ORACLE
PEOPLESOFT

PeopleTools 8.54: Portal Technology

November 2016

ORACLE

PeopleTools 8.54: Portal Technology
CDSKU
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Contents

Preface ..... **xi**
Understanding the PeopleSoft Online Help and PeopleBooks xi
PeopleSoft Hosted Documentation xi
Locally Installed Help xi
Downloadable PeopleBook PDF Files xi
Common Help Documentation xii
Field and Control Definitions xii
Typographical Conventions xii
ISO Country and Currency Codes xiii
Region and Industry Identifiers xiii
Using and Managing the PeopleSoft Online Help xiv
PeopleTools Related Links xiv
Contact Us xiv
Follow Us xiv
Part I: Getting Started with Portal Technology **17**
Chapter 1: Getting Started with Portal Technology **19**
Portal Technology Overview 19
Portal Technology Implementation 19
Chapter 2: Understanding PeopleSoft Pure Internet Architecture **23**
PeopleSoft Pure Internet Architecture Fundamentals 23
Database Server 24
Application Servers 24
Batch Server Environment 29
Web Server 30
Web Browser 32
Configuration and Deployment Options 33
Chapter 3: Understanding Portal Technology **37**
Portal Architecture 38
Node Definitions 40
Portal Templates and Template Pagelets 40
Navigation 41
Portal Registry 42
Portal Servlets 44
Portal URL Formats 46
Page-Based Template Proxying 52
Part II: Working with PeopleTools Portals **55**
Chapter 4: Working with Navigation Pages and Navigation Collections **57**
Understanding Navigation Pages 57
Understanding Navigation Collections 61
Common Elements Used on Navigation Pages 65
Setting Up Navigation Collection Options 66
Creating and Maintaining Navigation Collections 75
Publishing Navigation Collections 95
Chapter 5: Administering Portals **101**
Understanding Portal Administration 101
Understanding Folder Navigation 102

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Common Elements Used in This Chapter.... 104
Administering Portal Definitions.... 106
Administering Folders.... 109
Administering SmartNavigation Folders.... 115
Administering Content References.... 139
Managing Folders and Content References.... 158
Managing General Portal Settings.... 163
Managing Portal Objects.... 167
Reviewing Menu Item Information.... 170
Publishing Pagelets to Other Portals.... 176
Running Portal Utility Processes.... 180
Chapter 6: Using PeopleTools Branding Features.... 189
Understanding PeopleTools Branding.... 189
Creating System Definitions and Objects for Branding.... 194
Administering System Branding.... 203
Applying Branding to Other Objects.... 217
Additional Branding Tasks.... 221
Chapter 7: Modifying the Portal Interface.... 241
Using Predefined Template HTML Definitions.... 241
Using Predefined Homepage HTML Definitions.... 242
Using Custom Portal Headers.... 250
Changing Link Colors.... 250
Applying Changes.... 251
Chapter 8: Working with Portal Templates.... 253
Understanding Portal Templates.... 253
Applying Template Types.... 253
Developing Portal Templates.... 265
Chapter 9: Using Portal Caching Features.... 271
Understanding Portal Caching.... 271
Administering Server-Based Caching.... 272
Administering Browser-Based Caching.... 278
Administering Web Server-Based Navigation Caching.... 282
Chapter 10: Configuring the Portal Environment.... 285
Understanding the Authentication Domain.... 285
Configuring Web Profiles.... 287
Defining Portal Nodes.... 326
Implementing Single Signon Functionality.... 329
Redirecting Pages with Sensitive Data.... 331
Overriding Page Encoding.... 331
Importing Menu Groups into the Portal Registry.... 331
Improving Same-Server Performance Under SSL.... 336
Using SSL Accelerators.... 337
Using Reverse Proxy Servers.... 338
Using Firewalls with Reverse Proxy Servers.... 341
Using Reverse Proxy Servers with SSL Accelerators.... 342
Applying Load Balancing.... 344
Chapter 11: Working with Conditional Navigation.... 347
Understanding Conditional Navigation.... 347
Configuring Conditional Navigation.... 350
Overriding Conditional Navigation.... 358
Working with Other PeopleTools Features in Conditional Navigation.... 359

