staging tables.
4. The user interacts with:
   - The Error Handling page (to select a message for review or correction).
   - Application pages (for editing data errors).
5. The PeopleSoft system updates:
   - Message queue (if the original message was selected from the queue).
   - Staging tables (if the original data was extracted for staging tables).

# Setting Up and Maintaining Message Errors

This section discusses how to:

- Create error-correction pages and the necessary hooks to the error handling utility.
- Set up the error handling utility.
- Set up row security.
- Set up workflow notification in PeopleSoft Application Designer.
- Test the error handling utility.

## Pages Used to Set Up and Maintain Message Errors

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Data Maint Setup1 (data maintenance setup 1) | EO_EIP_CTL_SETUP1 | Enterprise Utilities, Integration Definitions, Data Maintenance Utility | Set up the Error Handling utility and correct message errors. |
| Data Maint Setup2 (data maintenance setup2) | EO_EIP_CTL_SETUP2 | Enterprise Utilities, Integration Definitions, Data Maint Utility Setup2 | Set up row security. |
| EO EIP Data Maint (EO EIP data maintenance) | EIP_DTA_CTL | Enterprise Utilities, Integration Definitions, Review Centralized Error | Test the error handling utility.<br>Test subscription processes must error out. Create a message subcontract with *Error* status, or create a stage-based incoming transaction with a status value of 1 *(Error)*.<br>**Note.** Error conditions depend on processes that are specific to the application. |

## Creating Error-Correction Pages

Use PeopleSoft Application Designer to create error-correction pages and components.

**Note.** The Header Details page and the Error Details page are applicable only if data resides in the messaging queues. Use one Header Details page and one Error Details page for each field.

When creating error-correction pages, follow these guidelines:

- Use EO_EIP_CTL as the main header table if you use staging tables.

  Developers can use unique EIP_CTL_ID generator functions (Generate_EIP_CTL_ID and Increment_EIP_CTL_ID) available in the FUNCLIB_EOEIP.EIP_CTL_ID FieldFormula event.

- Use the generate EIP_CTL_ID function to generate a unique key value for EIP_CTL_ ID, based on the subscription process instance.

  The EIP_CTL_ID function is the sole key in the EO_EIP_CTL header table.

- Hook to EO_EIP_CTL to use the Error Handling utility for staging-table-based error handling.

  See Chapter 16, “Using the Error Handling Utility,” Setting Up the Error Handling Utility, page 200.

- Modify the record PeopleCode in the derived work record EO_EIP_CTL_WRK.

  The derived work record called EO_EIP_CTL_WRK contains most of the processing logic for the utility. Although most of the codes are generic, you must write application-specific codes for the return button field (EIP_RETURN_BTN).

- Add codes to return the user to the main Error Handling page.

Use the EO_EIP_CTL_WRK derived record in your component and write component PeopleCode for the record to handle any unique requirements.

**Note.** This unique code belongs only to your component and cannot be shared.

### FUNCLIB_XXEIP Codes

Functions are created and stored in product-specific FUNCLIB_XXEIP records, with *XX* representing the product. Application developers can look up these function libraries for possible use as templates.

Common integration-related functions are placed in the FUNCLIB_EOEIP record.

### EIP_DETAIL_BTN FieldFormula

Function Copy_Detail_Errors (&WRK_FIELD, &J, &WRK_ROWSET) copies edit errors to a generic error subpage.

Function BuildQueueRowset (&WRK_ROWSET1 As Rowset, &WRK_ROWSET2 As Rowset, &MSG_ROWSET As Rowset, &SCROLL, &RECORD_FROM, &RECORD_TO) generically builds the queue-based transfer page. This function is useful for single record messages.

### FUNCLIB_EOEIP.EIP_CTL_ID FieldFormula

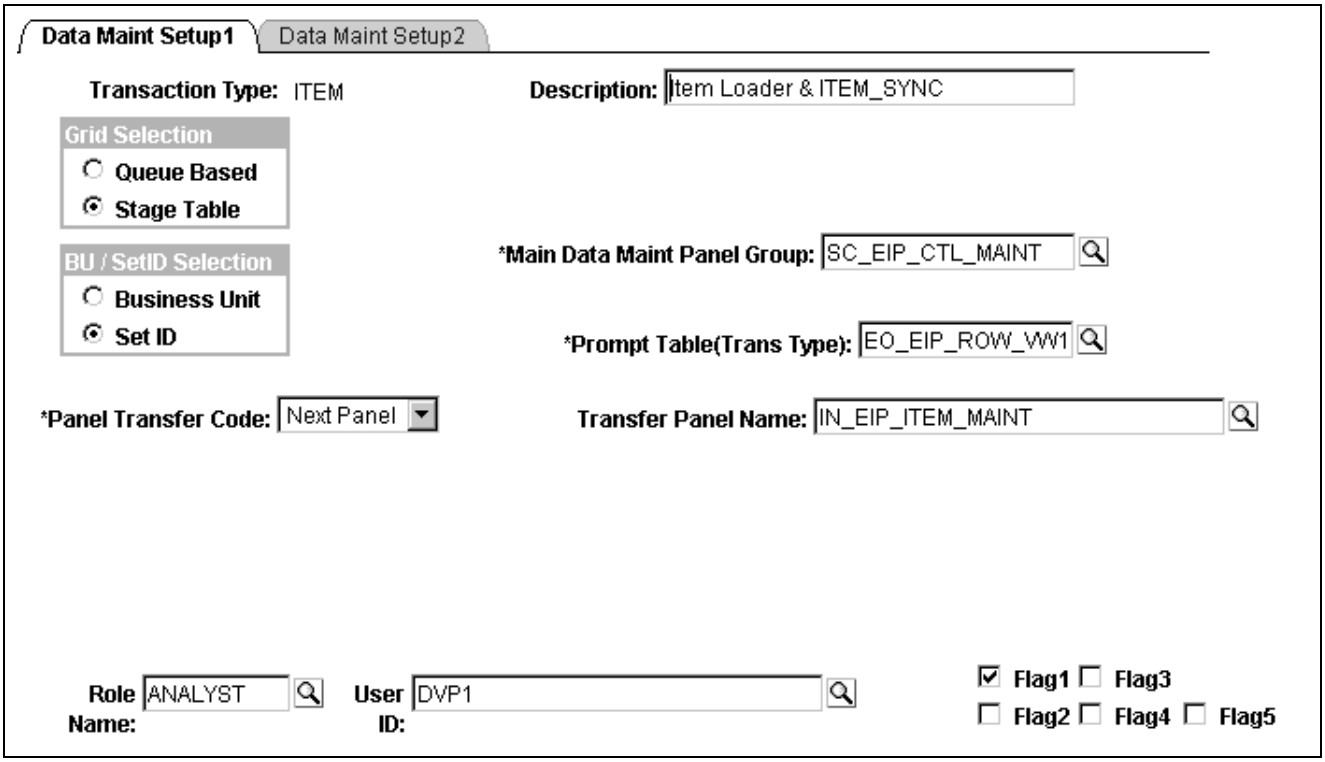
You need an EIP_CTL_ID every time that you process a message to maintain error handling. Use Function Generate_EIP_CTL_ID and Increment_EIP_CTL_ID to create EIP_CTL_IDs based on a new subscription process instance. For the Generate_EIP_CTL_ID function, invoke method 4 from the list of process instances.

**Note.** Although in some cases a random number generator is used for creating a new EIP_CTL_ID by calling Generate_EIP_CTL_ID(1), you should use the subscription process instance if possible. Generate_EIP_CTL_ID(4) uses the message subscription process identifier number. Remember to activate this on the message subscription properties by selecting the Generate Subscription Process Instance check box.

***Warning!*** The Generate_EIP_CTL_ID and Increment_EIP_CTL_ID functions call other functions in FUNCLIB_ININTFC (an FDM record). You must recreate these additional functions whenever you use Generate_EIP_CTL_ID and Increment_EIP_CTL_ID within applications.

## Setting Up the Error Handling Utility

Access the Data Maint Setup1 page.




Data Maint Setup1 page

**Queue Based** and **Stage Table** — PeopleTools messages are queue-based; bulk data or application data is usually staged.

**Business Unit** and **SetID** — Select a business unit or setID on which to base the data.

**Main Data Maint Panel Group** (main data maintenance panel group) — Select the name of the component for accessing the Error Handling/Data Maintenance page (EO_EIP_CTL_MAINT). This is the component that the user opens for error-correction activities. You should select *EIP_DTA_CTL*. This component uses the same fonts and images for all incoming data maintenance or error-correction activities within PeopleSoft for non-real-time integration.

**Prompt Table (Trans Type)** (prompt table [transaction type]) — Follow the navigation path to use the Error Handling utility, and select a prompt table for the data maintenance program (EIP_PROGRAM). The PeopleSoft system includes a common error handling menu structure that points to the main component EIP_DTA_CTL. Based on user privileges, the Error Handling utility prompts from a selected list of data maintenance programs.

**Panel Transfer Code** — Select *Next Panel* to receive a transfer panel name prompt, which gives you a selection of pages that are part of the main data maintenance component that you selected in the Main Data Maint Panel Group field. Select one page.

Selecting either *Transfer* or *Modal* opens additional fields.

**Transfer Panel Name** — Select the default value *EO_EIP_ROW_VW1*. This view enforces row-level security. During the correction phase, when you try to select an EIP_PROGRAM (transaction type), the utility prompts from a list of values for which you have the correct access privilege.

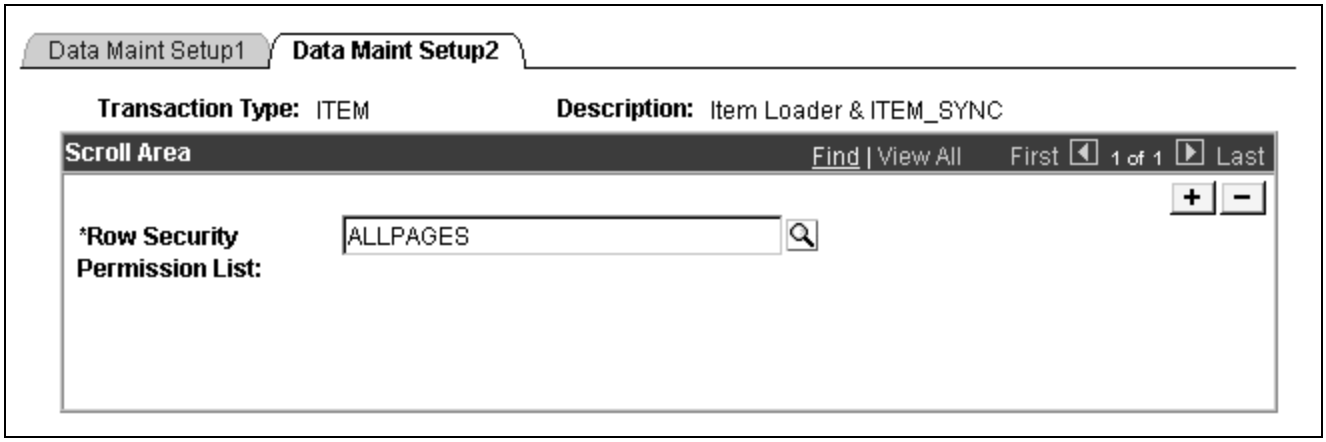
**Role Name** Select a name to set up workflow notification and to notify all role users listed.

**User ID** Select to notify only specific role users.

**Flag1, Flag2, Flag3, Flag4,** and **Flag5** Users often won't see these check boxes. Application developers use them for extra coding flexibility. Developers must create documentation for these check boxes to use them.

## Setting Up Row Security

Access the Data Maint Setup2 page.




Data Maint Setup2 page

**Row Security Permission List** Assign a row security class to the transaction type that you're setting up.

**Note.** The value that you select determines which user can see the given transaction type. You can enter multiple row security classes.

## Setting Up Workflow Notification in PeopleSoft Application Designer

You can set up an EO_WF_ERR Application Engine process to scan the EO_EIP_CTL table periodically to look for rows with *ERROR* status. When the system finds errors, it generates workflow notifications and routes them to the role users that you designated in the User ID field on the Data Maint Setup1 page. Clicking the worklist item in the workflow notification transfers the user to the Error Handling/Data Maint (error handling/data maintenance) page.

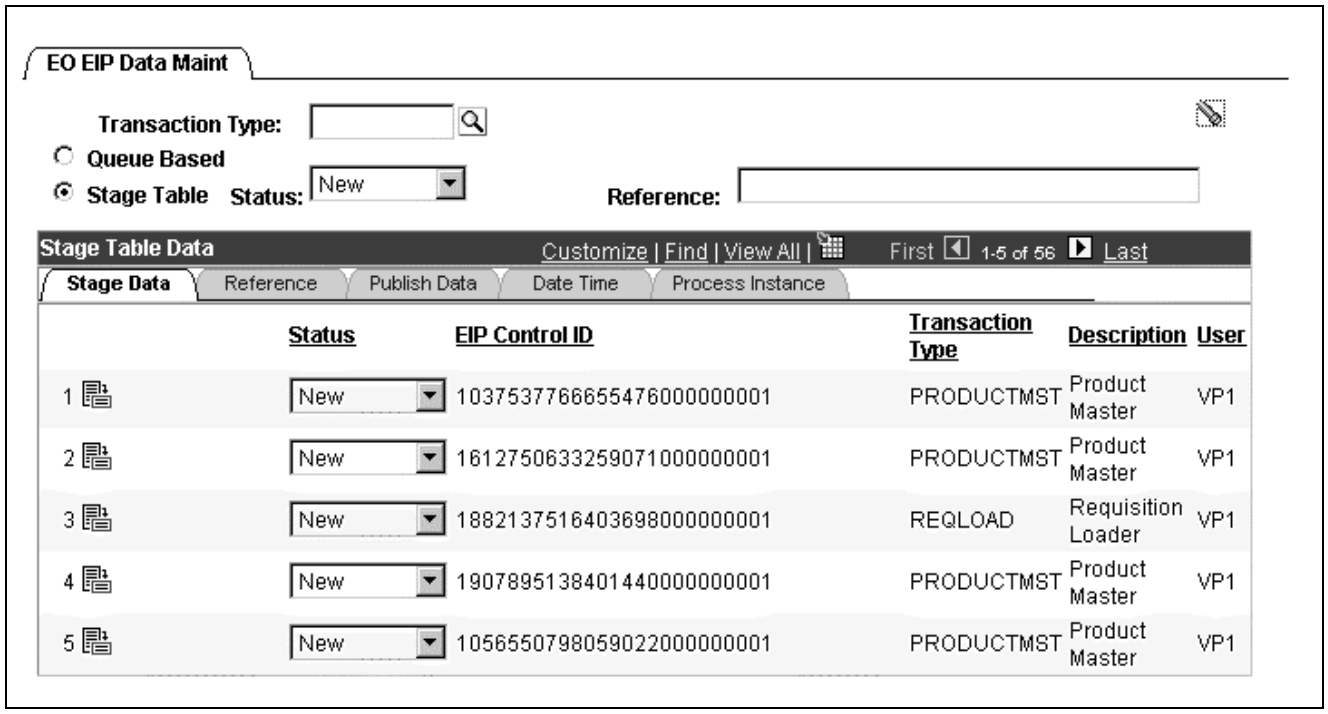
To set up workflow notification:

1. Open an instance of PeopleSoft Application Designer.
2. Select File, Open.
3. Select *Business Process* from the Definition drop-down menu.
4. Open business process EC_MANAGE_ERRORS.
5. Right-click the MANAGE ERRORS icon and select *View Definition.*
6. Double-click the Correct Errors icon to open the Step Definition dialog box.

7. Click the Attributes button to open the Step Attributes dialog box.
8. Complete the required navigation information to the error correction page in the Step Attributes dialog box and click OK.

## Testing the Error Handling Utility

Access the EO EIP Data Maint page.




EO EIP Data Maint page

**Queue Based** and **Stage Table** — Select whether the errors you want to check are queue-based or stored in a staging table.

**Status** — Select from *Cancelled, Complete, Error, Hold, In Process, Incomplete, New,* and *Reprocess.*

 — Click the Show Detail Entry button to transfer to the application page for necessary error-correction activities.

# Correcting Message Errors

This section discusses how to correct message errors.

## Understanding the Workflow Notification Process

Workflow error notification involves:

1. A process-scheduled Application Engine program scans data errors in the staging tables (or in the Subscription Contract Message queues) at a given interval.

   **Note.** You can also run a separate Workflow Notification Application Engine program for messaging queues.

2. When the Application Engine program finds errors in the actual message queue or the staging summary table, it opens respective component interfaces that invoke the designated temporary components for inserting error data into the underlying temp tables.
3. Saving the temporary page (through component interface calls) triggers workflow PeopleCode to send worklist notifications to designated users, depending on the role name or user ID that you selected on the Data Maint Setup1 page.

   **Note.** The Application Engine program traps only the first error for each field, but it can trap multiple errors for each record.

