PeopleSoft®

PeopleSoft Enterprise Components for PeopleSoft Enterprise HRMS and Campus Solutions 8.9 PeopleBook

April 2005

PeopleSoft Enterprise Components for PeopleSoft Enterprise HRMS and Campus Solutions 8.9 PeopleBook
SKU HRCS89MP1ECQ-B 0405
Copyright © 1988-2005 PeopleSoft, Inc. All rights reserved.

All material contained in this documentation is proprietary and confidential to PeopleSoft, Inc. ("PeopleSoft"), protected by copyright laws and subject to the nondisclosure provisions of the applicable PeopleSoft agreement. No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including, but not limited to, electronic, graphic, mechanical, photocopying, recording, or otherwise without the prior written permission of PeopleSoft.

This documentation is subject to change without notice, and PeopleSoft does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to PeopleSoft in writing.

The copyrighted software that accompanies this document is licensed for use only in strict accordance with the applicable license agreement which should be read carefully as it governs the terms of use of the software and this document, including the disclosure thereof.

PeopleSoft, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, PeopleTalk, and Vantive are registered trademarks, and Pure Internet Architecture, Intelligent Context Manager, and The Real-Time Enterprise are trademarks of PeopleSoft, Inc. All other company and product names may be trademarks of their respective owners. The information contained herein is subject to change without notice.

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 © 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 © 1998-2003 The OpenSSL Project. All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

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 © 1995-1998 Eric Young. All rights reserved.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Copyright © 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 © 2001 by Andrei Alexandrescu. This code accompanies the book:

Alexandrescu, Andrei. "Modern C++ Design: Generic Programming and Design Patterns Applied." Copyright © 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.

Helma Project

Copyright © 1999-2004 Helma Project. 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 HELMA 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.

Helma includes third party software released under different specific license terms. See the licenses directory in the Helma distribution for a list of these license.

Sarissa

Copyright © 2004 Manos Batsis.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Contents

General Preface

About This PeopleBook Preface xxi

PeopleSoft Application Prerequisites xxi

PeopleSoft Application Fundamentals xxi

Documentation Updates and Printed Documentation xxii

Obtaining Documentation Updates xxii

Ordering Printed Documentation xxii

Additional Resources xxiii

Typographical Conventions and Visual Cues xxiv

Typographical Conventions xxiv

Visual Cues xxv

Country, Region, and Industry Identifiers xxvi

Currency Codes xxvi

Comments and Suggestions xxvi

Common Elements Used in PeopleBooks xxvii

Preface

PeopleSoft Enterprise Components Preface xxix

Overview of PeopleSoft Enterprise Components xxix

Part 1
Getting Started With PeopleSoft Enterprise Components

Chapter 1

Getting Started With PeopleSoft Enterprise Components 3

Getting Started With PeopleSoft Directory Interface 3

Getting Started With Common Objects and Components 4

Getting Started With PeopleSoft Active Analytics Framework 4

Getting Started With Portal Utilities 5

Getting Started With Enterprise Integration 5

Part 2
PeopleSoft Directory Interface

Chapter 2
Using PeopleSoft Directory Interface.....9
Understanding PeopleSoft Directory Interface.....9
Overview of Using PeopleSoft Directory Interface.....10
Setting Up in PeopleSoft Application Designer and PeopleSoft Integration Broker.....11
Using the Directory Configurations Component.....11
Common Elements Used in This Chapter.....12
Defining and Configuring the Directory.....12
Pages Used to Define and Configure the Directory.....13
Configuring the Directory Connection.....13
Caching the Schema.....15
Deleting the Directory Configuration.....16
(Optional) Setting Up Directory Authentication.....16
Pages Used to Set Up Directory Authentication.....17
Using Map Authentication.....17
Viewing User Properties.....18
Setting Up Mappings.....19
Understanding Mappings.....20
Pages Used to Set Up Mappings.....20
Creating Mappings.....20
Modifying the Distinguished Name.....22
Specifying Distinguished Name Details.....23
Translating or Performing Functions With Database Values.....24
Specifying Distinguished Name Defaults.....26
Mapping Data to Directory Object Class Attributes.....26
Locating Delivered Messages.....28
(Optional) Setting Up Entry Membership Rules.....28
Pages Used to Set Up Entry Membership Rules.....28
Creating Entry Definitions.....28
Specifying Entry Membership Rules.....30
Loading Data into the Directory.....32
Understanding Directory Load Behavior.....32
Pages Used to Load PeopleSoft Data into the Directory.....32
Loading the Directory with PeopleSoft Data.....32

Chapter 3
Reviewing Directory Data and Generating Reports....35
Reviewing LDAP Directory Data....35
Pages Used to Review Directory Data....35
Running a Directory Audit....35
Running a Directory Search....36
Viewing PeopleSoft Directory Interface Reports....37
Pages Used to View PeopleSoft Directory Interface Reports....38
Viewing the Directory Audit Report....38
Viewing the Business Interlink Status Report....38

Part 3
PeopleSoft Common Objects and Components

Chapter 4
Formatting Addresses....43
Understanding Address Formats....43
Specifying Address Formats....43
Pages Used to Specify Address Formats....44
Describing Countries....44
Formatting Addresses....44
Validating Addresses....46
Specifying State Information....46

Chapter 5
Working With Currencies and Market Rates....49
Understanding Currencies and Market Rates....49
Understanding Currency and Market Rate Tables....49
Understanding Triangulation....50
Understanding Conversion Factor Fields and the Visual Rate....51
Understanding Application-Specific Requirements for Currency Conversion....52
Defining Currencies....52
Page Used to Define Currencies....52
Defining Currency Codes....53
Defining Currency Quotation Methods....54
Page Used to Define Currency Quotation Methods....54
Defining Currency Quotation Methods....54

Defining Market Rates....56
Pages Used to Define Market Rates....57
Defining Market Rate Indexes....57
Defining Rate Types....58
Creating Market Rate Definitions....59
Defining Market Rates....61
Accessing Exchange Rate Details....63
Calculating Cross, Triangulated, and Reciprocal Rates....65
Understanding the EOP_RATECALC process....65
Page Used to Calculate Cross, Triangulated, and Reciprocal Rates....65
Running the EOP_RATECALC Process....65
Using the Currency Exchange Calculator....67
Page Used to Use the Currency Exchange Calculator....68
Converting Amounts Using the Currency Exchange Calculator....68

Chapter 6
Using Datasets....69
Understanding Datasets....69
Defining Dataset Rules....69
Understanding Dataset Rules....69
Page Used to Define Dataset Rules....70
Creating Dataset Rules....70
Defining Dataset Roles....71
Page Used to Define Dataset Roles....71
Defining Dataset Roles....71
Defining Mobile Data Distribution....71
Understanding Mobile Data Distribution....72
Pages Used to Define Mobile Data Distribution....72
Defining Mobile Data Distribution Rules....72
Using Mobile User Rules....74

Chapter 7
Setting Up the Credit Card Interface....75
Understanding Credit Card Processing....75
Understanding a CyberSource Integration....75
Understanding an XML-Based Integration....76
Understanding Manual Processing....76
Setting Up the Credit Card Interface....76

Pages Used in Setting Up the Credit Card Interface....77
Defining Connection Parameters....77
Defining Accepted Credit Card Types....80
Testing the Credit Card Interface....81
Testing Credit Card Transactions....83
Configuring the Java Interlink Plug-In....86
Understanding the Java Interlink Plug-In....86
Setting Up the CyberSource API and Java Plug-in on Microsoft Windows NT Systems....87
Setting Up the CyberSource API and the Java Plug-in on Solaris Systems....88
Configuring the PeopleSoft Application Designer Plug-in....90
Configuring Access to Account Information....91
Configuring Proxy Server Support....94
Setting Up Tracing....94
Setting Parameter Check Logic....95
Setting the Decryption Option....95
Setting Transaction Inputs and Outputs....96
Identifying Required Parameters....96
Using Authorize-and-Bill Transactions....96
Using Authorize-Only Transactions....97
Using Bill-Only Transactions....98
Using Credit Transactions....99
Using the Address Verification Service....100
Using the Fraud Screen Service....101
Identifying Input Fields That Are Not Used or Supported....102

Chapter 8
Using Interactive Reports....103
Understanding Interactive Reports....103
Common Elements Used in This Chapter....103
Configuring Interactive Reports....104
Pages Used to Configure Interactive Reports....104
Defining Reports....104
Defining Servers....105
Defining ODBC Connections....106
Configuring the Environment....107
Mapping Queries....107
Viewing Query Prompts....108
Validating the Environment....109

Part 4
PeopleSoft Active Analytics Framework

Chapter 9
Understanding PeopleSoft Active Analytics Framework.....113
Understanding PeopleSoft Active Analytics Framework.....113
Understanding Policies and Trigger Points.....113
Understanding the Data Library.....114
Understanding the Action Framework.....115

Chapter 10
Building and Managing Policies.....117
Building Policies.....117
Prerequisites to Building Policies.....117
Pages Used in This Section.....118
Building, Editing and Activating Policies.....118
Managing Trigger Points.....124
Removing a Policy or Policy Groups.....125
Reordering Policy or Policy Groups.....126
Adding a Policy or Policy Group.....126
Reusing Policies.....126
Adding a Precondition.....126
Setting Execution Options.....127
Understanding Execution Options.....127

Chapter 11
Setting Up the Data Library.....133
Understanding the Data Library.....133
Using Data Library Components.....133
Pages Used in This Chapter.....134
Creating Implementations.....134
Registering an Implementation.....135
Creating Terms.....136
Understanding Term Properties.....136
Using Generic Implementations.....138
Using Contextual Implementations.....138
Managing Terms.....138

Testing Term Implementations.....140

Chapter 12
Managing Contexts.....143
Understanding Contexts.....143
Configuring Contexts.....144
Pages Used in This Chapter.....144
Generating Context Objects.....144
Managing Context Objects.....146

Chapter 13
Setting Up the Action Framework.....149
Understanding the Action Framework.....149
Architecture of the Action Framework.....149
Understanding Action Types.....149
Understanding How Actions Execute.....150
Registering Action Types and Action Type Bundles.....151
Pages Used in This Chapter.....151
Registering Action Types.....151
Registering Action Type Bundles.....153

Chapter 14
Administering the Framework.....155
Using the Setup Components.....155
Pages Used in This Chapter.....155
Setting Log and Installation Options.....156
Setting Data Library Log Options.....156
Installation Options.....157
Registering Action Objective.....159
Registering Trigger Types and Trigger Points.....160
Registering Trigger Types.....160
Registering Trigger Points.....161
Defining Subject Areas.....162
Registering Operators and Operator Sets.....164
Registering Operators.....164
Registering Operator Sets.....165
Registering Resolution Methods.....166

Registering Business Domain and Category....167
Register Business Domain....167
Register Category....167

Chapter 15
Considerations for Enabling the Framework....169
Considerations When Creating Contexts....169
Considerations When Creating Custom Operator Expressions....170
Considerations When Creating a Trigger Point....171
Enabling a Trigger Point in a PeopleCode Event....172
Enabling the Display Action in a PeopleSoft Application Page....172
Considerations When Creating a New Action Type....172
Configuring the Action Type....172
Creating Design and Runtime Application Class Code....176
Considerations When Creating an Application Class Implementation....177
Technical Details of Application Class Implementations....177
Example of an Application Class Accessing Implementation Binds....178
Sample Methods of an Application Class Resolving Terms....178
Example of How a Value Can Be Passed to the Data Library....179
Sample Code of an Application Class Implementation....179
Considerations When Creating a PS Query Implementation....180
Considerations When Creating a Policy....181

Part 5
Portal Utilities

Chapter 16
Working With Pagelet Wizard....185
Overview of Pagelet Wizard....185
Pagelet Wizard Data Sources....188
Pagelet Wizard Display Formats....189
Pagelet Wizard Display Sizes....195
Pagelet Wizard Pagelet Types....196

Chapter 17
Setting Up Pagelet Wizard Options.....201
Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types.....201
Pages Used to Define Pagelet Wizard Transformation Types, Display Formats, and Data Types.....202
Defining Transformation Types.....202
Defining Display Formats.....203
Defining Data Types.....205
Defining Pagelet Wizard XSL Prototypes.....207
Page Used to Define Pagelet Wizard XSL Prototypes.....207
Defining XSL Prototypes.....207
Defining Pagelet Wizard Headers and Footers.....209
Pages Used to Define Pagelet Wizard Headers and Footers.....209
Defining Headers.....209
Defining Header Images.....210
Defining Footers.....211
Defining Footer Images.....212
Defining Pagelet Wizard Pagelet Categories.....214
Page Used to Define Pagelet Wizard Pagelet Categories.....214
Defining Pagelet Categories.....214

Chapter 18
Using Pagelet Wizard.....217
Step 1: Entering Pagelet Identifying Information.....217
Page Used to Enter Pagelet Identifying Information.....217
Specifying Pagelet Information.....217
Step 2: Selecting a Pagelet Data Source.....219
Page Used to Select a Pagelet Data Source.....219
Selecting a Query Data Source.....219
Selecting an HTML Data Source.....221
Selecting a Search Record Data Source.....222
Selecting a Navigation Collection Data Source.....223
Step 3: Specifying Pagelet Data Source Parameters.....224
Understanding System Variables Supported as Data Source Parameters.....224
Common Elements Used in This Section.....225
Page Used to Specify Pagelet Data Source Parameters.....227
Specifying Query Data Source Parameters.....227
Specifying Search Record Data Source Parameters.....228
Specifying Navigation Collection Data Source Parameters.....230
Specifying Data Source Parameter Values.....231

Step 4: Selecting a Pagelet Display Format....231
Page Used to Select a Pagelet Display Format....232
Selecting a Display Format....232
Step 5: Specifying Pagelet Display Options....233
Common Elements Used in This Section....233
Pages Used to Specify Pagelet Display Options....235
Specifying Table Display Options....236
Specifying Passthru Display Options....238
Specifying Search List Display Options....238
Specifying Menu Display Options....239
Specifying List Display Options....241
Specifying Chart Display Options....243
Specifying Custom Display Options....246
Specifying Data Link Details....248
Step 6: Specifying Pagelet Publication Options....249
Common Elements Used in This Section....250
Pages Used to Specify Pagelet Publication Options....251
Specifying Publication Options....251

Chapter 19
Administering Pagelets....255
Common Elements Used in This Chapter....255
Deleting and Copying Pagelet Wizard Pagelets....256
Pages Used to Delete and Copy Pagelet Wizard Pagelets....257
Deleting Pagelets....257
Copying Pagelets....258
Generating Export and Import Scripts for Pagelet Wizard Pagelets....259
Understanding Export and Import Scripts for Pagelet Wizard Pagelets....260
Pages Used to Generate Export and Import Scripts for Pagelet Wizard Pagelets....260
Generating Export and Import Pagelet Scripts....260
Generating Export/Import Scripts for Pagelet Wizard Setup Data....262
Understanding Export and Import Scripts for Pagelet Wizard Setup Data....262
Publishing Pagelets to Other Portals....262
Pages Used to Publish Pagelets to Other Portals....263
Publishing a Pagelet to Multiple Portals....263
Publishing Multiple Pagelets to a Portal....264
Reviewing Pagelet Wizard Pagelet Details....266
Page Used to Review Pagelet Wizard Pagelet Details....266

Chapter 20
Using Pagelet Wizard Post-Transformation Processing.....267
Understanding Pagelet Wizard Post-Transformation Processing.....267
Inserting Message Catalog Text.....268
Formatting Date, Time, DateTime, Currency, and Number Elements.....269
Inserting a Content-Reference-Based Link.....270
Inserting a Content Reference URL as Text.....271
Generating Charts.....271
Inserting Pagelet Headers and Footers in Specific Locations.....271

Chapter 21
Working with Navigation Pages.....273
Describing Navigation Pages.....273
Describing Standard Navigation Pages.....273
Describing Custom Navigation Pages.....275
Common Elements Used on Navigation Pages.....275
Enabling Standard Navigation Page Display.....276
Enabling Navigation Page Display Caching.....277
To enable caching for a custom navigation page accessed by clicking a content reference link.....277
To enable caching for a folder or breadcrumb on a navigation page.....277
To enable caching for a standard navigation page accessed by clicking a menu navigation folder.....277

Chapter 22
Setting Up Navigation Collection Options.....279
Describing the Navigation Collection Option Default Hierarchy.....279
Defining Navigation Collection Options.....280
Pages Used to Define Navigation Collection Options.....281
Defining Navigation Collection System Options.....281
Defining Navigation Collection Registry Options.....285

Chapter 23
Using Navigation Collections.....291
Common Elements Used in This Chapter.....291
Describing Navigation Collections.....291
Describing Navigation Collection Publications.....292
Describing Folder and Link Sequence on Navigation Pages.....295
Describing Navigation Collection Images.....298

Creating and Maintaining Navigation Collections....299
Pages Used to Create and Maintain Navigation Collections....300
Searching for a Navigation Collection....301
Creating or Editing a Navigation Collection....303
Adding a Folder....306
Selecting a Source Folder....308
Selecting Navigation Collection Images....310
Editing a Folder....311
Adding a Link....312
Selecting a Source Link....314
Editing a Link....316
Publishing Navigation Collections....318
Pages Used to Publish Navigation Collections....318
Publishing a Navigation Collection....318

Chapter 24
Running Portal Utility Processes....323
Synchronizing Collection Security....323
Overview of the Sync Collection Security Process....323
Page Used to Synchronize Collection Security....324
Running the Sync Collection Security Process....324
Deleting Empty Portal Registry Folders....325
Overview of the Delete Empty Folders Process....325
Page Used to Delete Empty Portal Registry Folders....325
Cleaning Portal Projects....325
Overview of the Clean Portal Project Process....325
Page Used to Clean Portal Projects....327
Running the Clean Portal Project Process....327

Part 6
Enterprise Integration

Chapter 25
Understanding Enterprise Integration....333
Understanding PeopleSoft Messaging....333
Understanding PeopleSoft Business Interlinks....333
Understanding PeopleSoft Component Interfaces....333

Understanding File Layouts.....334
Understanding PeopleSoft Integration Broker.....334

Chapter 26
Understanding Integration Points.....335
Overview of Integration Points.....335

Chapter 27
Activating Messaging Integration Points.....337
Setting Up PeopleSoft Messaging Integration Points.....337
Pages Used to Set Up PeopleSoft Messaging Integration Points.....337
Activating Messages for Publication.....338
Setting Up Publication Rules.....338
Mapping Nodes to a Chunking Rule.....339
Assigning Business Units or SetIDs to a Chunking Rule.....340
Specifying OnRoute PeopleCode.....340
Setting Up Related Languages.....341
Understanding Related Language Tables.....341
Understanding Related Language Guidelines for Messaging.....342
Interpreting Component Processor Behavior.....343
Publishing a Message from a Component.....343
Publishing a Message from Batch Programs.....344
Subscribing to Data in a PeopleSoft Multilingual Environment.....344
Subscribing to Data in a Non-Multilingual Environment.....344
Examining Related-Language Messaging Scenarios.....345
Publishing a Non-Base Language Message to a Subscribing System With a Different Base Language and No Prior Data.....346
Switching the Preferred Language to Japanese and Updating the Same Employee's Name and Address.....346

Chapter 28
Assigning Publishing Rules.....347
Understanding Publishing Rules.....347
The Publish Utility.....347
Terms Used When Defining Publish Rules.....347
Assigning Full Table Publishing Rules.....348
Pages Used to Assign Full Table Publishing Rules.....349
Associating a Rule to a Message.....349

Mapping a Message Record to Another Record....350
Specifying Languages in Which to Publish Messages....352
Assigning Batch Publishing Rules....353
Pages Used to Assign Batch Publishing Rules....353
Associating a Rule to a Message....353
Mapping a Message Record to Another Record....354
Assigning an Application Program to a Publishing Rule....354
Setting Up Message Chunking....355
Pages Used to Set Up Message Chunking....356
Understanding Message Chunking....356
Identifying When to Use Chunking....357
Selecting Chunking Fields....357
Creating Chunking Rules....359
Defining the Chunking Rule Description....360
Maintaining Chunking Data for Business Units....360
Maintaining Chunking Data for SetIDs....361
Adding Nodes to Existing Chunking Rules....362
Assigning Business Units or SetIDs to a Chunking Rule....362
Assigning Chunking Rules to a Business Unit....363
Assigning Chunking Rules to a SetID....364
Creating Custom Chunking Tables....364
Creating a Custom Chunking Table....365
Creating a View for the Component Search Record....365
Creating Maintenance Pages....365
Creating a Component....366
Creating Routing PeopleCode....367

Chapter 29
Using the Error Handling Utility....369
Understanding the Error Handling Utility....369
Error Management Process....369
Setting Up and Maintaining Message Errors....370
Pages Used to Set Up and Maintain Message Errors....371
Creating Error-Correction Pages....371
Setting Up the Error Handling Utility....372
Setting Up Row Security....374
Setting Up Workflow Notification in PeopleSoft Application Designer....374
Testing the Error Handling Utility....375
Correcting Message Errors....375

Understanding the Workflow Notification Process.....375
Correcting Message Errors.....376

Chapter 30
Using the Effective Date Publish Utility.....377
Understanding the Effective Date Publish Utility.....377
Performing a Full Data Publish of Current Effective Data.....378
Pages Used for Full Data Publish of Current Effective Data.....379
Creating Effective-Dated Messages.....379
Defining the Message Node, Message Channel, and Message Definition.....380
Defining Chunking Rules and Ordering Views.....381
Creating Publish Rule Definitions.....382
Creating Run Controls for the Full Data Publish Program.....382
Defining Message Routing.....384
Publishing Incremental Messages of Current Effective Data.....384
Creating Message Definitions and Assigning the Message Channel.....385
Creating Subscription Processes That Open the Generic Effective-Dated Delay Function.....385
Publishing Effective-Dated Rows from the Delay Table.....386
Understanding Effective-Dated Row Publishing.....386
Page Used to Run the Effective Date Publish Utility.....387
Running the Effective Date Publish Utility.....387

Chapter 31
Using the Flat File Utility.....389
Understanding the Flat File Utility.....389
Processing Inbound Flat Files.....390
Initiating File Processing.....391
Pages Used to Initiate File Processing.....391
Setting Up Inbound Flat File Processing.....391
Initiating Inbound Flat File Processing.....393
Testing Inbound Flat File Processing.....394

Chapter 32
Using the XML Schema Utility.....401
Understanding the XML Schema Utility.....401
Generating the XML Schema.....401
Page Used to Generate XML Schema.....401

Generating the XML Schema.....401
Interpreting Sample Output.....402

Appendix A
Integration Point Naming Standards.....411
Standard Action and Event Verbs.....411
Standard Business Object Nouns.....414

Appendix B
PeopleSoft Design Patterns.....417
List of Design Patterns.....417

Glossary of PeopleSoft Terms.....421

Index 443

About This PeopleBook Preface

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

This preface discusses:

- PeopleSoft application prerequisites.
- PeopleSoft application fundamentals.
- Documentation updates and printed documentation.
- Additional resources.
- Typographical conventions and visual cues.
- Comments and suggestions.
- Common elements in PeopleBooks.

Note. PeopleBooks document only page elements, such as fields and check boxes, 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.

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

You should be familiar with navigating the system and adding, updating, and deleting information by using PeopleSoft menus, and pages, forms, or windows. 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 applications.

Note. Application fundamentals PeopleBooks are not applicable to the PeopleTools product.

For some applications, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals PeopleBook. Most PeopleSoft product lines have a version of the application fundamentals PeopleBook. The preface of each PeopleBook identifies the application fundamentals PeopleBooks that are associated with that PeopleBook.

The application fundamentals PeopleBook consists of important topics that apply to many or all PeopleSoft applications across one or more product lines. 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 the appropriate application fundamentals PeopleBooks. They provide the starting points for fundamental implementation tasks.

Documentation Updates and Printed 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 website. 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, https://www.peoplesoft.com/corp/en/login.jsp

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 website, access the PeopleBooks Press website under the Ordering PeopleBooks topic. The PeopleBooks Press website is a joint venture between PeopleSoft and MMA Partners, the book print vendor. Use a credit card, money order, cashier's check, or purchase order to place your order.

Telephone

Contact MMA Partners at 877 588 2525.

Email

Send email to MMA Partners at peoplebookspress@mmapartner.com.

See Also

PeopleSoft Customer Connection, https://www.peoplesoft.com/corp/en/login.jsp

Additional Resources

The following resources are located on the PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps
Interactive Services Repository	Interactive Services Repository
Hardware and software requirements	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation & Software, Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation & Software, Installation Guides and Notes
Integration information	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation and Software, Pre-built Integrations for PeopleSoft Enterprise and PeopleSoft EnterpriseOne Applications
Minimum technical requirements (MTRs) (EnterpriseOne only)	Implement, Optimize + Upgrade, Implementation Guide, Supported Platforms
PeopleBook documentation updates	Support, Documentation, Documentation Updates
PeopleSoft support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Prerelease Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes

Resource	Navigation
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.
- Country, region, and industry identifiers.
- Currency codes.

Typographical Conventions

This table contains the typographical conventions that are used in PeopleBooks:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system 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. We also use italics when we refer to words as words or letters as letters, as in the following: Enter 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 the W key.
`Monospace font`	Indicates a PeopleCode program or other code example.

Typographical Convention or Visual Cue	Description
“ ” (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.
. . . (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. Ampersands also precede all PeopleCode variables.

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.

If the note is preceded by *Important!,* the note 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.

Cross-References

PeopleBooks provide cross-references either under 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.

Country, Region, and Industry Identifiers

Information that applies only to a specific country, region, or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a country-specific heading: "(FRA) Hiring an Employee"

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Country Identifiers

Countries are identified with the International Organization for Standardization (ISO) country code.

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in PeopleBooks:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in PeopleBooks:

- USF (U.S. Federal)
- E&G (Education and Government)

Currency Codes

Monetary amounts are identified by the ISO currency code.

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 Used in 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. Select Always to run the request every time the batch process runs. Select Don't Run to ignore the request when the batch process runs.
Process Monitor	Click to access the Process List page, where you can view the status of submitted process requests.
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).
Request ID	An ID that represents a set of selection criteria for a report or process.
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.
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.
User ID	An ID that represents the person who generates a transaction.

PeopleSoft Enterprise Components Preface

This preface provides an overview of PeopleSoft Enterprise Component documentation included in this PeopleBook.

Overview of PeopleSoft Enterprise Components

PeopleSoft Enterprise Components include a variety of features that are generally available as common objects or components to all product lines delivered by PeopleSoft. This PeopleBook include the following 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.* Describes features generally available as common objects or components to all product lines, including interactive reports, currency and market rate components, datasets, and the credit card interface.
- *Active Analytics Framework.*Describes the framework and its features including the data library, the action framework, and considerations for technical users.
- *Portal Utilities.* Describes the Navigation Collection and Pagelet Wizard features, their setup options, and their publications. Also discusses portal utility batch processes.
- *Enterprise Integration.* Describes features related to integration from one PeopleSoft product to another, and integration between PeopleSoft and third party products.

Note. Some of the page elements and colors that your product uses may differ from the screen shots presented in this PeopleBook. This book uses a generic style sheet for the purposes of illustration only.

PART 1

Getting Started With PeopleSoft Enterprise Components

Chapter 1
Getting Started With PeopleSoft Enterprise Components

CHAPTER 1

Getting Started With PeopleSoft Enterprise Components

This chapter provides information about getting started in the following PeopleSoft Enterprise Component areas:

- PeopleSoft Directory Interface
- Common Objects and Components
- PeopleSoft Active Analytics Framework
- 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 PeopleSoft 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\Services\NTDS\Parameters\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

PeopleSoft 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 5, "Working With Currencies and Market Rates," page 49.

See Chapter 6, "Using Datasets," page 69.

See Chapter 8, "Using Interactive Reports," page 103.

See Chapter 7, "Setting Up the Credit Card Interface," page 75.

Getting Started With PeopleSoft Active Analytics Framework

PeopleSoft Active Analytics Framework is a closed-loop decision-making system comprising a suite of tools. Refer to the following chapter for an overview of the framework:

See Chapter 9, "Understanding PeopleSoft Active Analytics Framework," page 113.

The framework includes:

- Components for building and managing policies.

  See Chapter 10, "Building and Managing Policies," page 117.
- A data library for registering terms.

  See Chapter 11, "Setting Up the Data Library," page 133.
- Components for creating and managing contexts.

  See Chapter 12, "Managing Contexts," page 143.
- An action framework.

  See Chapter 13, "Setting Up the Action Framework," page 149.
- Components for setting log options and registering elements of the framework.

  See Chapter 14, "Administering the Framework," page 155.
- Considerations for technical users, such as creating custom operator expressions, trigger points, a new action type and so on.

  See Chapter 15, "Considerations for Enabling the Framework," page 169.

Getting Started With Portal Utilities

PeopleSoft Enterprise Component portal utilities include components and processes that support Navigation Collections and Pagelet Wizard, 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 Navigation Collection implementation.

See Chapter 22, “Setting Up Navigation Collection Options,” Defining Navigation Collection System Options, page 281.

See Chapter 22, “Setting Up Navigation Collection Options,” Defining Navigation Collection Registry Options, page 285.

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

See Chapter 21, “Working with Navigation Pages,” Enabling Standard Navigation Page Display, page 276.

See Chapter 21, “Working with Navigation Pages,” Enabling Navigation Page Display Caching, page 277.

The following section covers transformation types, display formats, and data types that you may choose to set up as a part of your Pagelet Wizard implementation:

See Chapter 17, “Setting Up Pagelet Wizard Options,” Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types, page 201.

The following section covers Extensible Stylesheet Language (XSL) prototypes that you may choose to set up as a part of your Pagelet Wizard implementation:

See Chapter 17, “Setting Up Pagelet Wizard Options,” Defining Pagelet Wizard XSL Prototypes, page 207.

The following section covers header and footer definition options that you may choose to set up as a part of your Pagelet Wizard implementation:

See Chapter 17, “Setting Up Pagelet Wizard Options,” Defining Pagelet Wizard Headers and Footers, page 209.

The following section covers pagelet category definition options that you may choose to set up as a part of your Pagelet Wizard implementation:

See Chapter 17, “Setting Up Pagelet Wizard Options,” Defining Pagelet Wizard Pagelet Categories, page 214.

Getting Started With Enterprise Integration

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

If you are using integration points, the following sections may be helpful:

See Appendix A, “Integration Point Naming Standards,” page 411.

See Appendix B, “PeopleSoft Design Patterns,” page 417.

See Chapter 27, “Activating Messaging Integration Points,” Setting Up PeopleSoft Messaging Integration Points, page 337.

See Chapter 27, “Activating Messaging Integration Points,” Setting Up Related Languages, page 341.

See Chapter 29, “Using the Error Handling Utility,” Setting Up and Maintaining Message Errors, page 370.

PART 2

PeopleSoft Directory Interface

Chapter 2
Using PeopleSoft Directory Interface

Chapter 3
Reviewing Directory Data and Generating Reports

CHAPTER 2

Using PeopleSoft Directory Interface

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

- Define and configure the directory.
- Set up directory authentication.
- Load the directory schema.
- Set up mappings.
- Set up entry membership rules.
- Load data into the 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 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* (DN) 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 messages. Each 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. To do this, use one of the following options:

- 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.

See *Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Business Interlinks*

Overview of Using PeopleSoft Directory Interface

The section briefly describes the steps needed to use PeopleSoft Directory Interface, including:

- Setting up in PeopleSoft Application Designer and PeopleSoft Integration Broker.
- Using the Directory Configurations component

Setting Up in PeopleSoft Application Designer and PeopleSoft Integration Broker

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

Setting Up 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. PeopleSoft Directory Interface then 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 *Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Application Designer*

Setting Up 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 *Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Integration Broker*

Using the Directory Configurations Component

Access Directory Configurations component (PSDSSETUP) from the browser menu.

- Using the Directory Configurations 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	Select the directory product from the drop-down list box.
Default Connect DN	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 Directory Configurations component (PSDSSETUP) to define and configure the directory connection. In this section, we discuss how to:

- Configure the directory connection.
- Cache the schema.
- Delete the directory configuration.

See Also

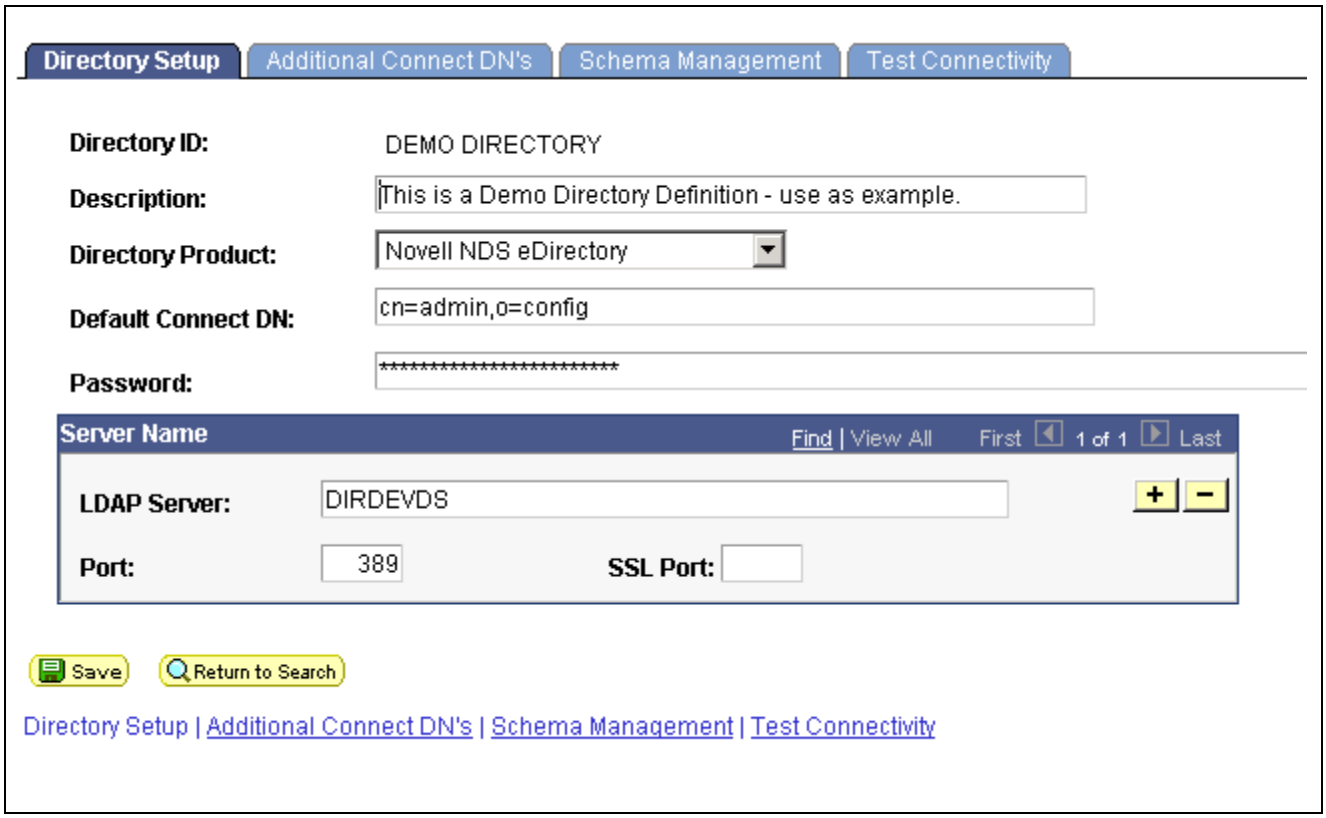
Enterprise PeopleTools 8.45 PeopleBook: Security Administration, "Employing LDAP Directory Services," Configuring the LDAP Directory

Pages Used to Define and Configure the Directory

Page Name	Object Name	Navigation	Usage
Directory Setup	DSDIRSETUP	Enterprise Components, Directory Interface, Definitions, Directory Configurations, Directory Setup	Enter values to configure the directory.
Additional Connect DN's	DSSERVERID	Enterprise Components, Directory Interface, Definitions, Directory Configurations, Additional Connect DN's	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 Deletions	Delete the directory configuration.

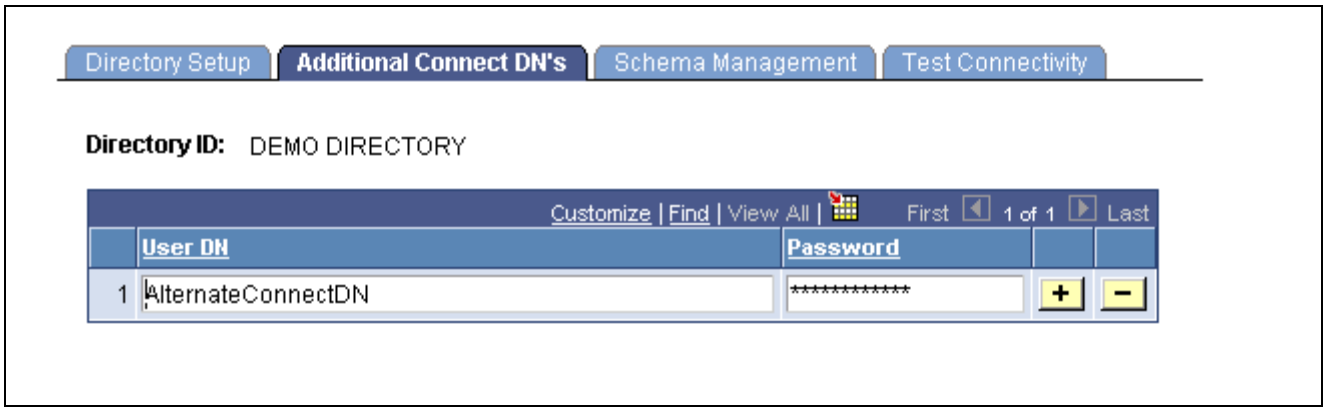
Configuring the Directory Connection

Access the Directory Setup page.



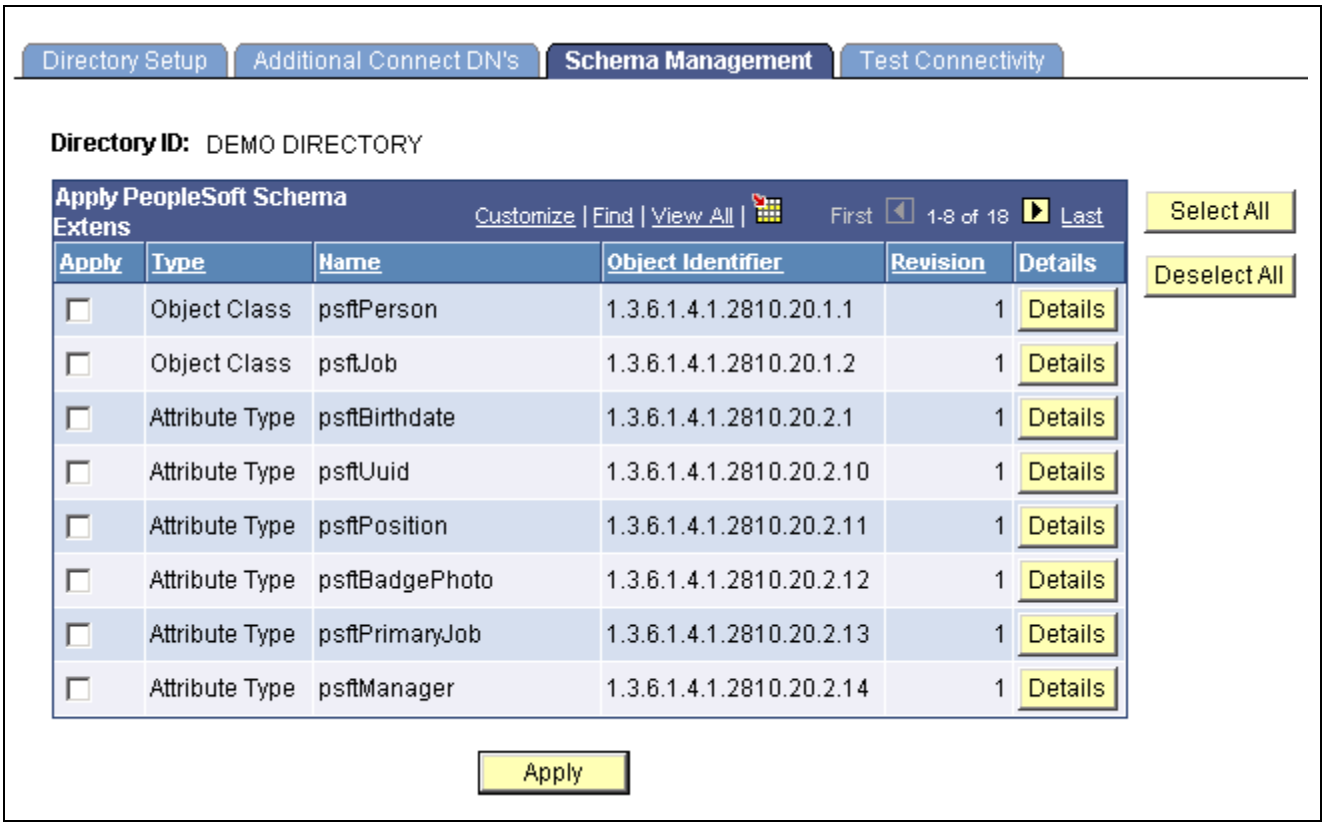
Directory Setup

Access the Additional Connect DN's page and add more Connect DNs and passwords, if needed.

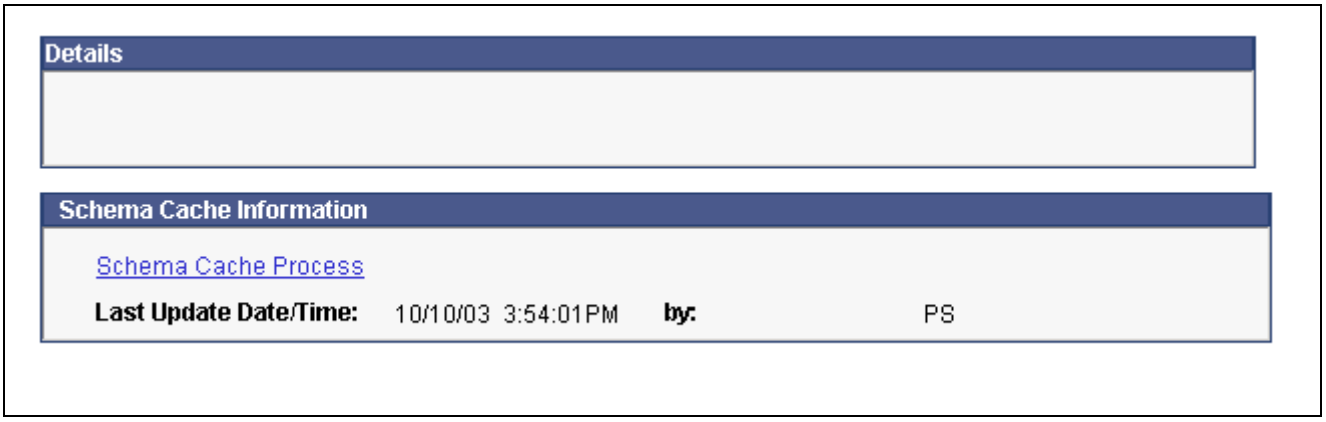


Additional Connect DN's

Access the Schema Management page.



Schema Management page (1 of 2)

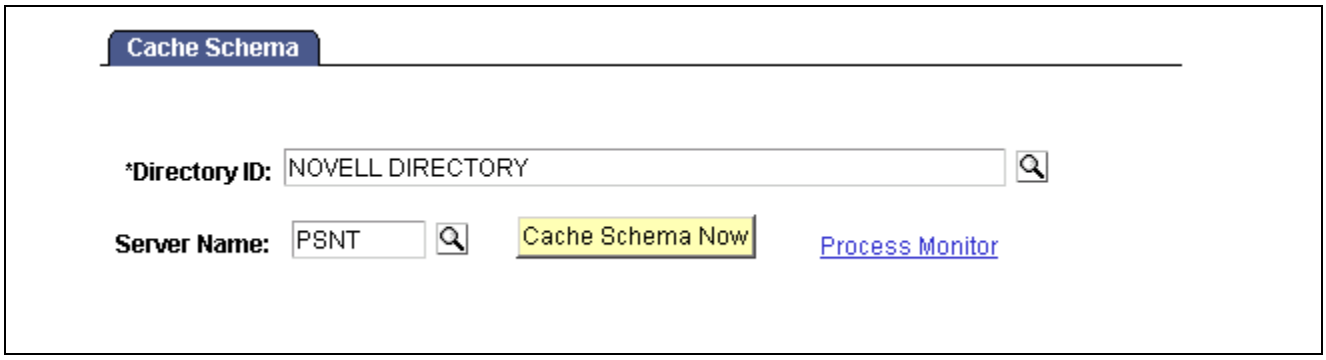


Schema Management page (2 of 2)

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

Caching the Schema

Access the Schema Caching page.

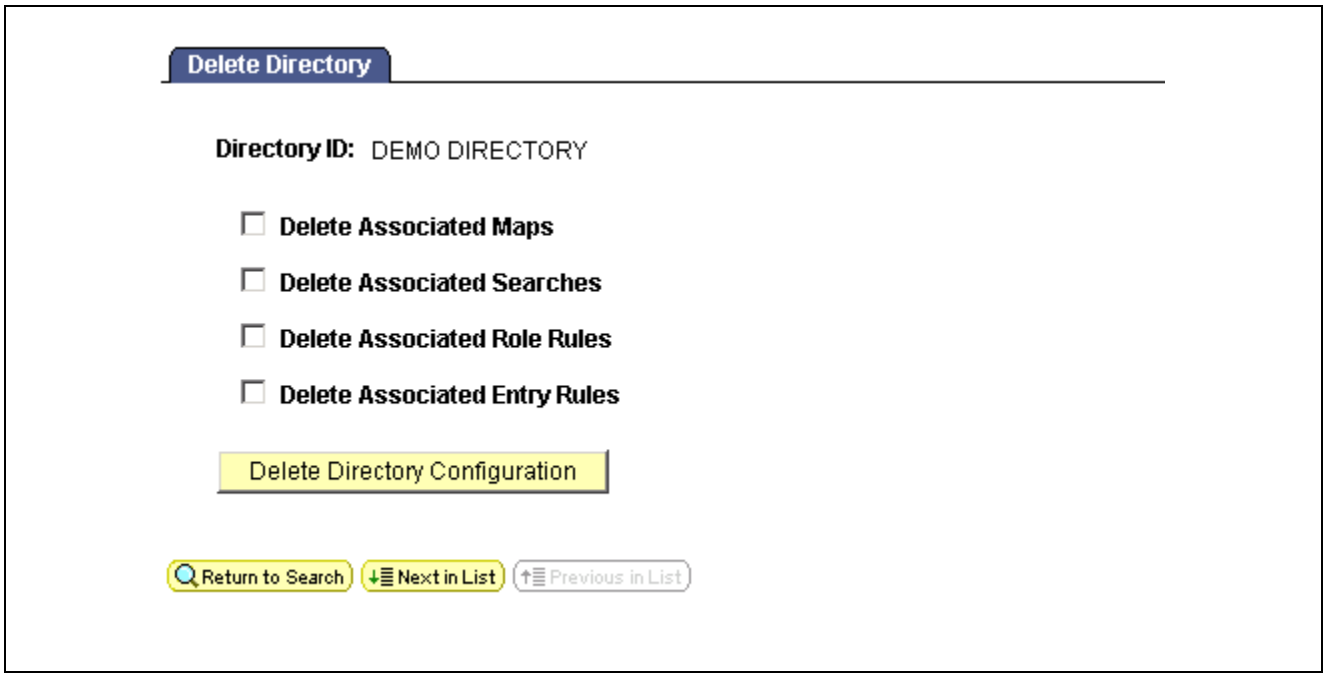


Cache Schema page

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

Deleting the Directory Configuration

Access the Delete Directory page.



Delete Directory page

Select the check boxes for the desired directory configuration deletions.

(Optional) Setting Up Directory Authentication

In this section, we discuss how to:

- Use map authentication.
- Viewing user properties.

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

See Also

Enterprise PeopleTools 8.45 PeopleBook: Security Administration, "Employing LDAP Directory Services," Creating the Authentication Map

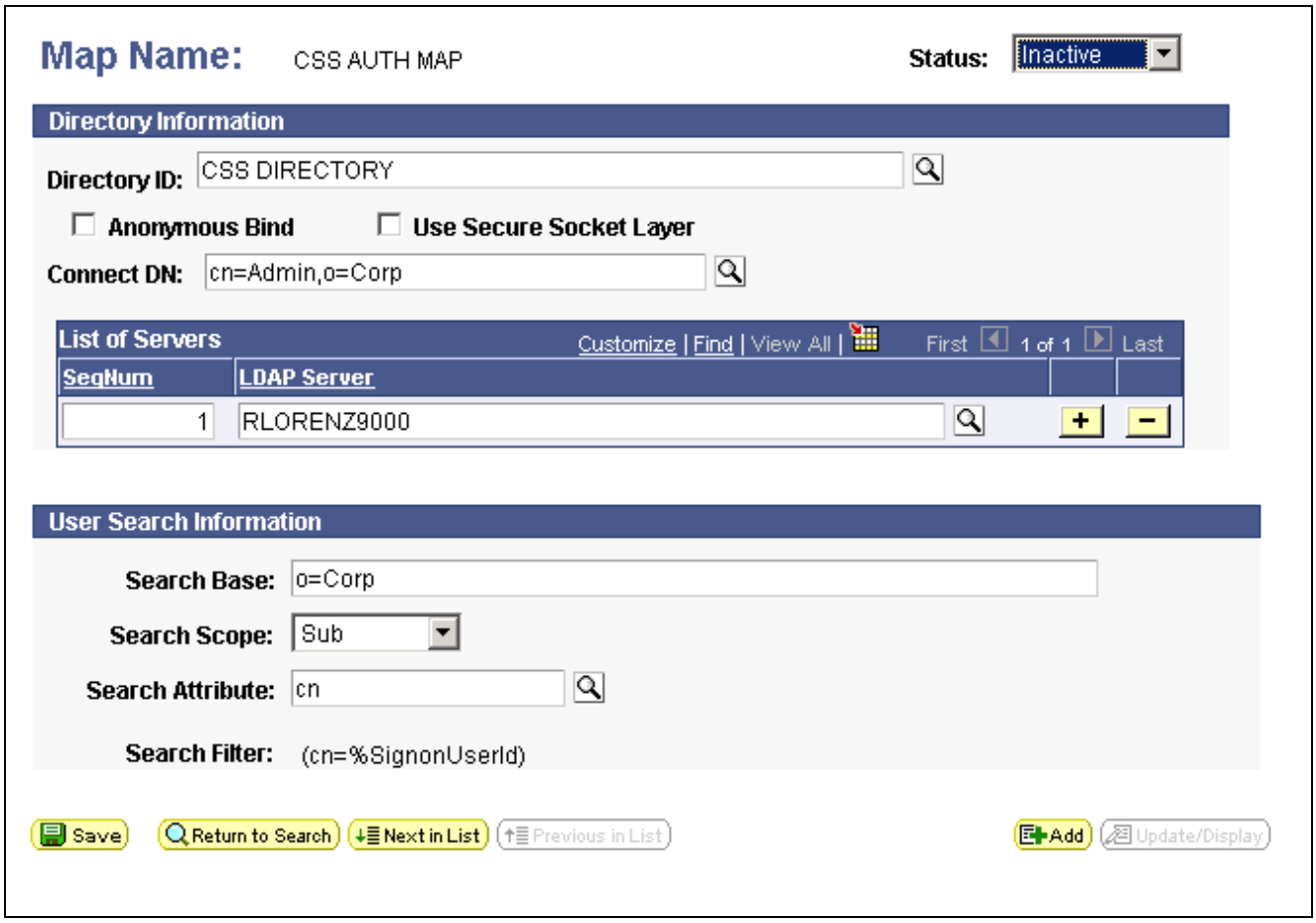
Enterprise PeopleTools 8.45 PeopleBook: Security Administration, "Employing LDAP Directory Services," Creating User Profile Maps

Pages Used to Set Up Directory Authentication

Page Name	Object Name	Navigation	Usage
Authentication	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 Authentication page.



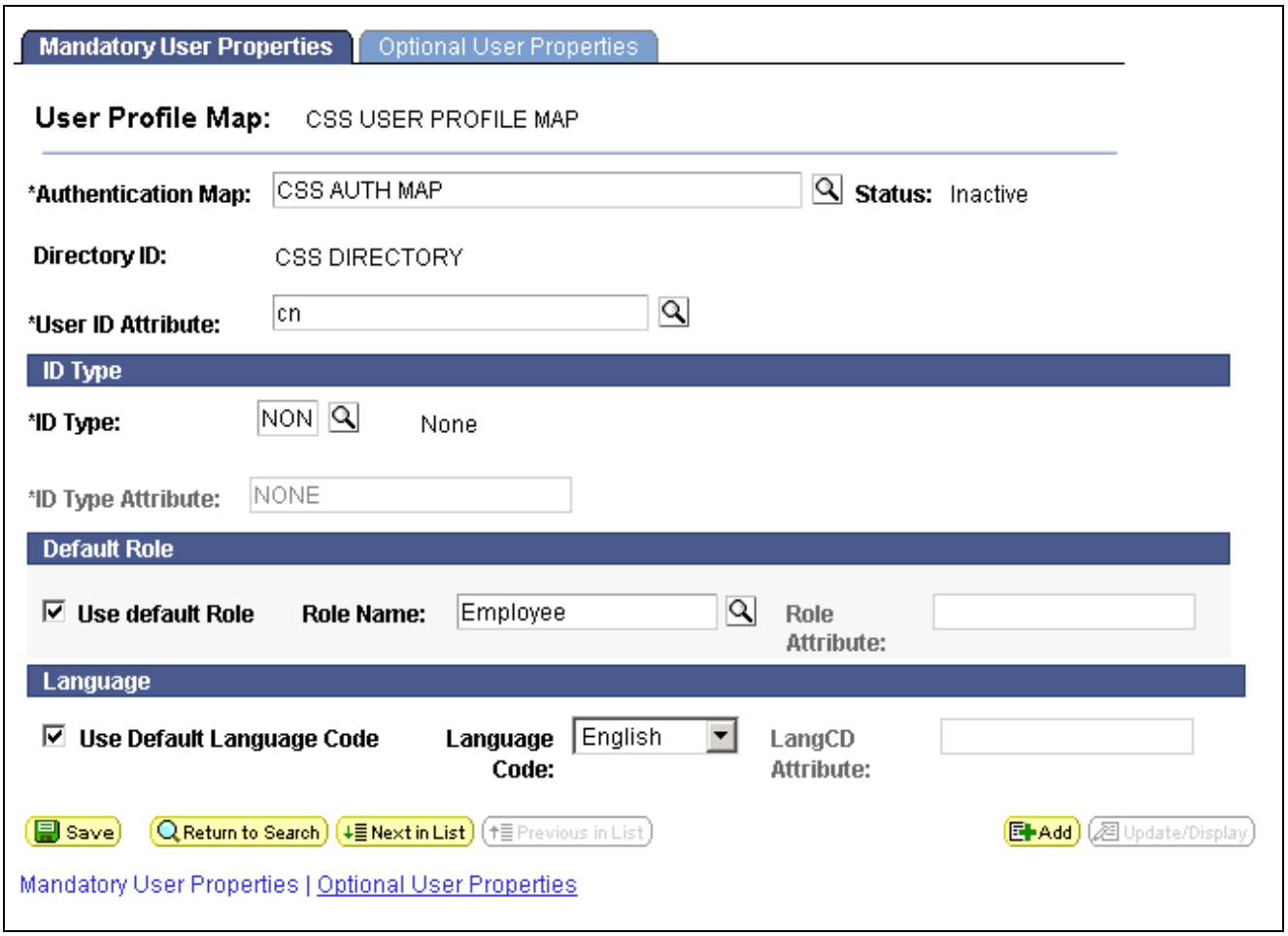
Authentication page

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

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

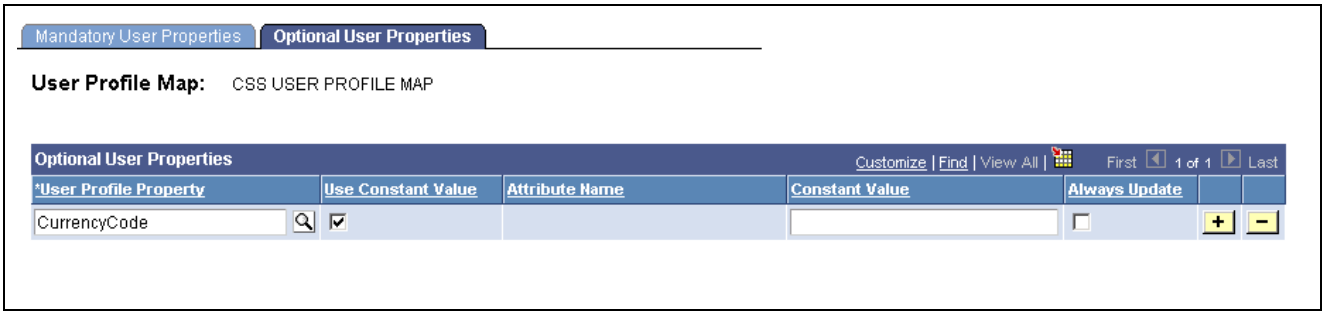
Viewing User Properties

Access the Mandatory or Optional User Properties page.



Mandatory User Properties page

Select the Authentication Map and set check boxes and field values 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

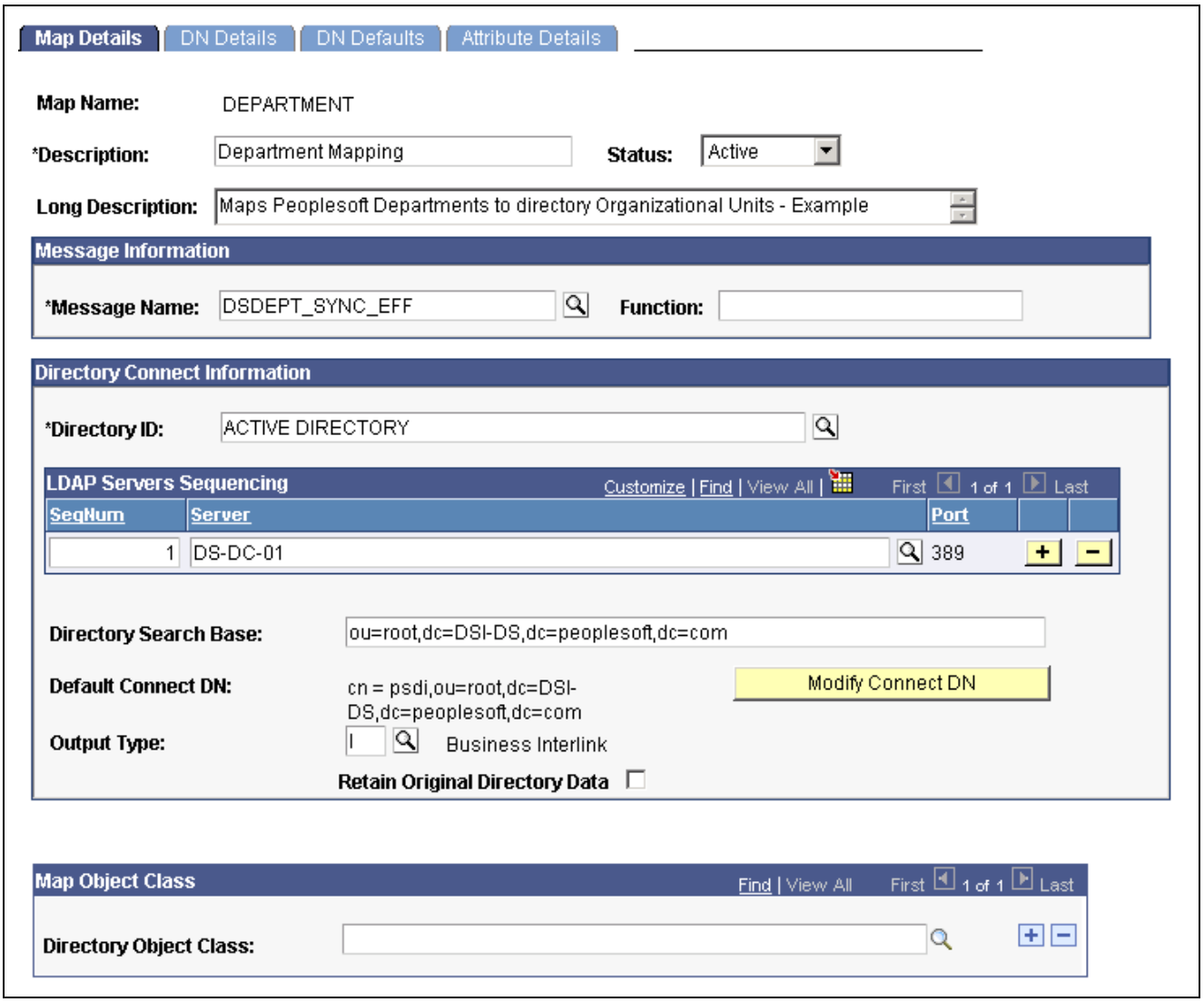
You map PeopleSoft data to the equivalent directory objects to keep the data synchronized. PeopleSoft Directory Interface receives PeopleSoft data from messages published whenever there is a business event associated with the messages identified in the Directory Mapping component. Each message contains 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. Additionally, you can define a constant value or a PeopleCode function that returns a value to populate data used in building temporary Directory Information Trees when not all data exists for an entry.

Pages Used to Set Up Mappings

Page Name	Object Name	Navigation	Usage
Map Details	EO_DSMAP	Enterprise Components, Directory Interface, Mappings, Directory maps, Map Details	Set up a mapping and enter the data relationship details between PeopleSoft data and directory data.
Modify Connect DN - Directory Interface	EO_DSUSERDN	Click the Modify Connect DN button on the Map Details page.	Allows you to modify the Connect DN.
DN Details	EO_DSDN	Enterprise Components, Directory Interface, Mappings, Directory maps, 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.
DN Defaults	EODS_DN_DEFAULTS	Enterprise Components, Directory Interface, Mappings, Directory maps, DN Defaults	Define a constant value or PeopleCode function that returns values that populate the blank values on the Directory Information Tree.
Attribute Details	EO_DSATTRIB	Enterprise Components, Directory Interface, Mappings, Directory maps, Attribute Details	Set up the relationship between the data in the 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

Status Select the appropriate status from the following values.

- *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) 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 Click to access the Modify Connect DN - Directory Interface 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

Access the Modify Connect DN - Directory Interface page.

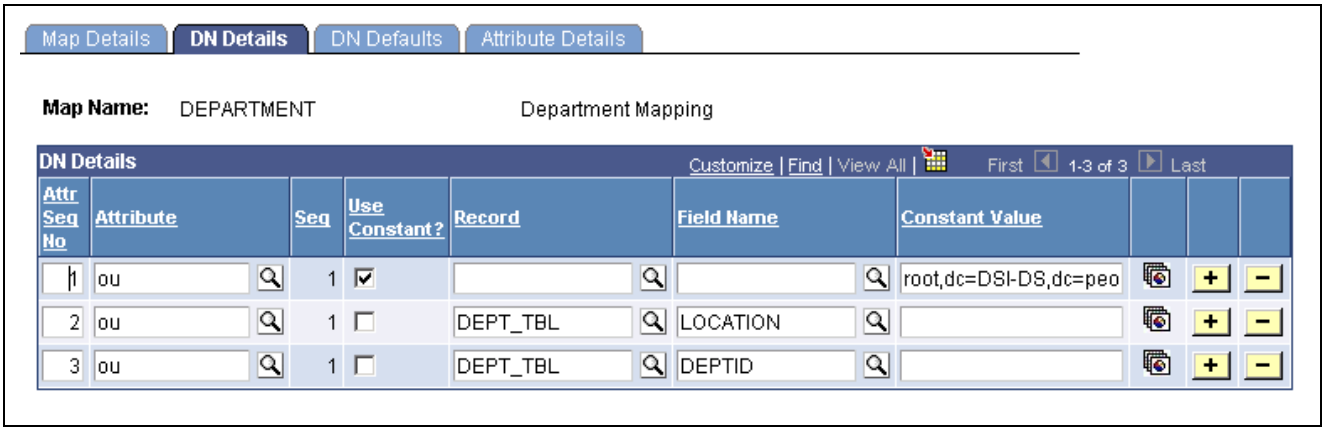
Use Default (Admin) DN (use default [administrative] distinguished name) Select to use the default connect distinguished name value that you set up in PeopleTools.

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 DN Attribute Function - Directory Interface 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 DN Attribute Function - Directory Interface 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):

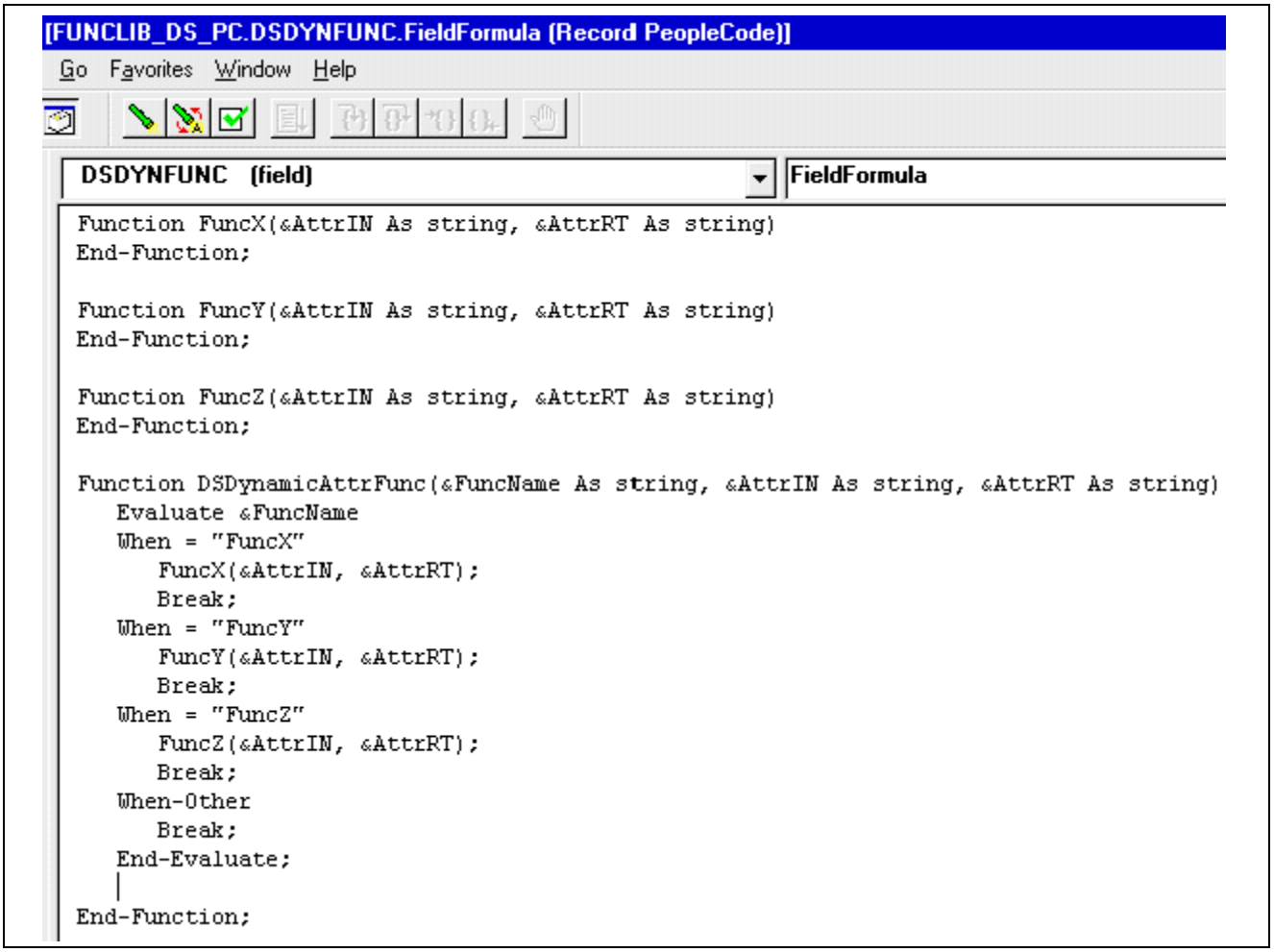
```
When = 'FuncX'
   FuncX(&AttrIn, &AttrRT);
   Break;
```

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

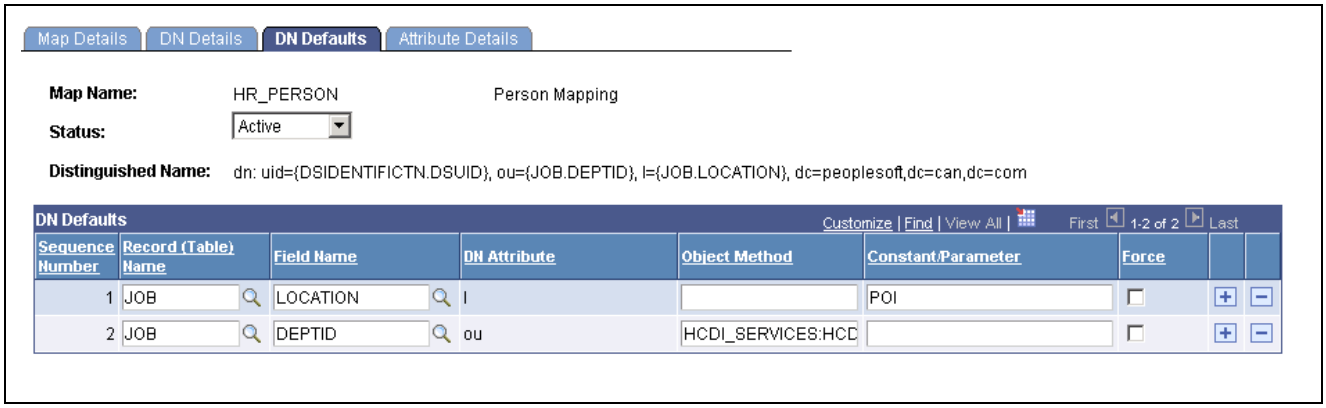
The following example displays the setup for functions FuncX, FuncY, and FuncZ.



Setup functions on the FUNCLIB_DS_PC.DSDYNFUNC FieldFormula

Specifying Distinguished Name Defaults

Access the DN Defaults page.



DN Defaults page

On the DN Defaults page, define a constant value or a PeopleCode method that returns a value to populate blank values when you are building Directory Information Trees. This page enables you to define defaults for any Record.Field value that is left blank in the data rowset of the Application Message that is used to populate the map. For example, you can enter defaults to fill in the blanks left by the lack of Department/Location data for the Persons of Interest constructed by the Directory Interface.

In the preceding screen shot, the value for JOB.DEPTID defaults to a method called HCDI_SERVICES:HCDIUtilities.DeptID. This method returns a DeptID constant.

Note. The syntax for the method needs to be fully qualified using the following format: Package_Name:App_Class_Name.Method_Name.

Seq (sequence number) Displays the sequence number for this attribute.

Record (Table) Name Select the record name for the value. This field is required.

Field Name Select the field name for the value. This field is required.

DN Attribute The name of the Distinguished Name attribute.

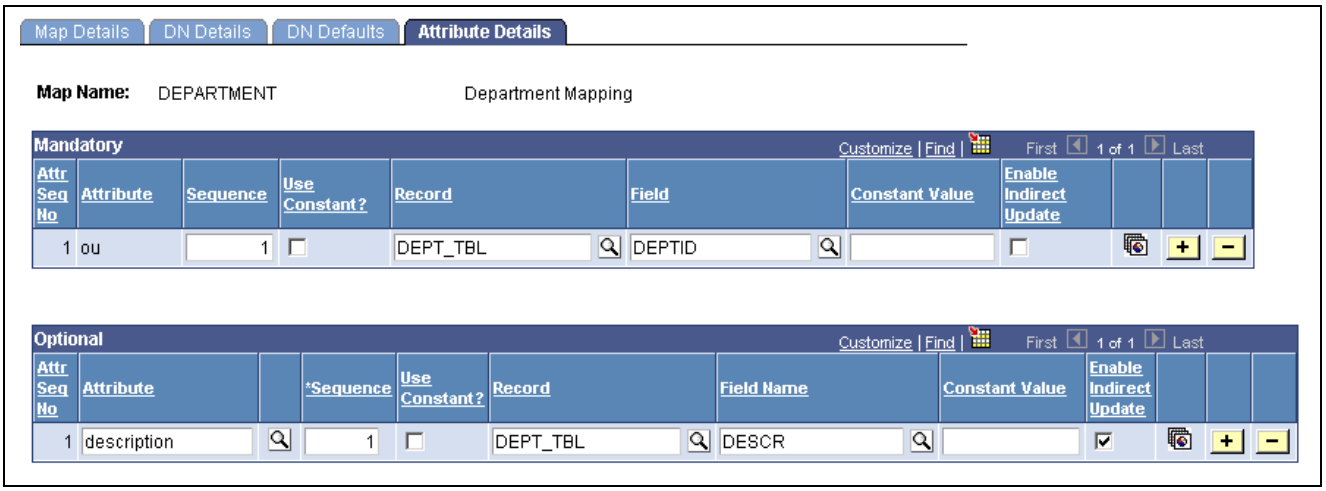
Object Method Enter the object method you are using to populate the value, if applicable. Leave this field blank if you are using a constant or parameter to populate the value.

Constant/Parameter Enter the values for the constant or the parameter, if applicable. Leave this field blank if you are using an object method to populate the value.

Force Select this checkbox to overwrite the Record.Field values at runtime, even if the values exist.

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

This section discusses how to:

- Create entry definitions.
- Specify entry membership rules.

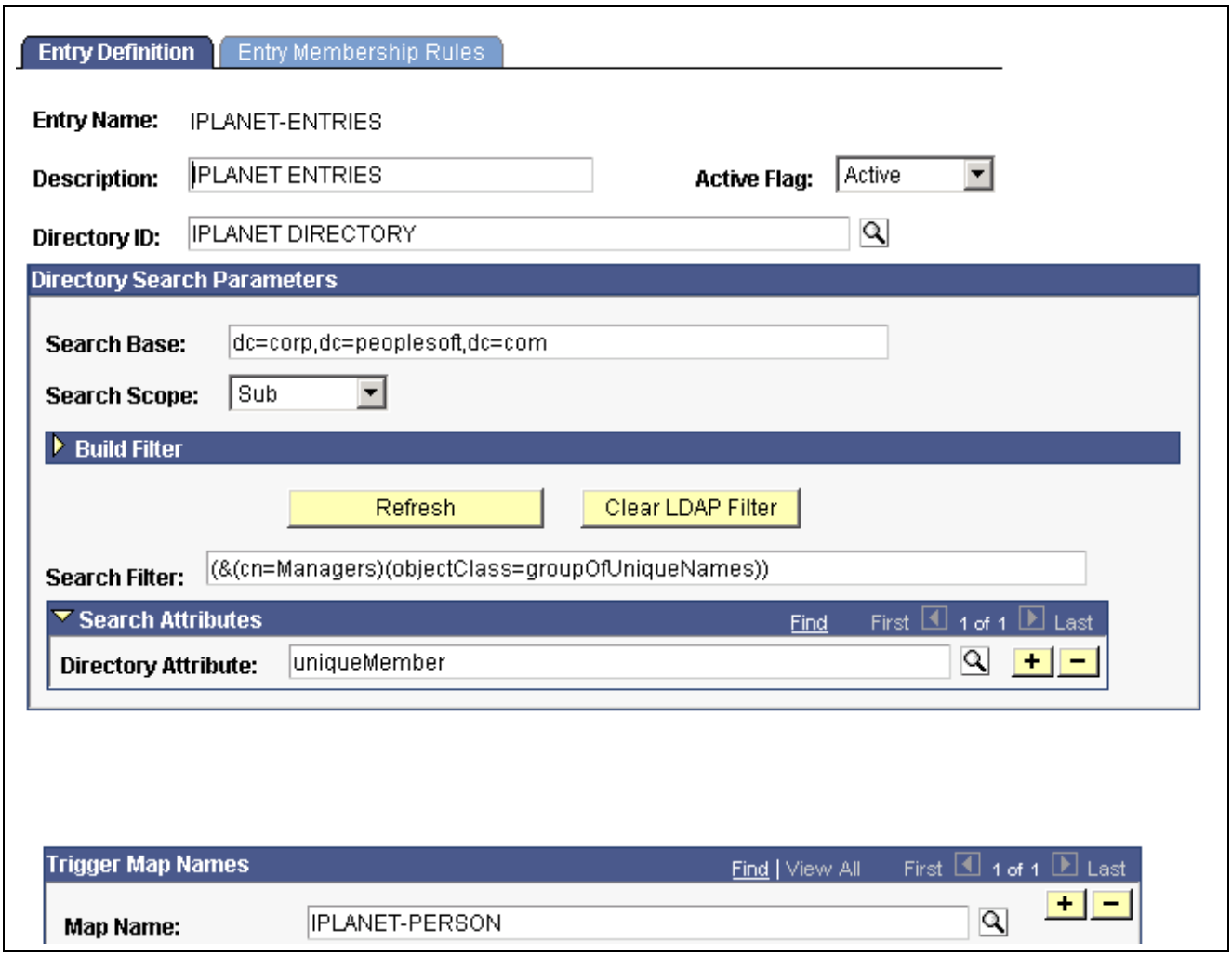
Entry membership rules enable you to modify a directory entry, such as a group, based on criteria stored in the PeopleSoft database. This feature provides a method to match any type of directory entry to rules that are meaningful in PeopleSoft. You can use membership rules to create any type of logical 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, Entry Rules, Entry Definition	Create a directory entry definition.
Entry Membership Rules	EO_DSSECRULES	Enterprise Components, Directory Interface, Membership Rules, Entry Rules, 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

( ) 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 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.

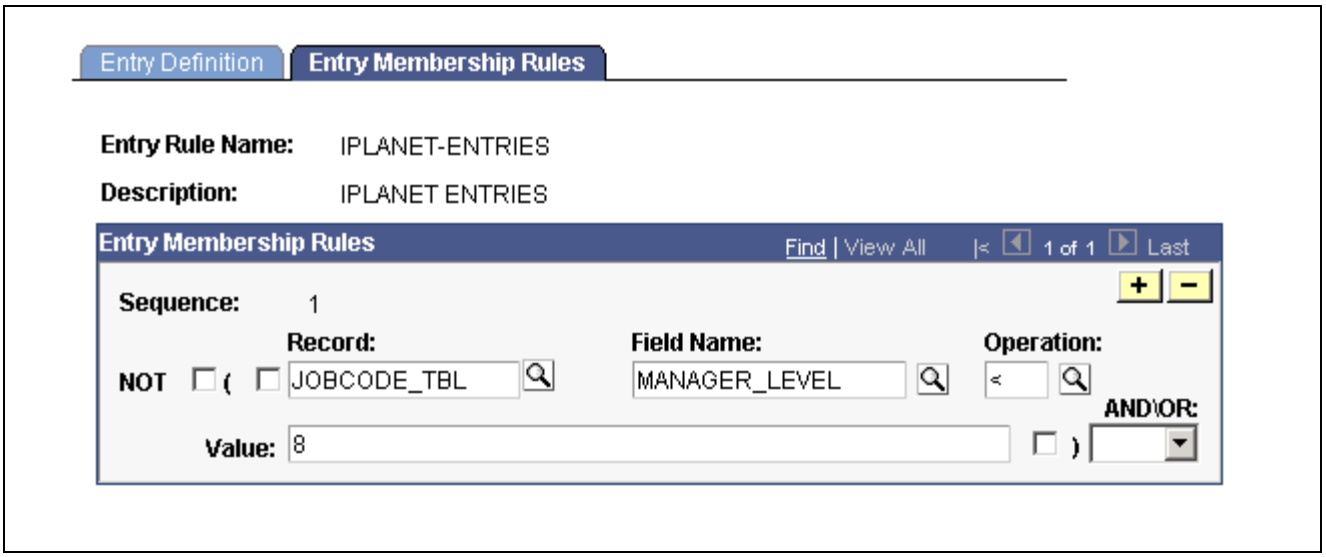
Trigger Message Names

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.

( ) 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 Enterprise HRMS 8.9 Application Fundamentals PeopleBook*

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, Run Directory Audit	Run the Directory Audit process.
Directory Search	EO_DSSRCHDIRECTORY	Enterprise Components, Directory Interface, 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.

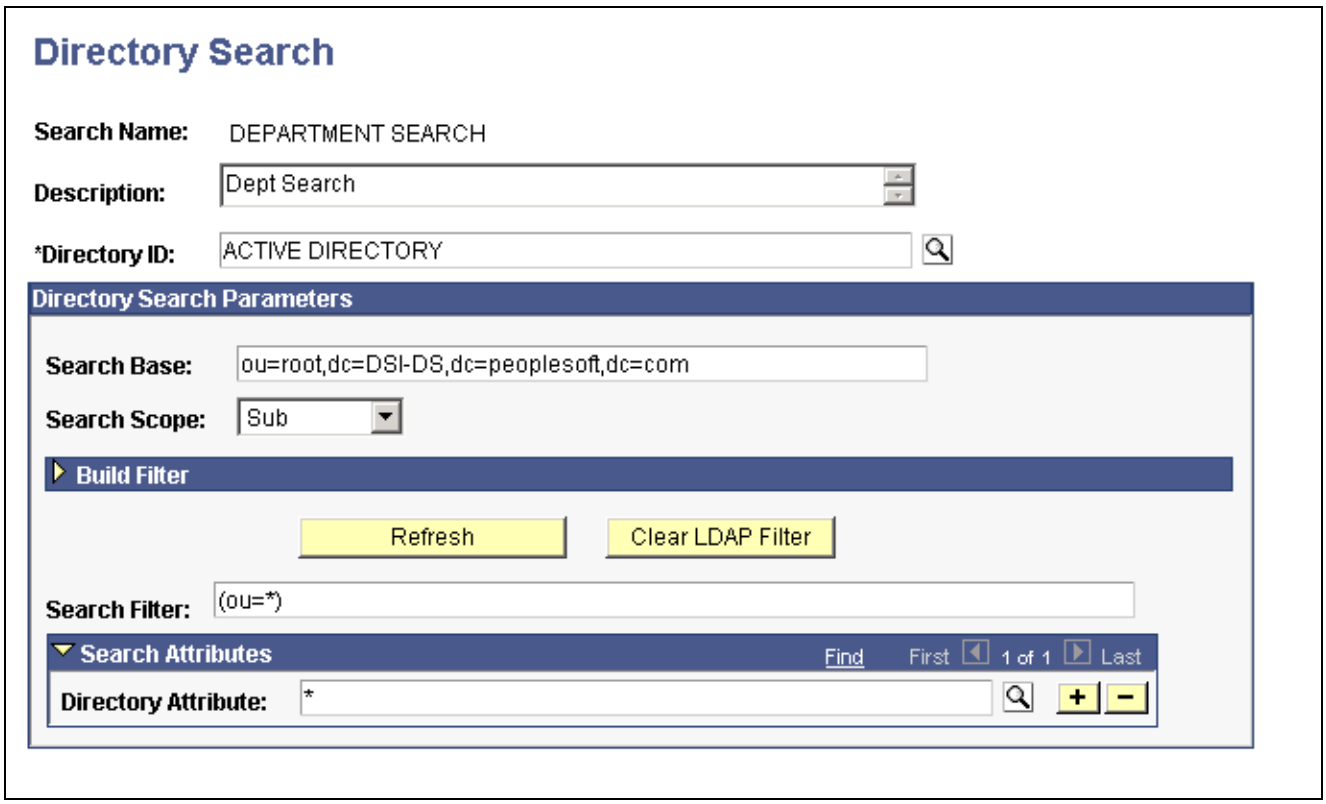
Note. For HRMS customers only, there is an alternative process named DSMAPINPUT FullSync Audit that you can use in place of the Directory Audit process. This new process does not replace the Directory Audit process; it is provided as an alternative to audit the data if performance becomes an issue.

See *PeopleSoft Enterprise HRMS 8.9 Application Fundamentals PeopleBook*

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 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 Build 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.

( ) 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, Bus Interlink Status Report	Generate the Directory BI Status report.

Viewing the Directory Audit Report

Access the Directory Audit Report page.

Directory 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

Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Process Scheduler

PART 3

PeopleSoft Common Objects and Components

Chapter 4
Formatting Addresses

Chapter 5
Working With Currencies and Market Rates

Chapter 6
Using Datasets

Chapter 7
Setting Up the Credit Card Interface

Chapter 8
Using Interactive Reports

CHAPTER 4

Formatting Addresses

This chapter provides an overview of address formats and discusses how to specify address formats.

Understanding Address Formats

You can format addresses for any country in the world. You can create country and state descriptions and street, suite, building, and postal zone formats to meet the needs of any country format. For example, the state description can be changed to county, province, district, or other geopolitical designation used by the country in question. In addition, PeopleSoft contains fully populated country code and state code tables, which we update each major release if national boundaries and designations have changed.

Specifying Address Formats

This section discusses how to:

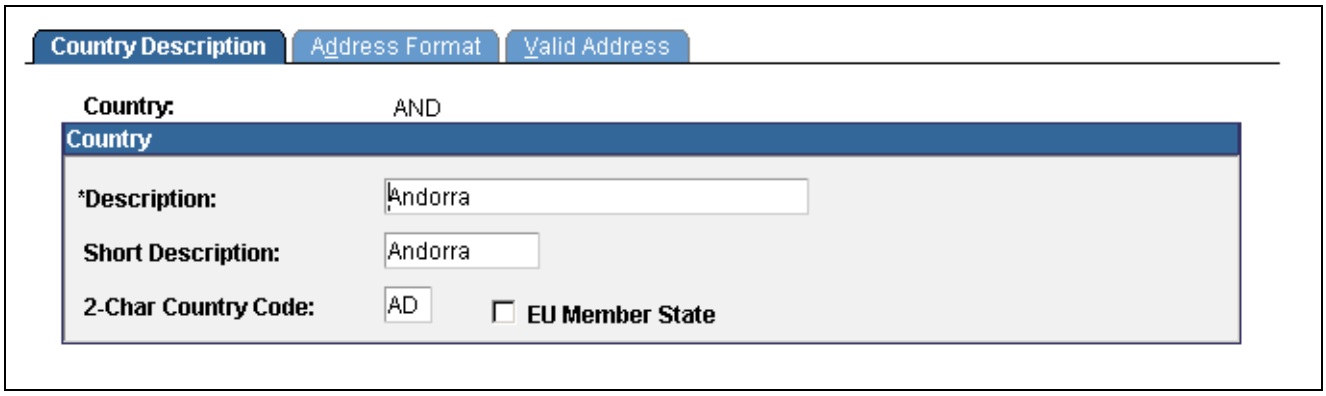
- Describe countries.
- Format addresses.
- Validate addresses.
- Specify state information.

Pages Used to Specify Address Formats

Page Name	Object Name	Navigation	Usage
Country Description	COUNTRY_DEFN	Set Up HRMS, Install, Country Table	Define countries to be used throughout the system. Used mainly for currencies and addresses.
Address Format	ADDR_FORMAT_TABLE	Set Up HRMS, Install, Country Table, Address Format	Customize fields and field descriptions, so addresses conform to the customary address format of the specified country. Once set, the format appears everywhere the system uses the address subrecord.
Valid Address	EO_ADDR_VALIDAT	Set Up HRMS, Install, Country Table, Valid Address	Add valid combinations of address fields.
State or Province Table	STATE_DEFN	Set Up HRMS, Install, State/Province	Add or review a state, province, county, or other geopolitical region within a country.

Describing Countries

Access the Country Description page.



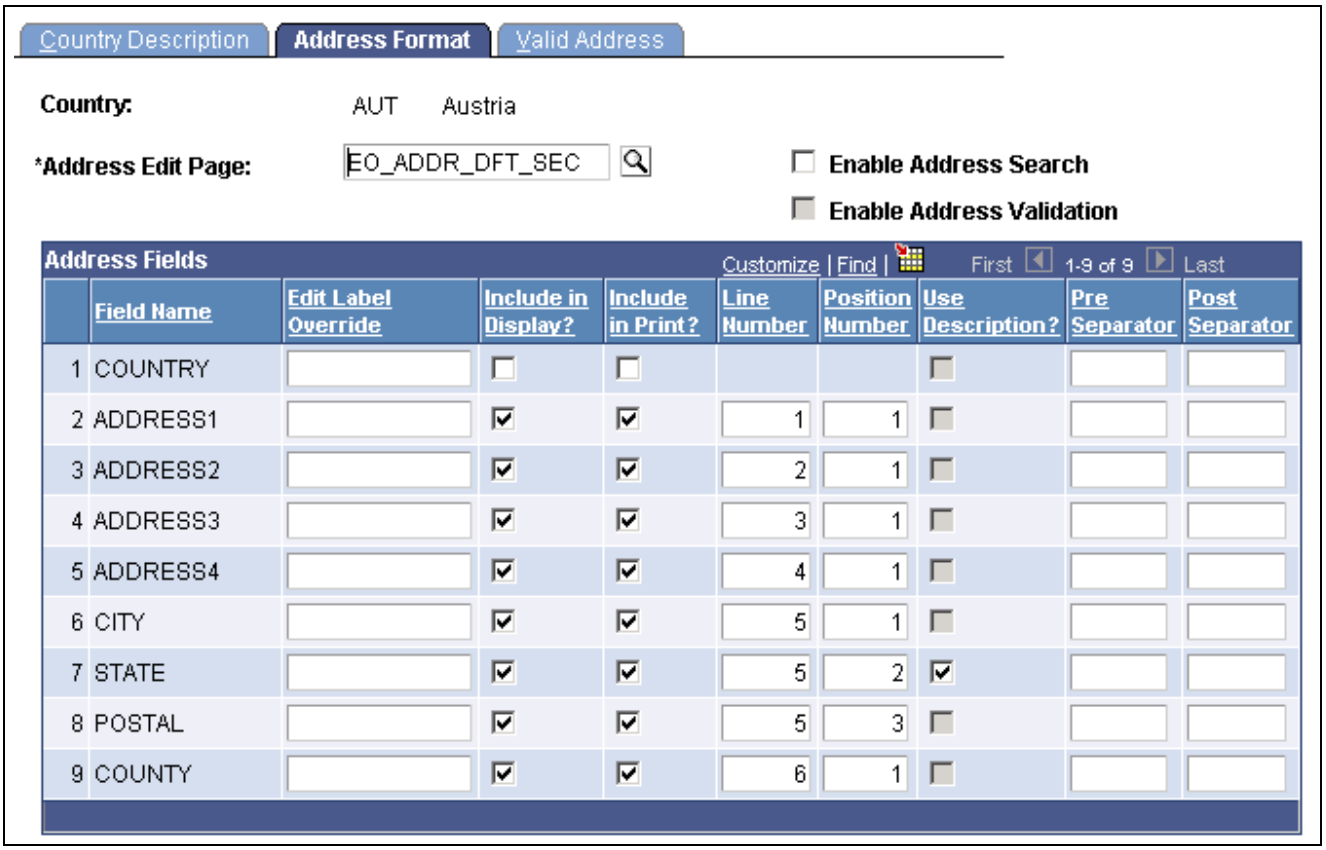
Country Table page

2-Char Country Code(two-character country code) Enter the Value Added Tax (VAT) registration ID for the country..

EU Member State (european union member state) Select if the country is a member of the European Union. Select to include the country in prompts for intrastate reporting.

Formatting Addresses

Access the Address Format page.



Address Format page

Use this page to set up three aspects of the address: editing, displaying, and printing.

Address Edit Page Displays the secondary page used for editing the address. You can create a new secondary page using PeopleSoft Application Designer. On the new address secondary page you create, use page fields from the DERIVED_ADDRESS record definition. You then need to add a secondary page control to ADDRESS_SBP pointing to your new secondary page. Once you complete these steps, the secondary page is accessible in this field.

Enable Address Search Select to enable users to search for a valid value. Selecting this check box enables the Used in Search column and the Valid Address page.

Enable Address Validation Available only when Enable Address Search is selected. Select to ensure validation of all values selected. When selected, users must select a value from the search list. If this check box clear, users can select from the search list or enter a new value.

Field Name Displays the field options available for the address page.

Edit Label Override (Optional) Enter an alternative label for the field. The new label is used when prompting for the field. You can customize address formats so that they conform to the address requirements of each location. For instance, for a U.S. address, you would change the Postal field to read ZIP Code.

Keep in mind the distinctions between county and state:

- *County:* The tertiary geopolitical region within a state; the level after country and state. (In the UK, the level of state is called a county; you would enter such counties in the State field.)
- *State:* The secondary geopolitical region within a country; a state in the U.S., a province in Canada, a county in the UK, and a department in France.

Used in Search Available only when you select the Enable Address Search check box. Select the fields you want users to be able to search on.

Include in Display Select to activate the corresponding address field as part of the standard address format for the selected country. To remove a field, clear the check box. When entering addresses, users enter a country code, and then the system updates the page to display the fields appropriate for that country.

Include in Print Select to print the field when printing reports.

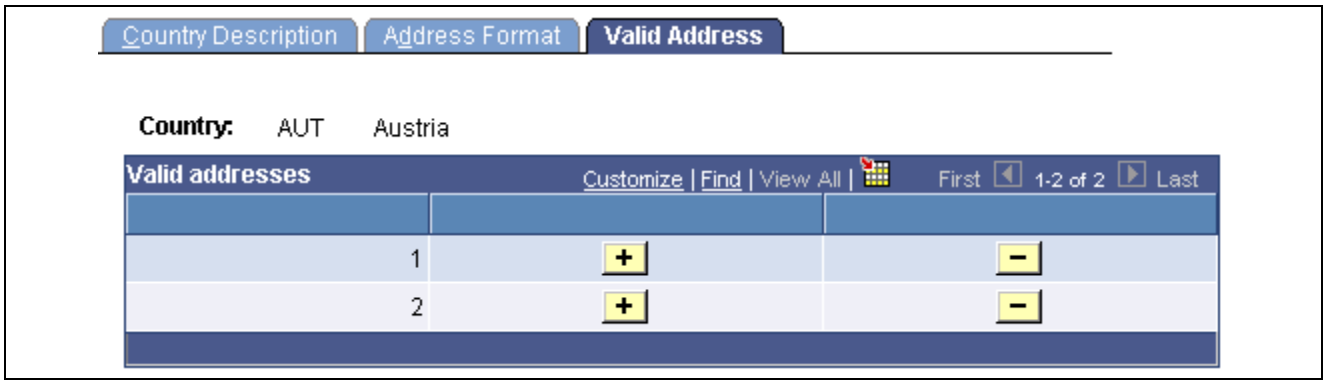
Line Number and **Position Number** Enter the physical location of the fields for displaying and printing.

Use Description Select to display the description for the field value. For example, for addresses in Japan, select this option to display the description of the state rather than the state code, since the code is numeric.

Pre Separator and **Post Separator** Enter characters to be used surrounding the address field. For example, in the United States, a comma generally follows the city name, such as in *San Francisco, CA*. In India, there are parentheses around the postal code, for example *(123)*.

Validating Addresses

Access the Valid Address page.

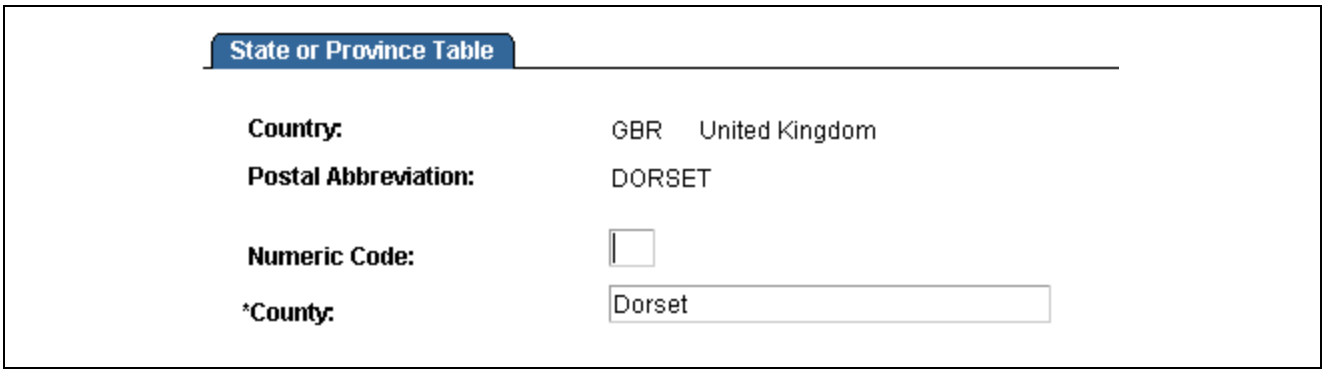


Valid Address page

To enable this page, select the Enable Address Search check box on the Address Format page, which enables the Used in Search column. The fields you selected to be used in search appear on the Valid Address page as columns. Enter the valid address combinations that the user can search for and select.

Specifying State Information

Access the State/Province page.



State Table page

This table provides states, provinces, and equivalent geopolitical entities for all supported countries, such as Dutch communities and French departments. The codes are based on standard postal codes.

Enter a numeric code for statistical and reporting purposes. The second enterable field, County in the preceding example, changes depending on the country. If the country is France, the field label appears as Department.

CHAPTER 5

Working With Currencies and Market Rates

This chapter provides an overview of currencies and market rates and discusses how to:

- Define currencies.
- Define currency quotation methods.
- Define market rates.
- Calculate cross and reciprocal rates.
- Use the Currency Exchange Calculator.

Understanding Currencies and 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 converting currency throughout PeopleSoft applications, and for defining and storing market rates. Market rate is a generic term for a currency exchange rate, an interest rate, or a future rate, for example.

In this section, we discuss:

- Currency and market rate tables.
- Triangulation.
- Conversion factor fields and the visual rate.
- Application-specific requirements for currency conversion.

Understanding Currency and Market Rate Tables

The following tables store currency and market rate data:

- CURRENCY_CD_TBL

  Stores currency code data.

- CURR_QUOTE_TBL

  Stores currency quotation method data.

- RT_INDEX_TABLE

  Stores market rate index data.

- RT_TYPE_TBL

  Stores rate type data.

- RT_RATE_TABLE

  Stores market rate data.

Understanding Triangulation

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

To support triangulation, PeopleSoft provides a means to define that you want a currency pair to triangulate through a fixed reference currency. The actual conversion process is done in a two-step procedure in which 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 or possibly three exchange rates that are relevant to the conversion. When viewing a triangulated conversion at a detailed 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. However, you can 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 run the EOP_RATECALC Application Engine 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	Direct /Triangulate /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.