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Troubleshooting Conditional Navigation....364
Part III: Working with Homepages, Dashboard Pages, and Pagelets....367
Chapter 12: Administering Homepages and Dashboard Pages....369
Understanding Homepages and Dashboard Pages....369
Managing Homepages....370
Managing Dashboard Pages....380
Searching for and Importing Pagelets....388
Chapter 13: Using Pagelet Wizard to Create and Manage Pagelets....393
Understanding Pagelet Wizard....393
Understanding Pagelet Wizard Data Sources....394
Understanding Pagelet Wizard Display Formats....396
Understanding Pagelet Wizard Display Sizes....402
Understanding Pagelet Wizard Pagelet Types....403
Defining Pagelet Wizard Setup Values....404
Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types....406
Defining Pagelet Wizard XSL Prototypes....414
Defining Pagelet Wizard Headers and Footers....416
Defining Pagelet Wizard Pagelet Categories....422
Step 1: Entering Pagelet Identifying Information....423
Step 2: Selecting a Pagelet Data Source....425
Step 3: Specifying Pagelet Data Source Parameters....441
Step 4: Selecting a Pagelet Display Format....461
Step 5: Specifying Pagelet Display Options....463
Step 6: Specifying Pagelet Publishing Options....485
Using Pagelet Wizard Post-Transformation Processing....495
Administering Pagelet Wizard Pagelets....499
Chapter 14: Developing and Registering Pagelets....511
Understanding Pagelets....511
Understanding PeopleSoft Portal Architecture....517
Understanding Pagelet Development....525
Understanding Single Signon and Pagelets....543
Developing PeopleSoft Component-Based Pagelets....550
Developing iScript-Based Pagelets....551
Developing Java-Based Pagelets....553
Developing Contextual Embeddable Pagelets....568
Administering Pagelets in the Portal Registry....569
Using Attributes to Enhance Pagelets....574
Handling Special Situations....580
Chapter 15: Configuring OBIEE Report Pagelets....583
Understanding OBIEE Integration Within the PeopleSoft Framework....583
Configuring OBIEE Server Setup....585
Configure the Web Server Files....586
Loading OBIEE Report Definitions....586
Viewing OBIEE Report Details....588
Creating OBIEE Report Pagelets....592
Part IV: Working with Related Information....595
Chapter 16: Configuring Interwindow Communication....597
Understanding Interwindow Communication....597
Publishing and Subscribing to IWC Messages....602
Chapter 17: Developing and Configuring Related Content Services....607
Understanding the PeopleSoft Related Content Framework....607