## Correcting Message Errors

To correct message errors:

1. Access the Data Maint Setup1 page to open the worklist.
2. Open the worklist.

   You are transferred to the Error Correction/Data Maint page (EO_EIP_CTL_MAINT) to correct the errors.
3. Find the error rows by using a combination of search criteria, such as transaction type, business unit or setID, and error status.
4. Click the Search icon.
5. Click the Show Detail Entry icon (to the left of each row) to select a specific transaction that represents a single message or a single row of header-level record from the staging table.
6. Correct the data.
7. Click the Return button to return to the Error Handling Summary page (EO_EIP_CTL_MAINT) and continue the session with other transactions.

**Note.** In this scenario, the corrected data does not update core PeopleSoft tables. Instead, it updates the message queue or staging tables.

# CHAPTER 17

# Using the Effective Date Publish Utility

This chapter discusses how to:

- Perform a full data publish of current effective data.
- Publish incremental messages of current effective data.
- Publish effective-dated rows from the delay table.

## Understanding the Effective Date Publish Utility

The Effective Date Publish utility enables you to design processes to update external systems that process only current data and don't use or recognize effective dating.

When working with effective dating and effective date publishing, you need to understand the following terms:

**Current Row** — The current row is the first row of data with an effective date equal to or less than the system date. Only one row can be the current row.

**Future Rows** — Future rows have effective dates greater than the system date (usually the current date).

**Historical Rows** — Historical rows have effective dates less than the current row.

**Effective Date** — An effective date is when a table row becomes effective, or the date that an action begins. The PeopleSoft system supports the concept of effective-dated rows.

**Note.** The EFFDT field is almost always a key. Specify the descending key attribute to display the row with the most recent effective date first.

**Effective Dating** — Automated effective dating saves changed data in a staging table for subsequent processing when the effective date becomes current. (Although data can be historical, current, or future, some third-party applications may support only current data. Thus, if a future-dated row is created within the PeopleSoft system, it must be delayed before transmission to the other system.)

**Effective Sequence** — An effective sequence serves two different purposes:

- If EFFSEQ is a required field, it enables the entry of more than one row with the same effective date when paired with EFFDT. The system

assigns a unique sequence number to each row that has the same effective date. It also enables the first EFFSEQ to be zero.

- If EFFSEQ is not a required field, it is not paired with EFFDT, has no special function, and can be used as a simple sequencing field.

**Effective Status** Effective status enables the system to select the appropriate effective-dated rows, when combined with the effective date field.

**Full Data Publish** The full data publish process seeds, or initially populates or repopulates, a copy of an entire table onto a remote database or legacy system. The entire contents of the table are published to all systems that require a copy of the table. Generally, full data replication occurs with setup tables (relatively static, low-volume tables that are keyed by setID) and occurs in an asynchronous manner.

When a full copy of the table exists on the external system, an incremental update provides a mechanism to keep the copy up-to-date with changes made on the master.

**Incremental Publish** The incremental publish process sends a message that contains only the rows where the data has been modified, plus the corresponding anchoring parent and grandparent rows. When a particular transactional event occurs, an incremental update of the transaction data is sent to other systems to notify them of the changes.

**Message Nodes** Each message node represents a publishing or subscribing system of a message. For example, the PeopleSoft Human Resources and PeopleSoft Financials databases are each defined as a message node even if they are both on the same server.

**Message Channels** Message channels group messages and the nodes to which they are published, so that messages are published sequentially. Each message must belong to only one message channel. Channels control the ordering of messages and define timeout parameters and error thresholds. Assign message nodes to a message channel when you define the message channel.

**Message Chunking** Chunking automatically breaks a message into several smaller messages based on the values in one or more of the fields in the level zero record. When publishing the entire contents of a table, you can use message chunking to publish only certain sets of data, or if a particular subscriber is interested in only a portion of the table.

**Request ID** Use the request ID to specify multiple requirements within the same run control.

**Run Control** You use run controls to produce full messages for objects at the same time. Run controls also associate publish rule definitions with the scheduled full publish process run. For example, you can set up a run control to publish both customer full messages and sales order full messages on a daily schedule.

# Performing a Full Data Publish of Current Effective Data

This section discusses how to:

- Create effective-dated messages.
- Define the message node, message channel, and message definition.
- Define chunking rules and ordering views.
- Create publish rule definitions.
- Create run controls for the Full Data Publish program.
- (Optional) Define message routing.

For full data messages that are intended for vendors who do not handle effective dating, use the Effective Date Publish utility and a current full message to publish only those rows that are currently active. Any future-dated rows are written to the delay table.

This section discusses the process involved in a full data publish of current effective data. It uses the CUSTOMER_FULLSYNC_EFF message as an example, but the methods and procedures that are described here apply to creating any effective-dated message.

## Pages Used for Full Data Publish of Current Effective Data

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Chunking Rule | CUSTOMER_FULLSYNC_EFF | Enterprise Components, Map Chunking Rules, Chunking Rule Definition | Define the chunking rule description. |
| Full Data Publish | EO_FULLDATAPUB | Enterprise Components, Initiate Processes, Full Data Publish | Create the run control for the Full Data Publish utility.<br>The run control associates publish rule definitions with the scheduled Full Publish process run. For example, you can set up a run control to publish both customers and sales orders full messages at the end of each day. |

## Creating Effective-Dated Messages

The structure of the current full message must be a clone of the original FullSync message structure. However, you must map effective-dated records to a record view that selects only those rows that contain current data.

The current full message for customer data, CUSTOMER_FULLSYNC_EFF, uses the following views that are created as ordering view records.

| Level | TARGET RECORDS | ORDERING VIEW RECORDS |
|---|---|---|
| Level 0 | CUSTOMER | CUSTOMER |
| Level 1 | CUST_ADDR_CNTCTC | CUST_ADDR_CNTCTC |
| Level 1 | CUST_ADDR_SEQ | CUST_ADDR_SEQ |
| Level 2 | CUST_ADDRESS (effective-dated) | CUST_ADDR_EF2VW (current effective-dated view) |
| Level 1 | CUST_CNTCT_SEQ | CUST_CNTCT_SEQ |
| Level 2 | CUST_CONTACT (effective-dated) | CUST_CNCT_EF2VW (current effective-dated view) |
| Level 3 | CUST_CNTCT_CARD (effective-dated) | CUST_CARD_EF_VW (current effective-dated view) |
| Level 3 | CUST_CNTCT_DOC (effective-dated) | CUST_DOC_EF_VW (current effective-dated view) |
| Level 3 | CUST_CNTCT_PHN (effective-dated) | CUST_PHN_EF_VW (current effective-dated view) |
| Level 3 | CUST_CNTCT_TYPE (effective-dated) | CUST_TYPE_EF_VW (current effective-dated view) |

## Defining the Message Node, Message Channel, and Message Definition

Create a full message definition that contains the necessary records in the publishing system. You can also set up message routing by using OnRouteTo PeopleCode.

**Note.** Remember to insert the message version first, otherwise you can't add the tables that compose the message.

Begin by setting up the node and the transaction and connector details by using the Integration Profile setup function of PeopleSoft Integration Broker. To set up the node:

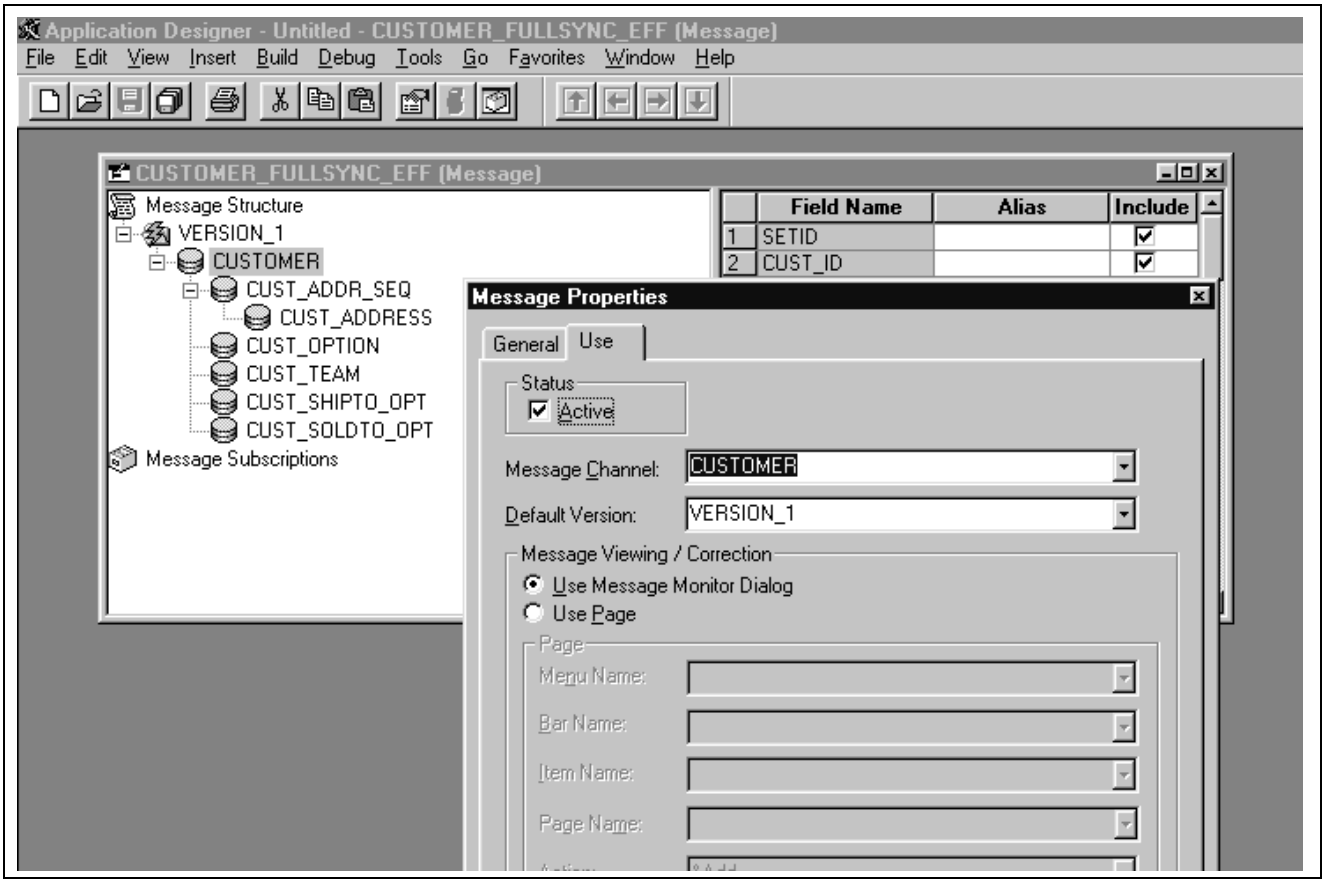
1. Create a node.
2. Set up the connector.

   The default target is PSHTTP, but you can instead provide the HTTP address for another target.
3. Associate the transaction to the node.

4. Provide the message name.
5. Designate whether the message is synchronous or asynchronous.
6. Create the message definition for the effective-dated message.

### Creating the Message Definition for the Message CUSTOMER_FULLSYNC_EFF

This example uses the message CUSTOMER_FULLSYNC.




Creating a message definition in PeopleSoft Application Designer

To create the message definition for the message CUSTOMER_FULLSYNC_EFF:

1. Open PeopleSoft Application Designer.
2. Create the current full message definition by copying the CUSTOMER_FULLSYNC message and adding the suffix *_EFF*.

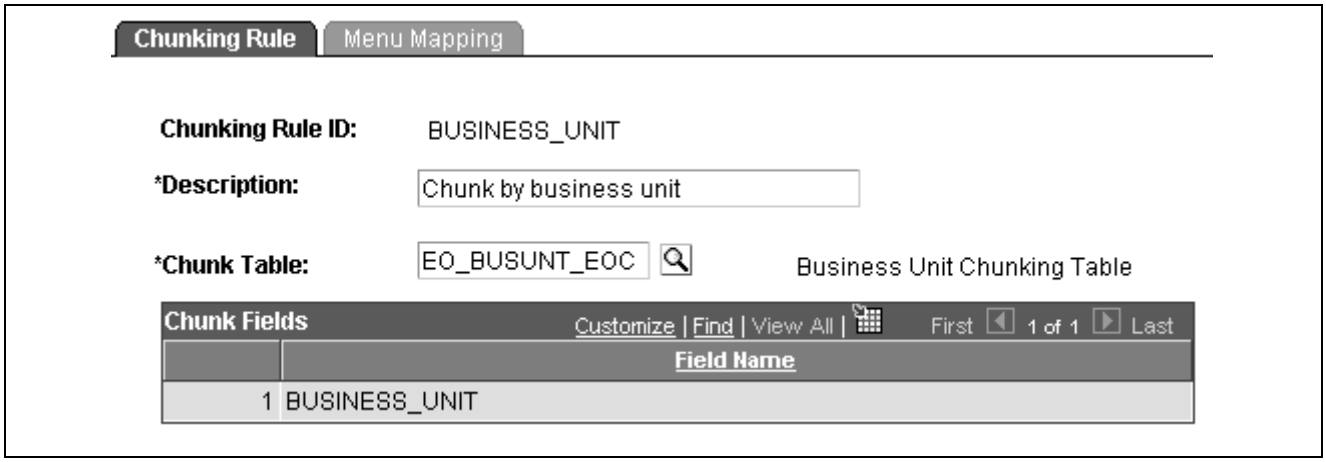
   **Note.** Messages must have the *_EFF* suffix to be effective-dated.

3. Assign the message channel.
4. Set the status to *Active*.
5. Click Save and close PeopleSoft Application Designer.

## Defining Chunking Rules and Ordering Views

In message chunking, all data within the message contains the same break field. (For example, if the break field is business unit, all transactions in the message are for the same business unit.)

The following message is chunked on setID by using the EO_SETID_EOC table:




Chunking Rule page

To ensure that the message publishes when you use chunking rules:

1. Add subscribing nodes to the chunking node table.
2. In PeopleSoft Application Designer, add OnRouteSend PeopleCode to return a list of subscribing nodes.

## Creating Publish Rule Definitions

For creating the current full message, the publish rule defines these options:

- Message header and trailer creation.
- Chunking rules.
- Ordering views.

Create publish rule definitions for each current full message definition. Considerations are:

- Specify only target records that are effective-dated.
- Select only the current effective row to list the ordering view record that should be used as an override when publishing the message.
- The Effective Date Publish utility makes a logic pass through the data for each publish rule definition.

  You can use this logic to order and chunk the data differently for each subscriber.

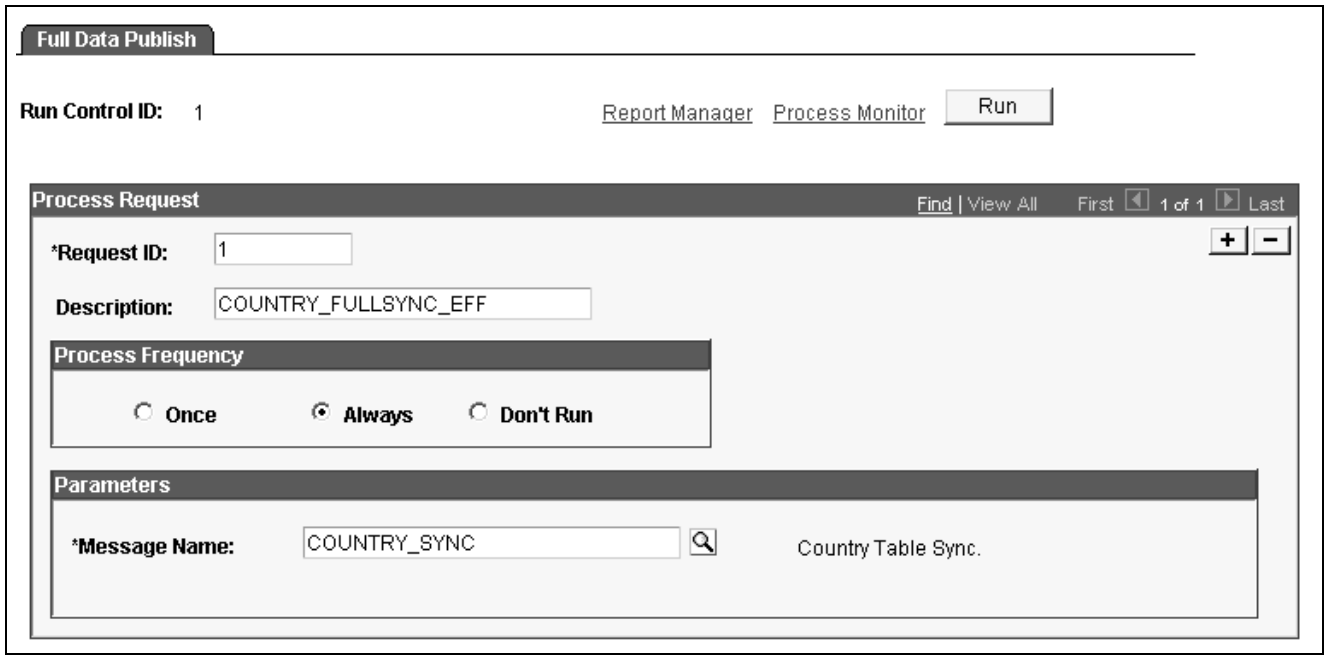
---

**Note.** When chunking a message, you must provide an ordering view for each record that includes the chunking fields. The fields in this view must appear in the same order as the primary keys, followed by any other keys that are needed for that record. If you override the normal key structure of the message records, you must provide the ordering views for each record to guarantee that the message reconstructs with the correct chunking, parent, or child key relationships.