See Also

Chapter 5, "Working With Currencies and Market Rates," Calculating Cross, Triangulated, and Reciprocal Rates, page 65

Understanding 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	Direct /Triangulate /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	Indirect /Triangulate /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 with units) = (1000 CHF / 100) × 119.335 = 1193.35 DEM

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

See Also

Chapter 5, "Working With Currencies and Market Rates," Defining Currency Quotation Methods, page 54

Understanding 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

Enterprise PeopleTools 8.45 PeopleBook: Global Technology

Enterprise PeopleTools 8.45 PeopleBook: PeopleCode Developer's Guide

Defining Currencies

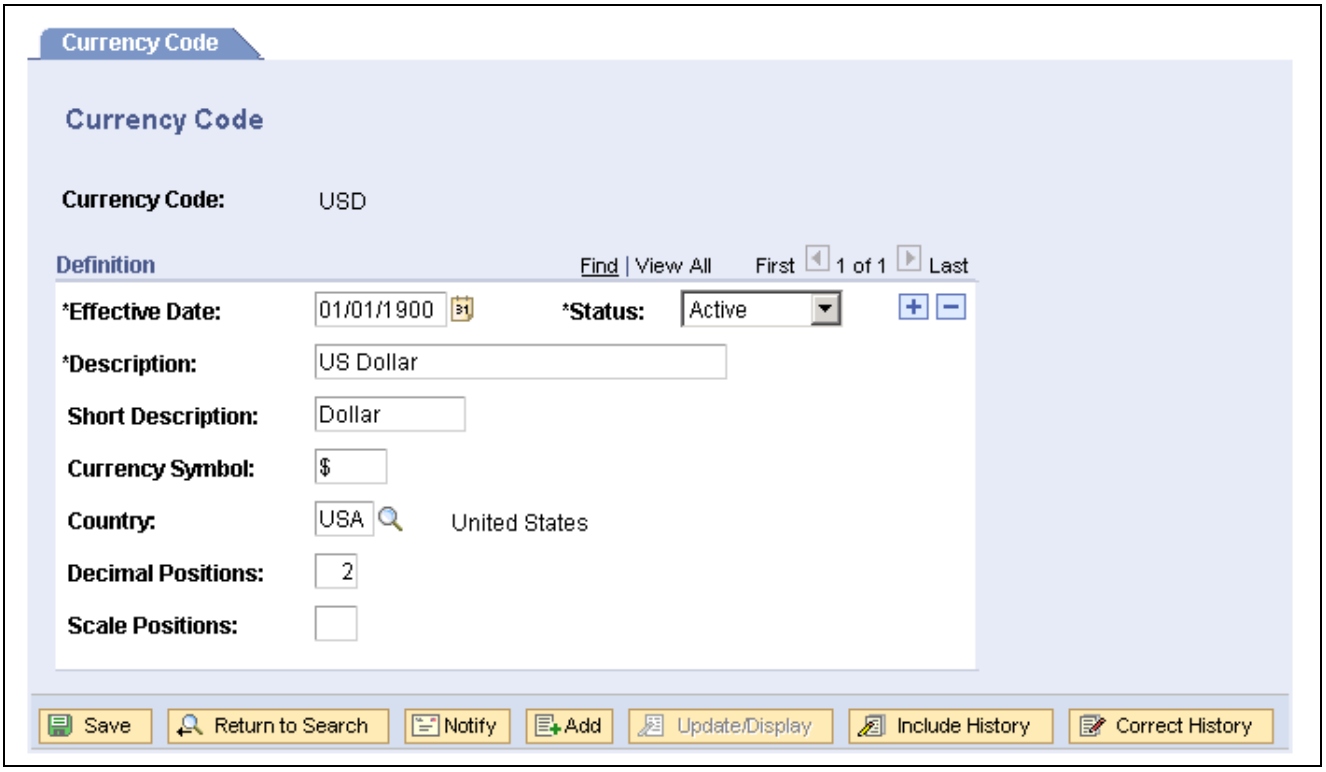
In this section, we discuss how to define currencies.

Page Used to Define Currencies

Page Name	Object Name	Navigation	Usage
Currency Code	CURRENCY_CD_TABLE	Depends on application.	Add and maintain currency codes. These currency codes are used to designate currencies throughout your PeopleSoft system.

Defining Currency Codes

Access the Currency Code page.



Currency Code page

Status Indicate whether the currency code is active or inactive. If you inactivate a currency code that is in use, existing transactions are unaffected. However, the currency code is unavailable for future selections.

Some PeopleSoft applications do not allow you to inactivate a currency code that is in use.

Currency Symbol PeopleSoft delivers many currencies with a currency symbol such as $ for Australian dollar (AUD) or £ for British pound (GBP). You can enter new symbols for delivered currencies or for currencies that you might add.

Country Select the code for the country from which the currency originates.

Note. PeopleSoft delivers fully populated country, state, and province code tables and updates these tables as national boundaries and designations change.

Decimal Positions Enter the number of decimal positions that should appear in the notation for the currency. For example, there are two decimal positions for Australian dollars (500.00 AUD), but no decimal positions for Japanese yen (500 JPY).

Scale Positions Enter the scale positions you want to round for this currency. This controls how many numbers appear to the left of the decimal when displayed. The data is actually stored with full precision in the database itself.

For example, if you want all million-dollar amounts displayed as the number of millions without the zeros, enter 6 as your scale position. In this case, 24,000,000 is displayed as 24, but is stored in the database as 24,000,000.

Defining Currency Quotation Methods

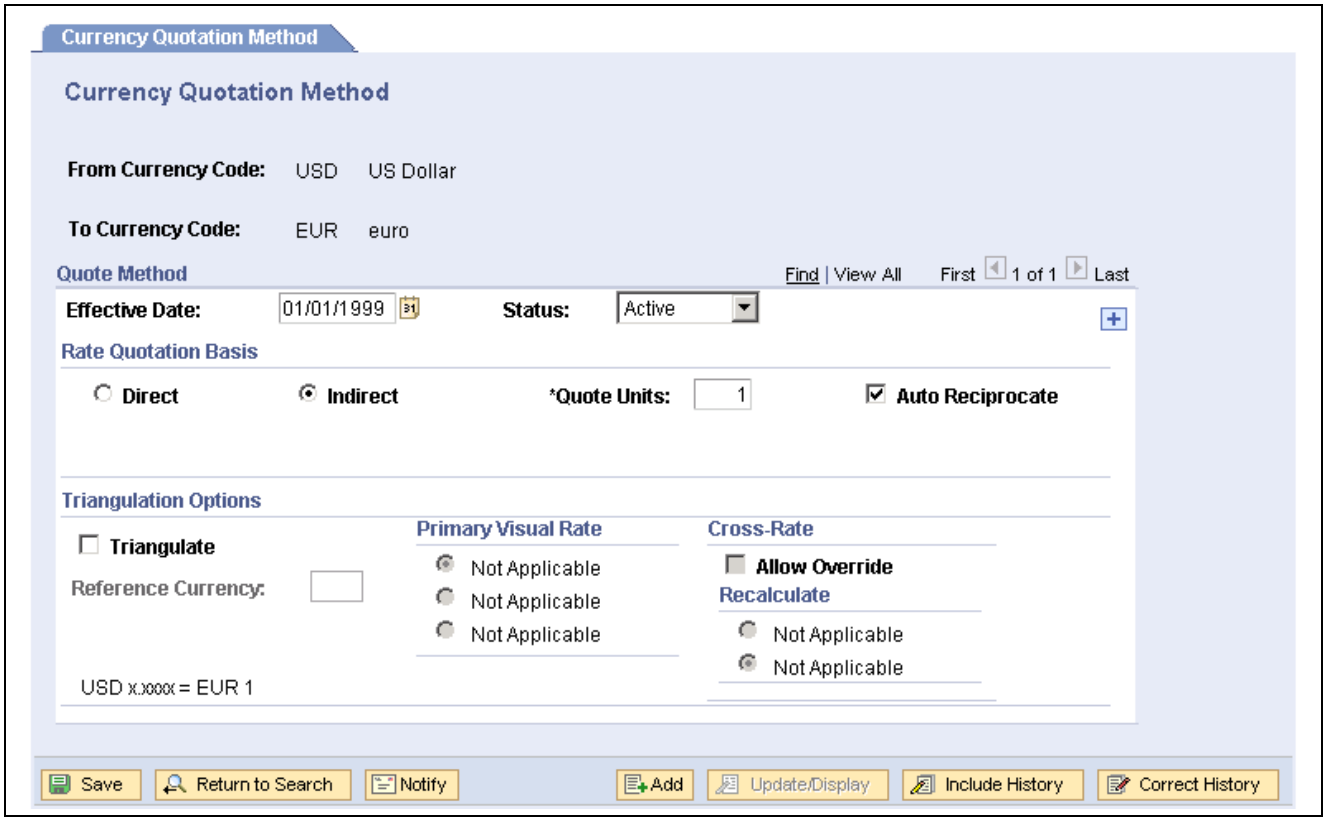
In this section, we discuss how to define currency quotation methods.

Page Used to Define Currency Quotation Methods

Page Name	Object Name	Navigation	Usage
Currency Quotation Method	CURR_QUOTE_PNL	Depends on application.	Set up and maintain a currency quotation method for each from-currency and to-currency pair.

Defining Currency Quotation Methods

Access the Currency Quotation Method page.



Currency Quotation Method page

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.

See Chapter 5, "Working With Currencies and Market Rates," Understanding Conversion Factor Fields and the Visual Rate, page 51.

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.

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.

See Chapter 5, "Working With Currencies and Market Rates," Defining Market Rates, page 56.

Direct and **Indirect**

Indicate whether the rates for this currency pair are quoted directly or indirectly. For example, when defining a currency quotation method for USD and FRF:

- Select *Direct* if you want 1 USD to equal x.xxxx FRF.
- Select *Indirect* if you want x.xxxx USD to equal 1 FRF.

Even currency quotation methods for currency pairs that triangulate must be classified as either direct or indirect. In this case, the value is used to display the calculated cross rate.

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.

Quote Units

Enter a quote unit for the exchange rate, as is common business practice for some currencies. This field can have any value, but is usually a power of 10.

Sometimes called scaling factors, quote units 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.

Auto Reciprocate

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

For example, if you create a currency quotation method for USD to EUR. The reciprocal currency quotation method for EUR to USD is automatically created, regardless of this setting.

When you create a rate for USD to EUR on the Market Rates page, the EUR to USD reciprocal rate is automatically created if this Auto Reciprocate option is selected for the currency pair.

If the either rate for the currency pair is updated on the Market Rates page, the reciprocal rate is updated as long as the Auto Reciprocate option is selected for one of the currencies in the pair.

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 you want to be the primary visual rate. This is the rate displayed on the primary pages and reports. For online applications, other components of the rate can be viewed and modified on the Exchange Rate Detail page.

Allow Override For triangulated currency pairs, select to enable users to override the cross rates on the Market Rates page and Exchange Rate Detail page.

Recalculate If the Allow Override option is selected, select to indicate which of the two other rates should be recalculated to bring the triangle back into balance again. Because the triangulated rate is initially a calculated rate, if you allow it to be overridden, the rates used to initially calculate it must be recalculated.

Defining Market Rates

In this section, we discuss how to:

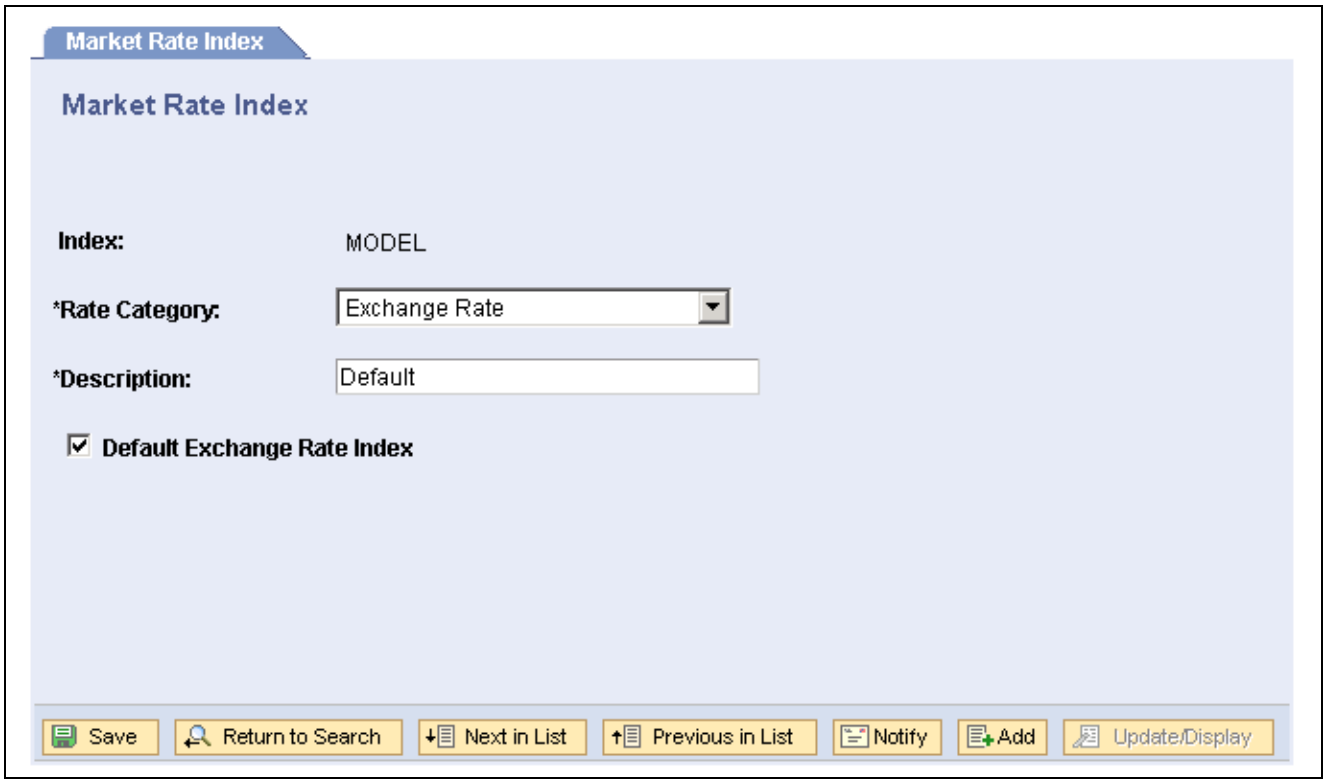
- Define market rate indexes.
- Define market rate types.
- Create market rate definition.
- Define market rates.
- Access rate definition details.
- Access exchange rate details.

Pages Used to Define Market Rates

Page Name	Object Name	Navigation	Usage
Market Rate Index	RT_INDEX_TBL	Depends on application.	Create market rate indexes, which provide a means of organizing market rates in the PeopleSoft system.
Rate Type	RT_TYPE_TBL	Depends on application.	Define rate types that further categorize market rates. Examples of rate types include current, commercial, floating, average, and historical.
Market Rate Definition - Rate Definition	RT_RATE_DEF_TBL	Depends on application.	Define tolerance limits for rates and determine what action occurs if a new rate falls outside the tolerance limit.
Market Rates	RT_RATE_PNL	Depends on application.	Maintain and view market rates. The fields available on the page vary depending on the rate category.
Rate Definition	RT_RATE_DEF_SEC	Select the Rate Definition link on the Market Rates page.	View market rate definition details, including the maximum variance and error handling definitions specified for the currency pair on the Rate Definition page.
Exchange Rate Detail	EXCH_RT_DTL	Click the Exchange Rate Detail button on the Market Rates page.	Access exchange rate detail information.

Defining Market Rate Indexes

Access the Market Rate Index page.



Market Rate Index page

Market rate indexes are stored in the RT_INDEX_TABLE table.

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 selected market rate index is the default exchange rate index. This field is available only if:

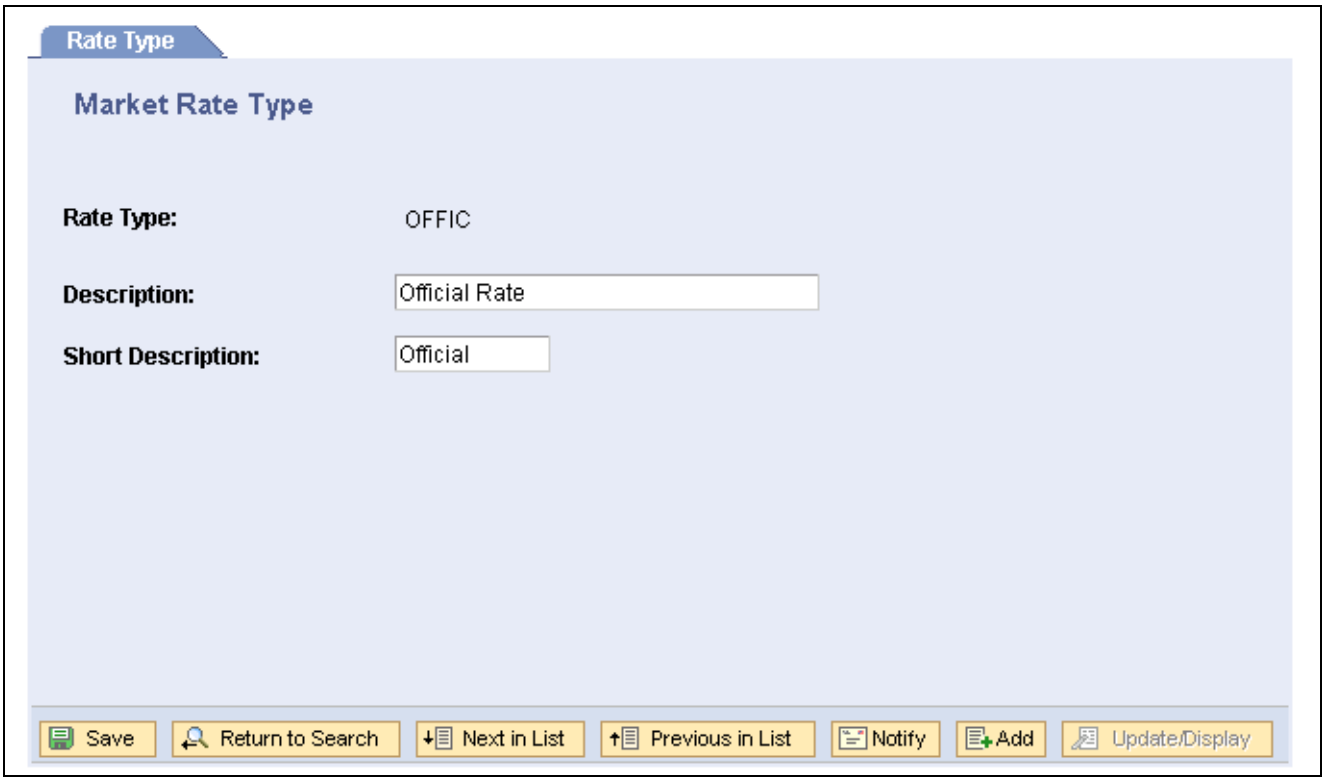
- The Rate Category field is set to *Exchange Rate.*
- No other index is currently defined as the default exchange rate index.

The Market Rates Index page does not ensure that a default market rate index has been defined. However, if no default has been defined, the Market Rate Default view does not return any data.

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 an index defined as the default exchange rate index.

Defining Rate Types

Access the Rate Type page.



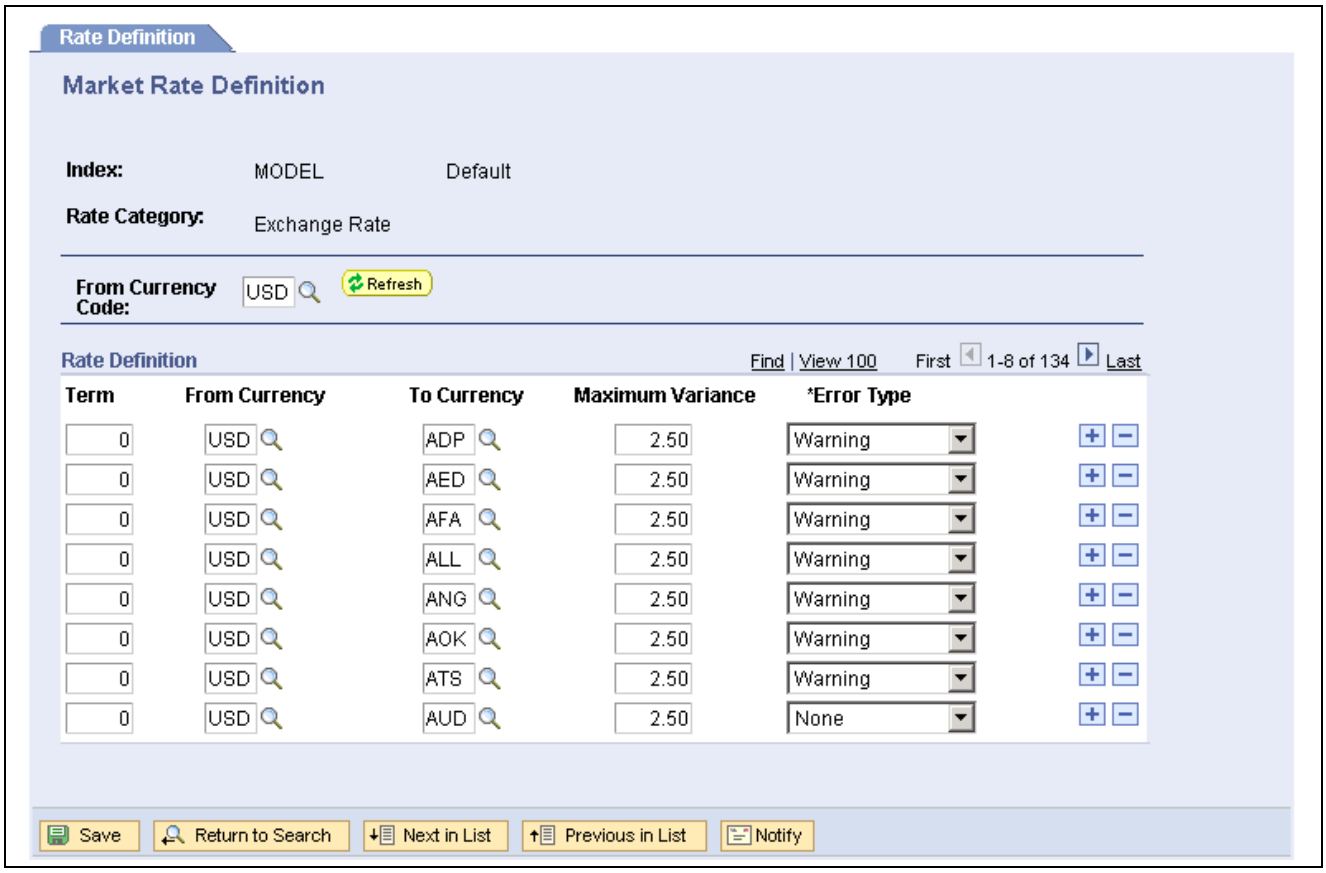
Rate Type page

Rate types are stored in the RT_TYPE_TBL edit table. Rate types serve as categories within a market rate index. For example, some common types of exchange rates are official rate, spot rate, and free market rate.

Enter a description and short description to define each market rate type that you use.

Creating Market Rate Definitions

Access the Market Rate Definition - Rate Definition page.



Market Rate Definition - Rate Definition page

Market rate definitions specify the valid term, currency, and other appropriate field combinations for market rates. For example, if you have a market rate definition for an exchange rate with a term of 30, a from-currency of CHF, and a to-currency of USD, you can enter a rate using this combination on the Market Rates page.

If you have not created a market rate definition on this page when you create the a market rate on the Market Rates page, the system automatically creates one for you using the default values of 2.5 percent maximum variance and warning message processing.

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.

Note. The information you see on this page depends on the selected market rate index. For example, if you select an index associated with a rate category of *Interest Rate,* fields on this page display interest-related data.

From Currency Code Enter the from-currency code with which you want to populate all From Currency fields on the page.

Refresh Click to populate the From Currency field with the currency you selected in the From Currency Code field.

Term Enter the desired term expressed in days. A zero term indicates that the spot rate = zero term. Only PeopleSoft Treasury uses non-zero terms; all other applications must use a zero term for spot rate.

From Currency In addition to using the From Currency Code field to populate all From Currency field on this page, you can also manually enter the appropriate

from-currency. This value is used with its associated To Currency field value as part of an exchange rate pair. When you use triangulation, include a definition row for each of the currency pairs involved in the triangulation.

To Currency Enter the appropriate to-currency. This value is used with its associated From Currency field value as part of an exchange rate pair.

Currency This field displays when you are working with a rate definition with a rate category set to *Interest Rate*.

Select the currency for which you are creating an interest rate definition.

The From Currency and To Currency fields do not display.

Day Count Basis This field displays when you are working with an interest rate definition.

Select an interest basis:

30/360

30E/360

Actual/360

Actual/365

Actual/Actual

Maximum Variance Enter the percentage of variance that is allowed when a user maintains or overrides a market rate. If the change exceeds the tolerance, an error results. The default value is 2.50 (2.5%).

Error Type Select the type of error that results when the defined maximum variance is exceeded during data entry.

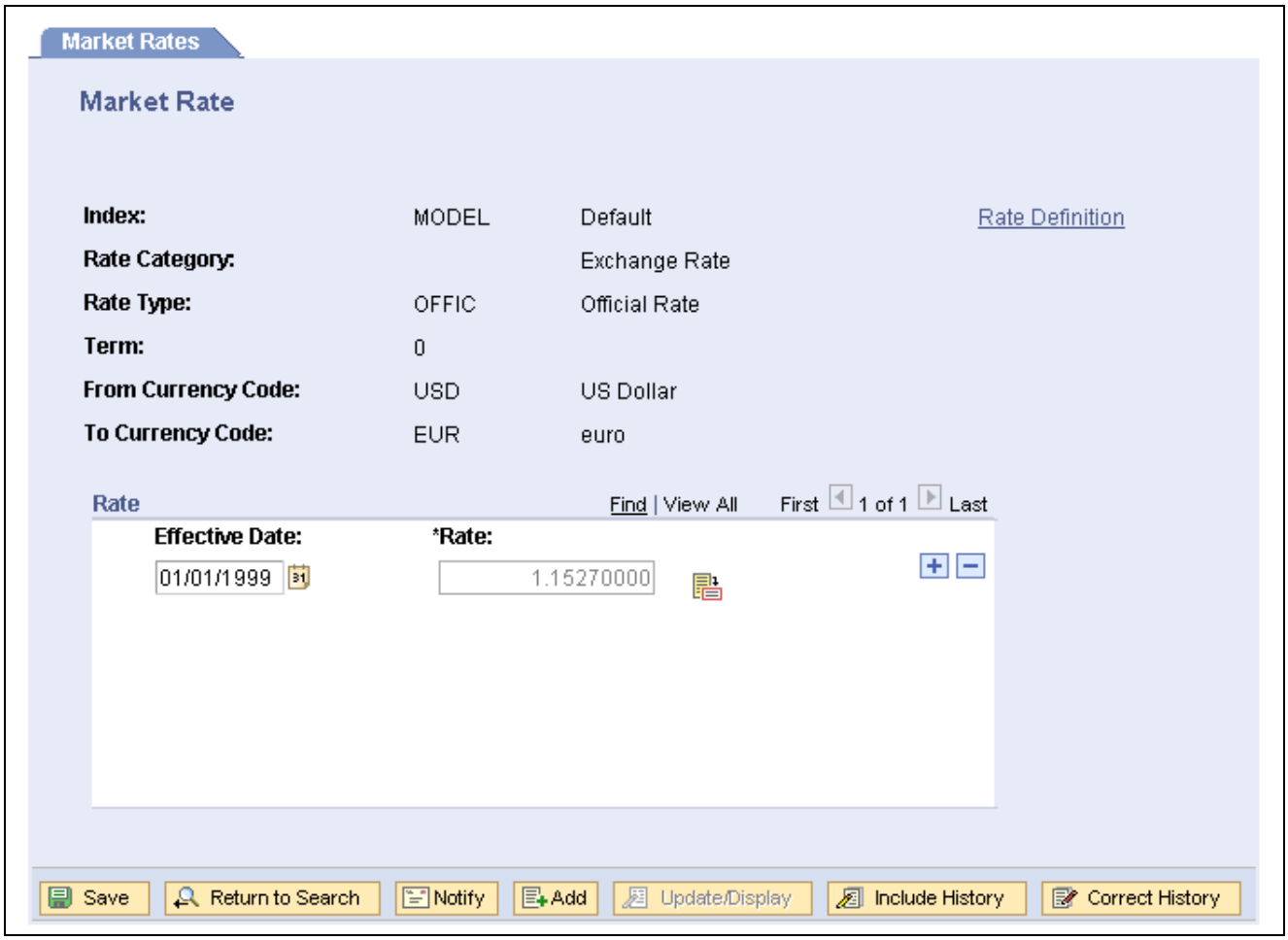
None. No error processing occurs and the new rate is used, even if it exceeds the maximum variance.

Stop. Processing halts and the system prevents you from saving the new rate.

Warning. This is the default value. A warning appears that you can ignore and proceed to save the new rate.

Defining Market Rates

Access the Market Rates page.



Market Rates page

The data you enter on this page is stored in the RT_RATE_TABLE table, which is the common repository for all types of market rates including exchange rates and interest rates.

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 option is clear for the exchange rate's quotation method on the Currency Quotation Method page.

Note. When working with interest rates, the From Currency Code and To Currency Code fields may contain the same field value.

Rate

Rate

Displays the visual rate. If you are working with a triangulated exchange rate, this field displays the primary visual rate, which is typically the cross rate, but can also be one of the other component rates of the triangle.

During online maintenance of market rates, you don't view or change RATE_MULT and RATE_DIV values directly, but instead access this visual rate, which is calculated by page logic based on RATE_MULT, RATE_DIV,

and the currency quotation method defined for the currency pair on the Currency Quotation Method page. The visual rate is stored temporarily on a page work record.

Exchange Rate Detail Click to access the Exchange Rate Detail page, where you can view all three visual rates of a triangulated exchange rate.

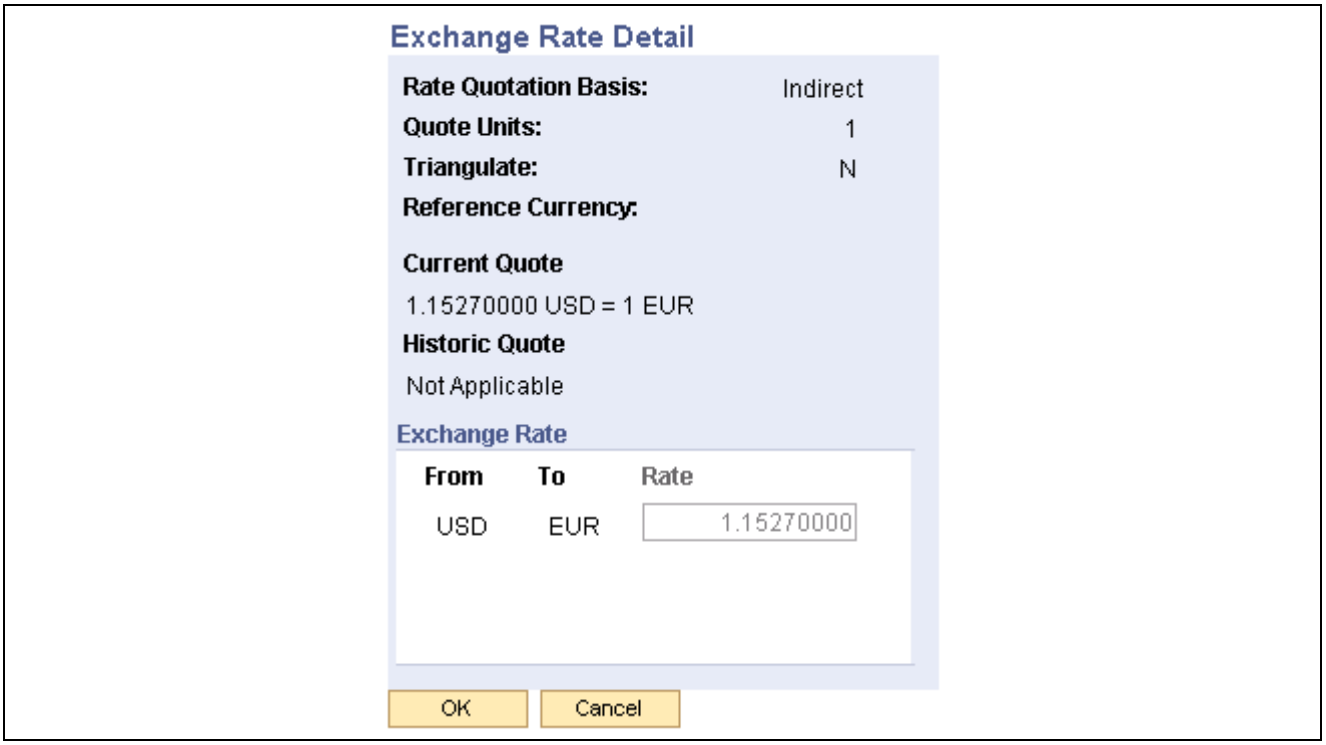
If a quotation method has been defined for the currency pair and the Auto Reciprocate option for the currency quotation method is selected, then creating or maintaining a rate for a currency pair on this page automatically creates or updates the rate for the reciprocal currency pair. For example, if you change the USD-to-GBP rate, the GBP-to-USD rate is automatically updated. You can only auto-reciprocate currency pairs for which currency quotation methods have been defined on the Currency Quotation Method page.

See Chapter 5, "Working With Currencies and Market Rates," Defining Currency Quotation Methods, page 54.

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

Accessing 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. For triangulated rates, you can update rate values for all three components of the triangulated rate.

Rate Quotation Basis Displays the quotation basis for the exchange rate as defined in the Currency Quotation Method page.

Quote Units Displays the quote units for the exchange rate as defined in the Currency Quotation Method page.

Triangulate Displays the triangulation setting for the exchange rate as defined in the Currency Quotation Method page.

Reference Currency For triangulated exchange rates, 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.

For a direct, non-triangulated rate, this field displays quote units (or 1) to the left side of the equal sign and the visual rate on the right. For example:

1 USD = 1.40000000 CAD

For an indirect, non-triangulated rate, this field displays the visual rate to the left of the equal sign and quote units (or 1) on the right. For example:

1.400000000 CAD = 1 USD

For a triangulated rate, this field displays the two component rates of the triangle: the rate for converting the from-currency to the reference currency (USD to EUR) and the rate for converting the reference currency to the to-currency (FRF to EUR). For example:

1.25 USD = 1 EUR = 6.8 FRF

Historic Quote If page logic determines that the exchange rate, as stored in the database, is inconsistent with the current quotation method, this field displays a quote based on the current quotation method, instead of the quotation method active on the rate effective date.

Data provided in the historic quote field allows you to see how the exchange rate has changed over time, using a consistent quotation method, even if the quotation method has actually changed.

For example, if you are viewing a historical rate where FRF was converted to USD directly using a calculated reciprocal rate of 1 FRF = 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.

This field also displays a quote if the historic quote method was non-triangulated and the current quote method is triangulated.

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.

If the system determines that the exchange rate is consistent with the current quotation method, the field displays *Not Applicable.*

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 option box is selected for the exchange rate on the Currency Quotation Method page.

Calculating Cross, Triangulated, and Reciprocal Rates

In this section, we discuss how to run the EOP_RATECALC Application Engine process to calculate cross, triangulated, and reciprocal rates.

Understanding the EOP_RATECALC process

Run the EOP_RATECALC process to calculate rates and update the market rates table.

The process performs three functions:

- Generates cross rates for non-triangulated currency pairs.

  For example, an organization subscribes to a rate service that provides all rates respective to USD. Starting with a USD to Canadian dollar rate and a USD to Mexican peso rate, the system can calculate a new Canadian dollar to Mexican peso cross rate.

- Generates triangulated rates for triangulated currency pairs.

  For example, the EUR to an EPC (euro participating currency) fixed rate has been established on the market rate table and a new EUR to USD rate has just been entered. Using this information, the process can create a new USD to EPC triangulated rate. The difference between triangulated rates and cross rates affects how the data is stored in the database. When calculating a cross rate, you actually create a new rate. When calculating a triangulated rate, the individual components of the source rates are stored on the target.

- Generates reciprocal rates for those currency pairs that are not automatically reciprocated.

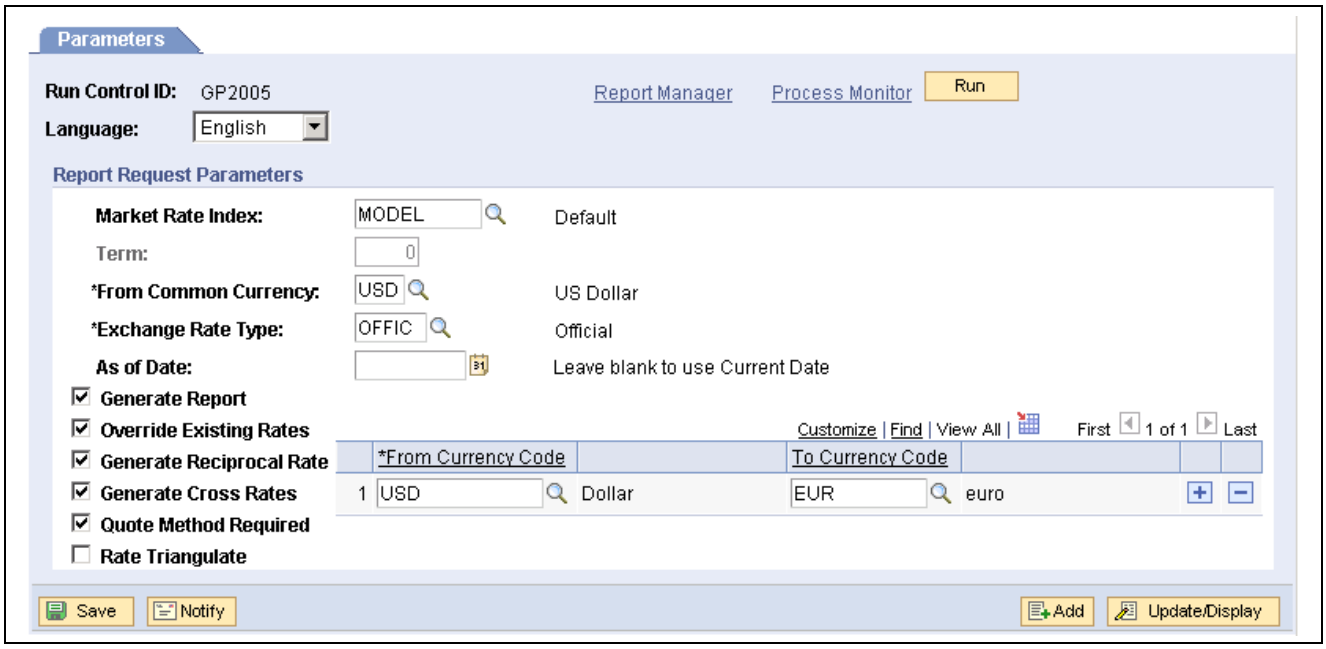
  For example, using a USD to CAD rate as the source, the process calculates the CAD to USD reciprocal. If currency quote methods are in place, the visual rate remains the same and there is a difference in how the data is stored in the database (RATE_MULT and RATE_DIV are inverse). If currency quote methods are not used, the process actually calculates an inverse rate, meaning that the visual rates will differ.

Page Used to Calculate Cross, Triangulated, and Reciprocal Rates

Page Name	Object Name	Navigation	Usage
Cross/Reciprocal Rate Calc - Parameters (cross/reciprocal rate calculation – parameters)	EO_RATECALC	Depends on application.	Set run control parameters to run the EOP_RATECALC Application Engine process, which sets up cross, triangulated, and reciprocal rates.

Running the EOP_RATECALC Process

Access the Cross/Reciprocal Rate Calc - Parameters page.



Cross/Reciprocal Rate Calc - Parameters page

Market Rate Index Select a market rate index. Applications other than PeopleSoft Treasury should use the default index that you select for the exchange rate.

Term This value defaults from the value entered on the Market Rate Definition page.

From Common Currency Select a currency code to calculate a reciprocal rate.

Exchange Rate Type Select the exchange rate type to use for this calculation

As of Date Select the effective date of the newly created exchange rates, which are the output of the process. The as of date also determines the rates used as the basis for the calculations, which are the input of 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. For example, a USD to EPC (euro participating currency) triangulated rate effective April 1, 2004 might be comprised of the EUR to USD rate also effective April 1, 2004 and the fixed EUR to an EPC rate effective on the date the newly participating EPC officially becomes a euro participating currency.

Generate Report Select to generate a report that displays the cross, triangulated, and reciprocal rate calculations performed by the process.

Override Existing Rates Select to have the calculated rates override rates for the exchange rate type, regardless of the as of date.

Generate Reciprocal Rate Select to calculates reciprocal rates for currency pairs that do not have the Auto Reciprocate option select on the Currency Quotation Method page.

You can select this option alone, or in combination with the Generate Cross Rates and Rate Triangulate options.

This process does not directly manipulate the exchange rates. The system uses numerator and denominator values instead, such that the following is true:

(from-currency / RATE_DIV) × RATE_MULT = to-currency

Fro example, suppose you want a reciprocal rate between USD and CHF and assume a two-to-one ratio. If the exchange rate for USD to CHF is quoted directly (either using a direct quote method that you selected or using the system default), this rate is stored as RATE_MULT = 2 and RATE_DIV = 1. The rate is represented as 1 USD = 2 CHF, with a visual rate of 2.

In turn, the CHF to USD rate must be indirect. The reciprocal is a simple exchange, storing the rate as RATE_MULT = 1 and RATE_DIV = 2. The visual rate remains 2.

If quote methods are not being used, the CHF to USD rate must be quoted directly (the default), so the reciprocal rate is actually a calculated inverse. This rate is stored as RATE_MULT = 0.5 and RATE_DIV = 1, with a visual rate of 0.5.

In this example between USD and CHF, using a quote method and using a calculated inverse produced the same end result, 1/2 equals 0.5. But in actual practice, the manipulation of exchange rates is a major task and is one of the reasons for establishing the currency quote method.

Generate Cross Rates Select to automatically generate cross rates. For example, to generate cross currency rates for USD, CAD, and MXP, you enter USD to CAD = 1.473 and USD to MXP = 9.8793. The system automatically generates CAD to MXP = 9.8793/1.473 = 6.7069246.

If you choose to generate cross rates, the From Cur (from-currency) and To Cur (to-currency) fields display and you must select a from-currency and a to-currency. You can enter a wild card of *%* in either or both fields to indicate from all or to all currencies.

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

Select to convert two currencies through a third currency. If you select Rate Triangulate, the From Cur (from-currency) and To Cur (to-currency) fields display and you must select a from-currency and a to-currency. You can enter a wild card of *%* in either or both fields to indicate from all or to all currencies.

Quote Method Required Select to indicate that you want the process to perform selected calculations only if the currency pairs have an existing currency quotation method definition.

Using the Currency Exchange Calculator

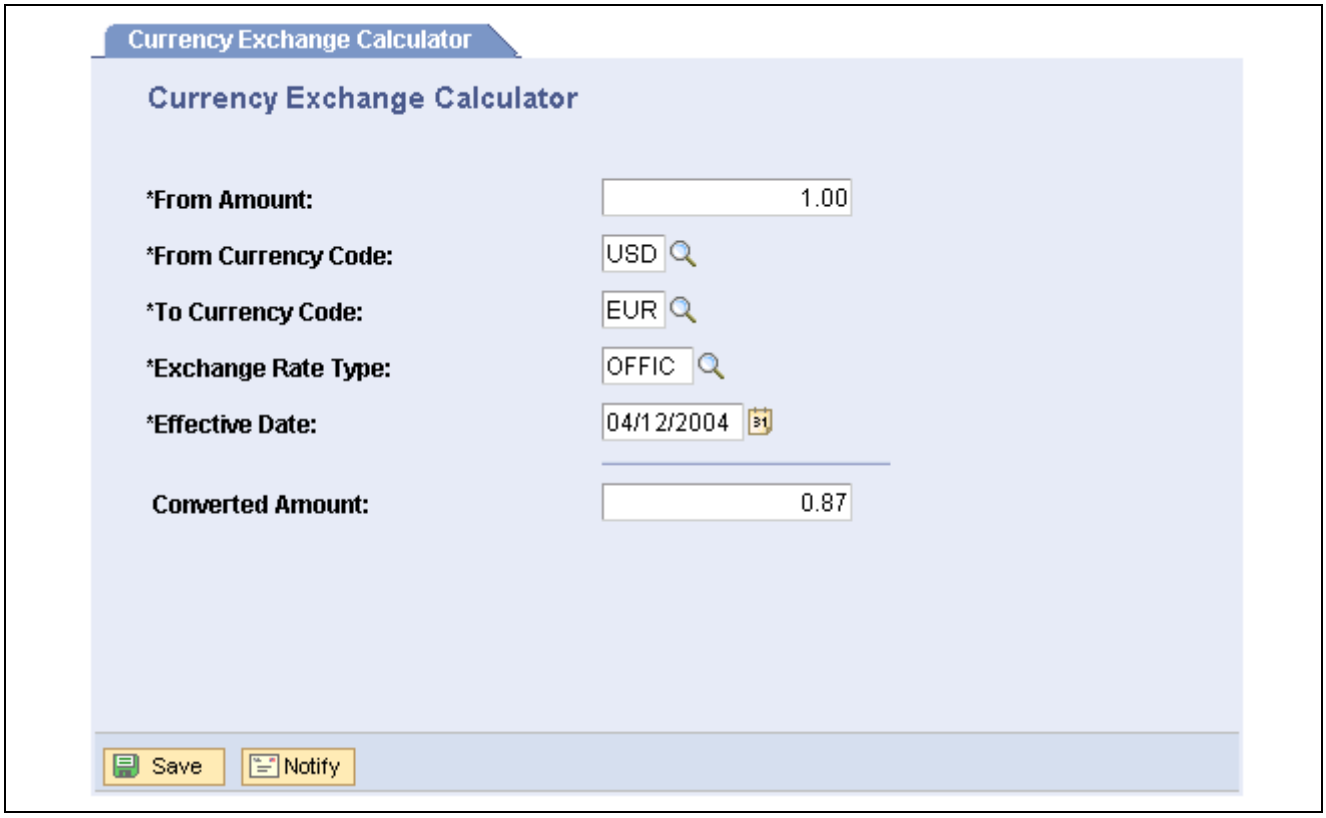
In this section, we discuss how to convert amounts using the Currency Exchange Calculator.

Page Used to Use the Currency Exchange Calculator

Page Name	Object Name	Navigation	Usage
Currency Exchange Calculator	CURRENCY_EXCHNG_PN	Depends on application.	Calculate currency exchange between currencies. This tool enables you to select a rate type other than the base currency, but does not enable you to override the exchange rate.

Converting Amounts Using the Currency Exchange Calculator

Access the Currency Exchange Calculator page.



Currency Exchange Calculator page

From Amount The currency exchange is based on the from amount that you enter and the current exchange rate set up on the Market Rates page.

See Chapter 5, "Working With Currencies and Market Rates," Defining Market Rates, page 61.

From Currency Code Select the currency code from which to calculate the exchange amount.

To Currency Code Select the currency code to which to calculate the exchange amount.

Exchange Rate Type Select the type of exchange rate to use for this calculation.

Converted Amount Click Save to calculate the amount and display it in this field.

CHAPTER 6

Using Datasets

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

- Define dataset rules.
- Define dataset 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 a user's dataset roles. The result of dataset rules is a set of data appropriate to the user's dataset 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 dataset 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 dataset role's security rules. Defining dataset rules creates Structured Query Language (SQL) statements that select the dataset displayed for each rule.

To use dataset rules and roles:

1. Define dataset rules, which are based on a synchronized record.

   You define a dataset rule to return a subset of rows from the selected synchronized record based on the dataset role to which you will link the rule.

   These dataset rules are based on views that can join to any record in your PeopleSoft system.

   For each rule condition, the user specifies a field that comes from the search record name defined in the dataset rule. When the specified field has neither a prompt or translate table edit, the following system variables, which are delivered as system data to all applications, can be used to filter the condition:

   - %Blank

- %Date
- %EmployeeID
- %PersonID
- %Time
- %UserID

2. Assign the dataset rules to dataset roles according to role security and data requirements.

   Each dataset role can have multiple dataset rules. You can use existing dataset roles, or create new dataset roles by selecting from existing PeopleTools security-based user roles.

   See *Enterprise PeopleTools 8.45 PeopleBook: Security Administration,* "Setting Up Roles"

3. Ensure that the original user roles on which dataset roles were based are associated with appropriate user IDs.

   Each user ID can have multiple user roles.

   See *Enterprise PeopleTools 8.45 PeopleBook: Security Administration,* "Administering User Profiles," Setting 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.

The number of rule conditions in a dataset rule is limited only by your performance requirements. You can set a series of rule conditions that can navigate through as many records as necessary.

Dataset Rules

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

Status Select *Active* or *Inactive.*

Rule Conditions

…((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.

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

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 Dataset Roles

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

Page Used to Define Dataset Roles

Page Name	Object Name	Navigation	Usage
Dataset Roles	EOEC_MP_ROLE	Enterprise Components, Component Configurations, Datasets, Dataset Roles	Define dataset roles that associate existing PeopleTools user roles with dataset rules.

Defining Dataset Roles

Access the Dataset Roles page.

Select an existing dataset role for editing, or create a new dataset role by selecting from existing PeopleTools security-based user roles.

See *Enterprise PeopleTools 8.45 PeopleBook: Security Administration,* "Setting Up Roles"

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 rules.
- 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 dataset rules assigned to the mobile device user's dataset roles.

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

Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Mobile Agent

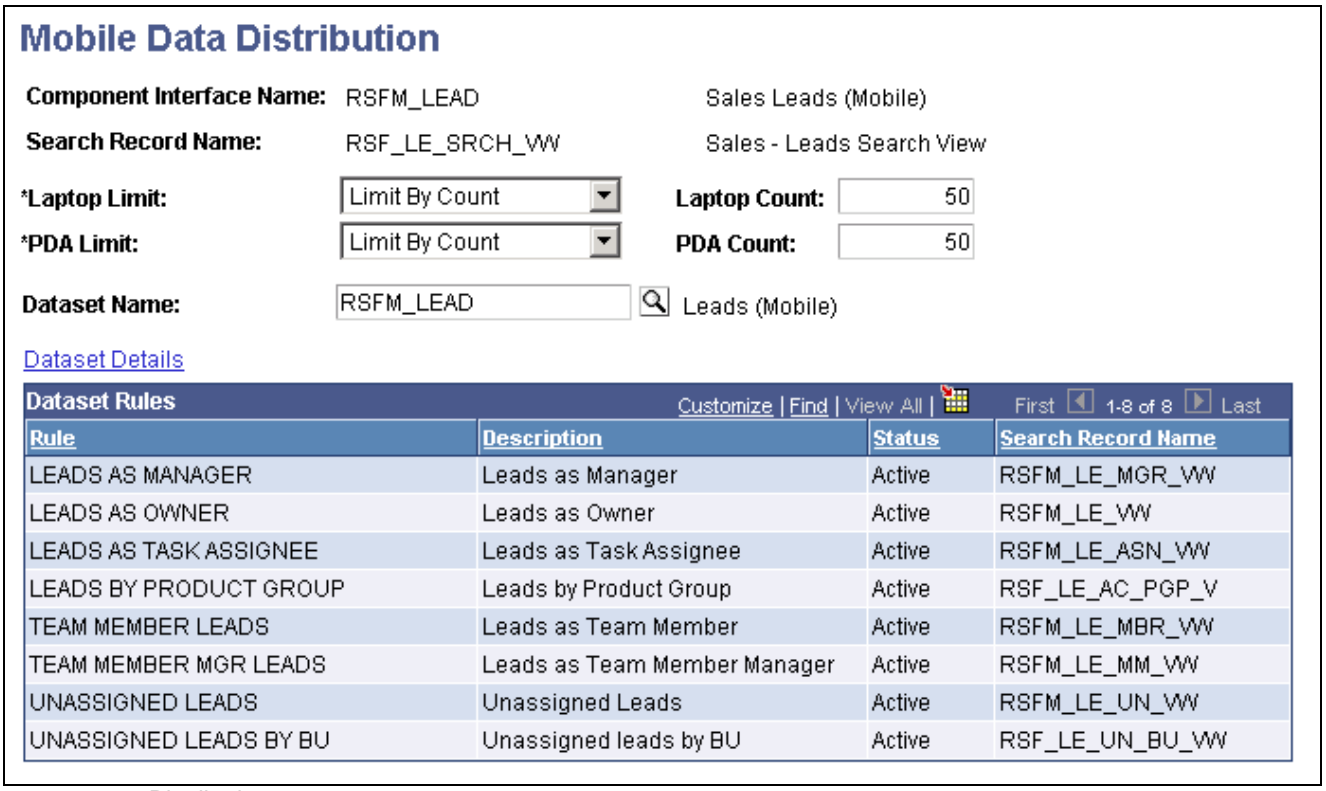
Enterprise PeopleTools 8.45 PeopleBook: PeopleCode Language Reference

Pages Used to Define Mobile Data Distribution

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

Defining Mobile Data Distribution Rules

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: Select to download all data matching the rule's conditions.

Limit By Count: Select to download only data matching the rule's conditions up to the specified count. The count is based on the number of rows returned. Referenced data is also included.

Referenced Data Only: Select to download only data only if it is referenced by another component. You would be more likely to select this option for setup data than for transaction data.

For example, if you have 50,000 products, but the leads you download are related to only 2,000 of the products, select *Referenced Data Only* so that you only download the 2,000 products, instead of the entire set.

If *Reference Data Only* is selected for products, downloads do not include products unless other objects such as leads and opportunities that reference products are 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 mobile data distribution rule.

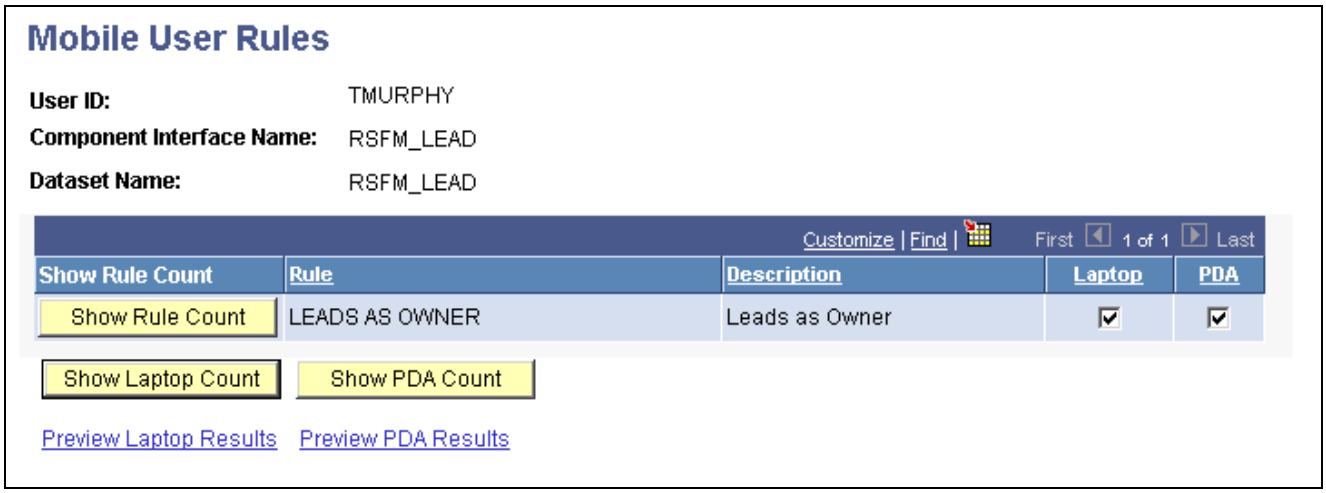
Dataset Details — Click to access the Dataset Rules page, where you can view and modify the selected dataset definition.

See Chapter 6, "Using Datasets," Creating Dataset Rules, page 70.

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

Show Rule Count — Click to view the number of results the mobile data distribution rule returns.

Show Laptop Count and **Show PDA Count** — Click to view the number of results the mobile data distribution rule returns to a laptop computer or PDA based on any limits set for the mobile device on the Mobile Data Distribution page.

See Chapter 6, "Using Datasets," Defining Mobile Data Distribution, page 71.

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

Note. If neither Laptop nor PDA is selected, no data from this mobile data distribution 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 mobile data distribution rule and any limits set for the mobile device on the Mobile Data Distribution page.

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.

See Also

Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Business Interlinks, "Business Interlinks for Runtime Plug-in Programming"

Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Business Interlinks, "Business Interlinks for Application Developers"

Understanding Credit Card Processing

Use the credit card processing interface to integrate with credit card processing vendors. You have the following options for credit card processing:

- CyberSource integration.
- XML-based integration.
- Manual processing.

Understanding a CyberSource Integration

If you use CyberSource, agents processing credit cards can use an application-specific credit card transaction page to submit the transaction to CyberSource for authorization, billing, authorization and billing, or credit. You can choose which types of transactions to permit.

Integration with other vendors may require customizations. There are third-party applications, such as TouchNet, that conform to CyberSource's API. TouchNet follows CyberSource's API standard.

The Credit Card Authorize Bill and Credit integration point enables interaction between PeopleSoft applications and CyberSource by way of the EOEC_CCI_TRANSACTION business interlink. Activate the business interlink by entering a CyberSource merchant ID on the Credit Card Interface Installation page.

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

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

See Also

http://www.cybersource.com

Understanding an XML-Based Integration

PeopleSoft delivers the following XML messages for use with XML-based credit card processing vendors. Ensure that the XML message is transformed into the format that the vendor is expecting.

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 authorize, bill, authorize and bill, or credit transaction.
EOEC_CCI_RESPONSE	The response to the request the credit card interface receives from the third-party vendor.

Agents can then process credit cards using their application-specific credit card transaction page to submit the transaction to the vendor for authorization, billing, authorization and billing, or credit. You can choose which types of transactions to permit.

If you use an XML-based interface, you must use PeopleSoft Integration Broker to set up the node to which the XML messages should be sent and 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.

See Also

Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Integration Broker

Understanding Manual Processing

If you decide not to use CyberSource or an XML-based credit card processing system, then your application-specific credit card transaction 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:

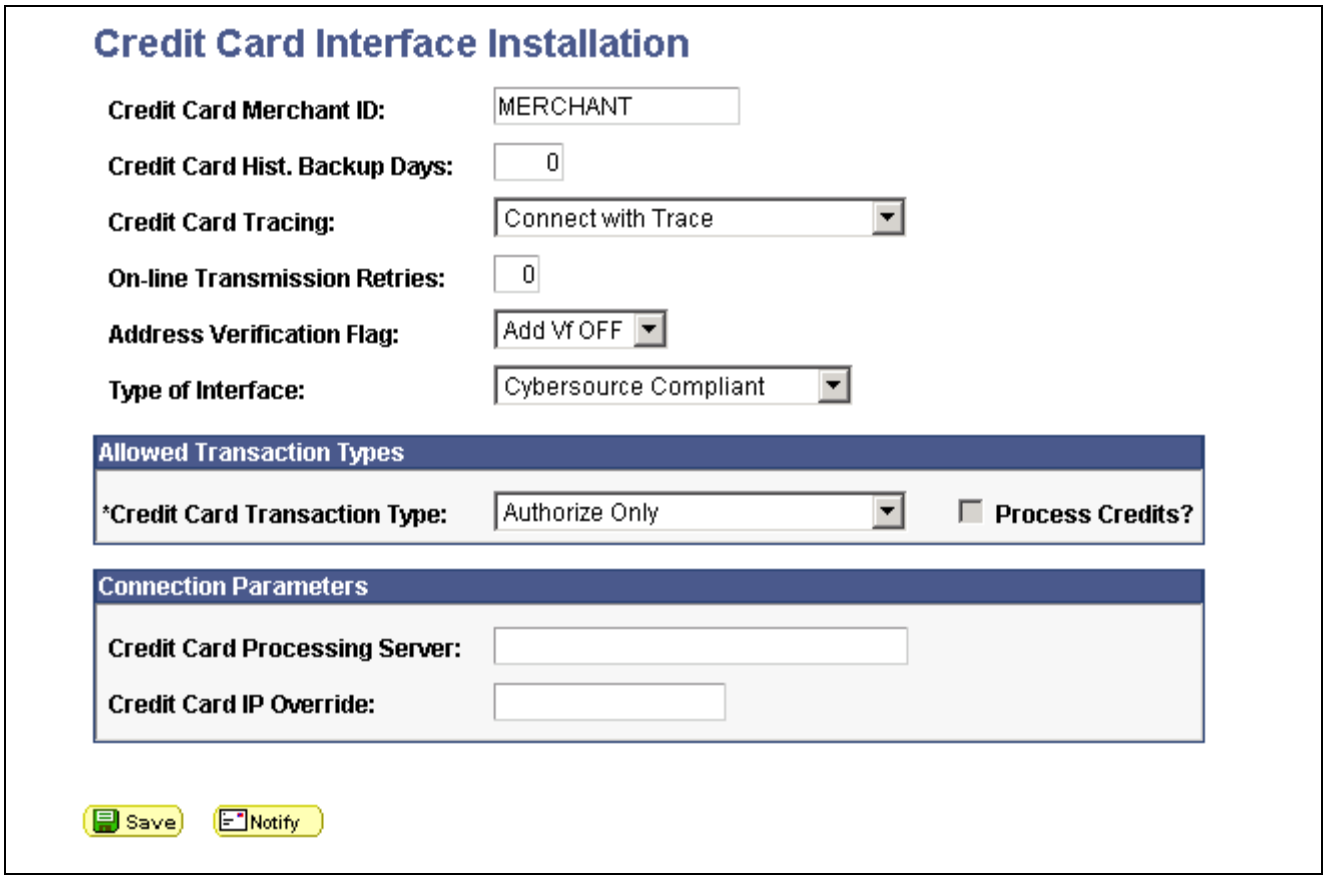
- Define connection parameters.
- Define accepted credit card types.
- Test the credit card interface.
- Test the credit card transaction.

Pages Used in Setting Up the Credit Card Interface

Page Name	Object Name	Navigation	Usage
Credit Card Interface Installation	EOEC_CCI_INSTAL	Enterprise Components, Component Configurations, Credit Card Interface, Setup	Set up connection parameters for credit card processing calls to CyberSource, or a CyberSource-compliant or XML-based third-party credit card processing vendor. Before you set up credit card processing options, establish your merchant account with CyberSource, or a CyberSource-compliant or XML-based vendor.
Credit Card Setup	EOEC_CCI_CARDTYPE	Enterprise Components, Component Configurations, Credit Card Interface, Credit Card Types	Define the types of credit cards you accept for credit card processing.
Test Credit Card Interface - Card Entry/Display	EOEC_CCI_TEST	Enterprise Components, Component Configurations, Credit Card Interface, Test Credit Card Interface, Card Entry/Display	Enter test credit card information, which you can then submit to verify that your credit card processing is functioning properly.
Test Credit Card Interface - Transaction	EOEC_CCI_TRANSACT	Enterprise Components, Component Configurations, Credit Card Interface, Test Credit Card Interface, Transaction	Enter test credit card transaction 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

Entering information on this page is required for CyberSource, CyberSource-compliant, and XML-based processing systems. Verify connection requirements with your vendor.

Credit Card Merchant ID Enter the merchant ID supplied by your vendor.

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.

Credit Card Tracing Select whether to keep a trace file of credit card transactions. Tracing functionality associated with this field applies to tracing for CyberSource and CyberSource-compliant vendors. Select from the following trace file options:

Connect with Trace: Connect to your vendor and create a trace file.

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

Production: Send data directly to your 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: This option is available only for CyberSource. Invoke a troubleshooting utility provided by CyberSource to aid in diagnosing connection problems. Results are displayed in the trace file.

See CyberSource documentation.

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

The way in which the trace file is created depends on the setting of the EOEC_CCI_TRANSACTION Business Interlink's trace_file parameter.

If no path and file name are specified in the trace_file parameter, which is the default, a crcard.txt trace file is created in one of the following directories:

- For Windows: c:\temp
- For UNIX: $PS_HOME/appserv/<db_name>

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

If only a path is specified for the trace_file parameter, the crcard.txt trace file is created or appended to as defined in the parameter.

If only a file name is specified for the trace_file parameter, the trace file is created or appended to as defined in the parameter in one of the following directories:

- For Windows: c:\temp
- For UNIX: $PS_HOME/appserv/<db_name>

If the path and file name are specified for the trace_file parameter, the trace file is created or appended to as defined in the parameter.

See Chapter 7, "Setting Up the Credit Card Interface," Setting Up Tracing, page 94.

Tracing for an XML-based interface is controlled through PeopleSoft Integration Broker as part of the setup for synchronous node transactions.

See *Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Integration Broker,* "Configuring Nodes and Transactions," Editing Transaction Message Details

On-line Transmission Retries

Enter a value between *0* and *9* to specify how many times the system should attempt to retransmit transactions 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.

See Chapter 7, "Setting Up the Credit Card Interface," Using the Address Verification Service, page 100.

Type of Interface

Cybersource Compliant: Select for CyberSource or a CyberSource-compliant vendor. A CyberSource-compliant vendor is one that complies with CyberSource's APIs.

XML Compliant: Select for an XML-based vendor.

Allowed Transaction Types

Credit Card Transaction Type Select the types of transactions your agents are allowed to submit. Disallowed transaction types are not available on the application-specific credit card transaction page. Select from:

Authorize Only: Your vendor verifies that the card is valid for the charge. For example, the customer has enough credit to pay for the order, the card is not stolen, and so forth. The vendor does not bill the credit card.

Bill Only: Your vendor bills the card without first verifying that the card is valid for the charge. Select this option if you have pre-authorized the transaction and you want to submit the transaction for billing only.

Authorize and Bill: Your vendor performs both authorization and billing during the same transaction. The vendor charges the customer's credit card on receiving authorization.

Credit Only: Your vendor credits the customer's credit card.

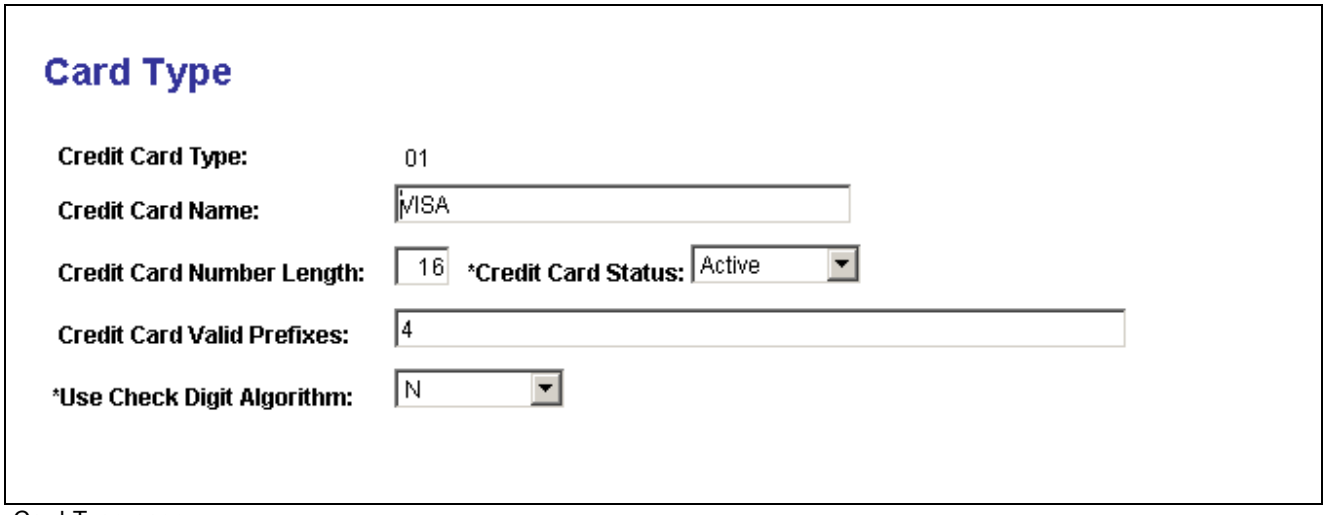
Process Credits Select to permit agents to submit credit transactions as well as billing transactions. Available only if you selected either *Authorize and Bill* or *Bill Only* in the Credit Card Transaction Types field.

Connection Parameters

CyberSource, and CyberSource-compliant and XML-based third-party vendors provide you with information to connect with their systems. Enter that information to enable PeopleSoft to make the connection when you submit a transaction for processing.

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 existing definitions and add new ones.

Credit Card Type The following credit card types are delivered:

01: Visa.

02: MasterCard.

03: Diners Club/Carte Blanche.

04: American Express.

05: 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 your 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 application-specific credit card transaction page nor in the Test Credit Card Interface component.

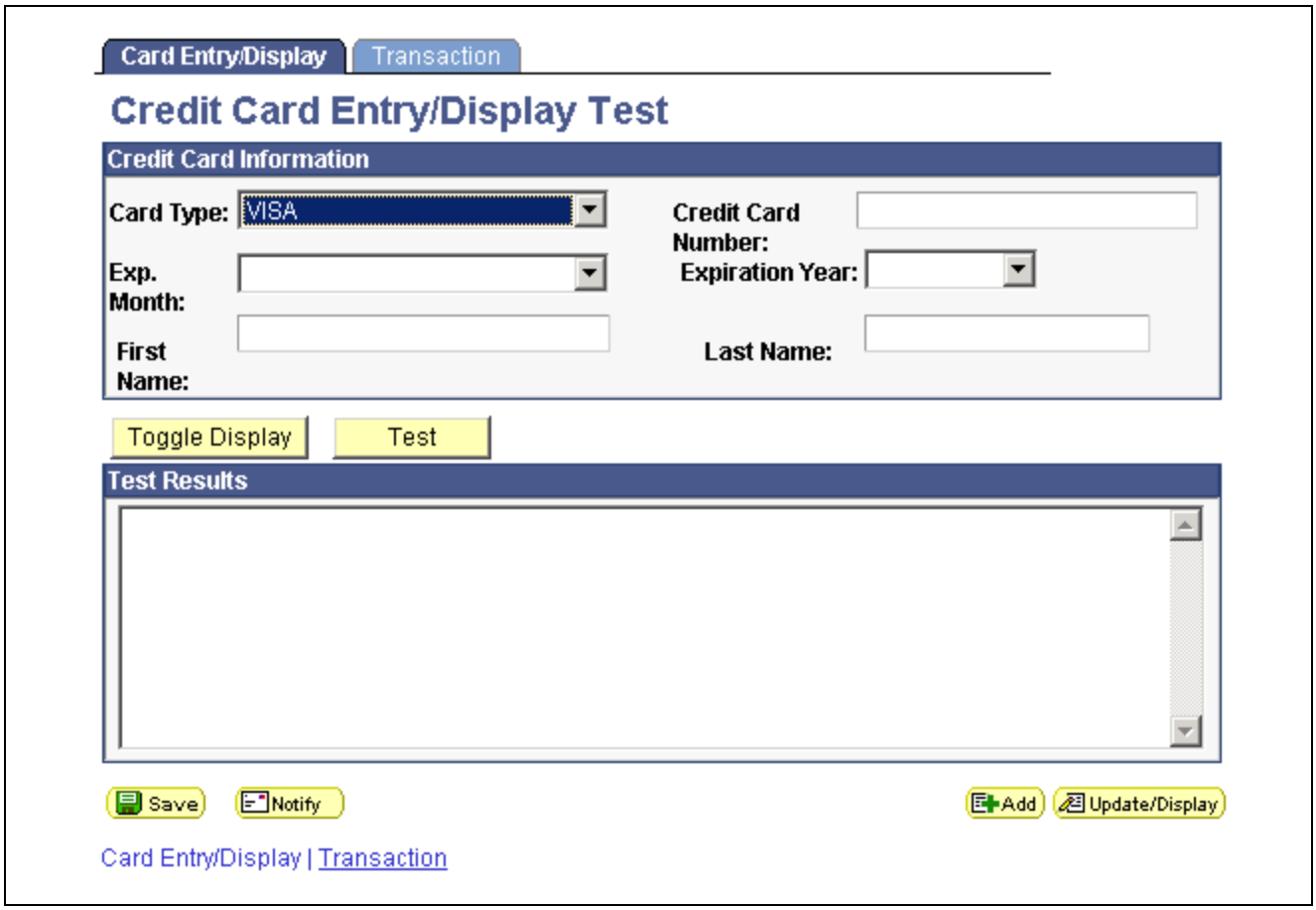
Credit Card Valid Prefixes Enter all valid prefixes for this type of credit card. Enter multiple prefixes in comma-separated format with no spaces in between. The system removes any characters other than numbers and commas when you move to another field.

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 modulus (MOD) 10 check digit algorithm to validate credit card numbers before transmitting requests to your vendor. The MOD 10 check digit algorithm verifies whether card numbers you enter into the system are legitimate.

Testing the Credit Card Interface

Access the Test Credit Card Interface - Card Entry/Display page.



Test Credit Card Interface - Card Entry/Display page

The test you can run on this page does the following:

- Verifies that the card number you enter meets the requirements defined in the Credit Card Valid Prefixes field for the associated card type on the Card Type page.
- If you have set the Use Check Digit Algorithm field value to *Y*, verifies that the card number is valid based on the MOD 10 check digit algorithm.
- Verifies that you have entered values in the Exp. Month, Expiration Year, First Name, and Last Name fields on this page.

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

Credit Card Type	Credit Card Number
American Express	378282246310005
Diners Club/Carte Blanche	38000000000006
Discover	6011111111111117

Credit Card Type	Credit Card Number
MasterCard	5555555555554444
Visa	4111111111111111

Card Type Select a card type to test. Available values are defined on the Credit Card Types page.

See Chapter 7, "Setting Up the Credit Card Interface," Defining Accepted Credit Card Types, page 80.

Credit Card Number Enter the credit card number to test.

Exp. Month (expiration month), **Expiration Year, First Name** , and **Last Name** Enter card information to test.

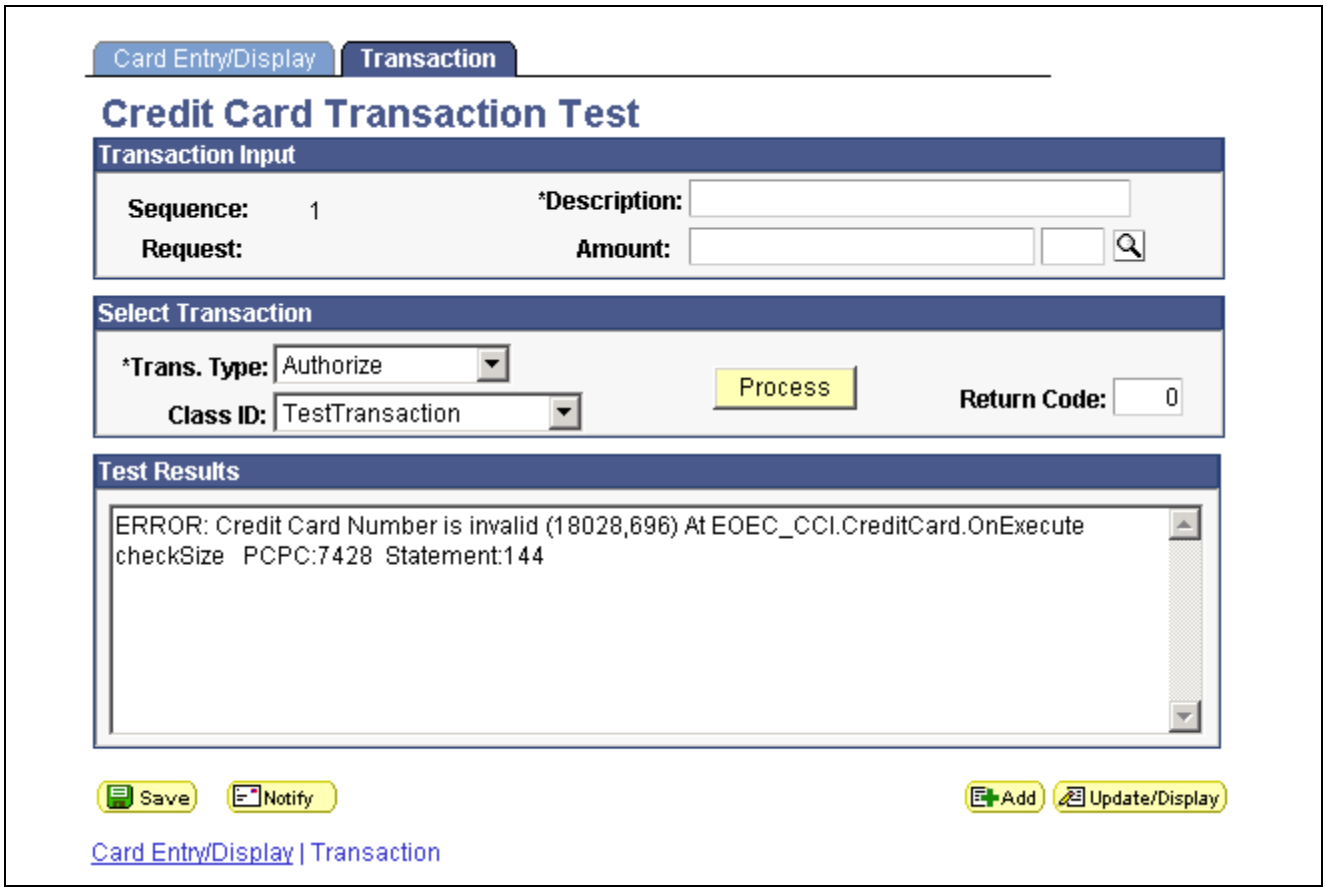
Toggle Display Click to switch between display-only and editable modes.

Test Click to begin the test.

Test Results The results of the test appear in this text box. If the card number is valid, the message VALID CARD NUMBER appears. If the card number is not valid, an explanatory message appears; the card number is incorrect or the card is expired, for example.

Testing Credit Card Transactions

Access the Credit Card Transaction Test page.



Credit Card Transaction Test page

Sequence and **Request**

Display a combination of numbers that distinguish the transaction from other transactions. They are similar to a run control or job number.

Amount

Enter a transaction amount and click the Look up button to select a transaction 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 *Test Transaction* (the default) or one of the following interface types that has been specified on the Credit Card Interface Installation page.

ProcessBrokerTransaction: Select if you want to test your XML-based interface.

Interlink Transaction: Select if you want to test your CyberSource-compliant interface.

See Chapter 7, "Setting Up the Credit Card Interface," Defining Connection Parameters, page 77.

Process Click to process the test transaction.

Return Code Enter a return code to test whether proper error messages and results are returned. Available codes and their descriptions are discussed in the Return Codes section below.

Test Results The results of the transaction test appear in this text box. Test result interpretations are discussed in the following sections.

Return Codes

You can enter any of the following return codes and click Process to view the corresponding description and error message in the Test Results area. These return codes and their corresponding errors can appear in multiple areas. For example, when using the Test Credit Card Interface component, they appear on this test page. When using the Business Interlink Tester, they are displayed in the various return fields. In an application, they appear as appropriate to that application's method of interacting with the credit card interface.

Return Code	Description
-3	Error Opening Trace File
-4	Vendor Error − ICS_INIT failed
-5	Unsupported Service
-6	Credit card number is invalid.
-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

Return Code	Description
-18	Email ID is required
-19	Zip Code is required
-20	City is required
-21	Country code is required
-23	Address 1 is required
-99	Trace Run Only

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 Solaris systems.

Understanding the Java Interlink Plug-In

Using PeopleSoft Business Interlink technology, the Java interface supports four types 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 from the Support Center in the CyberSource website. Download the zipped API files and use the instructions in this section to configure the environment. You must obtain the Java version of the API even if you are already using the C++ plug-in and API.

See http://www.cybersource.com

We support CyberSource API version 3.5.0.0 when used with the following:

- Java Development Kit (JDK) version 1.1.8, 1.2.2, or later; installed and running.
- Solaris 2.6 operating system on SPARC or Windows NT Server 4.0 on Intel.

- If you use a Java Virtual Machine that is not developed and supported by Sun Microsystems, you may encounter problems with the Internet Commerce Suite CyberSource Development Kit for Java.

Note. When you upgrade to PeopleTools 8.46 or higher, you must upgrade to Cybersource 3.7.11.

Note. Other UNIX platforms are not known to be supported, but please refer to the CyberSource website for a current listing of supported platforms.

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 on Windows NT.
- Set up the CyberSource API on Windows NT.
- Set up the plug-in on Windows NT.

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_HOME*>\class.

   You must include the name of the .jar file in the CLASSPATH entry; for example, <PS_HOME>\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*\cdkjava118-3500\properties\ICSClient.props to point to the key and certificate files located in the keys directory.

   Use a forward slash—not a backslash—to separate the directories, as follows:

```
# file with the location of your private key required
myPrivateKey=<CyberSource Home>/keys/<filename.pvt>
# file with the location of your cert required
myCert=<CyberSource Home>/keys/<filename.crt>
# file with the location of the CyberSource cert
serverCert=<CyberSource Home>/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_HOME*>\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_HOME*>*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 Solaris Systems

This section discusses how to:

- Set up the business interlink environment on Solaris.
- Set up the CyberSource API on Solaris.
- Set up the plug-in on Solaris.
- Configure the application server on Solaris.

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_HOME/appserv/classes.

   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_HOME/appserv/classes. You must include the name of the .jar file in the PS_CLASSPATH entry; for example, $PS_HOME/appserv/classes/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 backslash—to separate the directories, as follows:

```
# file with the location of your private key required
myPrivateKey=<CyberSource Home>/ics/keys/<filename.pvt>
# file with the location of your cert required
myCert=<CyberSource Home>/ics/keys/<filename.crt>
# file with the location of the CyberSource cert
serverCert=<CyberSource Home>/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_HOME*/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_HOME*/bin/InterfaceDrivers directory.

   Note. 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.

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 Solaris.

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

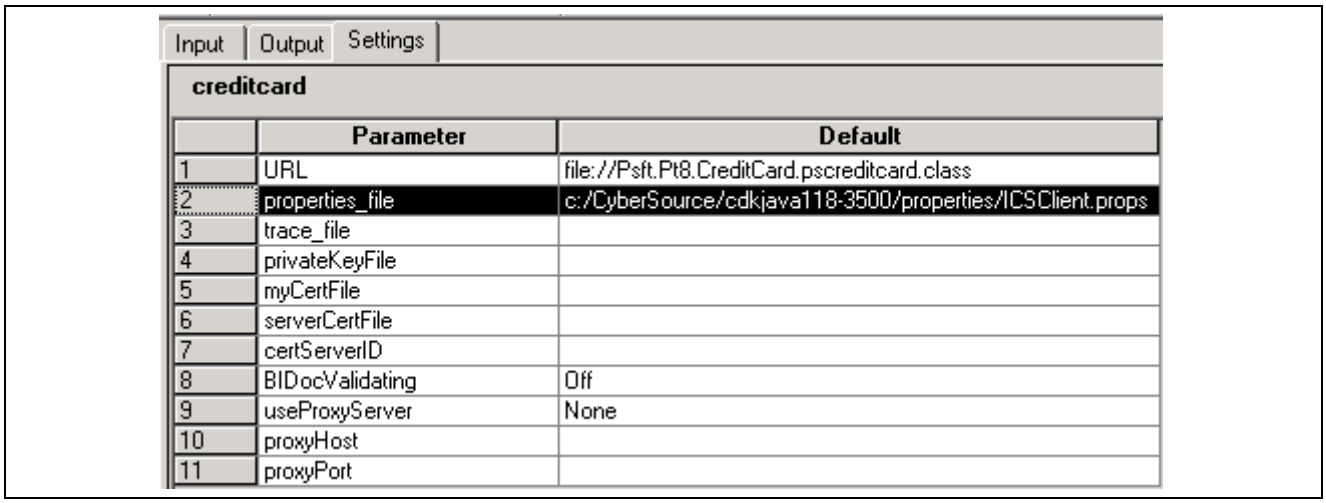
  The plug-in gets the certificate and key files from <CyberSource Home>\keys. Use this option if you have been using the C++ plug-in and want to continue to operate on a Microsoft Windows platform.

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.

Option 1: Keep All Configuration Parameters in a Properties File

Use this option if you are building a new credit card processing application. The following page shows settings for maintaining parameters in a properties file:



Settings page used to keep all configuration parameters in a properties file

To keep all of the configuration parameters in a properties file:

1. 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.