---

# Creating Run Controls for the Full Data Publish Program

Access the Full Data Publish page.




Full Data Publish page

**Request ID** Enter request IDs to group the Description, Process Frequency, and Message Name parameters under one unique process request. A single run control ID can encompass multiple request IDs.

**Parameters** Select the name of the message to publish.

The PeopleSoft system adds a run control for the currently effective FullSync message that is chunked by the setID CUSTOMER_FULLSYNC_EFF_SETID.

**Note.** If you insert a new row, the same run control component can publish more than one message, so you can produce both the full message and the current effective-dated full message from the same PeopleSoft Process Scheduler run.

**Note.** You must set up the run control parameters to start the Full Data Publish program.

## Performing Full Table Replication

After you click Run on the Full Data Publish page, you can perform a table replication publish from the following screen:

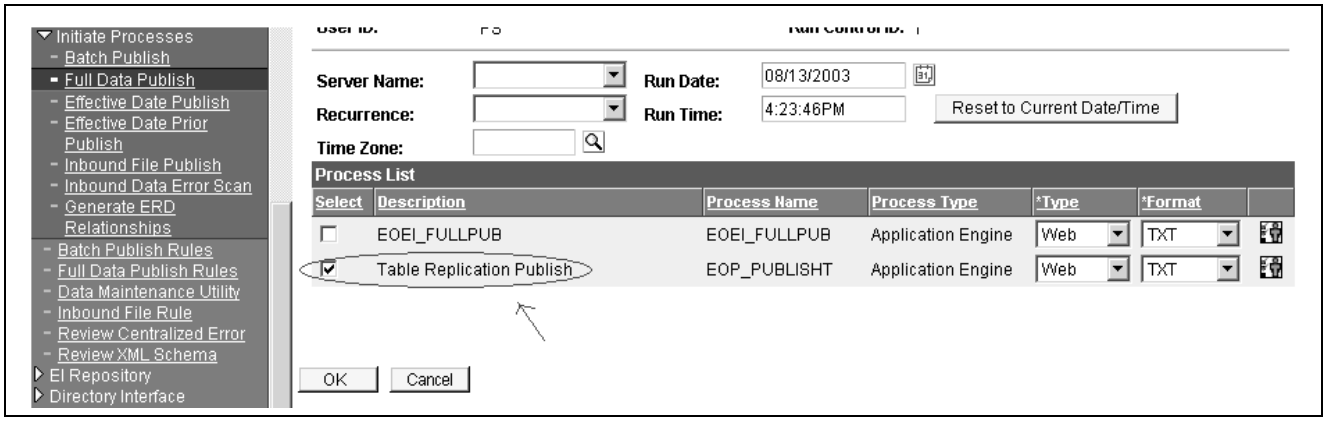



Table Replication Publish checkbox

To perform a full table replication, click the checkbox shown.

## Defining Message Routing

To define the message routing, you must insert PeopleCode. (Perform this step only if you're chunking a message.)

### Example

You want to route the customer message to the nodes that are defined in the SetID/Nodes page within the Publish Setup component. Add the following PeopleCode to the OnRouteSend PeopleCode that is associated with the message:

```
Declare Function GETSETIDNODES PeopleCode FUNCLIB_EOEIP.PUBLISH_ROUTE_PC Field⇒
Formula;
Local Message &MSG;
/* Call Function that looks at Setid of first transaction in the message and⇒
 returns a list of subscribing nodes to route the message */
&MSG = GetMessage();
GETSETIDNODES(&MSG, %Date);
```

# Publishing Incremental Messages of Current Effective Data

This section discusses how to:

- Create message definitions and assign the message channel in PeopleSoft Application Designer.
- Create subscription processes that open the generic effective-dated delay function.

For incremental messages, use subscription PeopleCode to copy current effective rows to a current incremental message for immediate publication, to strip out historic data, and to store future effective rows in a delay table. A regularly scheduled Application Engine program uses the delay table data as a trigger to publish future data when that data becomes effective.

## Creating Message Definitions and Assigning the Message Channel

You must create an incremental message definition that contains the necessary records in the publishing system. Specify the version first, otherwise you can't add the tables that compose the message. When the system requests a message channel, enter the channel definition that you previously selected.

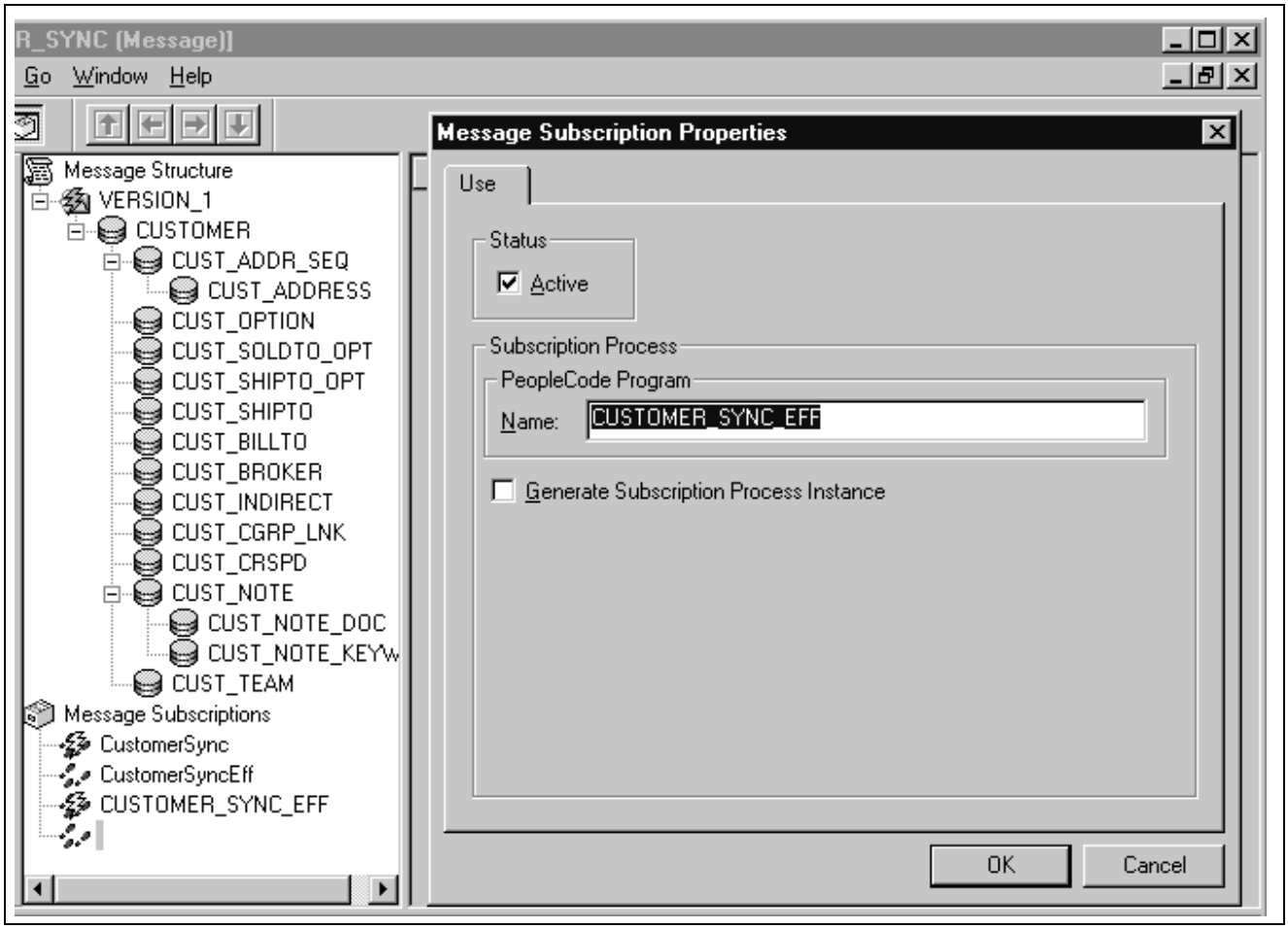
To create the message definition:

1. Open an instance of PeopleSoft Application Designer.
2. Create the current incremental message definition by copying the preferred message, such as the CUSTOMER_SYNC message, and adding the suffix *_EFF*.
3. Assign the message channel.
4. Set the status to *Active*.

## Creating Subscription Processes That Open the Generic Effective-Dated Delay Function

The PeopleSoft system includes a function called PROCESS_EFFDT_MSG. This function reads through an incremental message and processes the past, current, future, or non-effective-dated information. PROCESS_EFFDT_MSG resides within the record FUNCLIB_EOEIP, in the EFFDT_MSG_PC field.

Within PeopleSoft Application Designer, open the standard incremental message—CUSTOMER_SYNC in this example—and right-click Message Subscription to insert a new subscription (usually the message name with a suffix of *_EFF* attached). In this example, add a subscription to CUSTOMER_SYNC_EFF.




An incremental message within PeopleSoft Application Designer

When the Message Subscription Properties window appears:

1. Open the Process_Effdt_Msg generic function stored in record FUNCLIB_EOEIP.
2. Pass it the name of the current effective-dated incremental message.
3. Indicate whether only rows with *Active* effective status should be selected as the current effective-dated rows.

   If the &ACTIVE_EFFSTATUS parameter passed in is set to *False*, the current effective-dated row (whether active or inactive) is selected.
4. Pass the parameter set to True if only active effective-dated rows should be sent to the other system.

**Note.** The standard setting for the &ACTIVE_EFFSTATUS parameter is *False.*

# Publishing Effective-Dated Rows from the Delay Table

This section provides an overview of the publish of effective-dated rows from the delay table and discusses how to run the Effective Date Publish utility.

## Understanding Effective-Dated Row Publishing

Publishing effective-dated rows from a delay table requires:

- A future-dated entry in the delay table.
- The process request page.
- An Application Engine utility program that publishes future-dated information when it becomes current.

The Application Engine Effective Date Publish utility publishes effective-dated rows from a delay table by:

1. Retrieving from the delay table any entries that are effective within a date range.
2. Using the key strings from the delay table and record information for the current message to read the original application tables and retrieve the most current effective rows.
3. Publishing those rows to the current incremental message.

   The third-party application subscribes to the current incremental messages.

The PeopleSoft system allows for an end date range on the Effective Date Publish utility if the utility was not run for one day. The end date range enables the program to run on the next date.

You can run the Effective Date Publish utility multiple times during the day, and it deletes the information from the delay queue when the future effective date data becomes current and the message for that data has been published. The Effective Date Publish utility retrieves only the latest delay table information since the prior run.

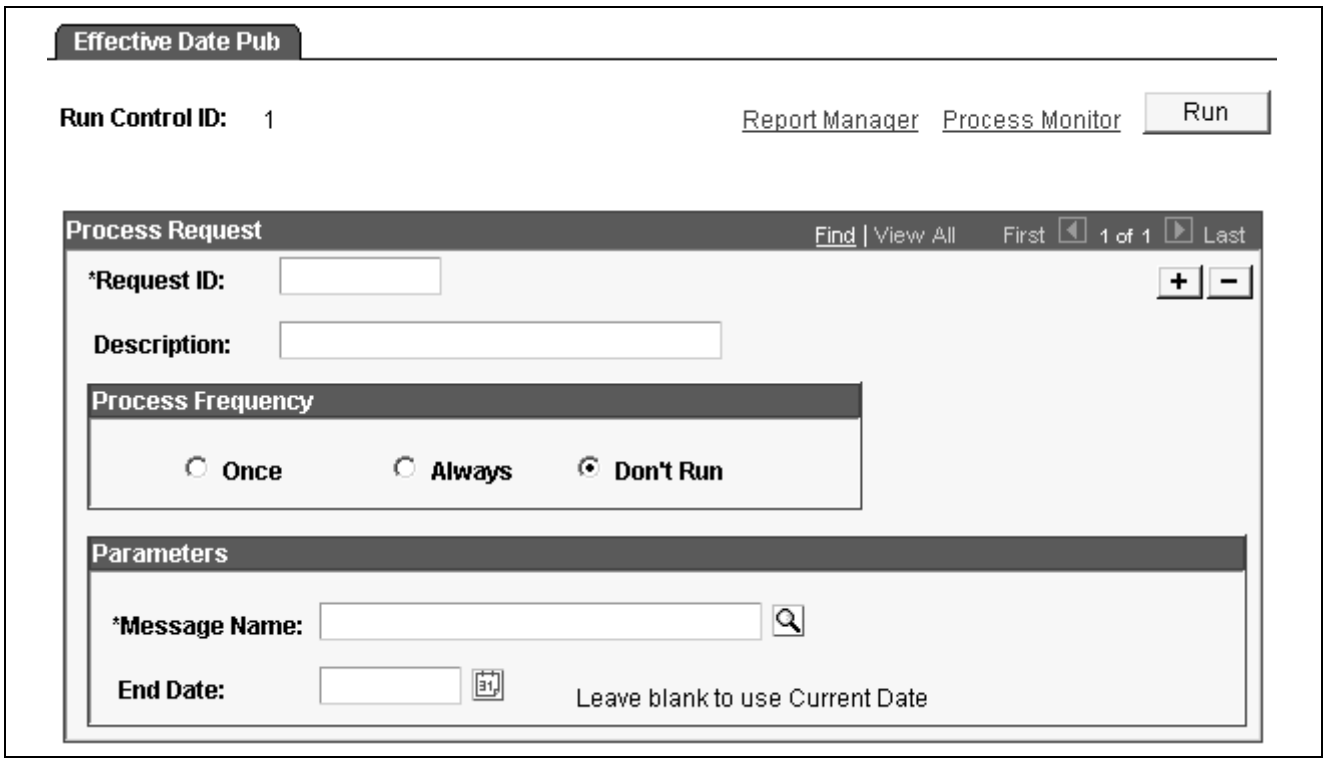
If the Effective Date Publish utility is invoked after not running for a period of time, it retrieves only the current row and publishes that as the active record. The presumption is that the subscribers want only the most current database information that is published.

## Page Used to Run the Effective Date Publish Utility

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| Effective Date Pub (effective date publish) | EO_EFFDTPUB | Enterprise Components, Initiate Processes, Effective Date Publish | Run the Effective Date Publish utility. |

## Running the Effective Date Publish Utility

Access the Effective Date Pub page.




Effective Date Pub page

**Message Name** Select the current incremental message to publish.

**End Date** Select the highest effective date to process from the delay table.

**Run** Click to run this request.

The Application Engine program uses the trigger records in the delay table and the end date parameter from the run control component to publish a current effective incremental message that contains all future-dated rows that are effective. This ensures that third-party systems that cannot manage future-dated records always receive currently active data on that data's effective date, even if that information was previously updated on the PeopleSoft system.

# CHAPTER 18

# Using the Flat File Utility

This chapter provides an overview of the Flat File utility and discusses how to:

- Process inbound flat files.
- Initiate file processing.
- Test inbound flat file processing.

## Understanding the Flat File Utility

When external systems send flat files to you for inbound transactions, you must develop complementary processes to translate incoming files into messages or translate outbound messages into files.

The flow for inbound file processing by using the Flat File utility is:

1. The utility receives a flat file in the form of a file layout object from an external system.

   The flat file consists of either:

   - The relevant data.
   - An index file that contains pointers to the data.

     Each index file lists the names of a set of data files to be processed. These files contain the application data, which is in one of the following formats: fixed record, Comma Separated Values (CSV), or XML.
2. The utility reads the file that is submitted for processing:
   - If the file is an index file, the Flat File utility loads the list of data files that are associated with each index file to be processed into a parameter table.
   - If it is a single data file, the utility inserts the single data file into a parameter table.

   **Note.** If additional fields in the file layout are not in the message definition, the additional fields are ignored during the copying of the flat file data to the message and are not included in the message.
3. The utility loops through the list of data files to be processed and reads each data file.
4. The utility copies the row sets of the data files into the message.
5. The utility publishes the message.
6. The subscribing systems receive the message and initiate normal inbound data processing.

# Processing Inbound Flat Files

You use the file layout definition to read and write from flat files.

To process inbound flat files:

1. Determine the necessary format of the inbound data.

   If there's an industry standard, use it for your file definition.

   If there's no industry standard, create a file layout object that mirrors your message object.
2. Identify the inbound process and its standard message.
3. Analyze the vendor's file structure and compare it to the standard message.

   Questions to answer include:

   - Can you use an existing message, or do you need to create a new one?
   - Can the customer conform to an existing enterprise integration point (EIP), or do you need to create one (along with corresponding subscription PeopleCode)?
4. Create the message definition.
5. Create a file layout definition with the same structure as the message definition to support the vendor file format.

   The hierarchical structure of the data in the File Layout Definition must match that of the message definition. For example, suppose a message has three levels: level zero, containing Record A, level one, containing records B and C, and level two, containing record D.

   All file layouts that are associated with this message must also have record A in level zero, record B and C in level one, and record D in level two.

   **Note.** The file layout does not need to contain the exact same fields as the message definition.

   For every record in your file layout, add a new file field, AUDIT_ACTN, as the first field in the record (except when the field already exists in the application table).