   Use a forward slash to separate directories, as follows:

```
# file with the location of your private key required
myPrivateKey=<CyberSource Home>/keys/<filename.pvt>
# file with the location of your cert required
myCert=<CyberSource Home>/keys/<filename.crt>
# file with the location of the CyberSource cert
serverCert=<CyberSource Home>/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.

Option 2: 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 Solaris.

Input | Output | Settings

creditcard

	Parameter	Default
1	URL	file://Psft.Pt8.CreditCard.pscreditcard.class
2	properties_file	
3	trace_file	
4	privateKeyFile	/cybersource/3500/unix/keys/ICS2Test.pvt
5	myCertFile	/cybersource/3500/unix/keys/ICS2Test.crt
6	serverCertFile	/cybersource/3500/unix/keys/CyberSource_SJC_US.crt
7	certServerID	CyberSource_SJC_US
8	BIDocValidating	Off
9	useProxyServer	None
10	proxyHost	
11	proxyPort	

Settings page used to pass input parameters and specify the path of the key and certificate files

Input | Output | Settings

creditcard

creditcard

	Input Name	Input Path	Default
1	service	service	1
2	merchant_ref	merchant_ref	10001
3	score_criteria_modified	fraud_screen_	N
4	fraud_screen	fraud_screen	Y
5	trace	trace	N
6	decrypt	decrypt	N
7	paramchk	paramchk	N
8	merchant	merchant	ICS2TEST
9	fname	fname	Rob
10	lname	lname	Smith
11	email	email	rs@rs.com
12	phone	phone	925/694-1775
13	addr1	addr1	123 Here St.
14	city	city	Pleasanton
15	state	state	CA
16	zip	zip	94588
17	country	country	USA
18	ccnum	ccnum	4111111111111111
19	expmo	expmo	03
20	expyr	expyr	2002
21	currency	currency	USD
22	amount	amount	100.00
23	avs_ignore	avs_ignore	N
24	server_host	server_host	ics2test.ic3.com:80

Input page - passing 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 configuration 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.

Option 3: Use the Java Plug-in as You Use the C++ Version

Use this option if you have been using the C++ plug-in and want to continue to operate on a Windows platform.

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.

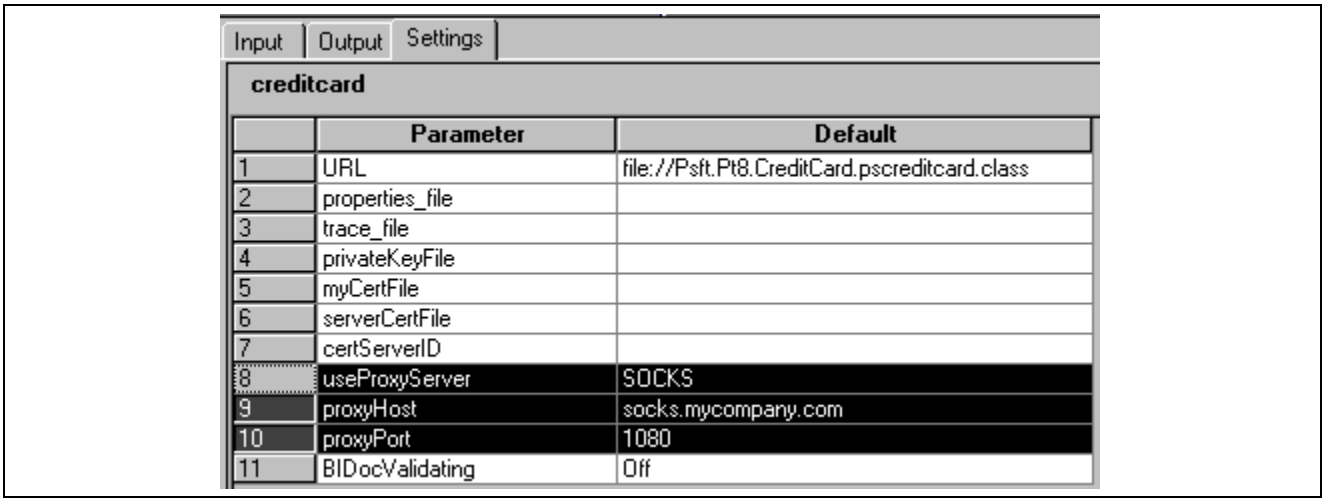
   Do not include the text *http://* in the server URL string.

Configuring Proxy Server Support

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

See CyberSource ICS2 Developer's Guide

The following example 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.

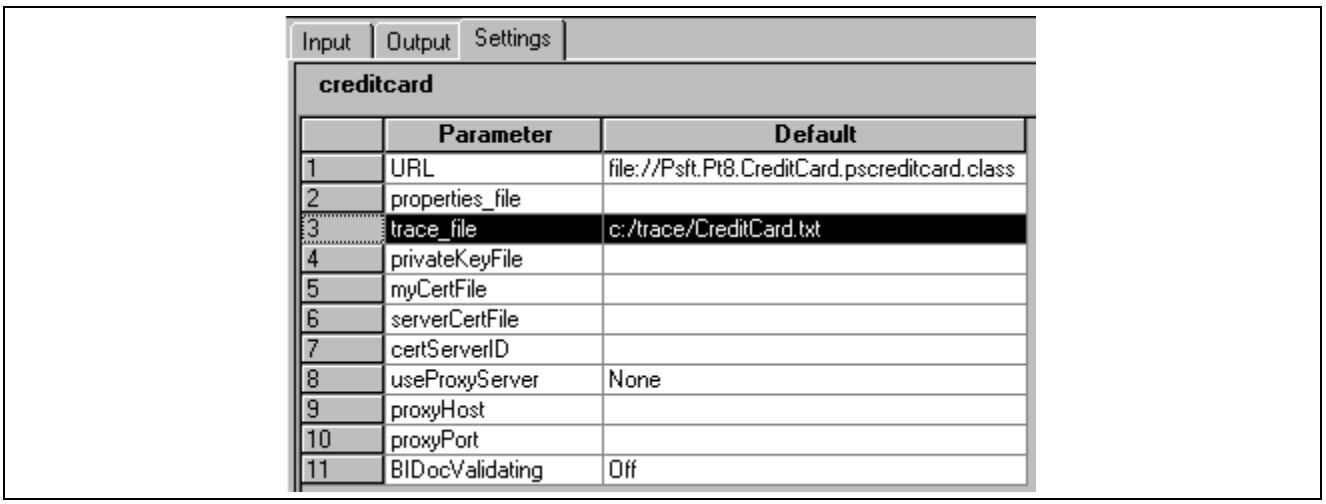
  The trace file writes to c:\temp\crcard.txt.

- 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 Solaris, or for writing the trace file to a directory other than c:\temp. Use forward slashes when specifying directory paths.

  See Chapter 7, "Setting Up the Credit Card Interface," Defining Connection Parameters, page 77.

The following screen shot illustrates the second method.



Settings page used to activate tracing

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 for available transactions and discusses how to:

- Use authorize-and-bill transactions.
- Use authorize-only transactions.
- Use bill-only transactions.
- Use credit transactions.
- Use the address verification service.
- Use the fraud screen service.
- Identify input fields that are not used or supported.

Identifying 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.

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.

Required input data includes:

- Service
- Merchant_ref
- Fname
- Lname
- Addr1
- City
- Country
- State
- Zip
- Email
- Expmo
- Expyr
- Ccnum
- Amount
- Currency

Output data includes:

- return_status
- Return_status_msg
- Ret_msg1
- Ret_msg2
- Ret_authcd
- Ret_authdttm
- Rqstid_out
- Bill_amount

Using Authorize-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.

Required input data includes:

- Service
- Merchant_ref
- Fname
- Lname
- Addr1
- City

- Country
- State
- Zip
- Email
- Expmo
- Expyr
- Ccnum
- Amount
- Currency

Output data includes:

- Return_status
- Return_status_msg
- Ret_msg1
- Ret_msg2
- Ret_authcd
- Ret_authdttm
- Rqstid_out

Using Bill-Only Transactions

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

Required input data includes:

- Service
- Merchant_ref
- Amount
- Currency
- Rqstid

Output data includes:

- Return_status
- Return_status_msg
- Ret_msg1
- Ret_msg2
- Ret_authcd
- Ret_authdttm
- Rqstid_out
- Bill_amount

- Trans_ref_no

Using Credit Transactions

You can 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 input data includes

- Service
- Merchant_ref
- Amount
- Currency
- Rqstid

Output data includes:

- Return_status
- Return_status_msg
- Ret_msg1
- Ret_msg2
- Ret_authcd
- Ret_authdttm
- Rqstid_out
- Credit_amount
- Trans_ref_no

Use Billing Information

Required input data includes:

- Service
- Merchant_ref
- Fname
- Lname
- Addr1
- City
- Country
- State
- Zip

- Email
- Phone
- Expmo
- Expyr
- Ccnum
- Amount
- Currency

Output data includes:

- Return_status
- Return_status_msg
- Ret_msg1
- Ret_msg2
- Ret_authcd
- Ret_authdttm
- Rqstid_out
- Credit_amount

Using the 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.

Value	Description
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.
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.

Using the Fraud Screen Service

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

See CyberSource documentation.

The Fraud_screen field determines if the fraud screen transaction request should be sent with the current transaction.

The Score_criteria_modified field 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 data input includes:

- Service
- Merchant_ref
- Fname
- Lname
- Addr1
- City
- Country
- State
- Zip
- Email
- Phone
- Expmo
- Expyr

- Ccnum
- Amount
- Currency

Output data includes:

- Return_status
- Return_status_msg
- Ret_msg1
- Ret_msg2
- Ret_authcd
- Ret_authdttm
- Rqstid_out
- Bill_amount (authorize/bill transaction)

Identifying Input Fields That Are Not Used or 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

CHAPTER 8

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 multidimensional 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.

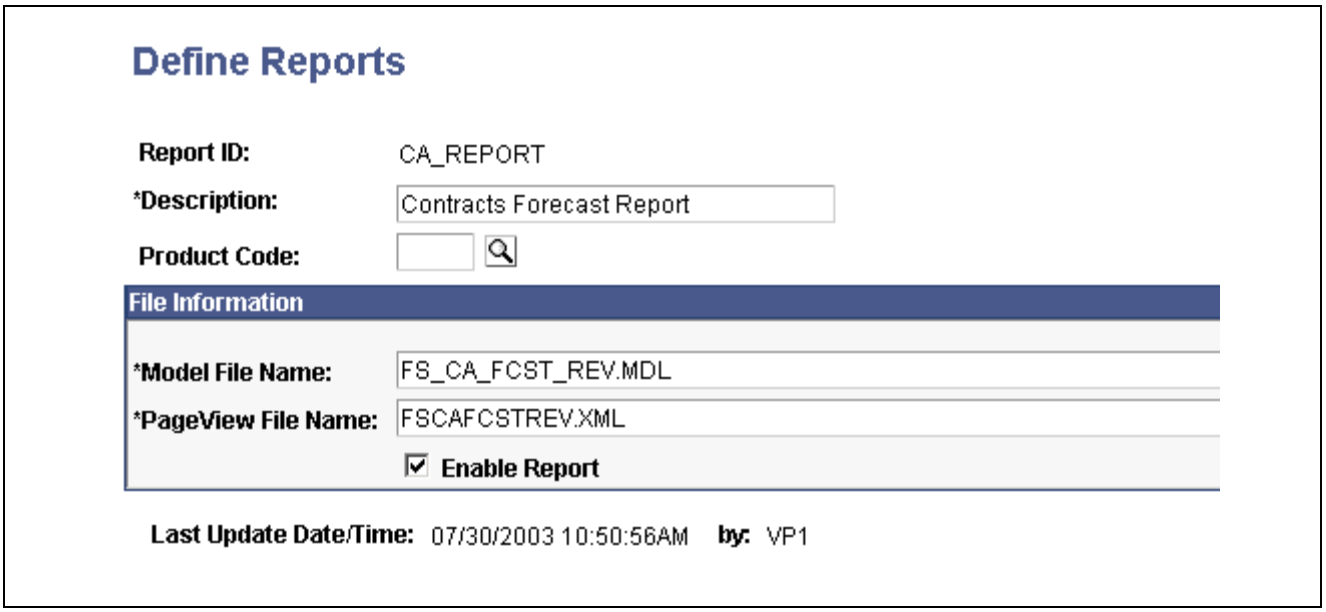
Configuring Interactive Reports

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 Report Definitions, Environment Configuration	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	Click the View Query Prompts link for the Query Name (Base Language) field on the Map Queries page.	Specifies the record and field containing the base language query prompt values when the report is launched.
Related Language Query Prompt	EOBF_QRY_PRMPT_LNG	Click the View Query Prompts link for the Query Name (Related Language) field on the Map Queries page.	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.

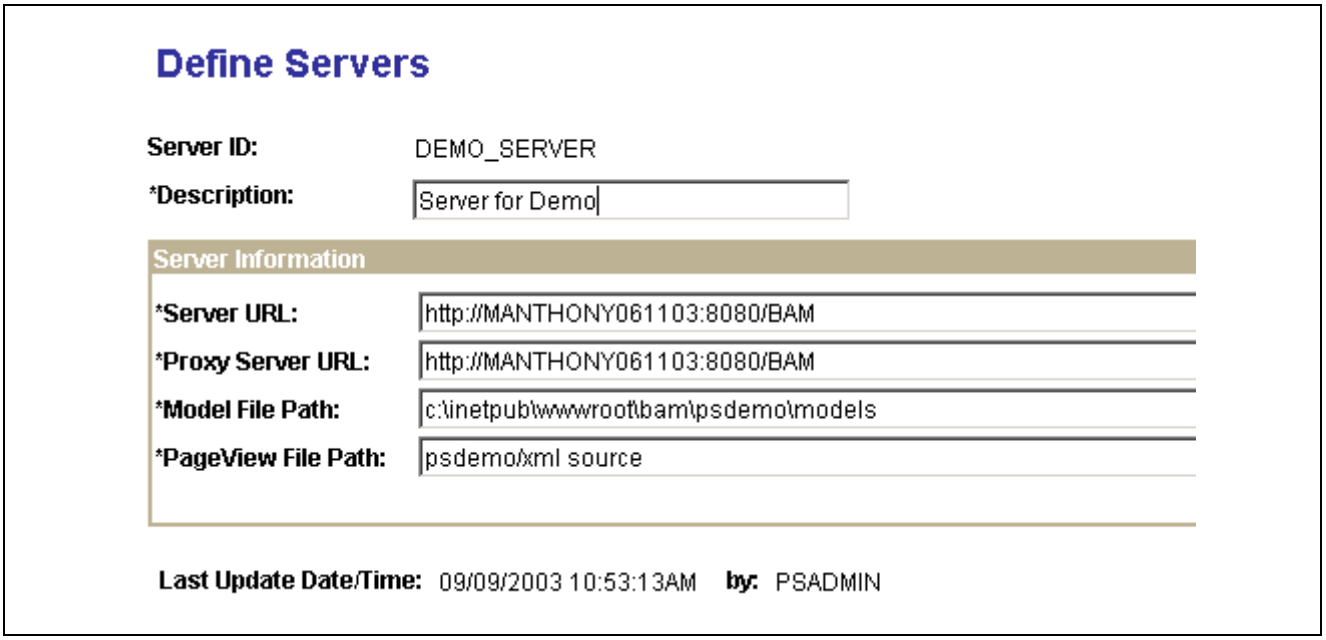


Define Interactive Reports

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.

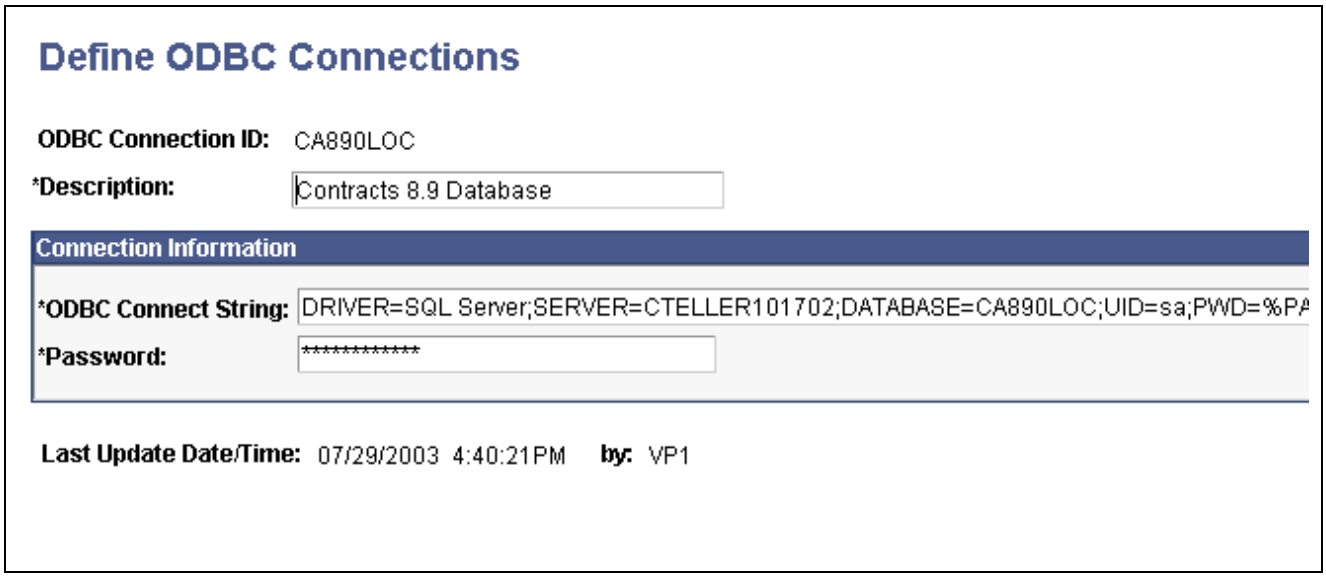
Model File Path Path of the .MDL (model) file shown on the Define Reports page. These files are usually on the PeopleSoft Business Analysis Modeler Analytic Web Server in the c:\inetpub\wwwroot\bam*PeopleSoft product*\models directory.

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.

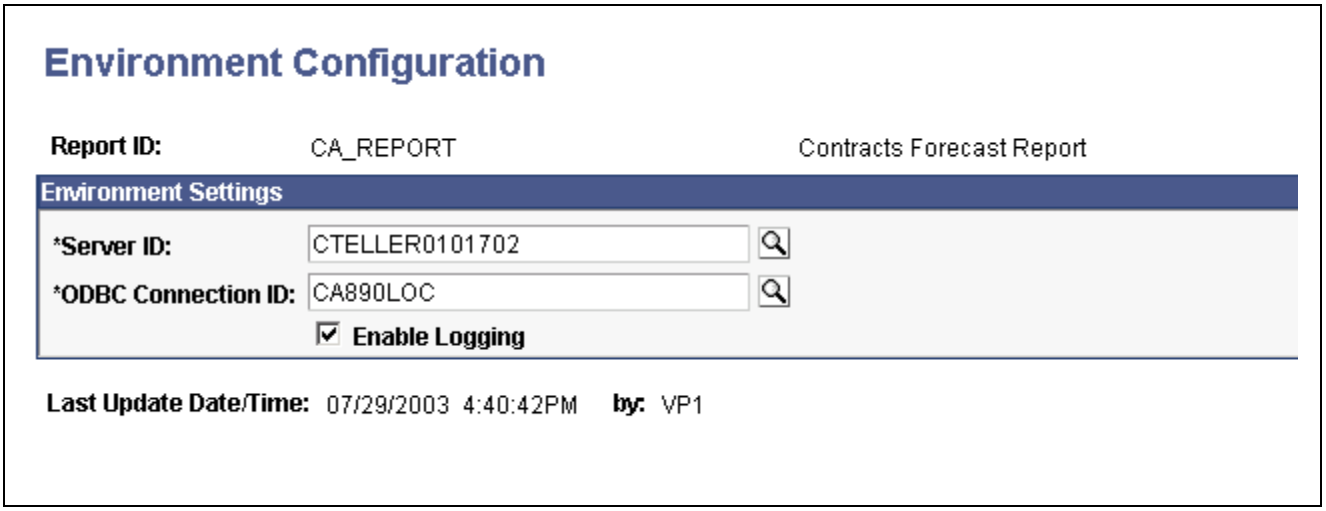
Password and **Confirm Password** Enter a password to connect to the database listed in the ODBC Connect String. Reenter the password in the Confirm Password field. The user ID listed must have a minimum of *Select* privileges on the database.

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

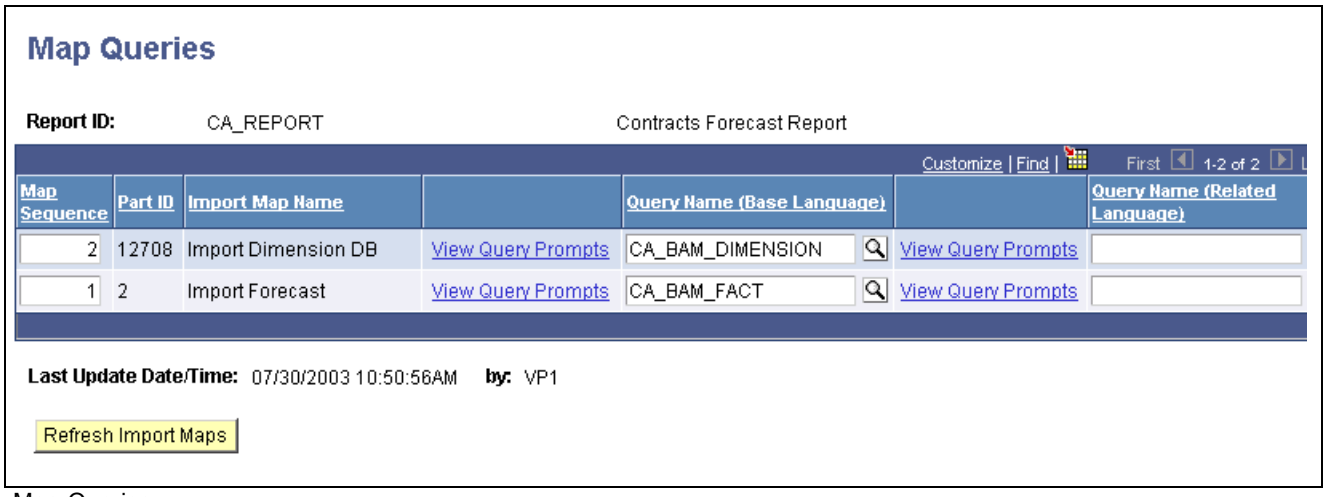
Select the Server ID and ODBC Connection ID to be used with this report.

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 8.8 Business Analysis Modeler Installation Documentation*

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.

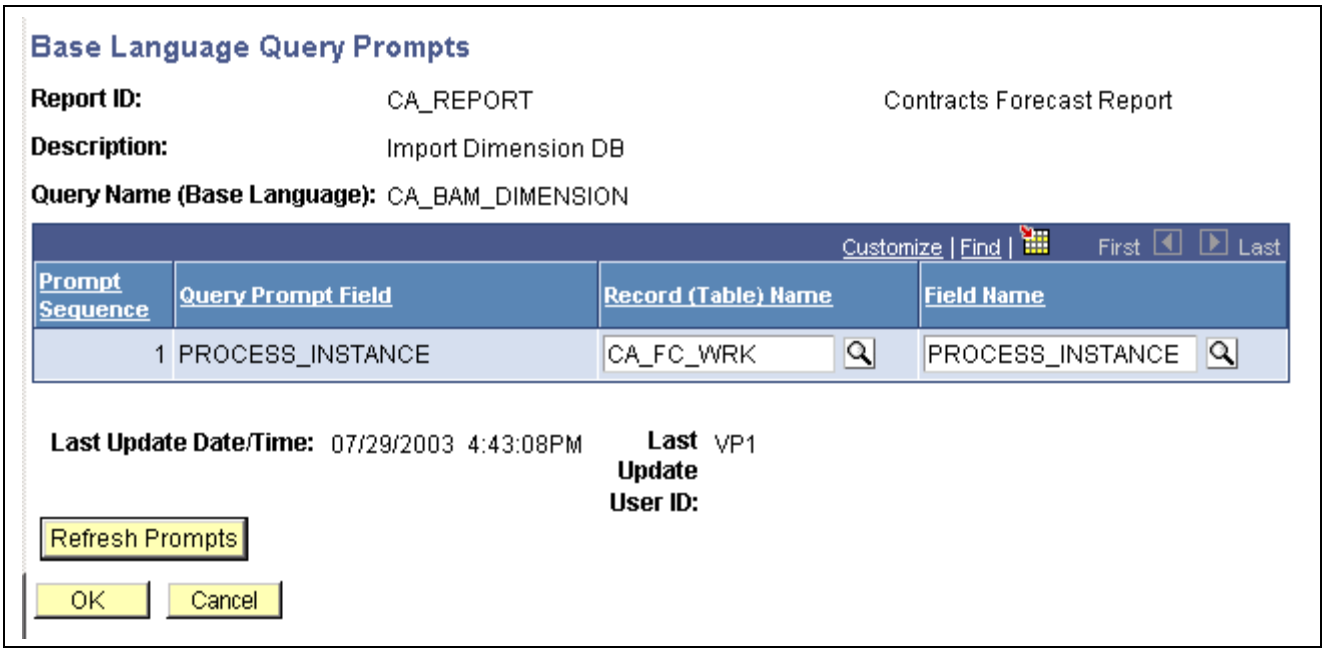
Query Name (Base Language) Identifies the PeopleSoft query used for the selection of data to the Import Map.

Query Name (Related Language) (Optional) If the report is used in a multilingual 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.

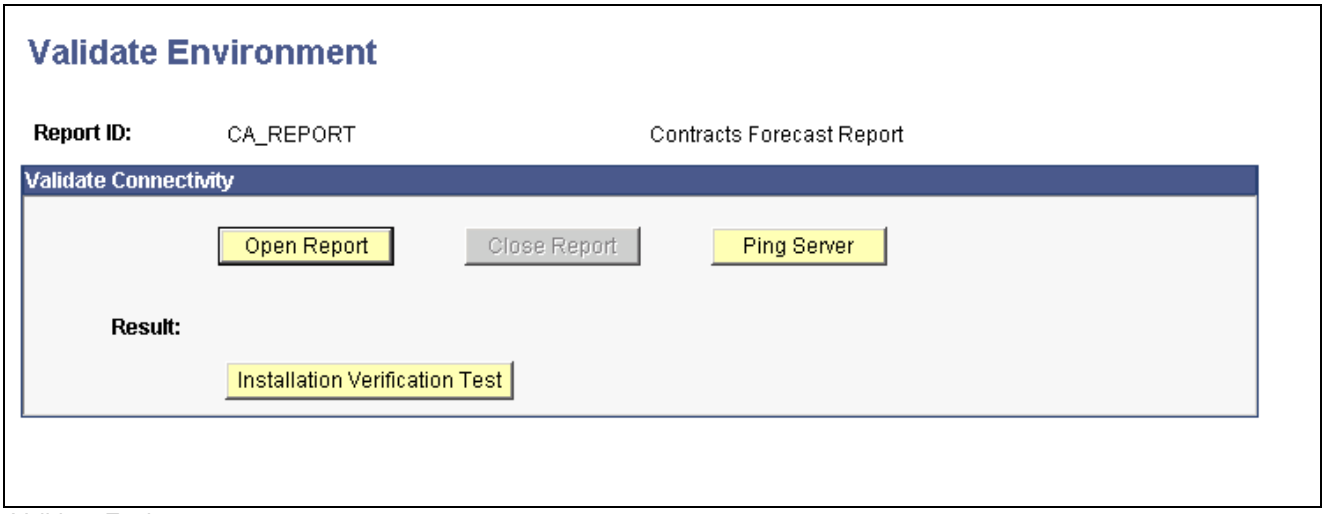


Base Language Query Prompt

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.

PART 4

PeopleSoft Active Analytics Framework

Chapter 9
Understanding PeopleSoft Active Analytics Framework

Chapter 10
Building and Managing Policies

Chapter 11
Setting Up the Data Library

Chapter 12
Managing Contexts

Chapter 13
Setting Up the Action Framework

Chapter 14
Administering the Framework

Chapter 15
Considerations for Enabling the Framework

CHAPTER 9

Understanding PeopleSoft Active Analytics Framework

This chapter provides an overview of PeopleSoft Active Analytics Framework and discusses:

- Policies and trigger points.
- The data library.
- The action framework.

Understanding PeopleSoft Active Analytics Framework

PeopleSoft Active Analytics Framework is a suite of tools comprising a closed-loop decision-making system where business-intelligent applications or transactions can respond when conditions are met and specific actions are recommended; for example:

- Giving a priority service or a better discount for high-value customers
- Send pertinent emails or notifications.
- Displaying alerts and warning messages.

PeopleSoft Active Analytics Framework provides components for setting up the analytic framework, which include managing the data library, building policies, and managing trigger points and actions. These components provide a way to define flexible business rules, called policies, that can be altered without modifying application code. Business analysts and other functional users define policies with an intuitive user interface.

Functional users can create policies that use data elements of various forms and shapes residing in different sources such as the transactional environment, data warehouses, legacy systems, and so on forth. The data elements are exposed to the business user as terms defined in the PeopleSoft Active Analytics Framework data library.

The extensible action framework supports the definition and execution of consequential actions. Application developers can create customized action types within product lines to accommodate their functional needs. PeopleSoft also delivers a built-in action type for displaying alerts to the user.

Understanding Policies and Trigger Points

Policies are the result of combining trigger points, conditions, and defined actions to complete a desired business request. The framework includes components for building policies.

Application developers and functional business analysts use a wizard-like interface to build, manage, and associate trigger points to policies. Business analysts can create interactive policies that react to customer behavior of particular interest to them. For example:

- If a banking customer deposits more than ten thousand dollars, a banking analyst can create a regulatory I.R.S. notification.
- If a high-value customer has logged three or more critical support issues with a call-center within a week, a business executive could send a letter of apology.

Policies supplement, but do not alter, normal transactional processing. Therefore, a policy cannot be used to stop a particular behavior due to some specified restriction. If the condition portion of a policy evaluates to true, the policy causes an action to be performed. If the specified condition evaluates to false, then no consequential actions occur. Policies are evaluated independently from each other. Therefore, if more than one policy is to be evaluated at the same time, the consequential actions of one policy cannot alter or influence the actions of another. Likewise, the sequence of policy execution cannot affect the results of another policy.

You construct a policy by defining one or more conditions, specifying one or more actions, and associating them to a trigger point. A policy cannot be activated without defining at least one condition and action. You can reuse defined policies with multiple trigger points if the elements of the policy agree with the contexts of the trigger points.

Trigger points are events from which the analytic decision engine is invoked by the application. The framework supports the registration of new trigger points, when needed. Examples of trigger points are:

- When a customer is identified.
- When a product is selected.
- After logging in to a self-service application.

Registration of a trigger point involves specifying:

- The type of actions to be invoked.
- The *context* in which the associated policies are to be executed.

During runtime, policies are triggered by specific trigger points within application components, resulting in defined actions being taken.

Note. Registration of a trigger point also involves introducing necessary code in the application to request the decision engine to evaluate the policies pertaining to a trigger point.

Understanding the Data Library

The data library is a repository for information within the PeopleSoft Active Analytics Framework. Each element in the data library is exposed by way of a *term*, which is a pointer to a unit of data within the PeopleSoft system. This data may reside in a relational database, or it may be derived at runtime.

Terms in the data library are organized hierarchically into functional categories called *subject areas*. Terms can be assigned to more than one subject area at a time; for example, if a term represents a customer it could be located both in a marketing subject area and in a financial subject area.

The data library enables functional users to:

- Access data (terms) residing in different sources such as an operational CRM environment, data warehouses, legacy systems and so on.

- Use the data in variety of contexts, such as input in rule applications, tokens in correspondence templates, or placeholders for customized text in questions or for presenting disparate pieces of information in different screen applications.

PeopleSoft Active Analytics Framework includes components to define new terms in the data library and can automatically create terms for data elements in a component.

Understanding the Action Framework

The action framework is a suite of components that are designed to manage actions and to invoke actions at runtime. The primary purpose of the action framework is to allow functional users to specify and configure the actions to be performed when policy conditions evaluate to true.

Also, PeopleSoft designed the action framework to enable application developers and IT personnel to introduce new action types for use in policies and to invoke them at runtime.

The action framework components:

- Register and maintain *action types*. An action type is metadata pertaining to a class of actions that can be performed at runtime.
- Register and maintain *action bundles*. Action bundles are groups of combinable action types.
- Embed and configure consequential actions within policies.
- Provide a display alert action type for use with all product lines.

CHAPTER 10

Building and Managing Policies

This chapter discusses how to:

- Build policies.
- Manage Trigger Points.

Building Policies

You build and manage policies with a wizard-like interface called a policy builder. During the creation of policies, you associate them to trigger points. At runtime, policies are evaluated by specific trigger points in application components, resulting in defined actions being taken.

The policy builder allows business analysts to change conditions or actions or both in policies to enable a business process change in an application component, without having to modify application code, or needing the help of IT personnel.

Note. Policies cannot be shared among different setIDs.

Prerequisites to Building Policies

Before building a policy, you should have already:

- Defined trigger points.
- Defined data library terms.
- Defined action types, categories, and action objectives.

See Chapter 14, "Administering the Framework," Registering Trigger Types and Trigger Points, page 160.

See Chapter 11, "Setting Up the Data Library," Creating Terms, page 136.

See Chapter 13, "Setting Up the Action Framework," page 149.

Pages Used in This Section

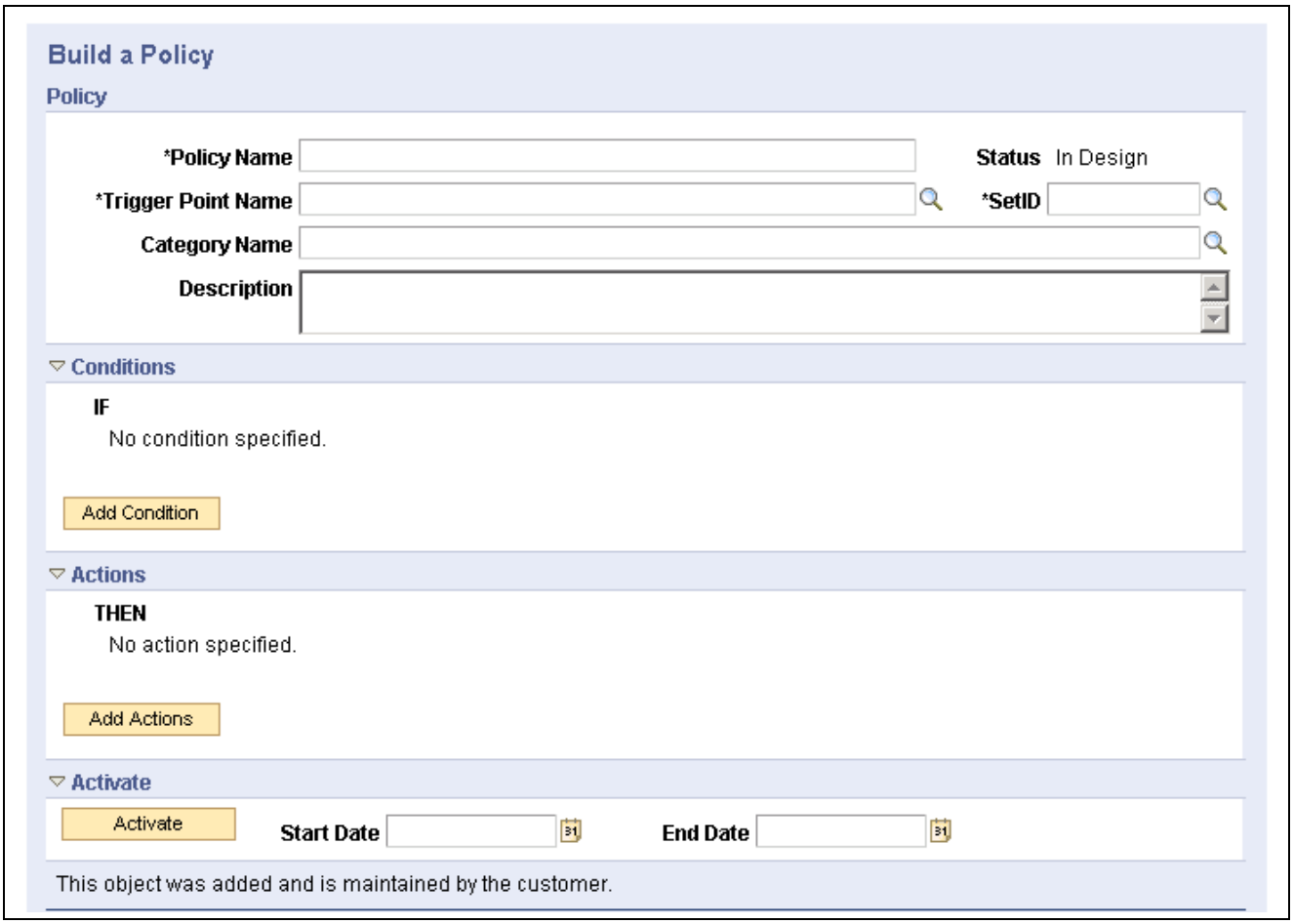
Page Name	Object Name	Navigation	Usage
Manage Policies	EOCF_RULE_DEFN	Enterprise Components, Active Analytics Framework, Policies, Manage Policies.	Used to build and manage policies.
Manage Trigger Point	EOCF_MANAGE_TP	Enterprise Components, Active Analytics Framework, Policies, Manage Trigger Point	Used to manage trigger points.

Building, Editing and Activating Policies

To build a new policy or edit an existing policy, access the Manage Policy page. If you want to edit an existing policy, use the search criteria to find the desired policy, then click the policy name on the results grid to open the policy definition.

Note. If you select a trigger point name as a search criterion, only policies directly associated to the trigger point are retrieved. To retrieve policies associated to *policy groups* of a trigger point, specify search criteria other than the trigger point name.

To create a new policy, click Build Policy.



Build Policy page

Policy Name	Unique and descriptive name of a policy.
Trigger Point Name	Descriptive name of a trigger point. A policy is always associated to at least one trigger point.
Category Name	Descriptive name of a policy category, used to functionally classify policies and aid in searching for policies.
SetID	Defined within the PeopleSoft system. Specify a setID and a trigger point for every policy. This value is used to select the valid set of policies associated to a trigger point at runtime; it also constrains the prompt list for the right-hand side values entered in conditions by performing a setID to setID indirection.
Status	This value defaults to In Design when the policy is new. Activation of a policy changes the status to Active.
Start Date, End Date	Dates defining the validity of a policy at runtime.

Enter the appropriate fields and click Add Conditions.

Adding and Editing Conditions

Click Add Condition on the Build Policy page.

There are two modes of specifying conditions:

- Basic.

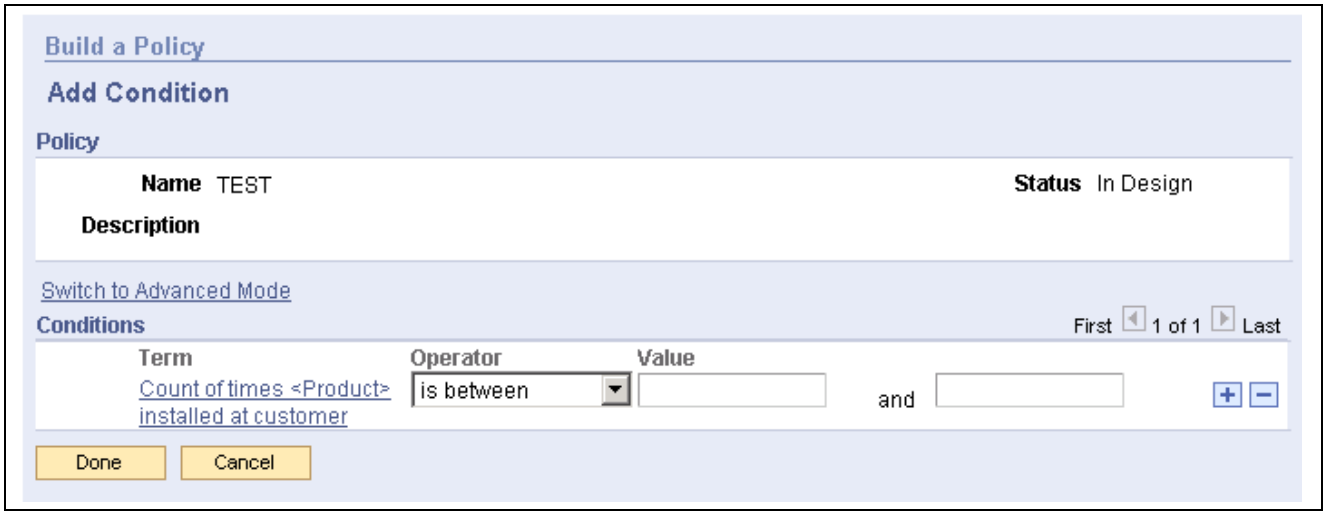
  This is the default mode; the logical operator defaults to AND.

- Advanced.

  You can group condition rows using parenthesis, specify logical operators (AND, OR), and specify terms as values in the right-hand side.

To add a condition row:

1. Select a term and, if it is hyperlinked, configure it.
2. Select an operator.
3. Enter or select one or more values on the right hand side.



Build a Policy page

Selecting Available Terms

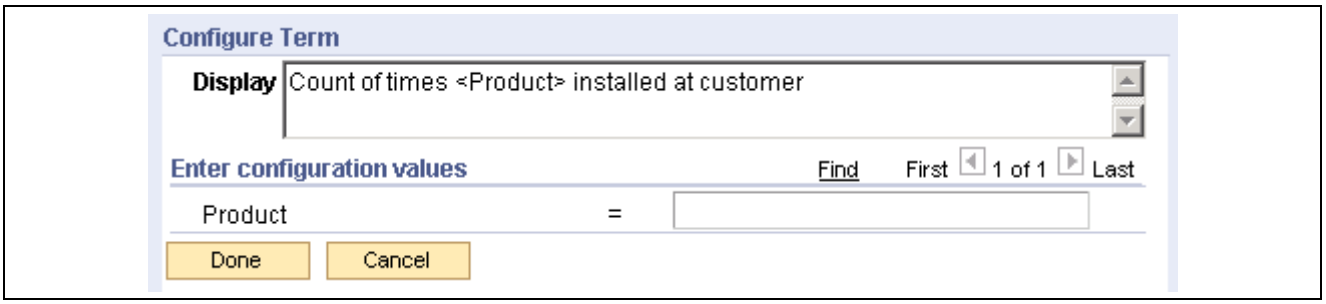
Click Select Term on the Add Condition page to display the Term Selection page.

The Term Selection page has two modes in which to search and select terms for a condition: browsing by subject area or searching by entering search criteria. Terms that appear are limited to those that can be resolved by the trigger point. Therefore, all terms having contextual implementations for the context pertaining to this trigger point, and all terms having generic implementations in which binds can be supplied by the context will be available.

Terms returning multiple rows are not available for use in conditions. Terms that retrieve data from detail rows in the component buffer cause the decision engine to evaluate the condition once for each row. The decision engine generates actions for every row for which the condition is true.

Configuring a Term

If you select a term that is configurable; that is, it has design time binds, it must be configured when you build the condition. Configurable terms are displayed as hyperlinks. Subsequently, while building the condition, click this hyperlink to configure the term.



Configure Term area

The prompt of valid values displayed for configurable terms relies on the prompt configuration specified in the Manage Terms component.

See Chapter 11, "Setting Up the Data Library," Creating Terms, page 136.

Selecting an Operator

The list of operators available when you select a term is dependent on the return data type defined for that term and the context of the selected trigger point.

Each operator defined in the Register Operators page supports certain data types; this determines which operators are available when you select a term in the condition builder. Furthermore, the selection of an operator determines how many fields are required on the right-hand side for entering values.

For example, if you select the is between operator, two right-hand side fields appear for entering values.

See Chapter 14, "Administering the Framework," Registering Operators and Operator Sets, page 164.

Entering Right-Hand Side Values of a Condition

Prompt options that you specify when defining a term determine the right-hand side field type. The field may be a prompt, dropdown list, or multi-select prompt. SetID-based prompt tables specified in the term definition use the setID value specified on the policy definition page to perform a setID to setID indirection, thus constraining the prompt list.

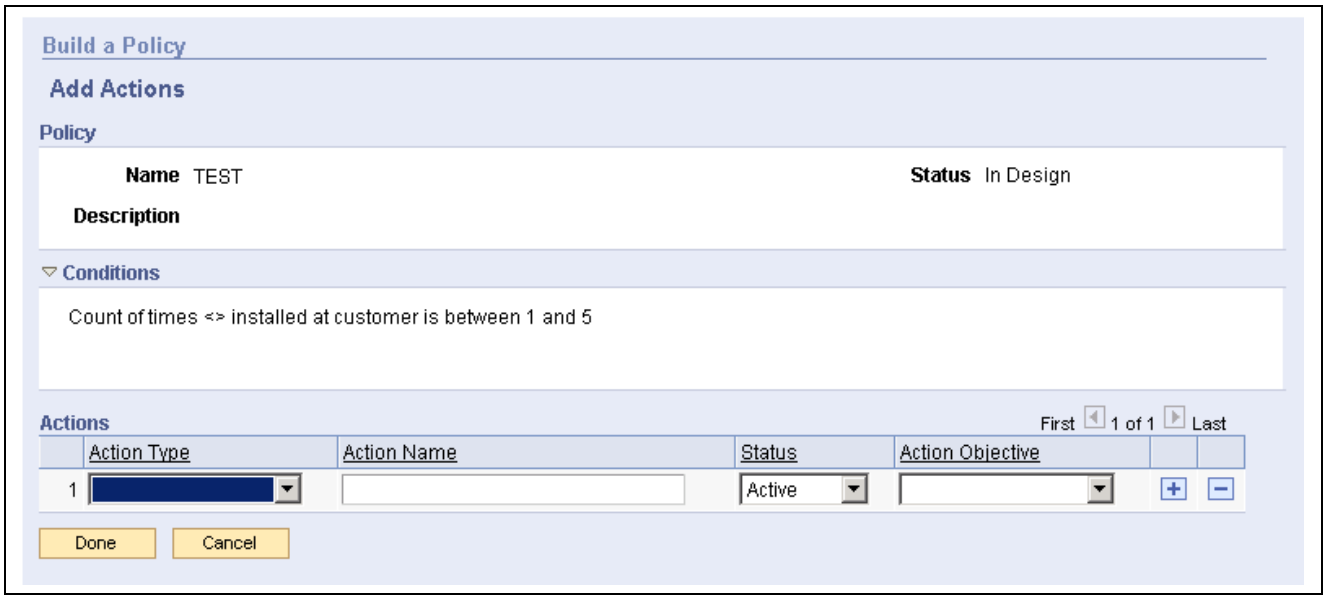
Note. Use a semi-colon as a separator in multi-select prompts.

In advanced mode, you can specify a term on the right-hand side of a condition—this term is resolved at runtime and the resolved value used as the right-hand side value.

After you enter the condition, click Done to save the condition. Before a successful save, the system validates that right-hand side values are present and are of the appropriate data types; that parenthesis match; and that configurable terms have been configured.

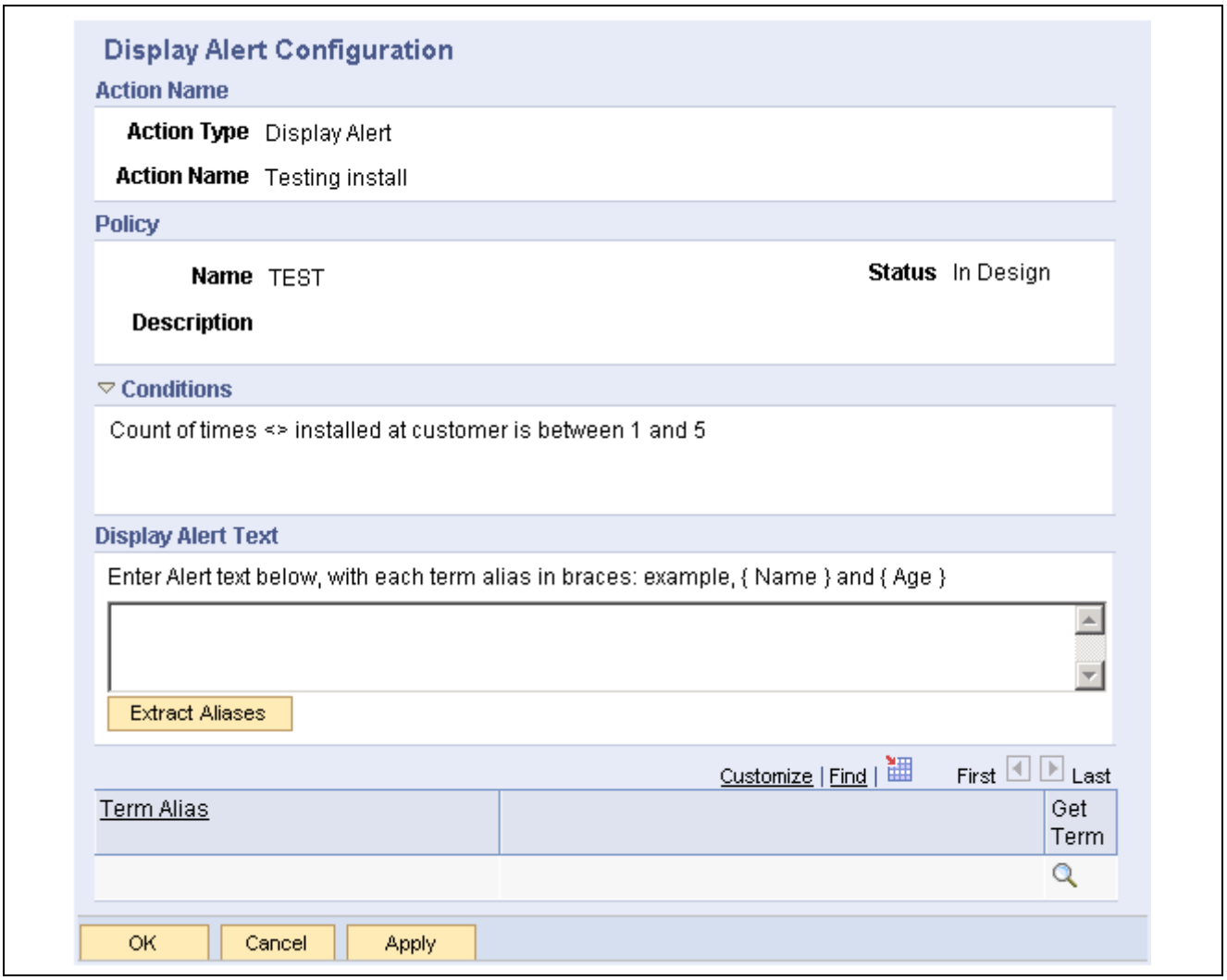
Adding, Editing and Configuring Actions

Click Add Actions or Edit Actions to display the Add or Edit Actions page.



Add Actions page

If the action type is configurable, the Configure button is enabled on the page. Individual action types have specific configuration pages.



Configuration page for Action

Note. PeopleSoft Active Analytics Framework delivers a display alert action type. Other action types may be delivered by individual product lines. Please refer to the appropriate product line PeopleBooks to get more information on delivered action types.

See Chapter 13, "Setting Up the Action Framework," Registering Action Types and Action Type Bundles, page 151.

Configuring the Display Alert Action

Access the Display Alert page by selecting *Display Alert* action type.

1. Click Configure.

   The Display Alert Configuration page appears.

2. Enter text in the Display Alert Text field and include any term aliases in braces.

   Term aliases are placeholders for dynamic content to be merged in the alert text at runtime.

3. Click Extract Aliases to extract term aliases to populate the grid, thus enabling you to map each alias to a term.

4. Click Get Term to map a term for each term alias in the grid.

   Only terms that return a single value can be used within the display text. Return data types of record or rowset, or terms with *Many* rows specified are not allowed.

5. Click OK or Apply to save the display alert configuration.

   This configuration is retrieved at runtime to generate the alert text and display it in a popup box.

Activate a Policy

Access the Build Policy page.

Click Activate. The system sets the status to active after executing validations. Activating a policy disables field editing, however, the policy may still be copied and associated to another trigger point.

Note. On activation of the policy, any modifications made are in effect only in new user sessions. Therefore, you must sign off and sign on to see any changes made to the policy.

Associating a Policy to Another Trigger Point

A policy can be associated to more than one trigger point within the same setID. Do this by:

- Adding a new row to the Associated Trigger Points grid.
- Selecting a valid trigger point from the dropdown list

Note. The trigger points available for selection in the drop-down list are constrained by the terms used in the policy condition and the actions configured in the policy. Also, a policy cannot be associated with multiple trigger points spanning multiple setIDs.

- Saving the policy associates the policy to the new trigger point.

Copying a Policy

Create a new policy by copying an existing one, provided you can use the same condition and actions. While copying a policy, you'll be prompted for a trigger point and setID. Selecting from the list of valid trigger points results in creating a new policy, which appears on the screen.

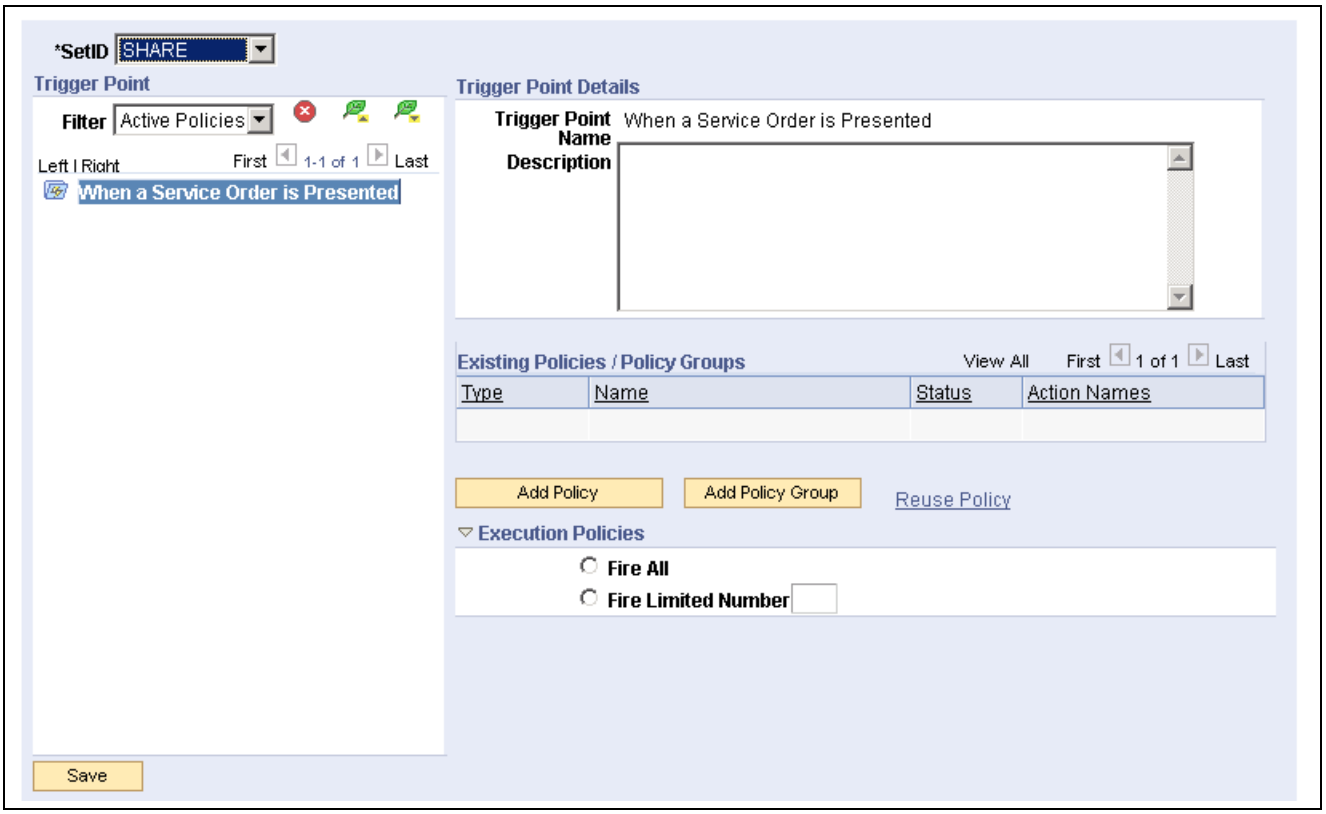
Note. When you copy a policy, the condition and actions, but not the action configurations, are copied to the new policy. Therefore, you need to reconfigure the actions by clicking Edit Actions. A reminder message appears when you're transferred to the new policy.

Managing Trigger Points

The Manage Trigger Point component provides a comprehensive view of policies that are associated to a specific trigger point. It displays policies and policy groups in a hierarchy, with the trigger point as the root and policy groups (if any) as parents of policies.

In addition, this component enables you to assign execution options at the trigger point level and at the policy group level, facilitating policy arbitration and better policy management.

Access the Manage Trigger Point page.



Manage Trigger Point page

The trigger point hierarchy on the left-hand side of the page displays all the policies and policy groups associated to the selected trigger point. The trigger point appears as the highest level item in the hierarchy while policy actions appear as the lowest level items.

Filter This field applies only to policies displayed in the hierarchy and the Existing Policies/ Policy Groups grid. It either displays active policies, in-design policies, or all policies depending on the selection.

SetID Toggle the value in this field to view policies for this setID.

Removing a Policy or Policy Groups

You can remove a policy or policy group from a trigger point by selecting the policy or policy group to be deleted, and then by clicking the Delete icon. Removing a policy or policy group from the trigger point disassociates it from the trigger point, but does not delete it from the database.

A policy group is not reusable—once the policy group is disassociated from a trigger point, it can not be referenced.

Any changes made to a trigger point by adding or removing policies or policy groups or by modifying execution options, take effect at runtime and only for new user sessions. Therefore, you must sign off and sign on to see any changes made.

Note. A policy that is associated to a single trigger point (either directly or within policy groups) cannot be removed from the trigger point. To disable such a policy, you must edit it and set the status to In Design.

Reordering Policy or Policy Groups

To set priorities to policies, you may need to reorder policies or policy groups within a trigger point or a parent policy group. Reorder policies and policy groups by using the reorder icon. a parent group, which may be a Trigger Point or a parent Policy group.

Adding a Policy or Policy Group

Click Add Policy to create a new policy and associate it to the trigger point. The Build Policy component appears.

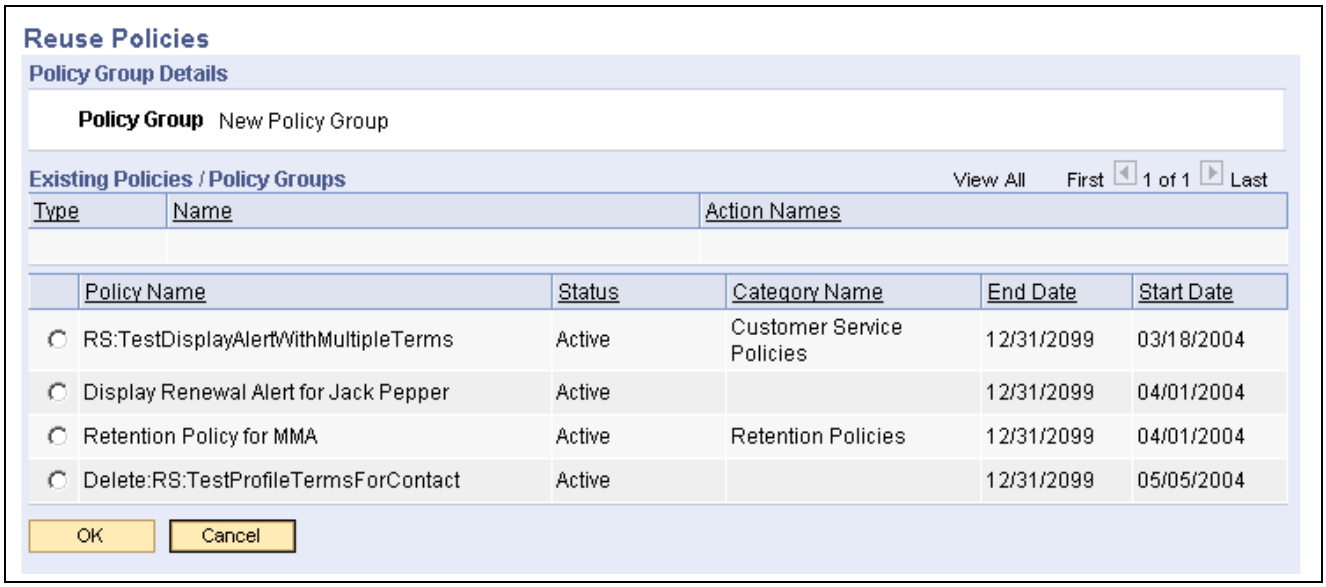
Click Add Policy Group to create a new policy group and to associate it with this trigger point.

A policy group may be used to set policy priority within a group, to de-activate policies, to nest child policy groups and to assign preconditions.

Reusing Policies

Reuse a policy in multiple trigger points and policy groups if the contexts are compatible.

Click the Reuse Policy hyperlink to reuse an existing policy. Select one of the policies listed in the grid by selecting the appropriate radio button. Click OK. The selected policy is associated to the trigger point or policy group.



Reuse Policy page

Note. If a policy is reused within a single trigger point through direct association to the trigger point, or by association to a policy group within the trigger point, you cannot remove this policy from either the trigger point or the policy group within the trigger point. If you want to remove such a policy, you must deactivate the policy by setting its status to In Design.

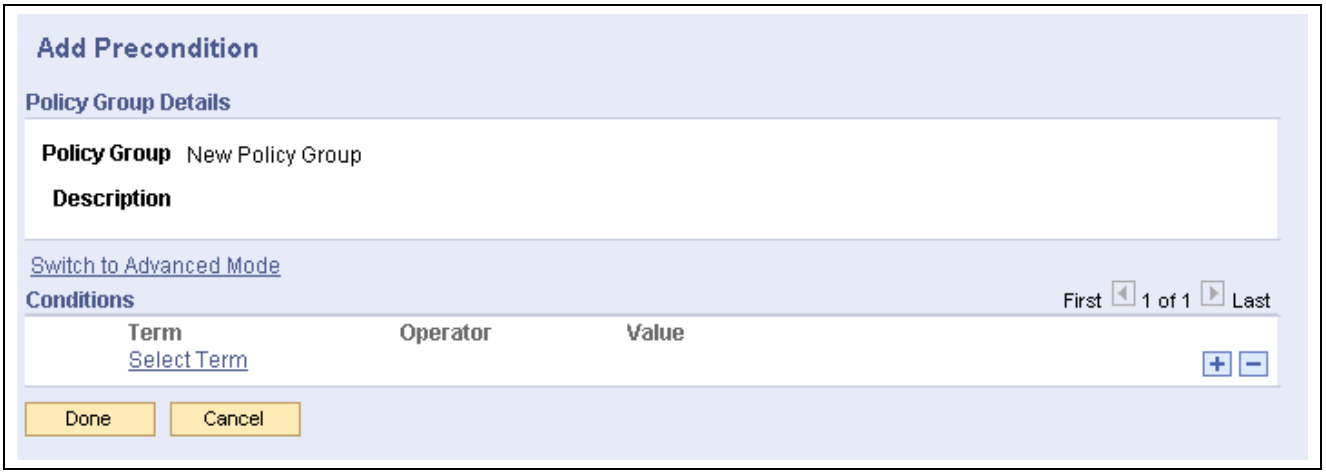
Adding a Precondition

Preconditions are combinations of one or more conditions. They are optional and only policy groups can have preconditions.

At runtime, the policies within a policy group are not evaluated unless the precondition evaluates to true.

For example, you could have a precondition defined for a self-service policy group, as "Is this user on the internet?" Consequently, all policies within that policy group would not be executed unless the precondition of being on the internet evaluates to true.

Click Add Precondition to display the Add Precondition page.



Add Precondition page

Select terms and operators; specify values on the right-hand side.

Setting Execution Options

The following execution options may be specified for a trigger point or a policy group:

Execute All	Default, if specified for a trigger point. Enables execution of all policies within a trigger point or policy group. During runtime, when executing policies within a policy group, the system adheres to the execution option specified for the policy group.
Execute Limited Number	Enables execution of a maximum (of the number specified) policies that evaluate to true.
Use Parent Execution Options	(Policy groups only). Default for policy groups. At runtime, this uses the execution options of either the trigger point or the parent policy group.

Understanding Execution Options

This section describes the execution options and various scenarios of how they can be used.

Execute All

Execute All tests all policies in a trigger point; that is, all policies in the trigger point can cause actions to occur if their conditions prove true.

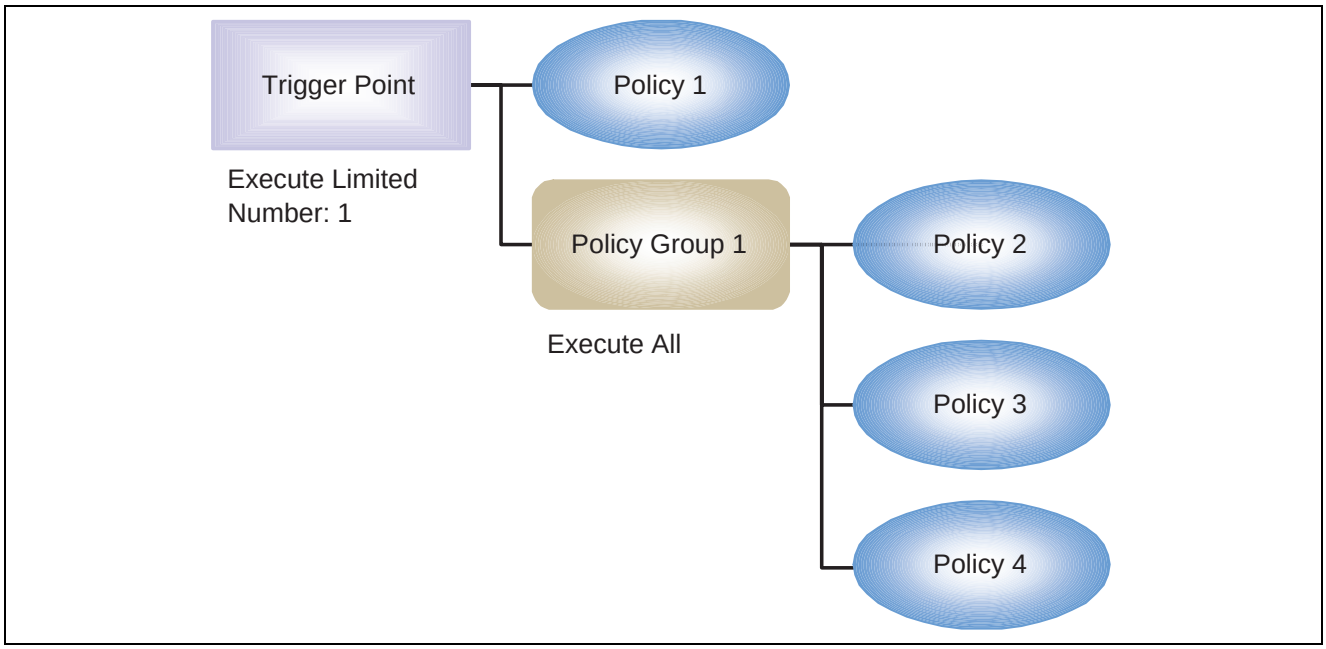
For example if a trigger point has three policies and the execution option is set to Execute All, all three policies are evaluated.

Execute Limited Number

Execute Limited Number evaluates all policies in the trigger point until one of the policies' conditions evaluates to true. Therefore, say you set Execute Limited Number to 1 for a trigger point that has three policies where Policy 1 evaluates to false and Policy 2 evaluates to true; Policy 3 will not be considered and Policy 2 is executed.

Various Scenarios of Execution Options

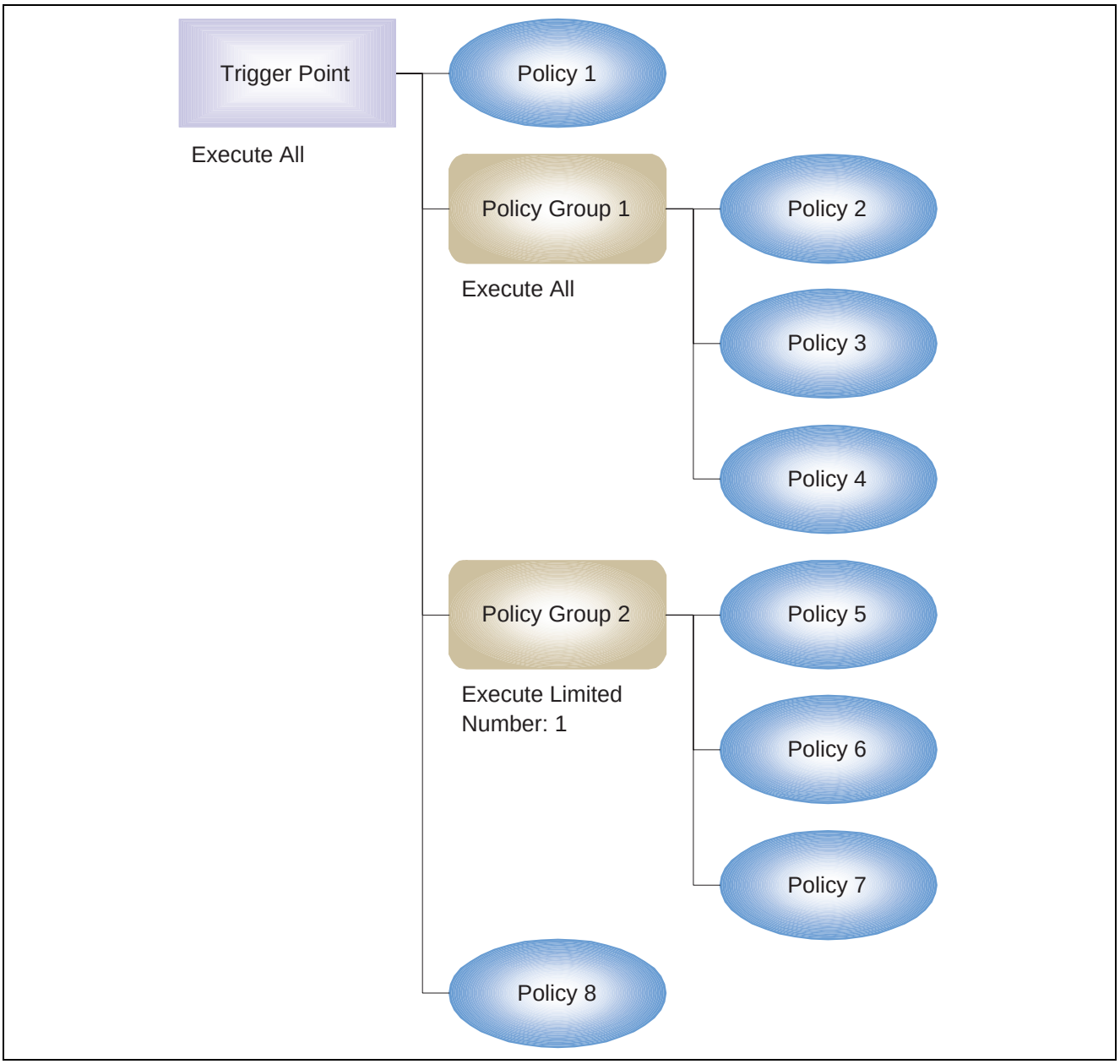
Policy groups may have their own execution options that could affect the option setting, however. For example, consider the following diagram:



Execution Options: Scenario 1

In Scenario 1, if Policy 1 proves false, then all policies in Policy Group 1 are evaluated because its execution option overrides the trigger point's execution option. Therefore, even though the trigger point is set to execute one policy, if Policy 2, 3, and 4 evaluate to true those three policies' actions will execute.

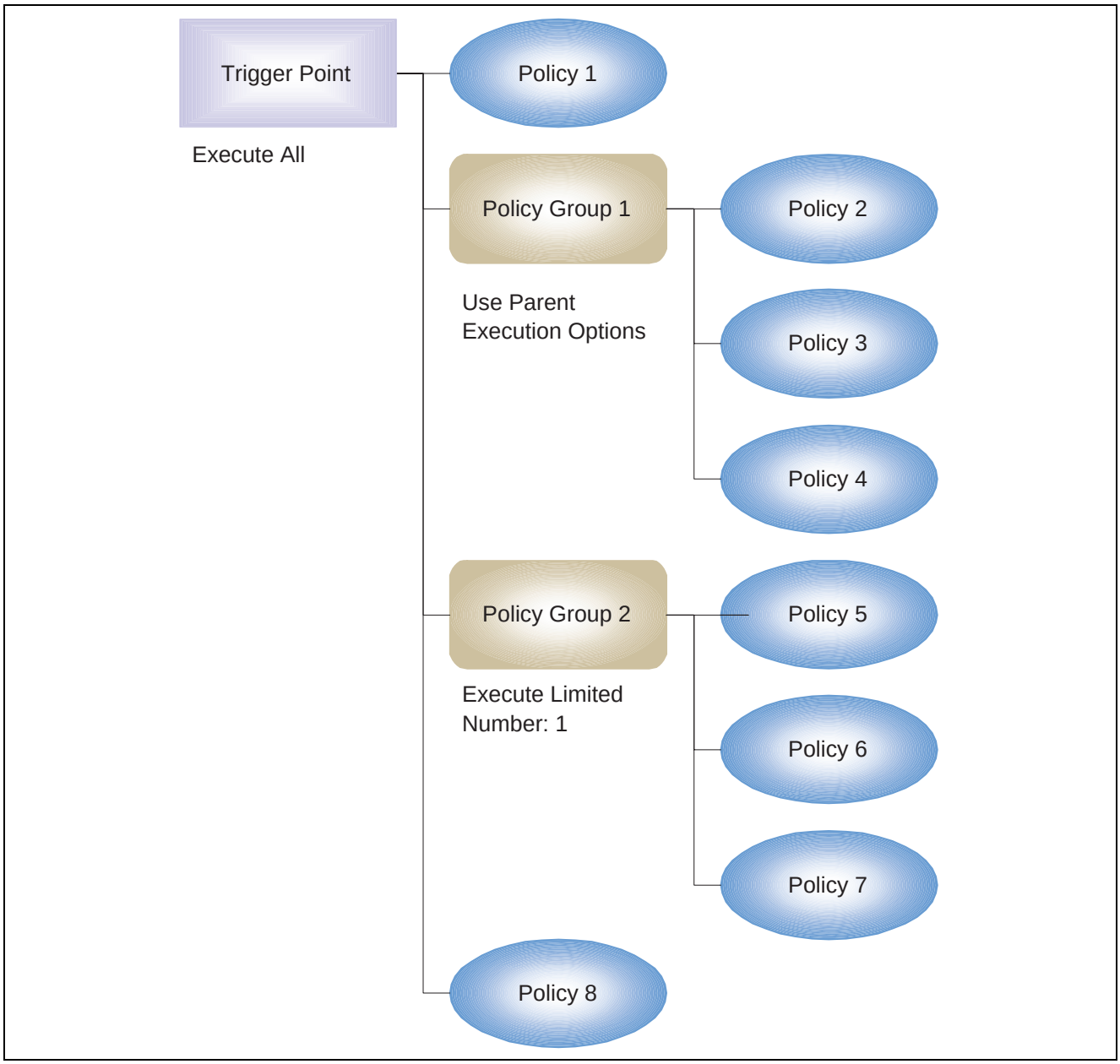
Consider Scenario 2:



Execution Options: Scenario 2

In this scenario, assume that all policy conditions are true, actions from Policy 1, 2, 3, 4, 5, and 8 will execute.

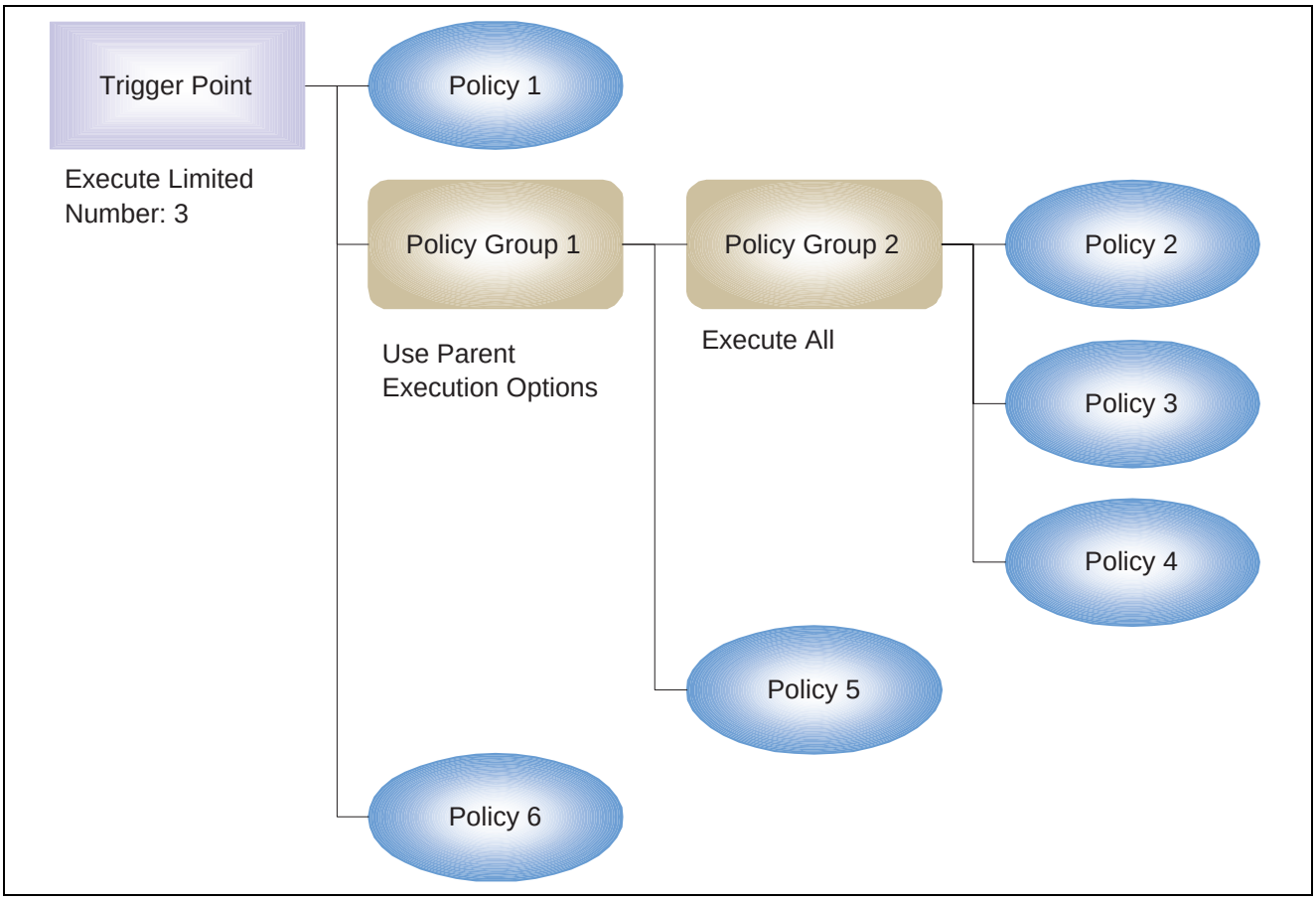
Consider Scenario 3:



Execution Options: Scenario 3

In this scenario, the trigger point's execution option is the default (as is Policy Group 1). Assuming that all policy conditions are true, this trigger point executes exactly as in Scenario 2; that is, actions from Policy 1, 2, 3, 4, 5, and 8 will execute.

Consider Scenario 4:



Execution Options: Scenario 4

In this scenario, assume that all policy conditions are true. Policy Group 2's execution option overrides Policy Group 1's; therefore, all of Policy Group 2's policy actions execute. However, Policy 5 is not considered, nor is Policy 6 because three policy actions have already executed (Policy 2, 3, and 4).

Guidelines for Setting Execution Options

Unless you have a specific reason to set options, PeopleSoft recommends using the default execution options. If there are only a few conditions that could be true for a trigger point, and these conditions are logically exclusive (only one could be true at a time), set the number of policies considered to one (Execute Limited Number =1) to improve performance.

If for whatever reason, only a few policy options should be considered, and there are no policy groups, setting a limitation for the trigger point is a good idea.

If you have a trigger point containing unrelated policies and a category of possible conditions that may be true, for which only one set of consequent actions should be taken, it's best to separate the unrelated policies into a separate policy group with a limitation on the number of policies allowed to fire.

CHAPTER 11

Setting Up the Data Library

This section gives an overview of the data library and discusses how to:

- Use the data library components.
- Create implementations.
- Create terms.
- Manage terms.
- Test term implementations.

Understanding the Data Library

The data library is a repository for information within the PeopleSoft Active Analytics Framework. Each element in the data library is exposed by way of a *term*, which is a pointer to a unit of data within the PeopleSoft system. This data may reside in a relational database, or it may be derived at runtime.

See Also

Chapter 9, "Understanding PeopleSoft Active Analytics Framework," page 113

Using Data Library Components

This section describes the components used to populate the data library.

Pages Used in This Chapter

Page Name	Object Name	Navigation	Usage
Manage Terms	EOCF_TERM_DEFN	Active Analytics Framework, Data Library, Manage Terms	Used to define and manage terms.
Subject Area	EOCF_TERM_SUBAREA	Active Analytics Framework, Data Library, Manage Terms, Subject Area	Used to define the subject area details.
Policy Options	EOCF_TERM_INACTION	Active Analytics Framework, Data Library, Manage Terms, Policy Options	Used to define policy options.
Extended Attributes	EOCF_TERM_ATTR	Active Analytics Framework, Data Library, Manage Terms, Extended Attributes	Used to add additional attributes to terms.
Define Implementation	EOCF_IMPL_DEFN	Active Analytics Framework, Data Library, Define Implementation	Used to define an implementation.
Test Term Implementation	EOCF_TEST_TERM	Active Analytics Framework, Data Library, Manage Terms, Test Term Implementation	Used to test a term's implementations.

Creating Implementations

An *implementation* refers to the mechanism through which the data is retrieved, derived or computed. The implementation knows either where the data physically resides or it knows the algorithm for deriving the value. All terms must be associated with an implementation unless the data to which the term refers is present in the component buffer.

An implementation can be associated with more than one term. Conversely, a term may require multiple implementations if it needs to be resolved from multiple contexts. Typically, application developers or IT personnel develop implementations.

Note. Terms that are resolved by accessing data available in the current operating component's buffer, do not need implementations to be developed. PeopleSoft Active Analytics Framework provides mechanisms to access data available in the component buffer.

PeopleSoft recommends that when multiple related terms will be accessed during a single business event, create a single implementation to return a rowset containing the data for several terms; then, specify which data element or field position in the rowset or record is to be used for the term.

Implementations are developed using:

- Application class.

Use application class implementations when retrieval or derivation of data involves writing procedural code; or, as a resolution method when data must be retrieved from an external source such as another PeopleSoft database or legacy systems. The application class can return data to the data library in a variety of forms: rowset, record, date, datetime, string, number, date array, datetime array, string array, number array and array of any. Use PeopleSoft Application Designer to develop the application classes.

Note. PeopleSoft does not recommend: 1) Using an application class to retrieve data from a component buffer; 2) Using an application class to retrieve the values for the binds directly by accessing the component buffer without registering them as implementation binds. Application classes must use APIs to retrieve the values for the implementation binds (input parameters).

- PS Query .

  Use Query Manager in PeopleSoft Application Designer to develop PS Query-based implementations. PS Query implementations are not appropriate for applications that get data from external databases or systems. The data library invokes the appropriate queries based on the information provided when you register the implementation. The data returned is available to the data library in the form of rowset.

- SQL object.

  Use this implementation when the SQL used needs to be platform independent and the data need not undergo complex transformations. SQL Object implementations are not appropriate for applications that get data from external databases or systems. The data library invokes the appropriate SQL object based on the information provided when you register the implementation. The data returned is available to the data library in the form of a record or an array of any objects. Use PeopleSoft Application Designer to create an SQL Object.

- Record.Field

  Create a Record.Field–based implementation when the data can be retrieved directly from a table without going through complex transformations. The data returned is available to the data library in the form of string, date, datetime, number, string array, date array, datetime array and number arrays.

Registering an Implementation

With the exception of component buffer implementations, all implementations must be registered in the PeopleSoft Active Analytics Framework. Before registering an implementation, you should have already defined PeopleSoft Application Designer objects if using application class, PS Query or SQL Object implementation methods.

Specify the following in the registration component:

- Functional name.
- Resolution method used for that implementation
- Values for the parameters needed for invoking the implementation. The list of parameters vary depending upon the resolution method.
- List of binds that are expected by this implementation.

Note. The binds specified for an application class implementation are referenced in the application class object for retrieving the values. Therefore, changing these implementation bind names can have adverse effect on the term resolution.

For IT users: The list of implementation binds specified are being used for two purposes: 1) To allow implementations to access these bind values. For any implementation, bind values are passed by position regardless of the resolution method used. Application class-based implementations alone have the additional capability to access the bind values by name. 2) To allow the data library engine to use these binds to uniquely tag the data in application cache. If IT users "take a short cut" by retrieving the necessary data by directly accessing the context (by not registering the data as implementation binds), the data library engine may, as a result, tag the data with incomplete key information. This could cause the same cached data to be incorrectly reused for resolving terms for which it is not valid.

Creating Terms

A term is a user-friendly name that refers to the data library content. It's essentially a piece of information that could exist in the PeopleSoft system or an external system, or it could be derived. For example, the data could be available in the component buffer; retrieved using a PS Query or an SQL object; or, computed using an application class.

Terms are the building blocks in policies. Functional users can build conditions for a policy using terms present in the data library. Terms must be registered in the PeopleSoft Active Analytics Framework before they can be used.

Registering a term is a multi-step process that includes:

- Developing an implementation.
- Registering the implementation.
- Defining the term.
- Associating the term to one or more subject areas.
- Testing and activating the term.

Understanding Term Properties

Defining a term involves specifying the following:

- Term name, code and type (constant or variable).
- Data type.

  The data library supports primitive data types of string, number, datetime, date , time; and PeopleSoft-specific data types of record and rowset.
- Number of rows to be returned, scalar or vector (returning an array). (Terms that are record or rowset data types have number of rows set to *One*).

  Note. Terms returning a vector (where value of number of rows is many) do not appear in the term list while building a condition for a policy.

- User binds.

These are values that would be supplied either during the construction of a condition or at the time of associating the term with the application. Not all terms will have the binds; however, user binds may make a term more reusable.

- Optionally, details about how the data needs to be captured for user binds:whether a prompt or dropdown list needs to be shown and how to derive the values.
- Which implementation needs to be used for resolving the term.
- Whether the term can potentially be resolved from any context; or only from specific contexts
- How the data library needs to extract the data from the content returned by the implementation.
- Prompt details for the term.

  Specifying prompt details for a term is needed only when the term will be used to build a condition. The prompt details convey how the data needs to be captured on the right-hand side for a term participating in a condition.
- Configuration details for prompts.

  The details you provide are used during the construction of a condition. When a term is selected as an element in a condition, the right-hand side widget will be constructed based on the configuration details specified for the prompts. You can configure the following prompt types :

  - Translate.

    Specify a translate field name in which its values appear to the end user on the right-hand side of a condition.

  - Dropdown.

    Specify a record name, data field name, and description field name. The record and data field names supply the valid data values to display in the dropdown list; the description field provides a user-friendly description of the data value.

  - Prompt.

    Specify a record name, data field name, and description field name. The record and data field names supply the valid data values to display in the prompt; the description field provide a user-friendly description of the data value.

  - Custom.

    Specify a custom application class, a data field name, and a description field name. Valid values are retrieved by executing the specified application class method and presented as a prompt.
- Scope of each term implementation.

  - When caching is activated for a term, data that is cached is uniquely identified by the implementation ID and all of the implementation bind values for that implementation.
  - When scope is specified as the trigger point, after the first invocation of a term, subsequent references to the same term in one or more policies associated to the same trigger point forces the data library engine to retrieve the data from the cache, provided all the values for the implementation binds match those of the ones belonging to the data present in the cache.
  - When scope is defined as a component, the longevity of the data is for a specific instance of the transaction.
  - When scope is defined as global, the cached data is available for the entire user session.
  - When scope is defined as *Do Not Cache*, data is retrieved by invoking the implementation every time.
- Association of a term to subject areas.

Subject areas act as file cabinets. You must assign a term to at least one subject area, but you can associate it to more than one.

Note. PeopleSoft Active Analytics Framework does not format the data. It is the term user's or term implementer's responsibility to format it according to their needs. For example, the term Current Date is always resolved using the standard YYYYMMDD format.

Using Generic Implementations

A generic implementation can resolve terms within the requesting context. You define generic implementations for terms when they can be used in various contexts and when any new contexts may want to use that term.

Examples of generic implementations are:

- Customer-specific measures such as customer value, the number of cases reported in a period of time or the number of telephone interactions with the customer.
- Customer profile information such as, first and last name, email address, customers within a segment.

Using Contextual Implementations

If the input data needed for invoking an implementation is too specific and cannot be supplied outside of the component, then the implementation must be associated with the component's context. For example, terms such as case status, order creation date, or case description, cannot be resolved from components other than those in which they are present.

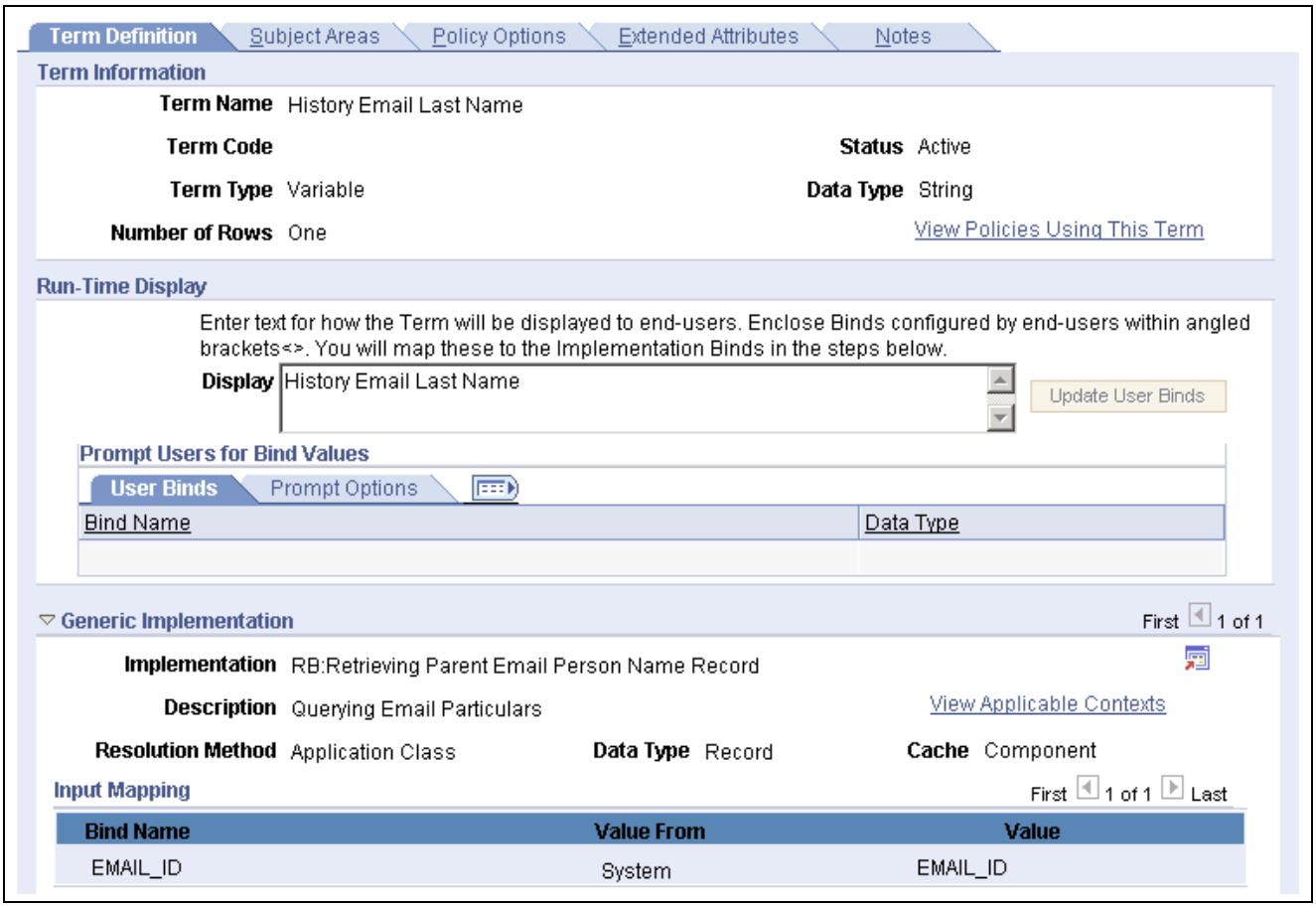
Terms that have different implementations depending upon their contexts will have an implementation associated with a specific context. For example, the term *Revenue for a customer/ segment / segment group*is computed differently depending on the context from which it originates. The implementation specific to the customer context calculates the revenue value from that customer. The implementation specific to a segment context calculates the revenue value generated from all the customers belonging to that segment, and so on for segment group.

Managing Terms

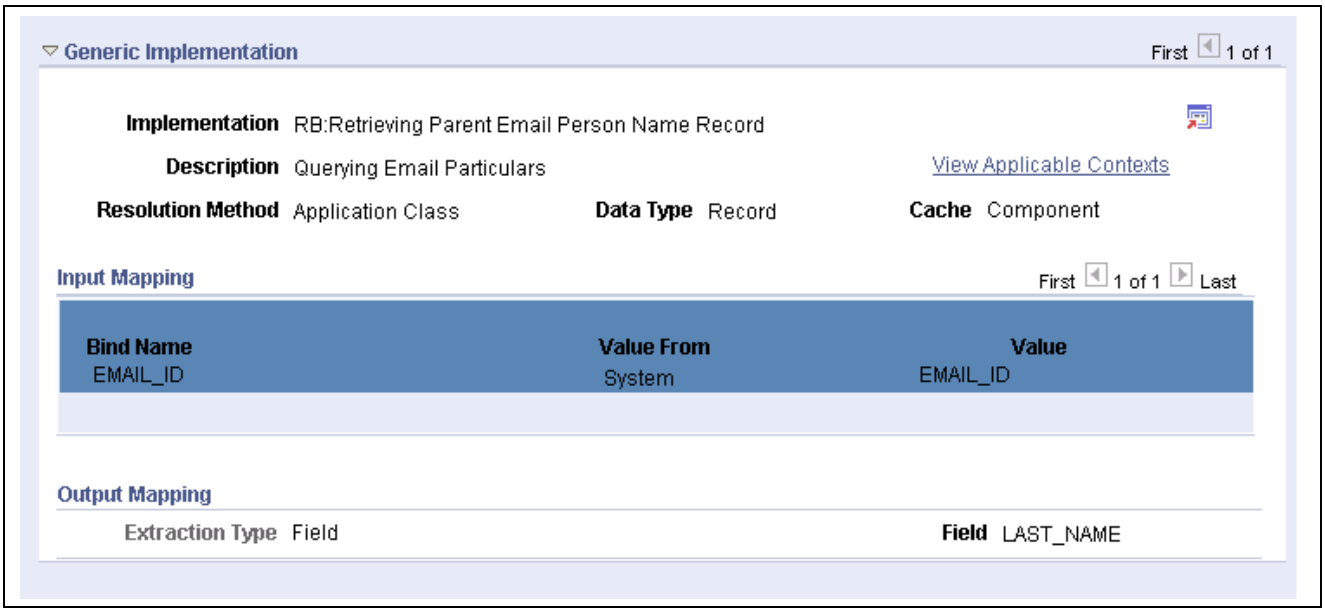
Before defining a term, you should have already:

- Created context definitions if you're using a contextual implementation.
- Registered any implementations if the term is not accessing the component buffer.
- Created prompt records in PeopleSoft Application Designer if prompt options need to be specified.

Access the Manage Terms page.



Manage Terms page (1 of 2)



Manage Terms (2 of 2)

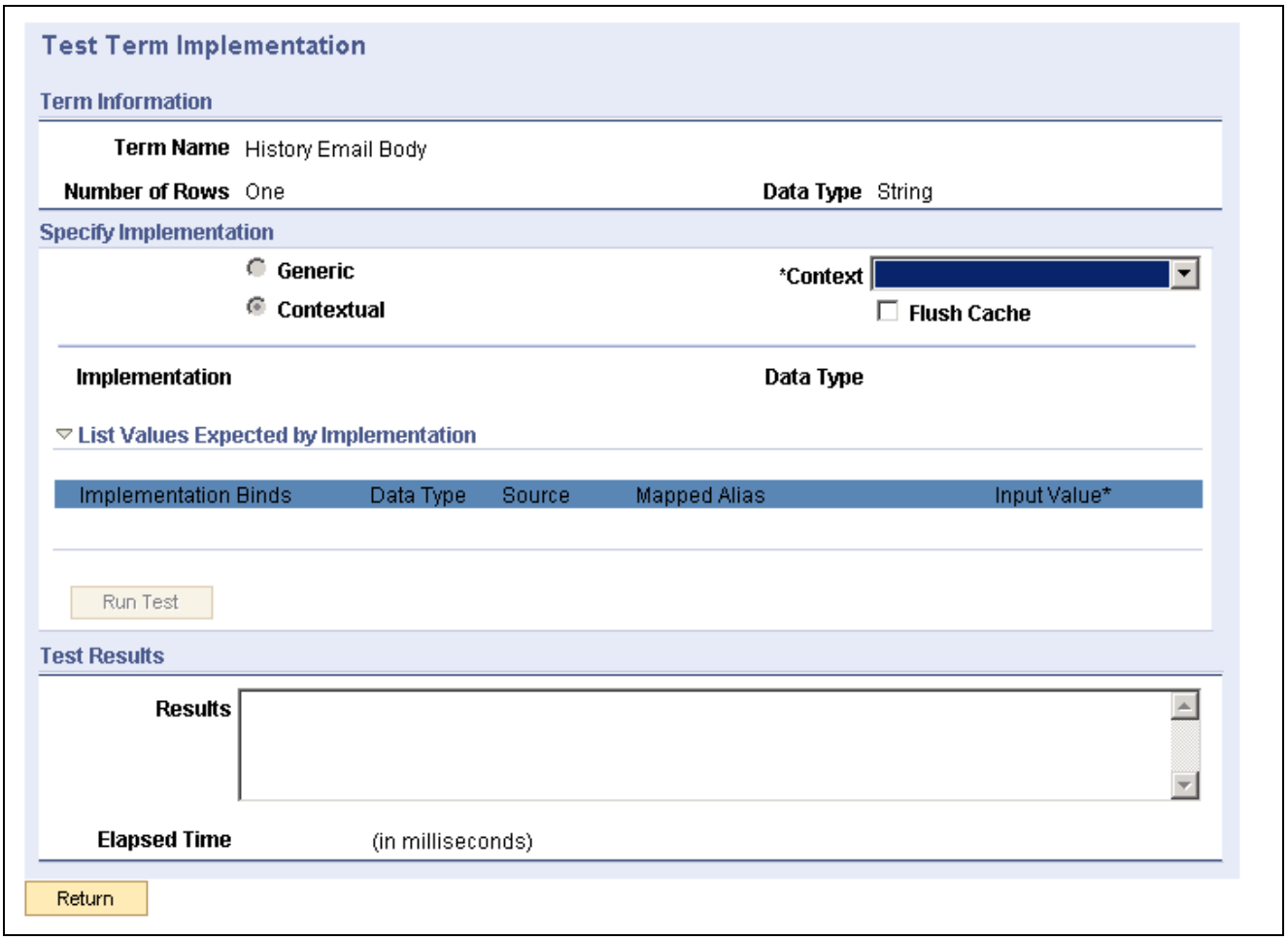
Term Name Unique identifier of the term; label that will be displayed to the functional users. Though allowed, PeopleSoft recommends that special characters not be used in term names.

Term Type	Specify that a term is a variable or constant. Variable terms must have at least one implementation.
Term Code	Uniquely identifies a term when accessing a term programmatically. This is user-defined.
Number of Rows	Number of rows to be returned, one or many (scalar or vector). If this field is *Many*, the term cannot participate in policy conditions.
	Note. Applications directly integrating with the data library are responsible for converting the resolved output value of a term (which will be of data type *any*) to the appropriate data type.
Status	Valid values are *Active* , *Inactive* and *In-Design*. Only active terms are used in policy conditions and other applications.
Data Type	Return data type of the term. Possible values are string, number, date, datetime, time, record and rowset.
Run-Time Display	Specify user binds for this term, which will be needed when the resolved value of the term depends on user defined binds.
Prompt Users for Bind Values	Specify the bind name; (optional) specify prompt options.
Generic Implementation	Specify a generic implementation. Generic implementations are resolved by deriving the bind values from the runtime context. Terms having generic implementations can be resolved by multiple contexts. You specify a generic implementation by selecting an existing implementation from the prompt or creating a new one using the Create button. Click on the View applicable contexts hyperlink to view the list of contexts in which this term would be resolved.
Contextual Implementation	Select an implementation specific to a context. Contextual implementations are resolved by deriving the bind values from this specific context.
Input Mapping	Maps the implementation binds to context variables or to constant values. If the term has user binds, one or more implementation binds must be mapped to the user binds. For generic implementations, this mapping is critical for this term to be resolved by multiple contexts.
Output Mapping	Specify the extraction parameters for a term implementation—such that a subset of the value returned by the implementation is returned as the resolved value of the term.

Note. Use caution when making changes to the term definition after the term has been associated to one or more policies. Changes to term attributes such as data type, number of rows, implementation category, and implementation details; or, changing a non-configurable term to a configurable term and vice versa, could have significant impact on the policies which reference this term. It is possible that these changes could result in invalidating these policies. Before making any of these changes, view the policies using a term by clicking the hyperlink View Policies Using This Term.

Testing Term Implementations

Access the Test Term Implementation page.



Test Term Implementation page

Specify Implementation Specify whether you want to test the generic or contextual implementation defined for the term.

- Contextual. If you select this you must specify a context name from the drop-down list. Displays the list of bind values required by this implementation in a grid.
- Generic. If you select this, the context from where the term is resolved must be specified.

Check Flush Cache if you do not want the system to fetch the value for this implementation from the memory cache.

List values.. Enter sample values for the parameters expected by the implementation and click Run Test.

Test Results Displays the resolved value of a term implementation being tested and the elapsed time to retrieve the value.

Note. Context variable implementations of a term cannot be tested. Also, terms that have application class implementations accessing data from a component buffer or directly from the context, cannot be tested in the Term Tester page. Testing these terms will result in an error message.

CHAPTER 12

Managing Contexts

This chapter provides an overview of contexts and discusses how to configure contexts.

Understanding Contexts

Contexts are a key component of how the PeopleSoft Active Analytics Framework works—their purpose is to describe the computing environment from which the decision engine is invoked and select the appropriate term implementation at runtime.

Online and Generic Contexts

Contexts can be online or generic. Online contexts must be associated with a PeopleSoft online page, whereas generic contexts can be used anywhere (including online pages).

The data elements within a context are called *context variables*. Online contexts have built-in context variables—they are the fields of the online page. However, the data elements in an online context are not limited to the online fields; both online and stand-alone contexts can have additional context variables defined.

Within a given context, all context variables must be assigned unique code names called *aliases*. Aliases are used to associate data with the input binds required by term implementations. For example, a field representing the customer ID may be named CUSTOMER_ID in the application page, but may have an alias of CUST_ID. Using aliases enables terms to be used in the largest possible set of contexts. Page variables exist on a one-to-one basis with their underlying fields; that is, two or more context variables may point to the same page field as long as the aliases of these context variables are distinct. This enables the context user to have aliases named both CUSTOMER_ID and CUST_ID, and thereby facilitate term re-use. Context variables are exposed to the user by corresponding term definitions.

At runtime, applications may request the data library engine to automatically populate the online context in memory, provided the request is made from the component from which the online context can be constructed. In case of generic contexts, applications can either construct the context explicitly by populating values for these context variables, or request the data library to automatically copy values from level 0 context variables of an online context to similar context variables of a generic context.

Context variables can be:

- Page variables that correspond to fields within PeopleSoft application pages.

  Page variables are also referred to as native context variables, as they are native to the pages from which their values come.

  Note. Use caution when altering the component structure after generating page variables for an online context (using Generate Context component). Changes done to existing fields on the application page might invalidate the corresponding context variables and the terms from which they're resolved.

- Constants.

  Generic contexts usually consist of named constants. Constant-type context variables may not have specific values associated to them when the context is registered; some of these variables may get values at runtime.

- Term variables.

  These context variables are data library terms that have been inserted into the context. A context may have any number of terms included within it. However, if a term's implementations require binds, then the context must provide them from its page variables and constants. Term variables can be used to add extended data elements for implementation binds and actions without having to customize the application.

Configuring Contexts

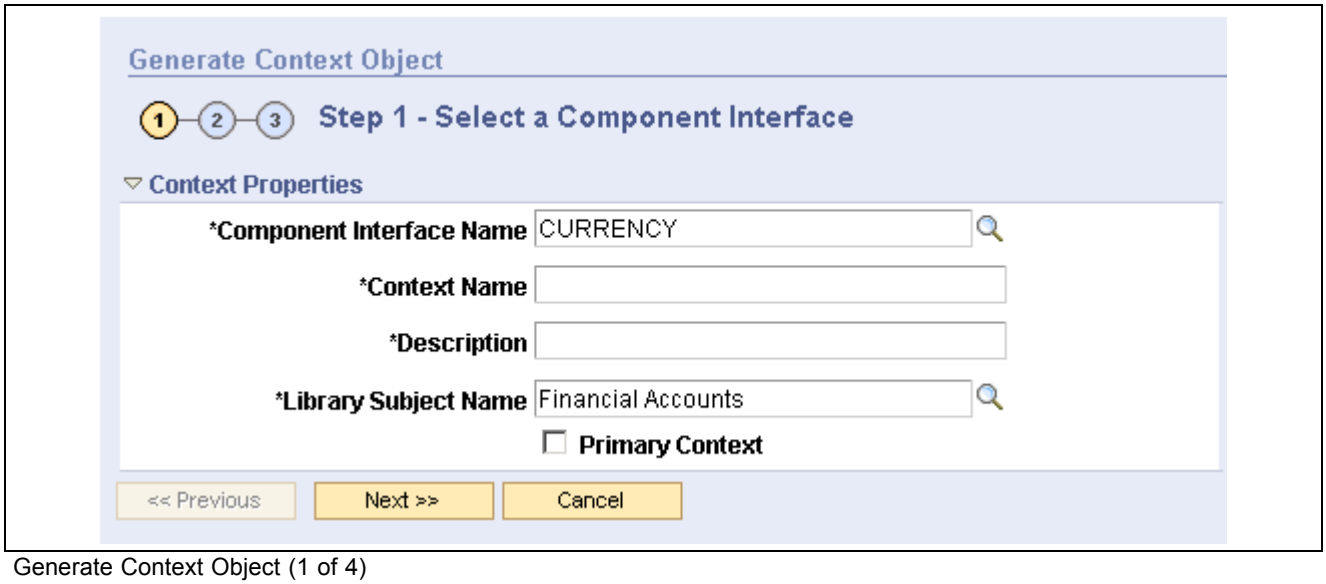
This section describes the components used to configure contexts.

Pages Used in This Chapter

Page Name	Object Name	Navigation	Usage
Generate Context Object	EOCF_CTX_IMPORT	Active Analytics Framework, Setup, Generate Context Object	Used to generate context objects.
Manage Context Object	EOCF_CONTEXT_DEFN	Active Analytics Framework, Setup, Manage Context Object	Manages context objects.

Generating Context Objects

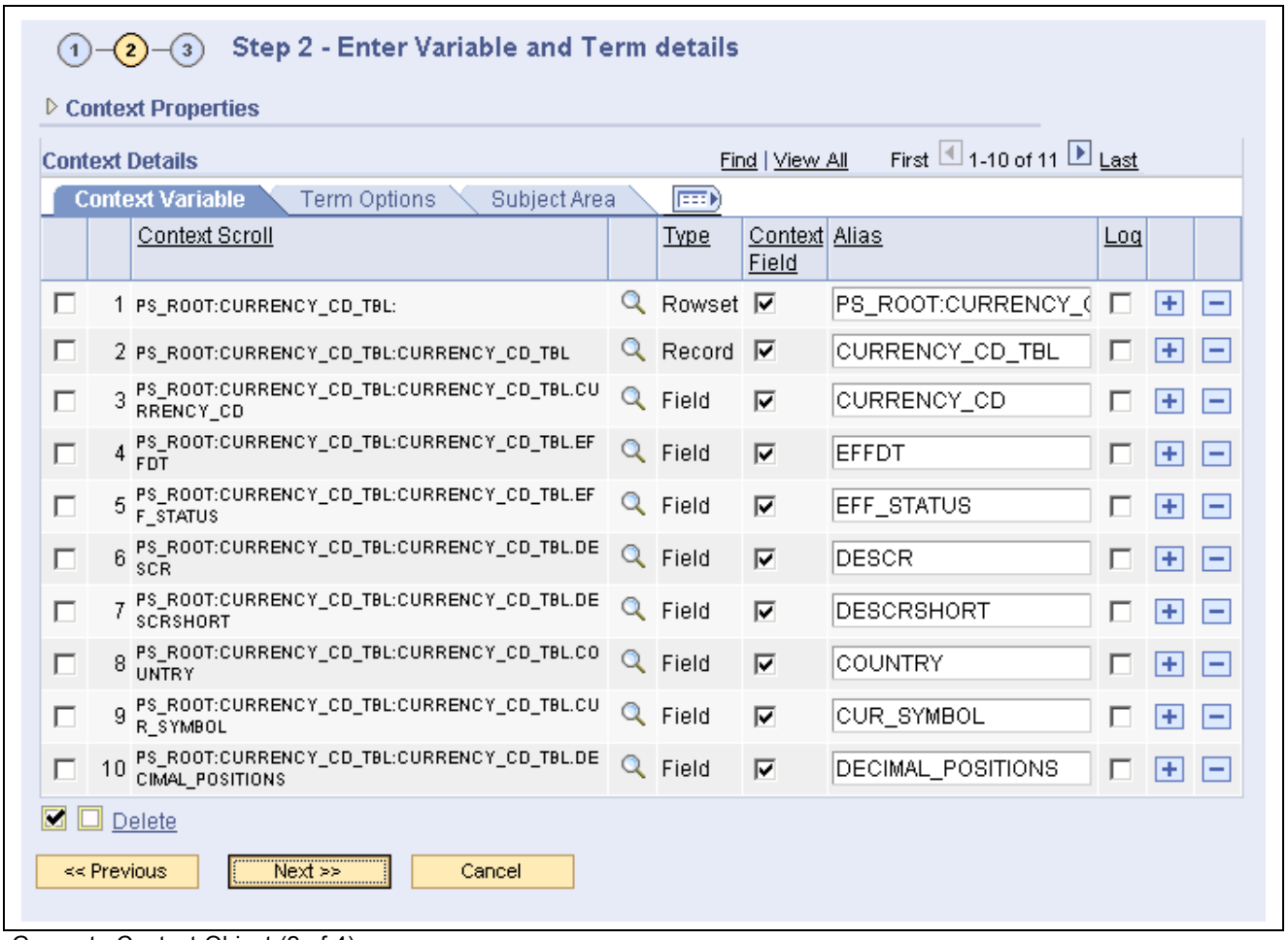
Access the Generate Context Objects page.



Generate Context Object (1 of 4)

Component Interface Name Used to extract the contents of a PeopleSoft component for use in generating the context. A component interface is required to create an online context.

Context Name	Required, unique and descriptive name of a context.
Description	Required, appropriate description about the intent of this context.
Library Subject Name	Default subject area for terms created by this process. The prompt for this field shows the subject area selection list.
Primary Context	Select this checkbox to denote the context to be generated as the primary context for this online transaction.

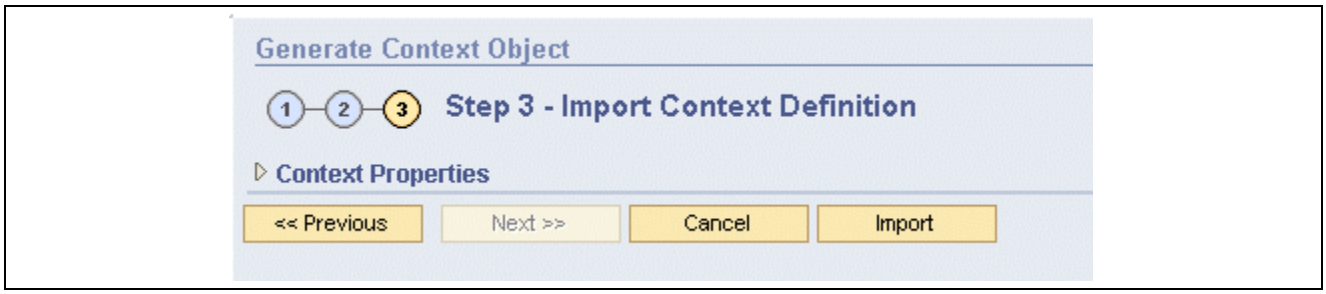


Generate Context Object (2 of 4)

Review and configure the context variables and terms that were automatically generated based upon the component interface you selected. Select the Log checkbox to denote that the field should be logged when a context is persisted at runtime (this is a key to the transaction, and the process automatically sets this field).

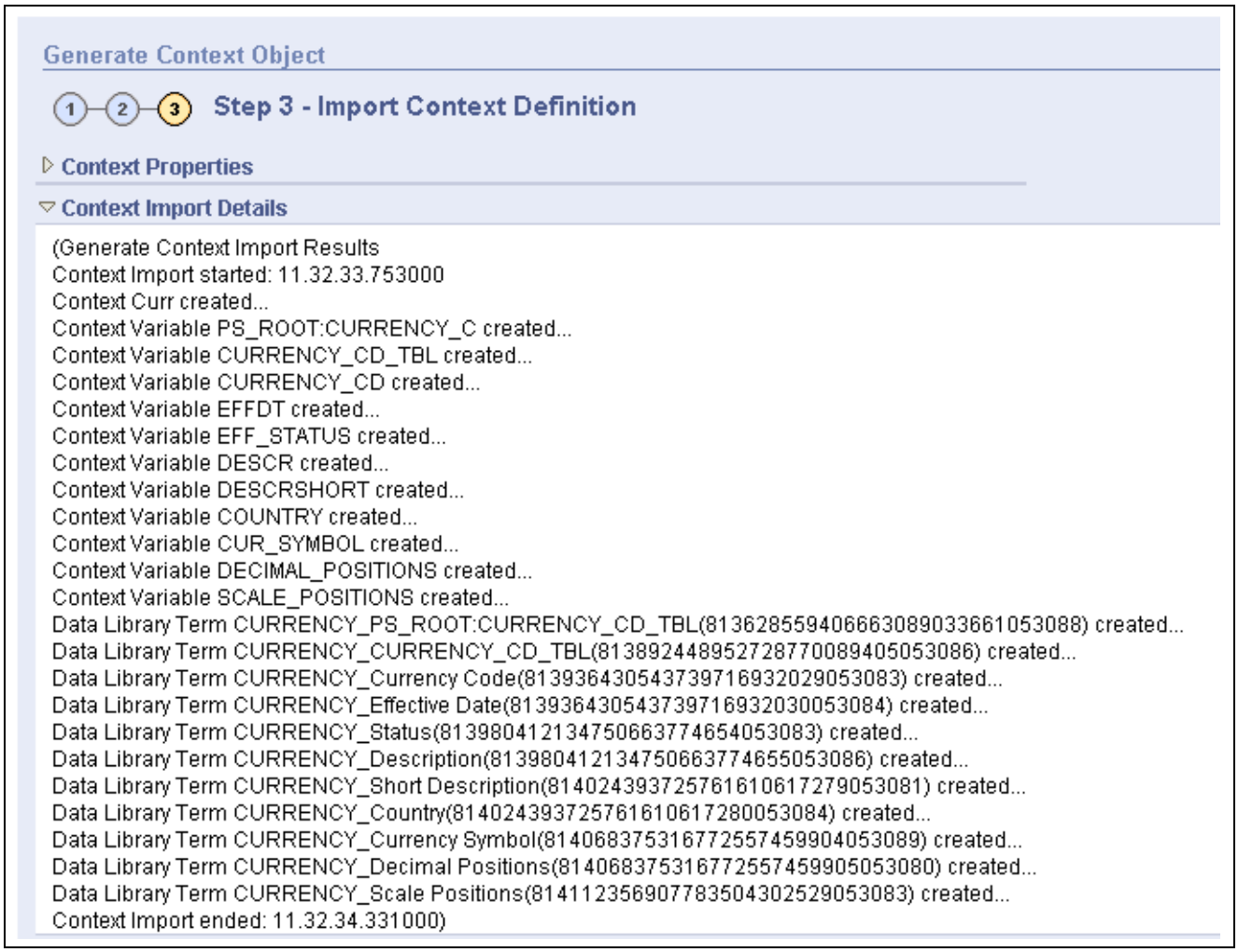
Select the Term Options tab to create a new term for each context variable that is created. Also, you can add a contextual implementation to an existing term for this context variable

Select the Subject Area tab if you want to override the default subject area chosen on the first page.



Import Context Definition page (3 of 4)

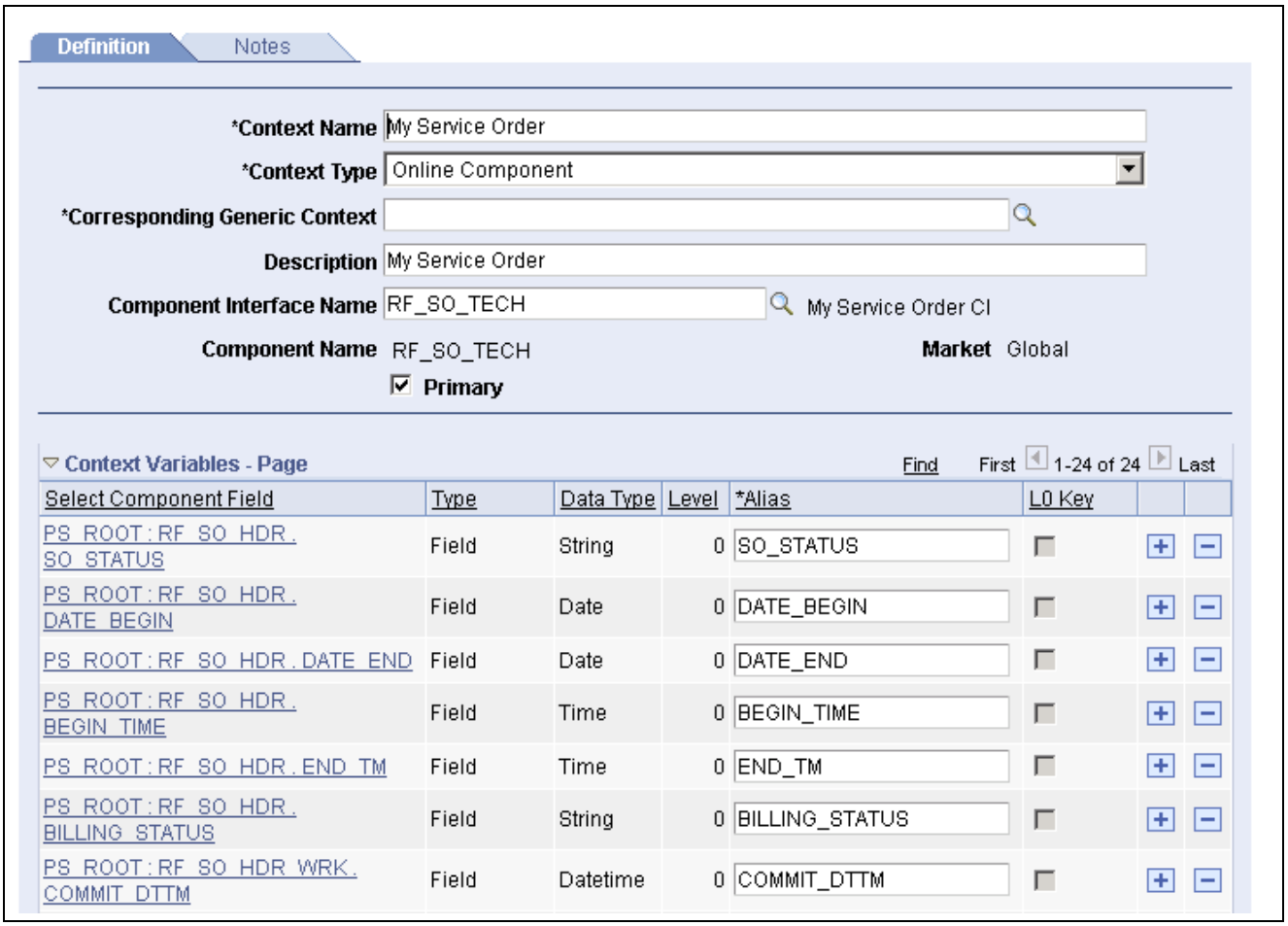
Select Import to create the context and terms. The Generate Context Object screen appears listing the import details.



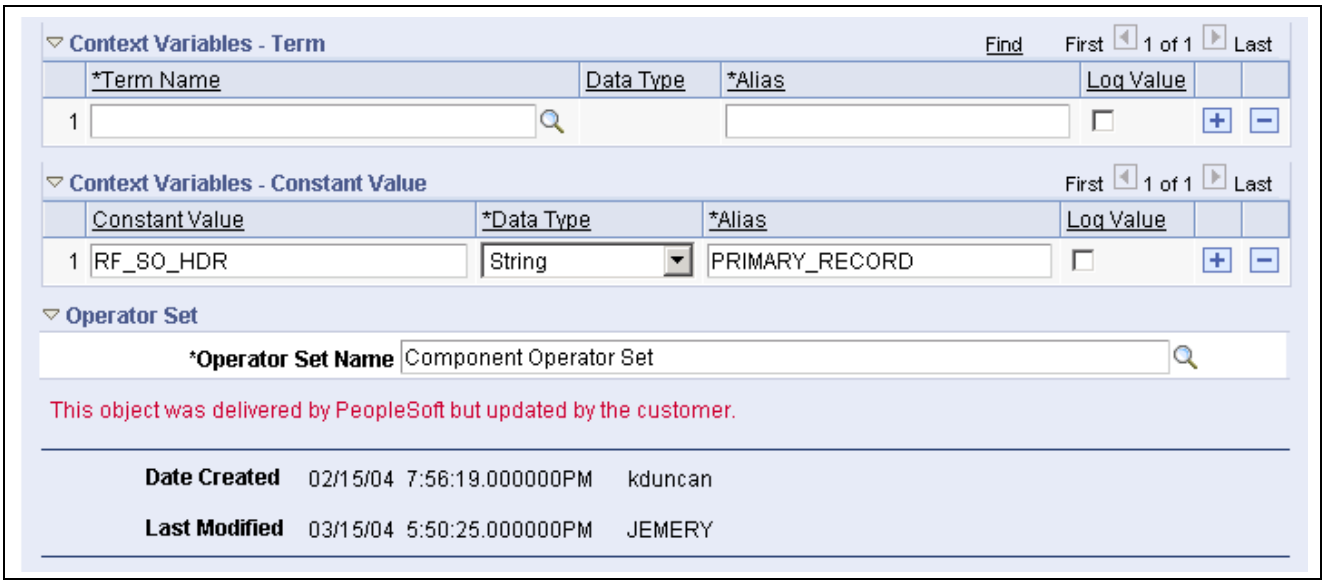
Generate Context Object (4 of 4)

Managing Context Objects

Access the Manage Context Object page.



Manage Context Object (1 of 2)



Manage Context Object (2 of 2)

Corresponding Generic Context This field is used to programmatically create a standalone context from its corresponding online context. Applications directly embedding data library

terms need this information to supply context information to the data library runtime engine.

Context Variables Page This section is used to define the page context variables.

- Select Component Field. Click to select a component field from the displayed component buffer hierarchy.
- Type. This column specifies the object type of the selected component field. Valid values are *Field, Record, Rowset*.
- Data Type. Specifies the PeopleSoft data type of the selected component field.
- Level. Specifies the scroll level of the selected component field.
- Alias. Unique identifier for a context variable.
- L0 Key. Specifies whether a component field is a key field for this component.

Context Variables Term This section defines the context variables of type term.

- Term name. Select a term name from the prompt. The resolved value of this term is used as the value of this context variable.
- Data Type. Specifies the data type of the term selected.

Context Variables Constant This section defines the context variables of type constant.

- Constant Value.
  - Specify a constant value to be used as the runtime value for this context variable.
  - Leave blank and a value has to be supplied at runtime.
- Data Type. Specify the data type of this context variable.

Note. A constant value specified for a context variable cannot be overridden at run time.

Operator Set Optionally, specify an operator set to be used for this context. This operator set is used to present the list of valid operators for selected terms in policy conditions. A default operator set is automatically filled in.

CHAPTER 13

Setting Up the Action Framework

This chapter provides an overview of the action framework and discusses how to:

- Register action types.
- Register action type bundles.

Understanding the Action Framework

The action framework allows users to specify the actions to be executed when policy conditions evaluate to true within a trigger point.

Architecture of the Action Framework

The architecture of the action framework comprises:

- An action type registration component for creating and maintaining action types by programmers and IT staff. An action type is metadata pertaining to a particular class of actions that might be performed at runtime.
- An action type bundle registration component for creating and maintaining action type bundles, or classes of actions that can be combined. An action type can be in only one action type bundle.

  Note. In this release, if other display action types are created by a PeopleSoft product line, they must be combinable with the delivered display alert action type, and must be registered in the display action type bundle.

- A design-time environment facilitating the embedding and configuring of consequent actions within policies.
- A runtime environment that allows the decision engine to invoke particular actions as needed.
- A generic display alert action type, which is available to all product lines using the framework. It can be used to display important information about a customer or a suggestion for a course of action. For example, if a high-value customer calls on the phone, a popup window appears with an alert and a suggestion that the phone connection be routed to a manager.

Understanding Action Types

An action type refers to a category of actions that can be associated with a policy. For use in the framework, an action type must be registered with the following information:

- Location of the code that handles the design-time and runtime aspects of the action type.

- Configuration requirements.

  When adding actions of specific action type to a policy, policy-specific configuration requirements must be set before the policy can be enacted.

- Whether actions of an action type terminate the analytic processing.

  For example, if the action is a transfer from an application page to another page, this terminates the framework processing that was triggered by a trigger point associated with the original page.

Note. A terminal action type is not combinable.

- Whether the action can be combined with other actions at runtime. Referred to as a *combinable* action.

  For example, the display alert action type can be combinable; therefore, when two display alert actions are specified, they are combined , and a single popup window appears that includes both alerts in the popup.

- Whether the action is part of an action bundle.

  If actions of *different* action types are combinable with each other, the action types must be included in an action type bundle. Consequently, if combinable action types are not combinable with certain other action types, then the action type does not need to be part of an action type bundle.

- Triggering environment—where the action can be deployed.

  For example, the display alert action executes only from an application page, not from an Application Engine program.

- The trigger types and trigger points that include the action type as a valid action type.

  For example, you may want an action to be valid for a ComponentPostBuild trigger type, specifically the *When a Customer is Presented* trigger point, associated to this trigger type.

Understanding How Actions Execute

After the framework evaluates conditions for all a trigger point's policies, the actions associated with the policies having true conditions are forwarded to the action framework. The action framework is responsible for firing the actions.

Only one terminal action can be fired for any trigger point. Because a trigger point may be associated with one or more policies, and each policy may include more than one action, it is possible that more than one terminal action is associated with a trigger point. In this case, the first terminal action in the first policy that evaluates to true is executed. If present, a terminal action fires after executing all the non-terminal actions.

Individual action types determine how actions can be combined. Some action types, such as the display alert, combine all their actions from the same trigger point. Therefore at runtime, all the display items appear in the same popup window.

Note. In order that display actions execute, popup blockers must be turned off.

An action that is not combinable will not affect the execution of another action.

Registering Action Types and Action Type Bundles

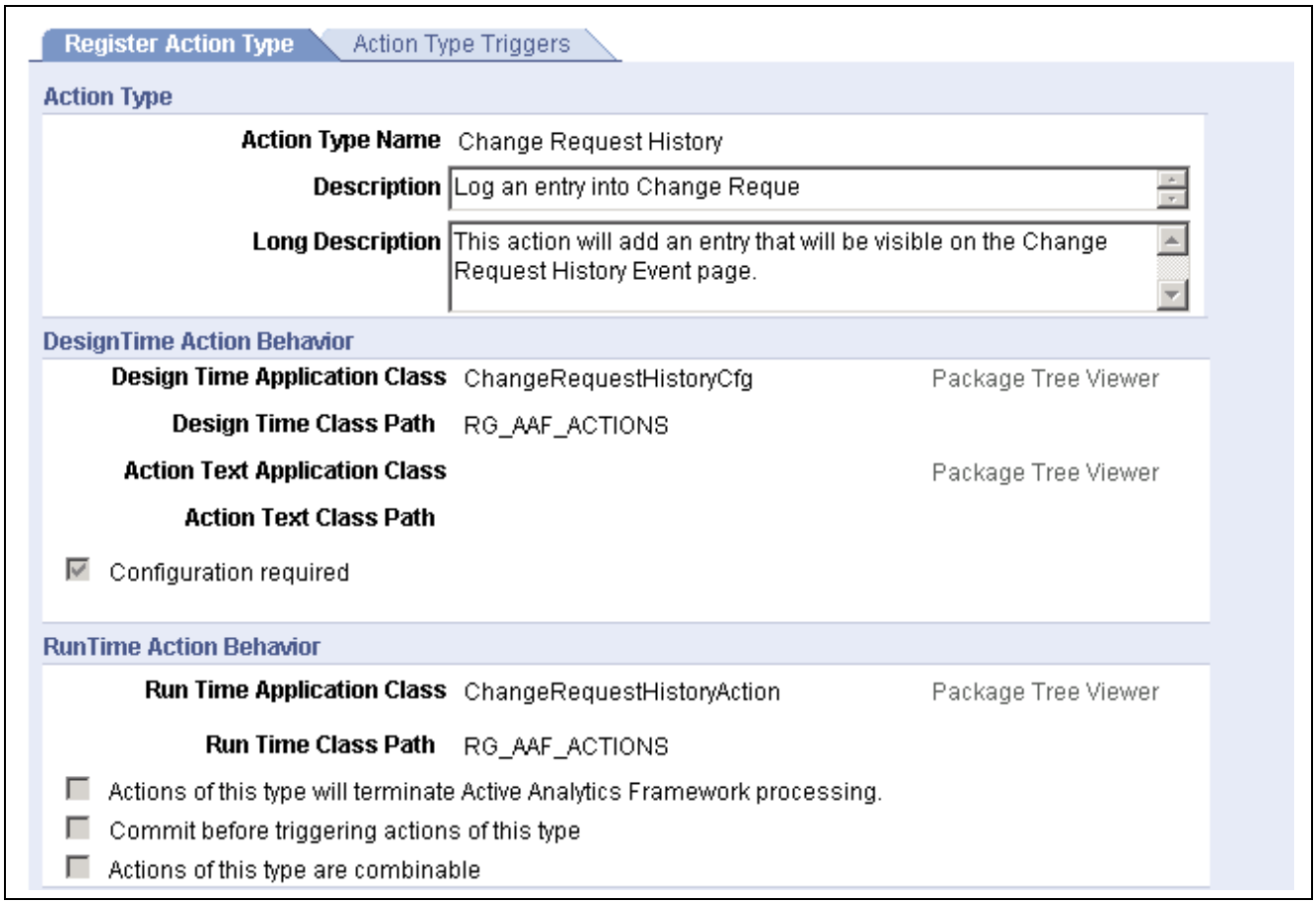
This section describes the components used to register action types and action bundles.

Pages Used in This Chapter

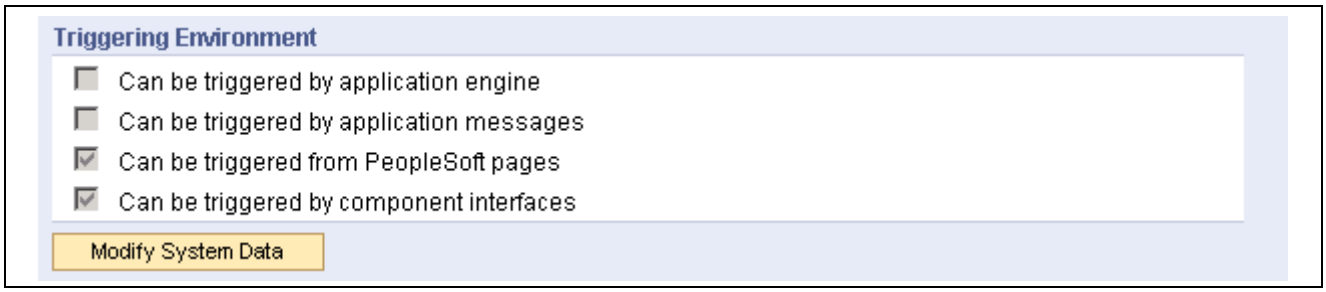
Page Name	Object Name	Navigation	Usage
Register Action Type	EOCF_ACTN_TYPE_REG	Enterprise Components, Active Analytics Framework, Action Framework, Register Action Type	Registers an action type.
Action Type Triggers	EOCF_ACT_TYP_EVNTS	Enterprise Components, Active Analytics Framework, Action Framework, Register Action Type, Action Type Triggers	Specifies action type triggers.
Register Action Type Bundle	EOCF_ACTION_BUNDLE	Enterprise Components, Active Analytics Framework, Action Framework, Register Action Type Bundle	Defines bundles of combinable action types.

Registering Action Types

This section describes how to register action types. Access the Register Action Type page.



Register Action Type page (1 of 2)



Register Action Type page (2 of 2)

Action Type Name Name of a class of similar actions.

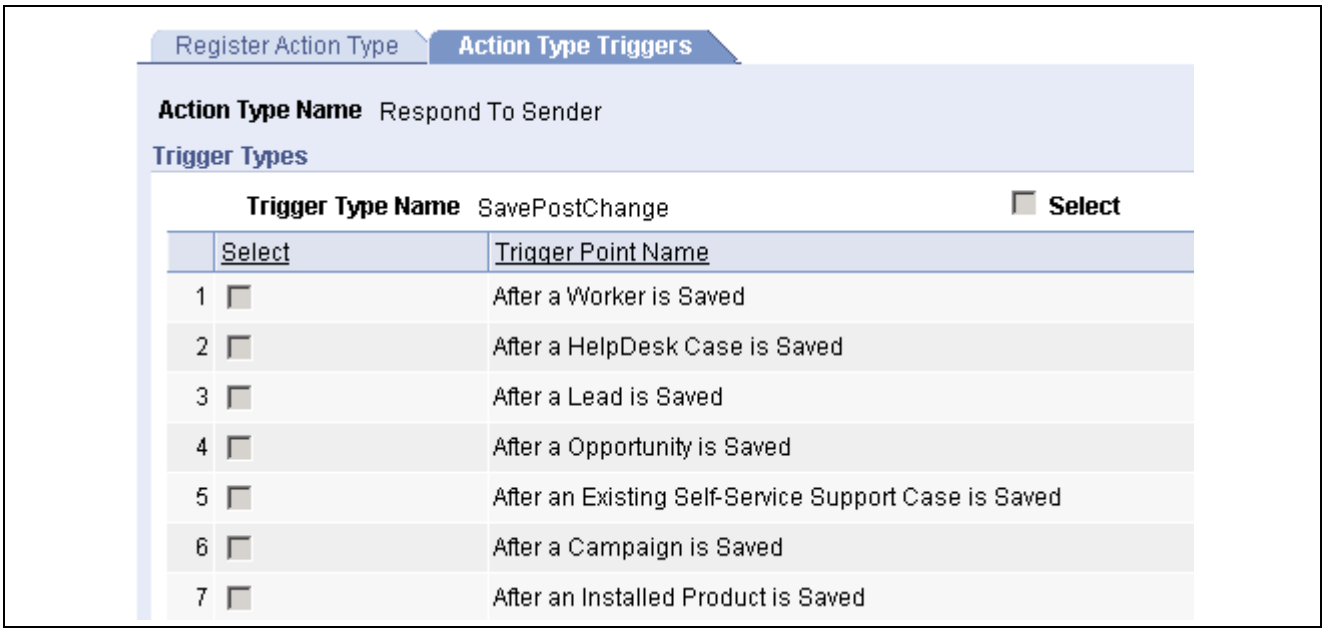
DesignTime Action Behavior Specify the design time details of the action type. When adding actions of this action type in a policy, these design time specifications are used to present the action type configuration page and store the configuration.

RunTime Action Behavior Specify the runtime details of the action type. The application class details specified here are executed at runtime to trigger actions of this type.

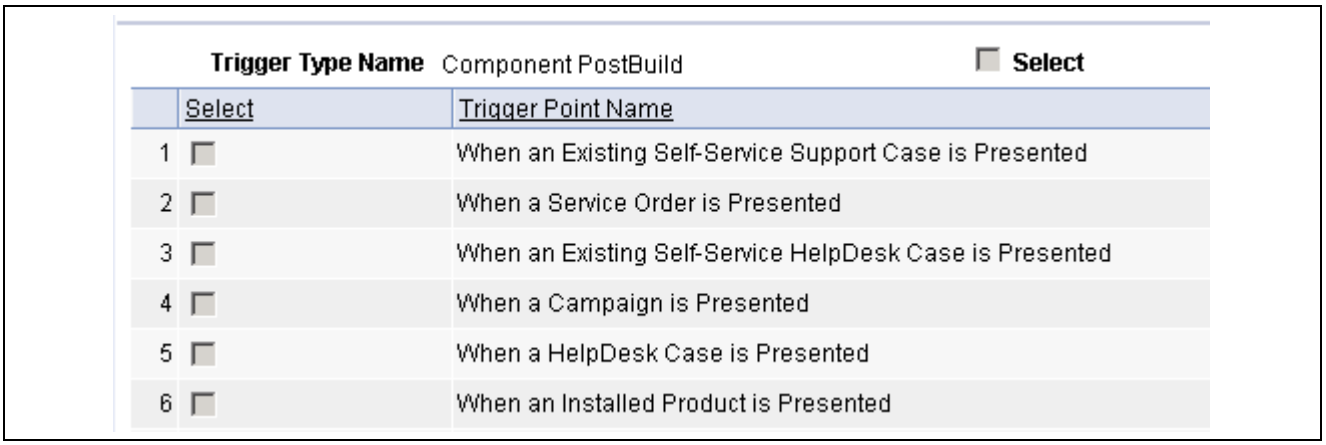
Triggering Environment Specify the triggering environments to be supported for this action type.

Registering Action Type Triggers

Access the Action Type Trigger page.



Action Type Triggers page (1 of 2)



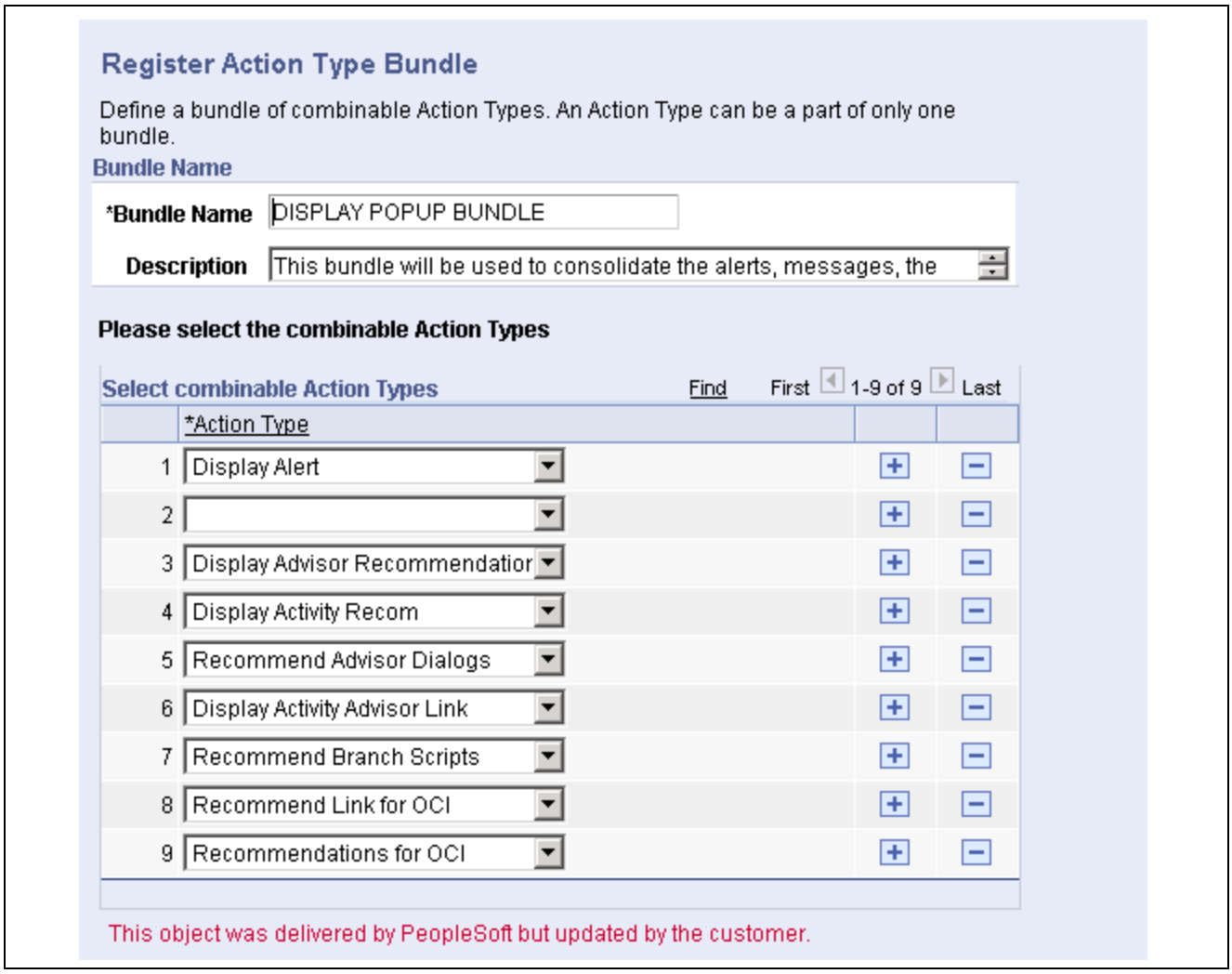
Action Type Triggers (2 of 2)

Select the appropriate checkboxes to associate this action type with listed trigger types and trigger points.

- Selecting a trigger type makes this action type available for the selected trigger type.
- One or more trigger points may be selected only if the corresponding trigger type is selected.

Registering Action Type Bundles

This section describes how to register action type bundles. Access the Register Action Type Bundle page.



Register Action Type Bundle page

Enter a name and description for this action type bundle. Select the action types that can be combined from the dropdown list in each row.

CHAPTER 14

Administering the Framework

This chapter discusses how to:

- Set logging options for the data library.
- Set installation options.
- Register trigger types and trigger points.
- Define subject areas.
- Register operators and operator sets.
- Register resolution methods.
- Register business domains and categories.
- Register action objectives.

Using the Setup Components

This section lists the components used for administering the PeopleSoft Active Analytics Framework.

Pages Used in This Chapter

Page Name	Object Name	Navigation	Usage
Data Library Logging	EOCF_DL_LOG_SET	Enterprise Components, Active Analytics Framework, Setup, Data Library Logging	Used to set Data Library logging options.
Install Options	EOCF_INSTALL	Enterprise Components, Active Analytics Framework, Setup, Install Options	Used to set Active Analytics Framework installation options.
Register Trigger Type	EOCF_EVTYP_DEFN	Enterprise Components, Active Analytics Framework, Setup, Register Trigger Type	Used to define trigger types.
Register Trigger Point	EOCF_EVENT_DEFN	Enterprise Components, Active Analytics Framework, Setup, Register Trigger Point	Used to define trigger points.

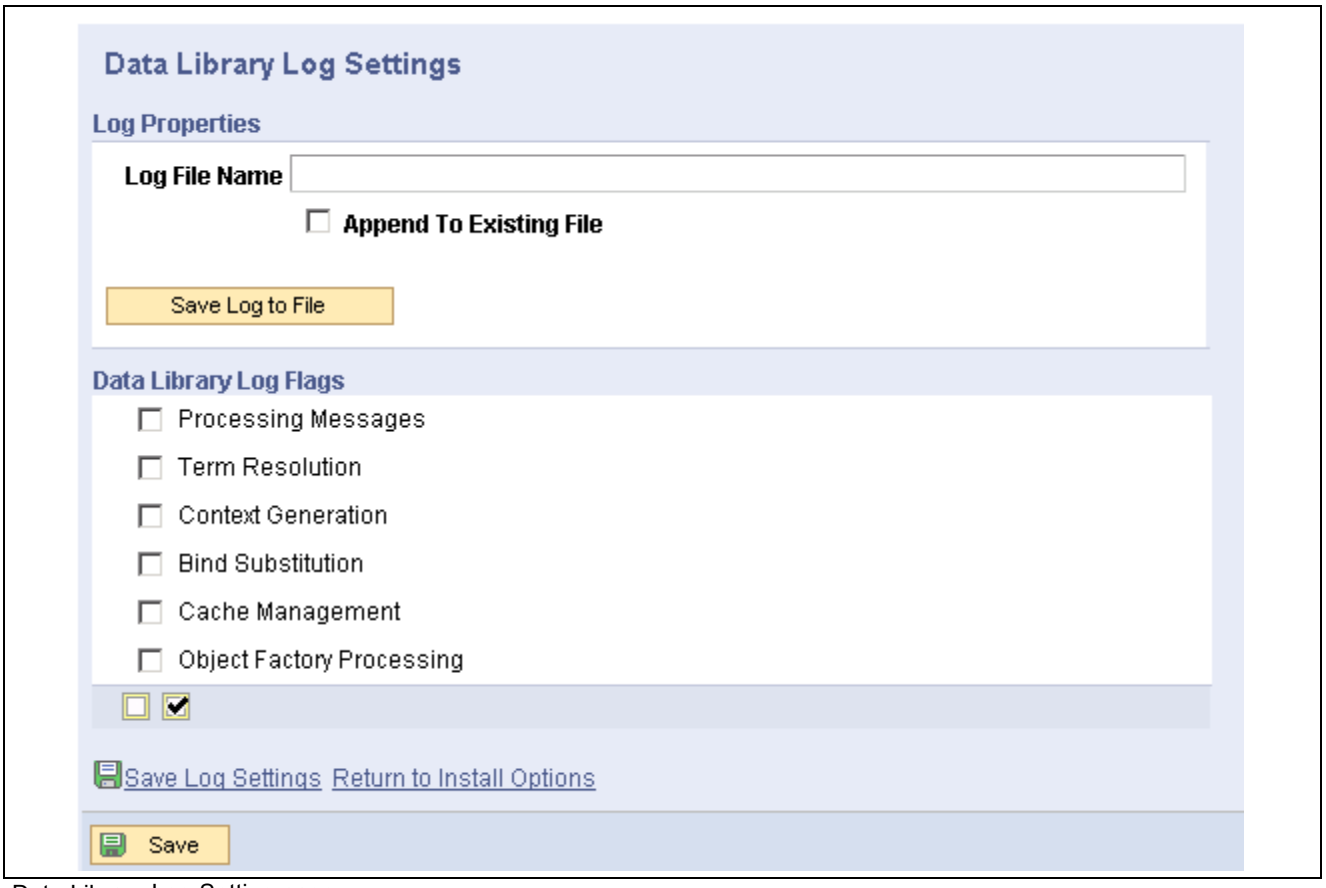
Page Name	Object Name	Navigation	Usage
Define Subject Area	EOCF_SUBJ_HIER	Enterprise Components, Active Analytics Framework, Setup, Define Subject Area	Used to define subject areas.
Register Operators	EOCF_OPERATOR_DEFN	Enterprise Components, Active Analytics Framework, Setup, Register Operators	Used to define operators.
Register Operator Sets	EOCF_OPERSET_DEFN	Enterprise Components, Active Analytics Framework, Setup, Register Operator Sets	Used to define operator sets.
Register Resolution Method	EOCF_DEF_RESLMTHD	Enterprise Components, Active Analytics Framework, Setup, Register Resolution Method	Used to define the resolution method.
Register Business Domain	EOCF_BUS_DOMAIN	Enterprise Components, Active Analytics Framework, Setup, Register Business Domain	Used to define a business domain, which is used to functionally classify trigger points.
Register Category	EOCF_IACATEGORY	Enterprise Components, Active Analytics Framework, Setup, Register Category	Used to functionally classify policies.
Register Action Objective	EOCF_ACTION_OBJ	Enterprise Components, Active Analytics Framework, Action Framework, Register Action Objective	Used to functionally categorize actions defined in policies.

Setting Log and Installation Options

This section describes the logging and installation options

Setting Data Library Log Options

Access the Data Library Log Settings page.

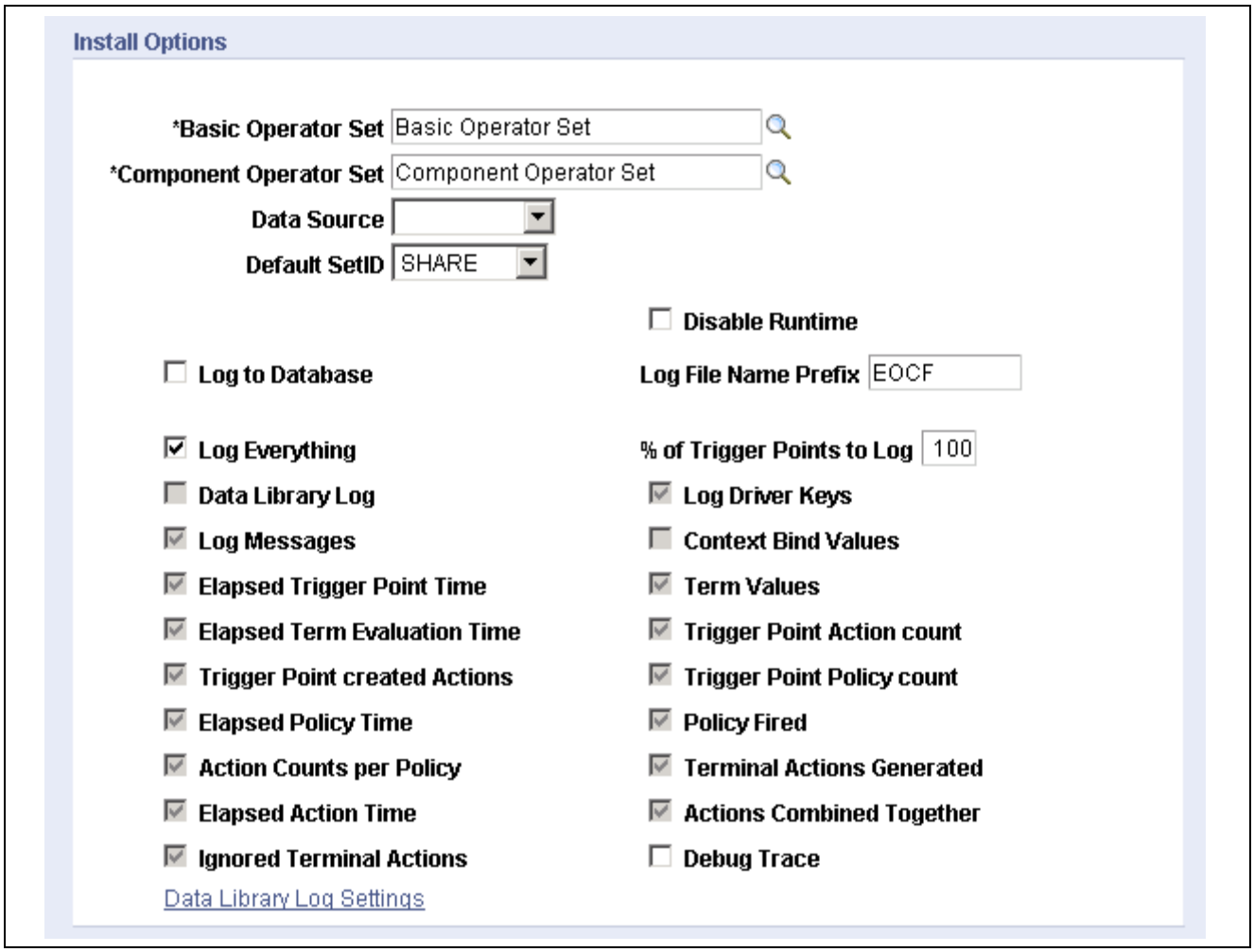


Data Library Log Setting page

Log File Name	Enter a name for the log file and select Append to an Existing File, if desired.
Term Resolution	Log technical details of resolution of individual terms.
Context Generation	Log technical details of the generation of contexts.
Bind Substitution	Log technical details of implementation binds substituted with values from the runtime context.
Cache Management	Log technical details of usage of cache.
Object Factory Processing	Log technical details of the loading of framework object definitions.

Installation Options

Access the Install Option page.



Install Options page

Data Source The product line for this database, for example CRM or HMS.

Default SetID The setID value to use for a component that does not use set control

% of Trigger Points to Log When logging for performance monitoring, the impact of logging can be alleviated by only logging a percent of trigger points executed. Specify a percent value between 1 and 100.

Disable Runtime This option disables all trigger points from running.

Log to Database Determines if the log will be written to the database, or to a text file.

Log File Name Prefix If a file is logged instead of the database, this prefix is prepended to each user's log file.

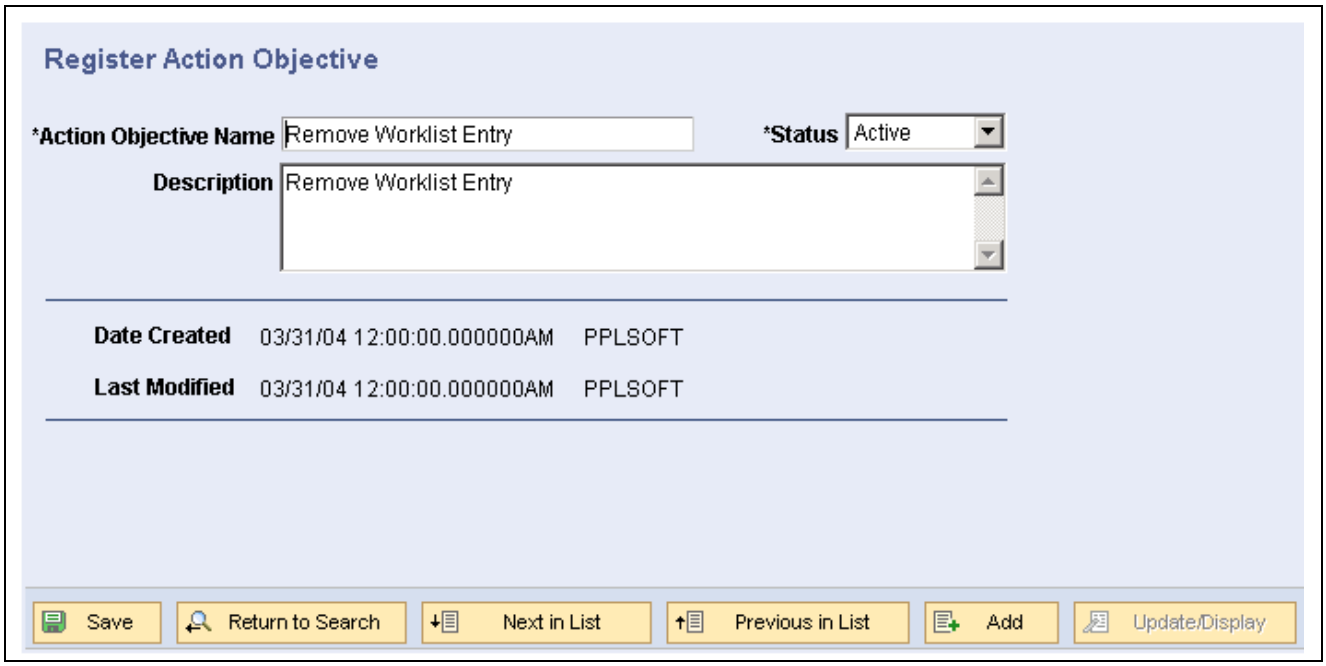
Note. The decision engine log file is created in the default files directory for this application server instance. Typically, the application server uses the directory %PS_HOME%\appserv\<servername>\files. The application server creates this directory as needed; therefore, if no log files have been written, it may not yet exist.

Log Everything If checked, causes all normal logging options are activated.

Data Library Logging	Includes the data library log entries in the decision engine log.
Log Driver Keys	Log the primary key values driving each trigger point's context (transaction).
Log Messages	Log messages that policies or actions may generate.
Context Bind Values	Log values retrieved directly from the operant context.
Elapsed Trigger Point Time	Log the time required to complete a trigger point.
Elapsed Term Evaluation Time	Log the time required to evaluate a term.
Term Values	Log the values of evaluated data library terms.
Trigger Point Action Count	Log the number of actions created by the execution of a trigger point.
Trigger Point Policy Count	Log the number of policies evaluated for a trigger point.
Trigger Point created Actions	Log the actions created during the evaluation of policy conditions for a trigger point.
Elapsed Policy Time	Log the time required to evaluate the condition portion of a policy.
Policy Fired	Log whether a policy condition was true or false.
Action Counts per Policy	Log the number of actions created by a policy.
Terminal Actions Generated	Log terminal actions created by a policy.
Elapsed Action Time	Log the time it takes an action to complete.
Actions Combined Together	Log actions that were combined during execution.
Ignored Terminal Actions	Log any terminal actions that could not be invoked.
Debug Trace	Log debugging information, not normally used.

Registering Action Objective

Access the Register Action Objective page.



Register Action Objective page

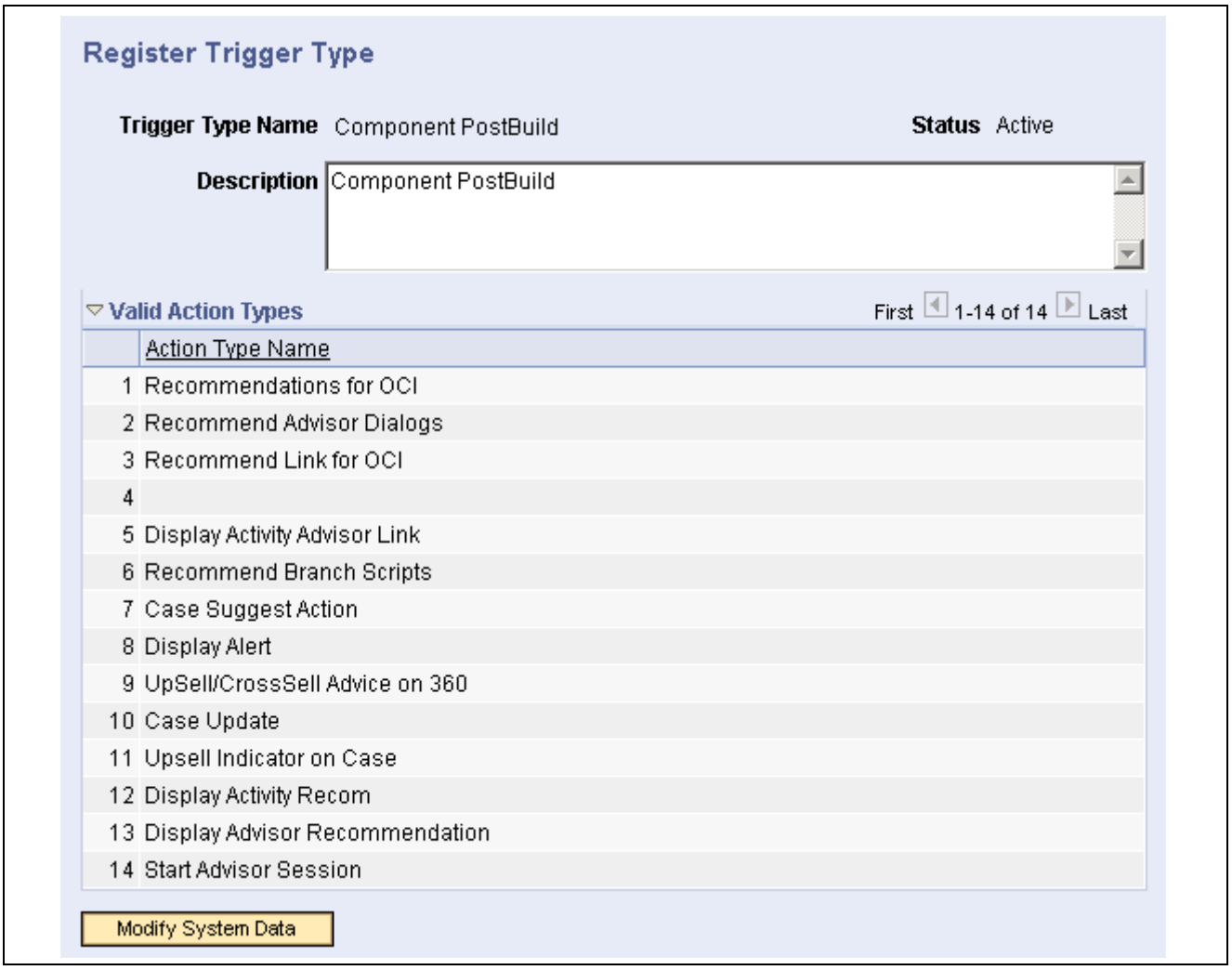
Enter the Action Objective Name and a description.

Registering Trigger Types and Trigger Points

This section describes the components used to register trigger types and trigger points.

Registering Trigger Types

Access the Register Trigger Type page.

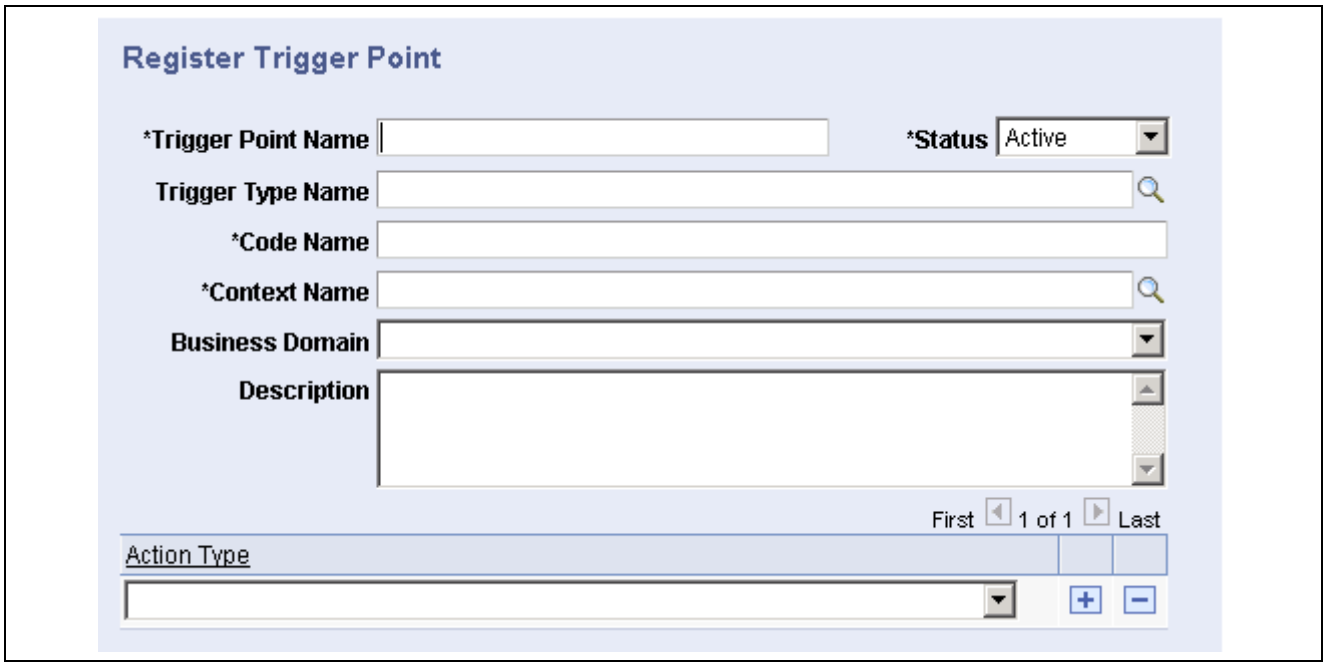


Register Trigger Type page

Enter details about the trigger type you are defining and select the valid actions.

Registering Trigger Points

Access the Register Trigger Point page.



Register Trigger Point page

Trigger Point Name Unique and descriptive name of the trigger point.

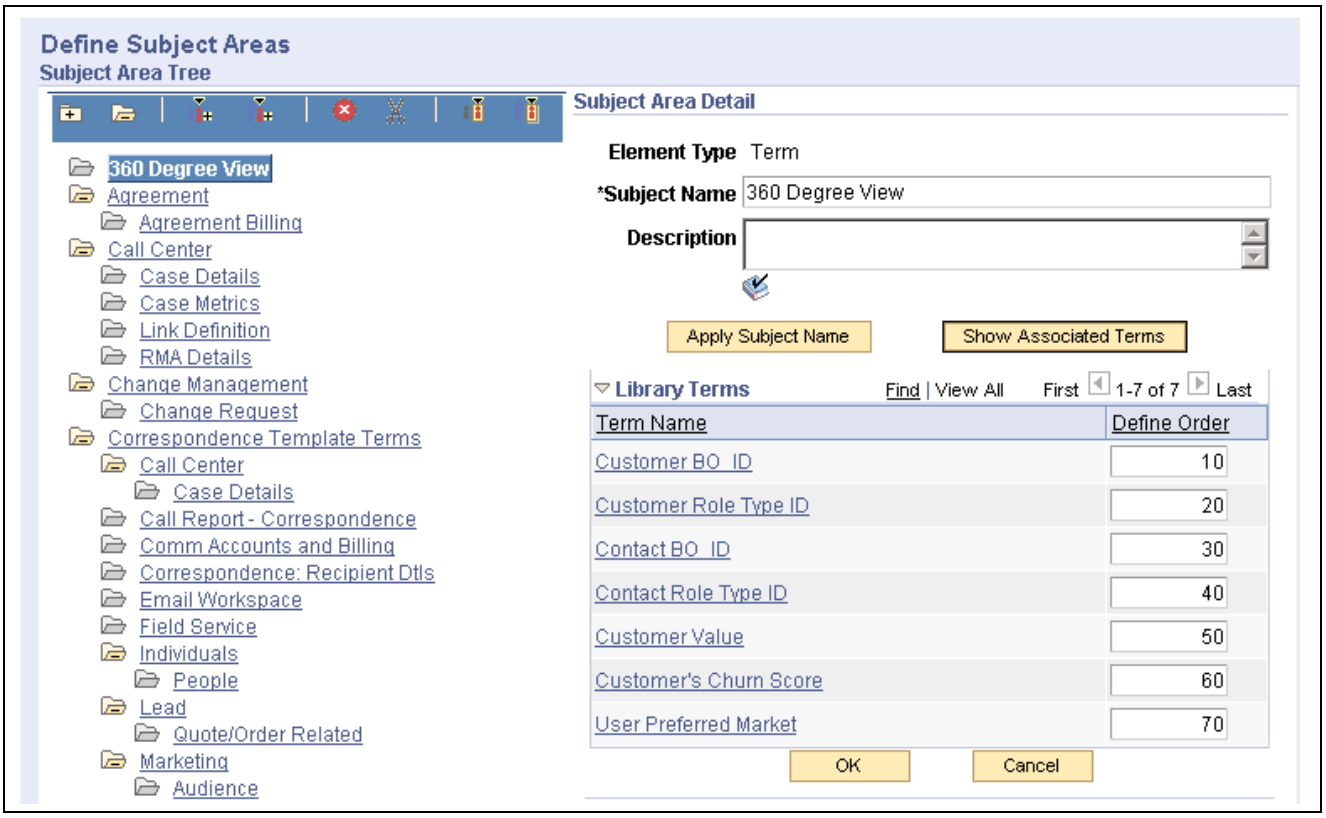
Trigger Type Used by IT developers to figure out which PeopleSoft event in the component needs to be enabled with this trigger point. Also restricts the list of valid action types that may be used when creating policies for this trigger point.

Code Name Used to reference the trigger point when programmatically enabling this trigger point for a PeopleSoft component. Changes made to the code name, after enabling this trigger point for a PeopleSoft component, may disrupt the execution of policies for this trigger point at runtime.

Context Name Defines the context for the trigger point.

Defining Subject Areas

Access the Define Subject Area page.



Define Subject Areas page

Subject Area Detail

Enter a Subject Name and Description and click Apply Subject Name. The subject area name may not be unique.

Click Show Associated Terms to display the list of terms associated to this subject area node. Order the terms appearing for a subject area node by specifying display order numbers. The order depicts how the terms are displayed for a subject area node in the term list presented in the condition builder.

Subject Area Tree

The subject area tree displayed on the left-hand side has the following icons:

- *Expand All*. Expands all subject area nodes.
- *Collapse All*. Collapses all subject area nodes.
- *Add Sibling*. Adds a new sibling node under the subject area node currently selected. Enter a subject area name and description on the right hand pane and click Apply Subject Name to apply changes.
- *Add Child*. Adds a new child node for the subject area node currently selected. Enter a subject area name and description on the right-hand side and click Apply Subject Name to apply changes.
- *Delete Node*. Deletes the selected node.
- *Cut*. Cuts the currently selected node.
- *Paste as Sibling*. Pastes the node previously cut as a sibling of the currently selected node.

- *Paste as Child*. Pastes the node previously cut as a child of the currently selected node.

Registering Operators and Operator Sets

This section describes the components used to register operators and operator sets.

Operators within PeopleSoft Active Analytics Framework are defined in text expressions that are used to evaluate a condition at runtime. The operand values in the text expressions must be substituted with an operator definition for integration with policies. The substitution of operand values is ensured by the correct placement of meta-text corresponding to the desired operand or source term.

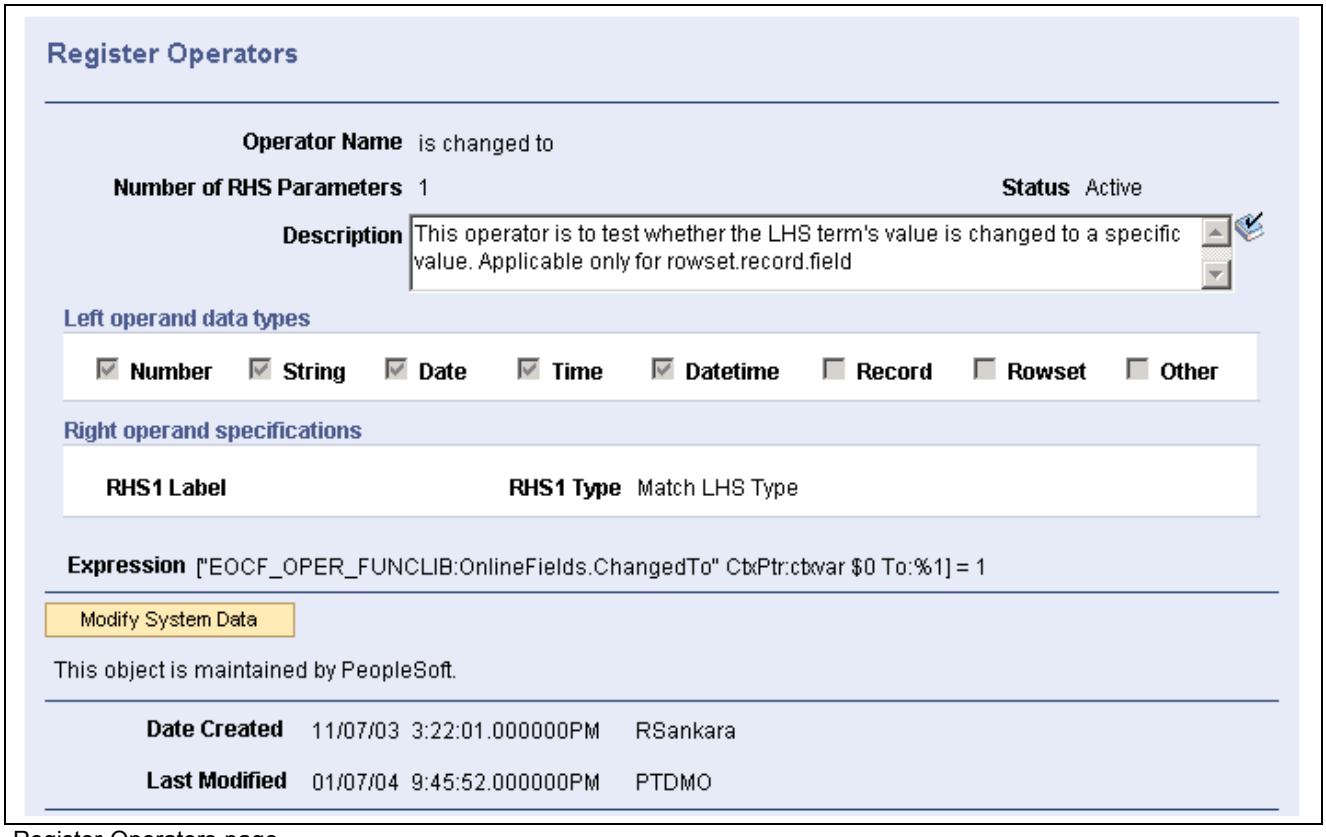
To refer to the left-hand-side of a condition, use %0 as the value. To refer to right-hand side operands 1–4 , use the text values %1, %2, %3 , and %4, respectively. For example, the following operator expression determines if the numeric term's value is less than a right-hand side value:

%0 < %1

Note. If you edit operators, policies that reference them must be recompiled. Do this by editing and saving the policy.

Registering Operators

Access the Register Operators page.



Register Operators page

Register custom operators by defining the left-hand side and right-hand side operand specifications and entering expression text.

Operator Name Enter the name of the operator you are defining. Operator names may not be unique as long as the expression texts are different.

Number of RHS Parameters Specify the number of right-hand side parameters that are required to evaluate the boolean value for this operator. The condition builder uses this to display one or more right-hand side fields to enter values.

Note. In this release, the condition builder interface supports a maximum of 2 right-hand side parameters for an operator.

Left operand data types Specify the supported data types for the left-hand side operand of this operator. This data is used to constrain the list of operators in the condition builder.

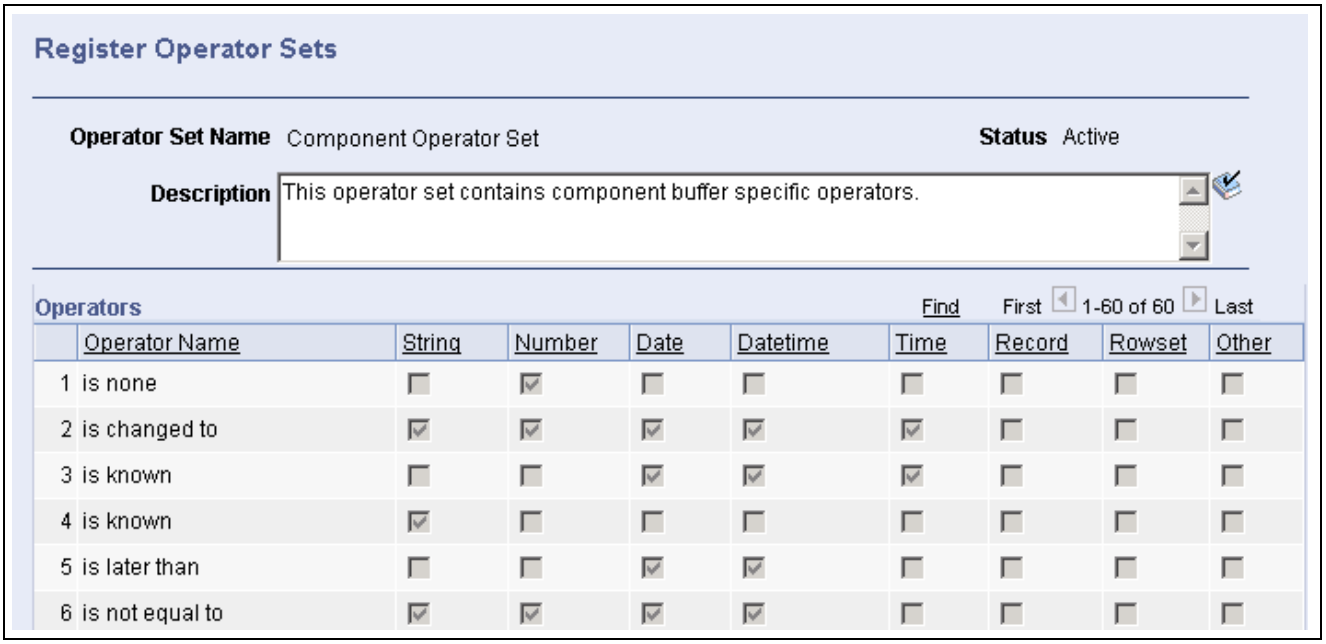
Right operand specifications Specify the type for each of the right-hand side parameters.

- *Match LHS Type*. Supports the left-hand operand data types.
- *Fixed Type*. Specify a fixed data type for this parameter.
- *Multi-Select*. Enables selection of multiple values for this parameter at condition building time.

Expression Specify the technical expression defining how the operator works.

Registering Operator Sets

Access the Register Operator Set page.



Register Operator Sets page

Operator sets define the list of valid operators for a term selected on the left-hand side of a condition.

Enter a unique Operator Set Name and description, and add valid operators for this set.

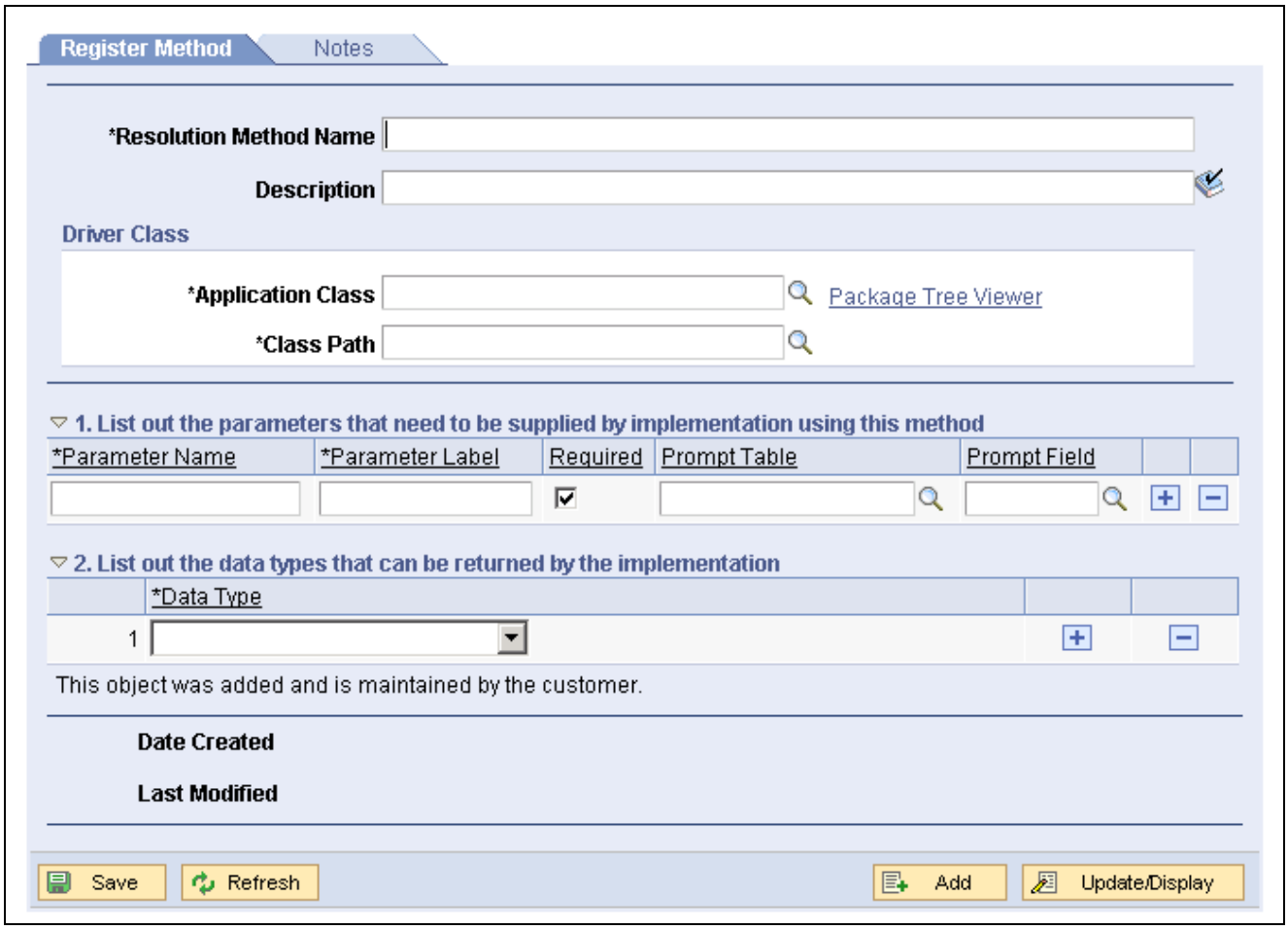
Registering Resolution Methods

This section describes how to register a resolution method.

Resolution methods are used to build implementations. PeopleSoft supports the following resolution methods:

- Application class.
- PS Query.
- SQL Object.
- Record.Field.

Access the Register Resolution Method page.



Register Method page

Resolution Method Name Unique name of the resolution method.

Driver Class The application class that encapsulates the data retrieval mechanism for this resolution method.

Parameters... Lists the parameters that an implementation using this method needs to supply.

Datatypes... Lists the valid data types that can be returned by an implementation using this resolution method.

Registering Business Domain and Category

This section describes how to register a business domain and how to register a category.

Register Business Domain

Access the Register Business Domain page.

Enter the Business Domain name and Description and select the appropriate status.

Register Category

Access the Register Category page

Enter the Category Name and Description and select the appropriate status.

CHAPTER 15

Considerations for Enabling the Framework

This chapter discusses considerations when:

- Creating contexts.
- Creating custom operator expressions.
- Creating a trigger point.
- Enabling a trigger point in a PeopleCode event.
- Enabling a display action.
- Creating a new action type.
- Creating an application class implementation.
- Creating a PS query implementation.

Considerations When Creating Contexts

To create an online context, you must be able to specify a component interface. You may encounter situations where sets of components share the same base records and display the same information onscreen, however for different purposes. In this case, decide if you need to create two contexts or if you can use one.

Whether to Create One or Two Contexts

Determine if the two contexts' common terms should be registered as the same term objects, or as different term objects. When common terms have the same meaning, it is most economical to re-use the common term definitions between the two contexts.

Create two contexts if:

- The components have actionable fields that they do not share in common.
- One or the other of the components need significant customization.

Considerations when Exposing Component Buffer Fields as Terms

Carefully consider which component buffer fields to be made available as terms.

- Keep the big picture in mind—by selecting terms that may be reused at some point.

  However, you would not want to expose every component field as a term. Some of the transactional components that allow users to perform complex tasks may have component fields that are used as work fields to implement the component, but which do not represent functional data. Exposing these fields as terms could overpopulate the data library.

- PeopleSoft recommends that you expose terms of those data elements on the component that are displayed to the end user and the elements that comprise the component's search keys.
- Work-fields may contain valuable data and may be exposed as terms, but if these fields have not yet been populated by the time that the values from the corresponding terms are requested, unpredictable results may occur. If you know for certain that the work-fields will contain data at the appropriate times, such as in the example of a work-field whose data is computed as part of component pre-build, then there is no harm in exposing the data element as a term. Although there is a Manage Context component for adding terms to a context, configuring terms is easier in the Generate Context component.

Considerations When Naming Terms

Note the following when you name terms:

- The framework prefixes each alias with the name of the context to reduce the creation of duplicate names.
- In record and rowset objects, the term name defaults to the name of the page buffer scroll, which will not be meaningful for the end user. PeopleSoft recommends this name be changed to something that makes sense for the user.
- Generally, rowset objects' names should be plural and record object term names should be singular. For example, if a component's scroll-level 1 contained purchase order line items, the rowset object is named Detail Items and the record object is named Line Item.
- If any terms are filed under alternate subject areas, use the Overridden Subject Area prompt to do so. PeopleSoft recommends that you carefully consider that all the names generated make sense before importing the context. Whenever context terms have the same functional meaning, have the context refer to pre-existing terms rather than creating new ones.

Considerations When Creating Custom Operator Expressions

Each operator definition contains an *operator expression*, a text template that defines the meaning of the operator and how operands are integrated to the expression. All operator expressions must be boolean expressions.

Comparisons supported in operator expressions are: =, <. >, <=. >=, <> (not equal). All comparisons are type-driven; for example, a number cannot be compared to a string.

If type conversion is necessary, the value to be converted may be preceded by the tokens string, number, date, datetime, or time, as needed.

Values are placed in expression text with two types of substitution tokens, location number and ? (question mark).

Location Number As An Operand

Location number determines the operand in the expression that is being referred to. For the left-hand side term, the location number is zero; for the right-hand side elements, the range is from one to four, depending on which right-hand side parameter is being referred to. Usually, the location number is preceded by a percent-sign. The percent-sign and location number combination form a complete operand token.

A complete operand token refers to the value of the configured operand. When creating a condition, an expression-text entry will be substituted for the operand-token. This expression-text entry corresponds to the definition of the operand value's source.

Sometimes a term ID value needs to be provided in the expression instead of a location number. In this rare case, instead of a percent sign, a dollar-sign is used to precede the location-number.

Parenthesis around sub-expressions are allowed, so long as each left parenthesis has a corresponding right parenthesis, and so long as the sub-expressions denoted within parentheses are valid. (While “(%0) > %1” and “(%0 > %1) are valid, “(%0 >) %1” is not.) Constants are available for the current date, date-time and the current time by means of the items “%today”, “%timestamp” and “%now”.

Use of Term Code

In addition to referring to the results of terms that have been configured in the Operator Definition page, references to terms may be embedded directly in the expression text by means of the term code. A reference to a term is made by placing the term code name (which may not contain spaces) within square brackets. The bind parameters needed for the term must be part of the context, therefore the operator should be part of a restricted operator set that’s available only where the operator will be known to be valid.

Parameters may also be manually specified for a term by following the term code name with a parameter specification, which consists of a name, a colon symbol, and the value to be bound to the parameter.

In the following example, a term is used as an input to another term in a condition to detect if a customer’s car is out of warranty:

[WarrantyExpirationDate VIN:[PrimaryVehicle DriversLicense:%0]] < %today

Generally this type of construction is awkward and difficult to maintain, but it is available if needed. A more maintenance-friendly facility that doesn’t require the configuration overhead of a term definition is a member-function call on an application class.

There are two rules that must be satisfied to use this facility:

- The class must not require any constructor arguments.
- The method to be invoked must accept an *array of any* as its only parameter.

To use this facility, set up an operator expression as if you are referring to a term; instead of using a term code name, use a quoted string value giving the fully qualified name of the class to be instantiated followed by a period and the name of the member function to be invoked. For example:

["MyAppPackage:MySubPackage:MyClass.MyMethod" MyParm1:%0 MyParm2:%2] = %1

Should an application class method need to return a boolean result, the standard practice is to return a number, and to compare the number to 0 for false values or to 1 for true values.

Note. Be aware that when you create operators for a specific purpose, they are neither valid nor relevant except in a specific situation. Therefore, the operator should belong to an operator set that corresponds to the context or term for which the custom operator makes sense.

Considerations When Creating a Trigger Point

Use caution when introducing new trigger points, especially in the FieldChange PeopleTools event. These trigger points, depending on where they are set, could result in significant overhead to the system.

In addition, introducing a new trigger point increases the likelihood of adding policies. These policies, depending on the number of terms and the mode of retrieval, could increase system overhead. This can result in a unfavorable user experience and throughput.

PeopleSoft recommends that FieldChange event trigger points be introduced only when it is critical. Consider deferring the execution of policies to a Save event.

Enabling a Trigger Point in a PeopleCode Event

The events in which a trigger point is to be executed must include PeopleCode. The following is a minimal example:

```
Declare Function GetRuleService PeopleCode FUNCLIB_EOCF.EOCF_DE_PUBLIC Field⇒
Formula;  GetRuleService().reset.SET_CONTROL = &mySetControlValue; GetRule⇒
Service.fireEvent(My_Trigger_Point_Code_Name);
```

Note. The use of the reset property on the first line is critical to the framework's correct functioning. If reset is not used, the state of the RuleService (the runtime API for triggering the decision engine) is unknown from previously executed PeopleCode.

Enabling the Display Action in a PeopleSoft Application Page

To enable the provided display alert action type:

- Add the PeopleTools sub-page EOCF_DISPLAY_SUBPG at Level 0 to each page within the component.
- Ensure that no other fields overlap the read-only display alert sub-page, otherwise the display alert action will not work correctly.
- Ensure that the roles having access to the transactional component that requests the framework to evaluate policies, have access to EOCF9002 permission list.
- Ensure that all popup blockers have been turned off.

Note. Any display action types created must be registered in the display action type bundle in addition to the delivered display alert action type.

Considerations When Creating a New Action Type

When creating a new action type, consider the following:

- Location of the code that handles the design-time and runtime aspects of the action type.
- Configuration requirements of the new action type before being enabled.
- Whether or not the action type is a terminal action.

  For example, if the action is to transfer to another page, this terminates framework processing. If there are several terminal actions associated with a trigger point, remember only the first of the terminal actions will be chosen, and it fires after all other non-terminal actions have been fired.

Configuring the Action Type

If your action type requires a configuration page, perform the following:

- Create record definition(s), keyed by EOCF_ACTION_ID, to capture design time configuration data.

- Create an action configuration component and page.
- Insert EOCF_ACTN_CFG_SBPG in the configuration page. This displays the action type and action name on the configuration page.
- Insert the policy information subpage, EOCF_RULEINFO_SBP, on the page. This displays policy information including the policy name and condition text.
- If you need to access terms, put a clone of EOCF_SRCH_DSP_AL as a secondary page on your action configuration page; and add the following code in the page's activate. (See the display alert action configuration page, EOCF_DSPL_ALRT_CFG, as an example of a configuration page)

```
import EOCF_CLF_RB:Definition:Rule:Rule;
import EOCF_CLF_RB:UI:*;

Component SearchBuilder &cobjSearchBuilder;
Component SearchConfig &cobjSearchConfig;
Global Rule &gobjRule;
Local RuleBuilder &objRuleBuilder = create RuleBuilder();
 &cobjSearchConfig = create EOCF_CLF_RB:UI:SearchConfig();

/*** Build Search page display **/
/*** Add where clause to filter terms by scalar **/
&cobjSearchConfig.AddCustomWhereClause(FetchSQL(SQL.EOCF_WHERECL_TMSCALAR), 0, "");
&cobjSearchBuilder = create SearchBuilder(EOCF_ACTION_WRK.EOCF_CONTEXT_ID, &cobj⇒
SearchConfig);
&cobjSearchBuilder.BuildDisplay();
/*** Populate Rule info **/
&objRuleBuilder.PopulateRuleInfo(&gobjRule);
```

- Insert a push button on the configuration page to transfer to this secondary page (for the purpose of choosing a term).

  Some of the terms on your configuration page may be configurable. Therefore, you must paste another subpage, EOCF_TERMCFG_SBP, on the configuration page for this purpose.
- Insert the following page activate code on your action configuration page to initialize the term configuration subpage:

```
import EOCF_CLF_RB:Definition:Rule:Rule
import EOCF_CLF_RB:UI:*;
import EOCF_CLF_RB:UI:ConfigBuilder;
import EOCF_CLF_RB:Definition:RuleTermConfig;
import EOCF_CLF_DL:Factory:DataLibraryFactory;
import EOCF_CLF_DL:Definition:Term:TermDefn;
import EOCF_CLF_DL:Utility:DLConstants;
Local Rowset &rs_level0, &rsActionTypeDefn;
Local number &bundleId, &actionId;
Local number &actionTypeId;
Local string &isTerminal, &isCombinable, &isConfigurable;
Local string &aeTriggered, &appMsgTriggered, &ciTriggered, &piaTriggered;
Local string &commit, &path, &id, &dtAppClassId, &appClsPath, &dtAppClsPath, &rtApp⇒
ClassId, &rtAppClsPath, &actTypeName, &actionName;
Global EOCF_CLF_RB:Definition:Rule:Rule &gobjRule;
```

```
Local EOCF_CLF_RB:UI:RuleBuilder &objRuleBuilder = create EOCF_CLF_RB:UI:Rule⇒
Builder();
Local EOCF_CLF_RB:Definition:RuleTermConfig &objRuleTermConfig = create EOCF_CLF_RB:⇒
Definition:RuleTermConfig();
Local EOCF_CLF_DL:Factory:DataLibraryFactory &objDlFactory = create EOCF_CLF_DL:⇒
Factory:DataLibraryFactory();
Local EOCF_CLF_DL:Definition:Term:TermDefn &objTermDefn = create EOCF_CLF_DL:⇒
Definition:Term:TermDefn();
Local EOCF_CLF_DL:Utility:DLConstants &objDLConstants = create EOCF_CLF_DL:Utility:⇒
DLConstants();
/*********************************************************************/
/* CREATE A CONFIG BUILDER TO CREATE UI ELEMENTS THAT ENABLE TERM    */
/* CONFIGURATION-TO BE DONE ONLY BY ACTIONS THAT NEED TO ACCESS DATA */
/* LIBRARY TERMS, AND THUS HAVE THE TERM-PICKER SCROLL AND THE TERM  */
/* CONFIGURATION SUBPAGE ON THE PAGE */

Local ConfigBuilder &objConfigBuilder;
Local Rowset &rs1 = GetLevel0()(1).GetRowset(Scroll.EOCF_DS_ALT_TRM);
&objConfigBuilder = create ConfigBuilder();
&objConfigBuilder.InitDisplay();
For &i = 1 To &rs1.ActiveRowCount
   &termId = &rs1(&i).EOCF_DS_ALT_TRM.EOCF_LIB_TERM_ID.Value;
   If All(&termId) Then
      If All(&rs1(&i).EOCF_DS_ALT_TRM.EOCF_CONFIG_ID.Value) Then
         /*******************************/
         &objRuleTermConfig = create EOCF_CLF_RB:Definition:RuleTermConfig();
         /*******************************/
         &objRuleTermConfig.EOCF_CONFIG_ID = &rs1(&i).EOCF_DS_ALT_TRM.EOCF_CONFIG_⇒
ID.Value;
         &objRuleTermConfig.getConfiguredTerm( True);
         &rs1(&i).EOCF_DS_AL2_WRK.EOCF_GOTO_BTN.Label = &objRuleTermConfig.Get⇒
ConfigTermLabel(EOCF_ACTION_WRK.SETID.Value);
         &rs1(&i).EOCF_DS_AL2_WRK.EOCF_GOTO_BTN.Enabled = True;
      Else
         &objTermDefn = &objDlFactory.getTermDefn(&termId, False, &obj⇒
DLConstants.ALLOCATIONTYPE_READONLY);
         &rs1(&i).EOCF_DS_AL2_WRK.EOCF_GOTO_BTN.Label = &objTermDefn.EOCF_TERM_⇒
LABEL;
         &rs1(&i).EOCF_DS_AL2_WRK.EOCF_GOTO_BTN.Enabled = False;
      End-If;
   Else
/*************/
&rs1(&i).EOCF_DS_AL2_WRK.EOCF_GOTO_BTN.Label = " ";
 &rs1(&i).EOCF_DS_AL2_WRK.EOCF_GOTO_BTN.Enabled = False;
   End-If;
End-For;

Local Grid &TERMGRID;
Local GridColumn &TERMCOLUMN, &LOOKUPCOLUMN;
```

```
&TERMGRID = GetGrid(Page.EOCF_DSPL_ALRT_CFG, "EOCF_DS_ALT_TRM");
&TERMCOLUMN = &TERMGRID.GetColumn("EOCF_GOTO_BTN");
&TERMCOLUMN.Label = " ";
&LOOKUPCOLUMN = &TERMGRID.GetColumn("EOCF_CONFIGURE");
&LOOKUPCOLUMN.Label = MsgGetText(18112, 2541, "Message not found - Get Term");
/*****************************************************************/
/*POPULATE RULE INFO - NEEDS TO BE DONE BY ALL CLF ACTIONS, i.e.,    */
/*ACTIONS WHOSE CONFIG PAGES HAVE BEEN NAVIGATED TO VIA RULE BUILDER */

&objRuleBuilder.PopulateRuleInfo(&gobjRule);
/*****************************************************************/
/* GET THIS ACTION'S ACTION TYPE NAME - TO BE DONE BY ALL ACTIONS */

&rs_level0 = GetLevel0();
&actionId = &rs_level0(1).EOCF_ACTION_WRK.EOCF_ACTION_ID.Value;
&actionTypeId = &rs_level0(1).EOCF_ACTION_WRK.EOCF_ACTION_TYP_ID.Value;
&actionName = &rs_level0(1).EOCF_ACTION_WRK.EOCF_ACTION_NAME.Value;

/* Get details of Action Type */
rem SQLExec(SQL.EOCF_ACTTYP_DTLS2_SEL, &actionTypeId, &actTypeName, &dtAppClassId,⇒
 &dtAppClsPath, &rtAppClassId, &rtAppClsPath, &isTerminal, &isConfigurable, &is⇒
Combinable, &aeTriggered, &appMsgTriggered, &ciTriggered, &piaTriggered, &Commit⇒
Flag, &dtCommit, &descr);

/***** Get details of Action Type ***/
/**** Modified by SB to enable rel. language processing for action type name ***/

&rsActionTypeDefn = CreateRowset(Record.EOCF_ACTN_TYPE);
&numrows = &rsActionTypeDefn.Fill("WHERE EOCF_ACTION_TYP_ID = :1", &actionTypeId);

If &numrows > 0 Then
   &actTypeName = &rsActionTypeDefn(1).GetRecord(1).EOCF_ACT_TYP_NAME.Value;
   &dtAppClassId = &rsActionTypeDefn(1).GetRecord(1).EOCF_DT_APPCLASSID.Value;
   &dtAppClsPath = &rsActionTypeDefn(1).GetRecord(1).EOCF_DT_APPCLSPATH.Value;
   &rtAppClassId = &rsActionTypeDefn(1).GetRecord(1).EOCF_RT_APPCLASSID.Value;
   &rtAppClsPath = &rsActionTypeDefn(1).GetRecord(1).EOCF_RT_APPCLSPATH.Value;
   &isTerminal = &rsActionTypeDefn(1).GetRecord(1).EOCF_IS_TERMINAL.Value;
   &isConfigurable = &rsActionTypeDefn(1).GetRecord(1).EOCF_IS_CFG_FLAG.Value;
   &isCombinable = &rsActionTypeDefn(1).GetRecord(1).EOCF_IS_CMBIN_FLAG.Value;
   &aeTriggered = &rsActionTypeDefn(1).GetRecord(1).EOCF_AE_TRIGGERED.Value;
   &appMsgTriggered = &rsActionTypeDefn(1).GetRecord(1).EOCF_APPMSG_TRIGGR.Value;
   &ciTriggered = &rsActionTypeDefn(1).GetRecord(1).EOCF_CI_TRIGGERED.Value;
   &piaTriggered = &rsActionTypeDefn(1).GetRecord(1).EOCF_PIA_TRIGGERED.Value;
   &CommitFlag = &rsActionTypeDefn(1).GetRecord(1).EOCF_COMMIT_FLAG.Value;
   &dtCommit = &rsActionTypeDefn(1).GetRecord(1).EOCF_DT_COMMIT.Value;
   &descr = &rsActionTypeDefn(1).GetRecord(1).DESCR.Value;
End-If;

/* Populate the ActionTypeName field for display */
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_ACT_TYP_NAME.Value = &actTypeName;
```

```
/*********************************************************************/
/* PLEASE DO THE FOLOWING IF YOU NEED THE DETAILS OF THE ACTION TYPE */
/* THAT WERE ENTERED AT THE TIME OF ACTION TYPE Registration*/
/*  This Display Alert action Does not need these details*/
/*
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_DT_APPCLASSID.Value = &dtAppClassId;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_DT_APPCLSPATH.Value = &dtAppClsPath;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_RT_APPCLASSID.Value = &rtAppClassId;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_RT_APPCLSPATH.Value = &rtAppClsPath;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_IS_CFG_FLAG.Value = &isConfigurable;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_IS_CMBIN_FLAG.Value = &isCombinable;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_IS_TERMINAL.Value = &isTerminal;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_APPMSG_TRIGGR.Value = &appMsgTriggered;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_CI_TRIGGERED.Value = &ciTriggered;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_AE_TRIGGERED.Value = &aeTriggered;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_PIA_TRIGGERED.Value = &piaTriggered;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_COMMIT_FLAG.Value = &CommitFlag;
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_DT_COMMIT.Value = &dtCommit;
&rs_level0(1).EOCF_DSTIME_WRK.DESCR.Value = &descr;
*/
/*********************************************************************/
/* PLEASE DO THE FOLLOWING IF  THE ACTION IS A COMBINABLE ONE AND YOU */
/* NEED THE INFO REGARDING THE ACTION BUNDLE OF WHICH IT IS A PART   */
/* This Display Alert Action , though combinable, does not need this */
/*
SQLExec(SQL.EOCF_ACT_BUNDLE_SEL, &actionTypeId, &ActnBundleId);
&rs_level0(1).EOCF_DSTIME_WRK.EOCF_ACT_BUNDLE_ID.Value = &ActnBundleId;*/
/*********************************************************************/
```

Creating Design and Runtime Application Class Code

To create the design-time and runtime application class code:

1. Create an application package named *<your product code>_AAF_ACTIONS*.
2. Create a class in the package to be your design-time action application class.

   It must extend the EOCF_CLF_AF:ActionCfgBase and must implement the designAction() method. This class is responsible for transferring from the policy builder page to the action configuration page previously created.

   All relevant information regarding the rule and the context will be available to the configuration page.

   (See the EOCF_CLF_ACTIONS:DisplayAlertCfg as an example.)
3. Create another class in the package to be your runtime application class.

   It must extend EOCF_CLF_AF:ActionBase and must implement the runAction() method. This class is responsible for launching the action.

   All the relevant information that the action needs to perform will be available to it.

   (See the EOCF_CLF_ACTIONS:DisplayAlert as an example.)
4. Register the action type in the Register Action Type page.

Specify the name of your new action type, the names of your design-time and runtime application classes and other action type characteristics .

Specify the trigger types and trigger points that may have policies using this action type. For example you may want an action to be triggered during ComponentPostBuild trigger types, specifically the trigger point "When a Defect is Presented, in Quality Management." After completing these configuration steps, the action type will be available for use.

5. If an action type is combinable with other action types, register it in an action type bundle in addition to the other combinable action types.

   All display action types must be included in a single action type bundle.

6. If your new action is configurable, click Configure on the page.

   This transfers you to the action configuration page for more specifications.

7. Insert the subpage EOCF_DISPLAY_SUBPG in Level 0 on all pages that you want display-enabled.

Note. This display subpage must not overlap any other field and there should be no popup blockers running.

Considerations When Creating an Application Class Implementation

PeopleSoft recommends:

- Implementations of related terms be grouped in a single class in which different methods are responsible for deriving the term's value. This practice facilitates maintaining a manageable number of application classes.
- In situations where the probability that multiple related terms are accessed during a single business event is high:
  - It is preferable to have a single implementation return a rowset, record, or vector containing the data for as many of the terms as possible.
  - When defining the term, specify which data element in the rowset, record, or vector is to be used for the term.

Technical Details of Application Class Implementations

The data library calls the appropriate application class based on the class and package information registered with the implementation. The application class is instantiated automatically and the specified application class method is invoked. The application class constructor must be coded to accept an instance of the AccessMethod class.

For example:

```
class LeadOppMetrics
 method LeadOppMetrics(&_oAccessMethod As EOCF_CLF_DL:Runtime:AccessMethods:Access⇒
Method);
 method LeadCountbyBO();
    private instance EOCF_CLF_DL:Runtime:AccessMethods:AccessMethod &ioAccess⇒
Method;
end-class;
```

The AccessMethod object allows the application class to access values for all the implementation binds (input parameters) needed by the implementation and all the information about the term being resolved.

Example of an Application Class Accessing Implementation Binds

The following partial code shows how an application class can access any of its implementation binds.

```
method GetCaseID Local any &CaseID;
  &CaseID = &ioAccessMethod.getBindValueByName("CASE_ID");
  Local EOCF_CLF_DL:Runtime:Results:ResultsScalar &oResultsScalar;
  &oResultsScalar = create EOCF_CLF_DL:Runtime:Results:ResultsScalar(&CaseID);
      &ioAccessMethod.Results = &oResultsScalar;
end-method;
```

The application class does not need to know how the data is retrieved and passed by the data library engine. The data could have been passed by the calling application; could have been defined as a constant in context definition; or, could have been defined as a term and resolved by the data library engine. The application class can retrieve the data either by position or by name.

```
/* BO_ID is the name of the implementation bind;the bind name specified here⇒
 matches the one to be specified at the time of registering the implementation in⇒
 &nBOID = &ioAccessMethod.getBindValueByName("BO_ID"); Here the request is⇒
 made to retieve the value for the first bind. */
&nBOID = &ioAccessMethod.getBindValueByPosition(1);
```

Another way the calling application can pass additional information to implementations or policy actions is to have the calling application use the addUserVariable method to add the information to the context object. To access this information, the application class uses the getUserVariable method to retrieve the content.

For example:

```
&AdditionalData = &oAccessMethod.AMContext.getUserVariable("ExtraInfo);
```

Note. The shape of the data received by an application class using any of the previous methods will be a PeopleCode Any object. Therefore, it's the responsibility of the application class to convert the object to the appropriate data type before the value is consumed. If the application class has a method to be invoked by the data library engine, then that method should not require any parameters.

Sample Methods of an Application Class Resolving Terms

The following code is an example of an application class having several methods for resolving terms.

```
class OperatorInfo
 method Get_Person_Name();
  method Get_Person_Salutation();
 method Get_Person_Title();
 method Get_Person_Gender();
 method Get_Person_BirthDate();
  method OperatorInfo(&_oAccessMethodParam As EOCF_CLF_DL:Runtime:AccessMethods:⇒
AccessMethod);
  private
   instance EOCF_CLF_DL:Runtime:AccessMethods:AccessMethod &ioAccessMethod;
end-class;
```

If the data library engine invokes an application class method, that method is responsible for deriving the results and for populating the results in the AccessMethod object. The data library engine transmits the results to the calling application. If an application class method is not specified in the implementation definition, the constructor is responsible for performing these tasks.

The data library provides several mechanisms through which the application class can return the output value to the data library:

- The application class instantiates one of the following application classes with the output value that needs to be passed to the engine.
  - `ResultsRecord Record &oResultsRecord = create EOCF_CLF_DL:Runtime:Results:ResultsRecord(RecordVariable);`
  - `ResultsRowset Rowset &oResultsRowset = create EOCF_CLF_DL:Runtime:Results:ResultsRowset(RowsetVariable);`
  - `ResultsScalar Scalar &oResultsScalar = create EOCF_CLF_DL:Runtime:Results:ResultsScalar(ScalarVariable);`
  - `ResultsVector Vector &oResultsVector = create EOCF_CLF_DL:Runtime:Results:ResultsVector(VectorVariable);`
- The type of class to be instantiated depends upon the type of data that needs to be returned.
- The instantiated object is assigned to the member of AccessMethod object.

Note. Application class objects are not cached. When the data library engine invokes the application class because the data is not available in the cache, the application class is instantiated by invoking the constructor, then the method specified in the implementation. Therefore, the instance variables created in the constructor cannot be shared across multiple methods of the same class to resolve different terms.

See Chapter 11, "Setting Up the Data Library," Understanding Term Properties, page 136.

Example of How a Value Can Be Passed to the Data Library

The following sample is a partial code listing of how a value can be passed to the data library:

```
method GetCaseID
Local any &CaseID;
   &CaseID = &ioAccessMethod.getBindValueByName("CASE_ID");
/* use the appropriate sub class of Results class for assigning value */
Local EOCF_CLF_DL:Runtime:Results:ResultsScalar &oResultsScalar;
&oResultsScalar = create EOCF_CLF_DL:Runtime:Results:ResultsScalar(&CaseID);
&ioAccessMethod.Results = &oResultsScalar;end-method;
```

Sample Code of an Application Class Implementation

The following is a sample of an application class implementation.