   You can associate more than one file layout to a single message. For example, vendor A may have a different number of fields than vendor B, so you may have two file layouts: one for A and one for B.

   Specify the file ID uniquely to include a row in a file, which is necessary in mapping the data to its proper record. Include start and end points when dealing with more than one record in a file layout.

   **Note.** Each record in the file layout has a file record ID attribute. Do not confuse this with the file layout ID. The file layout ID determines whether a new layout is encountered for multiple file layout processing.

   When you subscribe to the message and normal inbound data processing begins, you can invoke the SetDefault PeopleCode function to set the default values for fields that were not present in the input file.
6. Update or create the inbound file rule pages.
7. Create subscription PeopleCode in PeopleSoft Application Designer to process the message.

Have the standard inbound process subscribe to and process the message normally. The standard message definition should have a subscription process that initiates the normal inbound processing for that object to which you hook your application logic to process the file data.

8. Test the inbound flat file processing.

**Note.** You can process multiple inbound flat files at one time. By specifying an inbound index file as part of the Flat File utility parameters, the system reads all input files within the index file and uses the associated file layout object and message to convert the data. Similarly, specify a wildcard in the filename in the inbound file rule component, but make sure that all files that meet the wildcard criteria correspond to the file layout and message mapping that are defined.

# Initiating File Processing

This section discusses how to:

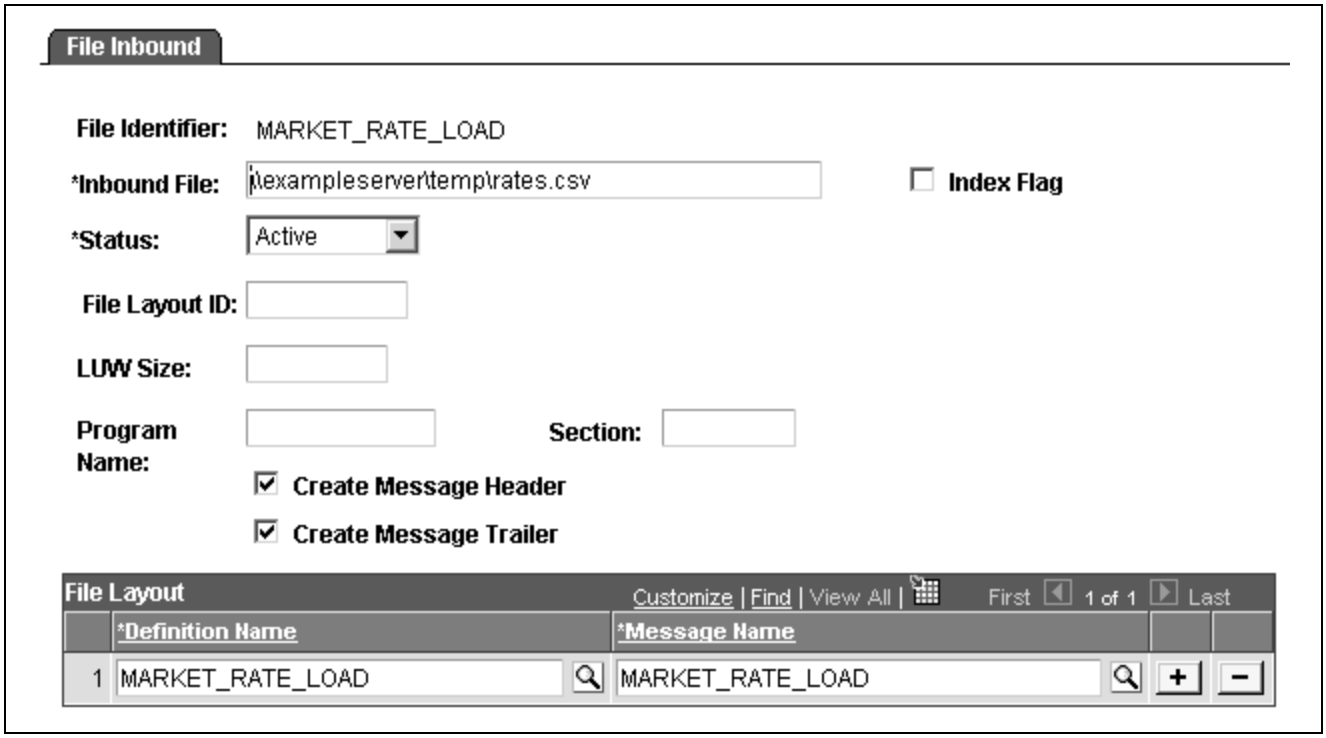
- Set up inbound flat file processing.
- Initiate inbound flat file processing.

## Pages Used to Initiate File Processing

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| File Inbound | EO_FILE_INBOUND | Enterprise Components, Inbound File Rule | Set up inbound flat file processing. |
| Inbound File | EO_FILETOMSG | Enterprise Components, Integration Definitions, Initiate Processes, Inbound File Publish | Initiate inbound flat file processing. This file-to-message processing function reads the file rowset and publishes it as a message. |

## Setting Up Inbound Flat File Processing

Access the File Inbound page.




File Inbound page

**File Identifier** Displays the inbound file that you are associating with the rule.

**Inbound File** Enter the index file name or the data file name. Specify the full path information. The PeopleCode program uses the *%filepath_absolute* variable when opening the file.

**Index Flag** Select to distinguish between the index and the data file.

**Status** Select whether this inbound file rule is *Active* or *Inactive.* The default value is *Inactive.*

**File Layout ID** Select a layout to associate with the file.

**LUWSize** (logical unit of work size) Enter the number of level zero rows that are in each message, to limit the message size. The output message is normally determined by the MaxMessageSize system parameter.

**Program Name** and **Section** Select a PeopleSoft Application Engine program and section to invoke when the utility finishes processing data.

**Create Message Header** Select to create a header message. Use the header message as a trigger in the subscription process to initialize tables before receiving the data messages. Default value is selected.

**Create Message Trailer** Select to create a trailer message. Use the trailer message as a trigger in the subscription process to indicate that all the data messages have been received. Default value is selected.

### File Layout

**Definition Name** and **Message Name**

If the File Layout ID field is blank, this field should contain only one entry.

If the File Layout ID field is not blank, this scroll area must contain an entry for each file layout definition name that is specified in the inbound file.

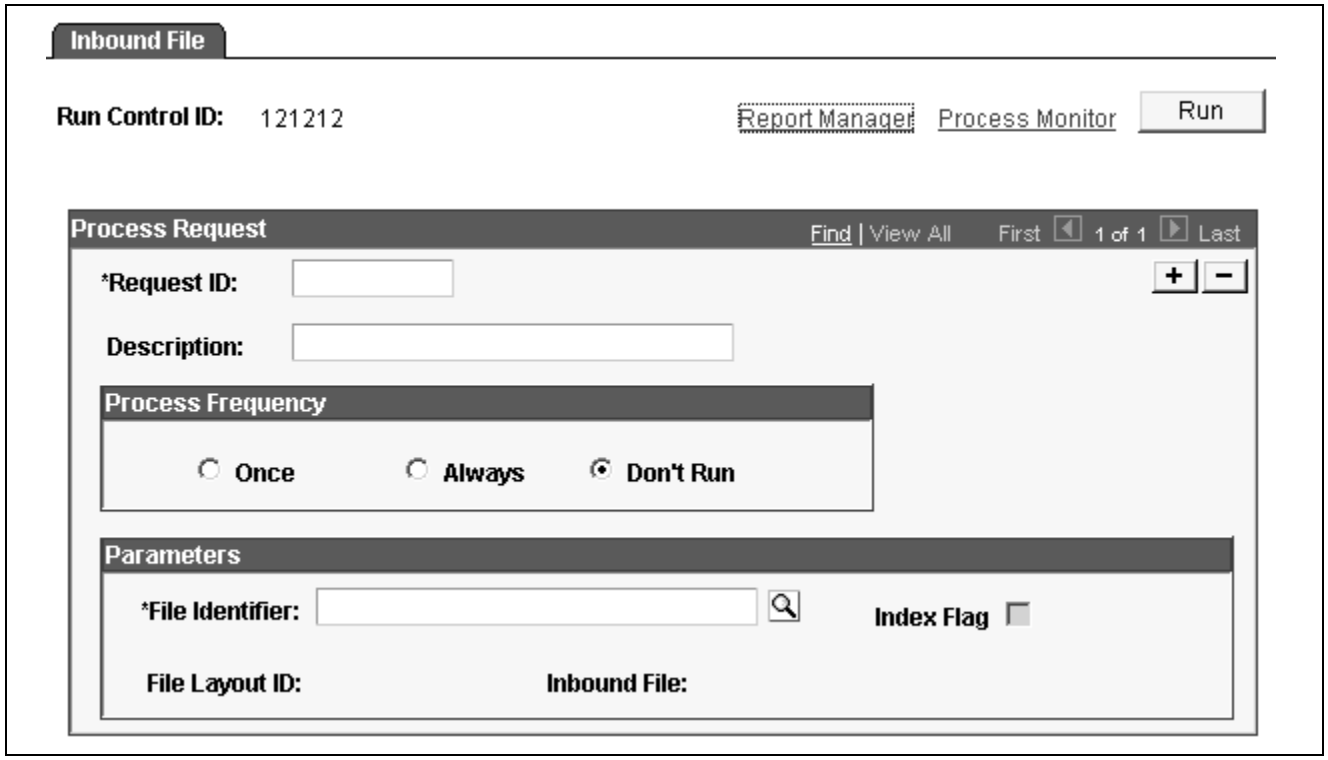
---

**Note.** Use the wildcards * and ? for the file name but not for the directory path. The file layout and message mapping must be valid for all files that meet the wildcard criteria.

---

## Initiating Inbound Flat File Processing

Access the Inbound File page.




Inbound File page

### Parameters

**File Identifier** Select or enter the name of the file identifier that you set up in the File Inbound page. The file identifier is tied to the publish rules.

**Run** Click to run this request.

### Publishing a New Message

The Inbound File page runs an Application Engine process that initiates the file-to-message processing. The file-to-message processing function reads the file rowset and publishes it as a message.

If an index file exists when the inbound conversion process runs, the Application Engine program loads the list of files to be converted into a parameter table and completes a commit. The Application Engine program uses the list of files within the parameter table to restart the processing if a particular flat file fails. If a single data file is provided, then the rowset processing immediately begins.

The file publish process goes through each of the rowsets of the file layout and copies them into the message row sets.

If the audit action (AUDIT_ACTN) exists in the file, it is copied to the PSCAMA record. If the audit action does not exist in the file, the publishing process uses the default value that is specified in the file layout field property.

The Flat File utility publishes a new message when one of the following exists:

- Maximum message size is exceeded.
- Logical unit of work publish size is reached.
- A new file layout is detected.
- End of file is reached.

The Application Engine program completes a commit every time a message is published from a file. After conversion, the flat file remains in the parameter table with a status of *Processed.*

**Note.** The file layout should exactly match the message layout (excluding the PSCAMA record) and should use the same character set as that used by the file: either American National Standards Institute or Unicode.

# Testing Inbound Flat File Processing

To test inbound files:

1. Create a sample flat file, or ask the third-party vendor for a sample flat file.
2. Launch the Flat File utility.
    a. Through the browser, sign in to PeopleSoft Internet Architecture.
    b. Select Enterprise Components, Management, Inbound File Rule.
3. Run the Application Engine program to convert the sample flat file to a message by running Message Monitor.

    Use Message Monitor to ensure the inbound file processing created a publish message that contains the sample flat file data.

    a. Verify that the standard inbound subscription process received the message and processed it into the application tables.
    b. Determine whether the values become the inherited values (if you used the inherited value feature in file layout).
    c. Validate that the production or staging tables loaded with the correct field values.

        For production tables, look in the PeopleSoft application pages.

        For staging tables, use either the PeopleSoft application pages or run a query by using PeopleSoft Query.

d. Ensure that the date formats conform.