```
import EOCF_CLF_DL:Runtime:AccessMethods:*;
import EOCF_CLF_DL:Runtime:Resolution:*;
import EOCF_CLF_DL:Contexts:*;
import EOCF_CLF_DL:Runtime:Results:*;
```

```
class CaseRecordInformation
   method CaseRecordInformation(&_oAccessMethod As EOCF_CLF_DL:Runtime:Access⇒
Methods:AccessMethod);
   method GetResultRecord();

private
   instance EOCF_CLF_DL:Runtime:AccessMethods:AccessMethod &ioAccessMethod;
end-class;

method CaseRecordInformation
   /+ &_oAccessMethod as EOCF_CLF_DL:Runtime:AccessMethods:AccessMethod +/

   &ioAccessMethod = &_oAccessMethod;
   end-method;

method GetResultRecord
   Local Record &result;
   Local number &nResolved_value, &nCaseID;
   Local string &sBusUnit;

/* Retrieving the implementation bind using the API */
   &nCaseID = &ioAccessMethod.getBindValueByName("CASE_ID");

   SQLExec("SELECT BUSINESS_UNIT FROM PS_RC_CASE WHERE CASE_ID = :1", &nCaseID, &s⇒
BusUnit);
   &result = CreateRecord(Record.RC_CASE);
   &result.CASE_ID.Value = &nCaseID;
   &result.BUSINESS_UNIT.Value = &sBusUnit;
      &result.SelectByKey();
/* Passing the values (in this case, it is a record) back to the data
   library engine - this record could be used to resolve multiple terms */
   &ioAccessMethod.Results = (create EOCF_CLF_DL:Runtime:Results:ResultsRecord⇒
(&result));

end-method;
```

Considerations When Creating a PS Query Implementation

PeopleSoft recommends that when multiple related terms may be accessed during a single business event:

- Create a single PS Query implementation to return a rowset containing the data for several terms
- Specify which data element or field position in the rowset or record is to be used for the term.
- Ensure appropriate privileges have been set for accessing the objects present in the query.

Note. Use caution when using this type of implementation because the query may execute very slowly.

Considerations When Creating a Policy

PeopleSoft Active Analytics Framework is neither a programming tool nor a substitute for PeopleCode. Do not use the framework as a way to program policies that do not need to be configurable. If policies are static, the use of PeopleCode is the best method. The framework should not be used as a tool to configure the presentation layer: for dynamically hiding or unhiding fields, making the fields editable or not, and so on. Furthermore, the data library should not be used when there is no need for the data abstraction layer.

Use the PeopleSoft Active Analytics Framework when there's a need for business users to configure business processes with business rules, without having to customize the application.

Before building policies, consider that each term used in conditions or in actions within policies, could negatively affect performance of the application component, especially if the term's retrieval mechanism is time consuming.

PART 5

Portal Utilities

Chapter 16
Working With Pagelet Wizard

Chapter 17
Setting Up Pagelet Wizard Options

Chapter 18
Using Pagelet Wizard

Chapter 19
Administering Pagelets

Chapter 20
Using Pagelet Wizard Post-Transformation Processing

Chapter 21
Working with Navigation Pages

Chapter 22
Setting Up Navigation Collection Options

Chapter 23
Using Navigation Collections

Chapter 24
Running Portal Utility Processes

CHAPTER 16

Working With Pagelet Wizard

This chapter discusses:

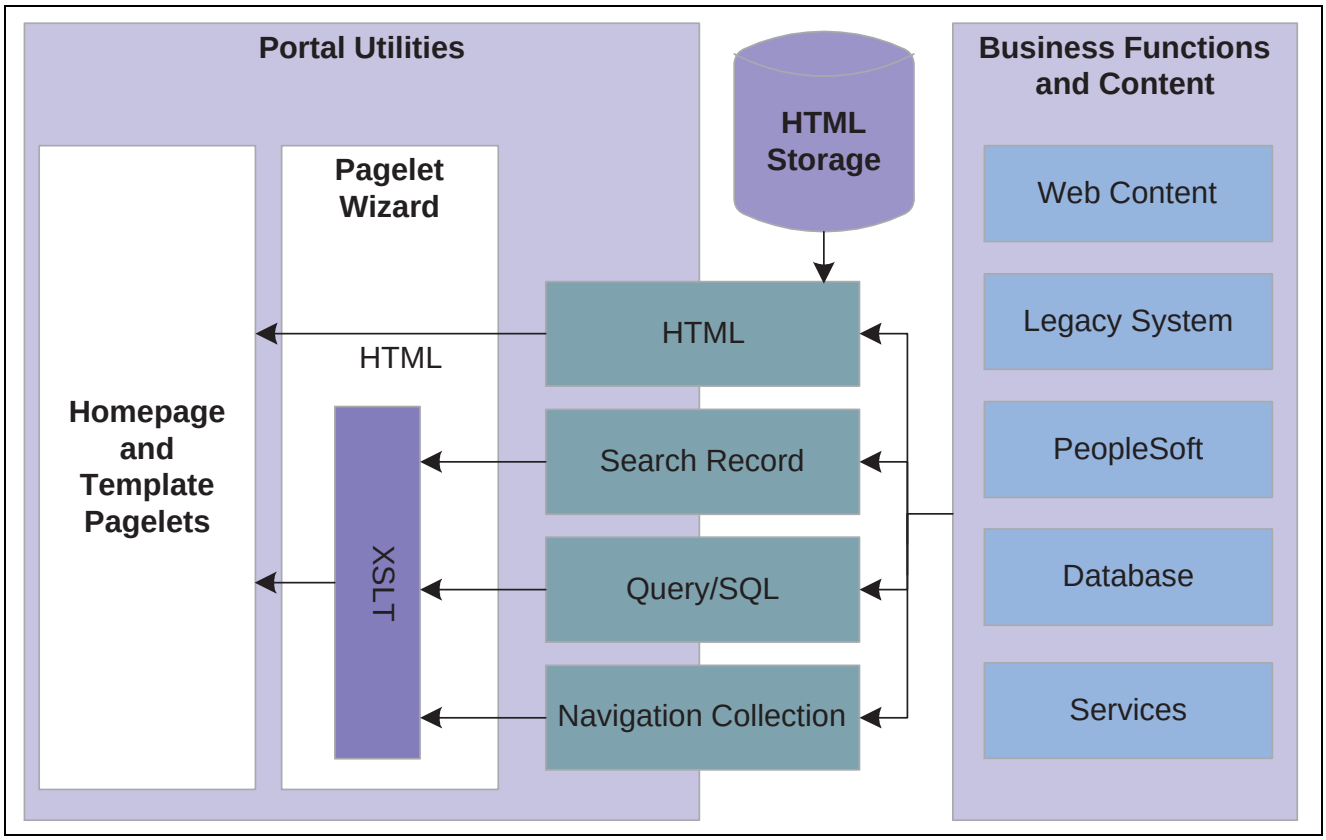
- An overview of Pagelet Wizard.
- Pagelet Wizard data sources.
- Pagelet Wizard display formats.
- Pagelet Wizard pagelet display sizes.
- Pagelet Wizard pagelet types.

Note. Examples of PeopleSoft Pure Internet Architecture pages and pagelets in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages and pagelets may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page and pagelet elements displayed remain the same.

Overview of Pagelet Wizard

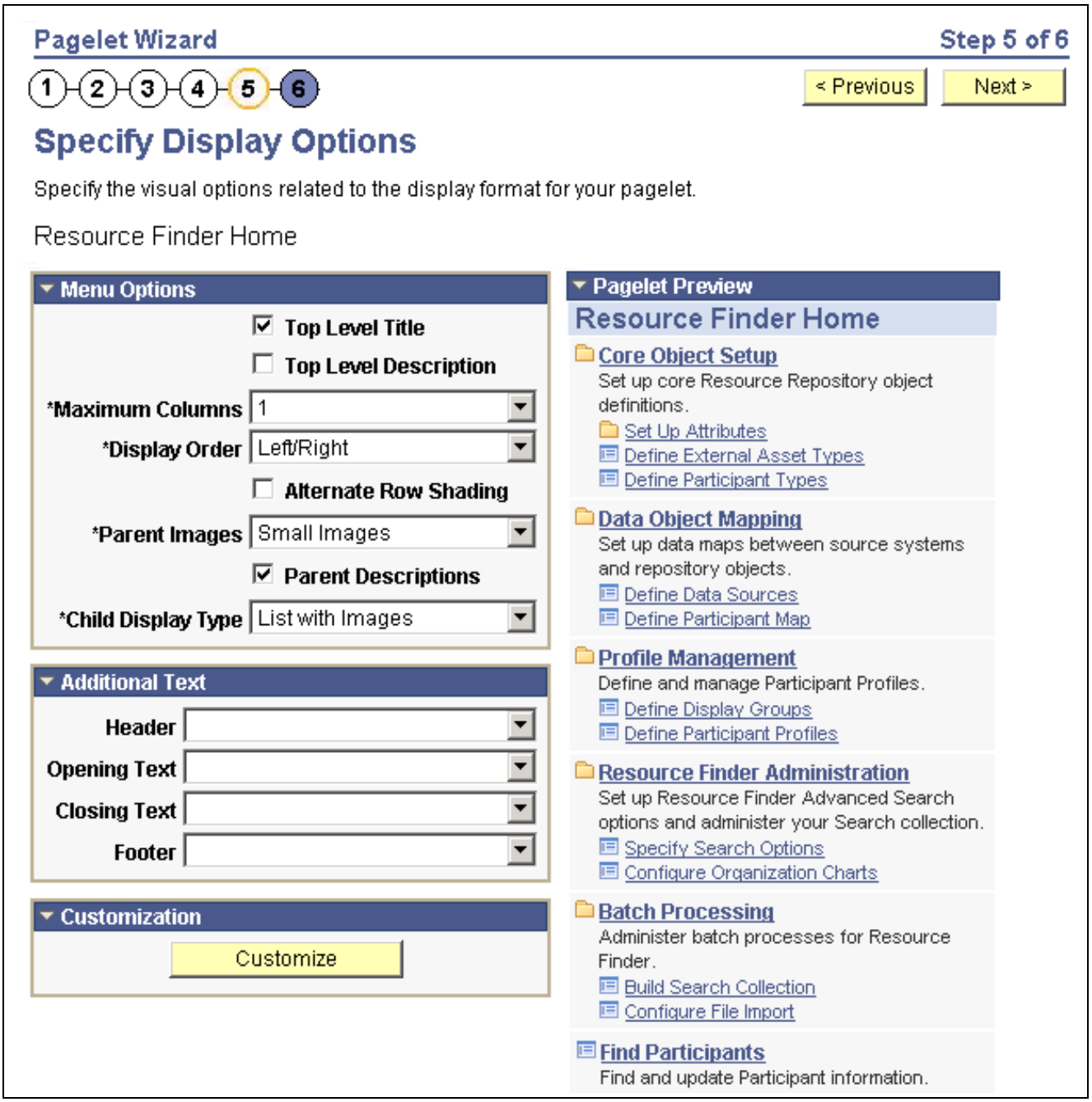
Pagelets are small pages that provide display-only snapshots of PeopleSoft system or non-PeopleSoft system content. Pagelet Wizard provides a user-friendly, browser-based graphical user interface (GUI) that leads you through the series of steps involved in creating and publishing a pagelet.

Portal administrators and nontechnical users can use Pagelet Wizard to integrate and transform data from a variety of data sources, both internal and external to the PeopleSoft system. The following diagram illustrates a high-level Pagelet Wizard architectural process flow:



Pagelet Wizard architectural process flow

You do not need to have PeopleSoft-specific application development tools or skills to use Pagelet Wizard to create pagelets. During the pagelet-creation process, the Pagelet Wizard GUI presents a series of numbered steps. Each step appears in a numbered path at the top of the Pagelet Wizard component to indicate where you are in the pagelet creation process.



Numbered steps on the Specify Display Options page

This labeled path assists you in navigating through the steps, and it enables you to review or edit pagelet definition values. Once you've completed a step, Pagelet Wizard automatically takes you to the next appropriate step.

The Pagelet Wizard GUI leads you through the following steps that are typically used to create a pagelet:

1. Specify pagelet information.
2. Select data source.
3. Specify data source parameters.
4. Select display format.

5. Specify display options.
6. Specify publishing options.

Note. Before you start building a pagelet with Pagelet Wizard, you should familiarize yourself with all data sources. This enables you to evaluate the best way to retrieve the data that you want to display on the pagelet.

See Also

Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Data Sources, page 188

Pagelet Wizard Data Sources

Pagelet Wizard supports the creation of pagelets using the following data sources:

HTML Use to present appropriately sized static HTML pages in Pagelet Wizard pagelets. HTML data source pagelets are compatible with the passthru and custom display formats.

Navigation Collections Use to create Pagelet Wizard pagelets from existing sets of folders and links stored in Navigation Collections. Navigation Collection data source pagelets are compatible with the menu and custom display formats.

PeopleSoft Query Use to access PeopleSoft application data or any tables within the PeopleSoft environment through Pagelet Wizard pagelets. PeopleSoft Query data source pagelets are compatible with the table, list, chart, passthru, and custom display formats.

Search record Use to create Pagelet Wizard pagelets that provide quick access to specific transactions on a target transaction page. The search record data type enables you to select a basic search record or a component interface.

Use the basic search record data type for simple search lookup pages.

Use the component interface search record data type when the basic search record does not provide the required functionality; for example, when security or PeopleCode considerations need to be taken into account.

Search record data source pagelets are compatible with the search list display format.

You can define your own data types on the Define Data Types page.

See Also

Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Display Formats, page 189

Chapter 23, "Using Navigation Collections," page 291

Chapter 17, "Setting Up Pagelet Wizard Options," Defining Data Types, page 205

Chapter 18, "Using Pagelet Wizard," Step 2: Selecting a Pagelet Data Source, page 219

Pagelet Wizard Display Formats

Pagelet Wizard supports the creation of pagelets based on the following display formats:

- Table.
- List.
- Chart.
- Menu.
- Search list.
- Passthru.
- Custom (Extensible Stylesheet Language, or XSL).

Table Display Format

The table display format is appropriate for creating a pagelet that displays data from the PeopleSoft Query data source. Following is an example of a table pagelet displaying data from a PeopleSoft Query data source:

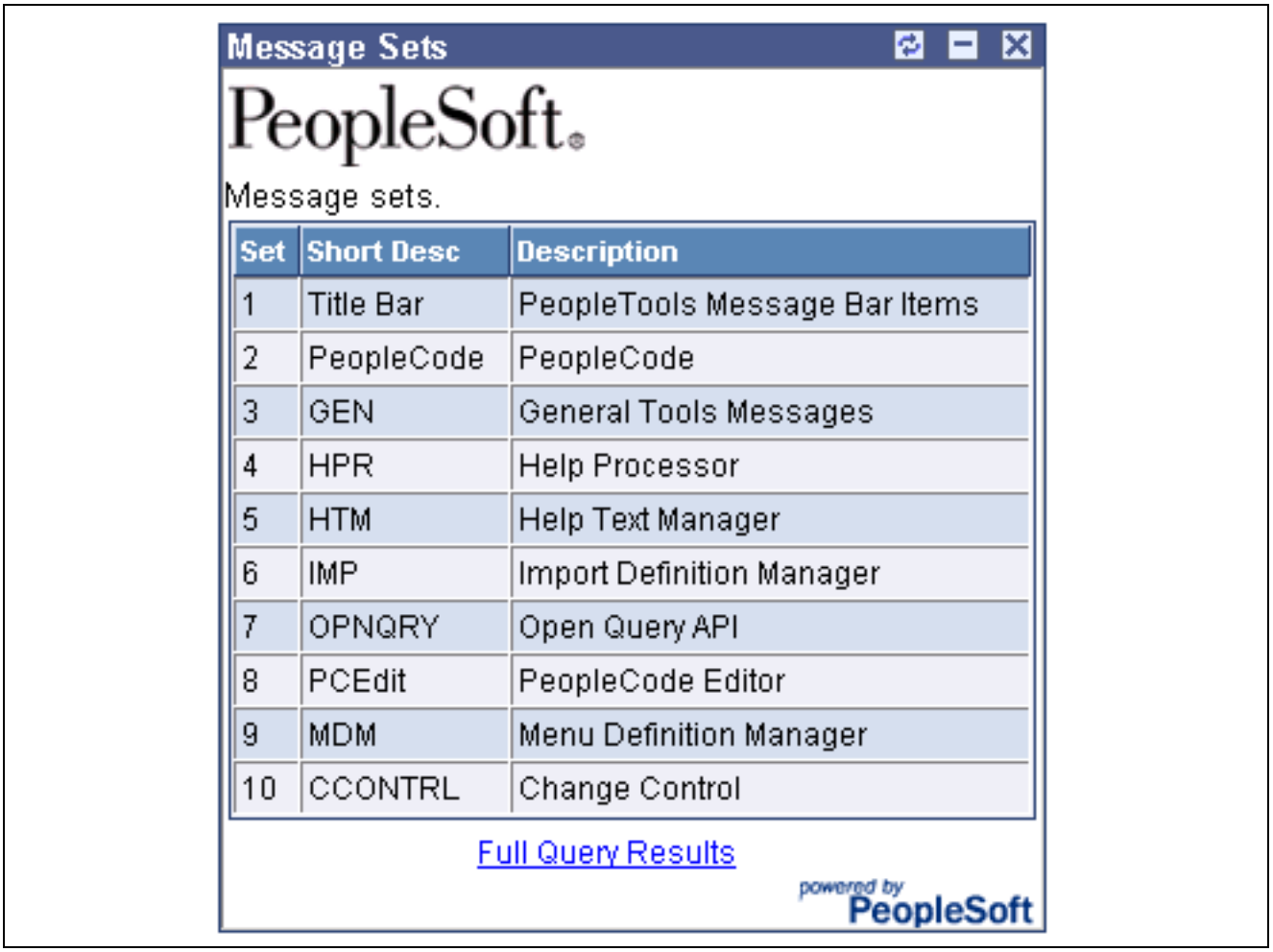


Table pagelet example

List Display Format

The list display format is appropriate for creating a pagelet that displays data from the PeopleSoft Query data source. Following is an example of a list pagelet displaying data from a PeopleSoft Query data source:

List pagelet example

Chart Display Format

The chart display format is appropriate for creating a pagelet that displays data from the PeopleSoft Query data source. Following is an example of a chart pagelet displaying data from a PeopleSoft Query data source:

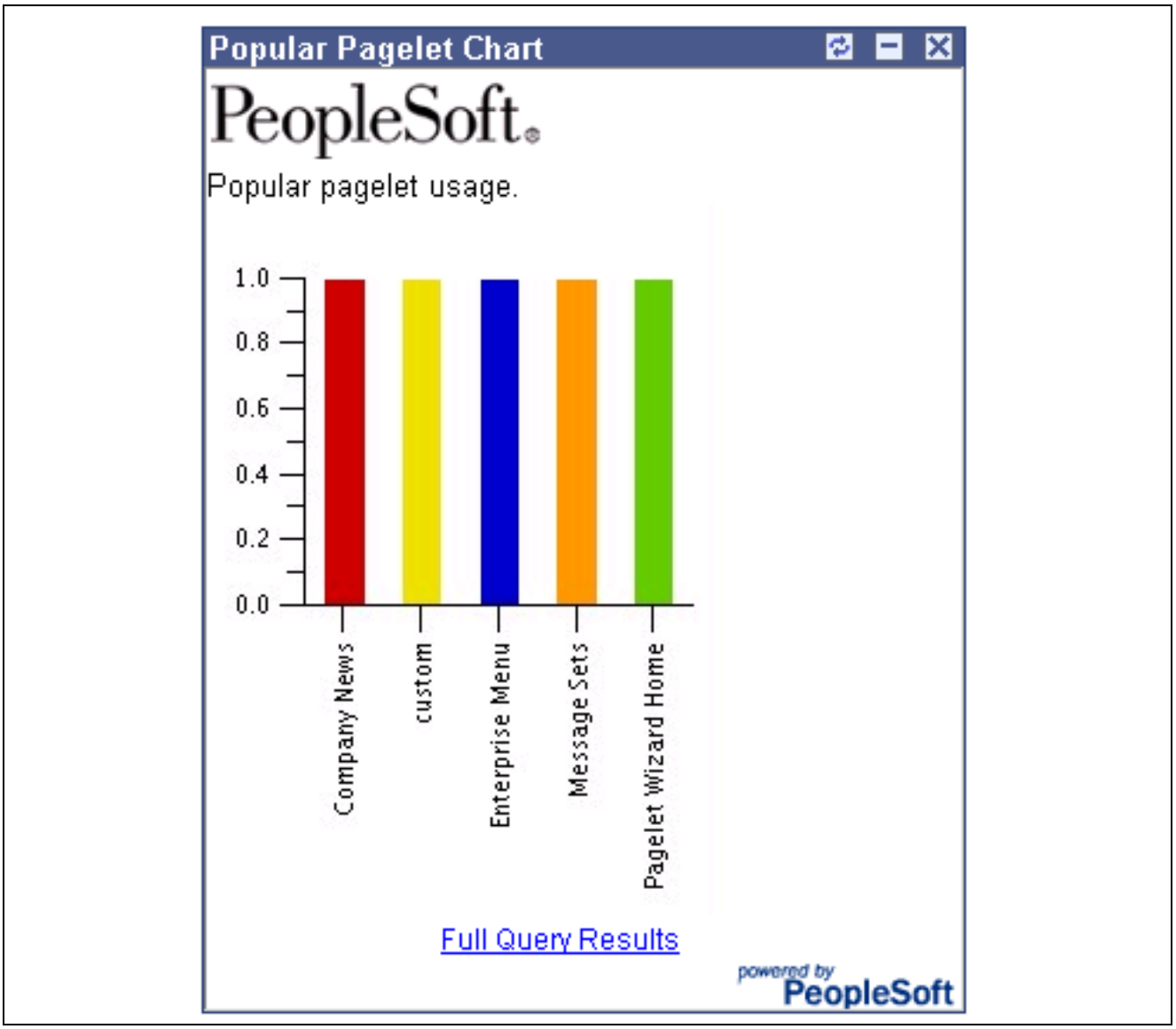


Chart pagelet example

Menu Display Format

The menu display format is appropriate for creating a pagelet that displays data from a Navigation Collection data source. Following is an example of a menu pagelet displaying data from a Navigation Collection data source:

Menu pagelet example

Search List Display Format

The search list display format is appropriate for creating pagelets that display data from the search record data source. Following are examples of search list pagelets displaying data from a search record data source:

Search list pagelet example (Show Results options set to N)

Search list pagelet example (Show Results option set to Y)

Passthru Display Format

The passthru display format is appropriate for creating a pagelet that displays data from the HTML and PeopleSoft Query data sources. Because the passthru display format performs no further transformations on data from the data source, it should only be used with data sources that return HTML that can be directly displayed as a pagelet. Following is an example of a passthru pagelet displaying data from an HTML data source:

Passthru pagelet example

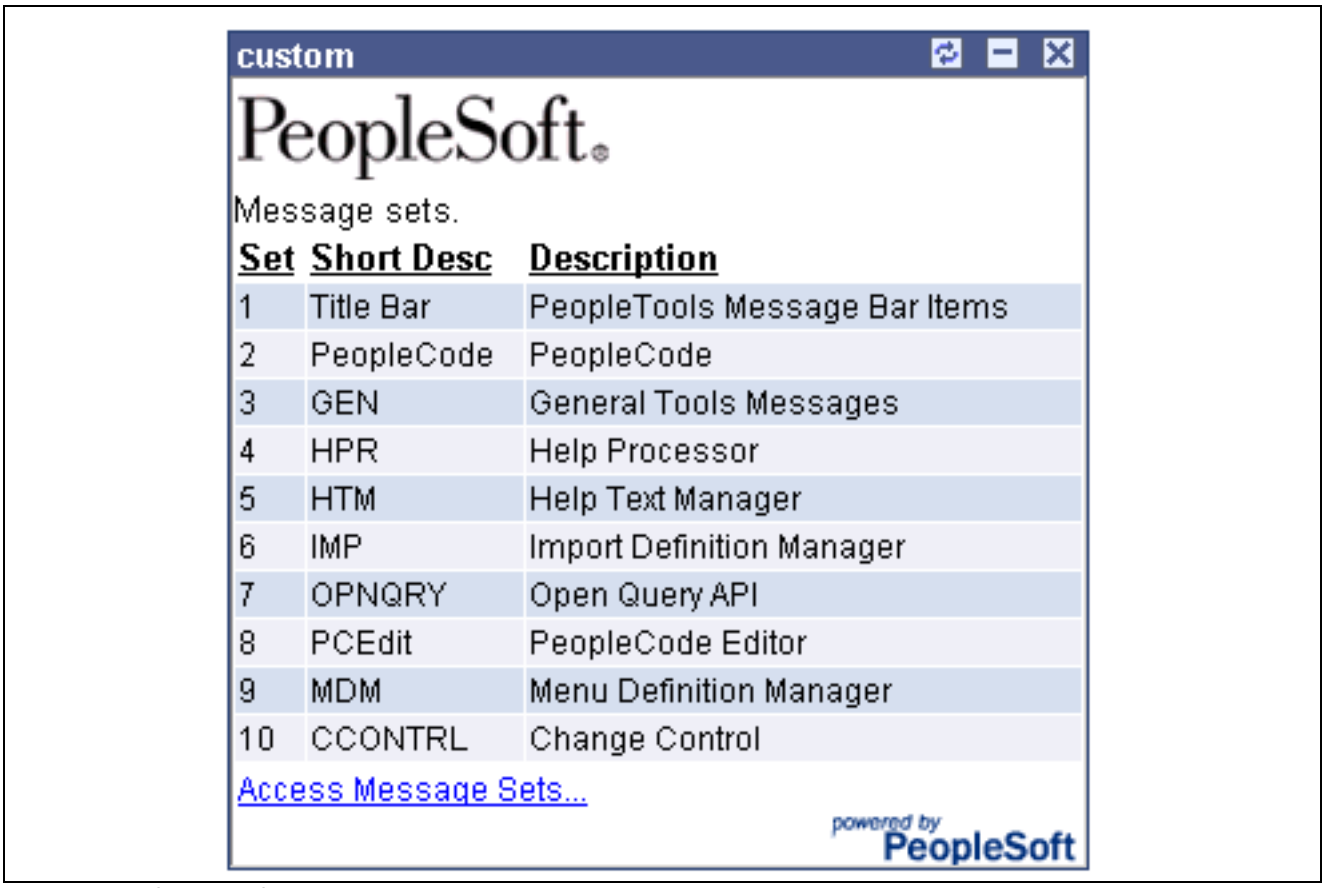
Custom Display Format

The custom display format is appropriate for creating a pagelet that displays data from the following data sources:

- HTML.
- Navigation Collection.
- PeopleSoft Query.
- Search record.

Use the custom display format when you want to use a display format other than the ones expressly listed in the Pagelet Wizard. When you select the custom display format, Pagelet Wizard enables you to select a format provided by one of the delivered XSL prototypes, or you can select an XSL template that you created on the Define XSL page.

Following is an example of a custom pagelet displaying data from a PeopleSoft Query data source:



Custom pagelet example

This custom pagelet uses the same PeopleSoft Query data source as the table pagelet example shown earlier; however, this example has a custom XSL template applied.

See Also

Enterprise PeopleTools 8.45 PeopleBook: PeopleCode API Reference, "Chart Class"

Chapter 17, "Setting Up Pagelet Wizard Options," Defining Pagelet Wizard XSL Prototypes, page 207

Chapter 18, "Using Pagelet Wizard," Step 4: Selecting a Pagelet Display Format, page 231

Pagelet Wizard Display Sizes

In addition to the display format, pagelet sizing is also important during pagelet creation. Pagelet Wizard uses pagelet sizing that follows a basic set of rules that enable pagelets to appear properly on a PeopleSoft portal homepage. Pagelet size corresponds to the homepage layout that is specified. You can specify either a two-column layout that displays one narrow pagelet and one wide pagelet, or a three-column layout that displays three narrow pagelets.

Pagelets that you design with the Pagelet Wizard should conform to the dimensions of the narrow column pagelet and (optionally) the wide column pagelet. A narrow pagelet should be 240 pixels wide. Subtracting the border and the internal margin provides 218 pixels for content. A wide pagelet should be 490 pixels wide. Subtracting the border and the internal margin provides 468 pixels for content.

If you display a pagelet that is wider than the homepage column that contains it, an error does not occur. The column expands to accommodate the content; however, this may require horizontal scrolling on the homepage, which can be inconvenient for the end user. In accordance with general pagelet design principles, make the pagelet height as short as possible.

See Also

PeopleSoft 8.8 Enterprise Portal PeopleBook, "Developing Pagelets," Selecting Pagelet Size

Pagelet Wizard Pagelet Types

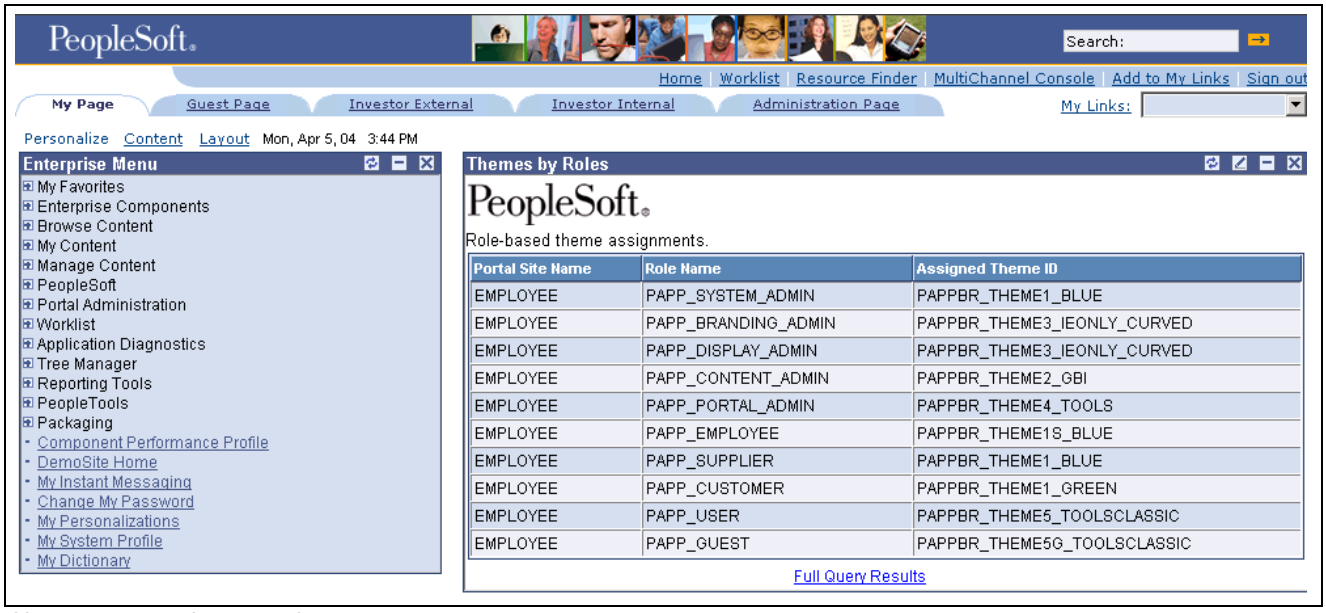
Pagelet Wizard supports the creation of the following pagelet types:

- Homepage pagelets.
- Template pagelets.

Homepage Pagelet

A homepage pagelet is a version of a pagelet that is available for use on the portal homepage. The layout and content of the homepage is usually configurable by the end user, and homepage pagelets can be designated as optional or required.

See *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portal Homepages and Pagelets," Selecting Tab Content



Homepage pagelet example

Note. The option to use Pagelet Wizard pagelets published as homepage pagelets is available if you have licensed PeopleSoft Enterprise Portal or a Portal Pack.

Template Pagelet

A template pagelet is a version of a pagelet that can be added to a PeopleSoft template. When used in conjunction with Intelligent Context Manager™ in the PeopleSoft Enterprise Portal, these pagelets can extract and leverage keys specified by the target transaction for the template.

Template pagelet example

Note. The option to use Pagelet Wizard pagelets published as homepage pagelets is available if you have licensed PeopleSoft Enterprise Portal or a Portal Pack.

See Also

Chapter 18, “Using Pagelet Wizard,” Step 6: Specifying Pagelet Publication Options, page 249

Enterprise PeopleTools 8.45 PeopleBook: Using PeopleSoft Applications, “Setting User Preferences,” Personalizing Your Homepage

PeopleSoft 8.8 Enterprise Portal PeopleBook, “Working With Intelligent Context Manager”

CHAPTER 17

Setting Up Pagelet Wizard Options

This chapter discusses how to:

- Define Pagelet Wizard transformation types, display formats, and data types.
- Define Pagelet Wizard Extensible Stylesheet Language (XSL) prototypes.
- Define Pagelet Wizard headers and footers.
- Define Pagelet Wizard pagelet categories.

Note. Examples of PeopleSoft Pure Internet Architecture pages in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page elements displayed remain the same.

Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types

This section discusses how to:

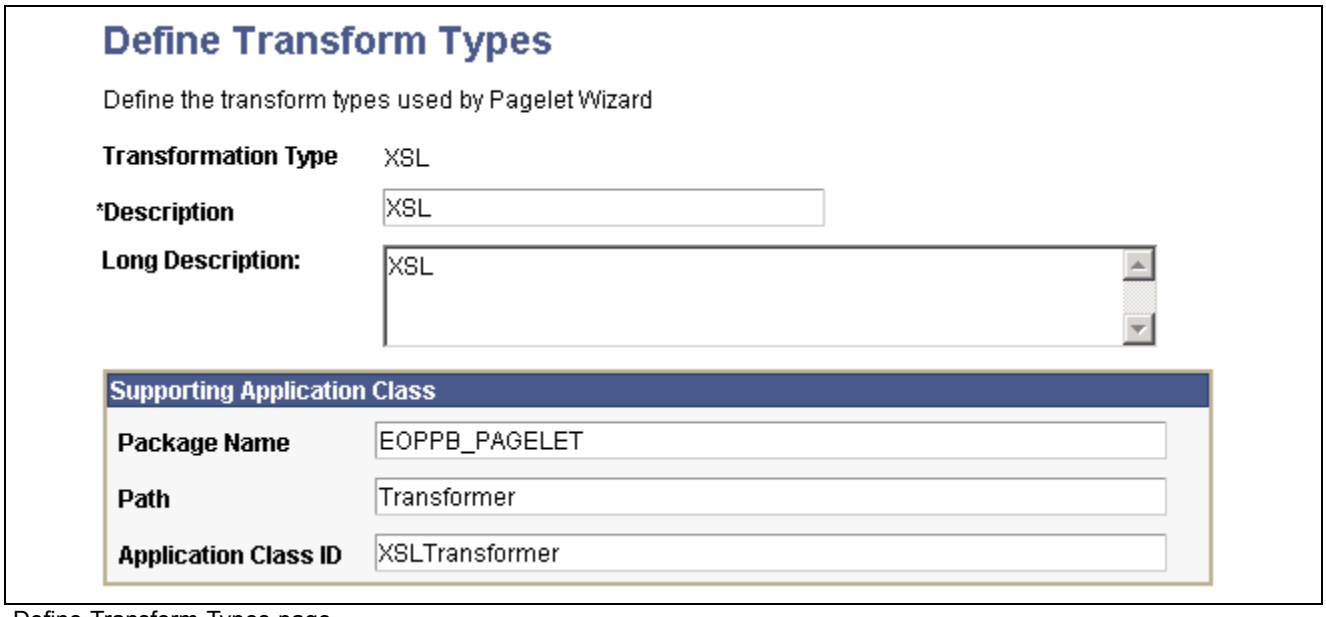
- Define transformation types.
- Define display formats.
- Define data types.

Pages Used to Define Pagelet Wizard Transformation Types, Display Formats, and Data Types

Page Name	Object Name	Navigation	Usage
Define Transform Types	EOPPB_TXFRMTYP_MTC	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Transform Types	Define transformation methods that can be applied to Pagelet Wizard pagelet content.
Define Display Formats	EOPPB_DISPFRMT_MTC	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Display Formats	Define display formats that can be used to format Pagelet Wizard pagelet content.
Define Data Types	EOPPB_DATATYPE_MTC	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Data Types	Define data types that can be used to supply Pagelet Wizard with pagelet content.
Document Type Definition (DTD)	EOPPB_DATADTD_SBP	Click the View Document Type Definition (DTD) link on the Define Data Types page.	Access the DTD that describes the XML generated by the selected data type.

Defining Transformation Types

Access the Define Transform Types page.



Define Transform Types page

Pagelet Wizard is delivered with the following transformation type definitions to support their associated display formats:

- PASSTHRU
- XSL

Warning! Do not alter delivered field values for these transformation types.

This Define Transform Types page enables you to build your own transformation types. For example, you could create a new transformation type called ABC123. You would then extend Pagelet Wizard to perform ABC123 transformations by building a supporting application class and registering it as a transformation type. You could then create builders for assembling pagelet displays that create ABC123 transformation definitions.

See Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Display Formats, page 189.

Transformation Type Enter the internal name that you want to assign to the transformation type.

Description For reference purposes only, enter a short description of the transformation type.

Long Description For reference purposes only, enter a more detailed description of the transformation type.

Supporting Application Class

The application class specified here refers to the PeopleCode that implements the transformation type.

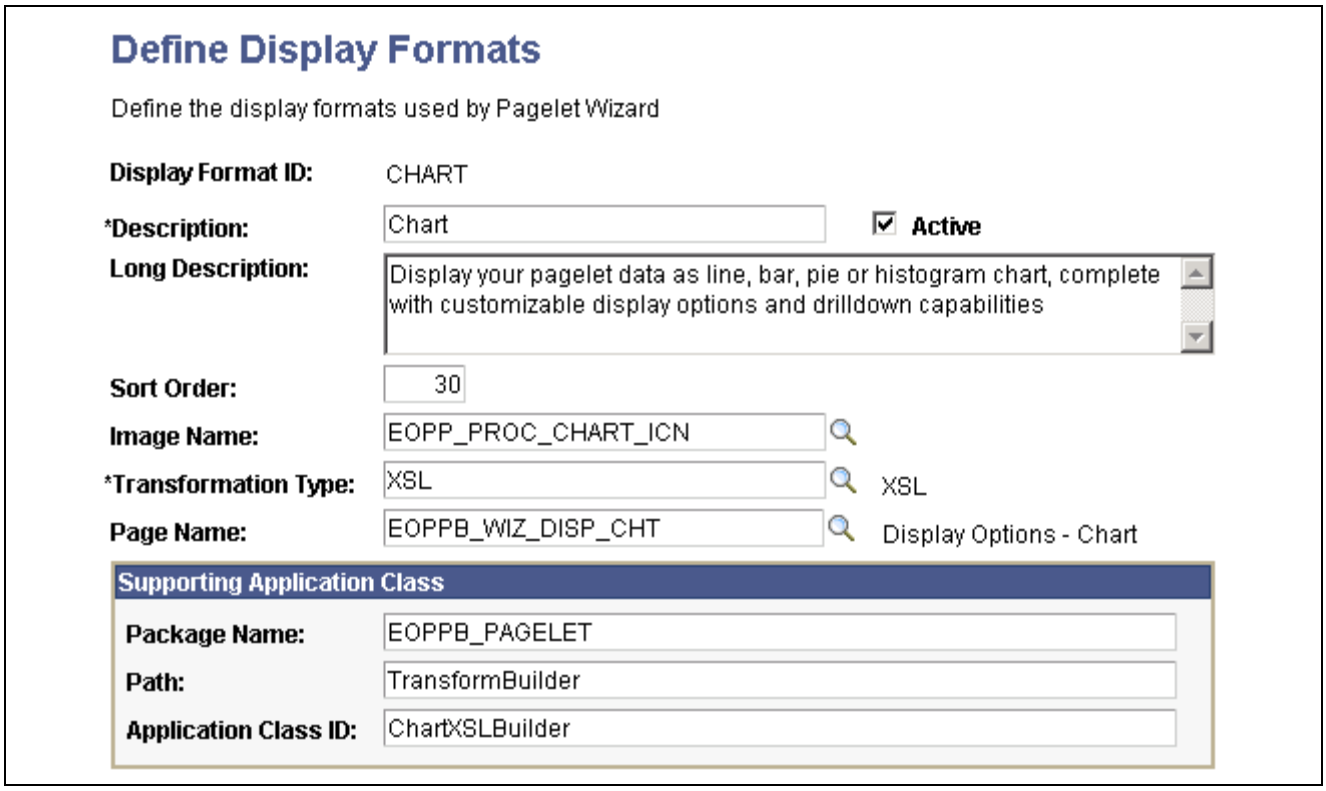
Package Name Enter the application class package name that you want to use for the transformation type.

Path Enter the application class path that you want to use for the transformation type.

Application Class ID Enter the application class ID that you want to use for the transformation type. The class must exist in the application package name that you have specified.

Defining Display Formats

Access the Define Display Formats page.



Define Display Formats page

A display format performs the following tasks:

- Defines a TransformBuilder, which is code that builds a particular type of display format (chart, menu, list, and so forth).
- Identifies the proper transformation to use with pagelets created with a specific display format builder.

For example, the chart display format defines the code that builds charts, and it specifies that Pagelet Wizard must use the XSL transformer for pagelets created using this display format.

Pagelet Wizard is delivered with the following display format definitions to support their associated display formats:

- CHART
- CUSTOM
- LIST
- MENU
- PASSTHRU
- SEARCHREC
- TABLE

Warning! Do not alter delivered field values for these display formats.

See Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Display Formats, page 189.

Display Format ID Enter the name that you want to assign to the display format.

Description Enter a short description of the display format. This short description appears as the display format name value on the Select Display Format page in Pagelet Wizard.

Active Select to activate the display format and make it a selectable value in Pagelet Wizard.

If you deactivate a display format, the display format is no longer available for selection on the Select Display Format page when accessed for associated data types.

Existing pagelets that have been created using a deactivated display format are unaffected; however, you can no longer use Pagelet Wizard to change the display settings for those pagelets.

Long Description Enter a detailed description of the display format. This long description appears as the display format description on the Select Display Format page in Pagelet Wizard.

Sort Order Enter a numeric value to specify the order in which you want the display format to appear as a selectable field value relative to other display formats on the Select Display Format page.

See Chapter 18, "Using Pagelet Wizard," Step 4: Selecting a Pagelet Display Format, page 231.

Image Name Select the image that you want to appear with the display format name and description on the Select Display Format page.

Transformation Type Display formats use a transformation technology. Select the transformation type that you want to use for the display format. Available transformation types are defined on the Define Transform Types page.

See Chapter 17, "Setting Up Pagelet Wizard Options," Defining Transformation Types, page 202.

Pagelet Wizard Page

Page Name Enter the object name of the page on which you want to be able to define display options for this display format.

Supporting Application Class

The application class specified here refers to the PeopleCode that implements the display format. This code is the builder that creates the definition that Pagelet Wizard uses (along with the transformer) to display the pagelet.

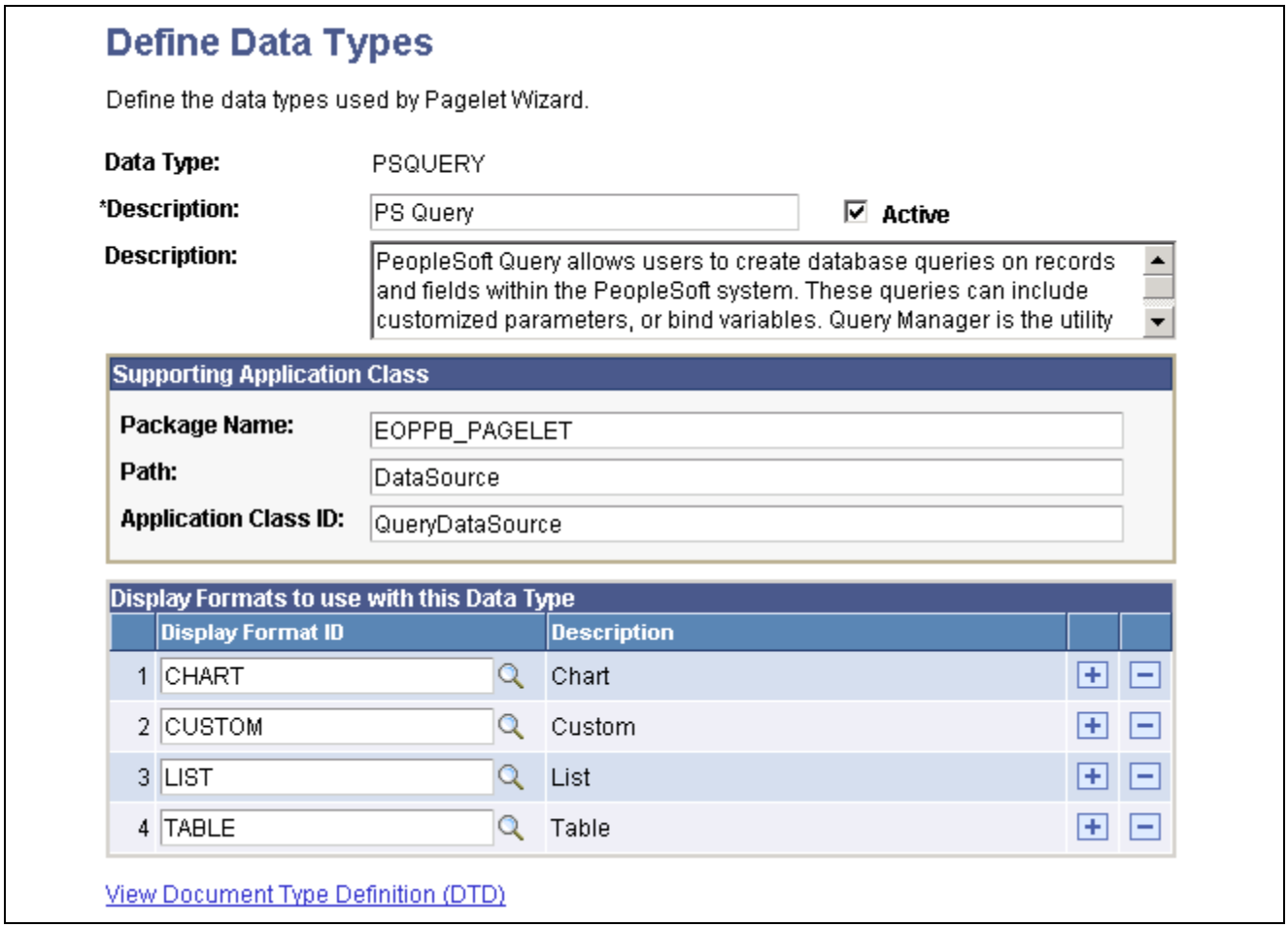
Package Name Enter the application class package name to use for the display format.

Path Enter the application class path to use for the display format.

Application Class ID Enter the application class ID to use for the display format. The class must exist in the application package name that you specify.

Defining Data Types

Access the Define Data Types page.



Define Data Types page

Pagelet Wizard is delivered with the following data type definitions to support their associated data sources:

- HTML
- NAVIGATION COLLECTION
- PSQUERY
- SEARCHREC

Warning! Do not alter delivered field values for these data types.

See Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Data Sources, page 188.

Data Type Enter the internal name that you want to assign to the data type.

Description Enter a short description of the data type. This short description appears as the prompt list value when you select data sources in Pagelet Wizard.

Active Select activate the data type and make it a selectable value in Pagelet Wizard.

If you deactivate a data type, the data type is no longer available for selection on the Select Data Source page.

If you deactivate a data type that is in use, associated pagelets do not display data. The pagelets display an error message instead.

Description Enter a detailed description of the data type. This long description appears as the data type description on the Select Data Source page in Pagelet Wizard.

Supporting Application Class

The application class specified here refers to the PeopleCode that implements the data type. This is the code that Pagelet Wizard calls to retrieve the data for the pagelet.

Package Name Enter the application class package name that you want to use for the data type.

Path Enter the application class path that you want to use for the data type.

Application Class ID Enter the application class ID that you want to use for the data type. The class must exist in the application package name that you specify.

Display Formats to use with this Data Type

Display Format ID Enter the display formats that are supported by the data type. You can select from existing display formats defined on the Define Display Formats page.

See Chapter 17, "Setting Up Pagelet Wizard Options," Defining Display Formats, page 203.

View Document Type Definition (DTD) Click to access the Document Type Definition (DTD) page, where you can access the DTD that describes the XML generated by the selected data type.

Defining Pagelet Wizard XSL Prototypes

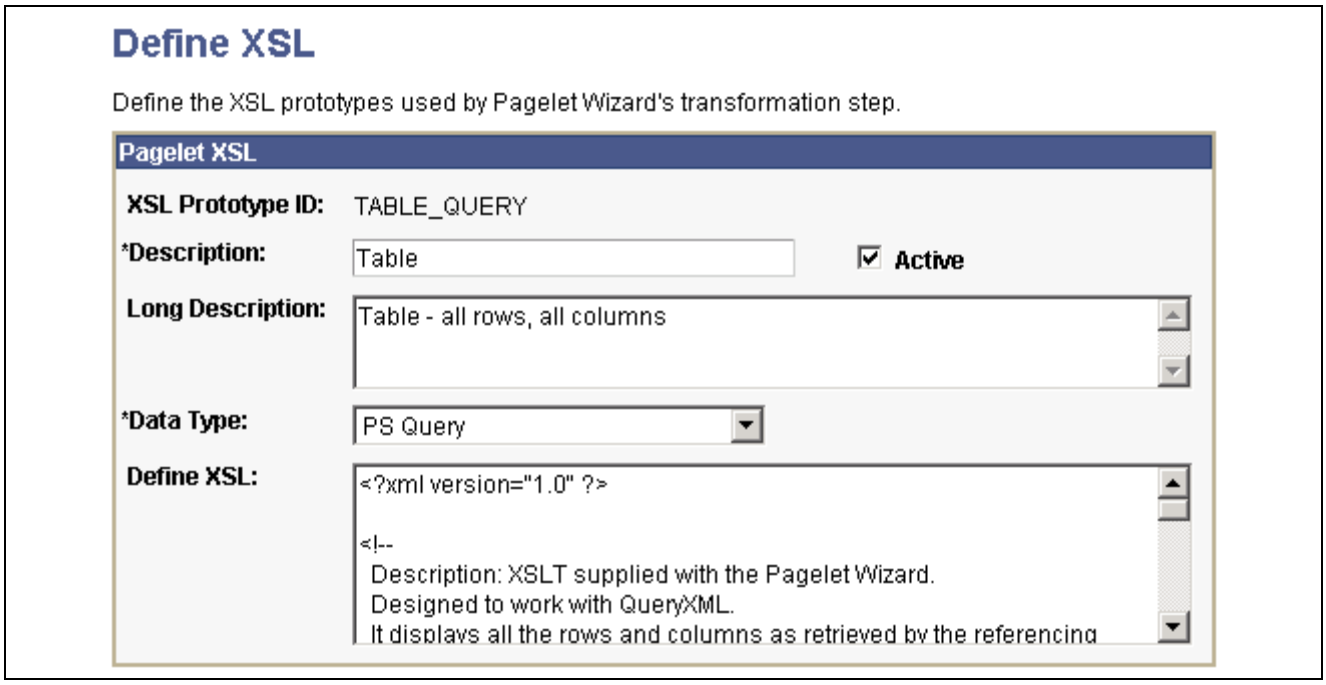
This section discusses how to define XSL prototypes for use in building Pagelet Wizard pagelets.

Page Used to Define Pagelet Wizard XSL Prototypes

Page Name	Object Name	Navigation	Usage
Define XSL	EOPPB_XSLPROT_MTCE	Enterprise Components, Portal Utilities, Pagelet Wizard, Define XSL	Create custom XSL definitions for use with Pagelet Wizard.

Defining XSL Prototypes

Access the Define XSL page.



Define XSL page

Pagelet Wizard is delivered with the following XSL prototypes to support their associated data sources and display formats:

- LIST_QUERY
- NUMBERED_LIST_QUERY
- TABLE_QUERY

Warning! Do not alter the delivered XSL for these XSL prototypes.

XSL Prototype ID Enter an ID for the XSL prototype.

Description Enter a description of the XSL prototype. The text that you enter here appears as available XSL template field values on the Specify Display Options page when you create a custom display format pagelet.

See Chapter 18, "Using Pagelet Wizard," Specifying Custom Display Options, page 246.

Active Select to activate the XSL prototype and make it an available XSL template field value on the Specify Display Options page for custom display format pagelets.

If you deactivate an XSL prototype, the prototype is no longer available for selection on the Specify Display Options page for the custom display format. Existing pagelets that have been created using the deactivated XSL prototype are unaffected.

Long Description For reference purposes only, enter a long description of the XSL prototype.

Data Type Select the applicable data type for the XSL prototype. You can assign only one data type per XSL prototype.

Define XSL Enter XSL code for formatting pagelet XML data in the desired pagelet format.

Defining Pagelet Wizard Headers and Footers

This section discusses how to:

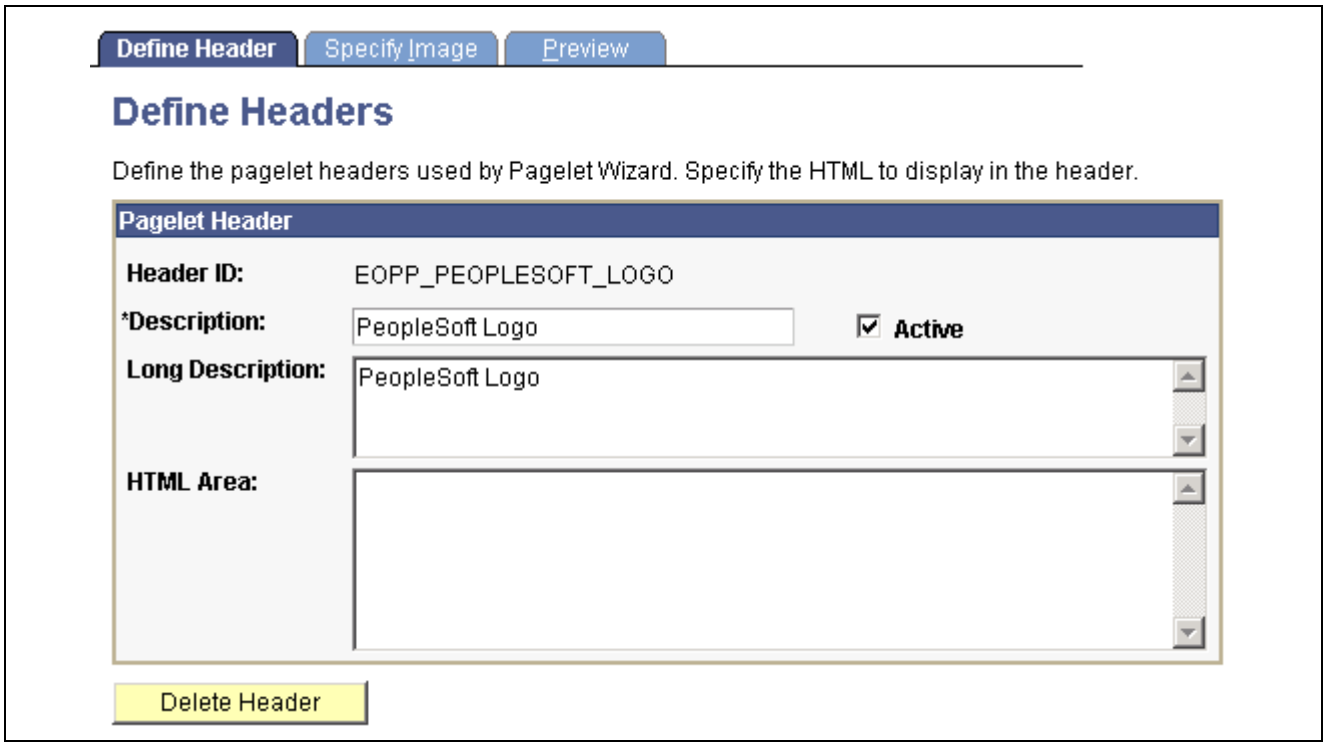
- Define headers.
- Define header images.
- Define footers.
- Define footer images.

Pages Used to Define Pagelet Wizard Headers and Footers

Page Name	Object Name	Navigation	Usage
Define Header	EOPPB_HDR_MTCE	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Headers, Define Header	Create a pagelet header design. Enter HTML that you want to appear on the header.
Define Headers - Specify Image	EOPPB_HDR_IMAGE	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Headers, Define Header, Specify Image	Specify an image that you want to appear on the pagelet header. Specify image properties.
Define Headers - Preview	EOPPB_HDR_PREVIEW	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Headers, Define Header, Preview	View the fully assembled pagelet header. After previewing the pagelet header, you can adjust the HTML and image parameters as needed.
Define Footer	EOPPB_FTR_MTCE	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Footers, Define Footer	Create a pagelet footer design. Enter HTML that you want to appear on the footer.
Define Footers - Specify Image	EOPPB_FTR_IMAGE	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Footers, Specify Image	Specify an image that you want to appear on the pagelet footer. Specify image properties.
Define Footers - Preview	EOPPB_FTR_PREVIEW	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Footers, Preview	View the fully assembled pagelet footer. After previewing the pagelet footer, you can adjust the HTML and image parameters as needed.

Defining Headers

Access the Define Header page.



Define Header page

Pagelet Header

Header ID Enter an ID for the pagelet header.

Description Enter a description of the pagelet header. The text that you enter here appears as the selectable field value in the Header field when users assign headers to pagelets on the Specify Display Options page.

Active Select to activate the pagelet header and make it a selectable field value in the Header field on the Specify Display Options page.

If you deactivate a header that is in use, the header no longer appears on associated pagelets. In addition, the header is not be available for selection on the Specify Display Options page.

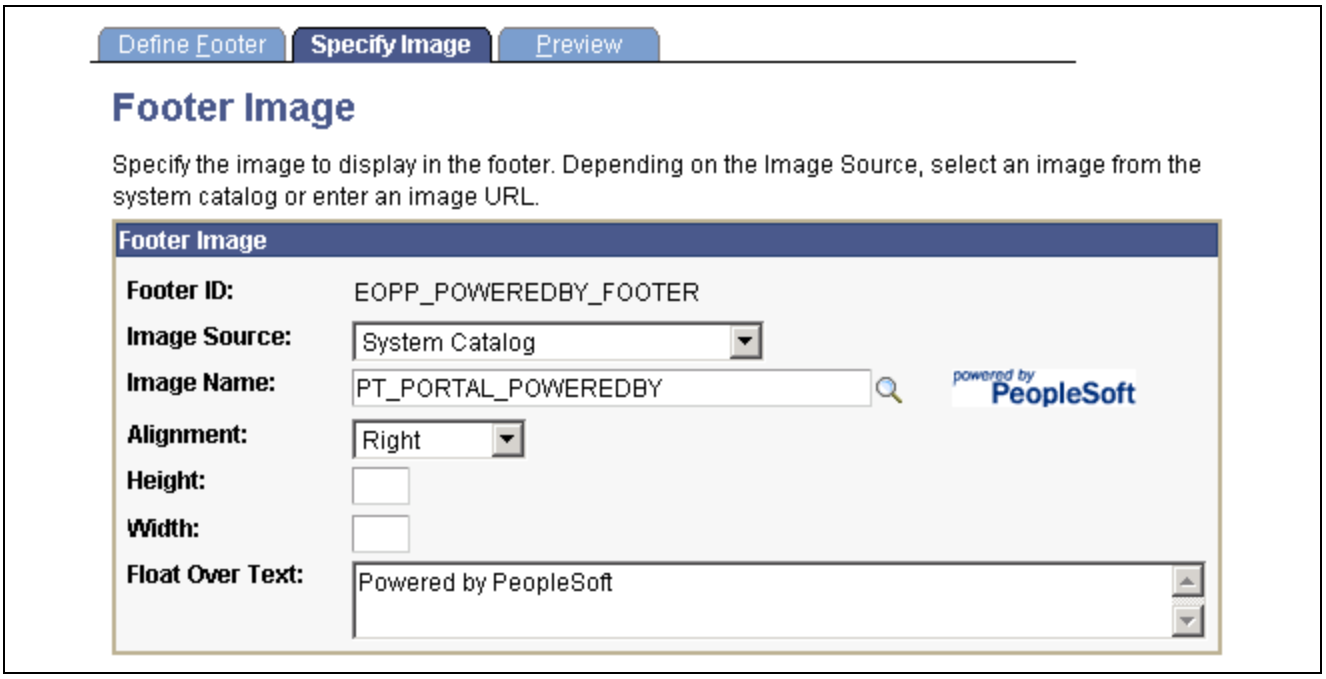
Long Description For reference purposes only, enter a long description of the pagelet header.

HTML Area Enter the HTML code that you want to use to generate the pagelet header. You can enter both HTML text and image references. Alternatively, you can include an image by selecting it on the Specify Image page.

Delete Header Click to delete the pagelet header design.

Defining Header Images

Access the Define Headers - Specify Image page.



Define Headers - Specify Image page

Header Image

Image Source Select the image source. Options are:

System Catalog. Select to specify an image from the PeopleTools system image catalog.

External Source. Select to specify an image from a source external to the PeopleTools system image catalog.

Image Name If you select the *System Catalog* image source, click the Search button to access a list of existing images.

If you select the *External Source* image source, enter a URL for the desired image file. All image types are supported. Ensure that the location of the image provides reliable access.

Alignment Select the alignment of the image. Options are:

Left: Select to align the image with the left margin of the pagelet.

Right: Select to align the image with the right margin of the pagelet.

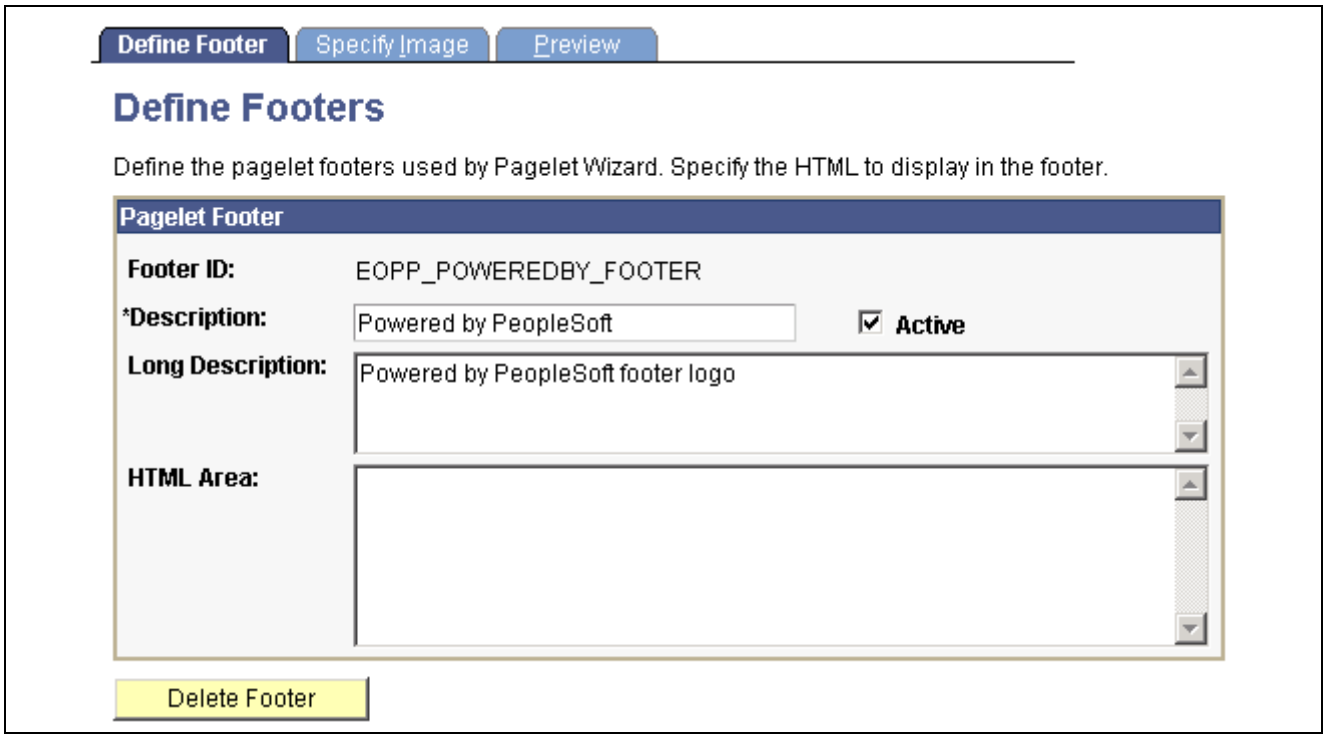
Height Enter the image height in pixels if the image needs to be resized. If you enter an image height, the system adjusts the image width to fit the image height.

Width Enter the image width in pixels if the image needs to be resized. If you enter an image width, the system adjusts the image height to fit the image width.

Float Over Text Enter text that you want to appear when the user moves the cursor over the image.

Defining Footers

Access the Define Footer page.



Define Footer page

Pagelet Footer

Footer ID Enter an ID for the pagelet footer.

Description Enter a description of the pagelet footer. The text that you enter here appears as the selectable field value in the Footer field when users assign footers to pagelets on the Specify Display Options page.

Active Select to activate the pagelet footer and make it a selectable field value in the Footer field on the Specify Display Options page.

If you deactivate a footer that is in use, the footer no longer appears on associated pagelets. In addition, the footer is not be available for selection on the Specify Display Options page.

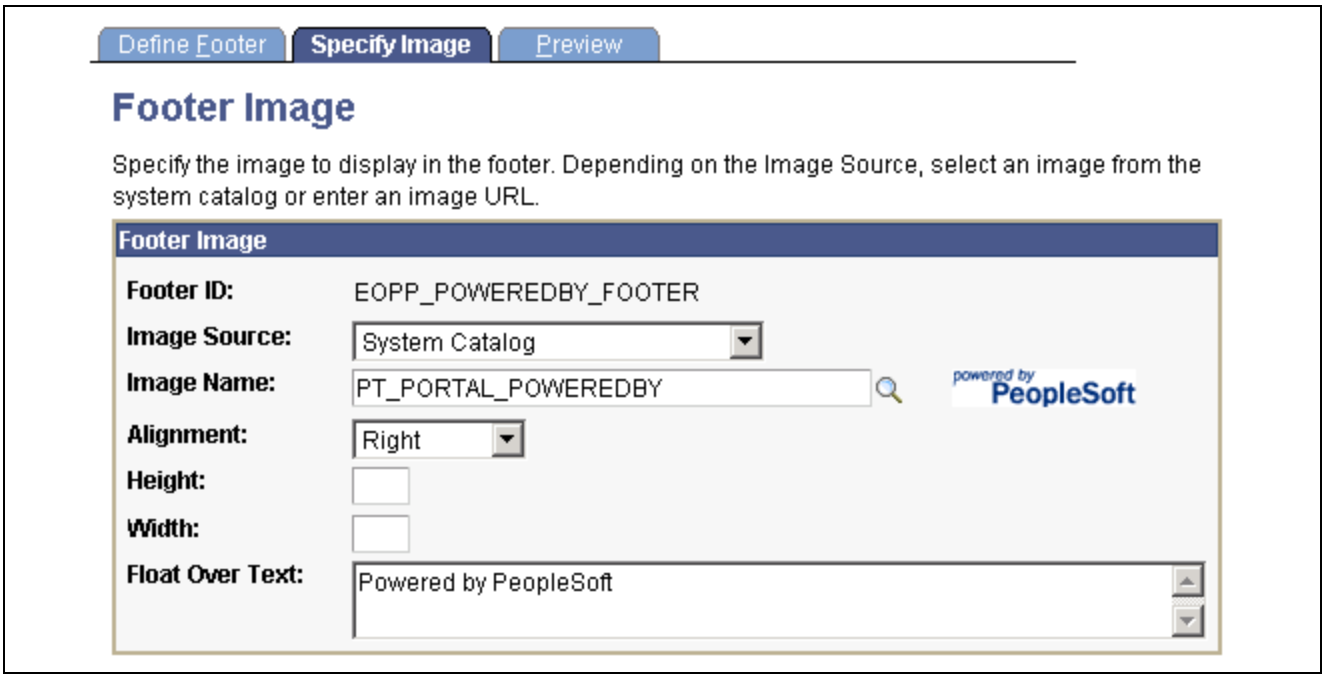
Long Description For reference purposes only, enter a long description of the pagelet footer.

HTML Area Enter that HTML code that you want to use to generate the pagelet footer. You can enter both HTML text and image references. Alternatively, you can include an image by selecting it on the Specify Image page.

Delete Footer Click to delete the pagelet footer design.

Defining Footer Images

Access the Define Footers - Specify Image page.



Define Footers - Specify Image page

Footer Image

Image Source Select the image source. Options are:

System Catalog: Select to specify an image from the PeopleTools system image catalog.

External Source: Select to specify an image from a source external to the PeopleTools system image catalog.

Image Name If you select *System Catalog* as the image source, click the Search button to access a list of existing images.

If you select *External Source* as the image source, enter a URL for the desired image file. All image types are supported. Ensure that the location of the image provides reliable access.

Alignment Select the alignment of the image. Options are:

Left: Select to align the image with the left margin of the pagelet.

Right: Select to align the image with the right margin of the pagelet.

Height Enter the image height in pixels if the image needs to be resized. If you enter an image height, the system adjusts the image width to fit the image height.

Width Enter the image width in pixels if the image needs to be resized. If you enter an image width, the system adjusts the image height to fit the image width.

Float Over Text Enter text that you want to appear when the user moves the cursor over the image.

Defining Pagelet Wizard Pagelet Categories

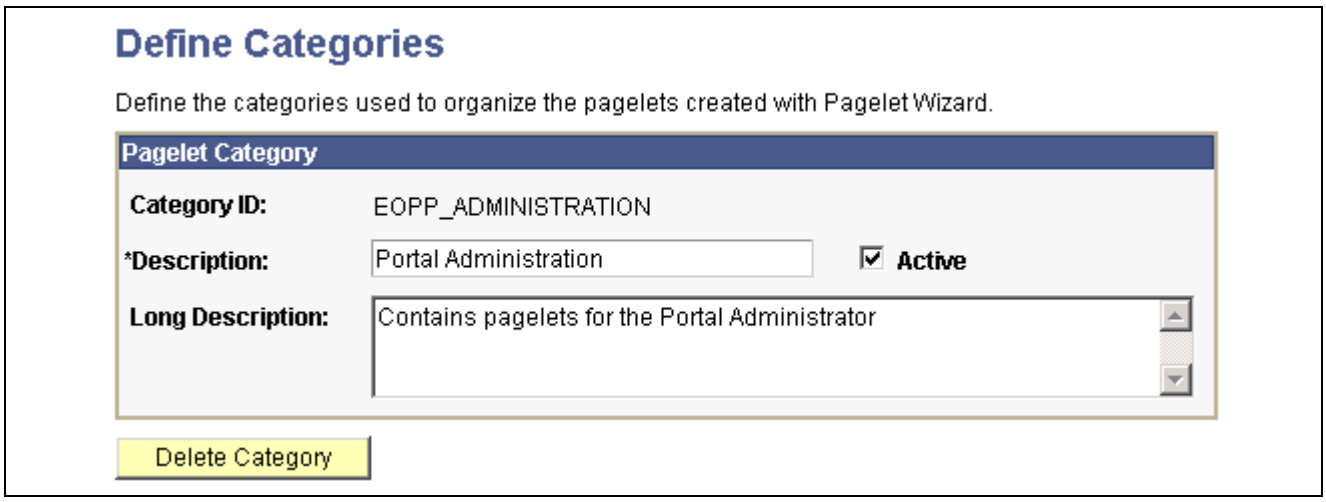
This section discusses how to define Pagelet Wizard pagelet categories.

Page Used to Define Pagelet Wizard Pagelet Categories

Page Name	Object Name	Navigation	Usage
Define Categories	EOPPB_PGLTCATG_MTC	Enterprise Components, Portal Utilities, Pagelet Wizard, Define Categories	Define categories used to organize pagelets created using Pagelet Wizard.

Defining Pagelet Categories

Access the Define Categories page.



Define Categories page

Category ID Enter a Pagelet Wizard pagelet category ID. You can use category IDs to organize Pagelet Wizard pagelets. The portal administrator can use this value when running pagelet reports and when searching for pagelets that need to be moved between different portal sites.

Categories provide another level of organization in addition to the owner ID. For example, you can use a departmental owner ID, and within that, categorize your pagelets by pagelet type.

Description Enter a description of the pagelet category. The text that you enter here appears as a field value in the Category ID field when users assign pagelets to categories on the Specify Pagelet Information page.

Active Select to activate the pagelet category and make it a selectable field value in the Category ID field on the Specify Pagelet Information page.

If you deactivate a category, the category is no longer available for selection on the Specify Pagelet Information page. Existing pagelets that have been created using the deactivated category are unaffected.

Long Description For reference purposes only, enter a long description of the pagelet category.

Delete Category Click to delete a pagelet category. You are prompted to confirm the deletion.

If you delete a category, the category is no longer available for selection on the Specify Pagelet Information page. Existing pagelets that have been created using the deactivated category still appear, but the pagelet category is removed from the pagelet definition in Pagelet Wizard.

CHAPTER 18

Using Pagelet Wizard

This chapter discusses how to:

1. Enter pagelet identifying information.
2. Select a pagelet data source.
3. Specify pagelet data source parameters.
4. Select a pagelet display format.
5. Specify pagelet display options.
6. Specify pagelet publication options.

Note. Before you start building a pagelet with Pagelet Wizard, familiarize yourself with all data sources. This enables you to evaluate the best way to retrieve the data that you want to display in the pagelet.

Examples of PeopleSoft Pure Internet Architecture pages in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page elements displayed remain the same.

See Also

Chapter 17, "Setting Up Pagelet Wizard Options," page 201

Step 1: Entering Pagelet Identifying Information

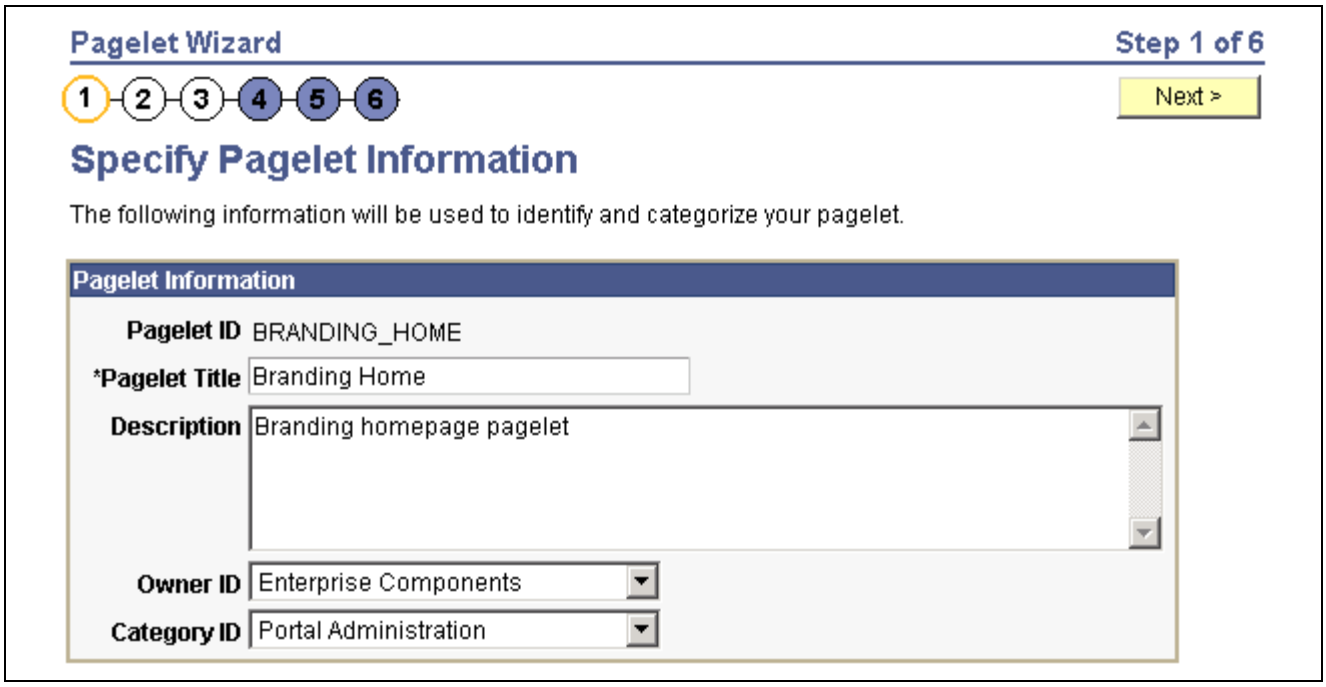
This section discusses how to specify pagelet information.

Page Used to Enter Pagelet Identifying Information

Page Name	Object Name	Navigation	Usage
Specify Pagelet Information	EOPPB_WIZ_INFO	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard	Define description and summary information for a pagelet.

Specifying Pagelet Information

Access the Specify Pagelet Information page.



Specify Pagelet Information page

Pagelet ID Enter an ID for the pagelet. Each pagelet that you build with Pagelet Wizard must have a unique ID.

Pagelet Title Enter a unique title for the pagelet. This value appears in the pagelet title bar, which is the solid bar at the top of the pagelet. If you change the pagelet title value for a published pagelet, you must unpublish and republish the pagelet for the change to take effect.

Description Enter additional descriptive information to identify the pagelet. This reference text is used in Verity and search record searches.

Owner ID Select the value that you want to use to identify pagelets created by the organization. The portal administrator can use this value when searching for pagelets 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 the organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer.

Category ID To organize Pagelet Wizard pagelets, select a category assignment for the pagelet. You create pagelet categories on the Define Categories page. The portal administrator can use this value when running pagelet reports and when searching for pagelets that need to be moved between different portal sites.

Categories provide another level of organization in addition to the owner ID. For example, you can use a departmental owner ID, and within that, categorize pagelets for that ID by pagelet type.

See Chapter 17, “Setting Up Pagelet Wizard Options,” Defining Pagelet Wizard Pagelet Categories, page 214.

Step 2: Selecting a Pagelet Data Source

This section discusses how to:

- Select a PeopleSoft Query data source.
- Select an HTML data source.
- Select a search record data source.
- Select a Navigation Collection data source.

You can define other data sources by using the Define Data Types page.

See Also

Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Data Sources, page 188

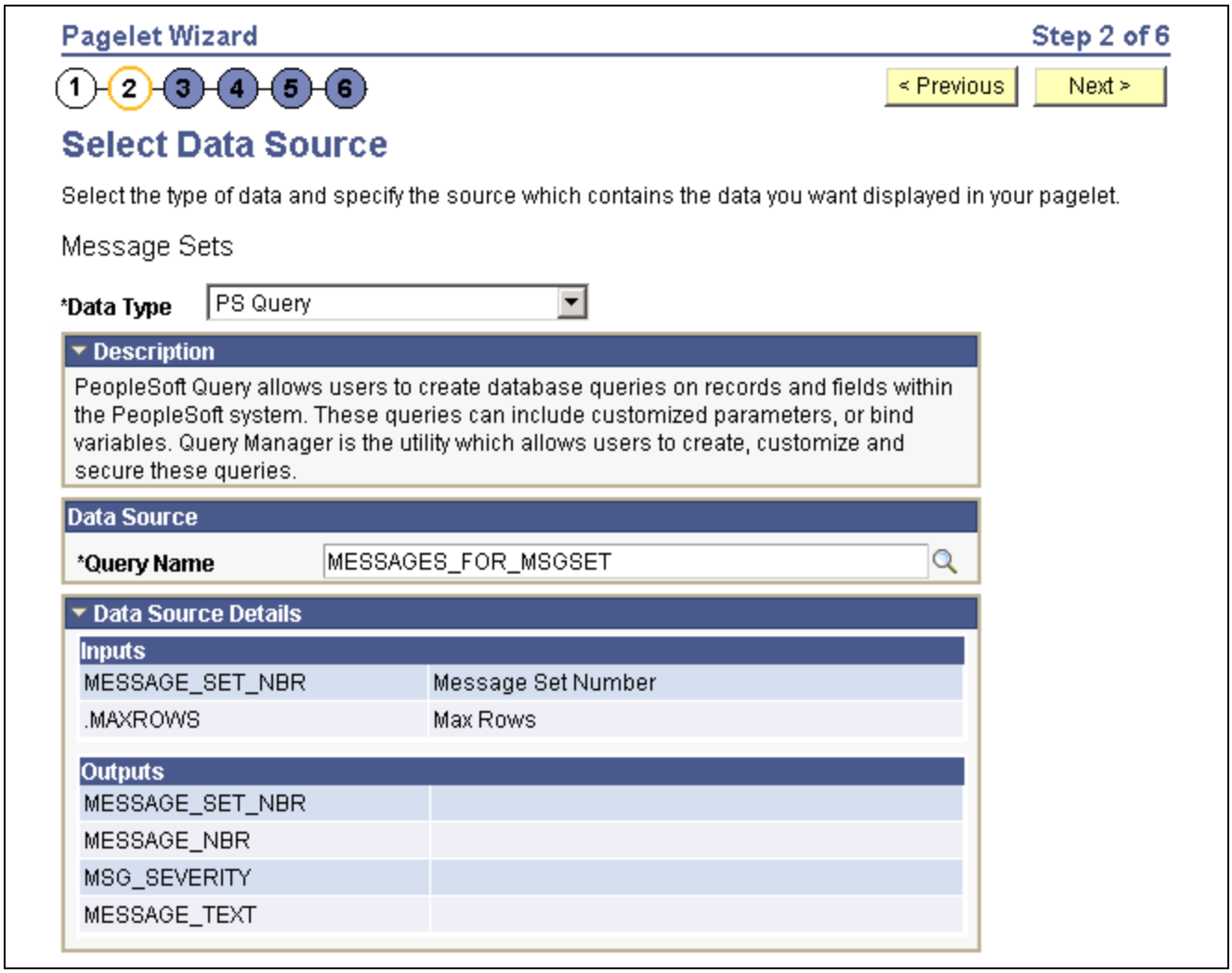
Chapter 17, "Setting Up Pagelet Wizard Options," Defining Data Types, page 205

Page Used to Select a Pagelet Data Source

Page Name	Object Name	Navigation	Usage
Select Data Source	EOPPB_WIZ_DATASRC	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Click the Next button or the icon for step 2.	Select the type of data source for the pagelet.

Selecting a Query Data Source

Access the Select Data Source page. Select *PS Query* in the Data Type field.



Select Data Source page (query data source)

Queries typically return a number of columns and rows. Queries for pagelets should be selected based on available display space on the homepage. In addition, avoid selecting long-running queries that can delay the display of the homepage.

Keep in mind that public query results are available to portal viewers only if they have appropriate access to the underlying data record. Permission lists in the users' role definitions must contain query access groups that include the records in the query definition.

Data Type Select *PS Query.*

Data Source

Query Name Select a query to serve as the data source for the pagelet. You can select any public query in the local database.

PeopleSoft queries provide SQL access to any table recognized by the PeopleSoft system. This includes external, remote tables that are recognized by the PeopleSoft system by way of a database link.

See *Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Query,* "Introduction to PeopleSoft Query"

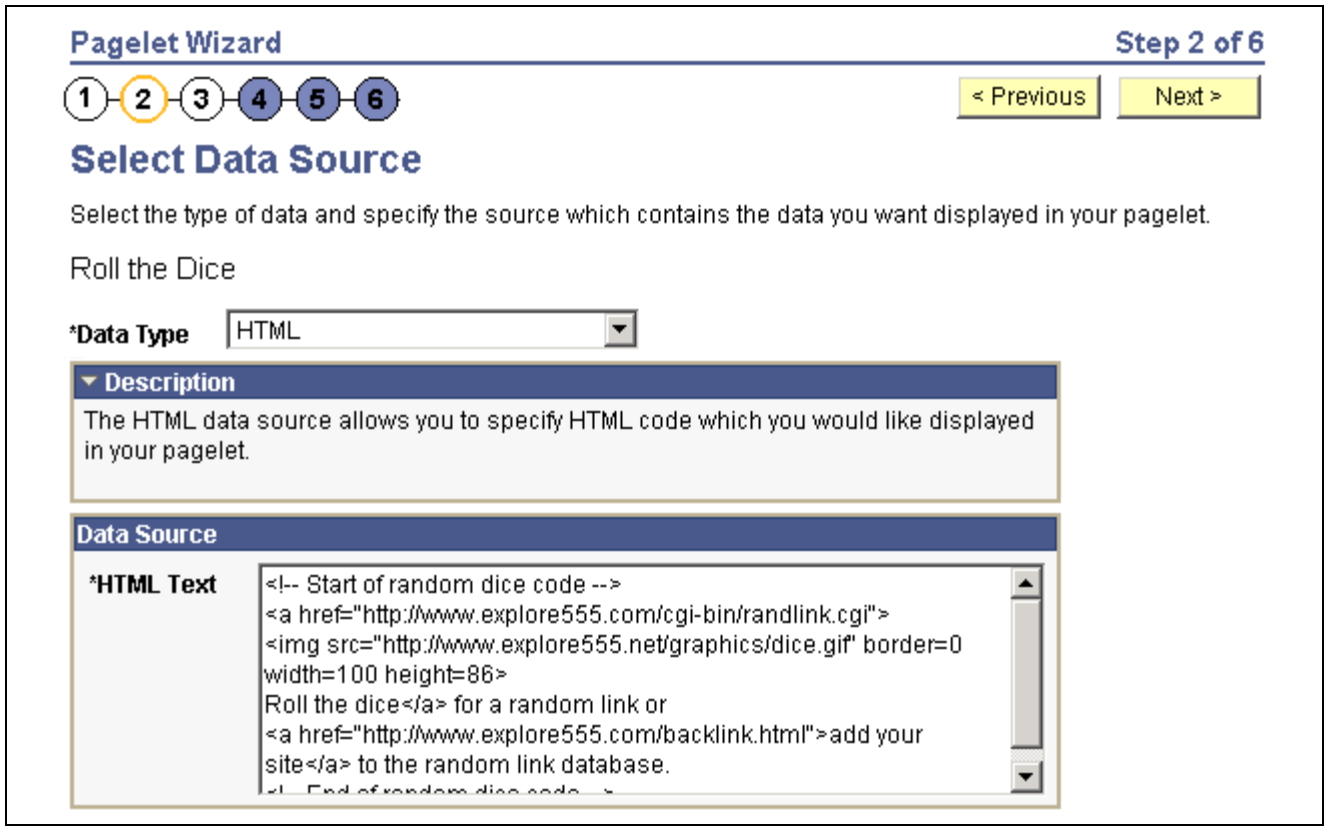
Data Source Details

Inputs Displays the data source parameter fields that correspond to the query's input prompts. Also displays the required system-supplied .MAXROWS query data source parameter that you can use to limit the amount of data returned to the pagelet.

Outputs Displays the data source parameter fields that correspond to the query's output.

Selecting an HTML Data Source

Access the Select Data Source page. Select *HTML* in the Data Type field.



Select Data Source page (HTML data source)

When you use HTML as the data source for a pagelet, the pagelet does not provide configurable parameters that enable criteria selection. Consequently, there is also no end user personalization for HTML data source pagelets.

When creating an HTML data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

Data Source

HTML Text Enter the HTML code that you want to use as the data source for the pagelet. You can enter an HTML document or fragment. This HTML can include embedded JavaScript.

Ensure that the HTML that you enter generates content that is appropriately sized for pagelet display.

See Chapter 17, "Setting Up Pagelet Wizard Options," Defining Data Types, page 205.

Selecting a Search Record Data Source

Access the Select Data Source page. Select *Search Record* in the Data Type field.

Select Data Source page (search record data source)

Search records typically return a number of columns and rows. When you select search records to be displayed on pagelets, consider the available display space on the homepage.

Data Type Select *Search Record.*

Data Source

Portal Name Select the portal in which the search record for the pagelet resides.

Content Reference Select the content reference for the search record.

Access Type Select the type of access for the search record data. Options are:

Basic Search Record: Select to have the search record pagelet retrieve data based on the basic search record defined in the selected portal.

CI: <component interface name>: Select when the basic search record does not provide the required functionality; for example, when security or PeopleCode considerations need to be taken into account This option is available if the selected search record has an associated component interface. Select to have the search record pagelet retrieve data by way of the component interface.

See *Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Component Interfaces,* "Developing the Component Interface"

Data Source Details

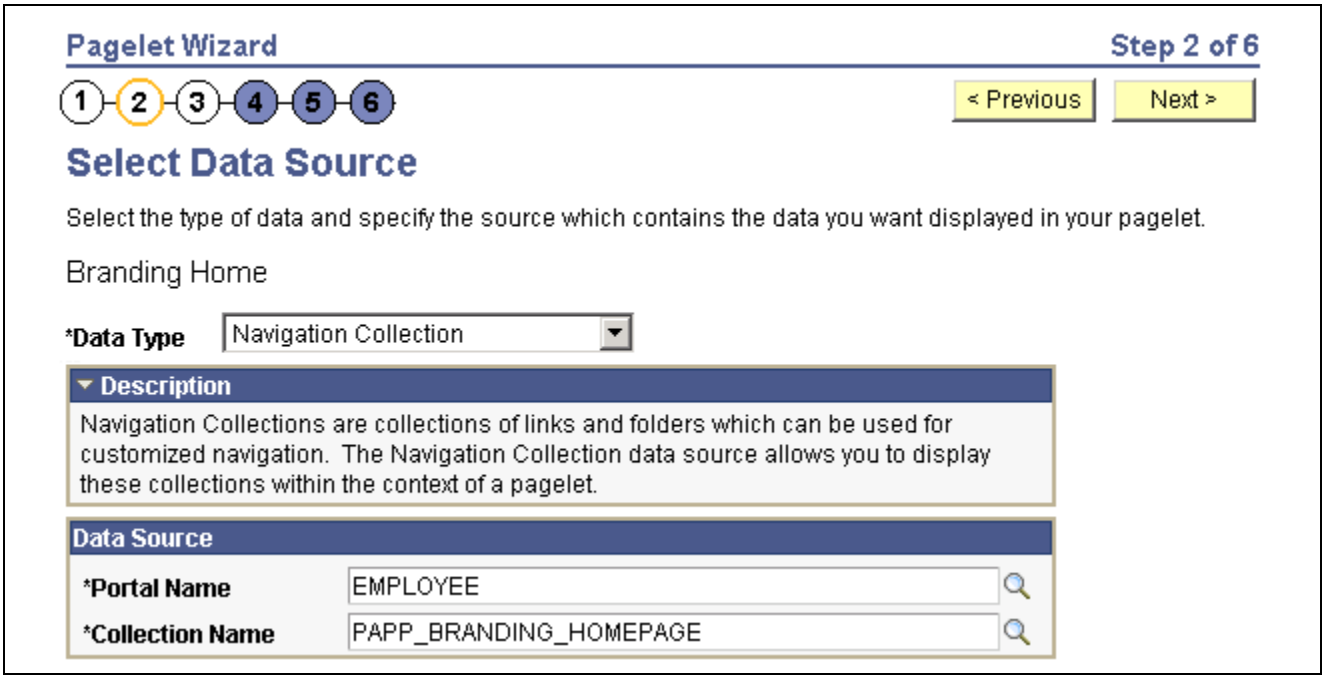
Inputs Displays search record key input fields, as well as other system-required input fields that are used to derive data that is displayed on the search record pagelet.

Outputs Displays search record key output fields, as well as other output fields for which search results are displayed on the search record pagelet. These output fields are derived from the drop-down list box fields on the search record.

If the SHOWRESULTS data source parameter field for the pagelet is set to *N,* the search key output fields appear as Search By drop-down list values on the pagelet.

Selecting a Navigation Collection Data Source

Access the Select Data Source page. Select *Navigation Collection* in the Data Type field.



Select Data Source page (Navigation Collection data source)

The Navigation Collection that you use as the data source must have already been created using the Navigation Collections feature. However, the Navigation Collection does not need to exist in its final form. Any changes that you make to the underlying Navigation Collection used as the data source are immediately reflected on any pagelets created from that Navigation Collection.

See Chapter 23, "Using Navigation Collections," page 291.

Note. Although it is possible to create a Navigation Collection pagelet directly from the Navigation Collections feature, pagelets created using this feature cannot be maintained in Pagelet Wizard. Only Navigation Collection pagelets created specifically with Pagelet Wizard can be maintained in Pagelet Wizard.

Data Type Select *Navigation Collection.*

Data Source

Portal Name Select the portal that contains the navigation collection for the pagelet.

Collection Name Select the name of the Navigation Collection that you want to use to create the pagelet.

Step 3: Specifying Pagelet Data Source Parameters

This section provides an overview of variables supported as data source parameters, lists common elements, and discusses how to:

- Specify query data source parameters.
- Specify search record data source parameters.
- Specify Navigation Collection data source parameters.
- Specify data source parameter values.

Understanding System Variables Supported as Data Source Parameters

When you set the Usage Type field to *System Variable* or *User Specified* on the Specify Data Source Parameters page, you can enter a system variable such as *%UserId,* which the system substitutes at runtime with the actual value represented by the variable. The following system variables are supported as data source parameters.

- %AuthenticationToken
- %ClientDate
- %ClientTimeZone
- %ContentID
- %ContentType
- %Copyright
- %Currency
- %Date

- %DateTime
- %DbName
- %DbServerName
- %DbType
- %EmailAddress
- %EmployeeId
- %HPTabName
- %IsMultiLanguageEnabled
- %Language
- %Language_Base
- %LocalNode
- %Market%MaxMessageSize
- %NavigatorHomePermissionList
- %Node
- %PasswordExpired
- %Portal
- %PrimaryPermissionList
- %ProcessProfilePermissionList
- %RowSecurityPermissionList
- %RunningInPortal
- %ServerTimeZone
- %Time
- %UserDescription
- %UserId

Common Elements Used in This Section

Description Displays a description of the data source parameter.

Usage Type Select the type of end-user accessibility that you want to grant for the data source parameter field when it appears on the pagelet. Options are:

- *Fixed:* Select to enter a fixed value for the data source parameter that the pagelet end user cannot modify.
- *System Variable:* Select to assign a system variable as the data source parameter value. The value of the system variable is automatically inserted into the parameter when the pagelet appears. When you select this option, the pagelet end user cannot modify the data source parameter.

For example, suppose that you specify *%UserId* as the system variable for a User data source parameter field. When the pagelet appears on a user's homepage, the User field is populated by the *%UserId* system variable, which is the user ID used to access the pagelet.

- *User Specified:* Select to enable end users to specify a data source parameter value for this field. When a pagelet contains a user-specified field, the Customize button appears on the pagelet title bar.

  End users can click this button to access a personalization page, where they can select a data source parameter value that they want to appear by default on the pagelet when they access it. They can select a value from a prompt, or they can manually enter their own value if no prompt values are available.

  If you change the usage type from or to *User Specified* for a data source parameter on a published homepage pagelet, you must unpublish and republish the pagelet.

Default Value

If you select *User Specified* as the usage type and the data source parameter is a required value, you must enter the default value that should appear before a user enters a value. If the data source parameter is not a required value, you do not have to enter a default value.

If you select a default value when defining available user-specified values on the Specify Data Source Parameter Values page, that default value populates this field.

If you select *User Specified* as the usage type, you can enter a system variable as the default value. For example, to make the current date the default value for a user-specified parameter, enter a default value of *%Date*.

If you select *System Variable* as the usage type, you must enter the system variable to use as the data source parameter value.

See Chapter 18, "Using Pagelet Wizard," Understanding System Variables Supported as Data Source Parameters, page 224.

If you select *Fixed* as the usage type, you must enter the fixed value.

You can enter a value that includes the % and * wildcards at the beginning or end of a value in the Default Value field.

If you select *Fixed* as the usage type and the data source parameter has associated prompt values, you can use the Look up Value button to search for available data source parameter values.

If you select *System Variable* as the usage type, you can use the Look up Value button to access a list of valid system variables.

If you select *User Specified* as the usage type, this button does not appear.

User Values

If you select *User Specified* as the usage type, click to access the Specify Data Source Parameter Values page.

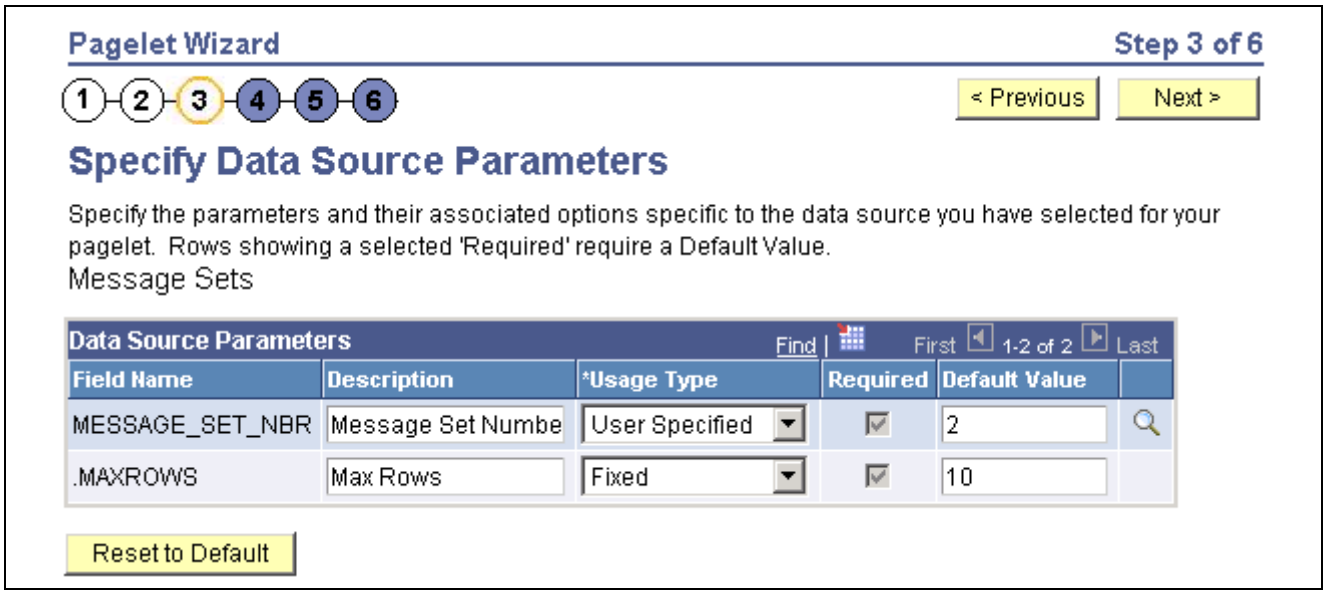
See Chapter 18, "Using Pagelet Wizard," Specifying Data Source Parameter Values, page 231.

Page Used to Specify Pagelet Data Source Parameters

Page Name	Object Name	Navigation	Usage
Specify Data Source Parameters	EOPPB_WIZ_DATAPRMS	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Click the Next button on the Select Data Source page, or click the icon for step 3.	Configure the data source parameters required for the data to be displayed on the pagelet. For instance, when entering data source parameters for a query pagelet, you specify input data that extracts output data to the pagelet. The parameters vary depending on the data source selected on the Select Data Source page. This page is accessible only when building a pagelet with a data source that allows modification of associated parameters. These data sources include PeopleSoft Query, search record, and Navigation Collections. **Note.** This step is not applicable when creating pagelets with an HTML data source.
Specify Data Source Parameter Values	EOPPB_WIZ_PRMVALS	Click the User Values button on the Specify Data Source Parameters page.	Enter values for the end user to select from when personalizing user-specific parameters on a pagelet. **Note.** This page is accessible only for data source parameters for which you have selected the *User Specified* usage type.

Specifying Query Data Source Parameters

Access the Specify Data Source Parameters page after selecting *PS Query* as the data type on the Select Data Source page.



Specify Data Source Parameters page (PeopleSoft Query data source)

Data Source Parameters

Pagelet Wizard analyzes the selected query and retrieves a list of predefined parameters and any related prompt list values. You cannot add to the list of parameters.

Field Name Displays the data source parameters that correspond to the query prompts associated with the selected query. Also displays the required system-supplied .MAXROWS query data source parameter that you can use to limit the amount of data returned on the pagelet.

Required This option is selected if the data source parameter must have a value entered on the pagelet to retrieve data successfully. For example, the .MAXROWS field is required.

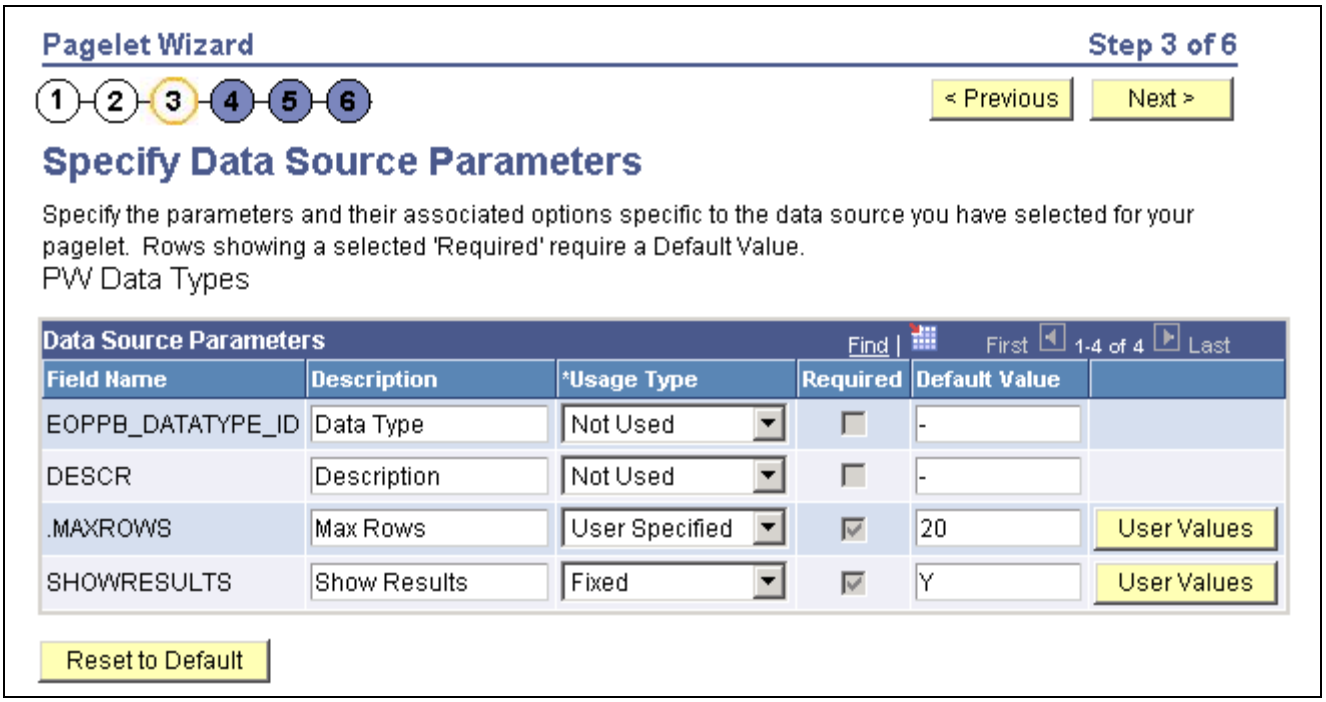
If this option is selected and display-only, the value is set in the application class at the application programming interface (API) level.

If you leave this option cleared, the data source parameter value can be clear at runtime, or it can be entered to refine the data selection criteria.

Reset to Default Click to reset the values on this page to the default values specified by the query data source API.

Specifying Search Record Data Source Parameters

Access the Specify Data Source Parameters page after selecting *Search Record* as the data type on the Select Data Source page.



Specify Data Source Parameters (search record data source)

Data Source Parameters

Pagelet Wizard analyzes the selected search record and retrieves a list of search keys required for the search record. You cannot add to the list of data source parameters.

For fields that are search keys but that are not displayed on the search prompt page or in the result set for the search record (for example, the setID, Business Unit, Operator ID, and Operclass fields), Pagelet Wizard assimilates the PeopleTools behavior and does not display the keys during design or at runtime. It simply binds in the same values that PeopleTools would use. These values are derived from the default field values established by the user in the PSRECDEFN table.

Field Name

Displays the data source parameters that correspond to the search keys required for the search record. Also displays the required system-supplied .MAXROWS and SHOWRESULTS search record data source parameters.

Use the .MAXROWS field to limit the amount of data returned to the pagelet.

Use the SHOWRESULTS field to determine whether search results appear on the pagelet.

If this field value is set to *Y,* search record results appear directly on the pagelet with links to transaction data.

In addition, if this field value is set to *Y,* the search results grid displays results based on the supplied data source parameter default key field values. If no defaults are supplied, all results appear, up to the .MAXROWS value defined for the pagelet.

If this field value is set to *N,* the pagelet displays a search prompt for the selected search record. The Search By drop-down list values available on the pagelet are the search keys for the search record.

If data source parameter default key field values have been defined, when the pagelet is initially accessed, the default value appears in the begins with pagelet field.

Required This option is selected if the data source parameter must have a value entered on the pagelet to retrieve data successfully. For example, the .MAXROWS and SHOWRESULTS fields are required.

If this option is selected and display-only, the value is set in the application class at the API level.

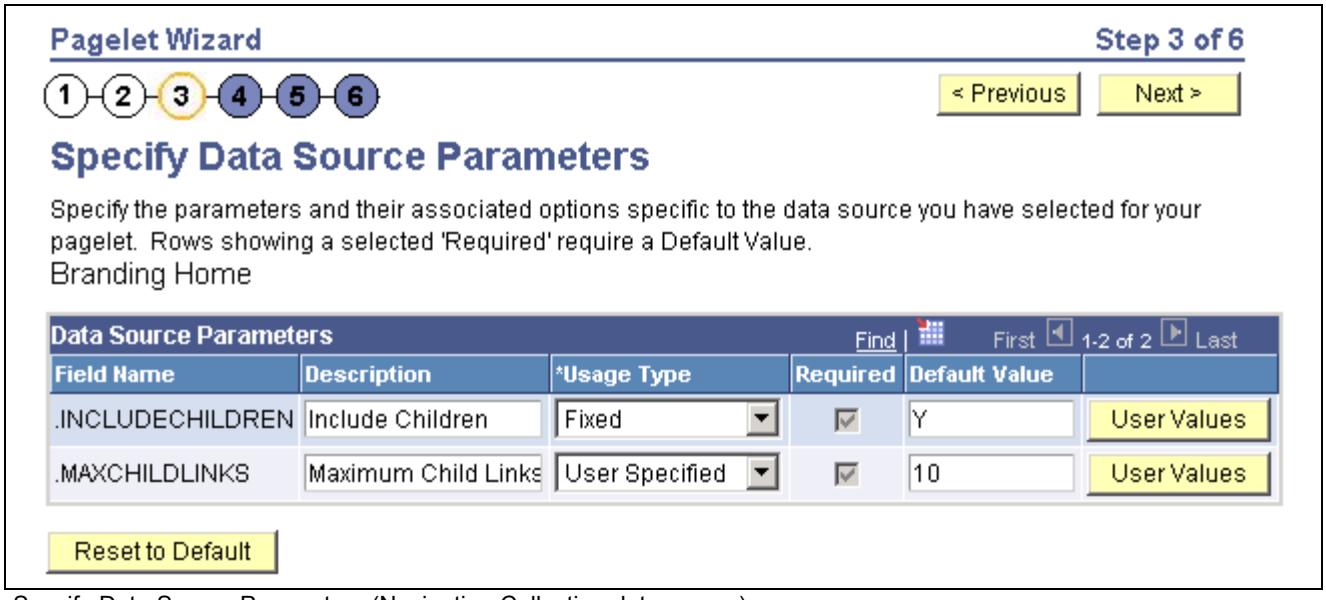
Search key values are not required because the search record pagelet supports partially populated keys so that you can define the pagelet to return all possible results.

If you leave this option clear, the data source parameter value can be blank at runtime, or the value can be provided to refine the data selection criteria.

Reset to Default Click to reset the values on this page to the default values specified by the search record data source API.

Specifying Navigation Collection Data Source Parameters

Access the Specify Data Source Parameters page after selecting *Navigation Collection* as the data type on the Select Data Source page.



Specify Data Source Parameters (Navigation Collection data source)

Data Source Parameters

Pagelet Wizard analyzes the selected query and retrieves a list of predefined parameters and any related prompt list values. You cannot add to the list of parameters.

Field Name Displays the following data source parameters required by the Navigation Collection data source.

.INCLUDECHILDREN: Determines whether the pagelet displays Navigation Collection child links.

.MAXCHILDLINKS: If the pagelet is defined to display child links, determines the maximum number of Navigation Collection child links to display.

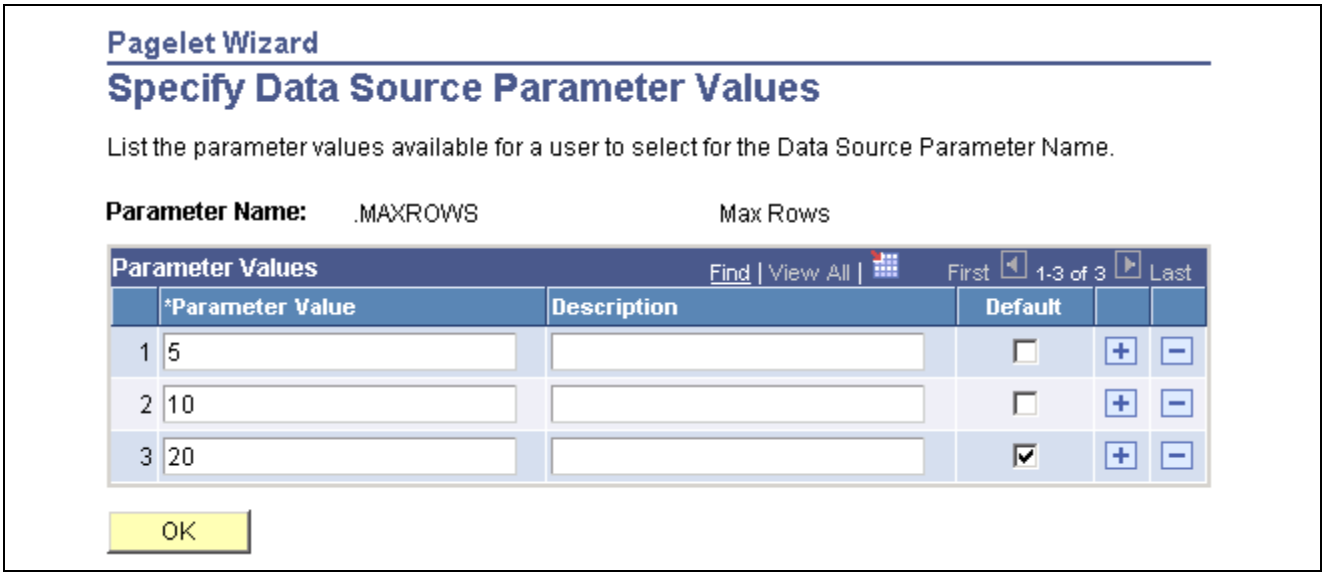
Required This option is selected if the data source parameter must contain a value on the pagelet to retrieve data successfully. For example, the .INCLUDECHILDREN field is required.

If this option is selected and display-only, the value is set in the application class at the API level.

Reset to Default Click to reset the values on this page to the default values specified by the Navigation Collection data source API.

Specifying Data Source Parameter Values

Access the Specify Data Source Parameter Values page.



Specify Data Source Parameter Values page

Parameter Name Displays the data source parameter for which you selected *User Specified* as the usage type on the Specify Data Source Parameters page.

Parameter Value Enter values from which you want pagelet users to be able to select when personalizing user-specifiable parameters on a pagelet.

Default Selecting a default value overrides the default value you defined for the data source parameter on the Specify Data Source Parameters page.

Step 4: Selecting a Pagelet Display Format

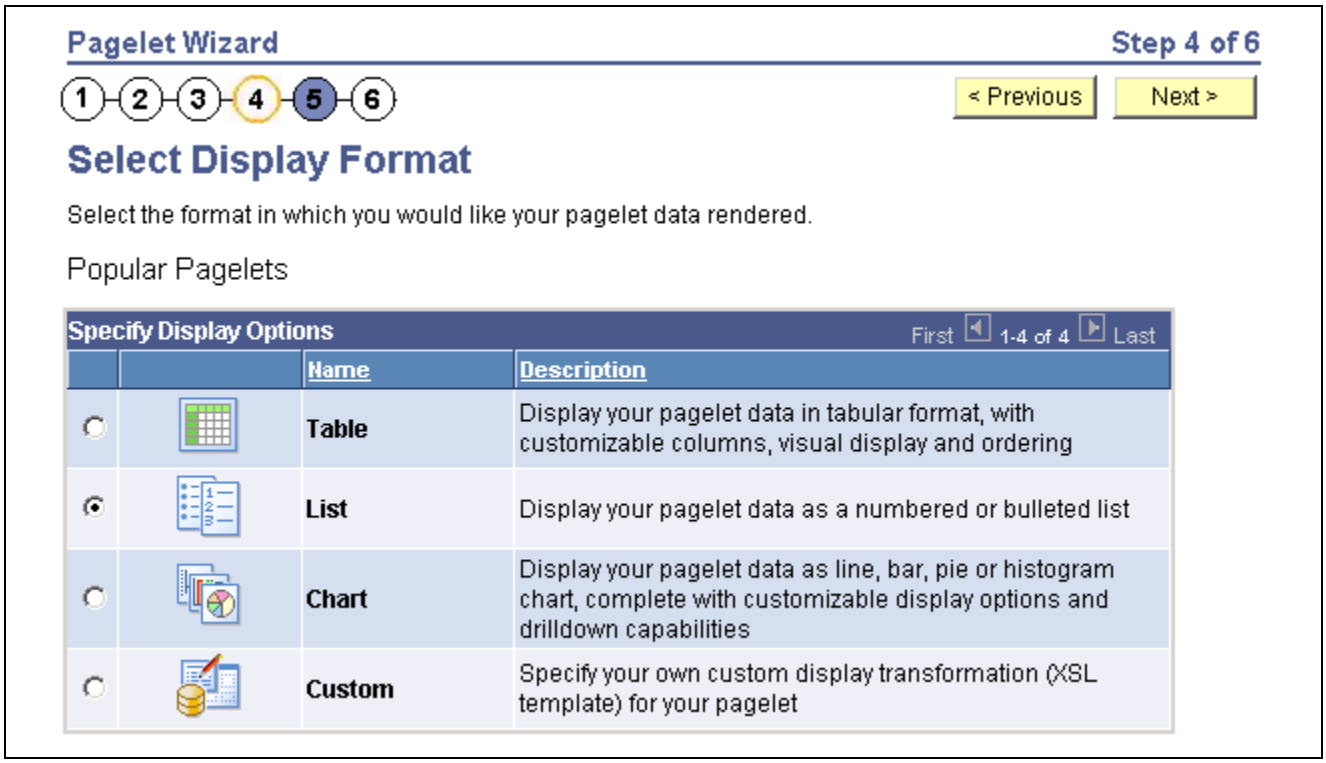
This section discusses how to select a display format.

Page Used to Select a Pagelet Display Format

Page Name	Object Name	Navigation	Usage
Select Display Format	EOPPB_WIZ_DISPFRMT	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Click the Next button on the Specify Data Source Parameters page, or click the icon for step 4.	Specify the data transformation method and display format for the pagelet.

Selecting a Display Format

Access the Select Display Format page.



Select Display Format page

Select the format to use for the pagelet. The display formats available for selection on this page depend on the data source you have selected on the Select Data Source page. Options are:

Table Available for query data source pagelets.

List Available for query data source pagelets.

Chart Available for query data source pagelets.

Menu Available for Navigation Collection data source pagelets.

Search List Available for search record data source pagelets.

Passthru Available for HTML and query data source pagelets.

Custom Available for HTML, Navigation Collection, query, and search record data source pagelets.

See Also

Chapter 18, "Using Pagelet Wizard," Step 5: Specifying Pagelet Display Options, page 233

Step 5: Specifying Pagelet Display Options

This section lists common elements and discusses how to:

- Specify table display options.
- Specify passthru display options.
- Specify search list display options.
- Specify menu display options.
- Specify list display options.
- Specify chart display options.
- Specify custom display options.
- Specify data link details.

Common Elements Used in This Section

This section discusses common elements used in this section.

Click the List Field Link Options button to access the Link Details page, where you can enter values that link elements in the pagelet to pages (components or URLs) that provide details about the data.

See Chapter 18, "Using Pagelet Wizard," Specifying Data Link Details, page 248.

Additional Text

The Additional Text group box contains options to add headers, footers, opening text, and closing text to a pagelet.

Header Select a pagelet header. You can select from headers that are defined on the Define Header page.

See Chapter 17, "Setting Up Pagelet Wizard Options," Defining Headers, page 209.

Opening Text Select the type of opening text that you want to appear on the pagelet. Options are:

Freeform Text/HTML: The Text field appears. Enter the opening text that appears on the pagelet.

Message Catalog Entry: The Message Set and Number fields appear. Select the message set and message number that appear as opening text on the pagelet. You can select from the messages defined in the database.

See *Enterprise PeopleTools 8.45 PeopleBook: System and Server Administration,* "Using PeopleTools Utilities," Message Catalog

Show "View Source Data" Link Displays when *PS Query* is selected as the data source for a pagelet. Select to display the Full Query Results link on the pagelet. Click the Full Query Results link to access the full results of the query used for the pagelet.

Closing Text Select the type of closing text that appears on the pagelet. Options are:

Freeform Text/HTML: The Text field appears. Enter the closing text that appears on the pagelet.

Message Catalog Entry: The Message Set and Number fields appear. Select the message set and message number that appear as closing text on the pagelet. You can select from the messages defined in the database.

See *Enterprise PeopleTools 8.45 PeopleBook: System and Server Administration,* "Using PeopleTools Utilities," Message Catalog

Footer Select a pagelet footer. You can select from footers that are defined on the Define Footer page.

See Chapter 17, "Setting Up Pagelet Wizard Options," Defining Footers, page 211.

Customization

The Customization group box contains the option to customize your pagelet.

Customize Click to customize the pagelet by applying a custom XSL template, or by directly editing the XSL resulting from the pagelet display options you have defined so far. When you click Customize, you switch from a non-custom display format selected on the Select Display Format page, to a custom format of your definition.

The XSL field is populated based upon preexisting display options set for the original non-custom display format. You can edit the XSL directly in the XSL field and retain as much of the original non-custom display options as you wish. Keep in mind that if you apply an XSL template using the XSL Template field, the XSL is regenerated based on the selected template and original non-custom display options you have set may be overridden. Making any edits to the XSL, either directly or by assigning a template, disconnects the pagelet definition from the original non-custom display format.

When you click the Customize button, the current pagelet definition is saved for restoration purposes. You can revert to this pre-customization pagelet definition by clicking the Undo Customizations button.

Clicking this button assigns your pagelet to the Custom design format on the Select Display Format page - Step 4. When you return to the Specify Publishing Options page - Step 5, you will access the version of the page for the Custom display format.

See Chapter 18, "Using Pagelet Wizard," Specifying Custom Display Options, page 246.

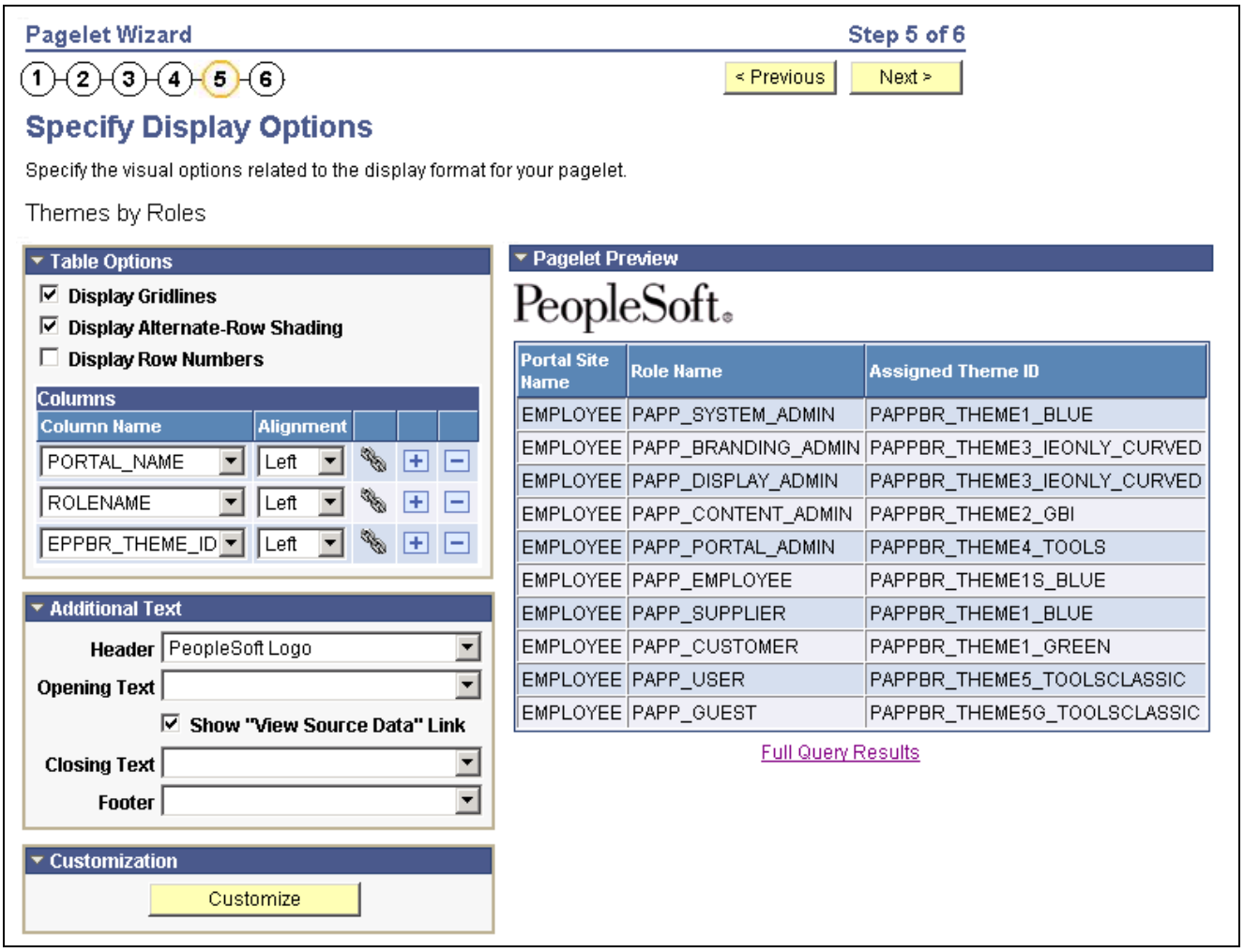
Pages Used to Specify Pagelet Display Options

Page Name	Object Name	Navigation	Usage
Specify Display Options (table format)	EOPPB_WIZ_DISP_TBL	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Select the *Table* as the display format and click the Next button on the Select Display Format page, or click the icon for the step 5.	Enter pagelet table formatting details and header and footer options. Preview the pagelet.