## Code Sample

The following is subscription code for the Market Rate Load message definition:

```
Declare Function Delete_Existing_Data PeopleCode FUNCLIB_EOEIP.SUBSCRIBE_MSG_PC⇒
 FieldFormula;
Declare Function Do_RelLang PeopleCode FUNCLIB_EOEIP.SUBSCRIBE_MSG_PC FieldFormula;
Local Message &MSG, &MSG2, &MSG3;
Local Rowset &MSG_ROWSET, &msg_rowset2, &MSG_ROWSET3;
Local number &I, &K, &S;
Local string &TEMP;
Local Record &REC;
Global string &MSG_LANG_CD;

&MSG = GetMessage();
If &MSG = Null Then
   Exit (1);
Else

   If &MSG.IsActive Then
      &MSG_ROWSET = &MSG.GetRowset();

      Evaluate &MSG_ROWSET(1).PSCAMA.MSG_SEQ_FLG.Value
      When "H"
         /* If the current message is the header msg, then prepare the table for⇒
 insert */
         Delete_Existing_Data(&MSG);
         Break;
      When "T"
         Break;
      When-Other

         /*build new message for the inserted row. used for multiple PeopleSoft⇒
 database  occurrences */
         &MSG2 = CreateMessage(Message.MARKET_RATE_SYNC);
         &msg_rowset2 = &MSG2.GetRowset();
         &S = 1;

         &MSG3 = CreateMessage(Message.MARKET_RATE_DEFN_SYNC);
         &MSG_ROWSET3 = &MSG3.GetRowset();
         &K = 1;

         For &I = 1 To &MSG_ROWSET.ActiveRowCount

            /* Set rate_mult and rate_div equal to each other for now. The rest of⇒
 this code figures          out what the true rate_mult and rate_div should be. ⇒
 When it figures it out, it will          overlay the appropriate field with the⇒
 derived value.*/
```

```
            &RATE_MULT = &MSG_ROWSET(&I).RT_RATE_TBL.RATE_MULT.Value;
            If All(&RATE_MULT) Then
               &MSG_ROWSET(&I).RT_RATE_TBL.RATE_DIV.Value = &RATE_MULT;
            End-If;

            /* Get the quotation method if the rate is for FX  */

            SQLExec("SELECT RT_CATEGORY FROM PS_RT_INDEX_TBL WHERE RT_RATE_INDEX =⇒
 :1", &MSG_ROWSET(&I).RT_RATE_TBL.RT_RATE_INDEX.Value, &RT_CATEGORY);

            If &RT_CATEGORY = "10" Then

               SQLExec("Select Rate_direct, quote_units, rate_decimals  from ps_⇒
curr_quote_tbl where from_cur = :1 and to_cur = :2 and effdt <= %DateIn(:3)",⇒
 &MSG_ROWSET(&I).RT_RATE_TBL.FROM_CUR.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TO_⇒
CUR.Value, &MSG_ROWSET(&I).RT_RATE_TBL.EFFDT.Value, &RATE_DIRECT, &UNITS, &RATE_⇒
DECIMALS);

               /* If currency pair is not in curr_quote_tbl, set quote_units = 1⇒
 and rate_decimals = 4  */

               If None(&UNITS) Then
                  &UNITS = 1;
                  &RATE_DECIMALS = 4;
                  &RATE_DIRECT = "D";
               End-If;

               /* Fox InDirect FX rates set RATE_MULT to 1 else set RATE_DIV to⇒
 1.*/

               If &RATE_DIRECT <> "I" Then
                  &MSG_ROWSET(&I).RT_RATE_TBL.RATE_DIV.Value = 1;
               Else
                  &MSG_ROWSET(&I).RT_RATE_TBL.RATE_MULT.Value = 1;
               End-If;
            Else

               /* for non FX rate set RATE_DIV to 1 */
               &MSG_ROWSET(&I).RT_RATE_TBL.RATE_DIV.Value = 1;

            End-If;

            /* CREATE AND INSERT TO RECORD */
            &REC = CreateRecord(Record.RT_RATE_VW);
            &MSG_ROWSET(&I).RT_RATE_TBL.CopyFieldsTo(&REC);
            &REC.RT_EFFDT.Value = &MSG_ROWSET(&I).RT_RATE_TBL.EFFDT.Value;
            &REC.Insert();

            /* build RATE SYNC message */
            If &S <> 1 Then
```

```
               &msg_rowset2.InsertRow(&S - 1);
            End-If;
            &MSG_ROWSET(&I).GetRecord(1).CopyFieldsTo(&msg_rowset2(&S).GetRecord⇒
(1));
            &MSG_ROWSET(&I).GetRecord(2).CopyFieldsTo(&msg_rowset2(&S).GetRecord⇒
(2));
            &S = &S + 1;

            /* For FX rates check for reciprocal currency pair in rate table,⇒
 update if present, insert if not*/

            If &RT_CATEGORY = "10" And

                  &MSG_ROWSET(&I).RT_RATE_TBL.FROM_CUR.Value <> &MSG_ROWSET(&I).RT_⇒
RATE_TBL.TO_CUR Then

               /* Check for reciprocal in the rt_rate_tbl */
               &TEMP = "";
               SQLExec("Select 'x' from ps_rt_rate_tbl where rt_rate_index =  :1⇒
 and term = :2 and from_cur = :3 and to_cur = :4 and rt_type = :5 and effdt =⇒
 %DateIn(:6)", &MSG_ROWSET(&I).RT_RATE_TBL.RT_RATE_INDEX.Value, &MSG_ROWSET(&I).RT_⇒
RATE_TBL.TERM.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TO_CUR.Value, &MSG_ROWSET(&I).RT_⇒
RATE_TBL.FROM_CUR.Value, &MSG_ROWSET(&I).RT_RATE_TBL.RT_TYPE.Value, &MSG_ROWSET⇒
(&I).RT_RATE_TBL.EFFDT.Value, &TEMP);

               If &TEMP <> "x" Then
                  /* From cur/to cur reciprocal does no exist, insert */
                  SQLExec("Insert into ps_rt_rate_tbl (RT_RATE_INDEX, TERM, FROM_⇒
CUR, TO_CUR, RT_TYPE, EFFDT, RATE_MULT, RATE_DIV) values (:1 , :2 , :3 , :4 , :5⇒
 ,%DateIn(:6) ,:7 ,:8)", &MSG_ROWSET(&I).RT_RATE_TBL.RT_RATE_INDEX.Value, &MSG_⇒
ROWSET(&I).RT_RATE_TBL.TERM.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TO_CUR.Value, &MSG_⇒
ROWSET(&I).RT_RATE_TBL.FROM_CUR.Value, &MSG_ROWSET(&I).RT_RATE_TBL.RT_TYPE.Value,⇒
 &MSG_ROWSET(&I).RT_RATE_TBL.EFFDT.Value, &MSG_ROWSET(&I).RT_RATE_TBL.RATE_⇒
DIV.Value, &MSG_ROWSET(&I).RT_RATE_TBL.RATE_MULT.Value);
                  &ACTION = "A"
               Else
                  /* entry exist so update */
                  SQLExec("Update ps_rt_rate_tbl set rate_mult = :7, rate_div = :8⇒
 where rt_rate_index =  :1 and term = :2 and from_cur = :3 and to_cur = :4  and rt_⇒
type = :5 and effdt = %DateIn(:6)", &MSG_ROWSET(&I).RT_RATE_TBL.RT_RATE_⇒
INDEX.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TERM.Value, &MSG_ROWSET(&I).RT_RATE_⇒
TBL.TO_CUR.Value, &MSG_ROWSET(&I).RT_RATE_TBL.FROM_CUR.Value, &MSG_ROWSET(&I).RT_⇒
RATE_TBL.RT_TYPE.Value, &MSG_ROWSET(&I).RT_RATE_TBL.EFFDT.Value, &MSG_ROWSET⇒
(&I).RT_RATE_TBL.RATE_DIV.Value, &MSG_ROWSET(&I).RT_RATE_TBL.RATE_MULT.Value);
                  &ACTION = "C";
               End-If;

               /* build RATE SYNC message for reciprocal */
               If &S <> 1 Then
                  &msg_rowset2.InsertRow(&S - 1);
```

```
               End-If;
               &MSG_ROWSET(&I).GetRecord(1).CopyFieldsTo(&msg_rowset2(&S).Get⇒
Record(1));
               &MSG_ROWSET(&I).GetRecord(2).CopyFieldsTo(&msg_rowset2(&S).Get⇒
Record(2));
               &msg_rowset2(&S).PSCAMA.AUDIT_ACTN.Value = &ACTION;

               &msg_rowset2(&S).RT_RATE_TBL.TO_CUR.Value = &MSG_ROWSET(&I).RT_RATE_⇒
TBL.FROM_CUR.Value;
               &msg_rowset2(&S).RT_RATE_TBL.FROM_CUR.Value = &MSG_ROWSET(&I).RT_⇒
RATE_TBL.TO_CUR.Value;
               &msg_rowset2(&S).RT_RATE_TBL.RATE_MULT.Value = &MSG_ROWSET(&I).RT_⇒
RATE_TBL.RATE_DIV.Value;
               &msg_rowset2(&S).RT_RATE_TBL.RATE_DIV.Value = &MSG_ROWSET(&I).RT_⇒
RATE_TBL.RATE_MULT.Value;

               &S = &S + 1;

               /* check for existence in the rt_rate_def_tbl for the currency pair⇒
 */
               &TEMP = "";
               SQLExec("Select 'x' from ps_rt_rate_def_tbl where rt_rate_index =  :⇒
1 and term = :2 and from_cur = :3 and to_cur = :4 ", &MSG_ROWSET(&I).RT_RATE_⇒
TBL.RT_RATE_INDEX.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TERM.Value, &MSG_ROWSET⇒
(&I).RT_RATE_TBL.FROM_CUR.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TO_CUR.Value,⇒
 &TEMP);
               If &TEMP <> "x" Then
                  SQLExec("Insert into ps_rt_rate_def_tbl (RT_RATE_INDEX, TERM,⇒
 FROM_CUR, TO_CUR, MAX_VARIANCE, ERROR_TYPE, INT_BASIS) values (:1, :2, :3, :4,⇒
 2.5, 'WAR', ' ') ", &MSG_ROWSET(&I).RT_RATE_TBL.RT_RATE_INDEX.Value, &MSG_ROWSET⇒
(&I).RT_RATE_TBL.TERM.Value, &MSG_ROWSET(&I).RT_RATE_TBL.FROM_CUR.Value, &MSG_⇒
ROWSET(&I).RT_RATE_TBL.TO_CUR.Value);

                  /* build DEFN SYNC message */
                  If &K <> 1 Then
                     &MSG_ROWSET3.InsertRow(&K - 1);
                  End-If;
                  &MSG_ROWSET(&I).GetRecord(1).CopyFieldsTo(&MSG_ROWSET3(&K).Get⇒
Record(1));
                  &MSG_ROWSET(&I).GetRecord(2).CopyFieldsTo(&MSG_ROWSET3(&K).Get⇒
Record(2));
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.MAX_VARIANCE.Value = 2.5;
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.ERROR_TYPE.Value = "WAR";
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.INT_BASIS.Value = " ";
                  &K = &K + 1;
               End-If;
               /* insert the reciprocal into the market rate defintion table if it⇒
 does not exist */
               &TEMP = "";
               SQLExec("Select 'x' from ps_rt_rate_def_tbl where rt_rate_index =  :⇒
```

```
1 and term = :2 and from_cur = :3 and to_cur = :4 ", &MSG_ROWSET(&I).RT_RATE_⇒
TBL.RT_RATE_INDEX.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TERM.Value, &MSG_ROWSET⇒
(&I).RT_RATE_TBL.TO_CUR.Value, &MSG_ROWSET(&I).RT_RATE_TBL.FROM_CUR.Value,⇒
 &TEMP);
               If &TEMP <> "x" Then
                  SQLExec("Insert into ps_rt_rate_def_tbl (RT_RATE_INDEX, TERM,⇒
 FROM_CUR, TO_CUR, MAX_VARIANCE, ERROR_TYPE, INT_BASIS) values (:1, :2, :3, :4,⇒
 2.5, 'WAR', ' ') ", &MSG_ROWSET(&I).RT_RATE_TBL.RT_RATE_INDEX.Value, &MSG_ROWSET⇒
(&I).RT_RATE_TBL.TERM.Value, &MSG_ROWSET(&I).RT_RATE_TBL.TO_CUR.Value, &MSG_⇒
ROWSET(&I).RT_RATE_TBL.FROM_CUR.Value);

                  /* build DEFN SYNC message */
                  If &K <> 1 Then
                     &MSG_ROWSET3.InsertRow(&K - 1);
                  End-If;
                  &MSG_ROWSET(&I).GetRecord(1).CopyFieldsTo(&MSG_ROWSET3(&K).Get⇒
Record(1));
                  &MSG_ROWSET(&I).GetRecord(2).CopyFieldsTo(&MSG_ROWSET3(&K).Get⇒
Record(2));
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.MAX_VARIANCE.Value = 2.5;
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.ERROR_TYPE.Value = "WAR";
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.INT_BASIS.Value = " ";
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.FROM_CUR.Value = &MSG_ROWSET⇒
(&I).GetRecord(1).TO_CUR.Value;
                  &MSG_ROWSET3(&K).RT_RATE_DEF_TBL.TO_CUR.Value = &MSG_ROWSET⇒
(&I).GetRecord(1).FROM_CUR.Value;
                  &K = &K + 1;
               End-If;
            End-If;

            If &MSG2.Size > %MaxMessageSize And
                  &MSG2.IsActive Then
               &MSG2.Publish();
               &S = 1;
               &MSG2 = CreateMessage(Message.MARKET_RATE_SYNC);
               &msg_rowset2 = &MSG2.GetRowset();
            End-If;

            If &MSG3.Size > %MaxMessageSize And
                  &MSG3.IsActive Then
               &MSG3.Publish();
               &K = 1;
               &MSG3 = CreateMessage(Message.MARKET_RATE_DEFN_SYNC);
               &MSG_ROWSET3 = &MSG3.GetRowset();
            End-If;

         End-For;
         If &MSG2 <> Null And
               &msg_rowset2(1).PSCAMA.AUDIT_ACTN.Value > "" And
               &MSG2.IsActive Then
```

```
            &MSG2.Publish();
         End-If;
         If &MSG3 <> Null And
               &MSG_ROWSET3(1).PSCAMA.AUDIT_ACTN.Value > "" And
               &MSG3.IsActive Then
            &MSG3.Publish();
         End-If;
      End-Evaluate;

   End-If;
End-If;
```

# CHAPTER 19

# Using the XML Schema Utility

This chapter provides an overview of the XML Schema utility and discusses using the XML Schema utility.

## Understanding the XML Schema Utility

PeopleSoft Open Integration Framework enables near real-time messaging and transactions by using a format that is based on XML to convey information between diverse applications in a standard way. To take advantage of this standardization, you must obtain clear XML definitions (schemas) for each application message, component interface, or business interlink.

The XML Schema utility provides the following features:

- Output options for the XML Schema utility, document type definition (DTD), or BizTalk definition for all application messages.
- The ability to create an XML definition for a single object, for all of the objects, or for all of the objects by a specific owner.
- A single flat file for each XML definition that is written to your system's %TEMP directory (when you use the Microsoft Windows client) or the server's common access file directory (when you use PeopleSoft Internet Architecture).
- An application foundation for future standards of XML definitions.

## Generating the XML Schema

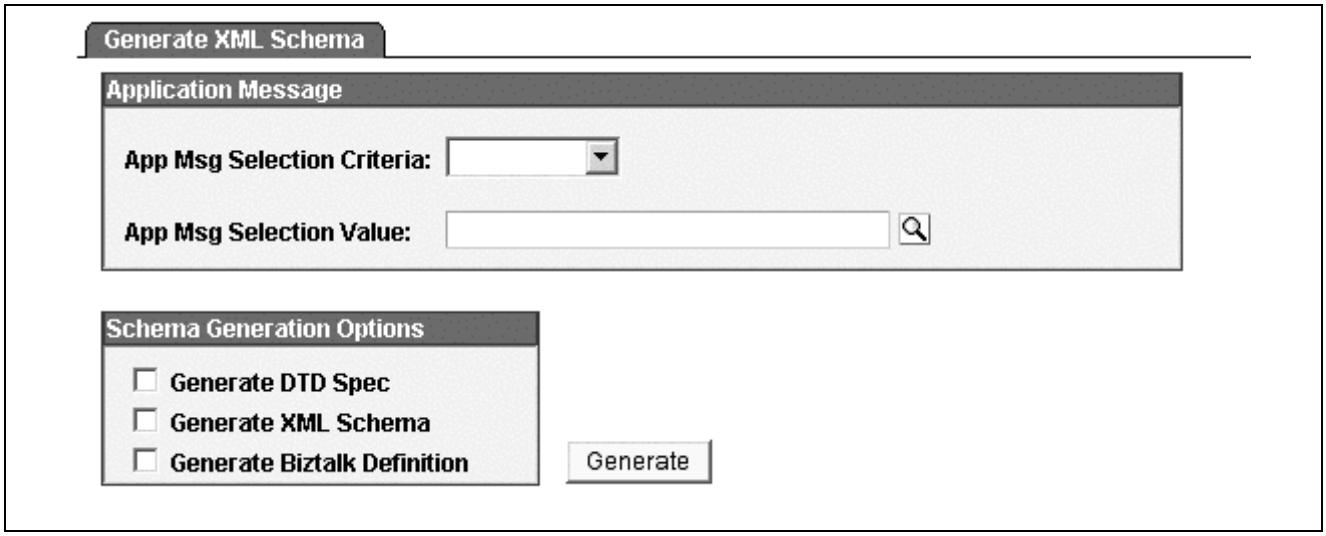
This section discusses how to generate the XML Schema.

### Page Used to Generate XML Schema

| Page Name | Object Name | Navigation | Usage |
|---|---|---|---|
| XML Schema | EO_GEN_XML_DATA | Enterprise Components, Integration Definitions, Review XML Schema | Generate DTDs, XML schemas, and BizTalk definitions. |

### Generating the XML Schema

Access the Generate XML Schema page.




Generate XML Schema page

**App Msg Selection Criteria** (application message selection criteria) — Select application selection message criteria. Values are *All Msg* (all messages), *Channel*, *Owner ID*, and *Single Msg* (single message). Depending on the selection criteria, you can enter the selection value (if already known) or search for the value.

**Generate DTD Spec** — Select for DTD spec output format.

**Generate XML Schema** — Select for XML schema output format.

**Generate Biztalk Definition** — Select for BizTalk definition output format.

**Generate** — Click to generate the selected output formats.

The utility queries the relevant PeopleTools tables to generate the selected types of XML schemas and writes the results to the server's file directory or your system's Temp directory, depending on the client you use.

To produce an XML schema, DTD, or BizTalk definition:

1. Define the selection criteria for application messages, component interfaces, and business interlinks.
2. Select XML schemas, DTDs, or BizTalk definitions for application message, component interface, and business interlink objects.

   PeopleCode sends a query to the PeopleTools tables to create the selected types of XML definitions.

   The XML Schema utility then writes the definitions to the file directory as specified by the PS_SERVDIR environment variable when using PeopleSoft Internet Architecture or the %TEMP directory of your system when using the Microsoft Windows client.

## Interpreting Sample Output

The following code shows three samples of output for the same application message (in this case, MARKET_RATE_TYPE_FULLSYNC) in BizTalk, DTD, and XML schema formats.

### Example: BizTalk

The following code shows MARKET_RATE_TYPE_FULLSYNC in BizTalk format:

```
<BizTalk xmlns="urn:schemas-biztalk-org:BizTalk/biztalk-0.81.xml">
<Body>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:element name="MARKET_RATE_TYPE_FULLSYNC" type="MARKET_RATE_TYPE_FULLSYNCType"⇒
/>

<xsd:complexType name="MARKET_RATE_TYPE_FULLSYNCType">
<xsd:sequence>
   <xsd:element name="FieldTypes" type="FieldTypesType"/>
   <xsd:element name="MsgData" type="MsgDataType"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldTypesType">
<xsd:sequence>
   <xsd:element name="RT_TYPE_TBL" type="FieldTypesRT_TYPE_TBLType"/>
   <xsd:element name="PSCAMA" type="PSCAMA"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldTypesRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE" type="FieldTypesFieldType"/>
   <xsd:element name="DESCR" type="FieldTypesFieldType"/>
   <xsd:element name="DESCRSHORT" type="FieldTypesFieldType"/>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="PSCAMA">
   <xsd:sequence>
   <xsd:element name="LANGUAGE_CD" type="LANGUAGE_CDType" minOccurs="0" maxOccurs=⇒
"1"/>
   <xsd:element name="AUDIT_ACTN" type="AUDIT_ACTNType"/>
   <xsd:element name="BASE_LANGUAGE_CD" type="BASE_LANGUAGE_CDType" minOccurs="0"⇒
 maxOccurs="1"/>
   <xsd:element name="MSG_SEQ_FLG" type="MSG_SEQ_FLGType" minOccurs="0" maxOccurs=⇒
"1"/>
   <xsd:element name="PROCESS_INSTANCE" type="PROCESS_INSTANCEType" minOccurs="0"⇒
 maxOccurs="1"/>
   <xsd:element name="PUBLISH_RULE_ID" type="PUBLISH_RULE_IDType" minOccurs="0"⇒
 maxOccurs="1"/>
   <xsd:element name="MSGNODENAME" type="MSGNODENAMEType" minOccurs="0" maxOccurs=⇒
"1"/>
   </xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="LANGUAGE_CDType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
```

```
</xsd:complexType>

<xsd:complexType name="AUDIT_ACTNType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="BASE_LANGUAGE_CDType"  >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="MSG_SEQ_FLGType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="PROCESS_INSTANCEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="PUBLISH_RULE_IDType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="MSGNODENAMEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>


<xsd:complexType name="MsgDataType">
      <xsd:sequence>
   <xsd:element name="Transaction">
      <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="RT_TYPE_TBL" type="MsgDataRT_TYPE_TBLType"/>
         <xsd:element name="PSCAMA" type="PSCAMA"/>
      </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
      </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MsgDataRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE">
<xsd:simpleType>
<xsd:restriction base="xsd:string" >
   <xsd:pattern value="[A-Z]{1-5}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCR">
```

```
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-30}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCRSHORT">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-10}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="FieldTypesFieldType">
   <xsd:attribute name="type" type="fieldtypes"/>
</xsd:complexType>

<xsd:simpleType name="fieldtypes">
<xsd:restriction base="xsd:string">
   <xsd:enumeration value="CHAR"/>
   <xsd:enumeration value="NUMBER"/>
   <xsd:enumeration value="DATE"/>
   <xsd:enumeration value="DATETIME"/>
   <xsd:enumeration value="TIME"/>
</xsd:restriction>
</xsd:simpleType>

</xsd:schema>
</Body>
</BizTalk>
```