Specify Display Options (passthru format)	EOPPB_WIZ_DISP_PST	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Select the *Passthru* as the display format and click the Next button on the Select Display Format page, or click the icon for step 5.	Enter pagelet passthru formatting details and header and footer options. Preview the pagelet.
Specify Display Options (search list format)	EOPPB_WIZ_DISP_SRC	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Select the *Search List* as the display format and click the Next button on the Select Display Format page, or click the icon for step 5.	Enter pagelet search list formatting details and header and footer options. Preview the pagelet.
Specify Display Options (menu format)	EOPPB_WIZ_DISP_NAV	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Select *Menu* as the display format and click the Next button on the Select Display Format page, or click the icon for step 5.	Enter pagelet menu formatting details and header and footer options. Preview the pagelet.
Specify Display Options (list format)	EOPPB_WIZ_DISP_LST	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Select *List* as the display format and click the Next button on the Select Display Format page, or click the icon for step 5.	Enter pagelet list formatting details and header and footer options. Preview the pagelet.

Page Name	Object Name	Navigation	Usage
Specify Display Options (chart format)	EOPPB_WIZ_DISP_CHT	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Select *Chart* as the display format and click the Next button on the Select Display Format page, or click the icon for step 5.	Enter pagelet chart formatting details and header and footer options. Preview the pagelet.
Specify Display Options (custom format)	EOPPB_WIZ_DISP_CUS	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Select *Custom* as the display format and click the Next button on the Select Display Format page, or click the icon for step 5.	Enter pagelet custom formatting details and header and footer options. Preview the pagelet.
Link Details	EOPPB_WIZ_LINK_DTL	Click the Link Options button on the Specify Display Options page.	Enter linking details that convert pagelet column data into drill-down links to pages that provide details about the pagelet data.

Specifying Table Display Options

Access the Specify Display Options page after selecting the *Table* option on the Select Display Format page.



Specify Display Options page (table display format)

Table Options

Select options that will display grid lines, alternate row shading, and row numbers on the table-formatted pagelet.

Columns

Column Name Displays possible column headings provided by the selected data source parameters. Use the drop-down list boxes to set the order in which the columns appear on the table-formatted pagelet. Use the Add button or the Delete button to add or remove columns.

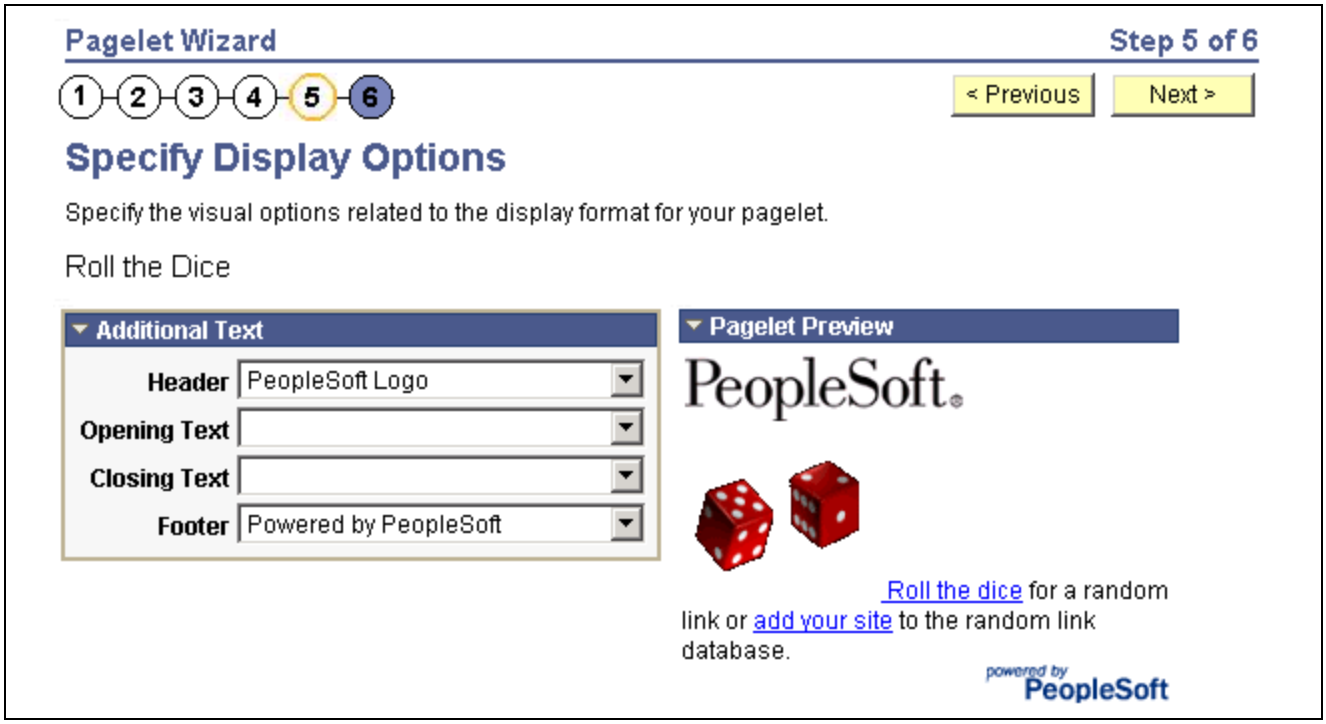
Alignment Specify the alignment of the text in the column. Options are *Center, Left,* and *Right.*

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Passthru Display Options

Access the Specify Display Options page after selecting the *Passthru* option on the Select Display Format page.



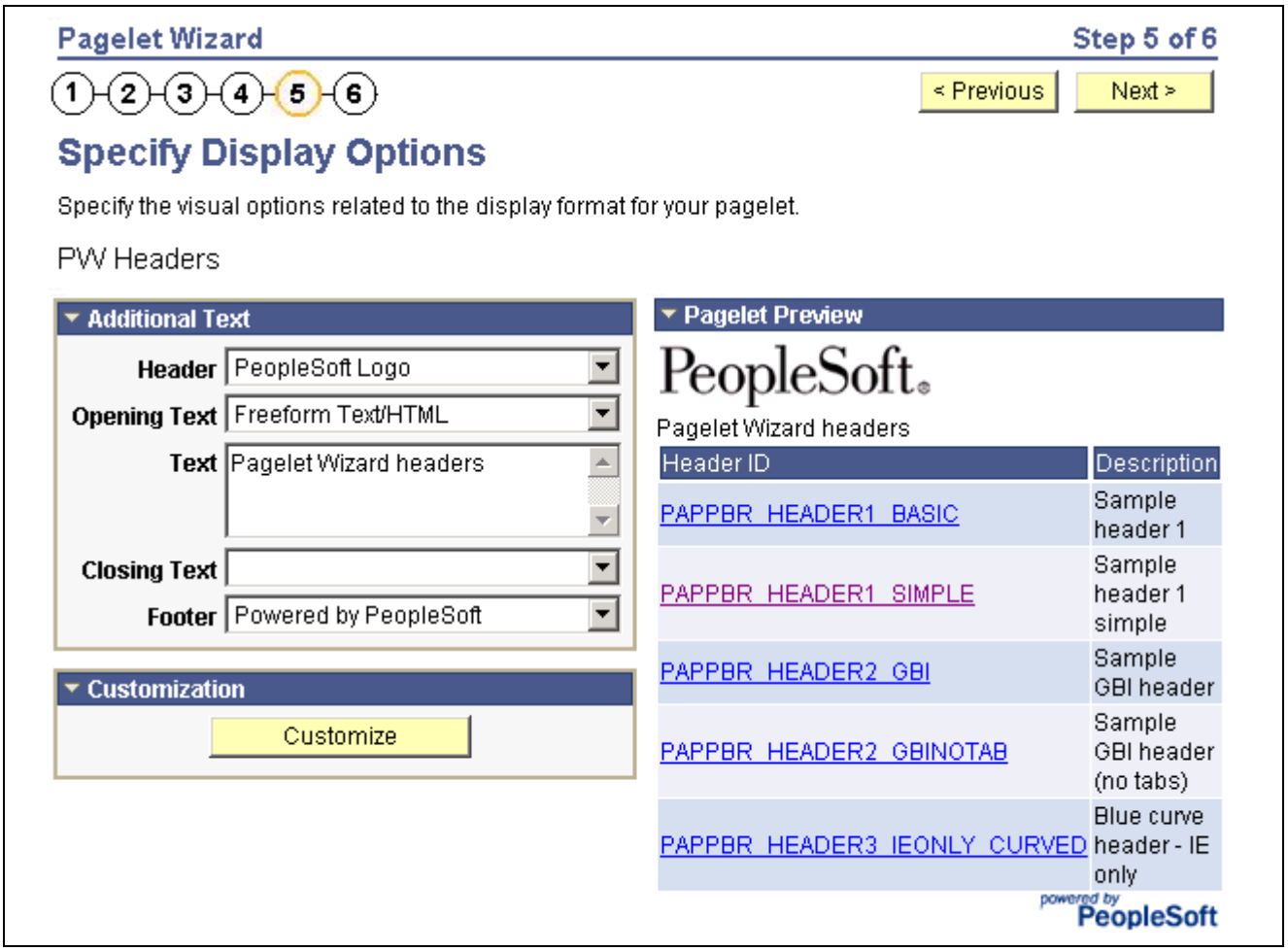
Specify Display Options page (passthru display format)

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Search List Display Options

Access the Specify Display Options page after selecting the *Search List* option on the Select Display Format page.



Specify Display Options page (search list display format)

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Menu Display Options

Access the Specify Display Options page after selecting the *Menu* option on the Select Display Format page.

Specify Display Options page (menu display format)

Menu Options

Top Level Title Select to have the menu pagelet display the top-level title defined in the underlying Navigation Collection.

Top Level Description Select to have the menu pagelet display the top-level description defined in the underlying Navigation Collection.

Maximum Columns Select the number of columns to display (options are *1* through *4*) on the Navigation Collection menu pagelet. Narrow pagelets display contents in one column. Wide pagelets can display contents in up to four columns.

See Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Display Sizes, page 195.

Display Order Select the order in which you want Navigation Collection elements to appear on the menu pagelet. When you create a Navigation Collection, you define the order of folders and links in the collection using options in the Placement Properties group box on the Edit Folder page and the Edit Link page. Options are:

Left to Right: Select to display the elements from left to right on the menu pagelet.

Top to Bottom. Select to display the elements from top to bottom on the menu pagelet.

Alternate Row Shading Select to display alternate row shading on the menu-formatted pagelet.

Parent Images Select the type of images you want to display for parent-level Navigation Collection elements (folders) on the menu pagelet. When you create a Navigation Collection, the collection definition includes specified large and small images that appear for parent-level elements in the Navigation Collection. Options are:

Large Images: Select to have the menu pagelet display the large folder image defined for the underlying Navigation Collection.

No Images: Select to prevent parent-level images from appearing on the menu pagelet.

Small Images: Select to have the menu pagelet display the small folder image defined for the underlying Navigation Collection.

Parent Descriptions Select to have the menu pagelet display parent-level folder descriptions defined in the underlying Navigation Collection.

Child Display Type Select the display type for child links on the menu pagelet. Options are:

Comma Delimited: Displays the maximum number of child links separated by commas.

List with Images: Displays the maximum number of child links accompanied by icons. This icon is the small content image selected in the Navigation Collection system or registry options for the underlying Navigation Collection.

Simple List: Displays the maximum number of child links, one child link per row. No bullets, numbers, or icons appear in the list.

Pagelet Preview

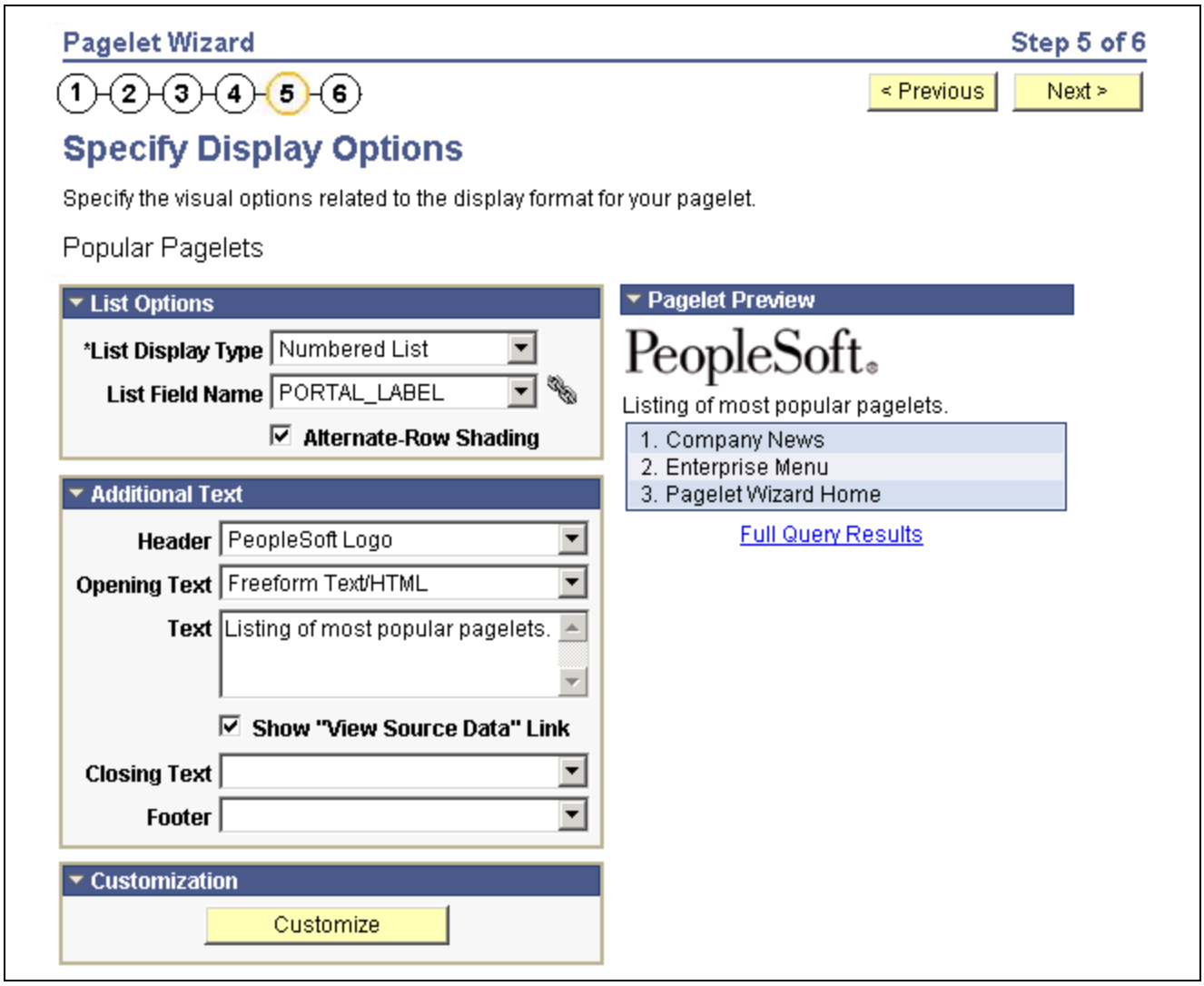
Displays a preview of the pagelet based on most display option settings on this page.

See Also

Chapter 23, “Using Navigation Collections,” page 291

Specifying List Display Options

Access the Specify Display Options page after selecting the *List* option on the Select Display Format page.



Specify Display Options page (list display format)

List Options

List Display Type Select the list format for displaying query results on the list pagelet. Options are:

Bulleted List: Displays query results in a bulleted list.

Numbered List: Displays query results in a numbered list.

Simple List: Displays query results in a list with no bullets or numbers.

List Field Name Select the field whose values are displayed in the list on the pagelet.

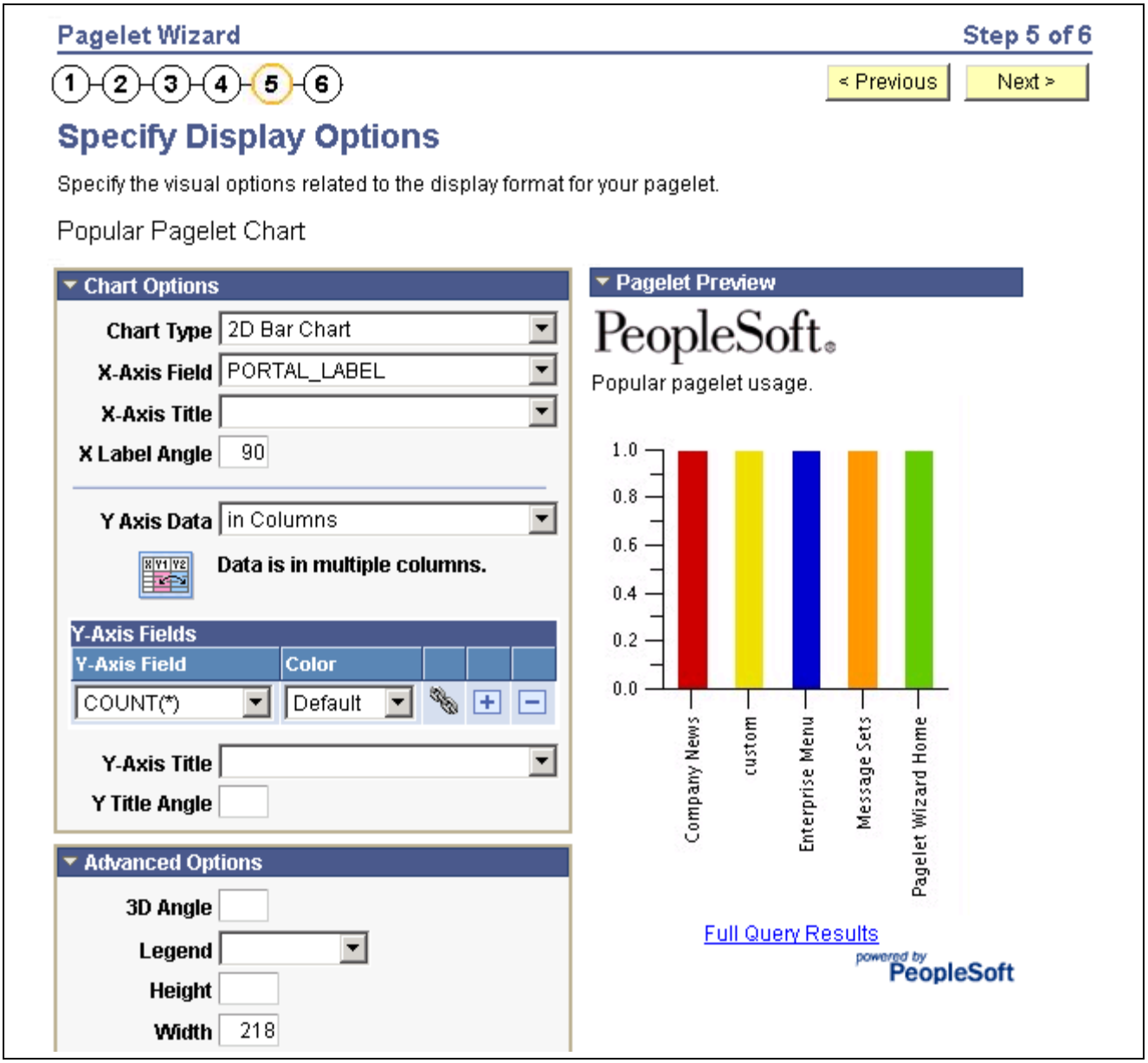
Alternate-Row Shading Select to use alternate-row shading on the list pagelet.

Pagelet Preview

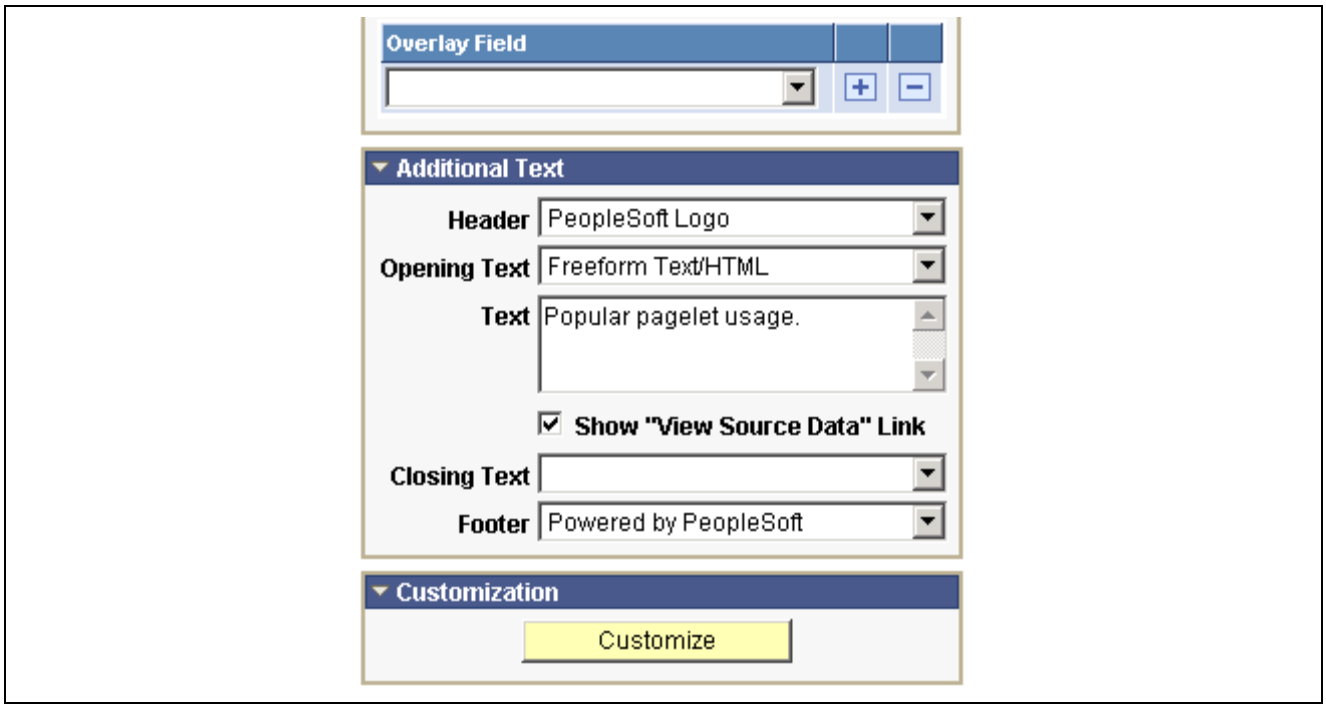
Displays a preview of the pagelet based on most display option settings on this page.

Specifying Chart Display Options

Access the Specify Display Options page after selecting the *Chart* option on the Select Display Format page.



Specify Display Options page (chart display format) (1 of 2)



Specify Display Options page (chart display format) (2 of 2)

To generate a chart pagelet using Pagelet Wizard, you include information related to the chart's *x* axis and *y* axis. The *y* axis is the axis that contains the query data. In most charts, the *y* axis is the vertical axis. However, in a horizontal bar chart, the *y* axis is the horizontal axis. The *x* axis is the axis against which the *y*-axis data is measured.

For example, a chart pagelet may have a *y* axis that displays the number of times that a pagelet has been accessed. This access count can be displayed against an *x* axis that shows the name of the pagelet accessed.

Chart Options

Chart Type Select the type of chart used to display the query results on the pagelet. Available chart formats are those supported by the PeopleCode Chart class.

See *Enterprise PeopleTools 8.45 PeopleBook: PeopleCode API Reference,* "Chart Class," Understanding the Chart Class

X-Axis Field Select the field used to derive *x*-axis values for the chart. Available values are determined by the data source's output fields.

X-Axis Label Select the field you want to use to derive the labels for the x-axis values for the chart. Available values are determined by the data source's output fields.

X Label Angle Enter the angle at which you want the text of the *x*-axis labels to appear along the *x*-axis on the pagelet.

Y Axis Data Select to indicate whether the *y*-axis data is stored in rows or columns in the selected query. This enables the Pagelet Wizard to read query data and generate the chart.

in Columns: Select if the *y*-axis data is stored in columns.

For example, you might have query data stored as shown below:

Quarter	Revenue	Profit
Q1	3562	1010
Q2	3473	1290
Q3	4568	1490
Q4	5668	2769

Y Axis Data (continued) *in Rows:* Select if the *y*-axis data to be used to generate the chart is stored in rows.

For example, you might have query data stored as shown below:

Quarter	Type	Value
Q1	Revenue	3562
Q1	Profit	1010
Q2	Revenue	3473
Q2	Profit	1290
Q3	Revenue	4568
Q3	Profit	1490
Q4	Revenue	5668
Q4	Profit	2769

Y-Axis Field Select the field that you want to use to derive *y*-axis values for the chart. Available values are determined by the data source's output fields.

If you set the Y Axis Data field to *in Columns,* you can select up to four *y*-axis fields to create a chart pagelet that displays data for multiple fields. For example, if you track sales figures for several departments over multiple years, you can assign each department its own *y*-axis value.

Color Select the color for the *y*-axis data. If you select multiple *y*-axis fields, selecting a different color for each field improves pagelet usability. This option is available only if the *y*-axis data is in stored columns.

Y-Axis Series This field is available if the Y Axis Data field is set to in *Rows.* Select the field used to generate a color-coded series of *y*-axis data. Colors are automatically assigned, but you can override them by supplying custom XSL.

Advanced Options

3D Angle If you select a 3D chart type, enter the angle at which you want the 3D data to be displayed.

Legend Select the area of the pagelet where the chart legend appears. Options are:

- *Bottom*
- *Left*
- *None*
- *Right*
- *Top*

Height Enter the height of the chart in pixels.

Width Enter the width of the chart in pixels.

If you do not enter height or width values, the chart is automatically sized based on the pagelet position and size (narrow or wide).

See Chapter 16, "Working With Pagelet Wizard," Pagelet Wizard Display Sizes, page 195.

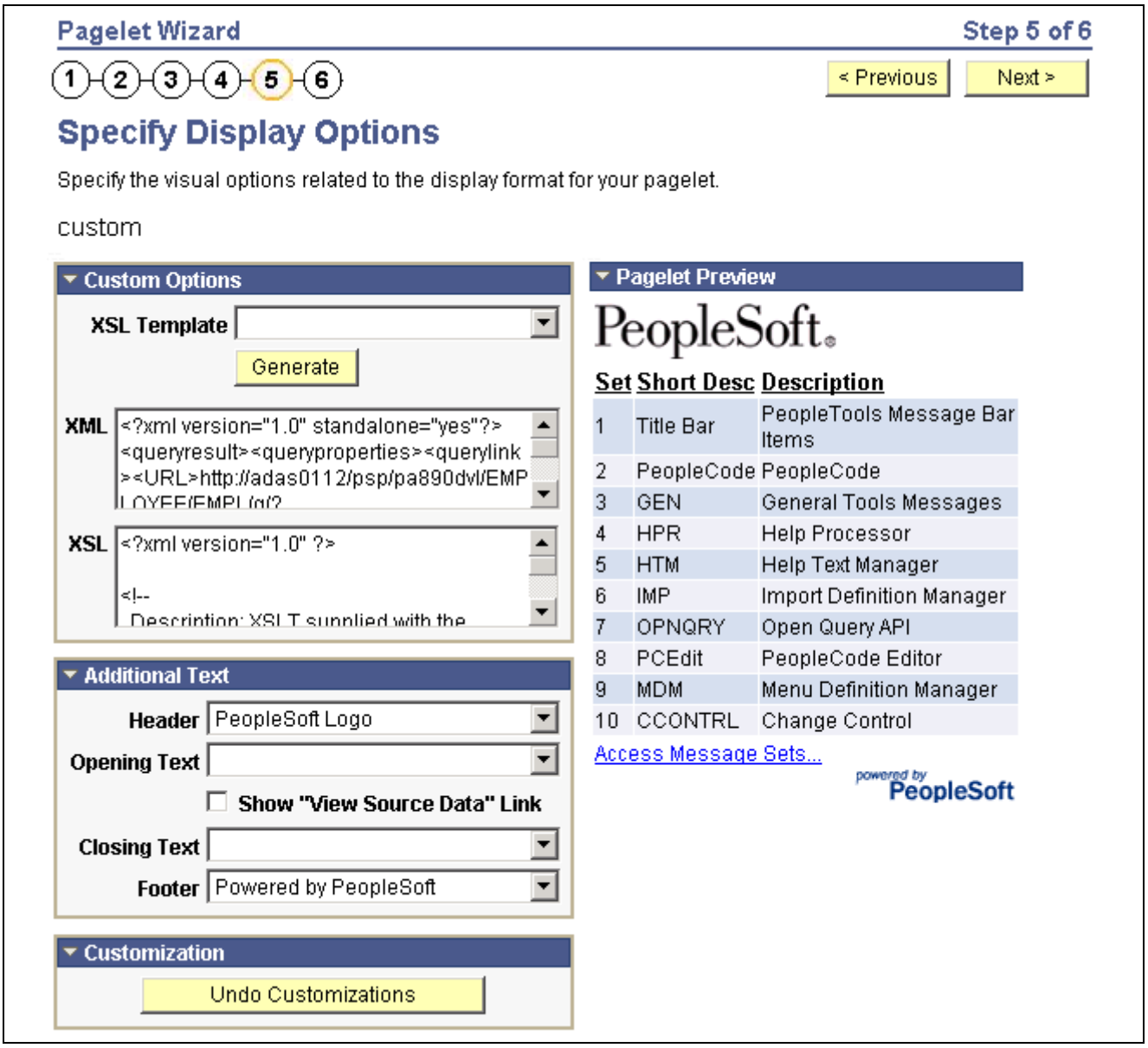
Overlay Fields Select up to three fields for which you want to display data using a line drawn over the background chart. Overlays apply to all charts types except 2D and 3D pie charts.

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Custom Display Options

Access the Specify Display Options page after selecting the Custom option on the Select Display Format page.



Specify Display Options page (custom display format)

Custom Options

XSL Template Select the XSL template used to generate the custom format pagelet. Available XSL templates are defined on the Define XSL page. Only XSL templates defined for the selected data source are displayed.

See Chapter 17, "Setting Up Pagelet Wizard Options," Defining Pagelet Wizard XSL Prototypes, page 207.

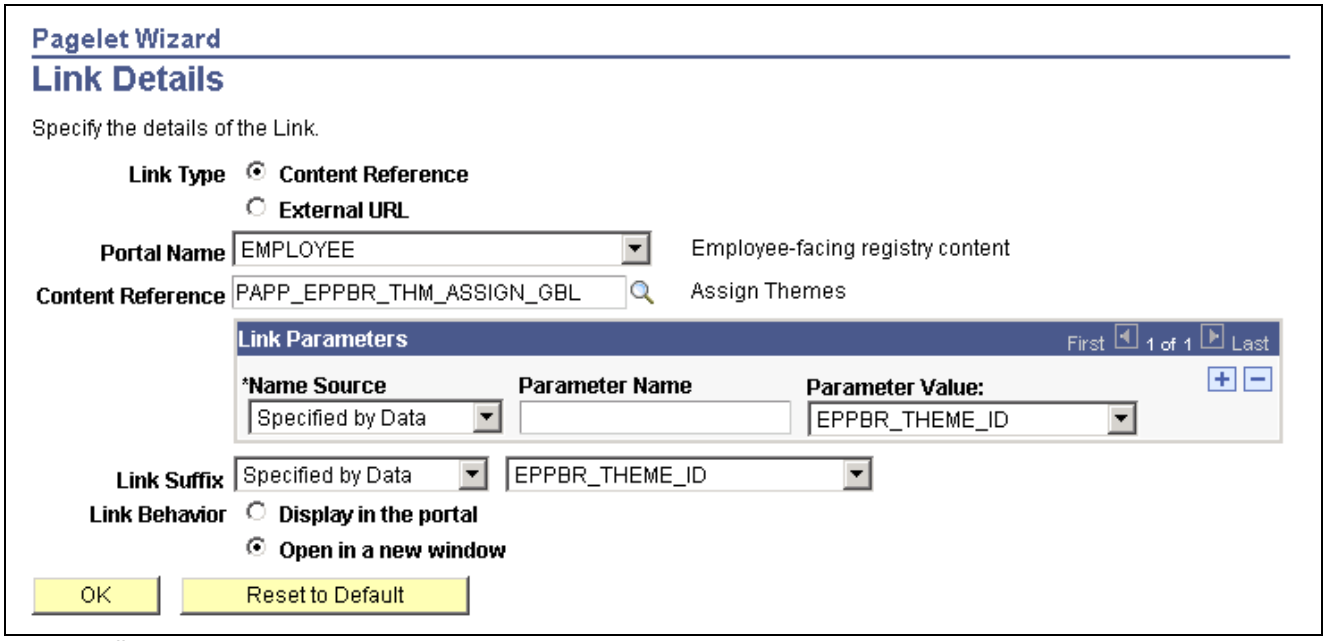
Generate Click to generate custom XSL for the pagelet based on the selected XSL template. This XSL is displayed in the XSL text box and is applied to the XML in the XML text box to generate the pagelet preview.

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Data Link Details

Access the Link Details page.



Link Details page

You can access this page for field or column names that are displayed on the Specify Display Options page when you access the page for the table, list, and chart formats. Entering information on the Link Details page enables you to link data that appears on the pagelet to additional information.

Pagelet Wizard builds the column data link as follows using link path, parameter, and suffix values:

http://baseURL/linkpath1/linkpath2/…/linkpathN/?parm1=val1&parm2=val2&…&parmN=valNlinksuffix

Link Type Select the type of link destination to define for the data. Options are:

Content Reference: Select to define data links to a content reference.

External URL: Select to define data links to an external URL.

Portal Name Select the portal in which the content reference resides. This field appears when you select *Content Reference* as the link type.

Content Reference Select the content reference to which the link points. This field appears when you select *Content Reference* as the link type.

Base URL Enter the URL to which you want the link to point. This field appears when you select *External URL* as the link type. Options are:

Specified by Data: Select to have the system dynamically generate the link information based on the selected data field.

Fixed Text Entry: Select to hardcode the link information by entering a value.

Link Paths

Path Source Select the method used to determine the link destination. Options are:

Specified by Data: Select to have the system dynamically generate the link information based on the selected data field.

Fixed Text Entry: Select to hardcode the link information by entering a value.

Link Path This field appears if you select *Specified by Data* as the path source. Select the data field used to generate the link.

Parameter Value This field appears if you select *Fixed Text Entry* as the path source. Enter a parameter value.

Link Parameters

Name Source Select the method used to determine the link destination. Options are:

Specified by Data: Select to have the system dynamically generate the link information based on the selected data field.

Fixed Text Entry: Select to hardcode the link information by entering a value.

Parameter Name Enter a value that corresponds to the value in the Parameter Value field.

Parameter Value If you have selected *Specified by Data* as the name source, select a data field.

If you have selected *Fixed Text Entry* as the name source, enter a parameter value.

Link Suffix Select the method used to determine the link destination.

Specified by Data: Select to have the system dynamically generate the link information based on the selected data field.

Fixed Text Entry: Select to hardcode the link information by entering a value.

If you have selected *Specified by Data* as the link suffix, select a data field.

If you have selected *Fixed Text Entry*as the link suffix, enter a parameter value.

Link Behavior Options are:

Display in portal: Select if you want the content accessed by the data link to be displayed within the portal.

Open in a new window: Select if you want the content accessed by the data link to be displayed in a new window.

Reset to Default Click to clear the values on this page and remove the link.

Step 6: Specifying Pagelet Publication Options

In this section, we discuss:

- Common elements used in this section.
- Specifying pagelet publication options.

Common Elements Used in This Section

This section discusses common elements used in this section.

Enable Caching Select to enable caching for the pagelet. Enabling caching improves performance because once the pagelet is stored in the cache memory using one of the caching types you define in the Caching Options group box, the next time the pagelet is accessed, it is retrieved and displayed from cache memory.

Caching Options

Entering caching options creates a content reference attribute for the new pagelet content reference. The content reference attribute contains the following information:

- Name: *PSCACHECONTROL*
- Label: *PSCache-Control*
- Attribute value: <Caching type field value>;max-age=<Minutes to Cache field value>

Once the pagelet is saved and registered, you can use Pagelet Wizard to alter caching options in the same way you can alter other registration information. However, altering caching option values will require that you re-register the pagelet.

See *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portals," Defining Content References

Caching Type *Private.* Select to enable user-based caching. For example, pagelet caching is stored per user. If user A accesses the pagelet, only user A will access the pagelet from the cache.

Public. Select to enable application-based caching. For example, pagelet caching is stored per web server instance. If a user accesses the pagelet using web server instance 1, only other users accessing the pagelets using web server instance 1 will access the pagelet from the cache.

Role. Select to enable role-based caching. For example, pagelet caching is stored per role combination. If user A has roles 1, 2, and 3, only another user with all roles 1, 2, and 3 will access the pagelet from the cache.

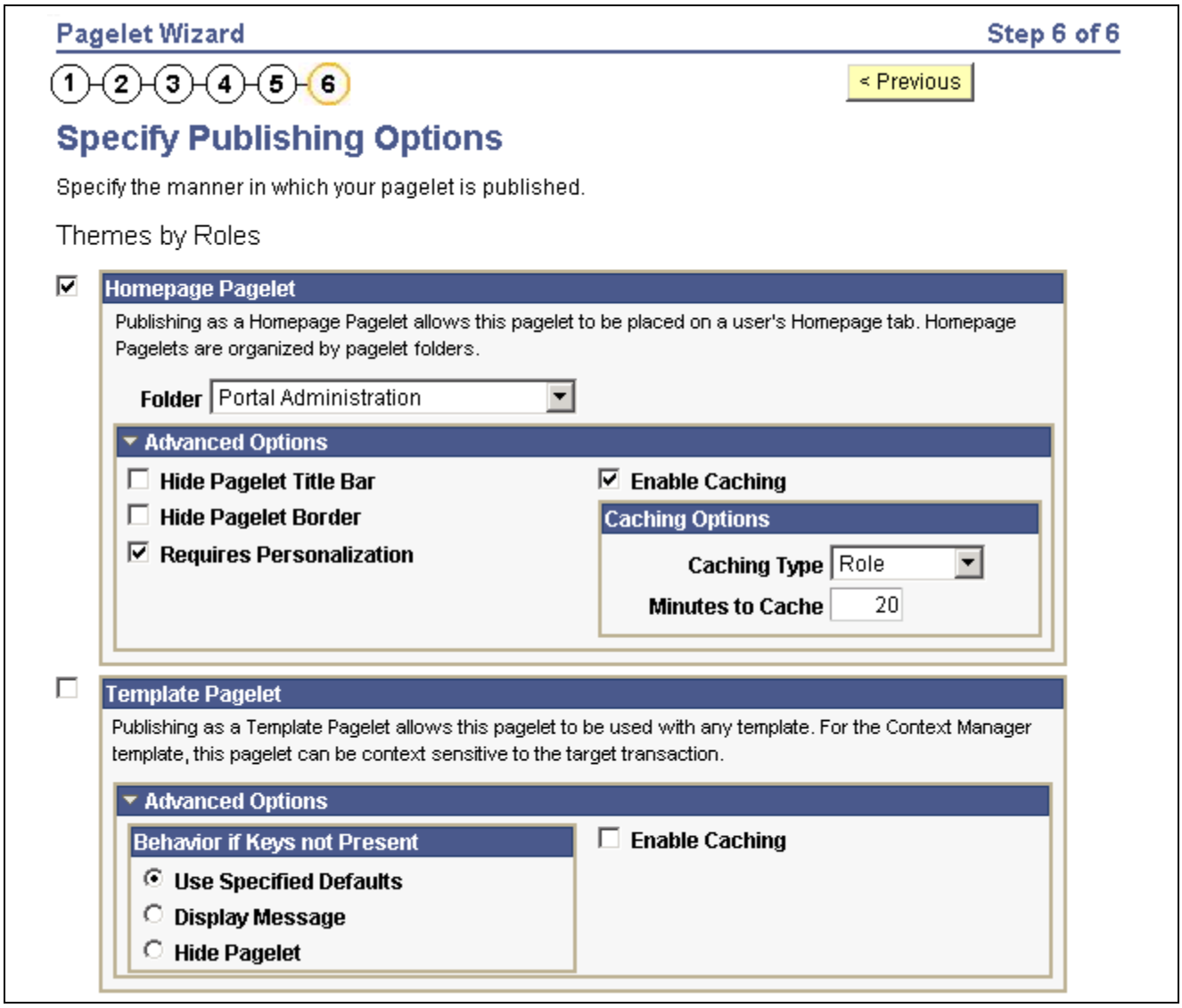
Minutes to Cache Enter the number of minutes for which you want the pagelet to remain in the cache. If you do not enter a value, a default value of 30 minutes is used.

Pages Used to Specify Pagelet Publication Options

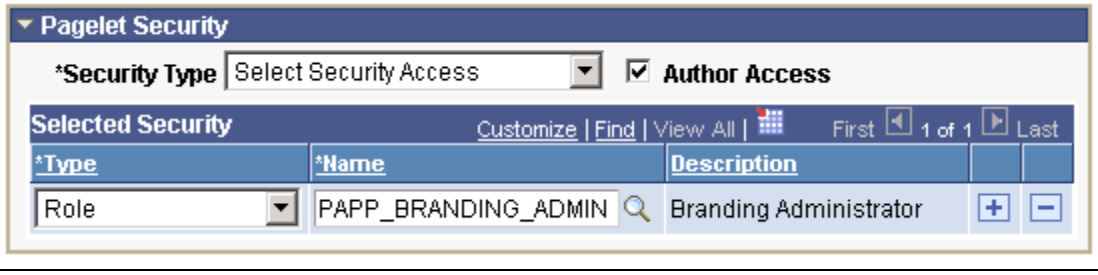
Page Name	Object Name	Navigation	Usage
Specify Publishing Options	EOPPB_WIZ_PUBOPT	Enterprise Components, Portal Utilities, Pagelet Wizard, Pagelet Wizard Click the Next button on the Specify Display Options page, or click the icon for step 6.	Specify the type of pagelet that you want to publish. Provide registration and security details, and register the pagelet.
Pagelet Creation Confirmed	EOPPB_WIZ_FINISH	Click the Finish button on the Specify Publishing Options page.	Confirm that the creation of the pagelet is complete.

Specifying Publication Options

Access the Specify Publishing Options page.



Specify Publishing Options page (1 of 2)



Specify Publication Options page (2 of 2)

When you access a pagelet definition on this page, the pagelet's registration option (Homepage Pagelet or Template Pagelet) is reflected only as it exists in the portal you are currently logged in to. Select the registration option to publish the pagelet to the portal where you are currently logged in.

For example, if you access ABC123 pagelet, the Homepage Pagelet option is selected only if the portal where you are currently logged in contains a CREF for the ABC123 pagelet. If there is no CREF for this pagelet in the current portal, even if there is a CREF for this pagelet in another portal on the same database, the Homepage Pagelet option will be clear. Select the Homepage Pagelet option to publish the pagelet to the current portal.

Homepage Pagelet

Select this option to publish the pagelet as a homepage pagelet that can be added to a user's homepage.

Note. The option to use Pagelet Wizard pagelets published as homepage pagelets is available if you have licensed PeopleSoft Enterprise Portal or a Portal Pack.

Folder Select the pagelet folder to which you want to assign the pagelet. The pagelet appears in this pagelet category on the Personalize Content page, which users access to add or remove pagelets from their homepages.

See *Enterprise PeopleTools 8.45 PeopleBook: Using PeopleSoft Applications,* "Setting User Preferences," Personalizing Content

Advanced Options

Hide Pagelet Title Bar Select to display the pagelet without a title bar.

Hide Pagelet Border Select to display the pagelet without a border.

Requires Personalization Select to specify that the pagelet requires end-user personalization. The pagelet displays a personalization button that accesses a personalization page. The pagelet requires that the end user access the personalization page to personalize and specify data source parameter values before it displays data. A message notifying the user of the personalization requirement appears on the pagelet. If this option is clear, the pagelet displays data using the default data source parameter values provided.

Template Pagelet

Select this option to publish the pagelet as a template pagelet that can be added to a PeopleSoft template. When used in conjunction with Intelligent Context Manager™, template pagelets can extract and leverage keys specified by the target transaction for the template.

Note. The option to use Pagelet Wizard pagelets published as template pagelets is available if you have licensed PeopleSoft Enterprise Portal or a Portal Pack.

Advanced Options

Behavior if Keys not Present Select one of the following options to indicate how pagelet data is displayed if target transaction keys are not present to populate the pagelet's parameters. Options are:

Use Specified Defaults: If target transaction keys are not present, the pagelet will display data using the defaults specified in Step 3: Specify Pagelet Data Source Parameters.

Display Message: If target transaction keys are not present, the pagelet does not display data but instead displays a message.

Hide Pagelet: If target transaction keys are not present, the pagelet does not appear.

Pagelet Security

If you change any pagelet security options for a registered pagelet, you must re-register the pagelet.

Security Type Select the type of security that you want to assign to the pagelet. Options are:

Public Access: Select to designate that all users can access the pagelet.

Select Security Access: Select to assign role- or permission list-based security to the pagelet. The Selected Security scroll area appears.

Author Access Select to provide the pagelet author with access to the pagelet regardless of any security restrictions assigned to the pagelet. This access is granted based on the author's user ID.

Selected Security

Type Select the type of security assigned to the pagelet. Specified security is applied to the content reference entries when the pagelets are registered. Options are:

Permission List: Select to assign permission list-based security to the pagelet. Select a permission list in the Name field.

Role: Select to assign role-based security to the pagelet. Select a role in the Name field.

Finish Click to complete the creation of the pagelet. The Pagelet Creation Confirmed page appears.

After you click the Finish button for a pagelet, the button no longer appears when you access the pagelet definition in Pagelet Wizard. Once a pagelet has been created, you can apply any modifications to the pagelet definition by clicking the Save button.

CHAPTER 19

Administering Pagelets

This chapter lists common elements used in adminstering pagelets and discusses how to:

- Delete and copy Pagelet Wizard pagelets.
- Generate export and import scripts for pagelets.
- Publishing pagelets to other portals.
- Reviewing Pagelet Wizard pagelet details.

Note. Examples of PeopleSoft Pure Internet Architecture pages in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page elements displayed remain the same.

Common Elements Used in This Chapter

This section discusses common elements used in this chapter.

Filter Pagelets

If appropriate, use the options in the Filter Pagelets group box to narrow the range of pagelets displayed in the Pagelets scroll area.

Data Type Select the data type of the pagelet that you want to appear in the Pagelets scroll area. Options are:

- *All Data Types.*
- *HTML.*
- *Navigation Collection.*
- *PS Query.*
- *Search Record.*

Published Type Select the pagelet type assigned to the pagelet that you want to appear in the Pagelets scroll area. Options are:

- *All Pagelets.*
- *Homepage Pagelet.*
- *Non-published Pagelet.*
- *Template Pagelet.*

Portal Name The portal name that you select here does not filter results, but rather serves to inform you which pagelets displayed in the Pagelet scroll area have been published in the selected portal. Pagelets are automatically published to the portal where the pagelet author is logged in when creating the pagelet.

Pagelet Keyword Enter keywords to narrow the range of pagelets that appear in the Pagelets scroll area. Pagelet elements that are used as keywords used include pagelet ID, title, and description.

Search Click to perform a pagelet search using the criteria entered in the Filter Pagelets group box.

Reset Click to clear the Pagelet Keyword field.

Pagelets

The Pagelets scroll area provides the following pagelet information.

Pagelet ID Displays the pagelet ID. Click to access the pagelet definition in the Pagelet Wizard component.

Pagelet Title Displays the pagelet title.

Homepage If selected, indicates that the pagelet has been published as a homepage page.

Template If selected, indicates that the pagelet has been published as a template pagelet.

Deleting and Copying Pagelet Wizard Pagelets

This section discusses how to:

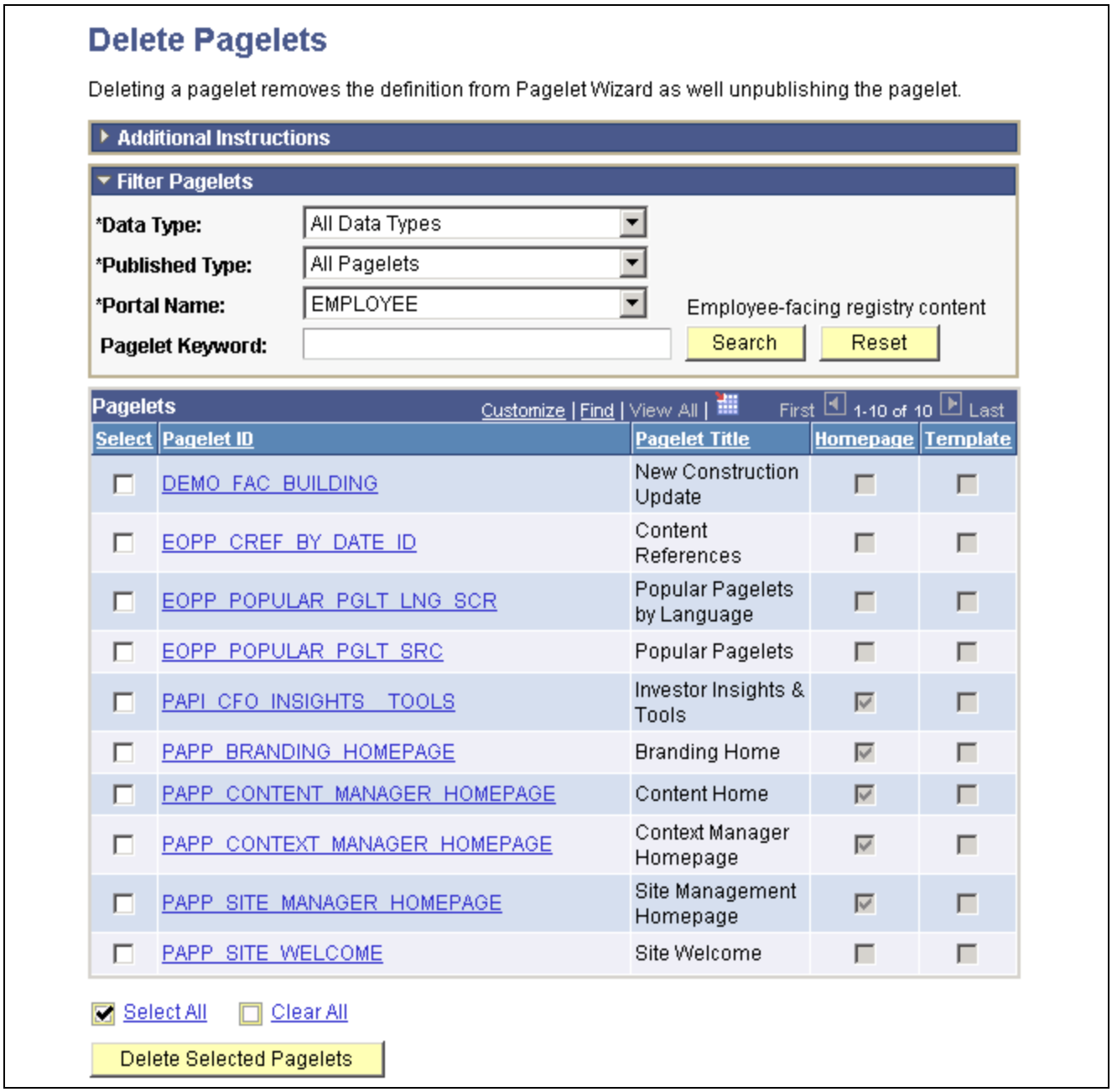
- Delete pagelets.
- Copy pagelets.

Pages Used to Delete and Copy Pagelet Wizard Pagelets

Page Name	Object Name	Navigation	Usage
Delete Pagelets	EOPPB_ADMIN_DELETE	Enterprise Components, Portal Utilities, Pagelet Wizard, Delete Pagelets	Delete one or more pagelets. Deleting a pagelet removes the pagelet definition from Pagelet Wizard, as well as deletes associated metadata and registry entries from the database and portal registry.
Copy Pagelets	EOPPB_ADMIN_SAVEAS	Enterprise Components, Portal Utilities, Pagelet Wizard, Copy Pagelets	Create a new pagelet that is a clone of an existing pagelet. The new pagelet has a unique pagelet ID. The pagelet copy inherits all aspects of the original pagelet, with the exception of registration information.

Deleting Pagelets

Access the Delete Pagelets page.



Delete Pagelets page

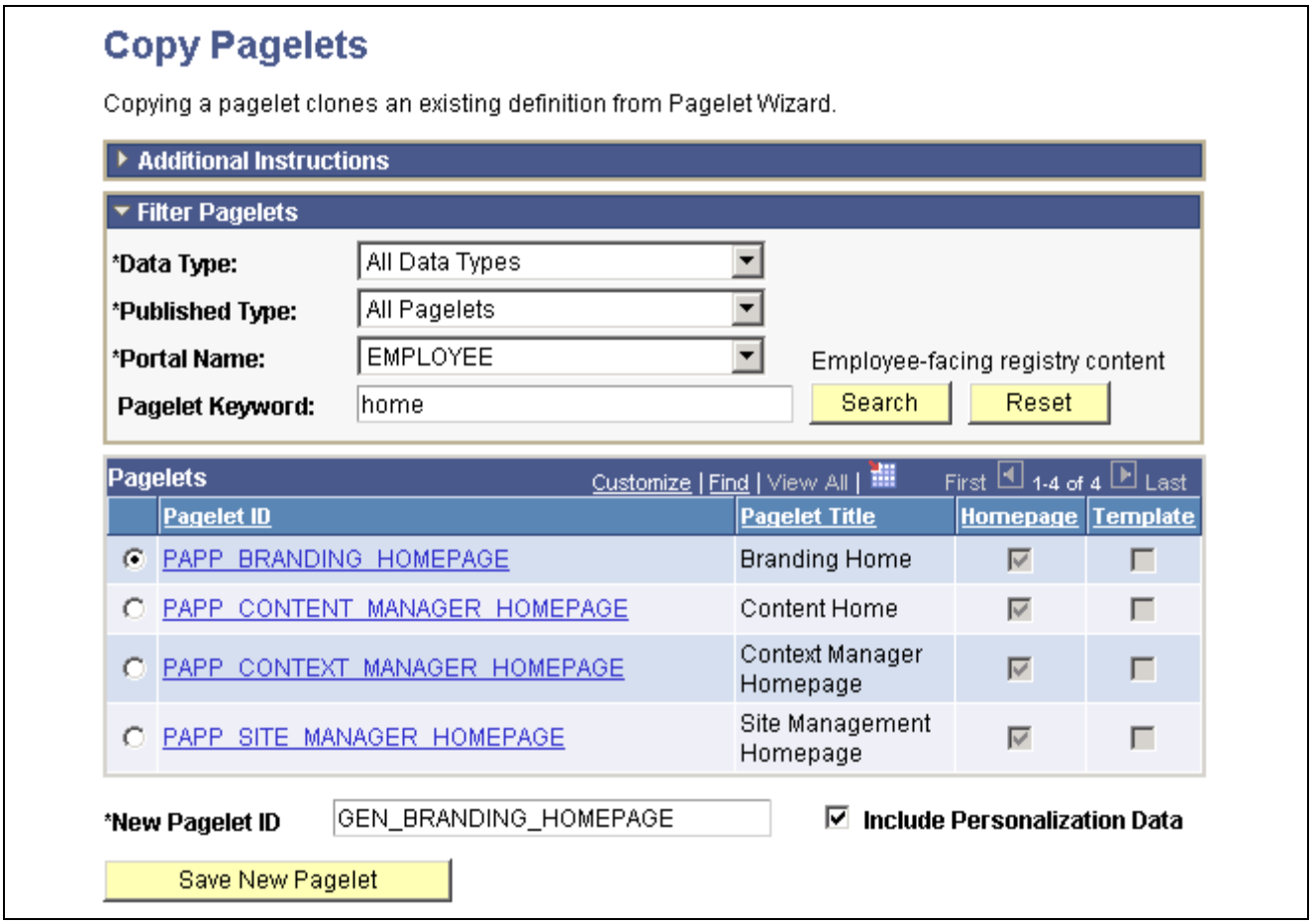
Deleting a pagelet removes the pagelet definition from Pagelet Wizard. In addition, associated metadata and registry entries are deleted from the database and portal registry.

Select Select the pagelets to delete.

Delete Selected Pagelets Click to delete selected pagelets. The system prompts you to confirm the deletion.

Copying Pagelets

Access the Copy Pagelets page.



Copy Pagelet page

The pagelet copy inherits all aspects of the original pagelet, with the exception of registration information.

Select Select the pagelet to copy.

New Pagelet ID Enter a unique pagelet ID for the new pagelet that will be copied from the selected existing pagelet.

Include Personalization Data Select to copy any end-user personalizations defined for the existing pagelet along with the new pagelet.

Save New Pagelet Click to create the cloned pagelet with the new pagelet ID. Once you create the copied pagelet, you must publish it before it can be used as a homepage pagelet or a template pagelet. You can access the new pagelet in the Pagelet Wizard and publish the pagelet during step 6 on the Specify Publishing Options page.

Generating Export and Import Scripts for Pagelet Wizard Pagelets

This section provides overviews of export and import scripts for Pagelet Wizard pagelets and export and import scripts for Pagelet Wizard setup data, and discusses how to generate export and import pagelet scripts.

Understanding Export and Import Scripts for Pagelet Wizard Pagelets

You can use these export and import scripts to copy Pagelet Wizard pagelet definitions from one database to another.

When you export or import a pagelet, the imported pagelet is not automatically published. You must access the pagelet definition in Pagelet Wizard in the target portal in the target database and publish the pagelet on the Specify Publishing Options page in step 6.

Alternatively, you can run the generated export and import Data Mover scripts (.DMS scripts) and copy the pagelet CREF by placing it into a PeopleSoft Application Designer project.

Note. These export and import scripts do not include the ability to export or import Pagelet Wizard setup data associated with the pagelet definitions being exported/imported.

See Also

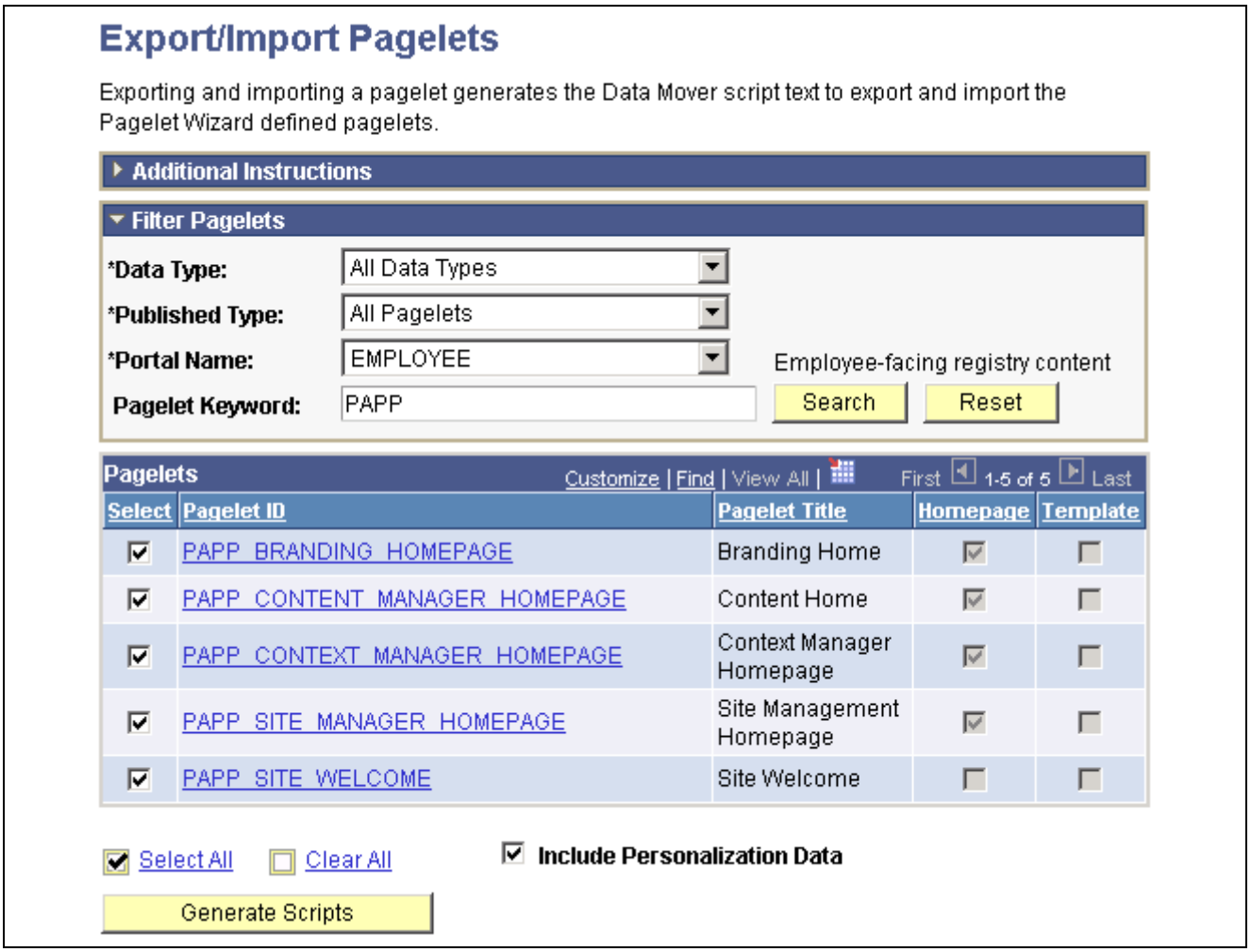
Chapter 19, "Administering Pagelets," Generating Export/Import Scripts for Pagelet Wizard Setup Data, page 262

Pages Used to Generate Export and Import Scripts for Pagelet Wizard Pagelets

Page Name	Object Name	Navigation	Usage
Export/Import Pagelets	EOPPB_ADMIN_EXPIMP	Enterprise Components, Portal Utilities, Pagelet Wizard, Export/Import Pagelets	Generate import and export scripts for multiple selected pagelets. You can then copy and save these scripts as .DMS scripts for execution in PeopleSoft Data Mover.
Data Mover Scripts	EOPPB_ADMIN_SCRIPT	Click the Generate Scripts button on the Export/Import Pagelets page.	View export and import scripts generated for the pagelets selected on the Export/Import Pagelets page.

Generating Export and Import Pagelet Scripts

Access the Export/Import Pagelets page.



Export/Import Pagelets page

Select Select the pagelets for which you want to generate export and import scripts.

Include Personalize Data Select to include any end-user personalizations defined for the selected pagelets in the export and import.

Generate Scripts Click to generate export and import scripts for the selected pagelets. The scripts appear on the Data Mover Scripts page.

Copy the generated scripts into PeopleSoft Data Mover to export the selected pagelet definitions from a source database and import them into a target database.

See *Enterprise PeopleTools 8.45 PeopleBook: Data Management,* "Using PeopleSoft Data Mover"

Note. There are other Pagelet Wizard-related Data Mover scripts delivered with PeopleSoft Enterprise Components for users who need more customizable scripting for pagelet migration between databases. You can find these scripts delivered in the pshome scripts directory.

Generating Export/Import Scripts for Pagelet Wizard Setup Data

In this section, we provide an overview of export/import scripts for Pagelet Wizard setup data.

Understanding Export and Import Scripts for Pagelet Wizard Setup Data

We deliver the following Data Mover scripts to export/import Pagelet Wizard setup data:

- EOPPB_SETUP_DEFN_IMP.DMS
- EOPPB_SETUP_DEFN_EXP.DMS

Use these scripts to copy the following types of Pagelet Wizard setup data from one database to another; from a development database to a production database, for example:

- Footers.
- Headers.
- Categories.
- XSL prototype templates.

You can find these scripts delivered in the pshome scripts directory.

Publishing Pagelets to Other Portals

This section discusses how to:

- Publish a pagelet from a source portal to multiple target portals that reside in the same database as the source portal.
- Publish multiple pagelets from a source portal to a target portal that resides in the same database as the source portal.

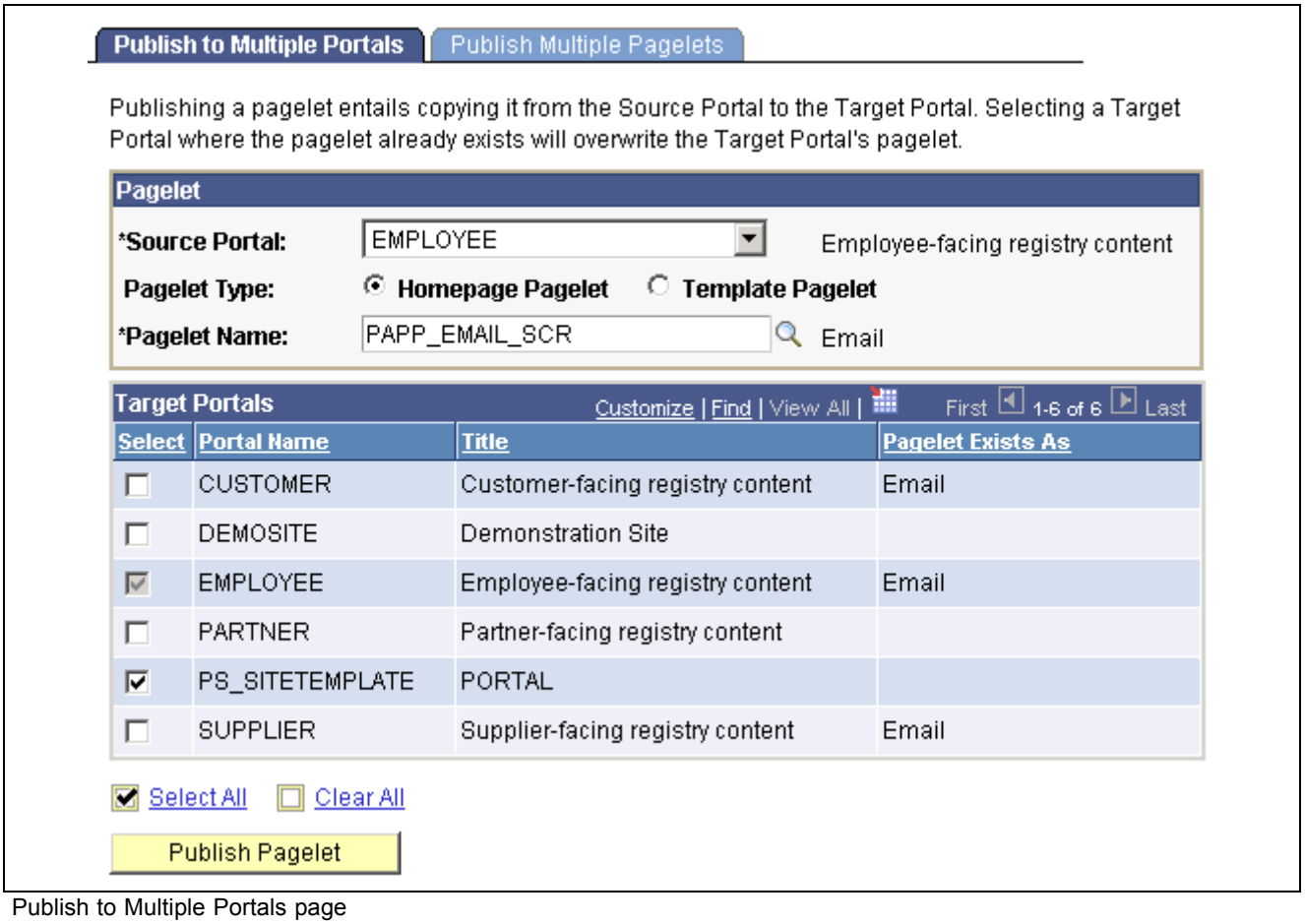
Note. The Publish Pagelets component can be used with Pagelet Wizard and Portal Pack pagelets.

Pages Used to Publish Pagelets to Other Portals

Page Name	Object Name	Navigation	Usage
Publish to Multiple Portals	EOPP_PMPUBPRTL	Enterprise Components, Portal Utilities, Publish Pagelets, Publish to Multiple Portals	Copy a pagelet from a source portal and publish it to multiple target portals that reside in the same database as the source portal.
Publish Multiple Pagelets	EOPP_PMPUBPGLT	Enterprise Components, Portal Utilities, Publish Pagelets, Publish Multiple Pagelets	Copy multiple pagelets from a source portal and publish them to a selected target portal that resides in the same database as the source portal.

Publishing a Pagelet to Multiple Portals

Access the Publish to Multiple Portals page.



Publish to Multiple Portals page

Pagelet

Source Portal Select the portal from which you want to copy a pagelet. You can select from portals that reside in the same database where you are currently logged in.

Pagelet Type Select the type of pagelet you want to copy. Options are:

- *Homepage Pagelet.*
- *Template Pagelet.*

Pagelet Name Select a pagelet that exists in the selected Source Portal registry. This is the pagelet content reference you want to publish to one or more other portal registries that reside in the database where you are currently logged in.

Target Portals

Warning! Selecting a target portal in which the selected pagelet already exists overwrites the pagelet content reference in the target portal.

Select Select to publish the selected pagelet to the selected target portal. Use the Portal Name and Title fields to identify the available target portals.

Pagelet Exists As If the pagelet you have selected to copy and publish to a target portal already exists in the target portal, this field is populated with the pagelet title as it exists in the portal registry.

For instance, in the example shown here the Email pagelet is selected for copying from the EMPLOYEE source portal to the PS_SITETEMPLATE target portal. The Pagelet Exists As field shows that this pagelet also already exists in the CUSTOMER and SUPPLIER target portals. If you copy the pagelet to the CUSTOMER and SUPPLIER target portals, the pagelet definitions in the CUSTOMER and SUPPLIER target portals are overwritten by the pagelet definition in the EMPLOYEE source portal.

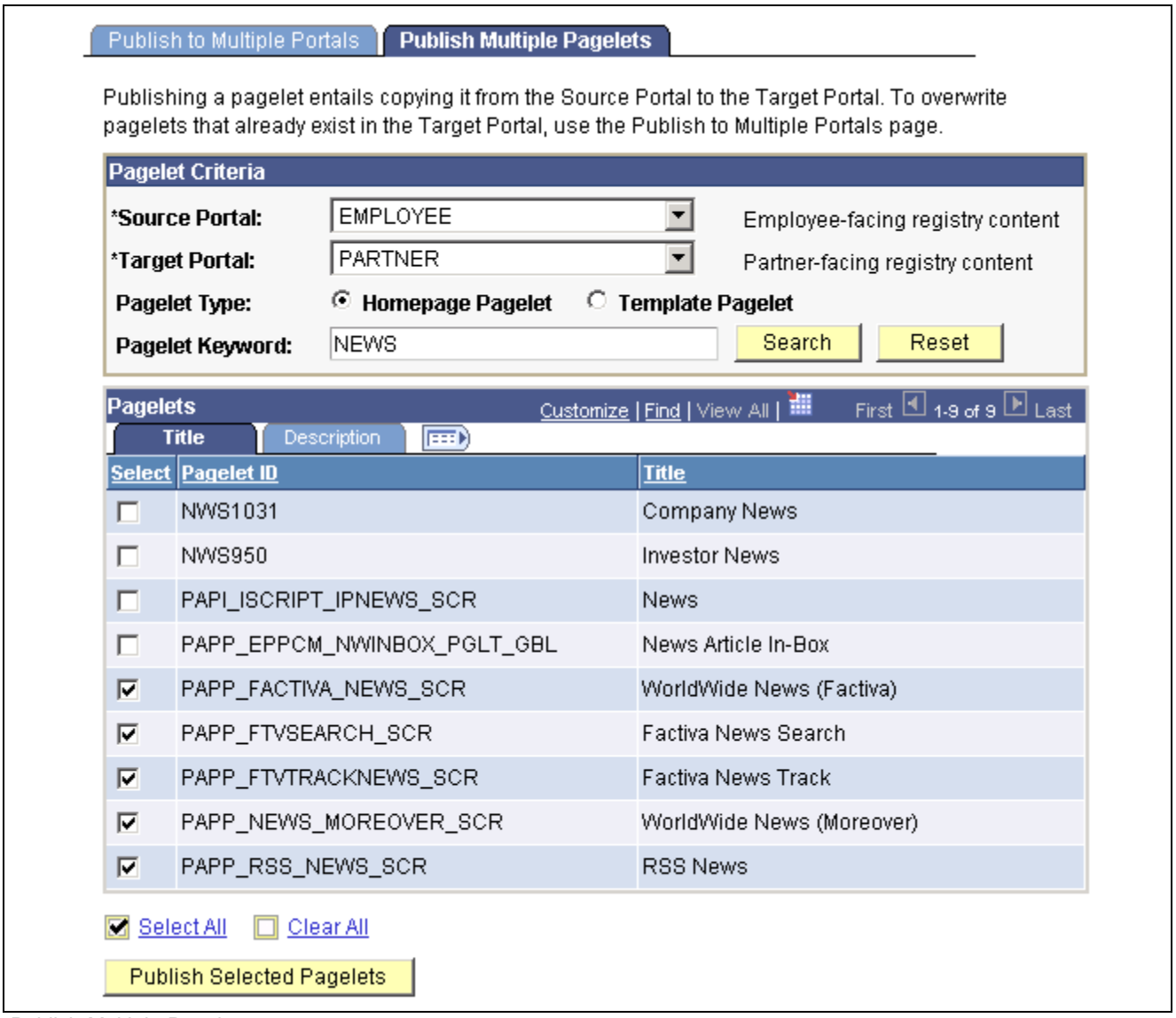
Publish Pagelet Select to copy the pagelet definition from the source portal and publish it in the selected target portals.

Content reference portal registry entries are created in the registries of all selected target portals using the same values that exist in the original pagelet registration in the source portal.

If the folder in which the original pagelet is registered does not exist for a selected target portal registry, it is dynamically generated.

Publishing Multiple Pagelets to a Portal

Access the Publish Multiple Pagelets page.



Publish Multiple Pagelets page

Pagelet Criteria

Source Portal Select the portal from which to copy a pagelet. You can select from portals that reside in the same database where you are currently logged in.

Target Portal Select the portal to which you want to copy pagelets. You can select from portals that reside in the same database where you are currently logged in.

After you select the source and target portals, the Pagelet group box displays a list of pagelets that are registered in the source portal but not the target portal.

Pagelet Type Select the type of pagelet you want to copy. Options are:

- *Homepage Pagelet.*
- *Template Pagelet.*

Pagelet Keyword Enter keywords to narrow the scope of pagelets displayed in the Pagelet group box.

Pagelets

Select Select to publish selected pagelets to the selected target portal. Use the Pagelet ID, Title, and Description fields on the Title and Description tabs to identify the pagelet.

Publish Selected Pagelets Select to copy the pagelet definition from the source portal and publish it any selected target portals.

Content reference portal registry entries are created in the target portal registry using the same values that exist for the original pagelet in the source portal registry.

If a folder in which an original pagelet is registered does not exist for a selected target portal registry, it is dynamically generated.

Reviewing Pagelet Wizard Pagelet Details

This section lists the page used to review Pagelet Wizard pagelet details.

Page Used to Review Pagelet Wizard Pagelet Details

Page Name	Object Name	Navigation	Usage
Review Pagelets	EOPPB_ADMIN_REPORT	Enterprise Components, Portal Utilities, Pagelet Wizard, Review Pagelets	Review information about existing pagelets created using Pagelet Wizard. This page provides basic identifying information about a pagelet including a pagelet's data source, display format, and pagelet type. The page also provides information about the user who created the pagelet and when it was created. In addition, it provides information about the last users to modify a pagelet and when it was modified.

CHAPTER 20

Using Pagelet Wizard Post-Transformation Processing

This chapter provides an overview of Pagelet Wizard post-transformation processing and discusses how to:

- Insert Message Catalog text.
- Format Date, Time, DateTime, Currency, and Number elements.
- Insert content-reference-based links.
- Insert content reference URLs as text.
- Generate charts.
- Insert pagelet headers and footers in specific locations.

Understanding Pagelet Wizard Post-Transformation Processing

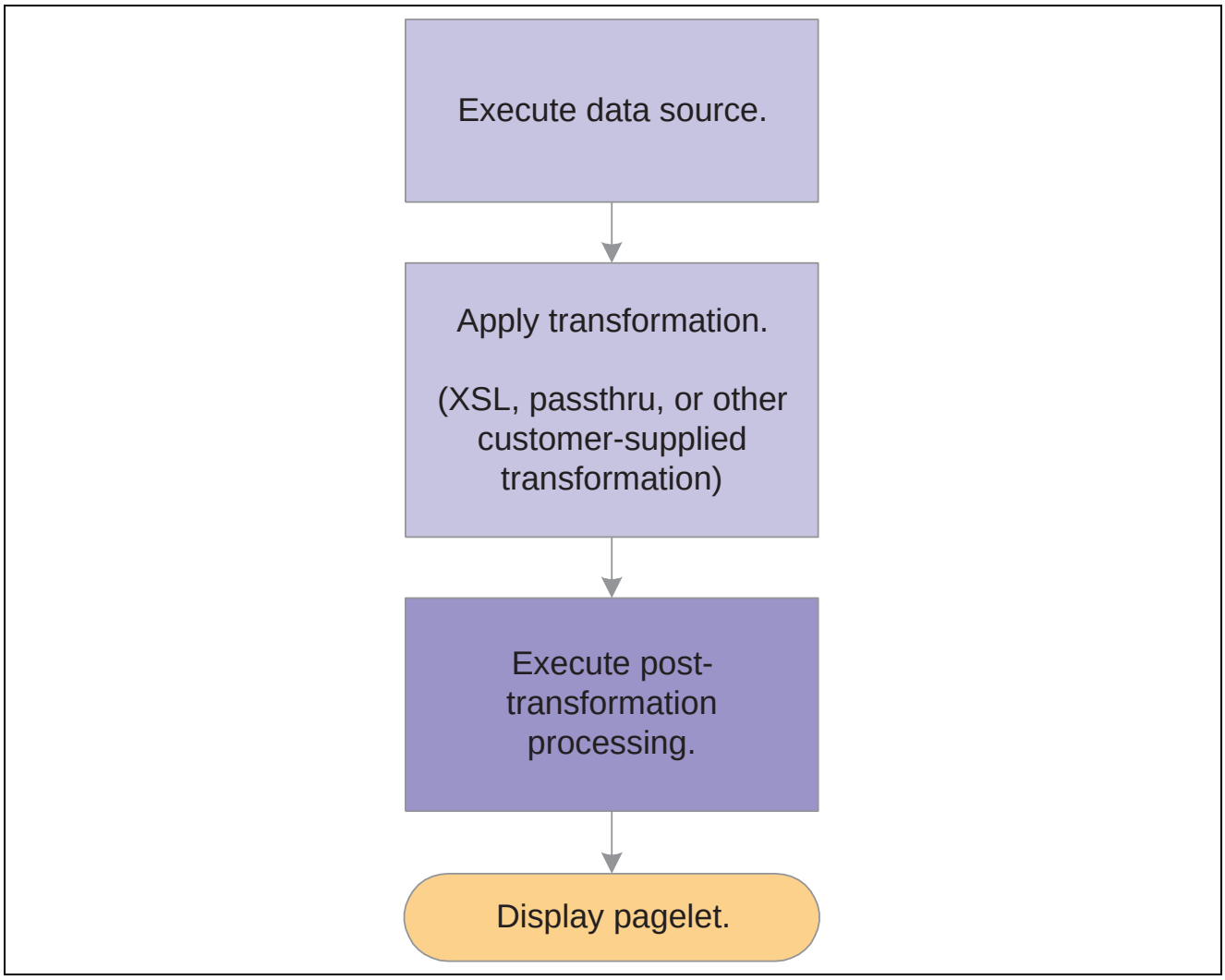
Pagelet Wizard enables you to use delivered Pagelet Wizard builder transformation tasks in custom and passthru transformations.

Listed in the order in which they are executed in post-transformation processing, the following transformation tasks are not automatically generated for custom and passthru transformations, but they can be added using embedded tags:

1. Insert Message Catalog text.
2. Format Date, Time, DateTime, Currency, and Number.
3. Insert content-reference-based links.
4. Insert content reference URLs as text.
5. Generate charts.
6. Insert pagelet headers and footers in specific locations.

These tasks are handled by the Post Transformation step, which is built into Pagelet Wizard.

The following diagram illustrates the point at which the Post Transformation step runs in the process flow for displaying a Pagelet Wizard pagelet.



Pagelet Wizard pagelet-display processing flow

The Post Transformation step runs after the completion of the Transformation step. Following the Post Transformation step, the Pagelet Wizard pagelet appears for the user. Pagelet Wizard uses embedded tags to identify the type of Post Transformation processing that occurs.

Tags that perform these post-transformation tasks can be placed in the HTML for HTML pagelets using the passthru transformation, or in the XSL for pagelets using a custom transformation. For all other transformations—such as chart, list, table, and so forth—Pagelet Wizard automatically generates the tags when necessary. These tags are primarily meant to be used by pagelet designers who want to create custom pagelets using functionality similar to what our delivered Pagelet Wizard builders provide.

Inserting Message Catalog Text

Use the following syntax to insert the text of a message catalog entry. Variables that are user-supplied are italicized.

Note. This processing step occurs only if the output of the Transformation step is XHTML-compliant.

See http://www.w3.org/TR/xhtml1/

Item to Insert	XHTML Tag	HTML After Processing
Message Catalog Text	<PSMSGTEXT MSGSET="*message set*" MSGNBR="*message number*" DEFAULT="*defaulttext*"/>	Message text.
Message Catalog Explain Text	<PSMSGEXPLAINTEXT MSGSET="*message set*" MSGNBR="*message number*" DEFAULT="*defaulttext*"/>	Message explanation text.

Formatting Date, Time, DateTime, Currency, and Number Elements

Use the following syntax to format a Date, Time, DateTime, Currency, or Number elements. Variables that are user-supplied are set in italics.

Note. This processing step only occurs if the output of the Transformation step is XHTML compliant.

See http://www.w3.org/TR/xhtml1/

Item to Insert	XHTML Tag	HTML After Processing
Date	<PSFORMAT TYPE="DATE" [FORMAT="*format string*"]>DateToFormat< /PSFORMAT> **Note.** FORMAT has the same syntax as the DateTimeToLocalizedString() PeopleCode Function.	Formatted Date.
Time	<PSFORMAT TYPE="TIME" [SOURCETZ="*Source Time Zone*"][DESTTZ="*Destination Time Zone*"] [FORMAT="*format string*"]>TimeToFormat< /PSFORMAT> **Note.** SOURCETZ and DESTTZ have the same format as the DateTimeToTimeZone() PeopleCode function. FORMAT has the same syntax as the DateTimeToLocalizedString() PeopleCode Function.	Formatted Time.

Item to Insert	XHTML Tag	HTML After Processing
DateTime	<PSFORMAT TYPE="DATETIME" [SOURCETZ="*Source Time Zone*"][DESTTZ="*Destination Time Zone*"] [FORMAT="*format string*"]>DateTimeToFormat< /PSFORMAT> **Note.** SOURCETZ and DESTTZ have the same format as the DateTimeToTimeZone() PeopleCode function. FORMAT has the same syntax as the DateTimeToLocalizedString() PeopleCode Function.	Formatted DateTime.
Number	<PSFORMAT TYPE="NUMBER" FORMAT="*format string*" [WIDTH="*width*"] [PRECISION= "*precision*"]>NumberToFormat< /PSFORMAT> **Note.** FORMAT, WIDTH, and PRECISION have the same format as the NumberToDisplayString() PeopleCode Function.	Formatted Number.
Currency	<PSFORMAT TYPE="CURRENCY" FORMAT="*format string*" [WIDTH= "*width*"] [PRECISION="*precision*"] [CURRENCY_CODE="*CODE* "] [CURRENCY_DECORATION= "*SYMBOL*/*CODE*/*BOTH* "]>NumberToFormat</PSFORMAT> **Note.** FORMAT, WIDTH, and PRECISION have the same format as the NumberToDisplayString() PeopleCode Function. CURRENCY_ CODE can be one of the supported PeopleSoft currency codes, or if left blank, defaults to the user's currency code. CURRENCY_DECORATION can have the value of *SYMBOL* (currency symbol appears before the number), *CODE* (currency code appears after the number), or *BOTH* (the default).	Formatted Number with currency symbol and/or currency code.

Inserting a Content-Reference-Based Link

Use the following syntax to insert a content-reference-based link into a Pagelet Wizard pagelet. Variables that are user-supplied are set in italics.

Note. This processing step only occurs if the output of the Transformation step is XHTML compliant.

See http://www.w3.org/TR/xhtml1/

Item to Insert	XHTML Tag	HTML After Processing
Content Reference Link	<PSREGISTERDLINK PORTAL= "*portal name*" CREF="*content reference name*" target="*target*" APPEND="*append*">Text of Link</PSREGISTEREDLINK>	<A HREF="http…URL of content reference…[?/&]append" target="target">Text of Link</A>

Inserting a Content Reference URL as Text

Use the following syntax to insert a content reference URL as text into a Pagelet Wizard pagelet. Variables that are user-supplied are set in italics.

Note. This processing step only occurs if the output of the Transformation step is XHTML compliant.

See http://www.w3.org/TR/xhtml1/

Item to Insert	XHTML Tag	HTML After Processing
Content Reference URL	<PSREGISTERDURL PORTAL="*portal name*" CREF="*content reference name*" APPEND="*append*"/>	http…URL of content reference…[? /&]append

Generating Charts

Embedded charts are generated in the Post Transformation step.

For details about chart generation and its syntax, refer to the ChartXSLBuilder included in the EOPPB_PAGELET Application Class. You can access the ChartXSLBuilder application class code using PeopleSoft Application Designer.

Inserting Pagelet Headers and Footers in Specific Locations

Pagelet Wizard automatically inserts a pagelet's header in front of the output of the Transformation step, and it inserts the pagelet's footer after the output of the Transformation step.

To alter this behavior, you must use special HTML comment tags. Here is the syntax of these tags:

Item to Insert	HTML Tag	XSL Equivalent
Header	<!--PWHEADER-->	<xsl:comment>PWHEADER</xsl:comment>
Footer	<!--PWFOOTER-->	<xsl:comment>PWFOOTER</xsl:comment>

CHAPTER 21

Working with Navigation Pages

This chapter provides an overview of navigation pages and their common elements, and discusses how to:

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

Note. Examples of PeopleSoft Pure Internet Architecture pages in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page elements displayed remain the same.

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.

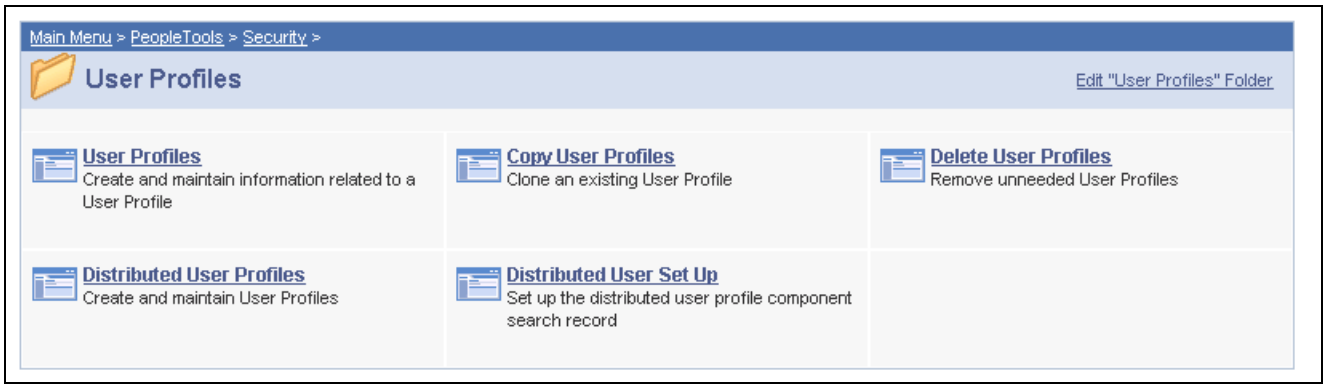
Describing Standard Navigation Pages

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 21, "Working with Navigation Pages," Enabling Standard Navigation Page Display, page 276.

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 23, "Using Navigation Collections," Describing Folder and Link Sequence on Navigation Pages, page 295.

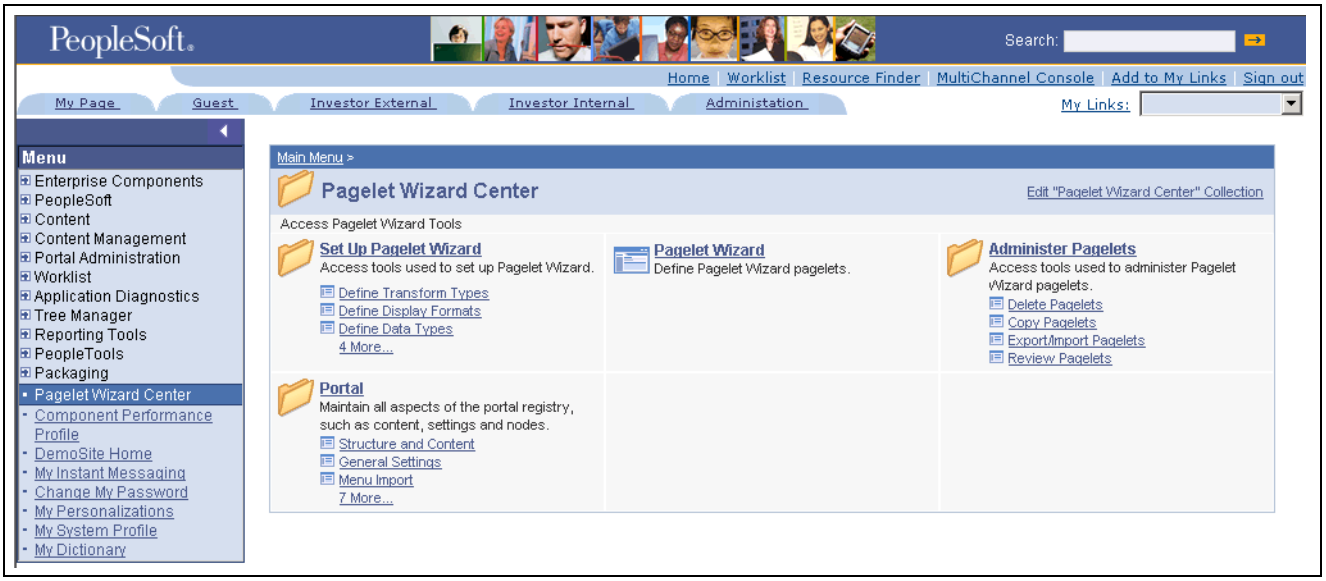
See Chapter 22, "Setting Up Navigation Collection Options," page 279.

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 and 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 Pagelet Wizard Center link in the menu navigation displays the Pagelet Wizard Center custom navigation page as shown here:



Pagelet Wizard 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 22, “Setting Up Navigation Collection Options,” page 279.

Common Elements Used on 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 22, “Setting Up Navigation Collection Options,” page 279.

Edit <folder name> Folder

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 *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portals," Defining Folders

Edit <Navigation Collection name> Collection

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 23, “Using Navigation Collections,” Creating and Maintaining Navigation Collections, page 299.

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

Enterprise PeopleTools 8.45 PeopleBook: Internet Technology, "Administering Portals," Managing General Portal Settings

Enterprise PeopleTools 8.45 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 21, “Working with Navigation Pages,” Describing Custom Navigation Pages, page 275.

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 23, “Using Navigation Collections,” Publishing Navigation Collections, page 318.

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 22, “Setting Up Navigation Collection Options,” page 279.

To enable caching for a 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 21, “Working with Navigation Pages,” Describing Standard Navigation Pages, page 273.

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 22

Setting Up Navigation Collection Options

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

Note. Examples of PeopleSoft Pure Internet Architecture pages in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page elements displayed remain the same.

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 (with attribute name EPPSC_IMAGE)	X
Small (child) image	X	X		
Style sheet	X	X		X
Show images	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 child links	X	X		X
Maximum columns	X	X		
Owner ID	X			X

System options are set on the System Options page.

See Chapter 22, “Setting Up Navigation Collection Options,” Defining Navigation Collection System Options, page 281.

Registry options are set on the Registry Options page.

See Chapter 22, “Setting Up Navigation Collection Options,” Defining Navigation Collection Registry Options, page 285.

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

See *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portals"

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

See Chapter 23, “Using Navigation Collections,” Creating and Maintaining Navigation Collections, page 299.

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 *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portals," Managing General Portal Settings

Defining Navigation Collection Options

In this section, we discuss how to:

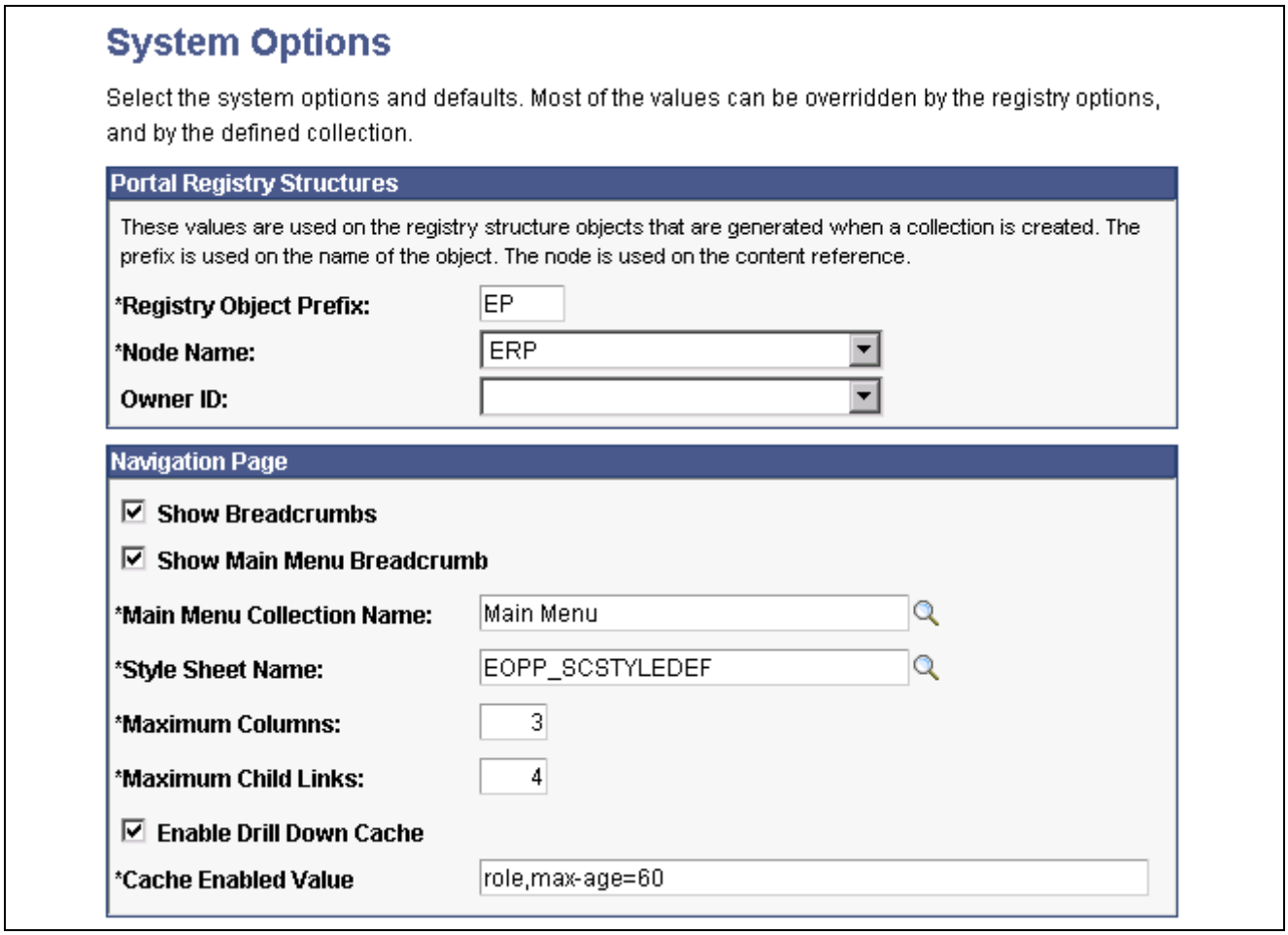
- Define Navigation Collection system options.
- Define Navigation Collection registry options.

Pages Used to Define Navigation Collection Options

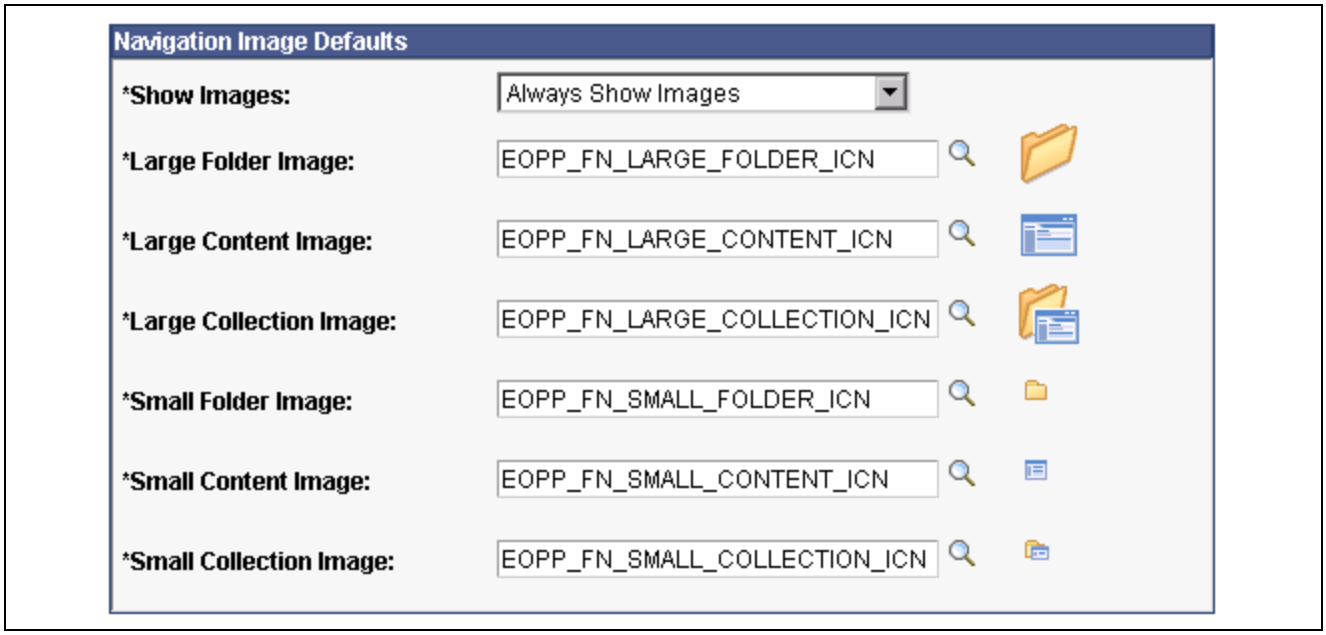
Page Name	Object Name	Navigation	Usage
System Options	EOPP_OPTIONS	Enterprise Components, Portal Utilities, System Options	Define system-level default values for Navigation Collection options. It may be possible to override these defaults at the registry, source reference, and Navigation Collection reference 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 overridden 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 22, "Setting Up Navigation Collection Options," Describing the Navigation Collection Option Default Hierarchy, page 279.

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 *Enterprise PeopleTools 8.45 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 *Enterprise PeopleTools 8.45 PeopleBook: Using PeopleSoft Applications,* "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 21, "Working with Navigation Pages," Enabling Navigation Page Display Caching, page 277.

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*.

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.

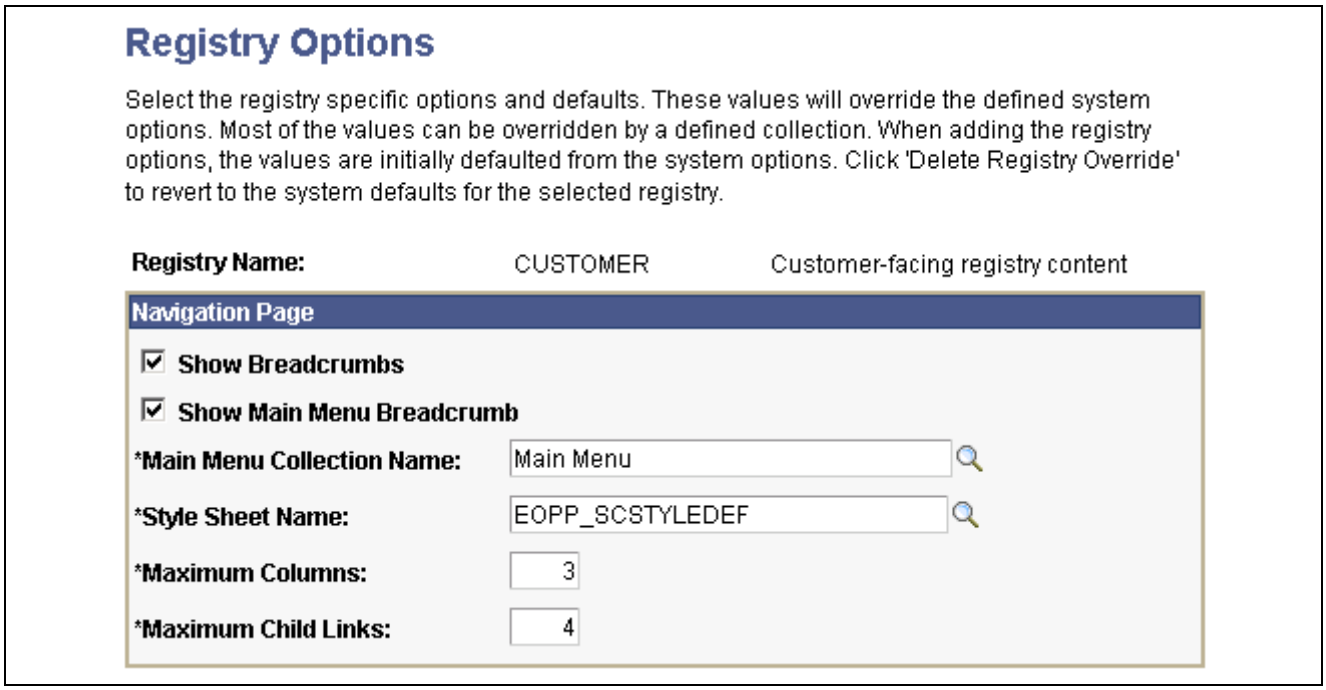
Small Folder Image Select the default image that you want to display for a folder within a folder on a Navigation Collection publication.

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

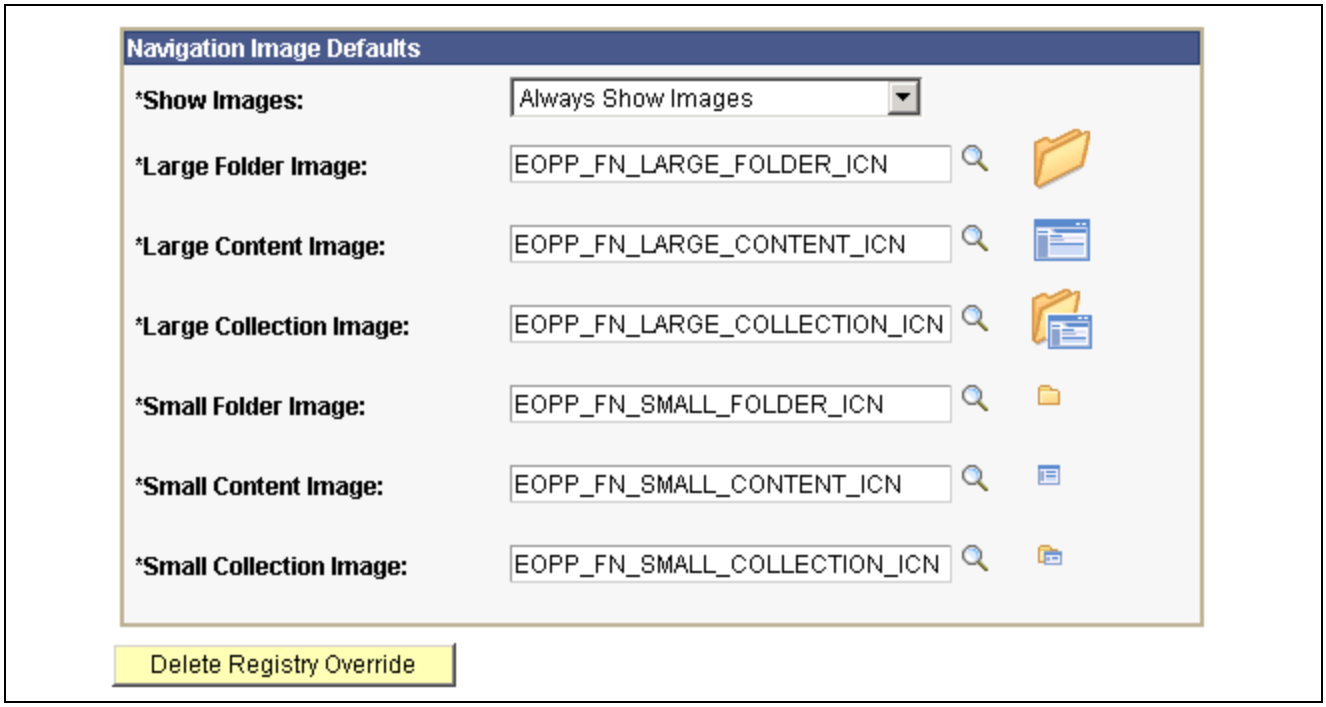
Small Collection Image Select the default image that you want to display for a Navigation Collection within a folder on a Navigation Collection publication.

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 22, "Setting Up Navigation Collection Options," Describing the Navigation Collection Option Default Hierarchy, page 279.

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

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 *Enterprise PeopleTools 8.45 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 *Enterprise PeopleTools 8.45 PeopleBook: Using PeopleSoft Applications,* "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 23

Using Navigation Collections

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

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

Note. Examples of PeopleSoft Pure Internet Architecture pages and pagelets in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages and pagelets may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page and pagelet elements displayed remain the same.

Common Elements Used in This Chapter

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

See *Enterprise PeopleTools 8.45 PeopleBook: Using PeopleSoft Applications,* "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 licensed 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.

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 23, “Using Navigation Collections,” Publishing Navigation Collections, page 318

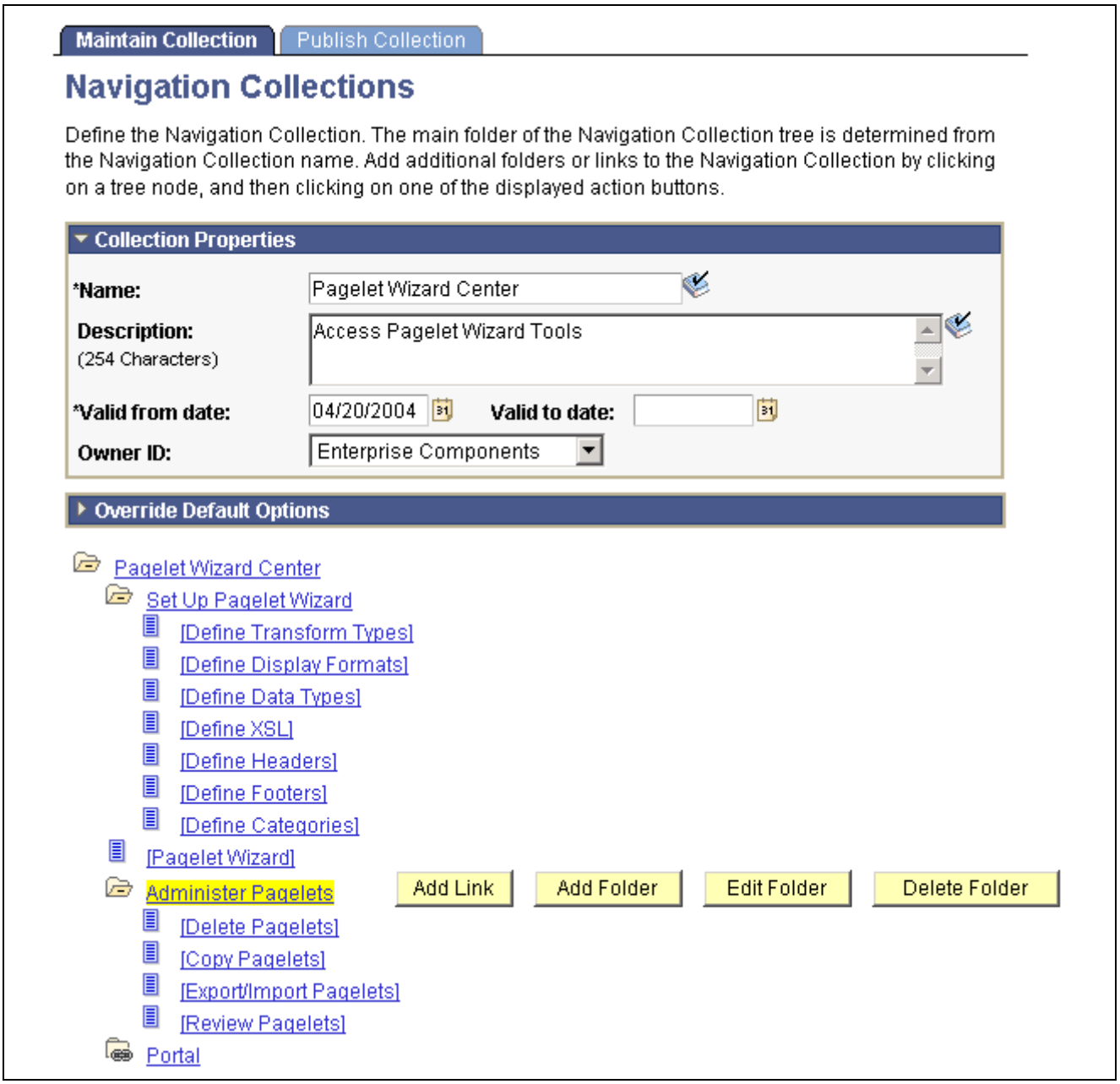
Chapter 24, “Running Portal Utility Processes,” Synchronizing Collection Security, page 323

Describing Navigation Collection Publications

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

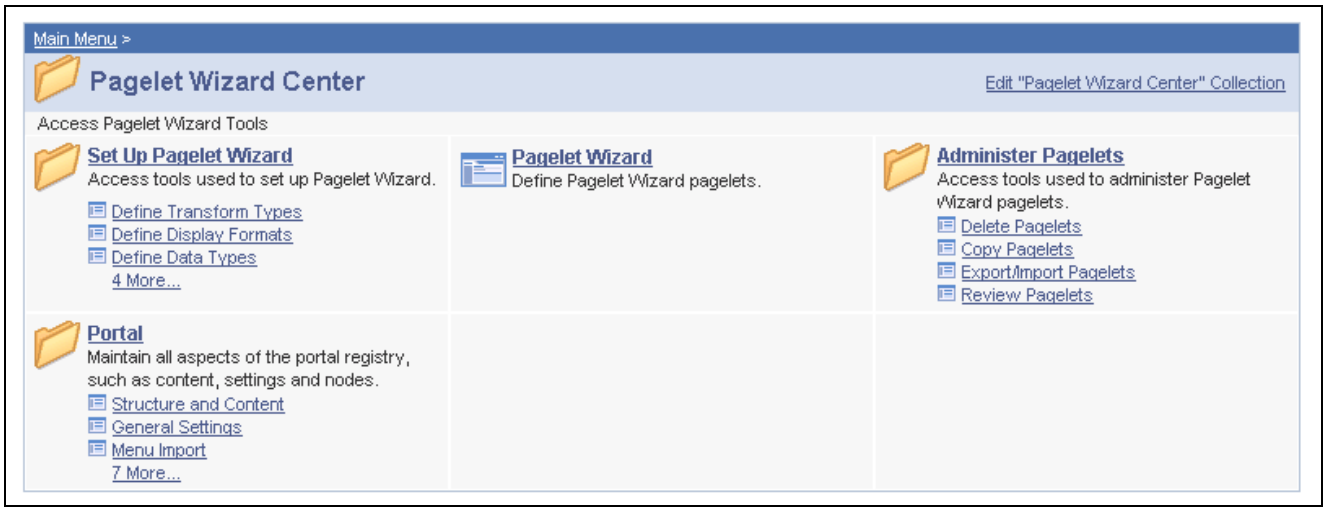
Note. The option to publish or access a Navigation Collection as a navigation pagelet is available if you have licensed PeopleSoft Enterprise Portal or a Portal Pack.

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



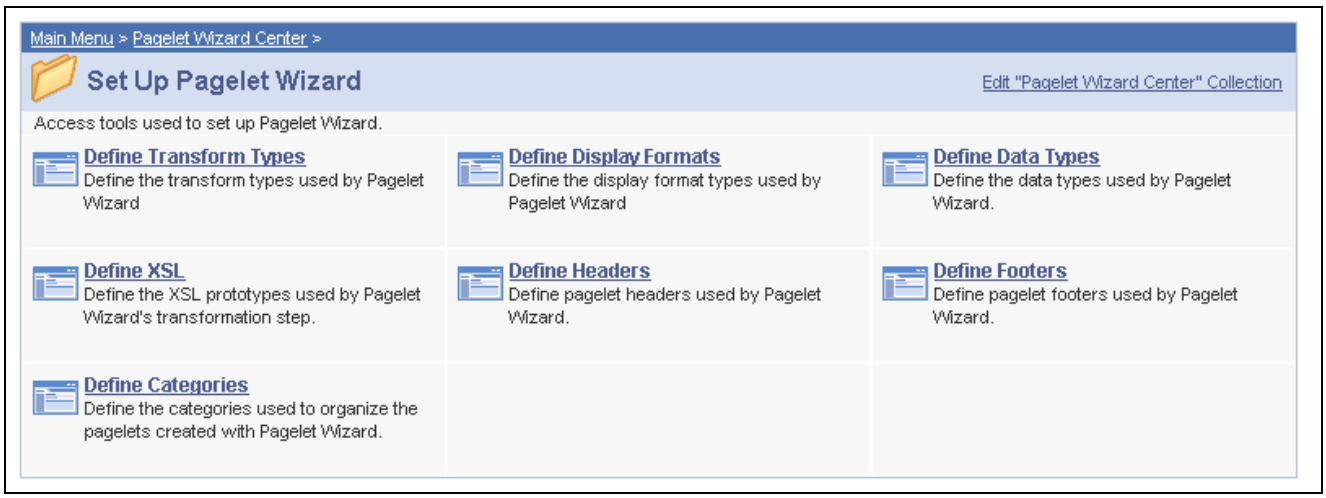
Maintain Collection page

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



Pagelet Wizard Center navigation page

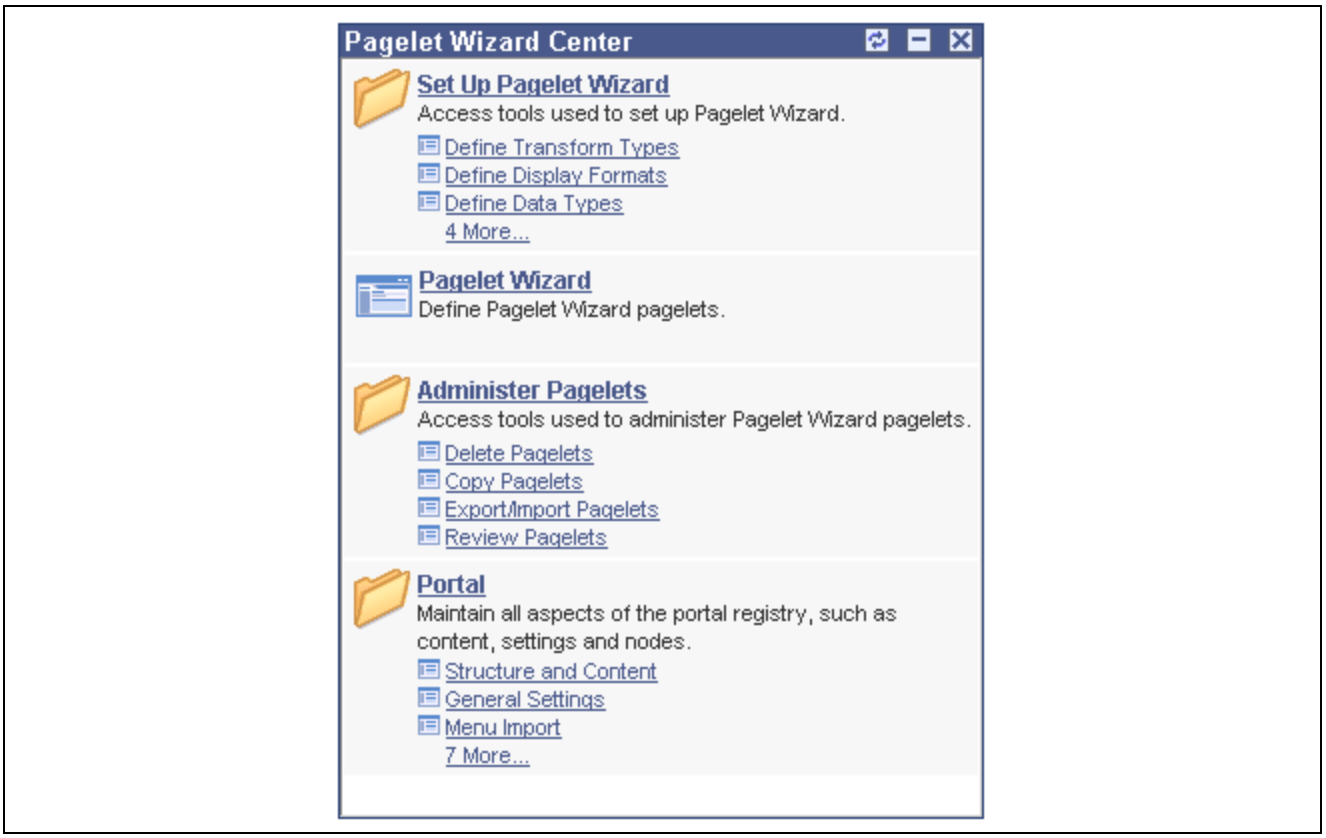
Top-level folders in the Navigation Collection may be transformed into bold and linked section headings that are used to organize the folders and content references they contain, as shown in these examples. However, the appearance of your published Navigation Collections is dependent on its assigned 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.



Set Up Pagelet Wizard 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.



Pagelet Wizard Center navigation pagelet

Note. The option to publish or access a Navigation Collection as a navigation pagelet is available if you have licensed 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 23, "Using Navigation Collections," Publishing Navigation Collections, page 318

Chapter 23, "Using Navigation Collections," Creating and Maintaining Navigation Collections, page 299

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.

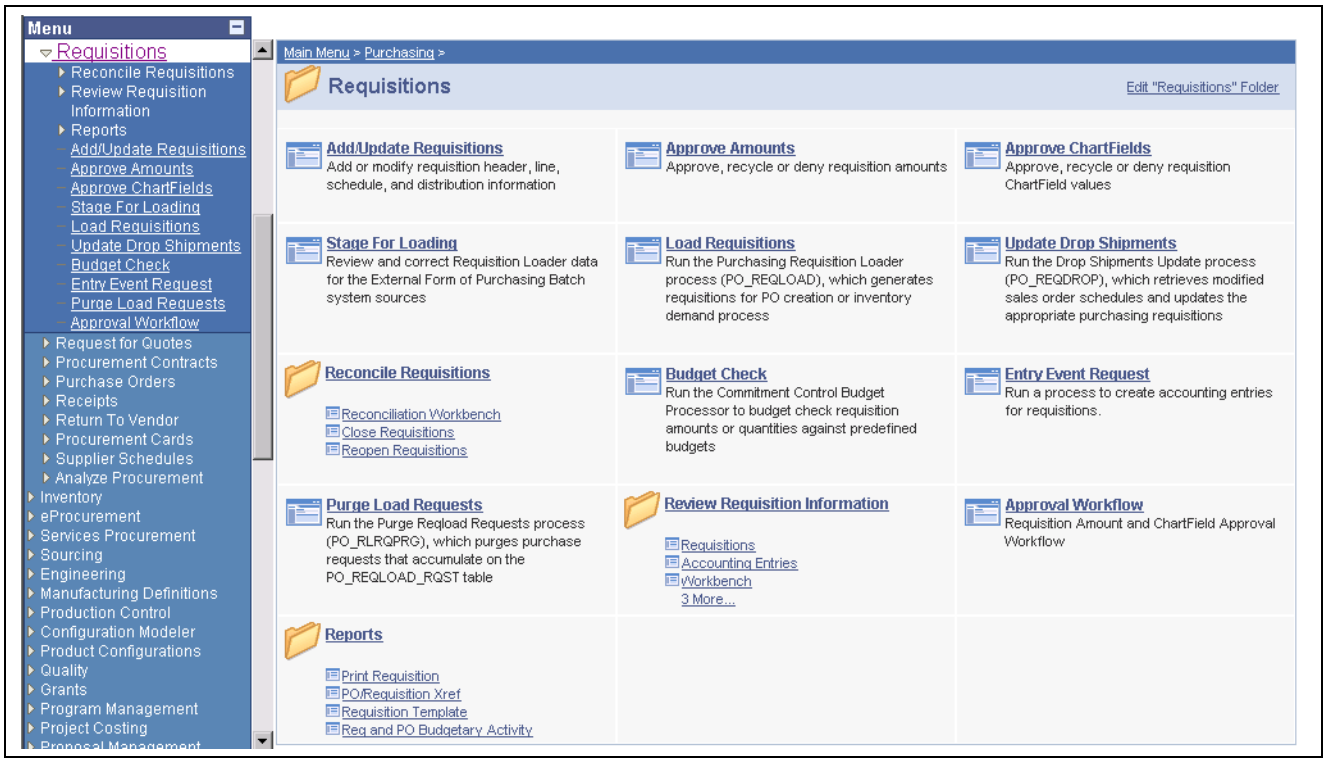
See Chapter 21, “Working with Navigation Pages,” Describing Standard Navigation Pages, page 273.

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

See *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portals," Administering Folders

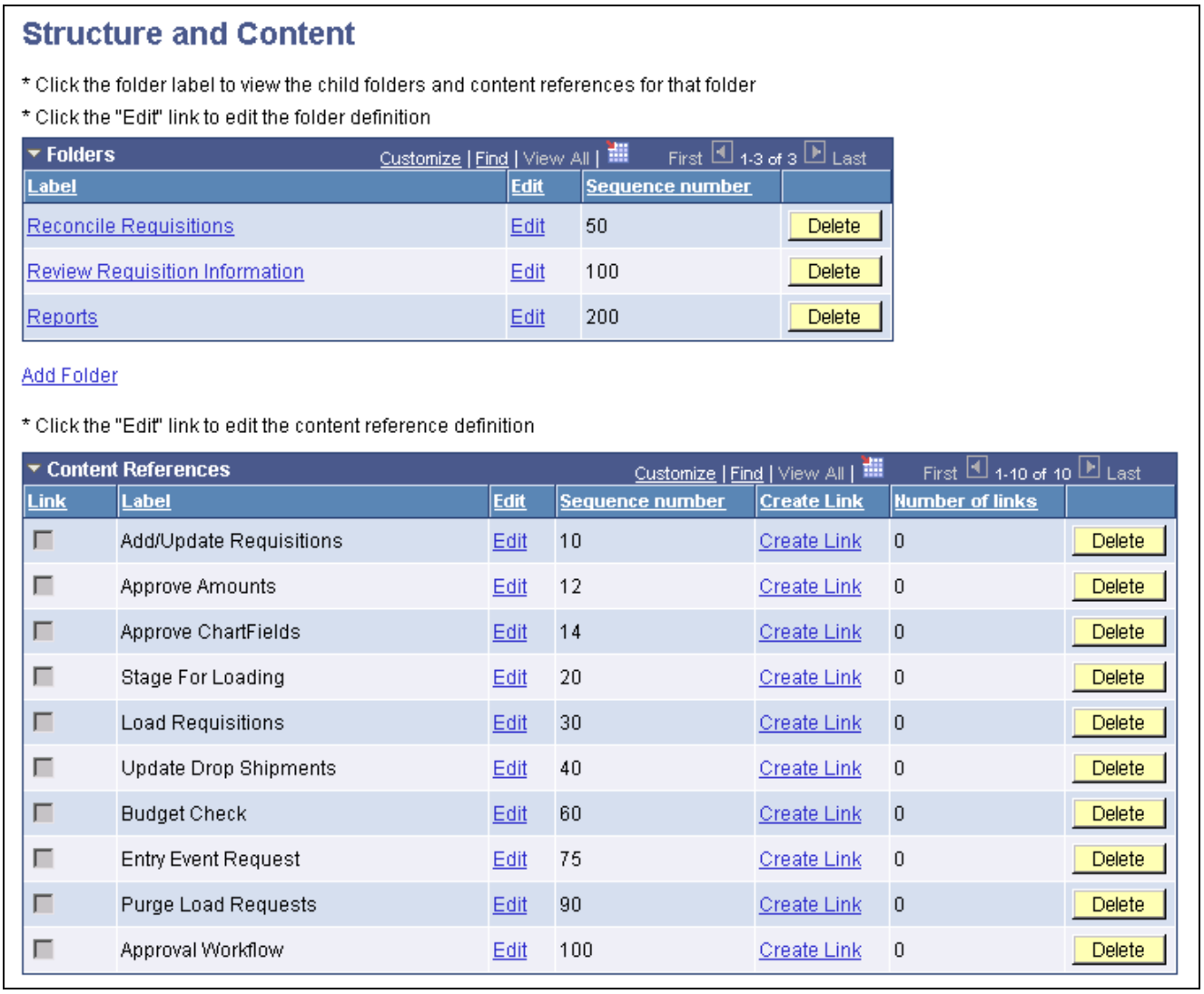
See *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portals," Administering Content References

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 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 21, “Working with Navigation Pages,” Describing Custom Navigation Pages, page 275.

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 23, “Using Navigation Collections,” Editing a Folder, page 311.

See Chapter 23, “Using Navigation Collections,” Editing a Link, page 316.

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.

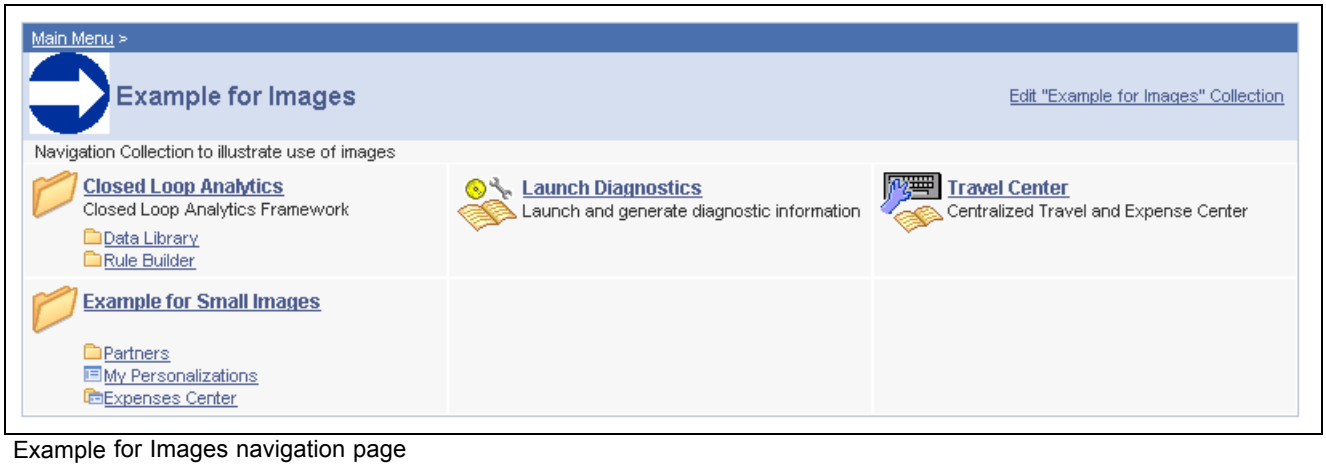
When you save the Navigation Collection, these modifications to the Placement Properties sequences will be updated to the registry.

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 22, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 279.

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 • 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. • Select the Add Collection link on the Find an Existing Collection page. • 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. • 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. • Click the Browse Image Catalog button on the Edit Folder page. • Click the Browse Image Catalog button on the Add Link page. • 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. • Click the Browse Portal Registry button on the Edit Link page. • 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. 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.

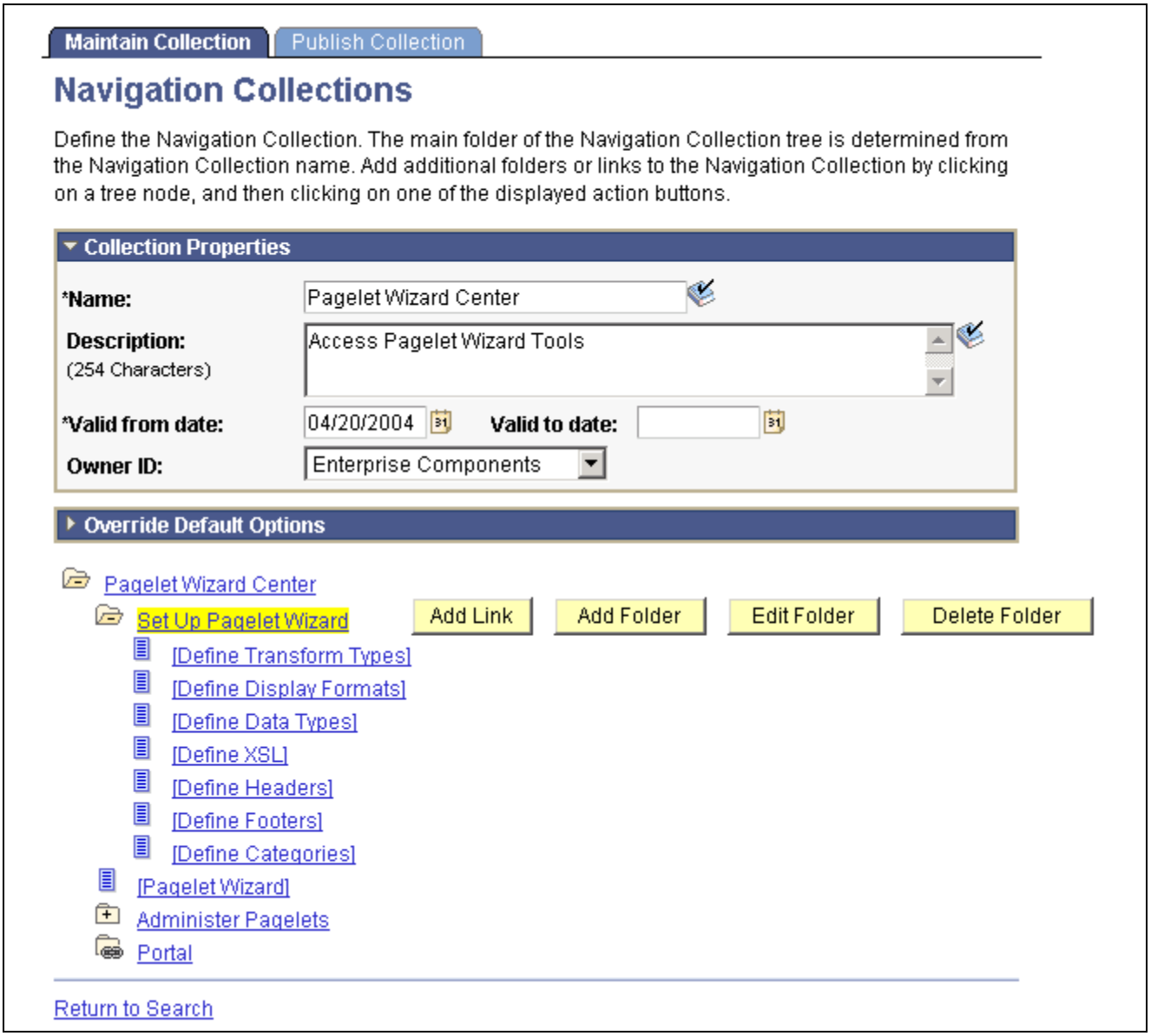
Delete

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 22, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 279.

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 22, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 279.

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 *Enterprise PeopleTools 8.45 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.

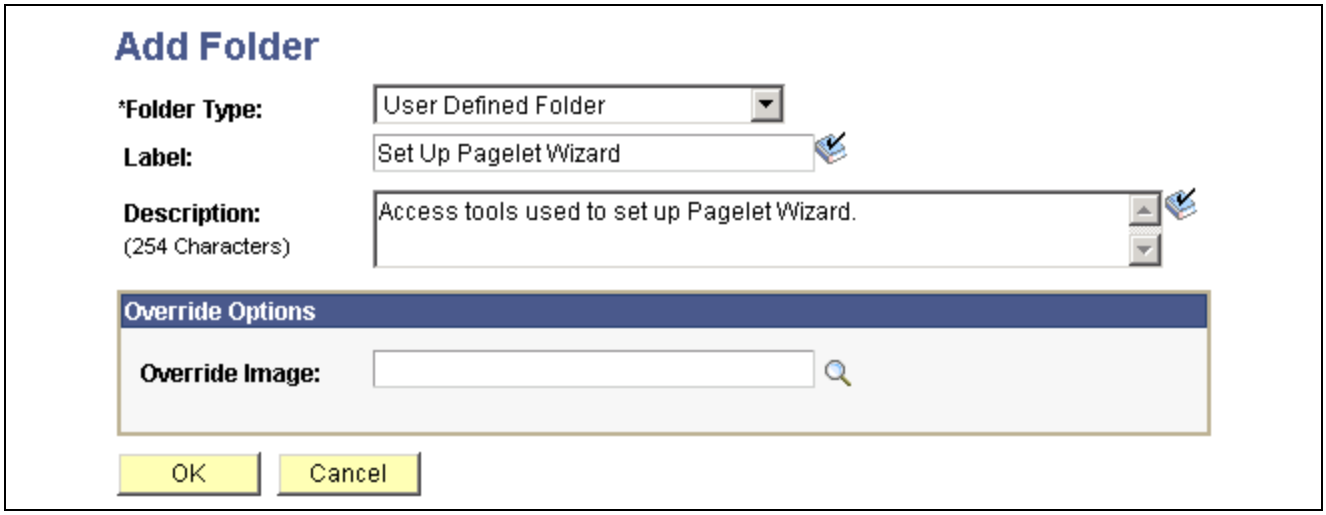
Select the associated folder name link to access options that enable you to edit specific aspects of the folder or delete the folder from the Navigation Collection.

Designates a content reference link in the hierarchy.

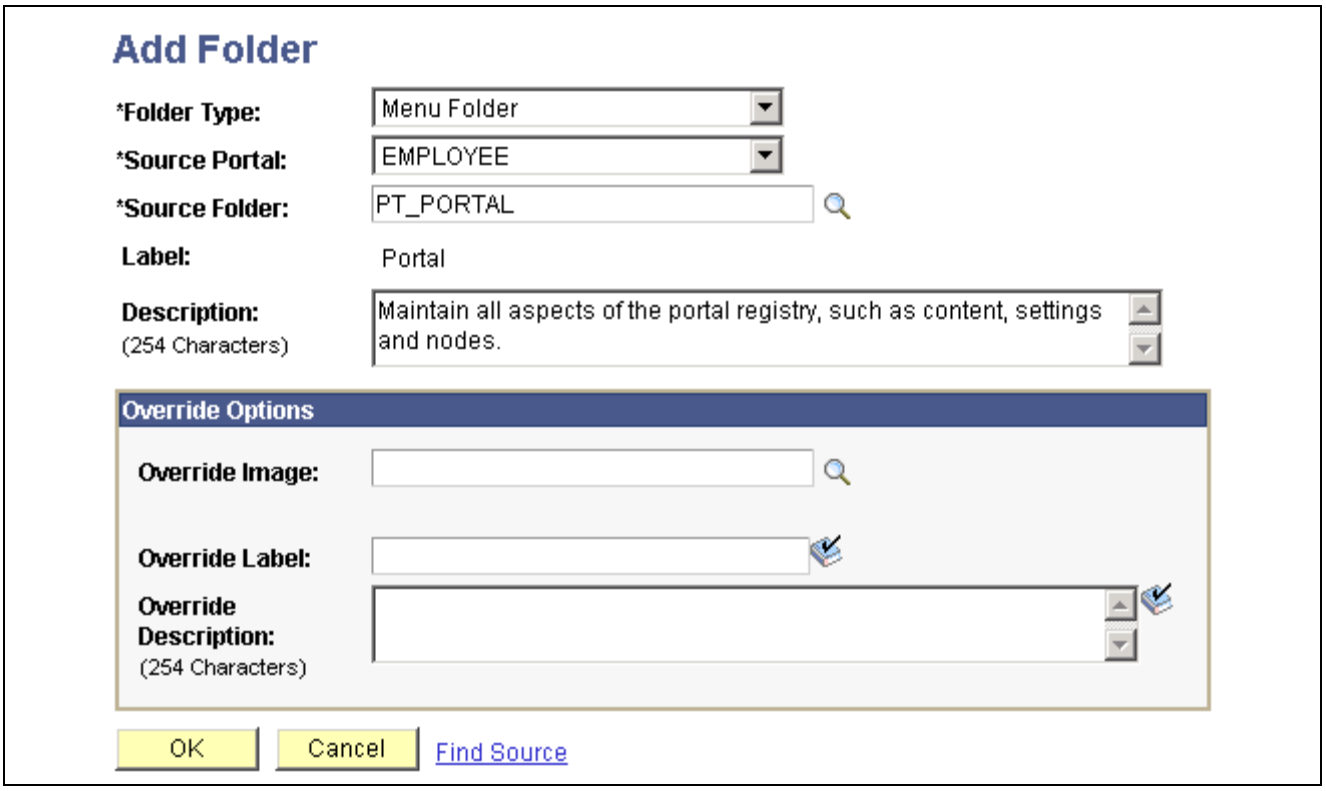
Click to access options that enable you to edit or delete the content reference from the Navigation Collection.

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 *Enterprise PeopleTools 8.45 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 22, "Setting Up Navigation Collection Options," Describing the Navigation Collection Option Default Hierarchy, page 279.

See Chapter 23, "Using Navigation Collections," Describing Navigation Collection Images, page 298.

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.

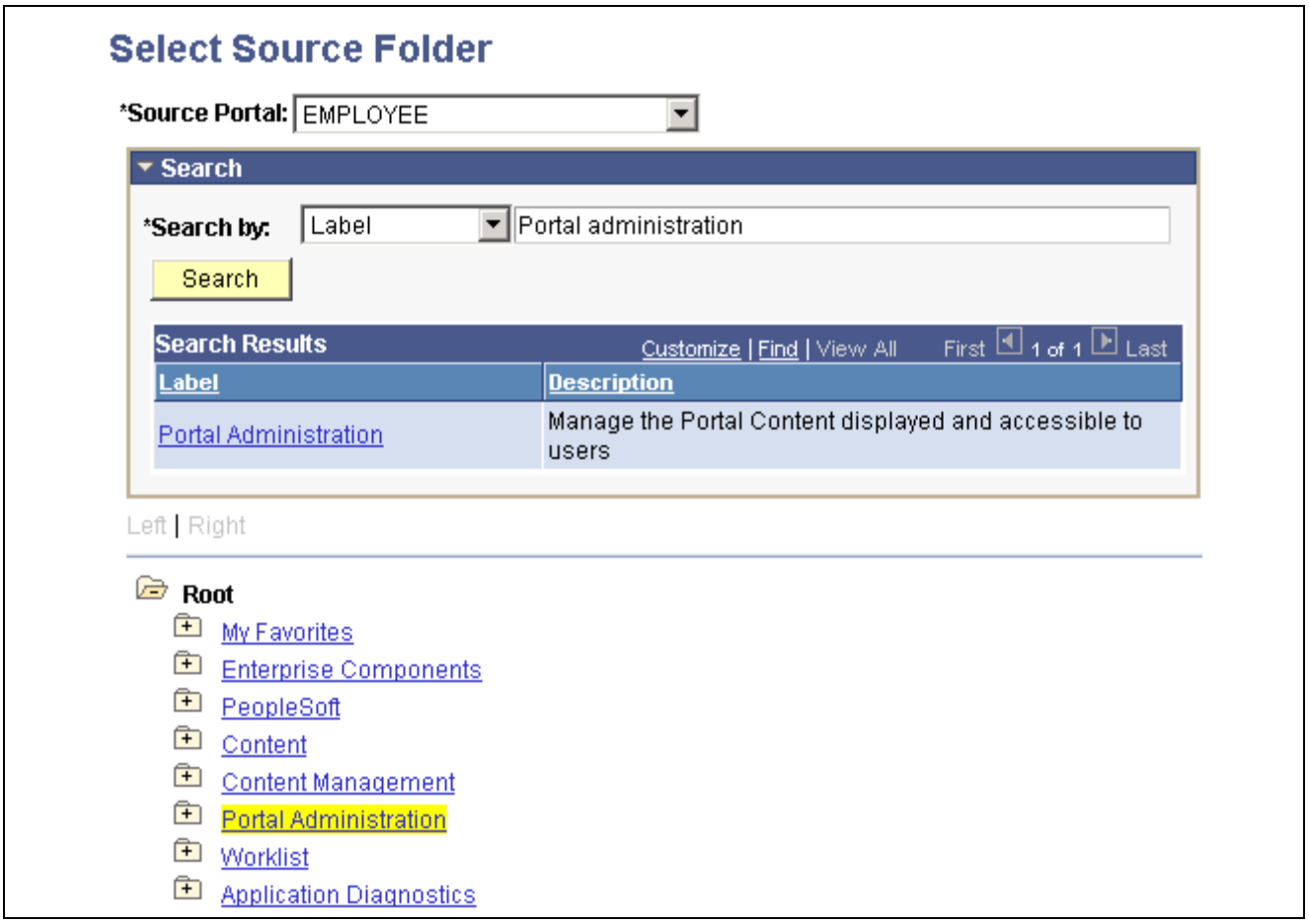
Find Source This link displays for users associated with the PTPT1300 permission list or Portal Administrator role when the Folder Type value is set to *Menu Folder.*

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 *Enterprise PeopleTools 8.45 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 *Enterprise PeopleTools 8.45 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.

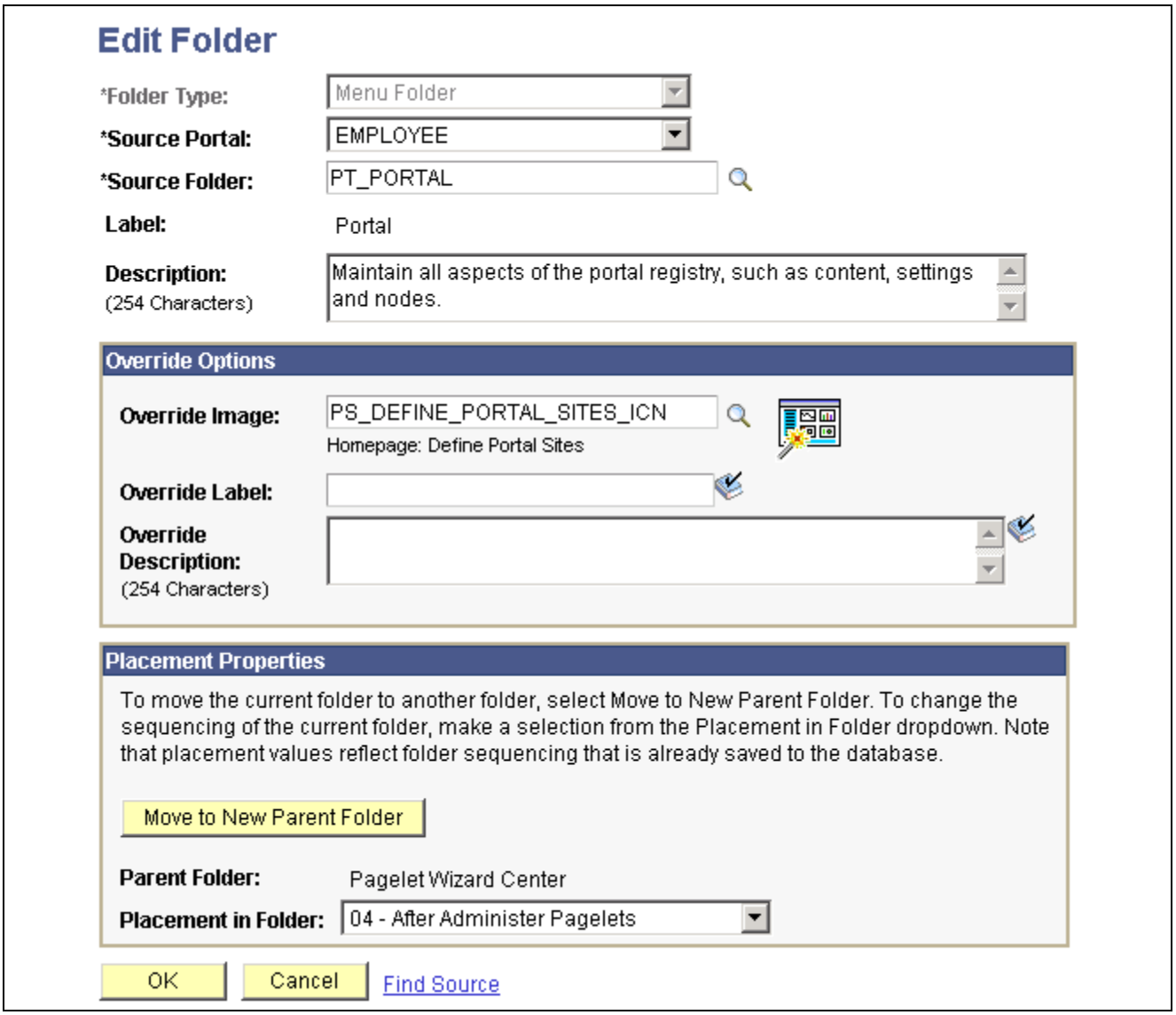
You can further refine your search by selecting the adjacent *begins with* or *contains* option.

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 23, "Using Navigation Collections," Adding a Folder, page 306.

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.

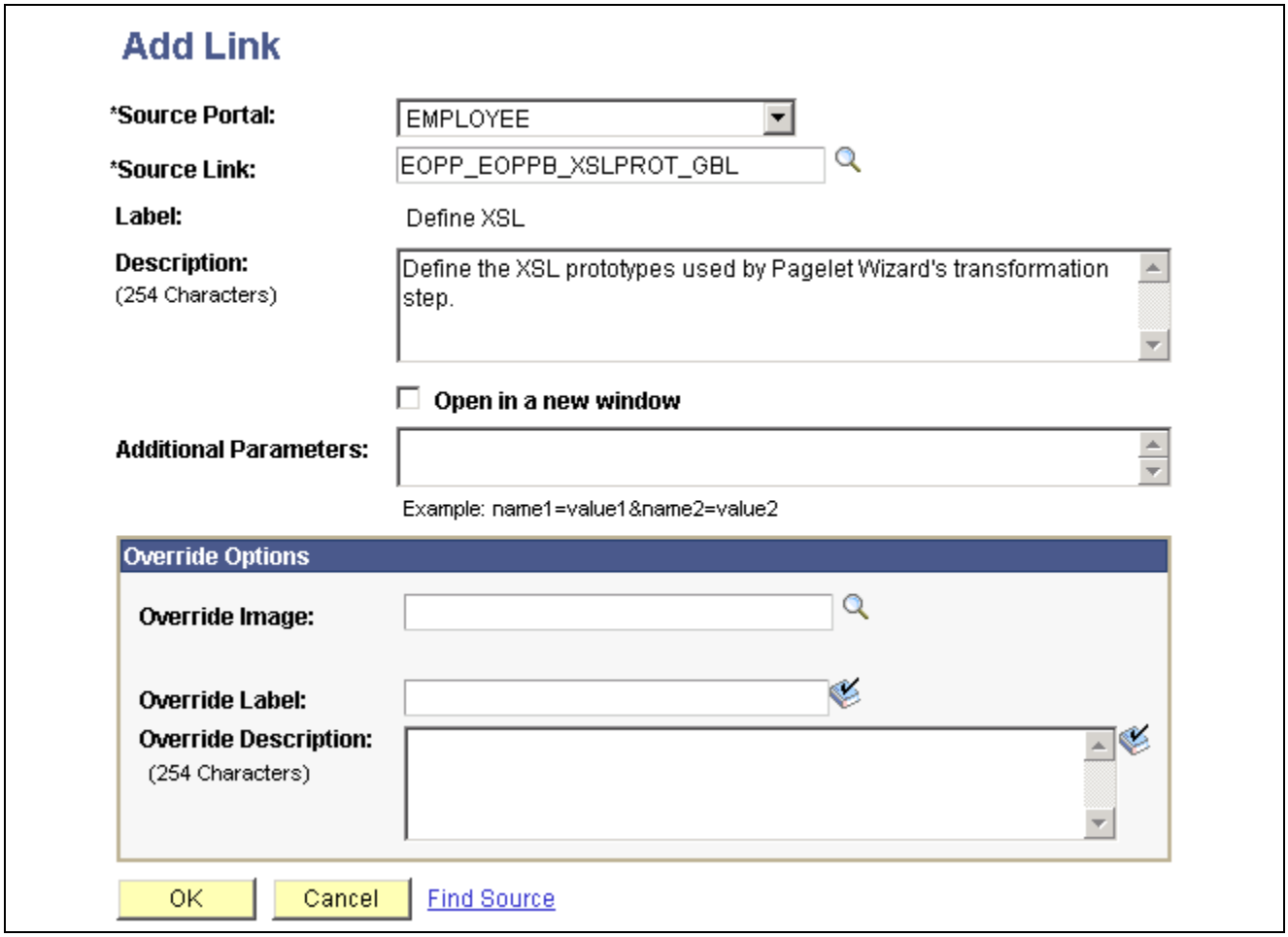
Parent Folder Displays the parent folder of the folder you are editing.

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 *Enterprise PeopleTools 8.45 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 22, “Setting Up Navigation Collection Options,” Describing the Navigation Collection Option Default Hierarchy, page 279.

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.

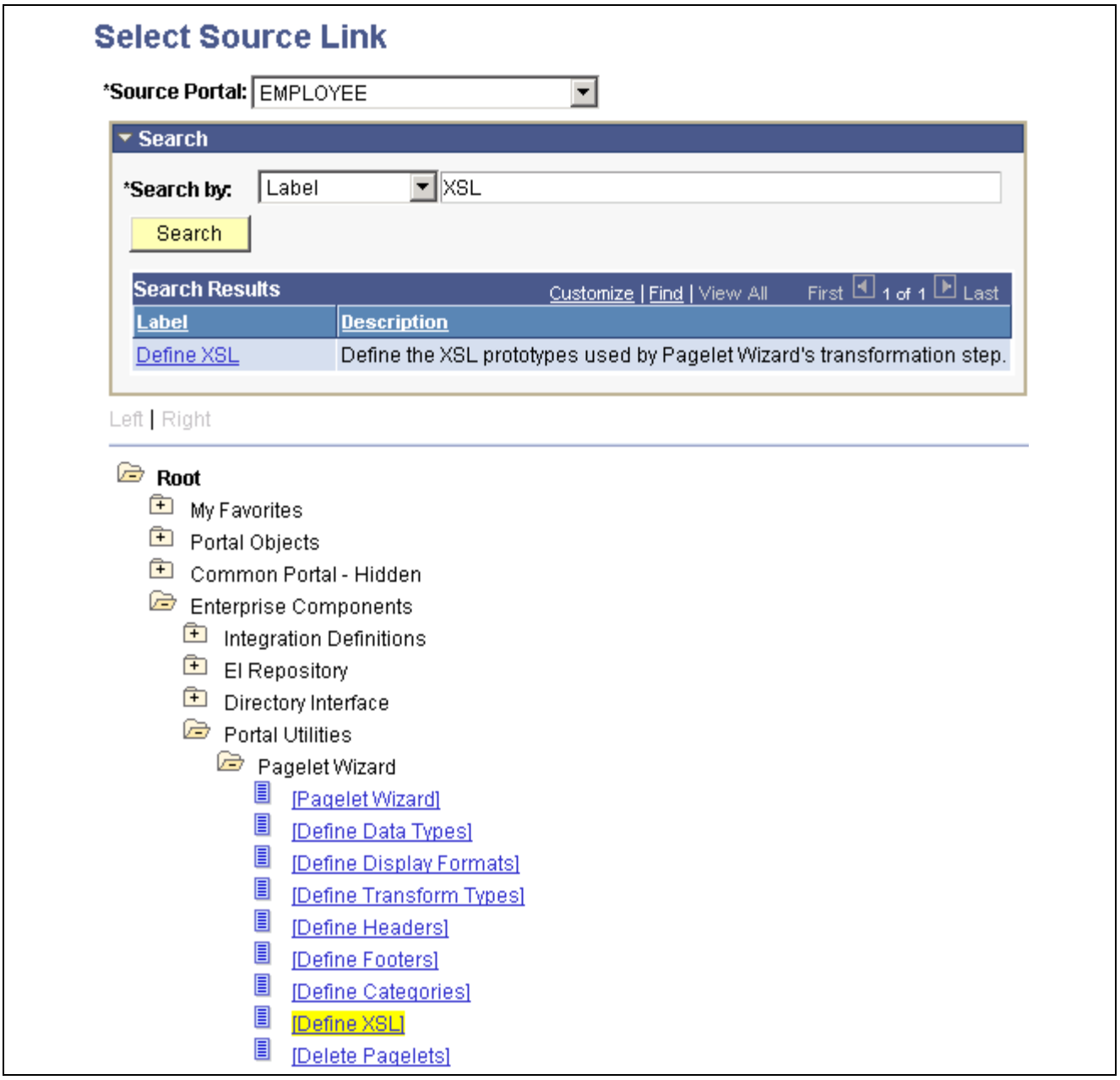
Find Source This link displays for users associated with the PTPT1300 permission list or Portal Administrator Role once a Source Link value has been selected.

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 *Enterprise PeopleTools 8.45 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 *Enterprise PeopleTools 8.45 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

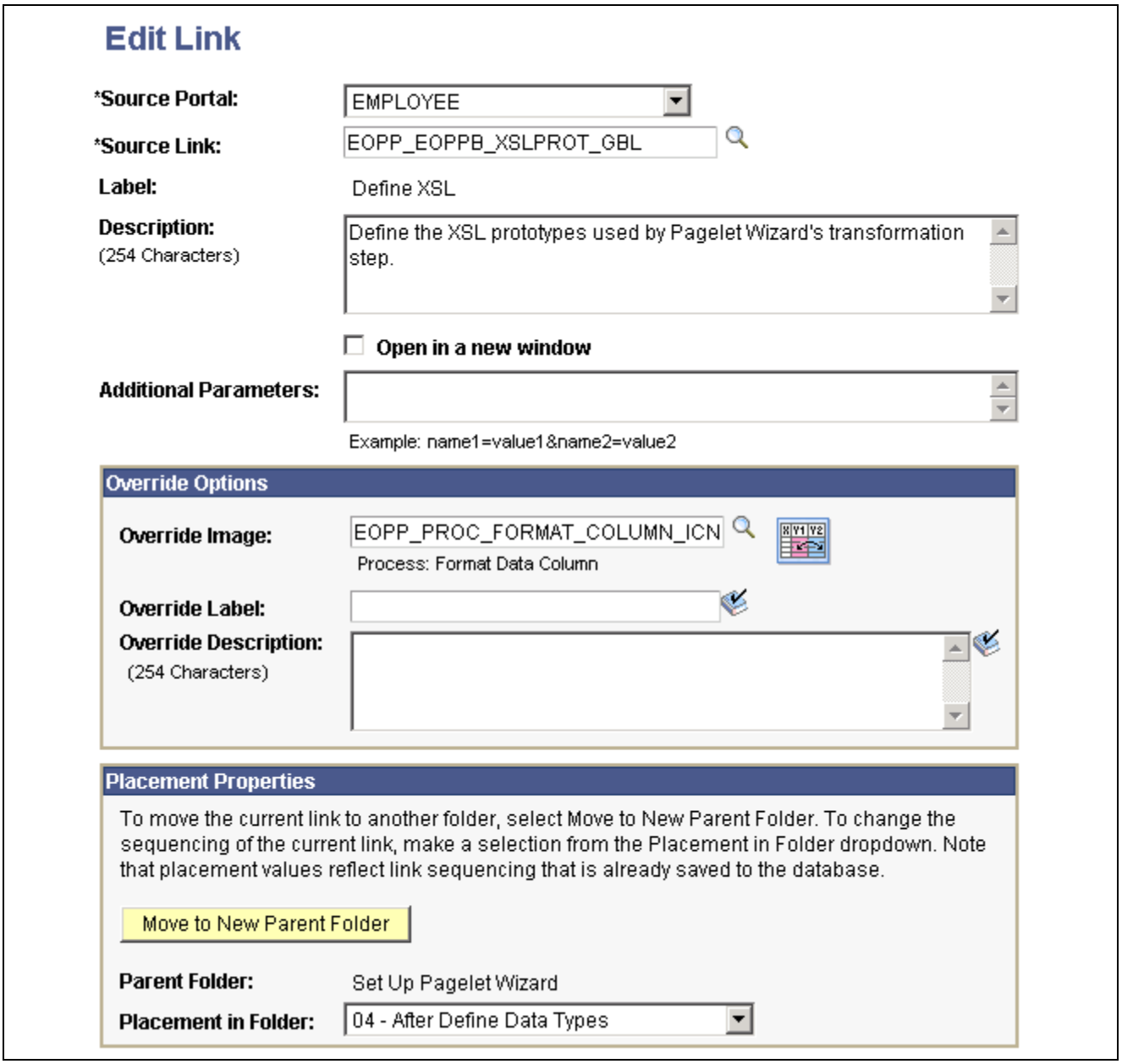
Label Select the linked label text to locate and select the content reference in the portal registry hierarchy displayed at the bottom of the page.

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 23, "Using Navigation Collections," Adding a Link, page 312.

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

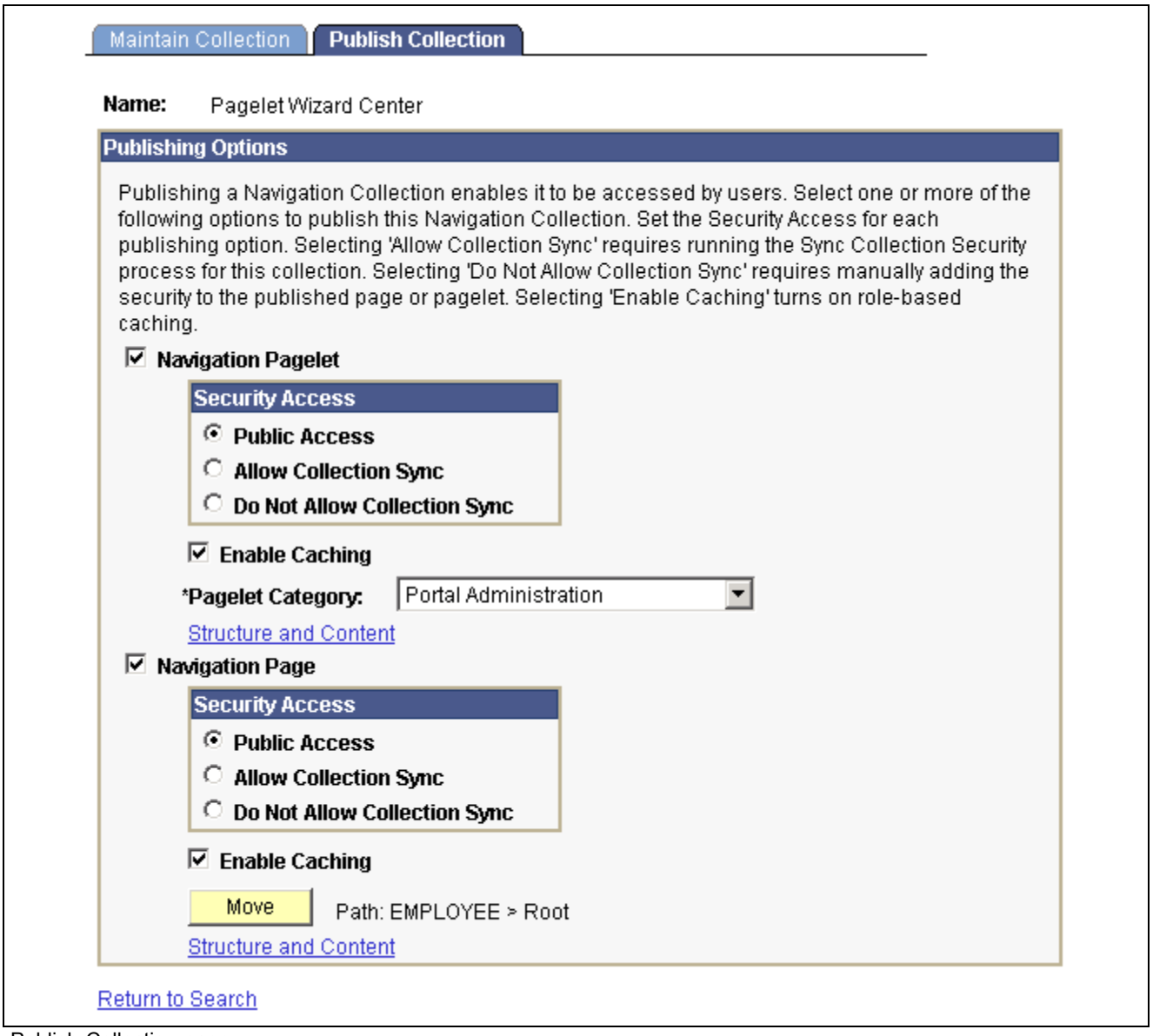
In this section, we discuss how to publish Navigation Collections.

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. **Note.** The option to publish or access a Navigation Collection as a navigation pagelet is available if you have licensed 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 licensed 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 24, "Running Portal Utility Processes," Synchronizing Collection Security, page 323.

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 *Enterprise PeopleTools 8.45 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 *Enterprise PeopleTools 8.45 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 24, "Running Portal Utility Processes," Synchronizing Collection Security, page 323.

See *Enterprise PeopleTools 8.45 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 24, “Running Portal Utility Processes,” Synchronizing Collection Security, page 323.

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 *Enterprise PeopleTools 8.45 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 23, “Using Navigation Collections,” Selecting a Source Folder, page 308.

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 24, “Running Portal Utility Processes,” Synchronizing Collection Security, page 323.

See *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Administering Portals," Administering Content References

CHAPTER 24

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.

Note. Examples of PeopleSoft Pure Internet Architecture pages in this chapter were captured with the PeopleTools default style sheet applied. PeopleSoft Pure Internet Architecture pages may display differently in your database depending on the applied style sheet. However, regardless of the style sheet applied, all page elements displayed remain the same.

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

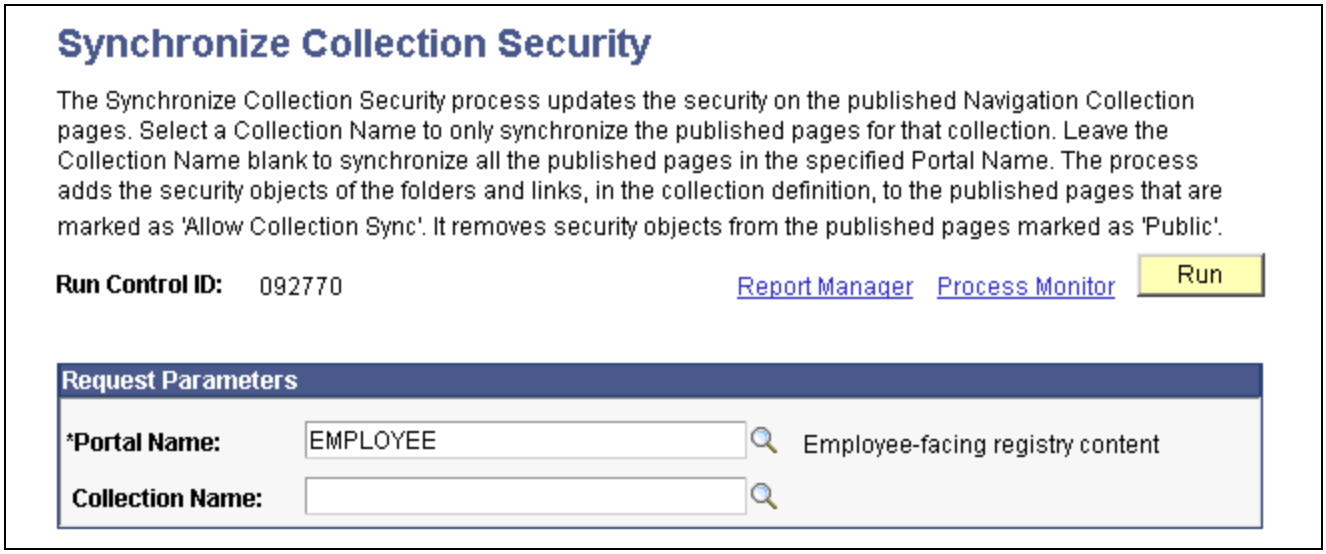
Chapter 23, "Using Navigation Collections," Publishing Navigation Collections, page 318

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 23, “Using Navigation Collections,” Publishing Navigation Collections, page 318

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 *Enterprise PeopleTools 8.45 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

Enterprise PeopleTools 8.45 PeopleBook: System and Server Administration, "Getting Started with System and Server Administration," PeopleSoft Configuration Manager

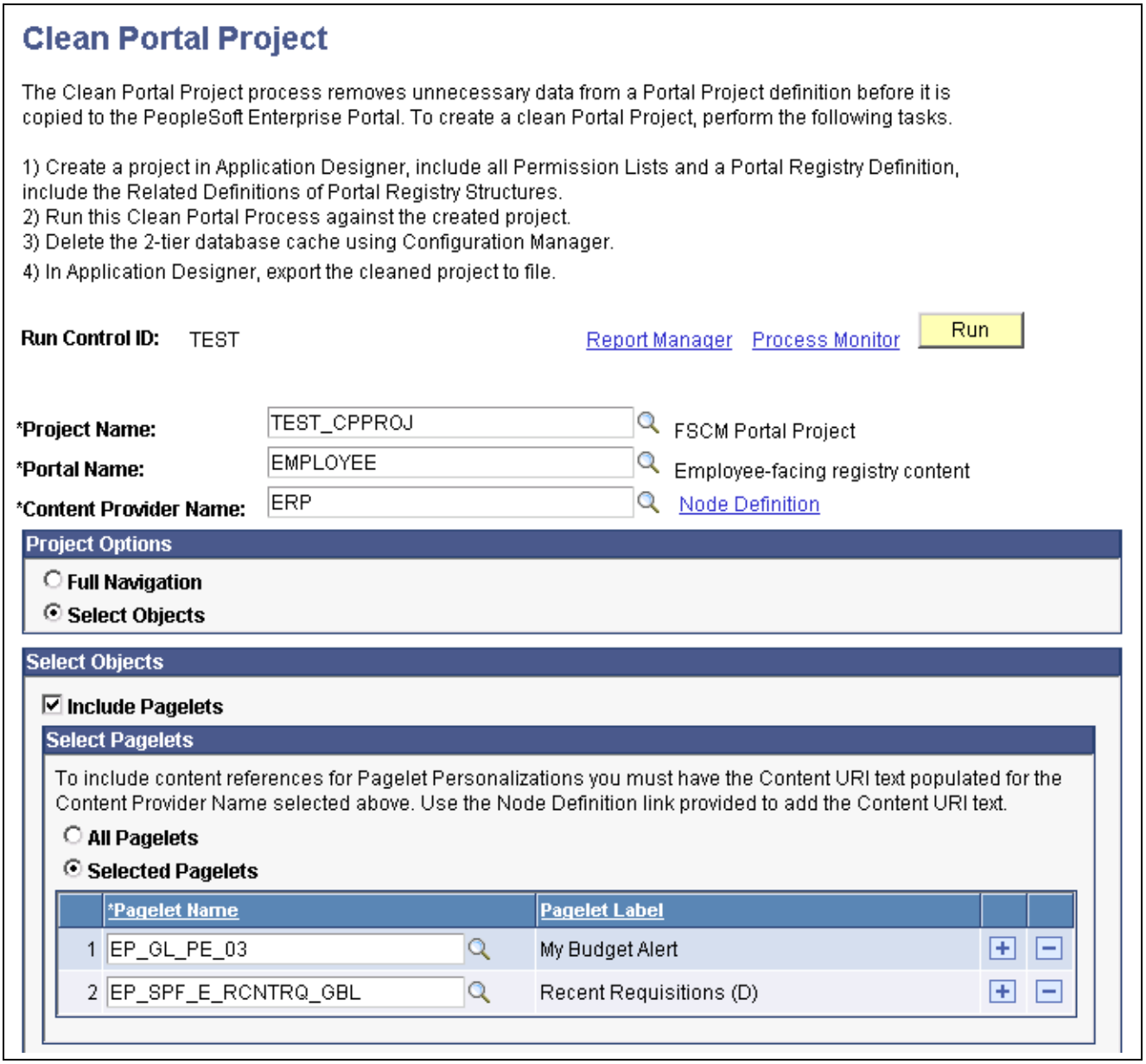
Enterprise PeopleTools 8.45 PeopleBook: PeopleSoft Application Designer

Page Used to Clean Portal Projects

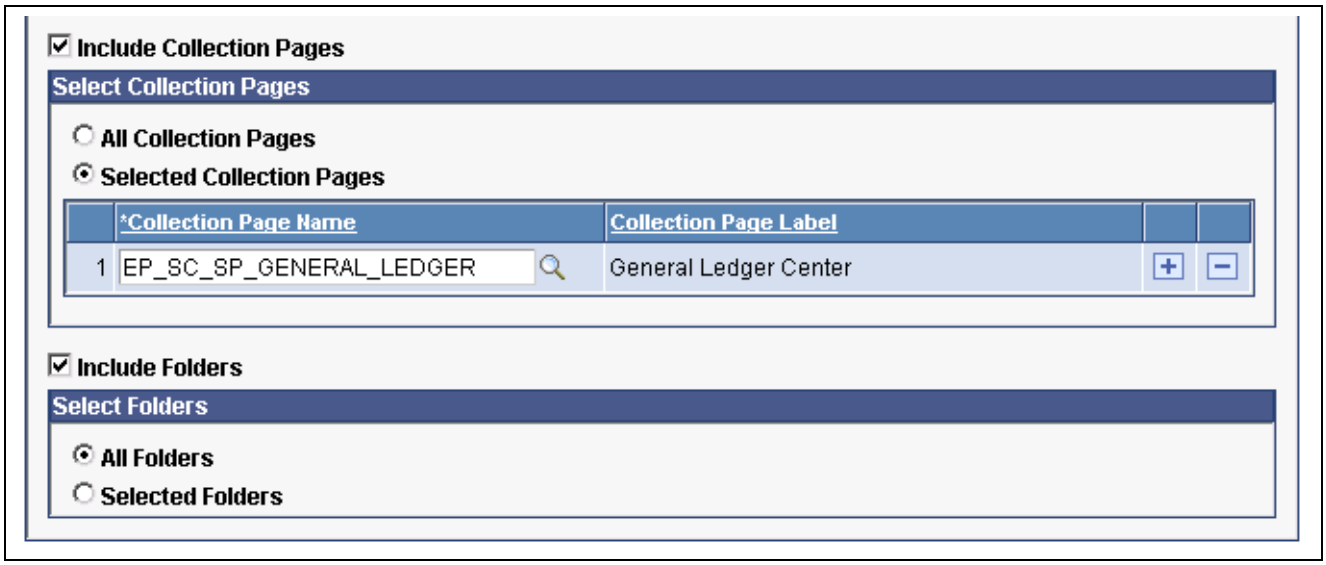
Page Name	Object Name	Navigation	Usage
Clean Portal Project	EOPP_CPPROJ_RUN	Enterprise Components, Portal Utilities, Clean Portal Project	Set run control parameters for the Clean Portal Project process.

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 *Enterprise PeopleTools 8.45 PeopleBook: Internet Technology,* "Configuring the Portal Environment," 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 6

Enterprise Integration

Chapter 25
Understanding Enterprise Integration

Chapter 26
Understanding Integration Points

Chapter 27
Activating Messaging Integration Points

Chapter 28
Assigning Publishing Rules

Chapter 29
Using the Error Handling Utility

Chapter 30
Using the Effective Date Publish Utility

Chapter 31
Using the Flat File Utility

Chapter 32
Using the XML Schema Utility

CHAPTER 25

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 26

Understanding Integration Points

This chapter discusses integration points.

Note. The term *integration point* is synonymous with the term *Enterprise Integration Point (EIP)* and may appear interchangeably in this documentation.

Overview of Integration Points

Provided by a PeopleSoft application, an integration point is an interface that is used to communicate with other PeopleSoft or external applications. An integration point provides integration details for PeopleSoft applications, including which technologies are involved, technology details, integration broker information for messaging, and how different integration points are related.

The integration point consists of data rules for the applications that it supports. The integration points that are delivered with PeopleSoft provide generic functionality so that they can be adapted for use with as many programs as possible.

An integration point can be implemented by using different technologies available in PeopleTools, such as messaging, component interfaces, business interlinks, XML links, and electronic data interchange (EDI).

Integration points can be associated with or used by application groups. An application group is a logical grouping of applications that use an integration point 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 integration point is owned by at least one application, but can be used in multiple applications. Therefore, if an application sends an integration point, and another application can receive the same integration point, the two systems should be interoperable, assuming the data structure and the rules of the integration point are implemented the same in both places.

However, sometimes two applications might use the same integration point but implement it in different ways. For example, one application that uses the Customer integration point 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 integration point can be a part of multiple application groups. For example, the Department Table integration point may be used by a number of application groups, including PeopleSoft Human Resources, CRM, and General Ledger.

See Interactive Services Repository on the PeopleSoft Customer Connection website.

CHAPTER 27

Activating Messaging Integration Points

This chapter discusses how to:

- Set up PeopleSoft Messaging integration points.
- Set up related languages.
- Examine related-language messaging scenarios.

Setting Up PeopleSoft Messaging Integration Points

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 integration points, and for information on how and with what applications to use them, consult the relevant application PeopleBook.

Pages Used to Set Up PeopleSoft Messaging Integration Points

Page Name	Object Name	Navigation	Usage
Batch Publish Rules	EO_MSGPUBATCH	Enterprise Components, Integration Definitions, Batch Publish Rules	Set up publication rules. 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, Integration Definitions, 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 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.

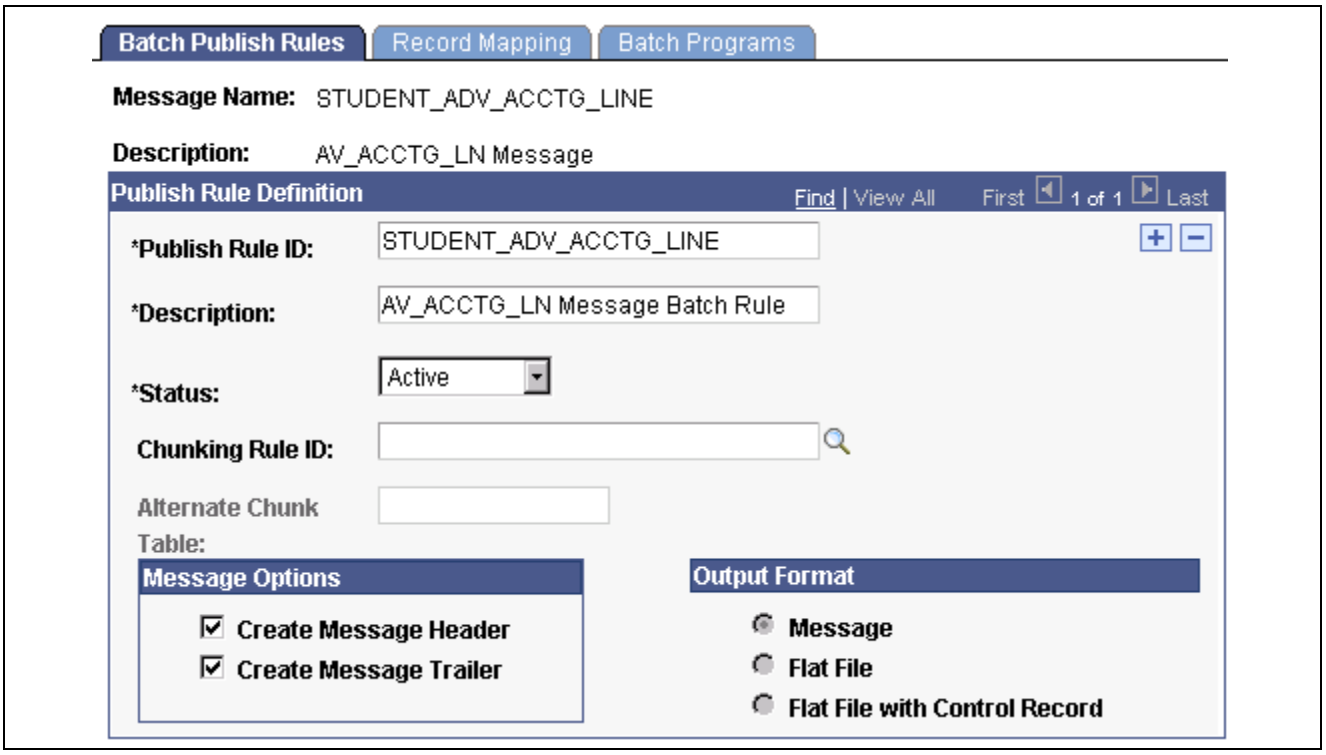
   The Open Object dialog box appears.
3. In the Definition menu, select *Message.*
4. Enter the message name in the Name field.
5. Click Open.

   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

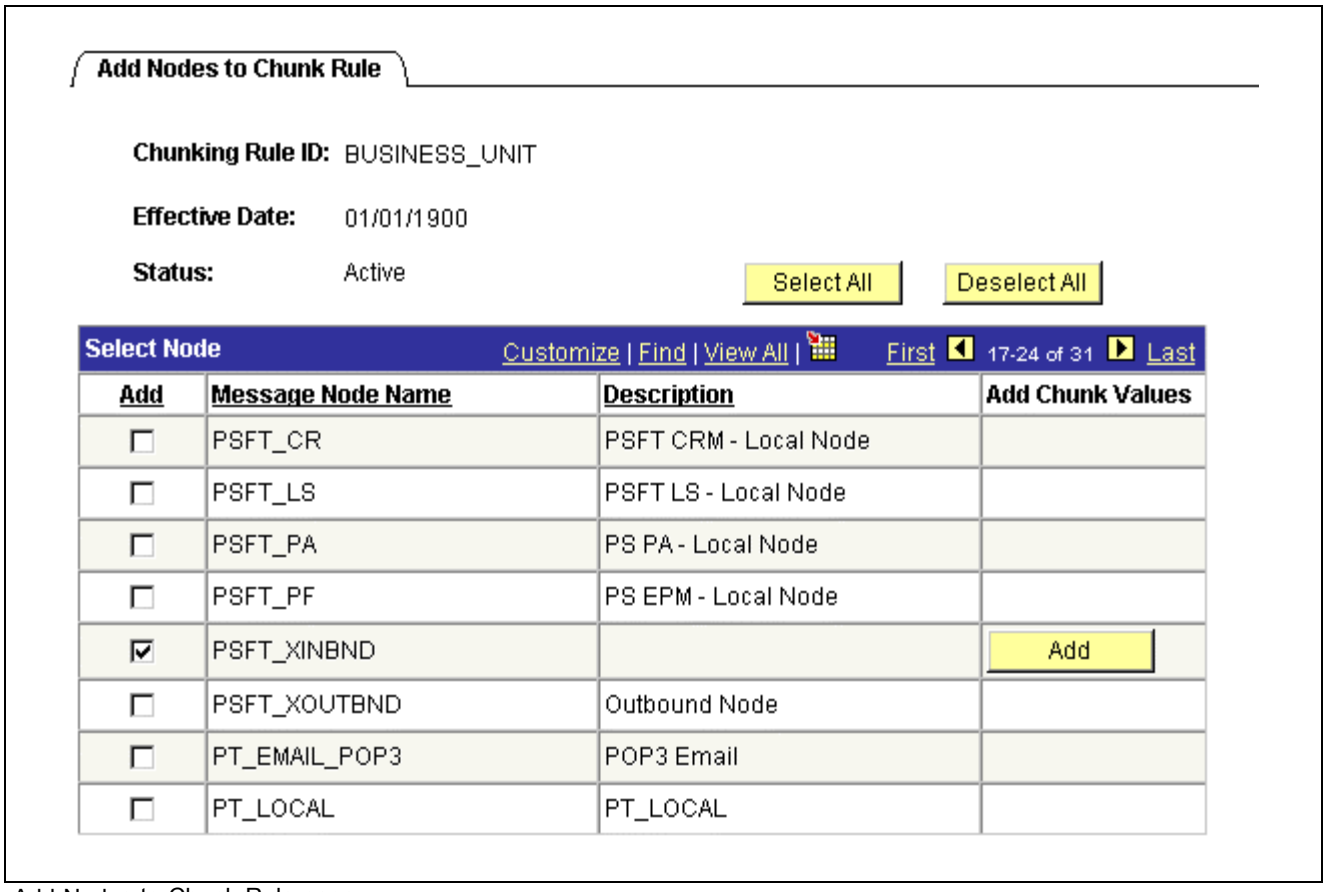
Many PeopleSoft systems rely on a message header and message trailer to trigger subscription PeopleCode to discard old table data and insert the new incoming data. As a general rule, all FullSync messages should use a header and trailer. Sync messages don't need headers and trailers.

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 28, “Assigning Publishing Rules,” Assigning Business Units or SetIDs to a Chunking Rule, page 362.

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

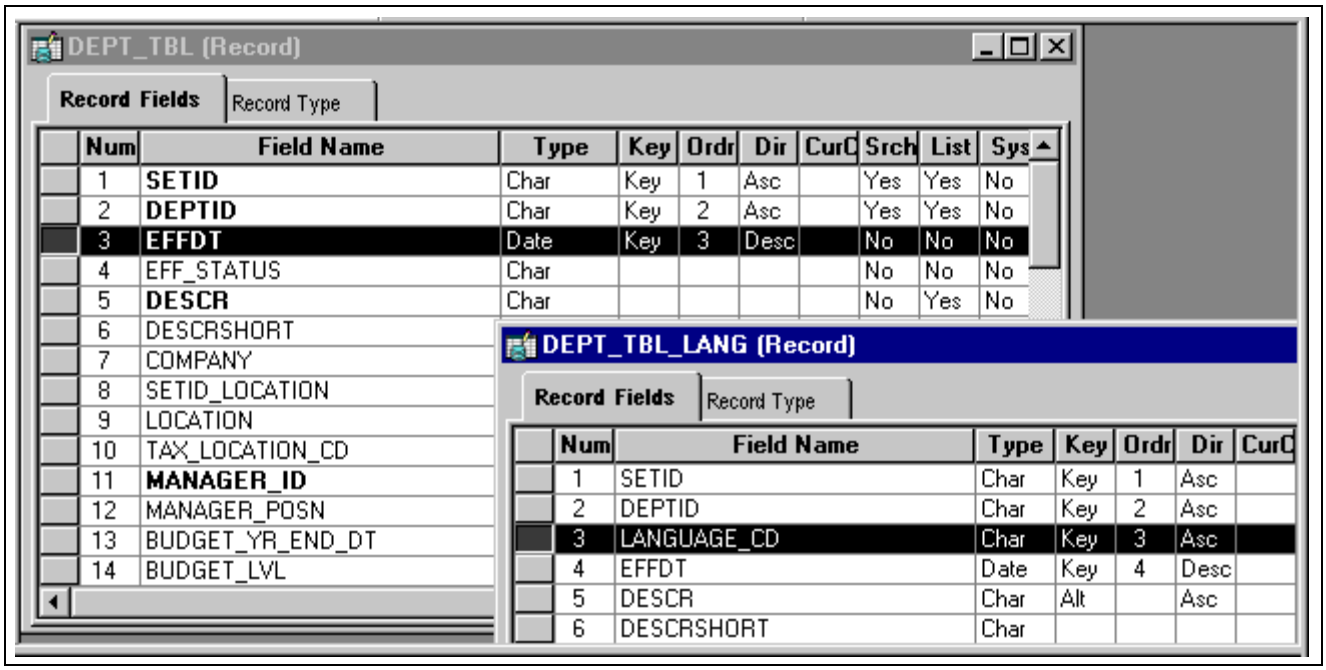
This section provides an overview of related language tables and related language guidelines for PeopleSoft Messaging and discusses how to:

- 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 28

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:

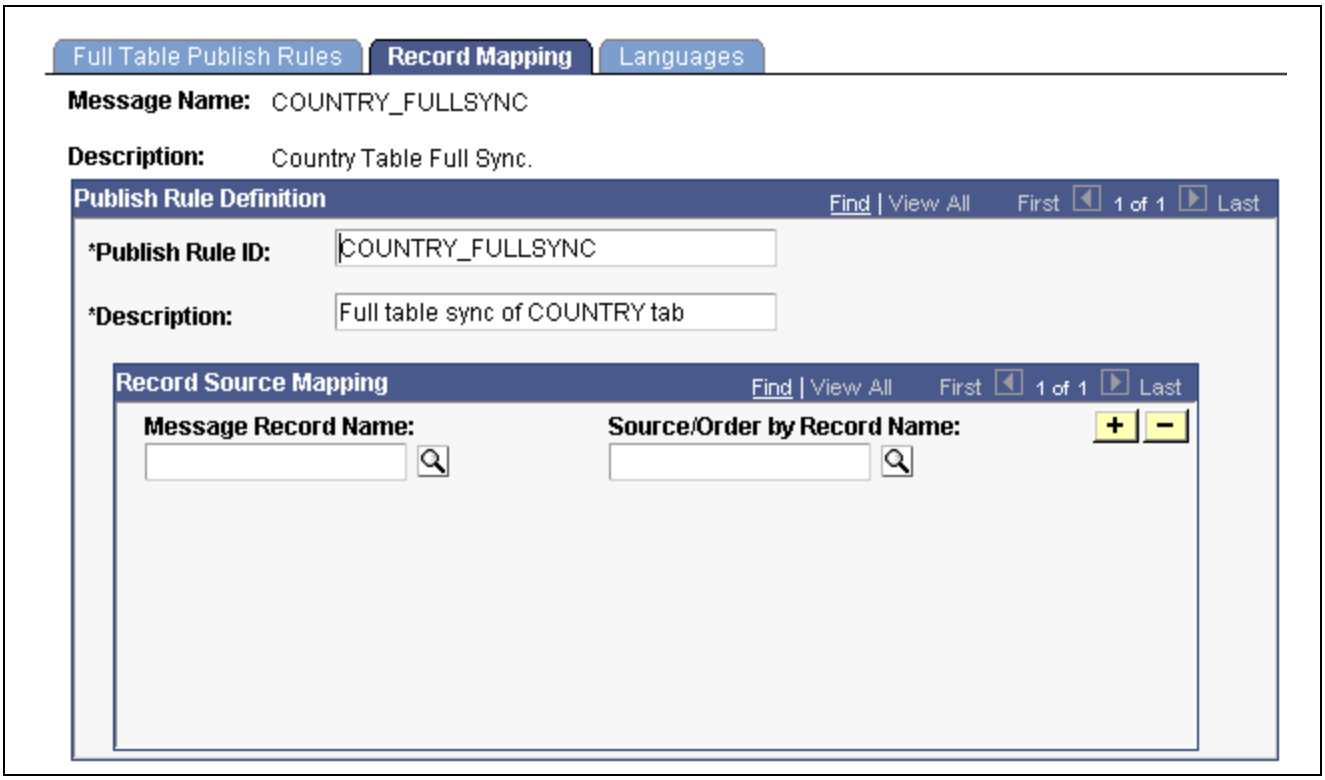
- Message (default value).
- Flat file.
- Flat file with control record.

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:

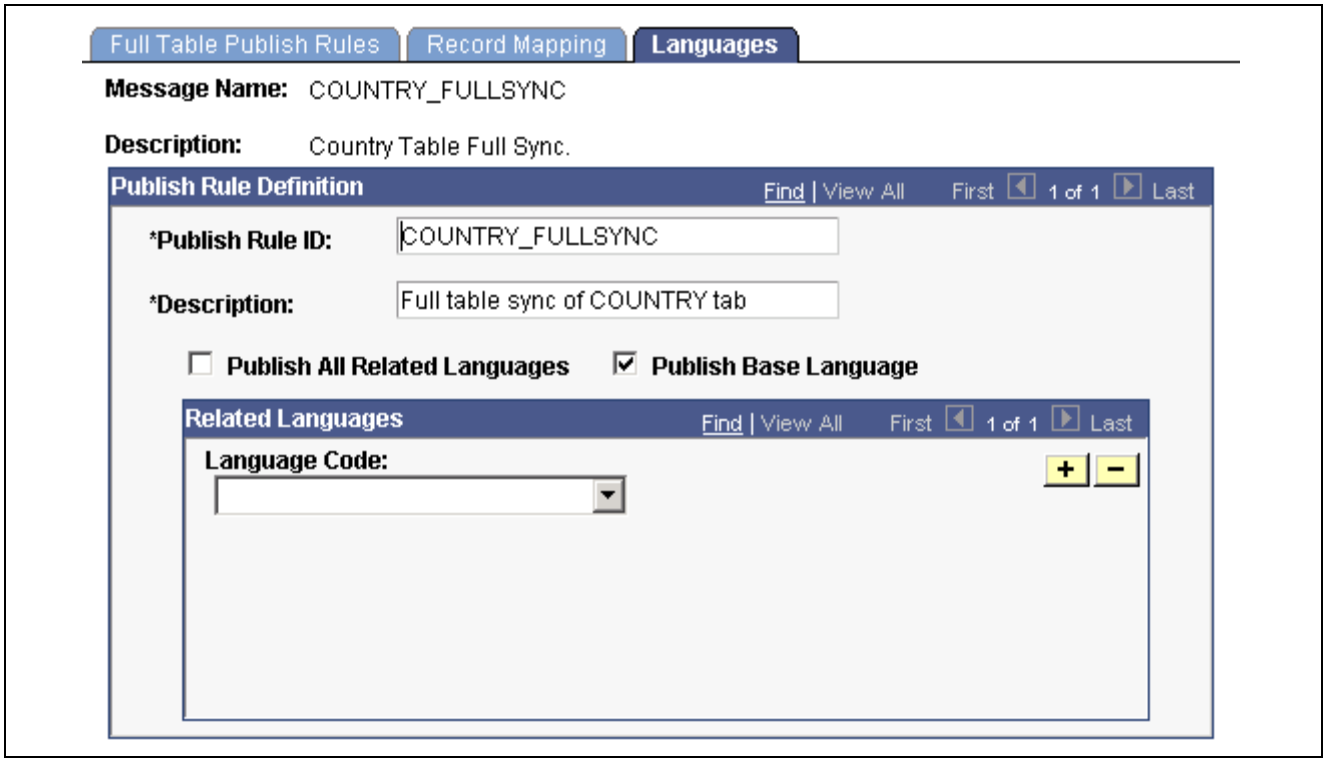
- The published data comes from staging tables or temporary tables.

  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.

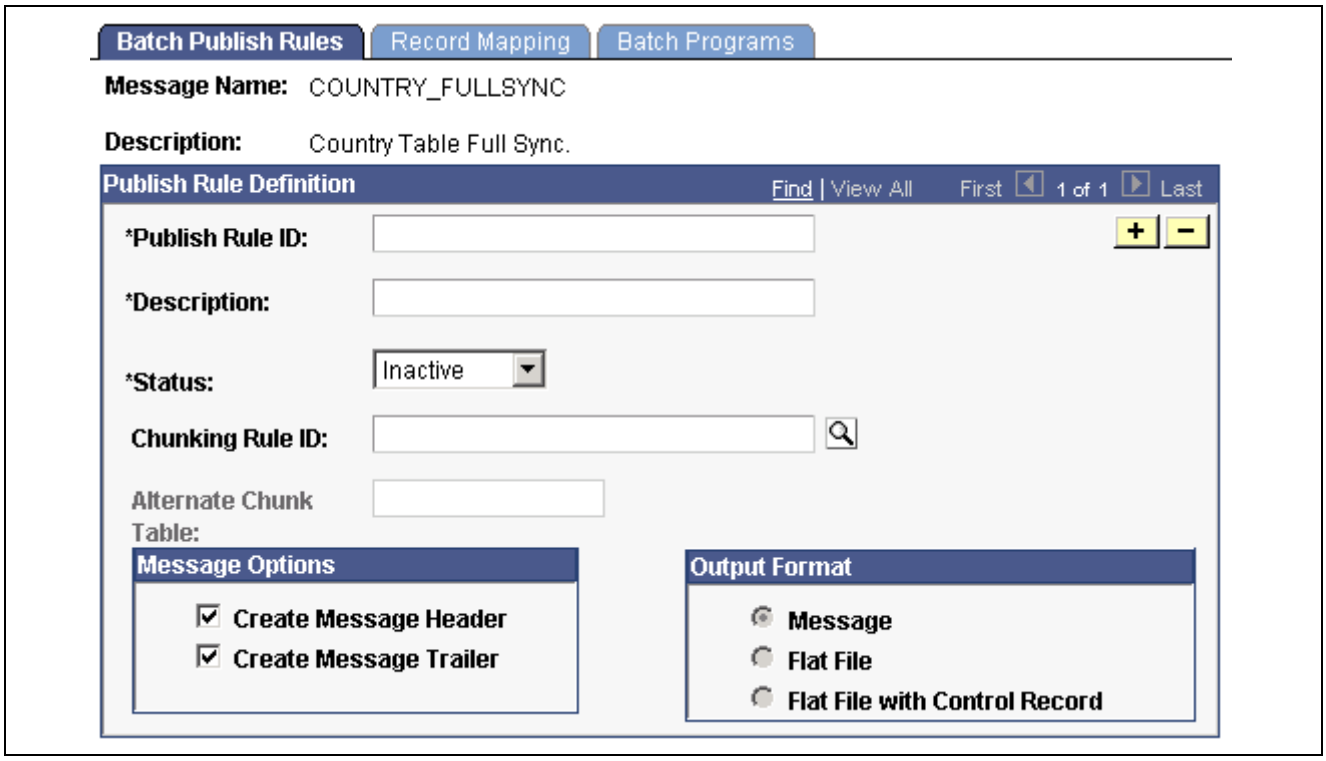
Note. You can link application programs to multiple publishing rules for the same message or different messages. The Publish utility treats each publishing rule as a separate publishing cycle.

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, SetIds	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_ADDSIDNODE_PNL	Click the Add button on the Add Notes to Chunk Rule page. This button is available after you have added nodes to the chunking rule.	Assign business units to a chunking rule.
Map Business Unit	EO_ADDNODEBU_PNL	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_CHUNKRULE	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.

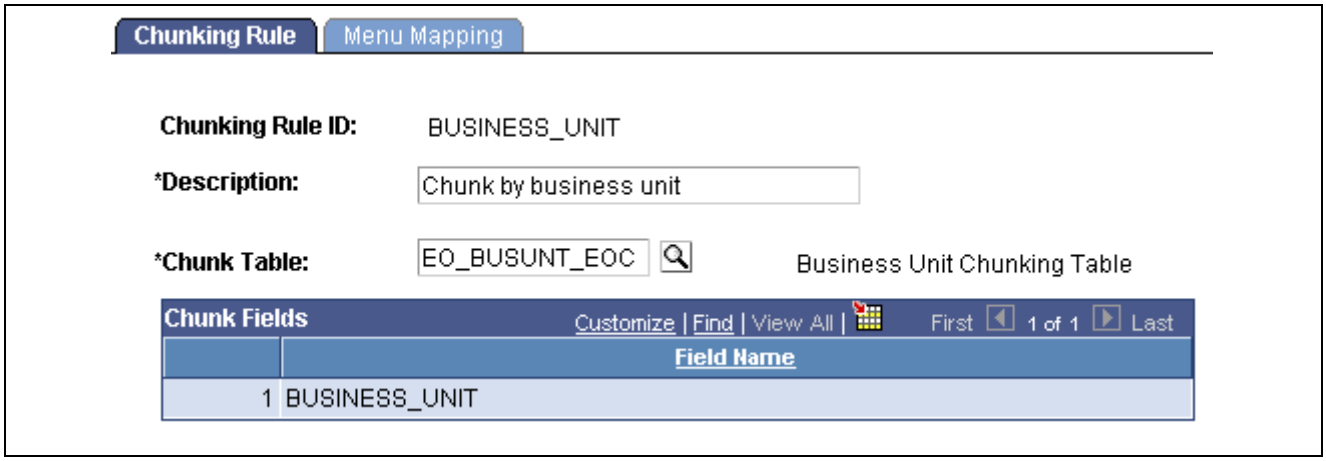
All user-defined chunking table names must end in *_EOC*. PeopleSoft provides two standard tables: EO_BUSUNIT_EOC for business unit values and EO_SETID_EOC for setID values. The different types of user-defined chunking tables are:

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: • CHUNK_RULE_ID • EFFDT • MSGNODENAME • Chunking fields 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.



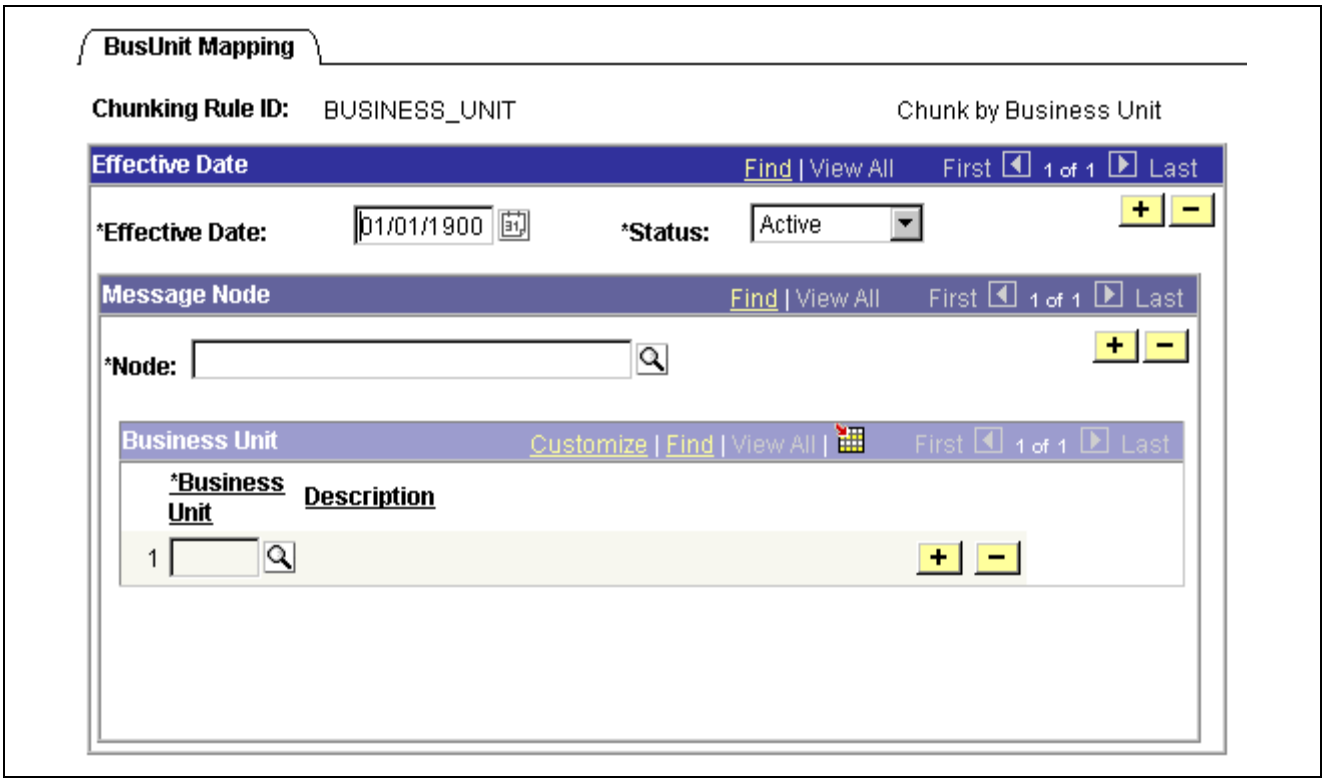
Chunking Rule page

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

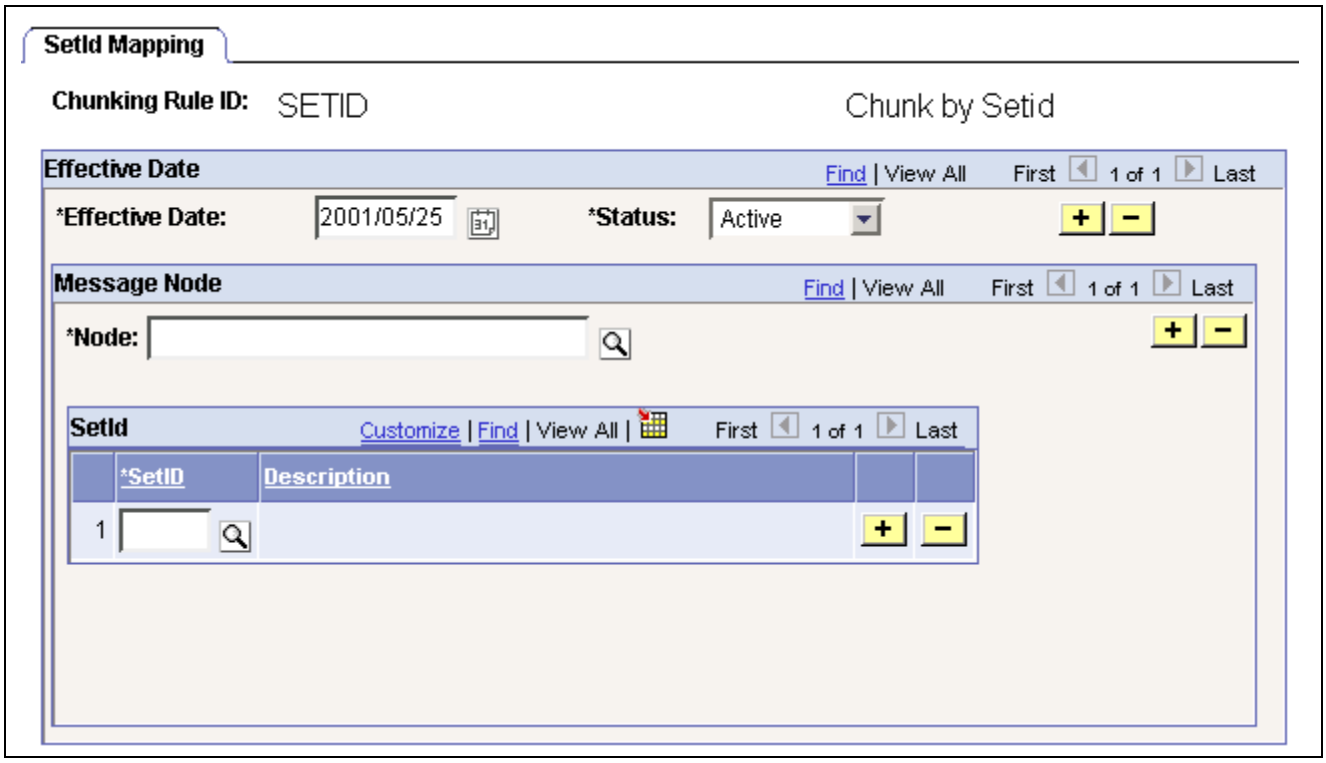
All four levels of the chunking rule tables appear.

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.

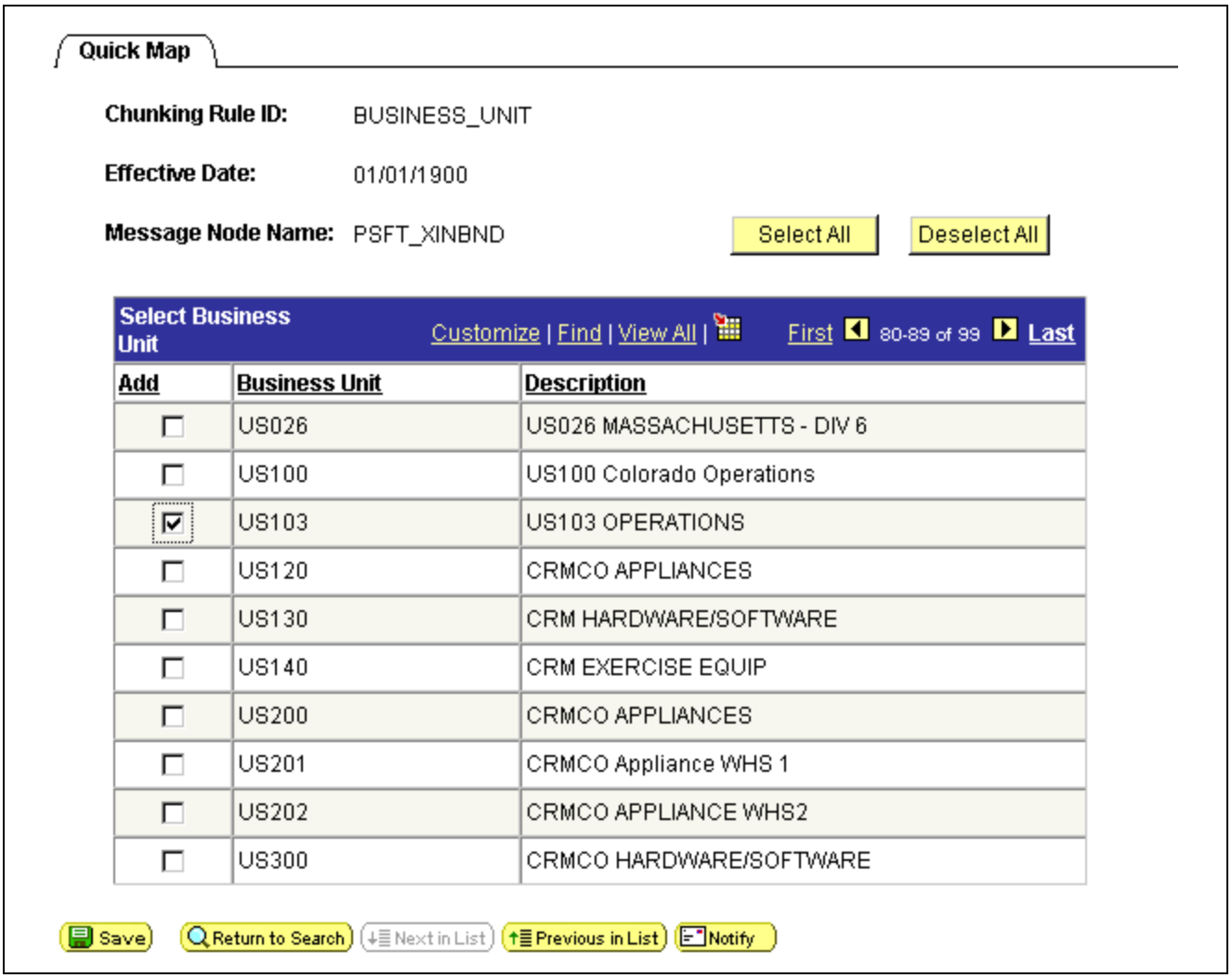
To add nodes to an existing chunking rule:

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

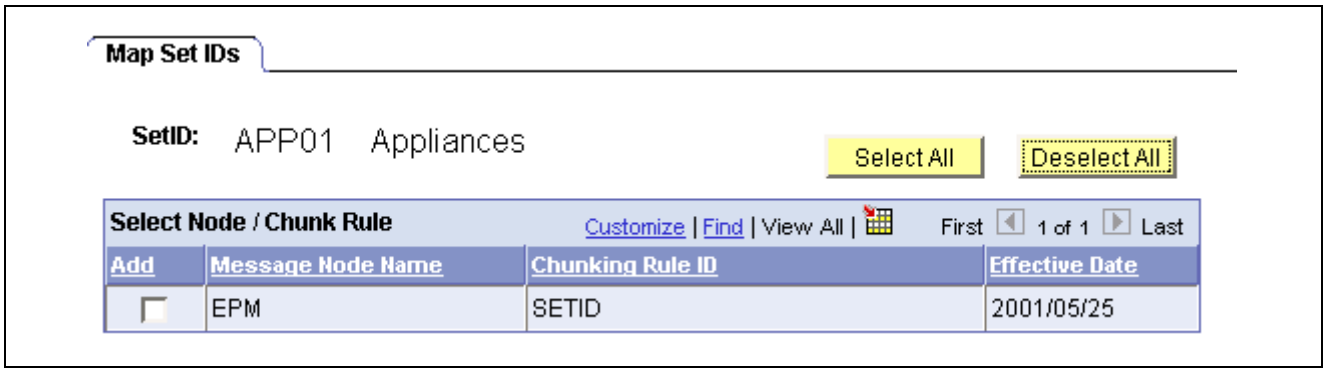
Access the Map Business Unit page.

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

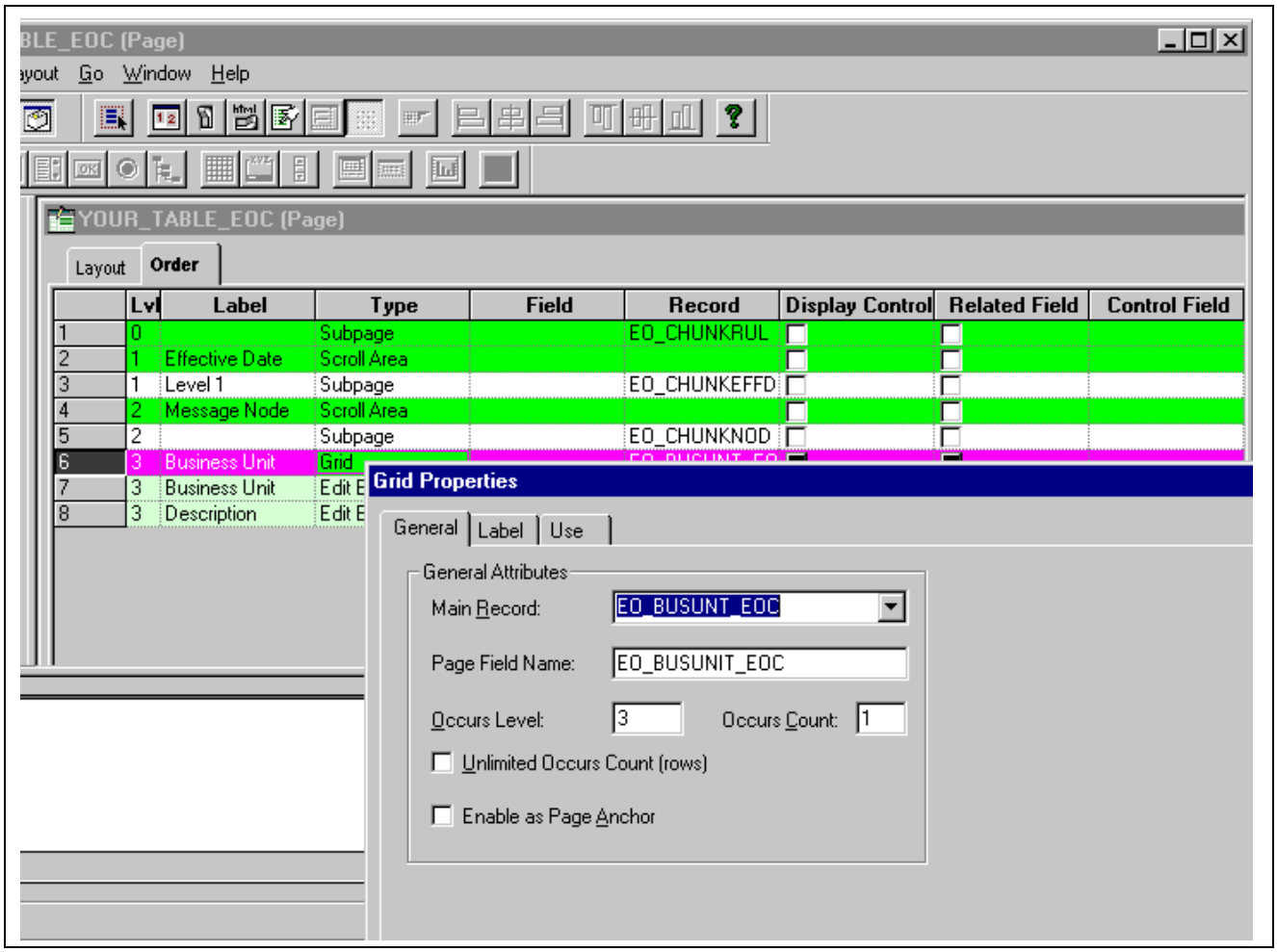
To create a view for the component search record:

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 29

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 integration point 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 Maintenance Utility, Data Maint Utility Setup2	Set up row security.
EO EIP Data Maint (EO EIP data maintenance)	EO_EIP_CTL_MAINT	Enterprise Utilities, Integration Definitions, Review Centralized Error	Test the error handling utility. 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)*. **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 29, "Using the Error Handling Utility," Setting Up the Error Handling Utility, page 372.

- 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

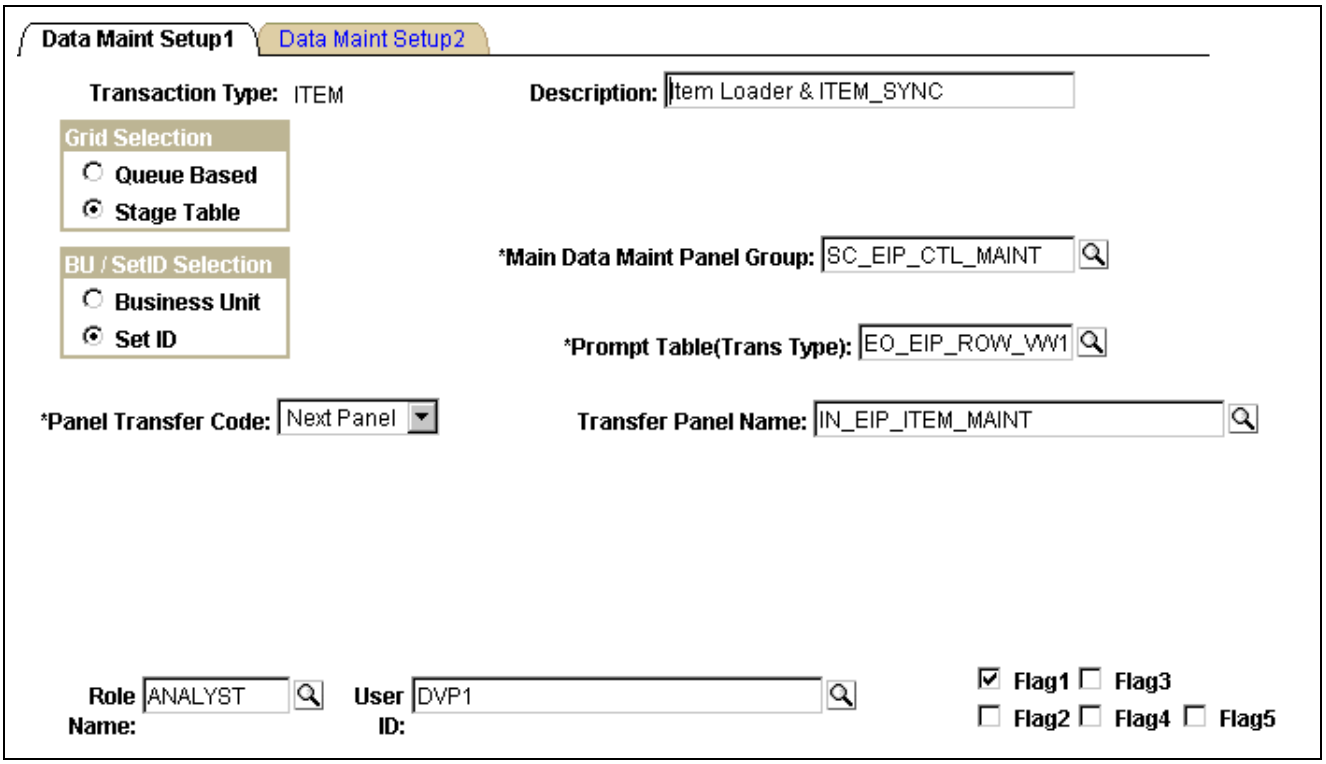
You need an EIP_CTL_ID every time that you process a message to maintain error handling. Use Function Generate_EIP_CTL_ID and Increment_EIP_CTL_ID to create EIP_CTL_IDs based on a new subscription process instance. For the Generate_EIP_CTL_ID function, invoke method 4 from the list of process instances.

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.

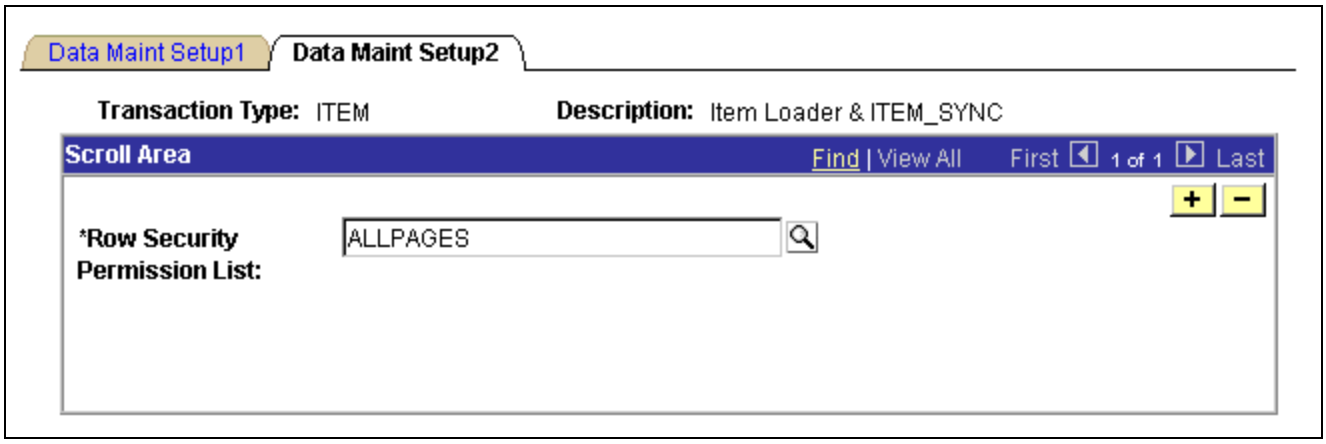
Transfer Panel Name — Select the default value *EO_EIP_ROW_VW1*. This view enforces row-level security. During the correction phase, when you try to select an EIP_PROGRAM (transaction type), the utility prompts from a list of values for which you have the correct access privilege.

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

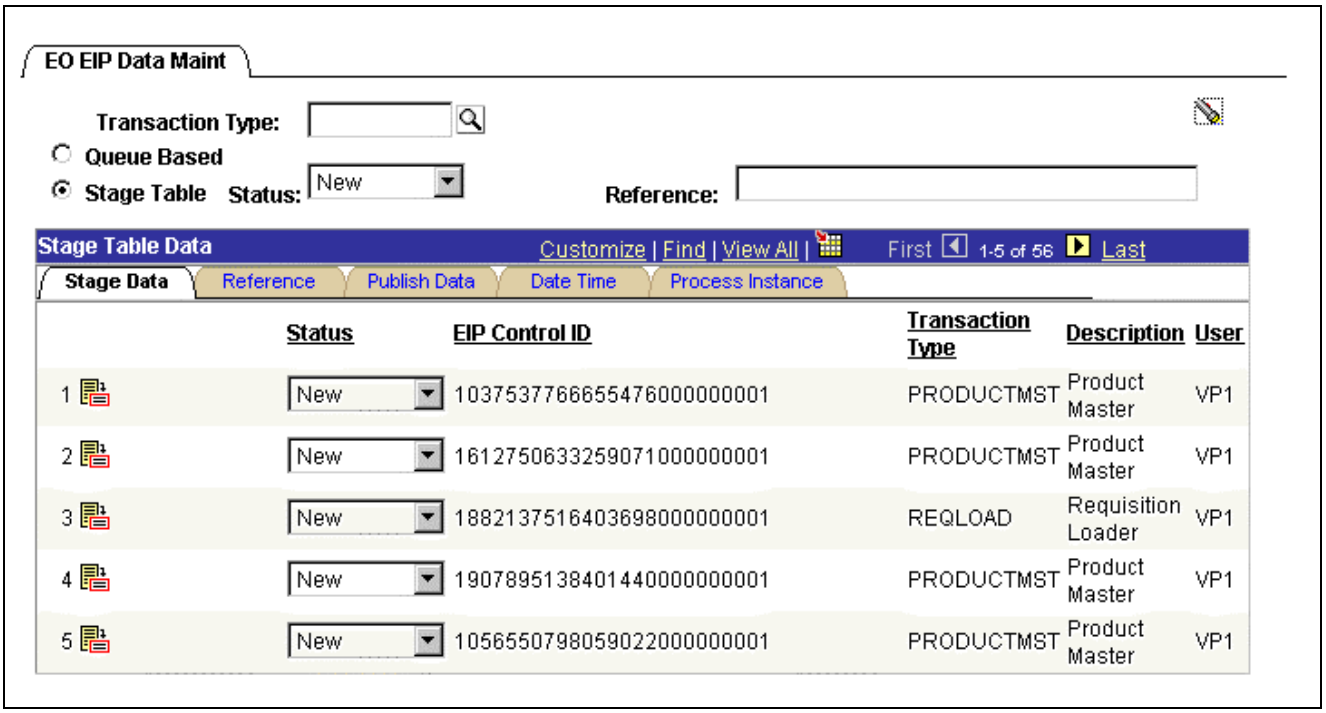
You can set up an EO_WF_ERR Application Engine process to scan the EO_EIP_CTL table periodically to look for rows with *ERROR* status. When the system finds errors, it generates workflow notifications and routes them to the role users that you designated in the User ID field on the Data Maint Setup1 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 30

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	EO_CHUNKRULE	Enterprise Components, Integration Definitions, Map Chunking Rules, Define Chunking Rules	Define the chunking rule description.
Full Data Publish	EO_FULLDATAPUB	Enterprise Components, Integration Definitions, Initiate Processes, Full Data Publish	Create the run control for the Full Data Publish utility. 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	TARGET RECORDS	ORDERING VIEW RECORDS
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.

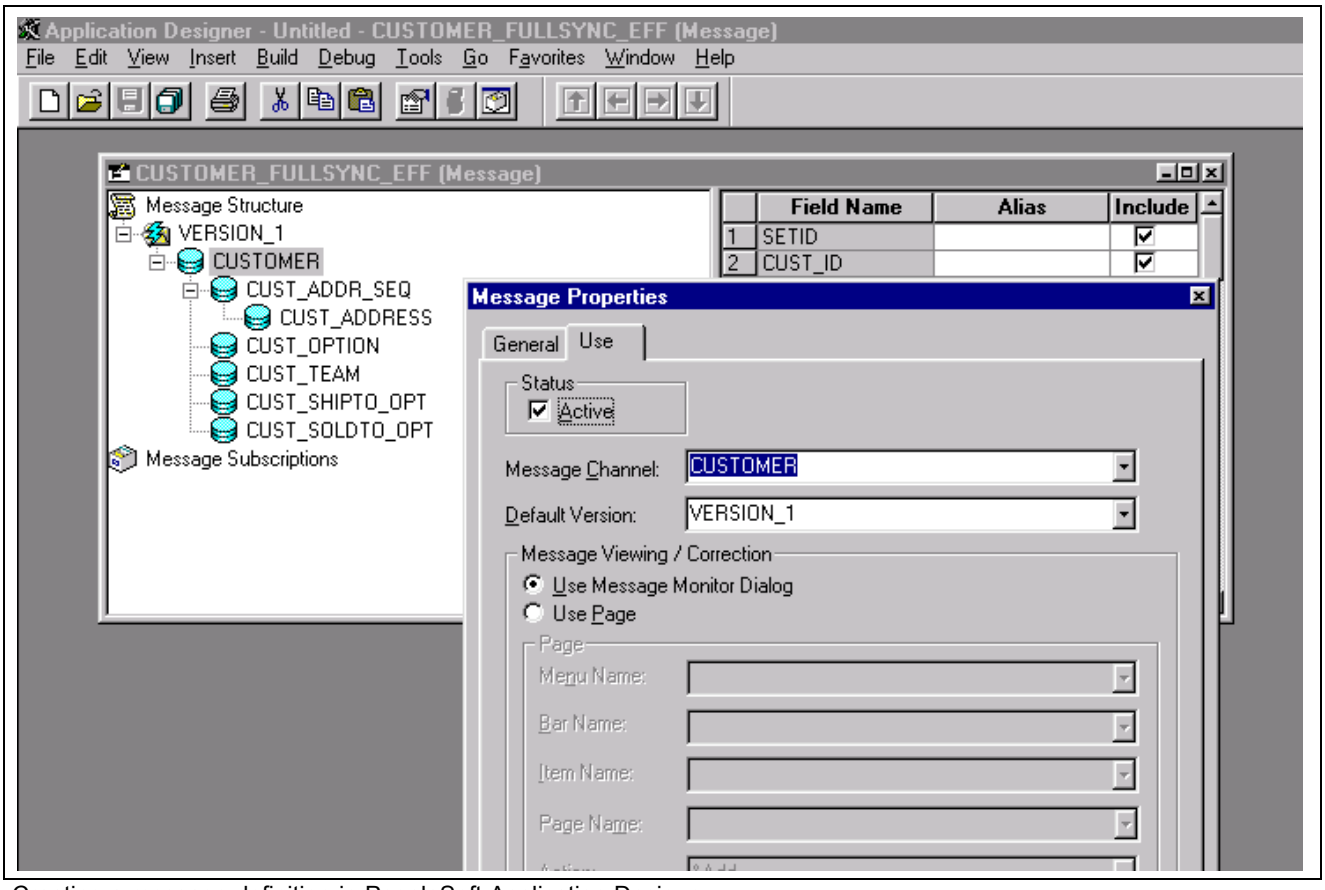
Begin by setting up the node and the transaction and connector details by using the Integration Profile setup function of PeopleSoft Integration Broker. To set up the node:

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*.

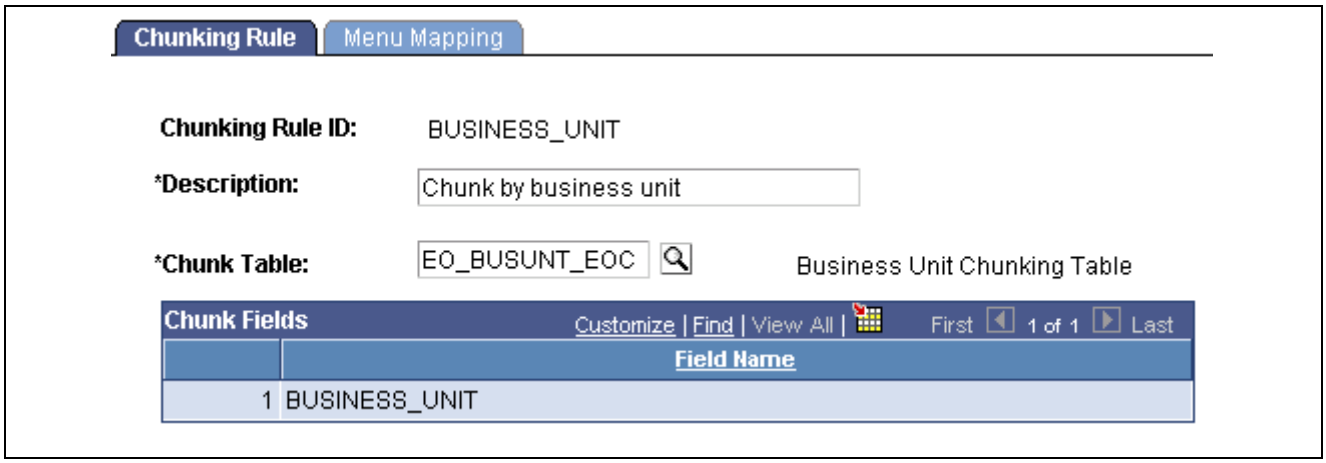
   Note. Messages must have the *_EFF* suffix to be effective-dated.

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:

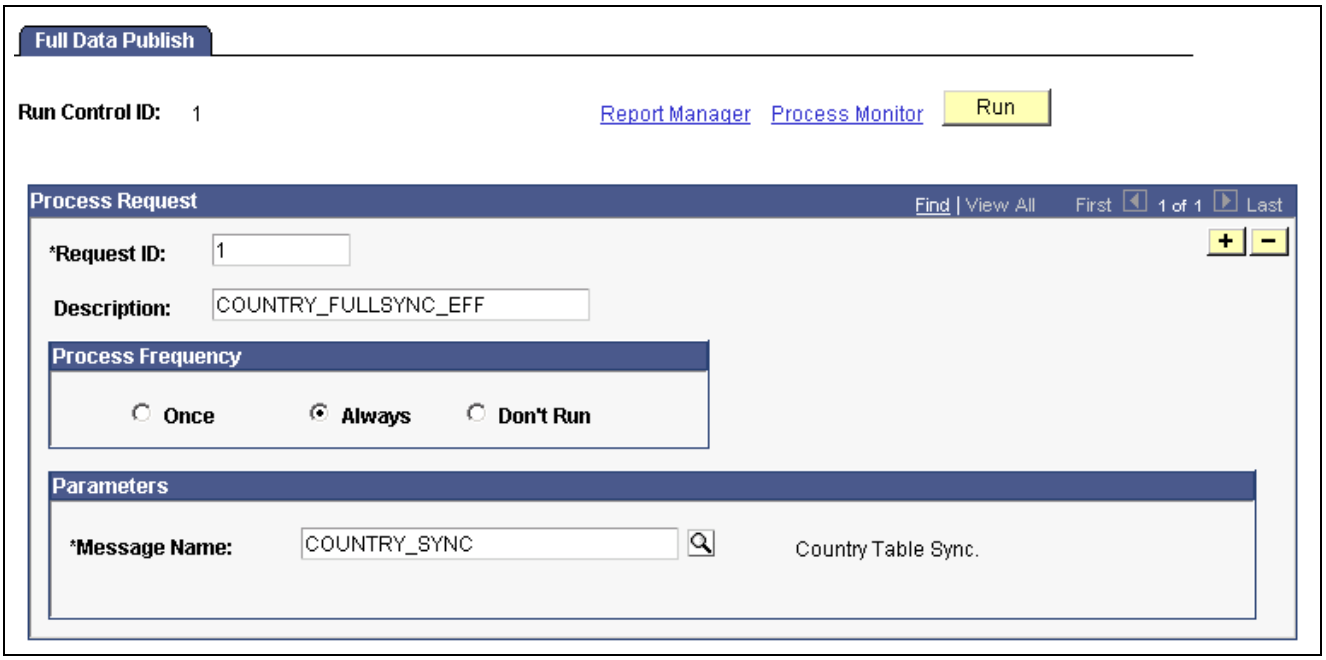
- Specify only target records that are effective-dated.
- Select only the current effective row to list the ordering view record that should be used as an override when publishing the message.
- The Effective Date Publish utility makes a logic pass through the data for each publish rule definition.

  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:

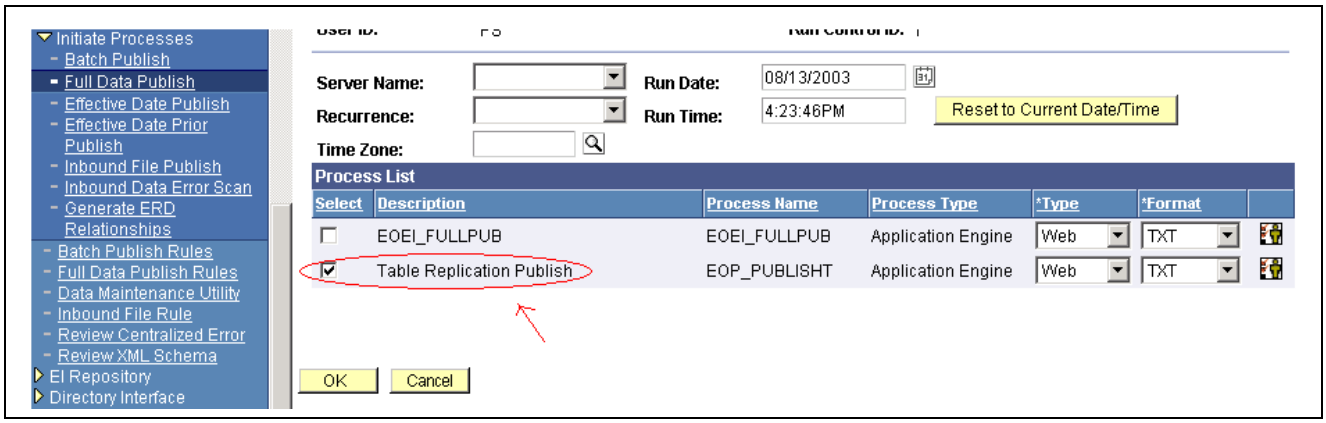


Table Replication Publish checkbox

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.

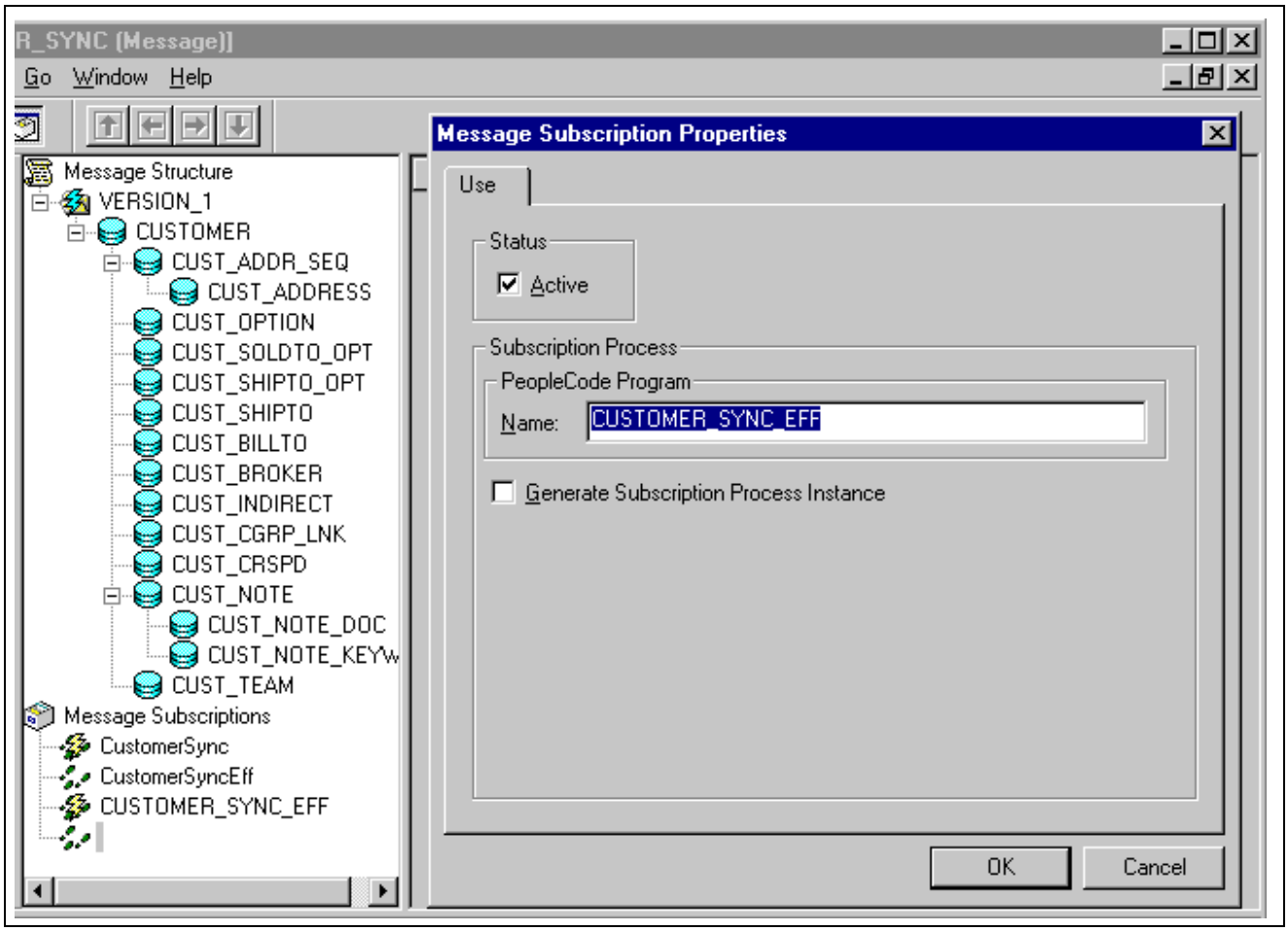
To create the message definition:

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.

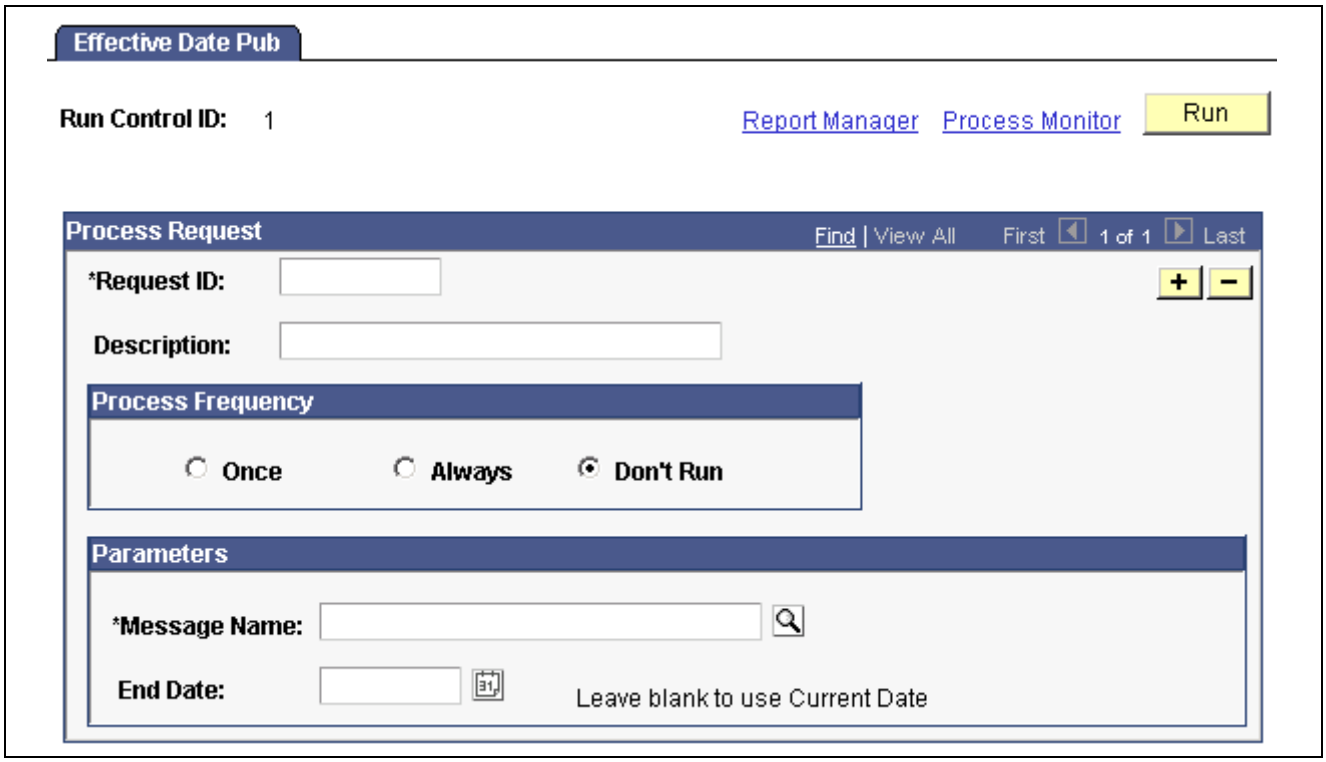
If the Effective Date Publish utility is invoked after not running for a period of time, it retrieves only the current row and publishes that as the active record. The presumption is that the subscribers want only the most current database information that is published.

Page Used to Run the Effective Date Publish Utility

Page Name	Object Name	Navigation	Usage
Effective Date Pub (effective date publish)	EO_EFFDATAPUB	Enterprise Components, Integration Definitions, 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 31

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 integration point, 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:

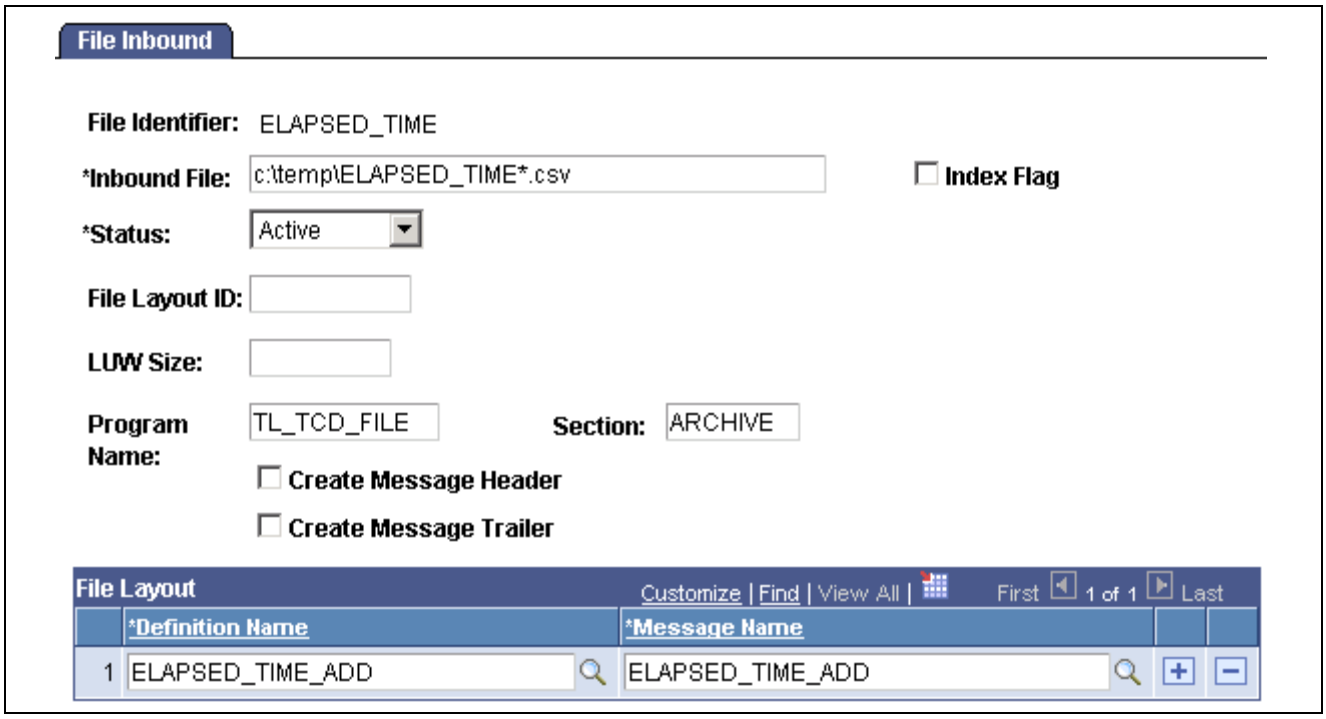
- Set up inbound flat file processing.
- Initiate inbound flat file processing.

Pages Used to Initiate File Processing

Page Name	Object Name	Navigation	Usage
File Inbound	EO_FILE_INBOUND	Enterprise Components, Integration Definitions, 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**

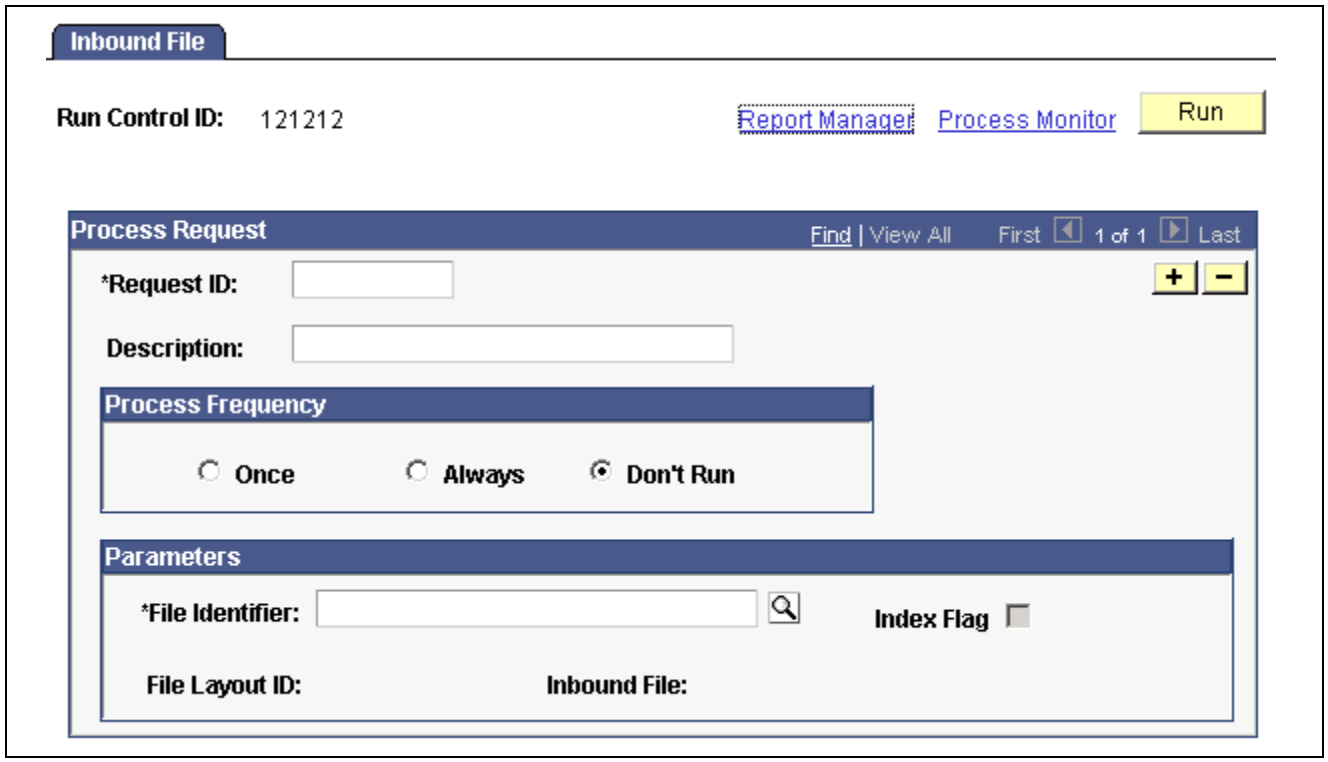
If the File Layout ID field is blank, this field should contain only one entry.

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 32

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

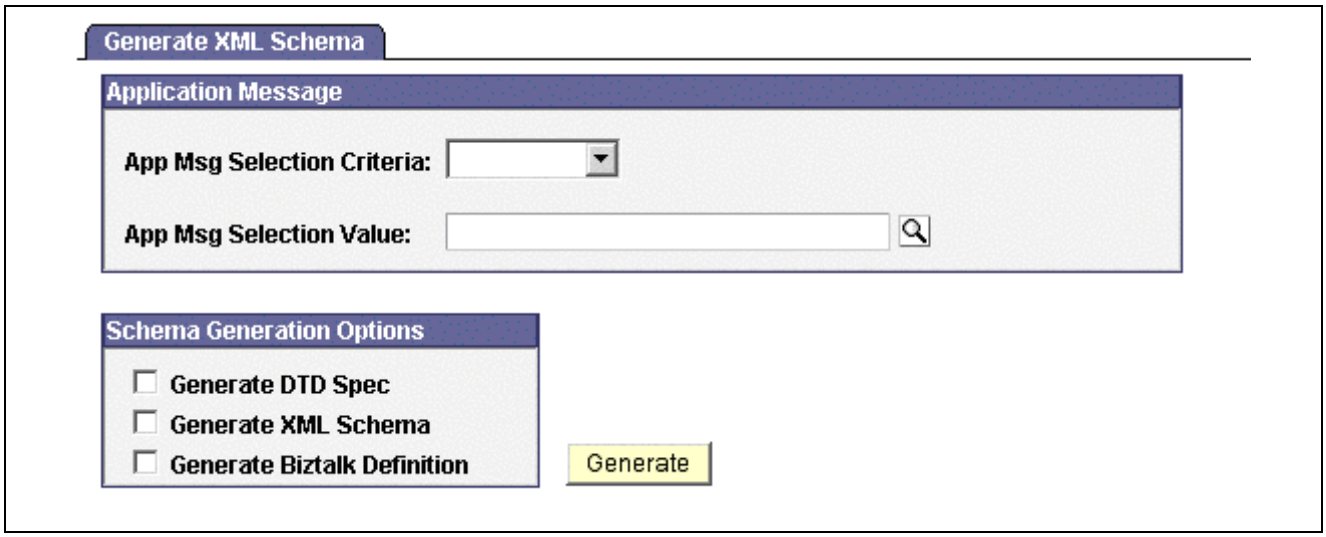
This section discusses how to generate the XML Schema.

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

Integration Point Naming Standards

Integration Point names describe the business purposes of integration points and provide a virtual wrapper for packaging one or more integration technology objects.

Integration point 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 integration points:

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. 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. 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.

academic career In PeopleSoft Enterprise Campus Solutions, all course work that a student undertakes at an academic institution and that is grouped in a single student record. For example, a university that has an undergraduate school, a graduate school, and various professional schools might define several academic careers—an undergraduate career, a graduate career, and separate careers for each professional school (law school, medical school, dental school, and so on).

academic institution In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.

academic organization In PeopleSoft Enterprise Campus Solutions, an entity that is part of the administrative structure within an academic institution. At the lowest level, an academic organization might be an academic department. At the highest level, an academic organization can represent a division.

academic plan In PeopleSoft Enterprise Campus Solutions, an area of study—such as a major, minor, or specialization—that exists within an academic program or academic career.

academic program In PeopleSoft Enterprise Campus Solutions, the entity to which a student applies and is admitted and from which the student graduates.

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.

address usage In PeopleSoft Enterprise Campus Solutions, a grouping of address types defining the order in which the address types are used. For example, you might define an address usage code to process addresses in the following order: billing address, dormitory address, home address, and then work address.

adjustment calendar In PeopleSoft Enterprise Campus Solutions, the adjustment calendar controls how a particular charge is adjusted on a student's account when the student drops classes or withdraws from a term. The charge adjustment is based on how much time has elapsed from a predetermined date, and it is determined as a percentage of the original charge amount.

administrative function In PeopleSoft Enterprise Campus Solutions, a particular functional area that processes checklists, communication, and comments. The administrative function identifies which variable data is added to a person's checklist or communication record when a specific checklist code, communication category, or comment is assigned to the student. This key data enables you to trace that checklist, communication, or comment back to a specific processing event in a functional area.

admit type In PeopleSoft Enterprise Campus Solutions, a designation used to distinguish first-year applications from transfer applications.

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.

analysis database In PeopleSoft Enterprise Campus Solutions, database tables that store large amounts of student information that may not appear in standard report formats. The analysis database tables contain keys for all objects in a report that an application program can use to reference other student-record objects that are not contained in the printed report. For instance, the analysis database contains data on courses that are considered for satisfying a requirement but that are rejected. It also contains information on

courses captured by global limits. An analysis database is used in PeopleSoft Enterprise Academic Advisement.

Application Messaging PeopleSoft Application Messaging enables applications within the PeopleSoft Enterprise product family to communicate synchronously or asynchronously with other PeopleSoft and third-party applications. An application message defines the records and fields to be published or subscribed to.

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.

audience In PeopleSoft Enterprise Campus Solutions, a segment of the database that relates to an initiative, or a membership organization that is based on constituent attributes rather than a dues-paying structure. Examples of audiences include the Class of '65 and Undergraduate Arts & Sciences.

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.

billing career In PeopleSoft Enterprise Campus Solutions, the one career under which other careers are grouped for billing purposes if a student is active simultaneously in multiple careers.

bio bit or bio brief In PeopleSoft Enterprise Campus Solutions, a report that summarizes information stored in the system about a particular constituent. You can generate standard or specialized reports.

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 activity The name of a subset of a detailed business process. This might be a specific transaction, task, or action that you perform in a business process.

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 process A standard set of 17 business processes are defined and maintained by the PeopleSoft product families and are supported by Business Process Engineering group at PeopleSoft. An example of a business process is Order Fulfillment, which is a business process that manages sales orders and contracts, inventory, billing, and so forth.

See also *detailed business process.*

business task The name of the specific function depicted in one of the business processes.

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.

campus In PeopleSoft Enterprise Campus Solutions, an entity that is usually associated with a distinct physical administrative unit, that belongs to a single academic institution, that uses a unique course catalog, and that produces a common transcript for students within the same academic career.

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.

category In PeopleSoft Enterprise Campus Solutions, a broad grouping to which specific comments or communications (contexts) are assigned. Category codes are also linked to 3C access groups so that you can assign data-entry or view-only privileges across functions.

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.

checklist code In PeopleSoft Enterprise Campus Solutions, a code that represents a list of planned or completed action items that can be assigned to a staff member, volunteer, or unit. Checklists enable you to view all action assignments on one page.

class In PeopleSoft Enterprise Campus Solutions, a specific offering of a course component within an academic term.

See also *course.*

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.*

clearance In PeopleSoft Enterprise Campus Solutions, the period of time during which a constituent in PeopleSoft Contributor Relations is approved for involvement in an initiative or an action. Clearances are used to prevent development officers from making multiple requests to a constituent during the same time period.

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.

cohort In PeopleSoft Enterprise Campus Solutions, the highest level of the three-level classification structure that you define for enrollment management. You can define a cohort level, link it to other levels, and set enrollment target numbers for it.

See also *population* and *division*.

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.

comm key See *communication key.*

communication key In PeopleSoft Enterprise Campus Solutions, a single code for entering a combination of communication category, communication context, communication method, communication direction, and standard letter code. Communication keys (also called *comm keys* or *speed keys*) can be created for background processes as well as for specific users.

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.

component interface 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.

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.

constituents In PeopleSoft Enterprise Campus Solutions, friends, alumni, organizations, foundations, or other entities affiliated with the institution, and about which the institution maintains information. The constituent types delivered with PeopleSoft Enterprise Contributor Relations Solutions are based on those defined by the Council for the Advancement and Support of Education (CASE).

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 Campus Solutions, a specific instance of a comment or communication. One or more contexts are assigned to a category, which you link to 3C access groups so that you can assign data-entry or view-only privileges across functions.

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.

course In PeopleSoft Enterprise Campus Solutions, a course that is offered by a school and that is typically described in a course catalog. A course has a standard syllabus and credit level; however, these may be modified at the class level. Courses can contain multiple components such as lecture, discussion, and lab.

See also *class*.

course share set In PeopleSoft Enterprise Campus Solutions, a tag that defines a set of requirement groups that can share courses. Course share sets are used in PeopleSoft Enterprise Academic Advisement.

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 cube In PeopleSoft Analytic Calculation Engine, a data cube is a container for one kind of data (such as Sales data) and works with in tandem with one or more dimensions. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and online analytical processing (OLAP) cubes in PeopleSoft Cube Manager.

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.

detailed business process A subset of the business process. For example, the detailed business process named Determine Cash Position is a subset of the business process called Cash Management.

dimension In PeopleSoft Analytic Calculation Engine, a dimension contains a list of one kind of data that can span various contexts, and it is a basic component of an analytic model. Within the analytic model, a dimension is attached to one or more data cubes. In PeopleSoft Cube Manager, a dimension is the most basic component of an OLAP cube and specifies the PeopleSoft metadata to be used to create the dimension's rollup structure. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and OLAP cubes in PeopleSoft Cube Manager.

directory information tree In PeopleSoft Directory Interface, the representation of a directory's hierarchical structure.

division In PeopleSoft Enterprise Campus Solutions, the lowest level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a division level, link it to other levels, and set enrollment target numbers for it.

See also *population* and *cohort*.

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.

equity item limit In PeopleSoft Enterprise Campus Solutions, the amounts of funds set by the institution to be awarded with discretionary or gift funds. The limit could be reduced by amounts equal to such things as expected family contribution (EFC) or parent contribution. Students are packaged by Equity Item Type Groups and Related Equity Item Types. This limit can be used to assure that similar student populations are packaged equally.

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.

financial aid term In PeopleSoft Enterprise Campus Solutions, a combination of a period of time that the school determines as an instructional accounting period and an academic career. It is created and defined during the setup process. Only terms eligible for financial aid are set up for each financial aid career.

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.

gap In PeopleSoft Enterprise Campus Solutions, an artificial figure that sets aside an amount of unmet financial aid need that is not funded with Title IV funds. A gap can be used to prevent fully funding any student to conserve funds, or it can be used to preserve unmet financial aid need so that institutional funds can be awarded.

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.

gift table In PeopleSoft Enterprise Campus Solutions, a table or so-called *donor pyramid* describing the number and size of gifts that you expect will be needed to successfully complete the campaign in PeopleSoft Contributor Relations. The gift table enables you to estimate the number of donors and prospects that you need at each gift level to reach the campaign goal.

GL business unit Abbreviation for *general ledger business unit.* A unit in an organization that is an independent entity for accounting purposes. It maintains its own set of accounting books.

See also *business unit*.

GL entry template Abbreviation for *general ledger entry template.* In PeopleSoft Enterprise Campus Solutions, a template that defines how a particular item is sent to the general ledger. An item-type maps to the general ledger, and the GL entry template can involve multiple general ledger accounts. The entry to the general ledger is further controlled by high-level flags that control the summarization and the type of accounting—that is, accrual or cash.

GL Interface process Abbreviation for *General Ledger Interface process.* In PeopleSoft Enterprise Campus Solutions, a process that is used to send transactions from PeopleSoft Enterprise Student Financials to the general ledger. Item types are mapped to specific general ledger accounts, enabling transactions to move to the general ledger when the GL Interface process is run.

group In PeopleSoft Billing and Receivables, a posting entity that comprises one or more transactions (items, deposits, payments, transfers, matches, or write-offs).

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.

initiative In PeopleSoft Enterprise Campus Solutions, the basis from which all advancement plans are executed. It is an organized effort targeting a specific constituency, and it can occur over a specified period of time with specific purposes and goals. An initiative can be a campaign, an event, an organized volunteer effort, a membership drive, or any other type of effort defined by the institution. Initiatives can be multipart, and they can be related to other initiatives. This enables you to track individual parts of an initiative, as well as entire initiatives.

inquiry access In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user only to view data.

See also *update access.*

institution In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.

integration A relationship between two compatible integration points that enables communication to take place between systems. Integrations enable PeopleSoft applications to work seamlessly with other PeopleSoft applications or with third-party systems or software.

integration point An interface that a system uses to communicate with another PeopleSoft application or an external application.

integration set A logical grouping of integrations that applications use for the same business purpose. For example, the integration set ADVANCED_SHIPPING_ORDER contains all of the integrations that notify a customer that an order has shipped.

item In PeopleSoft Inventory, a tangible commodity that is stored in a business unit (shipped from a warehouse).

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.

In PeopleSoft Receivables, an individual receivable. An item can be an invoice, a credit memo, a debit memo, a write-off, or an adjustment.

item shuffle In PeopleSoft Enterprise Campus Solutions, a process that enables you to change a payment allocation without having to reverse the payment.

joint communication In PeopleSoft Enterprise Campus Solutions, one letter that is addressed jointly to two people. For example, a letter might be addressed to both Mr. Sudhir Awat and Ms. Samantha Mortelli. A relationship must be established between the two individuals in the database, and at least one of the individuals must have an ID in the database.

keyword In PeopleSoft Enterprise Campus Solutions, a term that you link to particular elements within PeopleSoft Student Financials, Financial Aid, and Contributor Relations. You can use keywords as search criteria that enable you to locate specific records in a search dialog box.

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.

LMS Abbreviation for *learning management system.* In PeopleSoft Enterprise Campus Solutions, LMS is a PeopleSoft Student Records feature that provides a common set of interoperability standards that enable the sharing of instructional content and data between learning and administrative environments.

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.

mass change In PeopleSoft Enterprise Campus Solutions, mass change is a SQL generator that can be used to create specialized functionality. Using mass change, you can set up a series of Insert, Update, or Delete SQL statements to perform business functions that are specific to the institution.

See also *3C engine*.

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.

need In PeopleSoft Enterprise Campus Solutions, the difference between the cost of attendance (COA) and the expected family contribution (EFC). It is the gap between the cost of attending the school and the student's resources. The financial aid package

	is based on the amount of financial need. The process of determining a student's need is called *need analysis.*
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. 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.
payment shuffle	In PeopleSoft Enterprise Campus Solutions, a process allowing payments that have been previously posted to a student's account to be automatically reapplied when a higher priority payment is posted or the payment allocation definition is changed.
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 component processor. PeopleCode generates results based on existing data or user actions. By using various tools provided with PeopleTools, external services are available to all PeopleSoft applications wherever PeopleCode can be executed.
PeopleCode event	See *event.*
PeopleSoft Pure 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.
person of interest	A person about whom the organization maintains information but who is not part of the workforce.
personal portfolio	In PeopleSoft Enterprise Campus Solutions, the user-accessible menu item that contains an individual's name, address, telephone number, and other personal information.
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.

population In PeopleSoft Enterprise Campus Solutions, the middle level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a population level, link it to other levels, and set enrollment target numbers for it.

See also *division* and *cohort*.

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.

primacy number In PeopleSoft Enterprise Campus Solutions, a number that the system uses to prioritize financial aid applications when students are enrolled in multiple academic careers and academic programs at the same time. The Consolidate Academic Statistics process uses the primacy number indicated for both the career and program at the institutional level to determine a student's primary career and program. The system also uses the number to determine the primary student attribute value that is used when you extract data to report on cohorts. The lowest number takes precedence.

primary name type In PeopleSoft Enterprise Campus Solutions, the name type that is used to link the name stored at the highest level within the system to the lower-level set of names that an individual provides.

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 A PeopleSoft or third-party product. PeopleSoft organizes its software products into product families and product lines. Interactive Services Repository contains information about every release of every product that PeopleSoft sells, as well as products from certified third-party companies. These products are displayed with the product name and release number.

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.

product family A group of products that are related by common functionality. The family names that can be searched using Interactive Service Repository are PeopleSoft Enterprise, PeopleSoft EnterpriseOne, PeopleSoft World, and third-party, certified PeopleSoft partners.

product line The name of a PeopleSoft product line or the company name of a third-party certified partner. Integration Services Repository enables you to search for integration points by product line.

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.

prospects

In PeopleSoft Enterprise Campus Solutions, students who are interested in applying to the institution.

In PeopleSoft Enterprise Contributor Relations, individuals and organizations that are most likely to make substantial financial commitments or other types of commitments to the institution.

publishing

In PeopleSoft Enterprise Incentive Management, a stage in processing that makes incentive-related results available to participants.

rating components

In PeopleSoft Enterprise Campus Solutions, variables used with the Equation Editor to retrieve specified populations.

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.*

See *record input VAT flag.*

recname

The name of a record that is used to determine the associated field to match a value or set of values.

recognition

In PeopleSoft Enterprise Campus Solutions, the recognition type indicates whether the PeopleSoft Enterprise Contributor Relations donor is the primary donor of a commitment or shares the credit for a donation. Primary donors receive hard credit that must total 100 percent. Donors that share the credit are given soft credit. Institutions can also define other share recognition-type values such as memo credit or vehicle credit.

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.

reversal indicator In PeopleSoft Enterprise Campus Solutions, an indicator that denotes when a particular payment has been reversed, usually because of insufficient funds.

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.

SCP SCBM XML message Abbreviation for *Supply Chain Planning Supply Chain Business Modeler Extensible Markup Language message.* PeopleSoft EnterpriseOne Supply Chain Business Modeler uses XML as the format for all data that it imports and exports.

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.

search/match In PeopleSoft Enterprise Campus Solutions and PeopleSoft Enterprise Human Resources Management Solutions, a feature that enables you to search for and identify duplicate records in the database.

seasonal address In PeopleSoft Enterprise Campus Solutions, an address that recurs for the same length of time at the same time of year each year until adjusted or deleted.

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.

service impact In PeopleSoft Enterprise Campus Solutions, the resulting action triggered by a service indicator. For example, a service indicator that reflects nonpayment of account balances by a student might result in a service impact that prohibits registration for classes.

service indicator In PeopleSoft Enterprise Campus Solutions, indicates services that may be either withheld or provided to an individual. Negative service indicators indicate holds that prevent the individual from receiving specified services, such as check-cashing privileges or registration for classes. Positive service indicators designate special services that are provided to the individual, such as front-of-line service or special services for disabled students.

session In PeopleSoft Enterprise Campus Solutions, time elements that subdivide a term into multiple time periods during which classes are offered. In PeopleSoft Contributor Relations, a session is the means of validating gift, pledge, membership, or adjustment data entry . It controls access to the data entered by a specific user ID. Sessions are balanced, queued, and then posted to the institution's financial system. Sessions must be posted to enter a matching gift or pledge payment, to make an adjustment, or to process giving clubs or acknowledgements.

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 key process In PeopleSoft Enterprise Campus Solutions, a process that relates a particular transaction to the source of the charge or financial aid. On selected pages, you can drill down into particular charges.

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.

speed key See *communication key.*

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.

standard letter code In PeopleSoft Enterprise Campus Solutions, a standard letter code used to identify each letter template available for use in mail merge functions. Every letter generated in the system must have a standard letter code identification.

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.

tax authority In PeopleSoft Enterprise Campus Solutions, a user-defined element that combines a description and percentage of a tax with an account type, an item type, and a service impact.

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.

third party A company or vendor that has extensive PeopleSoft product knowledge and whose products and integrations have been certified and are compatible with PeopleSoft applications.

3C engine Abbreviation for *Communications, Checklists, and Comments engine.* In PeopleSoft Enterprise Campus Solutions, the 3C engine enables you to automate business processes that involve additions, deletions, and updates to communications, checklists, and comments. You define events and triggers to engage the engine, which runs the mass change and processes the 3C records (for individuals or organizations) immediately and automatically from within business processes.

3C group Abbreviation for *Communications, Checklists, and Comments group.* In PeopleSoft Enterprise Campus Solutions, a method of assigning or restricting access privileges. A 3C group enables you to group specific communication categories, checklist codes, and comment categories. You can then assign the group inquiry-only access or update access, as appropriate.

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.

tuition lock In PeopleSoft Enterprise Campus Solutions, a feature in the Tuition Calculation process that enables you to specify a point in a term after which students are charged a minimum (or *locked*) fee amount. Students are charged the locked fee amount even if they later drop classes and take less than the normal load level for that tuition charge.

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.

update access In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user to edit and update data.

See also *inquiry access.*

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.

worker A person who is part of the workforce; an employee or a contingent worker.

workset A group of people and organizations that are linked together as a set. You can use worksets to simultaneously retrieve the data for a group of people and organizations and work with the information on a single page.

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 link The XML Linking language enables you to insert elements into XML documents to create a links between resources.

XML schema An XML definition that standardizes the representation of application messages, component interfaces, or business interlinks.

XPI Abbreviation for *eXtended Process Integrator.* PeopleSoft XPI is the integration infrastructure that enables both real-time and batch communication with EnterpriseOne applications.

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

A

action bundles 115
action framework
  architecture 149
  definition of 115
  overview 149
action objective
  registering 159
action type bundles
  registering 151
action types 115
  configuring 172
  creating 172
  overview 149
  registering 151
actions
  adding, editing, configuring 121
  display alert 123
  how they execute 150
Add Nodes to Chunk Rule page 340
additional documentation xxii
Address Format page 44
address verification service, using for credit card processing 100
addresses, customizing formats 43
aliases 143
application class
  example of accessing binds 178
  sample of resolving terms 178
application class code
  creating for design and runtime 176
application class implementation 134
  creating 177
  sample code 179
application fundamentals xxi
application server, configuring for credit card processing on Solaris 90
Attribute Details page 26
authorize-and-bill transactions, using 96
authorize-only transactions, using 97

B

batch publish rules
  associating to a message 353
  characterizing 353
  pages used to assign 353
  setting up 338
Batch Publish Rules page 338, 353
bill-only transactions, using 98
binds
  application class accessing 178
business domain
  registering 167
Business Interlink Status report 38
business interlinks
  setting up for credit card processing on Solaris 88
  setting up for credit card processing on Windows NT 87
business units 340
  See Also chunking rules
BusUnit Mapping page 360

C

caching, enabling for navigation pages 277
Card Type page 80
category
  registering 167
chunking data 360, 361
Chunking Rule page 360, 379
chunking rules
  adding nodes to 362
  assigning business units to 340, 362
  assigning to a business unit 363
  assigning to a setID 364
  associating to publication rules 338
  creating 359
  defining 360
chunking tables, routing PeopleCode 367
Clean Portal Project process
  running 327
  understanding 325
combinable action 150
comments, submitting xxvi
common elements xxvii
component interfaces 333
conditions
  adding 119
  selecting terms for 120

setting right-hand side values 121
contact information xxvi
context objects
managing 146
context variables 143
characteristics of 143
contexts 114
configuring 144
creating 169
managing 143
online and generic 143
overview 143
contextual implementations 138
country
customizing address formats 43
Country Description page 44
country, describing 44
Credit Card Authorize Bill and Credit IP 75
credit card interface
configuring the application server on Solaris 90
configuring the PeopleSoft Application Designer plug-in 90
defining connection parameters 77
defining credit card types 80
describing error codes 85
describing required parameters 96
describing unsupported input fields 102
setting transaction inputs 96
setting transaction outputs 96
setting up 75, 76
testing 81, 83
understanding 75
understanding a CyberSource integration 75
credit card interface business interlinks
setting up on Solaris 88
setting up on Windows NT 87
credit card interface CyberSource API
setting up 86
setting up on Solaris 88, 89
setting up on Windows NT 87
Credit Card Interface Installation page 77
credit card interface integrations
understanding a manual 76
understanding options 75
understanding XML 76
credit card interface Java interlink plug-in
setting up 86
setting up on Solaris 88, 90
setting up on Windows NT 87, 88
understanding 86
credit card interface services
using address verification 100
using the fraud screen 101
credit card interface transactions
using authorize-and-bill 96
using authorize-only 97
using bill-only 98
using credit 99
credit card options
manual processing 75
XML-compliant vendors 75
credit card types, defining accepted 80
credit transactions, using 99
cross rates, calculating 65
cross-references xxv
currencies
converting ad hoc amounts 67
defining 52
defining quotation methods 54
describing tables 49
understanding 49
understanding conversion factor fields 51
understanding requirements for conversion 52
understanding the visual rate 51
understanding triangulation 50
currency codes, defining 53
currency exchange calculator, using 67
currency quotation methods, defining 54
custom chunking tables 355
See Also chunking tables
custom navigation pages
describing 275
describing folder and link sequence on 298
custom operator expressions
creating 170
Customer Connection website xxii
customizing address formats 43
CyberSource
processing credit cards with 75
understanding credit card processing with 75
CyberSource API
describing support requirements 86
setting up 86

setting up on Solaris 88, 89
setting up on Windows NT 87

D

data library
components 133
definition of 114
how a value is passed 179
overview 133
setting log options 156
subject areas 114
Data Maintenance Setup1 page 372
Data Maintenance Setup2 page 374
data subscription
in multilingual environment 344
in non-multilingual environment 344
data synchronization
understanding for LDAP 10
using LDIFs 10
dataset roles
creating 71
understanding 69
dataset rules
creating 70
defining 69
understanding 69
datasets
defining mobile data distribution rules 72
understanding 69
understanding mobile data distribution 71
using mobile data distribution user rules 74
Delete Empty Folders process
running 325
understanding 325
Directory Audit report 38
directory information trees, understanding 9
Directory Interface
configuring the LDAP directory 12
delivered business interlinks 10
generating LDAP directory reports 35
product requirements, recommendations 3
reviewing LDAP directory data 35
setting up LDAP directory authentication for 16
setting up mappings for 19
steps to use 10
understanding 9
understanding data mapping 9
using the configuration component 11
Directory Interface setup
setting up in PeopleSoft Application Designer 11
setting up in PeopleSoft Integration Broker 11
Directory Load page 32
display action
enabling 172
display alert action
configuring 123
display alert action type 149
Distinguished Name Details page 23
distinguished names defaults 26
distinguished names, understanding 9
DN, *See* distinguished names
DN Attribute Function - Directory Interface page 24
DN Defaults page 26
documentation
printed xxii
related xxii
updates xxii

E

Effective Date Publish page 387
effective date publish utility
definition 377
full data publish 378
pages used to run 387
effective-dated messages 379
See Also full data publish
creating a publish rule definition 382
defining chunking rules 381
defining message routing 384
effective-dated rows
publishing from the delay table 386
Enterprise Integration Points 335
See Also integration points
Entry Definition page 28
Entry Membership Rules page 30
entry membership rules, setting up 28
EO EIP Data Maint page 375
EOEC_CCI_RESPONSE message 76
EOEC_CCI_SYNC message 76
EOEC_CCI_TRANSACTION business interlink 75

EOP_RATECALC process
running 65
understanding 65
EOPP_CPPROJ 325
EOPP_FD_DEL 325
EOPP_SCSYNC 323
error correction
message setup and maintenance pages 371
messages 376
error handling utility
creating error correction pages 371
definition 369
row security 374
setting up 372
testing 375
workflow notification 375
error management process 369
exchange rates, accessing 63
Execute All option 127
Execute Limited Number option 127
execution options 127

F

File Inbound page 391
file layouts 334
flat file utility
about 389
inbound file processing 389, 393
inbound file processing pages 391
inbound file processing setup 391
inbound file processing testing 394
formatting
addresses 44
formatting data 138
fraud screen service, using for credit card processing 101
full data publish
creating a message definition 380
creating the run control 382
defining a message channel 380
defining message nodes 380
Full Data Publish page 382
Full Table Publish Rules page 349

G

generic contexts 143
generic implementations 138
glossary 421

I

implementations
contextual 138
creating a PS query 180
creating an application class 177
definition 134
definition of 134
generic 138
registering 135
testing terms 140
Inbound File page 393
incremental messages
creating the message definition 385
effective-dated delay function 385
Integration Broker 334
integration points 335
See Also message integration points
naming standards, nouns 414
naming standards, verbs 411

J

Java interlink plug-in
setting up for credit card processing 86
setting up for credit card processing on Solaris 88, 90
setting up for credit card processing on Windows NT 87, 88
understanding credit card processing with 86

L

Languages page 352
LDAP Data Interface Format files
synchronizing data using 10
LDAP data, synchronizing with PeopleSoft data 9
LDAP directory
configuring for Directory Interface 12
setting up authentication for Directory Interface 16
setting up mappings for use with Directory Interface 19
LDIF, *See* LDAP Data Interface Format files
location numbers 170

M

managing contexts 143

manual option, processing credit cards with 75
Map Business Unit page 363
Map Details page 20
Map SetID page 364
mapping, messages 28
market rate definitions, creating 59
market rate indexes, defining 57
market rate types, defining 58
market rates
accessing exchange rate details 63
creating definitions 59
defining 56, 61
defining indexes 57
defining types 58
describing tables 49
understanding 49
message chunking
creating a chunking rule 359
overview 356
pages used to set up 356
selecting chunking fields 357
when to use 357
message integration points
activating 338
setting up 337
setup pages 337
message publishing
from batch programs 344
from online component 343
message record mapping
batch publish 354
full publish 350
message routing 340
messages, delivered 28
messaging 333
MMA Partners xxii
mobile data distribution
defining rules 72
understanding 69, 71
using user rules 74
Modify Connect DN - Directory Interface page 22

N

naming terms 170
Navigation Collection options
defining 280
defining at the registry level 285
defining at the system level 281
describing default hierarchy for 279
Navigation Collections
adding folders to 306
adding links to 312
creating 303
describing images used for 298
describing publication options for 292
editing 303
editing folders in 311
editing links in 316
maintaining 299
publishing 318
searching for 301
selecting images for 310
synchronizing security for 323
understanding 291
navigation pagelets, describing 292
navigation pages
describing 292
describing common elements on 273
describing common elements used on 275
describing folder and link sequence on 295
enabling caching for 277
understanding 273
using breadcrumbs on 273
nodes 339
See Also chunking rules, adding nodes to
notes xxv

O

online contexts 143
operands
location number 170
operator
selecting 121
operator sets
registering 165
operators
registering 164

P

page processor 343
Pagelet Wizard
administering pagelets 255
copying pagelets 256
defining pagelet categories 214
defining transformation types 201, 202

defining XSL prototypes 207
deleting pagelets 256
describing display sizes 195
generating pagelet export and import scripts 259
overview 185
publishing a pagelet to multiple portals 263
publishing pagelets to another portal 264
publishing pagelets to other portals 262
reviewing pagelet details 266
setting up options 201
specifying pagelet information 217
specifying publication options 249
using 217
Pagelet Wizard data source parameters
specifying 224
specifying for Navigation Collections 230
specifying for queries 227
specifying for search records 228
specifying values for 231
using system variables as 224
Pagelet Wizard data sources
describing 188
selecting 219
selecting a Navigation Collection 223
selecting a query 219
selecting a search record 222
selecting HTML 221
Pagelet Wizard data types
defining 205
Pagelet Wizard data types, defining 201
Pagelet Wizard display formats
defining 201, 203
describing 189
describing the chart format 190
describing the custom format 194
describing the list format 190
describing the menu format 191
describing the passthru format 194
describing the search list format 192
describing the table format 189
selecting 231
Pagelet Wizard display options
specifying 233
specifying data links 248
specifying for charts 243
specifying for customs 246
specifying for lists 241
specifying for menus 239
specifying for passthrus 238
specifying for search lists 238
specifying for tables 236
Pagelet Wizard footers
defining 209, 211
defining images 212
Pagelet Wizard headers
defining 209
defining images 210
Pagelet Wizard pagelet types
describing 196
describing the homepage type 196
describing the template type 197
Pagelet Wizard post-transformation processing
formatting currency elements 269
formatting date elements 269
formatting datetime elements 269
formatting number elements 269
formatting time elements 269
generating charts 271
inserting content reference links 270
inserting content reference URLs 271
inserting footers 271
inserting headers 271
inserting message catalog text 268
understanding 267
using 267
pagelets
publishing to another portal 264
publishing to multiple portals 263
publishing to other portals 262
pages
add nodes to chunk rule 339, 340
batch publish rules 338, 353
BusUnit mapping 360
chunking rule 360, 379
data maintenance setup1 372
data maintenance setup2 374
effective date publish 387
EO EIP data maint 375
file inbound 391
full data publish 382
full table publish rules 349
inbound file 393
languages 352
map business unit 363
map SetIDs 364

quick map 362
record mapping 350
SetID mapping 361
pages used to
set address formats 44
PeopleBooks
ordering xxii
PeopleCode functions, setting up 24
PeopleCode, typographical conventions xxiv
PeopleSoft Active Analytics Framework
definition of 113
installation options 157
PeopleSoft Application Designer plug-in
configuring account access information for credit card processing 91
configuring for credit card processing 90
configuring proxy server support for credit card processing 94
setting parameter check logic for credit card processing 95
setting the decryption option for credit card processing 95
setting up tracing for credit card processing 94
PeopleSoft application fundamentals xxi
policies
activating 124
associating to another trigger point 124
building and managing 117
copying 124
definition of 113
editing 118
prerequisites to building 117
reusing 126
policy
activating 124
policy groups 118
setting execution options 127
policy or policy group
adding 126
removing 125
reordering 126
portal projects, cleaning 325
portal utilities
cleaning portal projects 325
creating Navigation Collections 299
defining at the registry level 285
defining at the system level 281
defining Navigation Collection options 280
deleting empty portal registry folders 325
publishing Navigation Collections 318
running batch processes 323
synchronizing Navigation Collection security 323
using Navigation Collections 291
preconditions
adding 126
prerequisites xxi
printed documentation xxii
PS query implementation
creating 180
PS Query implementation 135
publish rules
assigning an application program to 354
associating to a message 349
characterizing 349
full table, assigning 348
languages, specifying 352
overview 347

Q

Quick Map page 362

R

rate definitions, creating 59
rate types, defining 58
RATE_DIV, understanding 51
RATE_MULT, understanding 51
reciprocal rates, calculating 65
Record Mapping page 350
Record.Field implementation 135
registry options, defining for Navigation Collections 285
related documentation xxii
related languages
guidelines for messaging 342
scenarios 345
setup 341
resolution methods
registering 166
row security 374
See Also error handling utility

S

SetIDs Mapping page 361
setting execution options 127
  guidelines 131
Shortcut Collections, *See* Navigation Collections
SQL object implementation 135
standard navigation pages
  describing folder and link sequence on 295
  enabling 276
  understanding 273
State Table page 44, 46
state, customizing address formats 43, 46
subject areas 114
  defining 162
suggestions, submitting xxvi
Sync Collection Security process
  running 324
  understanding 323
system options, defining for Navigation Collections 281

T

term codes 171
term implementations
  testing 140
term properties 136
terms 421
  configuration of 137
  configuring 120
  creating 136
  exposing component buffer fields 169
  managing 138
  methods of an application class
    resolving 178
  naming 170
  prompt details 137
  registering 136
  scope of implementation 137
  user binds 136
Test Credit Card Interface - Card Entry/Display page 81
Test Credit Card Interface - Transaction page 83
TouchNet, understanding 75
triangulated rates, calculating 65
triangulation, understanding 50
trigger points
  creating 171
  definition of 113
  enabling in a PeopleCode event 172
  hierarchy 125
  managing 124
  registering 161
  registration of 114
  setting execution options 127
trigger types
  registering 160
typographical conventions xxiv

U

understanding address formats 43
Use Parent Execution Options option 127

V

visual cues xxv
visual rate, understanding 51

W

warnings xxv
workflow notification 375
  See Also error handling utility

X

XML Schema utility
  examples 402
  extracting table relationship 401
  overview 401
XML-based integrations, understanding credit card processing with 76
XML-based vendors
  processing credit cards with 75