### Example: DTD

The following code shows MARKET_RATE_TYPE_FULLSYNC in DTD format:

```
<!ELEMENT MARKET_RATE_TYPE_FULLSYNC (FieldTypes, MsgData)>

   <!ENTITY % recordtypes "class (R | SR) #REQUIRED" >
   <!ENTITY % fieldtypes "type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED" >

<!ELEMENT FieldTypes (RT_TYPE_TBL, PSCAMA)>

<!ELEMENT PSCAMA (LANGUAGE_CD?, AUDIT_ACTN, BASE_LANGUAGE_CD?, MSG_SEQ_FLG?,⇒
 PROCESS_INSTANCE?, PUBLISH_RULE_ID?, MSGNODENAME?)>
   <!ATTLIST PSCAMA class (R | SR) #REQUIRED>
   <!ELEMENT LANGUAGE_CD (#PCDATA)>
```

```
      <!ATTLIST LANGUAGE_CD type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT AUDIT_ACTN (#PCDATA)>
      <!ATTLIST AUDIT_ACTN type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT BASE_LANGUAGE_CD (#PCDATA)>
      <!ATTLIST BASE_LANGUAGE_CD type (CHAR | NUMBER | DATE | TIME | DATETIME)⇒
#IMPLIED>
   <!ELEMENT MSG_SEQ_FLG (#PCDATA)>
      <!ATTLIST MSG_SEQ_FLG type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT PROCESS_INSTANCE (#PCDATA)>
      <!ATTLIST PROCESS_INSTANCE type (CHAR | NUMBER | DATE | TIME | DATETIME)⇒
#IMPLIED>
   <!ELEMENT PUBLISH_RULE_ID (#PCDATA)>
      <!ATTLIST PUBLISH_RULE_ID type (CHAR | NUMBER | DATE | TIME | DATETIME)⇒
#IMPLIED>
   <!ELEMENT MSGNODENAME (#PCDATA)>
      <!ATTLIST MSGNODENAME type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
<!ELEMENT RT_TYPE_TBL (RT_TYPE, DESCR, DESCRSHORT)>
   <!ATTLIST RT_TYPE_TBL class (R | SR) #REQUIRED>

<!ELEMENT RT_TYPE (#PCDATA)>
      <!ATTLIST RT_TYPE type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT DESCR (#PCDATA)>
      <!ATTLIST DESCR type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT DESCRSHORT (#PCDATA)>
      <!ATTLIST DESCRSHORT type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>

<!ELEMENT MsgData (Transaction)>
<!ELEMENT Transaction (RT_TYPE_TBL, PSCAMA)>
```

## Example: XML Schema

The following code shows MARKET_RATE_TYPE_FULLSYNC in XML schema format:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">

<xsd:element name="MARKET_RATE_TYPE_FULLSYNC" type="MARKET_RATE_TYPE_FULLSYNCType"⇒
/>

<xsd:complexType name="MARKET_RATE_TYPE_FULLSYNCType">
<xsd:sequence>
   <xsd:element name="FieldTypes" type="FieldTypesType"/>
   <xsd:element name="MsgData" type="MsgDataType"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldTypesType">
<xsd:sequence>
   <xsd:element name="RT_TYPE_TBL" type="FieldTypesRT_TYPE_TBLType"/>
   <xsd:element name="PSCAMA" type="PSCAMA"/>
</xsd:sequence>
```

```
</xsd:complexType>

<xsd:complexType name="FieldTypesRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE" type="FieldTypesFieldType"/>
   <xsd:element name="DESCR" type="FieldTypesFieldType"/>
   <xsd:element name="DESCRSHORT" type="FieldTypesFieldType"/>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="PSCAMA">
   <xsd:sequence>
   <xsd:element name="LANGUAGE_CD" type="LANGUAGE_CDType" minOccurs="0" maxOccurs=⇒
"1"/>
   <xsd:element name="AUDIT_ACTN" type="AUDIT_ACTNType"/>
   <xsd:element name="BASE_LANGUAGE_CD" type="BASE_LANGUAGE_CDType" minOccurs="0"⇒
 maxOccurs="1"/>
   <xsd:element name="MSG_SEQ_FLG" type="MSG_SEQ_FLGType" minOccurs="0" maxOccurs=⇒
"1"/>
   <xsd:element name="PROCESS_INSTANCE" type="PROCESS_INSTANCEType" minOccurs="0"⇒
 maxOccurs="1"/>
   <xsd:element name="PUBLISH_RULE_ID" type="PUBLISH_RULE_IDType" minOccurs="0"⇒
 maxOccurs="1"/>
   <xsd:element name="MSGNODENAME" type="MSGNODENAMEType" minOccurs="0" maxOccurs=⇒
"1"/>
   </xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="LANGUAGE_CDType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="AUDIT_ACTNType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="BASE_LANGUAGE_CDType"  >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="MSG_SEQ_FLGType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="PROCESS_INSTANCEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>
```

```
<xsd:complexType name="PUBLISH_RULE_IDType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="MSGNODENAMEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>


<xsd:complexType name="MsgDataType">
      <xsd:sequence>
   <xsd:element name="Transaction">
      <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="RT_TYPE_TBL" type="MsgDataRT_TYPE_TBLType"/>
         <xsd:element name="PSCAMA" type="PSCAMA"/>
      </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
      </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MsgDataRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE">
<xsd:simpleType>
<xsd:restriction base="xsd:string" >
   <xsd:pattern value="[A-Z]{1-5}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCR">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-30}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCRSHORT">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-10}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="FieldTypesFieldType">
```

```
   <xsd:attribute name="type" type="fieldtypes"/>
</xsd:complexType>

<xsd:simpleType name="fieldtypes">
<xsd:restriction base="xsd:string">
   <xsd:enumeration value="CHAR"/>
   <xsd:enumeration value="NUMBER"/>
   <xsd:enumeration value="DATE"/>
   <xsd:enumeration value="DATETIME"/>
   <xsd:enumeration value="TIME"/>
</xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

# APPENDIX A

# EIP Naming Standards

EIP names describe the business purposes of integration points and provide a virtual wrapper for packaging one or more integration technology objects.

EIP names use an approved list of standard PeopleSoft business object nouns and action and event verbs.

## Standard Action and Event Verbs

The following standard verbs are used in the names and descriptions of EIPs:

| Verb | Description |
|---|---|
| Acknowledge | Indicates receipt of a processing request. Also conveys the result of the original request. (For example, your application acknowledges a purchase order (PO) when a PO has been issued and the corresponding business application acknowledges the receipt of the PO and responds with an acceptance or counter offer.) |
| Add | Use when a complete entity has already been constructed and needs to be communicated for the first time to another business application, and when there are business implications beyond what a Sync message would convey. Transactional messages may use Add; setup data should use Sync. |
| Adjust | Clarifies a specific process (for example, adjustment of inventory quantity on hand when neither Add, Change, or Delete conveys the full meaning). |
| Allocate | Clarifies a specific process (for example, allocating costs to different business applications when neither Add, Change, or Delete conveys the full meaning). |
| Approve | Use when an entity passes an approval process and is ready for the next business process step. |

| Verb | Description |
|---|---|
| Cancel | Use when there are business implications beyond a simple change or delete (for example, canceling a purchase order). |
| Change | Use when an entity is changed based on a business event and the change needs to be communicated to another business application. Encompasses business implications beyond what a Sync message would convey (for example, SalesOrderChange). Transactional messages can use Change; setup data should use Sync. |
| Confirm | Responds to a request from the receiving application to confirm. This function conveys the result of the original request (for example, when an inventory issue must be confirmed in an application based on an event in a warehouse management business application). |
| Create | Use when the processing must initiate the building of the document rather than moving the document from one system to another. |
| Find | Requests a list of items from a business application. The response to this request is List. Equivalent to a search dialog box in which you pass the search criteria over as the message. |
| FullSync | Replicates a complete entity, including all record instances, between business applications to initially seed the receiving application with that data. Use with all full message definitions. |
| Get | Requests a specific data entity from a business application. The response to this request is Show. Differs from Find in that the specific entity's key values are known and its details are being requested, whereas Find checks for existence and returns a list of values that match the Find criteria. |
| Issue | Clarifies a specific process (for example, issue material from inventory in cases where neither Add, Change, or Delete conveys the full meaning). |

| Verb | Description |
|---|---|
| List | Use when sending a list of multiple data entities in a summary format. The List verb can be used to respond to a Find or Get request, or in a publish scenario, to push information to other applications based on a business event. The results of a List can be used as is, or they can be used to select a specific instance of a document or entity to issue a detail Get request. |
| Load | Initiates the addition of a data entity to another business application where maintenance to the document passes to the receiving application permanently. When the request is passed, the sending application no longer has direct control over the document or entity. |
| Post | Clarifies a specific process (for example, a posted journal entry, in cases where neither Add, Change, or Delete conveys the full meaning). |
| Receive | Clarifies a specific process (for example, when you receive inventory against a PO, in cases where Change is not detailed enough for the business context). |
| Request | Requests specific data from a business application. The requested data should be passed back as a separate message. |
| Show | Sends information about a specific instance of a business entity. Can also be used to respond to a Get request, or in a publish scenario in which it pushes information to other applications based on a business event. |
| Sync | Communicates the need to update master files between business applications. Facilitates application integrity and ease of data entry for the business user by enabling a single point of input. Should contain only incremental messages of Add, Change, or Delete actions to the entity. Normally used for passing setup messages between applications. |

| Verb | Description |
|---|---|
| Transfer | Clarifies a specific process (for example, in the event of a transfer of material from one inventory location to another, in cases where neither Add, Change, or Delete conveys the full meaning). |
| Update | Clarifies a specific process (for example, in the event of an update of inspection information from one business application to another, in cases where neither Add, Change, or Delete conveys the full meaning). The event is not adding a document or changing fields, but communicating the occurrence of an event as well as the corresponding data that accompanies the event. Transactional data may use Update; setup data should use Sync. |

# Standard Business Object Nouns

Business objects appear in uppercase with underscores between key words.

The following table lists examples of business object names only; it is not intended to list or set conventions.

| Account_Chartfield | Deal | Names_Prefix_Suffix | Project_Category |
|---|---|---|---|
| Action_Reason | Department | Nations_Duevo | Project_Status |
| APE_Industry | Dept_Budget | Occupation_Illness | Project_Team |
| Applicant | Detail_Calendar | Par_Location | Project_Type_Cat_ Link |
| Bank | DirectDeposit | Par_Location_Count | Purchase_Order |
| BOM | Earnings | Payroll | Rating_Model |
| Budget | Expense | Payroll_Paysheet | Regulatory_Region |
| Bus_Unit_FS | Expense_Advance | Pension_Fund | Resource_Category |
| Bus_Unit_GL | Expense_Report | Person_ Accomplishment | Resource_SubCategory |
| Bus_Unit_HR | Expense_Sheet | Person_Competency | Resource_Type |

| Account_Chartfield | Deal | Names_Prefix_Suffix | Project_Category |
|---|---|---|---|
| Bus_Unit_PC | Fund | Person_Contract_ Belgium | Review_Scale |
| Bus_Unit_PF | Grant | Person_Credit_Card | Salary |
| Carrier | Image | Person_Disability | Salary_Matrix |
| Company | Industry_Inspection | Person_Diversity | Salary_Plan |
| Company_Credit_ Card | InterUnit | Person_Education | Salary_Structure |
| Company_Property | Inventory | Person_Names | SalesOrder |
| Competency | Item | Person_Property | Schedule |
| Consumer | Item_BusUnit | Person_PriorWork | State |
| Consumer_Usage | Item_Rev | Person_Visa_Citizen | State_Name |
| Contract | Item_Vendor | Position | Statute |
| ContractBelgium | Job_Code | Product_Chartfield | Statute_Belgium |
| Country | Journal | Product_Item | Vendor |
| Customer | Labor_Category | Product_Group | Visa_Permit |
| Credit_Card | Labor_Relations | Product_Price | Voucher |
| Credit_Card_ Merchant | Location | Product_UOM | Union |
| Currency | Market_Price | Project | UOM |
| Customer | Market_Rate | Project_Activity | Workforce |

# APPENDIX B

# PeopleSoft Design Patterns

PeopleSoft delivers a group of design patterns with each application. The Design Patterns page lists available design patterns.

## List of Design Patterns

The following table presents brief descriptions of the delivered design patterns:

| Design Pattern Name | Description |
|---|---|
| AE Row By Row Publish (application engine row by row publish) | In this design pattern, the transaction or setup data that you want to send out of PeopleSoft is updated by using an Application Engine program that performs procedural (row-by-row) processing; you want to publish these changes. Generally, messages are used with this design pattern. |
| Batch Publish | Use this design pattern to publish messages from a batch application. The batch application can be a COBOL or Structured Query Report program that takes either a procedural or set-based approach, or it can be an Application Engine set-based program. |
| Batch Subscribe | This design pattern enables you to perform edits against messages in sets. This can be a useful technique for high volume data, including millions of inbound rows. This design pattern is useful when you know that a single message definition may contain multiple instances of a transaction, or when you must reuse an existing batch program. |
| CI Subscribe (component interface subscribe) | This design pattern uses a component interface to edit incoming message data. This enables you to reuse existing business rules when processing data. |

| Design Pattern Name | Description |
|---|---|
| Component Publish | In this design pattern, the transaction or setup data that you want to send out of PeopleSoft is being updated by using a PeopleSoft component. In this case, the data is already in the component buffer, and the Publish PeopleCode function is used to publish a message. |
| EDI In | This design pattern is for inbound EDI documents.<br><br>You should only use EDI for existing EDI manager inbound transactions that must be supplied to an EDI partner, and you want to allow subscription to an XML message, or when you need to comply with other industry standards, such as SWIFT, BAI, or HL7, that have an existing EDI manager inbound map, and you want to convert to subscribing to an XML message. |
| EDI Out | This design pattern is for outbound EDI documents.<br><br>You should only use EDI for existing EDI manager inbound transactions that must be supplied to an EDI partner and you want to allow subscription to an XML message, or you need to comply with other industry standards, such as SWIFT, BAI, or HL7, that have an existing EDI manager inbound map, and you want to convert to publishing to an XML message. |
| Full Table Publish | Use this design pattern to populate an entire copy of a table onto a remote database or legacy system. Generally, full data replication occurs with setup tables, or relatively static, low-volume tables that are keyed by setID. When a copy of a table exists on the remote system, incremental updates can be used. |
| Full Table Subscribe | Use this design pattern to subscribe to messages that contain an entire copy of a table that is published from a remote database or legacy system. Generally, full data replication occurs with setup tables, or relatively static, low-volume tables that are keyed by setID. When a copy of a table exists on the remote system, incremental updates can be used. |
| No Pattern | No design pattern is specified. |
| PeopleCode Subscribe | Use this design pattern to subscribe to messages by using a PeopleCode program when additional processing is required. Use PeopleCode subscription when simple edits or no edits against the inbound data are needed before inserting the data into the application tables or staging tables. |

| Design Pattern Name | Description |
| --- | --- |
| Sync Reply | In this design pattern, another system initiates a request for information from PeopleSoft and waits for information to be returned. This information must be provided by PeopleSoft in a real-time synchronous mode and in a conversational style of interface before the other system can continue processing. Generally, business interlinks are used to satisfy this type of request. |
| Sync Request | Use this design pattern when a PeopleSoft application must call a third-party vendor's application to request information. This information must be provided in a real-time, synchronous mode. Generally, business interlinks are used to satisfy this type of request. |
| XML Reply | In this design pattern, another system initiates a request for information from PeopleSoft. This information must be provided by PeopleSoft in a real-time synchronous mode and in a conversational style of interface before the other system can continue processing. Generally, XMLDocs are used to satisfy this type of request. |
| XML Request | Use this design pattern when a PeopleSoft application must call a third party vendor's application to request information. This information must be provided in a real-time, synchronous mode. Generally, XMLDocs are used to satisfy this type of request. |

# Glossary of PeopleSoft Terms

**absence entitlement** This element defines rules for granting paid time off for valid absences, such as sick time, vacation, and maternity leave. An absence entitlement element defines the entitlement amount, frequency, and entitlement period.

**absence take** This element defines the conditions that must be met before a payee is entitled to take paid time off.

**accounting class** In PeopleSoft Enterprise Performance Management, the accounting class defines how a resource is treated for generally accepted accounting practices. The Inventory class indicates whether a resource becomes part of a balance sheet account, such as inventory or fixed assets, while the Non-inventory class indicates that the resource is treated as an expense of the period during which it occurs.

**accounting date** The accounting date indicates when a transaction is recognized, as opposed to the date the transaction actually occurred. The accounting date and transaction date can be the same. The accounting date determines the period in the general ledger to which the transaction is to be posted. You can only select an accounting date that falls within an open period in the ledger to which you are posting. The accounting date for an item is normally the invoice date.

**accounting split** The accounting split method indicates how expenses are allocated or divided among one or more sets of accounting ChartFields.

**accumulator** You use an accumulator to store cumulative values of defined items as they are processed. You can accumulate a single value over time or multiple values over time. For example, an accumulator could consist of all voluntary deductions, or all company deductions, enabling you to accumulate amounts. It allows total flexibility for time periods and values accumulated.

**action reason** The reason an employee's job or employment information is updated. The action reason is entered in two parts: a personnel action, such as a promotion, termination, or change from one pay group to another—and a reason for that action. Action reasons are used by PeopleSoft Human Resources, PeopleSoft Benefits Administration, PeopleSoft Stock Administration, and the COBRA Administration feature of the Base Benefits business process.

**action template** In PeopleSoft Receivables, outlines a set of escalating actions that the system or user performs based on the period of time that a customer or item has been in an action plan for a specific condition.

**activity** In PeopleSoft Enterprise Learning Management, an instance of a catalog item (sometimes called a class) that is available for enrollment. The activity defines such things as the costs that are associated with the offering, enrollment limits and deadlines, and waitlisting capacities.

In PeopleSoft Enterprise Performance Management, the work of an organization and the aggregation of actions that are used for activity-based costing.

In PeopleSoft Project Costing, the unit of work that provides a further breakdown of projects—usually into specific tasks.

In PeopleSoft Workflow, a specific transaction that you might need to perform in a business process. Because it consists of the steps that are used to perform a transaction, it is also known as a step map.

**agreement** In PeopleSoft eSettlements, provides a way to group and specify processing options, such as payment terms, pay from a bank, and notifications by a buyer and supplier location combination.

**allocation rule** In PeopleSoft Enterprise Incentive Management, an expression within compensation plans that enables the system to assign transactions to nodes and participants. During transaction allocation, the allocation engine traverses the compensation structure from the current node to the root node, checking each node for plans that contain allocation rules.

**alternate account** A feature in PeopleSoft General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level, as required for recording and reporting by some national governments.

**AR specialist** Abbreviation for *receivables specialist.* In PeopleSoft Receivables, an individual in who tracks and resolves deductions and disputed items.

**arbitration plan** In PeopleSoft Enterprise Pricer, defines how price rules are to be applied to the base price when the transaction is priced.

**assessment rule** In PeopleSoft Receivables, a user-defined rule that the system uses to evaluate the condition of a customer's account or of individual items to determine whether to generate a follow-up action.

**asset class** An asset group used for reporting purposes. It can be used in conjunction with the asset category to refine asset classification.

**attribute/value pair** In PeopleSoft Directory Interface, relates the data that makes up an entry in the directory information tree.

**authentication server** A server that is set up to verify users of the system.

**base time period** In PeopleSoft Business Planning, the lowest level time period in a calendar.

**benchmark job** In PeopleSoft Workforce Analytics, a benchmark job is a job code for which there is corresponding salary survey data from published, third-party sources.

**book** In PeopleSoft Asset Management, used for storing financial and tax information, such as costs, depreciation attributes, and retirement information on assets.

**branch** A tree node that rolls up to nodes above it in the hierarchy, as defined in PeopleSoft Tree Manager.

**budgetary account only** An account used by the system only and not by users; this type of account does not accept transactions. You can only budget with this account. Formerly called "system-maintained account."

**budget check** In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.

**budget control** In commitment control, budget control ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and terminate a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.

**budget period** The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.

**business event** In PeopleSoft Receivables, defines the processing characteristics for the Receivable Update process for a draft activity.

| | |
|---|---|
| | In PeopleSoft Sales Incentive Management, an original business transaction or activity that may justify the creation of a PeopleSoft Enterprise Incentive Management event (a sale, for example). |
| **business unit** | A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions. |
| **buyer** | In PeopleSoft eSettlements, an organization (or business unit, as opposed to an individual) that transacts with suppliers (vendors) within the system. A buyer creates payments for purchases that are made in the system. |
| **catalog item** | In PeopleSoft Enterprise Learning Management, a specific topic that a learner can study and have tracked. For example, “Introduction to Microsoft Word.” A catalog item contains general information about the topic and includes a course code, description, categorization, keywords, and delivery methods. A catalog item can have one or more learning activities. |
| **catalog map** | In PeopleSoft Catalog Management, translates values from the catalog source data to the format of the company’s catalog. |
| **catalog partner** | In PeopleSoft Catalog Management, shares responsibility with the enterprise catalog manager for maintaining catalog content. |
| **categorization** | Associates partner offerings with catalog offerings and groups them into enterprise catalog categories. |
| **channel** | In PeopleSoft MultiChannel Framework, email, chat, voice (computer telephone integration [CTI]), or a generic event. |
| **ChartField** | A field that stores a chart of accounts, resources, and so on, depending on the PeopleSoft application. ChartField values represent individual account numbers, department codes, and so forth. |
| **ChartField balancing** | You can require specific ChartFields to match up (balance) on the debit and the credit side of a transaction. |
| **ChartField combination edit** | The process of editing journal lines for valid ChartField combinations based on user-defined rules. |
| **ChartKey** | One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination. |
| **checkbook** | In PeopleSoft Promotions Management, enables you to view financial data (such as planned, incurred, and actual amounts) that is related to funds and trade promotions. |
| **Class ChartField** | A ChartField value that identifies a unique appropriation budget key when you combine it with a fund, department ID, and program code, as well as a budget period. Formerly called *sub-classification.* |
| **clone** | In PeopleCode, to make a unique copy. In contrast, to *copy* may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change. |
| **collection** | To make a set of documents available for searching in Verity, you must first create at least one collection. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents that match search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Because a collection can only store information for a single location, PeopleSoft maintains a set of collections (one per language code) for each search index object. |

**collection rule** In PeopleSoft Receivables, a user-defined rule that defines actions to take for a customer based on both the amount and the number of days past due for outstanding balances.

**compensation object** In PeopleSoft Enterprise Incentive Management, a node within a compensation structure. Compensation objects are the building blocks that make up a compensation structure's hierarchical representation.

**compensation structure** In PeopleSoft Enterprise Incentive Management, a hierarchical relationship of compensation objects that represents the compensation-related relationship between the objects.

**condition** In PeopleSoft Receivables, occurs when there is a change of status for a customer's account, such as reaching a credit limit or exceeding a user-defined balance due.

**configuration parameter catalog** Used to configure an external system with PeopleSoft. For example, a configuration parameter catalog might set up configuration and communication parameters for an external server.

**configuration plan** In PeopleSoft Enterprise Incentive Management, configuration plans hold allocation information for common variables (not incentive rules) and are attached to a node without a participant. Configuration plans are not processed by transactions.

**content reference** Content references are pointers to content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three categories: target content, templates, and template pagelets.

**context** In PeopleCode, determines which buffer fields can be contextually referenced and which is the current row of data on each scroll level when a PeopleCode program is running.

In PeopleSoft Enterprise Incentive Management, a mechanism that is used to determine the scope of a processing run. PeopleSoft Enterprise Incentive Management uses three types of context: plan, period, and run-level.

**control table** Stores information that controls the processing of an application. This type of processing might be consistent throughout an organization, or it might be used only by portions of the organization for more limited sharing of data.

**cost profile** A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.

**cost row** A cost transaction and amount for a set of ChartFields.

**current learning** In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's in-progress learning activities and programs.

**data acquisition** In PeopleSoft Enterprise Incentive Management, the process during which raw business transactions are acquired from external source systems and fed into the operational data store (ODS).

**data elements** Data elements, at their simplest level, define a subset of data and the rules by which to group them.

For Workforce Analytics, data elements are rules that tell the system what measures to retrieve about your workforce groups.

**dataset** A data grouping that enables role-based filtering and distribution of data. You can limit the range and quantity of data that is displayed for a user by associating dataset rules with user roles. The result of dataset rules is a set of data that is appropriate for the user's roles.

**delivery method**

In PeopleSoft Enterprise Learning Management, identifies the primary type of delivery method in which a particular learning activity is offered. Also provides default values for the learning activity, such as cost and language. This is primarily used to help learners search the catalog for the type of delivery from which they learn best. Because PeopleSoft Enterprise Learning Management is a blended learning system, it does not enforce the delivery method.

In PeopleSoft Supply Chain Management, identifies the method by which goods are shipped to their destinations (such as truck, air, rail, and so on). The delivery method is specified when creating shipment schedules.

**delivery method type**

In PeopleSoft Enterprise Learning Management, identifies how learning activities can be delivered—for example, through online learning, classroom instruction, seminars, books, and so forth—in an organization. The type determines whether the delivery method includes scheduled components.

**directory information tree**

In PeopleSoft Directory Interface, the representation of a directory's hierarchical structure.

**document sequencing**

A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in the system for statutory reporting and for tracking commercial transaction activity.

**dynamic detail tree**

A tree that takes its detail values—dynamic details—directly from a table in the database, rather than from a range of values that are entered by the user.

**edit table**

A table in the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft application, they can be validated against an edit table to ensure data integrity throughout the system.

**effective date**

A method of dating information in PeopleSoft applications. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. By using effective dates, you don't delete values; you enter a new value with a current effective date.

**EIM ledger**

Abbreviation for *Enterprise Incentive Management ledger.* In PeopleSoft Enterprise Incentive Management, an object to handle incremental result gathering within the scope of a participant. The ledger captures a result set with all of the appropriate traces to the data origin and to the processing steps of which it is a result.

**elimination set**

In PeopleSoft General Ledger, a related group of intercompany accounts that is processed during consolidations.

**entry event**

In PeopleSoft General Ledger, Receivables, Payables, Purchasing, and Billing, a business process that generates multiple debits and credits resulting from single transactions to produce standard, supplemental accounting entries.

**equitization**

In PeopleSoft General Ledger, a business process that enables parent companies to calculate the net income of subsidiaries on a monthly basis and adjust that amount to increase the investment amount and equity income amount before performing consolidations.

**event**

A predefined point either in the Component Processor flow or in the program flow. As each point is encountered, the event activates each component, triggering any PeopleCode program that is associated with that component and that event. Examples of events are FieldChange, SavePreChange, and RowDelete.

In PeopleSoft Human Resources, also refers to an incident that affects benefits eligibility.

**event propagation process**

In PeopleSoft Sales Incentive Management, a process that determines, through logic, the propagation of an original PeopleSoft Enterprise Incentive Management event and creates a derivative (duplicate) of the original event to be processed by other objects.

| | |
|---|---|
| | Sales Incentive Management uses this mechanism to implement splits, roll-ups, and so on. Event propagation determines who receives the credit. |
| **exception** | In PeopleSoft Receivables, an item that either is a deduction or is in dispute. |
| **exclusive pricing** | In PeopleSoft Order Management, a type of arbitration plan that is associated with a price rule. Exclusive pricing is used to price sales order transactions. |
| **fact** | In PeopleSoft applications, facts are numeric data values from fields from a source database as well as an analytic application. A fact can be anything you want to measure your business by, for example, revenue, actual, budget data, or sales numbers. A fact is stored on a fact table. |
| **forecast item** | A logical entity with a unique set of descriptive demand and forecast data that is used as the basis to forecast demand. You create forecast items for a wide range of uses, but they ultimately represent things that you buy, sell, or use in your organization and for which you require a predictable usage. |
| **fund** | In PeopleSoft Promotions Management, a budget that can be used to fund promotional activity. There are four funding methods: top down, fixed accrual, rolling accrual, and zero-based accrual. |
| **generic process type** | In PeopleSoft Process Scheduler, process types are identified by a generic process type. For example, the generic process type SQR includes all SQR process types, such as SQR process and SQR report. |
| **group** | In PeopleSoft Billing and Receivables, a posting entity that comprises one or more transactions (items, deposits, payments, transfers, matches, or write-offs).<br><br>In PeopleSoft Human Resources Management and Supply Chain Management, any set of records that are associated under a single name or variable to run calculations in PeopleSoft business processes. In PeopleSoft Time and Labor, for example, employees are placed in groups for time reporting purposes. |
| **incentive object** | In PeopleSoft Enterprise Incentive Management, the incentive-related objects that define and support the PeopleSoft Enterprise Incentive Management calculation process and results, such as plan templates, plans, results data, user interaction objects, and so on. |
| **incentive rule** | In PeopleSoft Sales Incentive Management, the commands that act on transactions and turn them into compensation. A rule is one part in the process of turning a transaction into compensation. |
| **incur** | In PeopleSoft Promotions Management, to become liable for a promotional payment. In other words, you owe that amount to a customer for promotional activities. |
| **item** | In PeopleSoft Inventory, a tangible commodity that is stored in a business unit (shipped from a warehouse).<br><br>In PeopleSoft Demand Planning, Inventory Policy Planning, and Supply Planning, a noninventory item that is designated as being used for planning purposes only. It can represent a family or group of inventory items. It can have a planning bill of material (BOM) or planning routing, and it can exist as a component on a planning BOM. A planning item cannot be specified on a production or engineering BOM or routing, and it cannot be used as a component in a production. The quantity on hand will never be maintained.<br><br>In PeopleSoft Receivables, an individual receivable. An item can be an invoice, a credit memo, a debit memo, a write-off, or an adjustment. |
| **KPI** | An abbreviation for *key performance indicator.* A high-level measurement of how well an organization is doing in achieving critical success factors. This defines the data value or calculation upon which an assessment is determined. |

**LDIF file** Abbreviation for *Lightweight Directory Access Protocol (LDAP) Data Interchange Format file.* Contains discrepancies between PeopleSoft data and directory data.

**learner group** In PeopleSoft Enterprise Learning Management, a group of learners who are linked to the same learning environment. Members of the learner group can share the same attributes, such as the same department or job code. Learner groups are used to control access to and enrollment in learning activities and programs. They are also used to perform group enrollments and mass enrollments in the back office.

**learning components** In PeopleSoft Enterprise Learning Management, the foundational building blocks of learning activities. PeopleSoft Enterprise Learning Management supports six basic types of learning components: web-based, session, webcast, test, survey, and assignment. One or more of these learning component types compose a single learning activity.

**learning environment** In PeopleSoft Enterprise Learning Management, identifies a set of categories and catalog items that can be made available to learner groups. Also defines the default values that are assigned to the learning activities and programs that are created within a particular learning environment. Learning environments provide a way to partition the catalog so that learners see only those items that are relevant to them.

**learning history** In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's completed learning activities and programs.

**ledger mapping** You use ledger mapping to relate expense data from general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as *rates*) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In PeopleSoft Enterprise Warehouse, you can map general ledger accounts to the EW Ledger table.

**library section** In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan (or template) and that is available for other plans to share. Changes to a library section are reflected in all plans that use it.

**linked section** In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan template but appears in a plan. Changes to linked sections propagate to plans using that section.

**linked variable** In PeopleSoft Enterprise Incentive Management, a variable that is defined and maintained in a plan template and that also appears in a plan. Changes to linked variables propagate to plans using that variable.

**load** In PeopleSoft Inventory, identifies a group of goods that are shipped together. Load management is a feature of PeopleSoft Inventory that is used to track the weight, the volume, and the destination of a shipment.

**local functionality** In PeopleSoft HRMS, the set of information that is available for a specific country. You can access this information when you click the appropriate country flag in the global window, or when you access it by a local country menu.

**location** Locations enable you to indicate the different types of addresses—for a company, for example, one address to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each address has a different location number. The primary location—indicated by a *1*—is the address you use most often and may be different from the main address.

**logistical task** In PeopleSoft Services Procurement, an administrative task that is related to hiring a service provider. Logistical tasks are linked to the service type on the work order so that different types of services can have different logistical tasks. Logistical tasks include both preapproval tasks (such as assigning a new badge or ordering a new

| | |
|---|---|
| | laptop) and postapproval tasks (such as scheduling orientation or setting up the service provider email). The logistical tasks can be mandatory or optional. Mandatory preapproval tasks must be completed before the work order is approved. Mandatory postapproval tasks, on the other hand, must be completed before a work order is released to a service provider. |
| **market template** | In PeopleSoft Enterprise Incentive Management, additional functionality that is specific to a given market or industry and is built on top of a product category. |
| **match group** | In PeopleSoft Receivables, a group of receivables items and matching offset items. The system creates match groups by using user-defined matching criteria for selected field values. |
| **MCF server** | Abbreviation for *PeopleSoft MultiChannel Framework server.* Comprises the universal queue server and the MCF log server. Both processes are started when *MCF Servers* is selected in an application server domain configuration. |
| **merchandising activity** | In PeopleSoft Promotions Management, a specific discount type that is associated with a trade promotion (such as off-invoice, billback or rebate, or lump-sum payment) that defines the performance that is required to receive the discount. In the industry, you may know this as an offer, a discount, a merchandising event, an event, or a tactic. |
| **meta-SQL** | Meta-SQL constructs expand into platform-specific Structured Query Language (SQL) substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, and PeopleSoft Application Engine programs. |
| **metastring** | Metastrings are special expressions included in SQL string literals. The metastrings, prefixed with a percent (%) symbol, are included directly in the string literals. They expand at run time into an appropriate substring for the current database platform. |
| **multibook** | In PeopleSoft General Ledger, multiple ledgers having multiple-base currencies that are defined for a business unit, with the option to post a single transaction to all base currencies (all ledgers) or to only one of those base currencies (ledgers). |
| **multicurrency** | The ability to process transactions in a currency other than the business unit's base currency. |
| **national allowance** | In PeopleSoft Promotions Management, a promotion at the corporate level that is funded by nondiscretionary dollars. In the industry, you may know this as a national promotion, a corporate promotion, or a corporate discount. |
| **node-oriented tree** | A tree that is based on a detail structure, but the detail values are not used. |
| **pagelet** | Each block of content on the home page is called a pagelet. These pagelets display summary information within a small rectangular area on the page. The pagelet provide users with a snapshot of their most relevant PeopleSoft and non-PeopleSoft content. |
| **participant** | In PeopleSoft Enterprise Incentive Management, participants are recipients of the incentive compensation calculation process. |
| **participant object** | Each participant object may be related to one or more compensation objects.<br>See also *compensation object.* |
| **partner** | A company that supplies products or services that are resold or purchased by the enterprise. |
| **pay cycle** | In PeopleSoft Payables, a set of rules that define the criteria by which it should select scheduled payments for payment creation. |
| **pending item** | In PeopleSoft Receivables, an individual receivable (such as an invoice, a credit memo, or a write-off) that has been entered in or created by the system, but hasn't been posted. |

**PeopleCode** PeopleCode is a proprietary language, executed by the PeopleSoft application processor. PeopleCode generates results based upon existing data or user actions. By using business interlink objects, external services are available to all PeopleSoft applications wherever PeopleCode can be executed.

**PeopleCode event** An action that a user takes upon an object, usually a record field, that is referenced within a PeopleSoft page.

**PeopleSoft Internet Architecture** The fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of a relational database management system (RDBMS), an application server, a web server, and a browser.

**performance measurement** In PeopleSoft Enterprise Incentive Management, a variable used to store data (similar to an aggregator, but without a predefined formula) within the scope of an incentive plan. Performance measures are associated with a plan calendar, territory, and participant. Performance measurements are used for quota calculation and reporting.

**period context** In PeopleSoft Enterprise Incentive Management, because a participant typically uses the same compensation plan for multiple periods, the period context associates a plan context with a specific calendar period and fiscal year. The period context references the associated plan context, thus forming a chain. Each plan context has a corresponding set of period contexts.

**plan** In PeopleSoft Sales Incentive Management, a collection of allocation rules, variables, steps, sections, and incentive rules that instruct the PeopleSoft Enterprise Incentive Management engine in how to process transactions.

**plan context** In PeopleSoft Enterprise Incentive Management, correlates a participant with the compensation plan and node to which the participant is assigned, enabling the PeopleSoft Enterprise Incentive Management system to find anything that is associated with the node and that is required to perform compensation processing. Each participant, node, and plan combination represents a unique plan context—if three participants are on a compensation structure, each has a different plan context. Configuration plans are identified by plan contexts and are associated with the participants that refer to them.

**plan template** In PeopleSoft Enterprise Incentive Management, the base from which a plan is created. A plan template contains common sections and variables that are inherited by all plans that are created from the template. A template may contain steps and sections that are not visible in the plan definition.

**planned learning** In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's planned learning activities and programs.

**planning instance** In PeopleSoft Supply Planning, a set of data (business units, items, supplies, and demands) constituting the inputs and outputs of a supply plan.

**portal registry** In PeopleSoft applications, the portal registry is a tree-like structure in which content references are organized, classified, and registered. It is a central repository that defines both the structure and content of a portal through a hierarchical, tree-like structure of folders useful for organizing and securing content references.

**price list** In PeopleSoft Enterprise Pricer, enables you to select products and conditions for which the price list applies to a transaction. During a transaction, the system either determines the product price based on the predefined search hierarchy for the transaction or uses the product's lowest price on any associated, active price lists. This price is used as the basis for any further discounts and surcharges.

**price rule** In PeopleSoft Enterprise Pricer, defines the conditions that must be met for adjustments to be applied to the base price. Multiple rules can apply when conditions of each rule are met.

**price rule condition** In PeopleSoft Enterprise Pricer, selects the price-by fields, the values for the price-by fields, and the operator that determines how the price-by fields are related to the transaction.

**price rule key** In PeopleSoft Enterprise Pricer, defines the fields that are available to define price rule conditions (which are used to match a transaction) on the price rule.

**process category** In PeopleSoft Process Scheduler, processes that are grouped for server load balancing and prioritization.

**process group** In PeopleSoft Financials, a group of application processes (performed in a defined order) that users can initiate in real time, directly from a transaction entry page.

**process definition** Process definitions define each run request.

**process instance** A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.

**process job** You can link process definitions into a job request and process each request serially or in parallel. You can also initiate subsequent processes based on the return code from each prior request.

**process request** A single run request, such as a Structured Query Report (SQR), a COBOL or Application Engine program, or a Crystal report that you run through PeopleSoft Process Scheduler.

**process run control** A PeopleTools variable used to retain PeopleSoft Process Scheduler values needed at runtime for all requests that reference a run control ID. Do not confuse these with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.

**product category** In PeopleSoft Enterprise Incentive Management, indicates an application in the Enterprise Incentive Management suite of products. Each transaction in the PeopleSoft Enterprise Incentive Management system is associated with a product category.

**programs** In PeopleSoft Enterprise Learning Management, a high-level grouping that guides the learner along a specific learning path through sections of catalog items. PeopleSoft Enterprise Learning Systems provides two types of programs—curricula and certifications.

**progress log** In PeopleSoft Services Procurement, tracks deliverable-based projects. This is similar to the time sheet in function and process. The service provider contact uses the progress log to record and submit progress on deliverables. The progress can be logged by the activity that is performed, by the percentage of work that is completed, or by the completion of milestone activities that are defined for the project.

**project transaction** In PeopleSoft Project Costing, an individual transaction line that represents a cost, time, budget, or other transaction row.

**promotion** In PeopleSoft Promotions Management, a trade promotion, which is typically funded from trade dollars and used by consumer products manufacturers to increase sales volume.

**publishing** In PeopleSoft Enterprise Incentive Management, a stage in processing that makes incentive-related results available to participants.

**record group** A set of logically and functionally related control tables and views. Record groups help enable TableSet sharing, which eliminates redundant data entry. Record groups ensure that TableSet sharing is applied consistently across all related tables and views.

**record input VAT flag** Abbreviation for *record input value-added tax flag*. Within PeopleSoft Purchasing, Payables, and General Ledger, this flag indicates that you are recording input VAT

| | |
|---|---|
| | on the transaction. This flag, in conjunction with the record output VAT flag, is used to determine the accounting entries created for a transaction and to determine how a transaction is reported on the VAT return. For all cases within Purchasing and Payables where VAT information is tracked on a transaction, this flag is set to Yes. This flag is not used in PeopleSoft Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in PeopleSoft Expenses, where it is assumed that you are always recording only input VAT. |
| **record output VAT flag** | Abbreviation for *record output value-added tax flag*.<br><br>See *record input VAT flag*. |
| **reference data** | In PeopleSoft Sales Incentive Management, system objects that represent the sales organization, such as territories, participants, products, customers, channels, and so on. |
| **reference object** | In PeopleSoft Enterprise Incentive Management, this dimension-type object further defines the business. Reference objects can have their own hierarchy (for example, product tree, customer tree, industry tree, and geography tree). |
| **reference transaction** | In commitment control, a reference transaction is a source transaction that is referenced by a higher-level (and usually later) source transaction, in order to automatically reverse all or part of the referenced transaction's budget-checked amount. This avoids duplicate postings during the sequential entry of the transaction at different commitment levels. For example, the amount of an encumbrance transaction (such as a purchase order) will, when checked and recorded against a budget, cause the system to concurrently reference and relieve all or part of the amount of a corresponding pre-encumbrance transaction, such as a purchase requisition. |
| **regional sourcing** | In PeopleSoft Purchasing, provides the infrastructure to maintain, display, and select an appropriate vendor and vendor pricing structure that is based on a regional sourcing model where the multiple ship to locations are grouped. Sourcing may occur at a level higher than the ship to location. |
| **relationship object** | In PeopleSoft Enterprise Incentive Management, these objects further define a compensation structure to resolve transactions by establishing associations between compensation objects and business objects. |
| **remote data source data** | Data that is extracted from a separate database and migrated into the local database. |
| **REN server** | Abbreviation for *real-time event notification server* in PeopleSoft MultiChannel Framework. |
| **requester** | In PeopleSoft eSettlements, an individual who requests goods or services and whose ID appears on the various procurement pages that reference purchase orders. |
| **role** | Describes how people fit into PeopleSoft Workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity. |
| **role user** | A PeopleSoft Workflow user. A person's role user ID serves much the same purpose as a user ID does in other parts of the system. PeopleSoft Workflow uses role user IDs to determine how to route worklist items to users (through an email address, for example) and to track the roles that users play in the workflow. Role users do not need PeopleSoft user IDs. |
| **roll up** | In a tree, to roll up is to total sums based on the information hierarchy. |
| **run control** | A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start a program that manipulates data. |
| **run control ID** | A unique ID to associate each user with his or her own run control table entries. |

**run-level context** In PeopleSoft Enterprise Incentive Management, associates a particular run (and batch ID) with a period context and plan context. Every plan context that participates in a run has a separate run-level context. Because a run cannot span periods, only one run-level context is associated with each plan context.

**search query** You use this set of objects to pass a query string and operators to the search engine. The search index returns a set of matching results with keys to the source documents.

**section** In PeopleSoft Enterprise Incentive Management, a collection of incentive rules that operate on transactions of a specific type. Sections enable plans to be segmented to process logical events in different sections.

**security event** In commitment control, security events trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.

**serial genealogy** In PeopleSoft Manufacturing, the ability to track the composition of a specific, serial-controlled item.

**serial in production** In PeopleSoft Manufacturing, enables the tracing of serial information for manufactured items. This is maintained in the Item Master record.

**session** In PeopleSoft Enterprise Learning Management, a single meeting day of an activity (that is, the period of time between start and finish times within a day). The session stores the specific date, location, meeting time, and instructor. Sessions are used for scheduled training.

**session template** In PeopleSoft Enterprise Learning Management, enables you to set up common activity characteristics that may be reused while scheduling a PeopleSoft Enterprise Learning Management activity—characteristics such as days of the week, start and end times, facility and room assignments, instructors, and equipment. A session pattern template can be attached to an activity that is being scheduled. Attaching a template to an activity causes all of the default template information to populate the activity session pattern.

**setup relationship** In PeopleSoft Enterprise Incentive Management, a relationship object type that associates a configuration plan with any structure node.

**share driver expression** In PeopleSoft Business Planning, a named planning method similar to a driver expression, but which you can set up globally for shared use within a single planning application or to be shared between multiple planning applications through PeopleSoft Enterprise Warehouse.

**single signon** With single signon, users can, after being authenticated by a PeopleSoft application server, access a second PeopleSoft application server without entering a user ID or password.

**source transaction** In commitment control, any transaction generated in a PeopleSoft or third-party application that is integrated with commitment control and which can be checked against commitment control budgets. For example, a pre-encumbrance, encumbrance, expenditure, recognized revenue, or collected revenue transaction.

**SpeedChart** A user-defined shorthand key that designates several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a SpeedChart definition.

**SpeedType** A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.

**staging** A method of consolidating selected partner offerings with the offerings from the enterprise's other partners.

**statutory account** Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft, this is equivalent to the Alternate Account (ALTACCT) ChartField.

**step** In PeopleSoft Sales Incentive Management, a collection of sections in a plan. Each step corresponds to a step in the job run.

**storage level** In PeopleSoft Inventory, identifies the level of a material storage location. Material storage locations are made up of a business unit, a storage area, and a storage level. You can set up to four storage levels.

**subcustomer qualifier** A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.

**Summary ChartField** You use summary ChartFields to create summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).

**summary ledger** An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. Summary ledgers increase speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.

**summary time period** In PeopleSoft Business Planning, any time period (other than a base time period) that is an aggregate of other time periods, including other summary time periods and base time periods, such as quarter and year total.

**summary tree** A tree used to roll up accounts for each type of report in summary ledgers. Summary trees enable you to define trees on trees. In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the *basis* tree). A summary tree structure specifies the details on which the summary trees are to be built.

**syndicate** To distribute a production version of the enterprise catalog to partners.

**system function** In PeopleSoft Receivables, an activity that defines how the system generates accounting entries for the general ledger.

**TableSet** A means of sharing similar sets of values in control tables, where the actual data values are different but the structure of the tables is the same.

**TableSet sharing** Shared data that is stored in many tables that are based on the same TableSets. Tables that use TableSet sharing contain the SETID field as an additional key or unique identifier.

**target currency** The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.

**template** A template is HTML code associated with a web page. It defines the layout of the page and also where to get HTML for each part of the page. In PeopleSoft, you use templates to build a page by combining HTML from a number of sources. For a PeopleSoft portal, all templates must be registered in the portal registry, and each content reference must be assigned a template.

**territory** In PeopleSoft Sales Incentive Management, hierarchical relationships of business objects, including regions, products, customers, industries, and participants.

**TimeSpan** A relative period, such as year-to-date or current period, that can be used in various PeopleSoft General Ledger functions and reports when a rolling time frame, rather

| | |
|---|---|
| | than a specific date, is required. TimeSpans can also be used with flexible formulas in PeopleSoft Projects. |
| **trace usage** | In PeopleSoft Manufacturing, enables the control of which components will be traced during the manufacturing process. Serial- and lot-controlled components can be traced. This is maintained in the Item Master record. |
| **transaction allocation** | In PeopleSoft Enterprise Incentive Management, the process of identifying the owner of a transaction. When a raw transaction from a batch is allocated to a plan context, the transaction is duplicated in the PeopleSoft Enterprise Incentive Management transaction tables. |
| **transaction state** | In PeopleSoft Enterprise Incentive Management, a value assigned by an incentive rule to a transaction. Transaction states enable sections to process only transactions that are at a specific stage in system processing. After being successfully processed, transactions may be promoted to the next transaction state and "picked up" by a different section for further processing. |
| **Translate table** | A system edit table that stores codes and translate values for the miscellaneous fields in the database that do not warrant individual edit tables of their own. |
| **tree** | The graphical hierarchy in PeopleSoft systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies. |
| **unclaimed transaction** | In PeopleSoft Enterprise Incentive Management, a transaction that is not claimed by a node or participant after the allocation process has completed, usually due to missing or incomplete data. Unclaimed transactions may be manually assigned to the appropriate node or participant by a compensation administrator. |
| **universal navigation header** | Every PeopleSoft portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like Home, Favorites, and signoff) the universal navigation header can also display a welcome message for each user. |
| **user interaction object** | In PeopleSoft Sales Incentive Management, used to define the reporting components and reports that a participant can access in his or her context. All Sales Incentive Management user interface objects and reports are registered as user interaction objects. User interaction objects can be linked to a compensation structure node through a compensation relationship object (individually or as groups). |
| **variable** | In PeopleSoft Sales Incentive Management, the intermediate results of calculations. Variables hold the calculation results and are then inputs to other calculations. Variables can be plan variables that persist beyond the run of an engine or local variables that exist only during the processing of a section. |
| **VAT exception** | Abbreviation for *value-added tax exception.* A temporary or permanent exemption from paying VAT that is granted to an organization. This terms refers to both VAT exoneration and VAT suspension. |
| **VAT exempt** | Abbreviation for *value-added tax exempt.* Describes goods and services that are not subject to VAT. Organizations that supply exempt goods or services are unable to recover the related input VAT. This is also referred to as exempt without recovery. |
| **VAT exoneration** | Abbreviation for *value-added tax exoneration.* An organization that has been granted a permanent exemption from paying VAT due to the nature of that organization. |
| **VAT suspension** | Abbreviation for *value-added tax suspension.* An organization that has been granted a temporary exemption from paying VAT. |
| **warehouse** | A PeopleSoft data warehouse that consists of predefined ETL maps, data warehouse tools, and DataMart definitions. |

**work order** In PeopleSoft Services Procurement, enables an enterprise to create resource-based and deliverable-based transactions that specify the basic terms and conditions for hiring a specific service provider. When a service provider is hired, the service provider logs time or progress against the work order.

**worksheet** A way of presenting data through a PeopleSoft Business Analysis Modeler interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.

**worklist** The automated to-do list that PeopleSoft Workflow creates. From the worklist, you can directly access the pages you need to perform the next action, and then return to the worklist for another item.

**XML schema** An XML definition that standardizes the representation of application messages, component interfaces, or business interlinks.

**yield by operation** In PeopleSoft Manufacturing, the ability to plan the loss of a manufactured item on an operation-by-operation basis.

**zero-rated VAT** Abbreviation for *zero-rated value-added tax.* A VAT transaction with a VAT code that has a tax percent of zero. Used to track taxable VAT activity where no actual VAT amount is charged. Organizations that supply zero-rated goods and services can still recover the related input VAT. This is also referred to as exempt with recovery.

# Index

## D



## E

## N



## P

## Q



## R



## S



## T



## V



## W



## X