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Related Content Services....619
Managing Related Content....646
Configuring Related Content, Related Actions, and Layout....653
Chapter 18: Creating and Configuring WorkCenter Pages....679
Understanding WorkCenter Pages....679
Understanding WorkCenter Pagelets....686
Understanding the WorkCenter Development and Implementation Process....697
Managing WorkCenter Pages....698
Creating and Managing WorkCenter Dashboards....706
Defining WorkCenter Page Layout Templates....710
Chapter 19: Developing and Deploying Activity Guides....713
Understanding Activity Guides....713
Creating and Maintaining Activity Guide Templates....717
Creating and Maintaining Action Items....727
Deploying Activity Guides....749
Managing Activity Guide Instances....753
Enhancing Activity Guides....756
Migrating Activity Guide Data....767
Part V: Working with Portlet Technologies....773
Chapter 20: Understanding Web Services for Remote Portlets (WSRP)....775
Understanding WSRP....775
Terminology....777
Implementing WSRP Protocol Scenario....778
Considerations for WSRP and Server Cluster Configuration....779
Chapter 21: Configuring WS-Security for WSRP Consumption and Production....781
Determining Security Requirements....781
Understanding the PeopleSoft WS-Security for WSRP....782
Setting Up WSRP Security Options....788
Setting Up SAML....792
Configuring WS-Security for PeopleSoft as a WSRP Producer....794
Defining Nodes with WS-Security....795
Chapter 22: Using WSRP to Consume and Produce Remote Portlets....799
Understanding Use of WSRP to Consume and Produce Remote Portlets....799
Setting Up the Local Gateway URL (Consumer)....799
Using the WSRP Consumption Wizard (Consumer)....800
Displaying Remote Portlets on a Homepage (Consumer)....806
Producing Remote Portlets (Producer)....809
Chapter 23: Working with JSR 168 Portlets....819
Understanding JSR 168....819
Understanding PeopleSoft Compliance with JSR 168....819
Configuring the Portlet Container....820
Describing the Portlet Entity Registry....820
Appendix A: Understanding Changes in Portal Configuration Settings....823
Understanding Changes in Portal Configuration Settings....823
General Properties....823
Security Properties....824
Virtual Addressing Properties....825
Cookie Rules....826
Caching Settings....826
Trace and Debug Properties....828
Look and Feel Settings....828

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix B: Configuring the Application for 8.4x Navigation.....831
Understanding Configuration of PeopleSoft Applications for 8.4x Navigation.....831
Reconfiguring the Default Template.....831
Reconfiguring the Navigation Pagelet.....833
Appendix C: Converting Frames-Based Templates to iFrame-Based Templates.....837
Understanding Conversion of Frames-Based Templates to iFrame-Based Templates.....837
Converting the eProcurement Shopping Cart Template.....837
Gathering Information About the Current Template.....837
Creating the iFrame Template iScript.....838
Creating the iFrame Template HTML Definition.....838
Setting Security for the Template iScript.....841
Converting the Template Content Reference.....842
Troubleshooting Errors.....842
Appendix D: Replacing SmartNavigation Images.....843
Understanding SmartNavigation Icons.....843
Creating Replacement Images.....844
Defining New Style Classes.....844
Modifying Folder Settings.....844
Writing PeopleCode.....845
Rowset Data Source.....845
Tree Data Source.....845
Appendix E: Related Content Reserved Words and Service Parameters.....849
Reserved Words Default Mapping.....849
Appendix F: Configuring UPK Topics as Related Content Services.....851
Understanding Oracle UPK and Related Content.....851
Prerequisites.....852
UPK Player Package Links.....853
Service Definitions.....856
Related Content Menus and Frames.....856
Creating UPK-Based Related Content Service Definitions.....857
Modifying the UPK Player Package Link.....857
Creating the UPK-Based Service Definition.....858
Assigning UPK-Based Services as Related Content.....860
Appendix G: Configuring OBIEE-Based Related Content Services.....861
OBIEE Configuration Prerequisites.....861
Appendix H: Configuring PeopleSoft Applications to Use Oracle WebCenter Related Content Services.....863
Understanding the Oracle WebCenter Integration.....863
Understanding WebCenter Spaces.....863
Understanding WebCenter Services.....864
Understanding Security Synchronization.....867
Related Language Considerations.....868
Prerequisites.....868
Configuring Oracle WebCenter Options.....869
Setting the WebCenter Integration Options.....870
Synchronizing Oracle WebCenter and PeopleSoft Application Security.....871
Running the WebCenter Security Sync Batch Process.....871
Managing WebCenter Links.....871
Managing Instance-Specific Links.....872
Managing Shared Links.....872
Adding Group Space Links.....873

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Adding External URL Links....874
Appendix I: Enabling Locale Switching Through Query Parameters in Oracle WebCenter....875
Registering Provider Classes....875
Appendix J: Configuring OBIEE Admin User Permissions in Enterprise Manager 11g....877
Configuring OBIEE Server Admin User Permissions in Enterprise Manager 11g....877
Appendix K: Language Support for Consuming and Producing Remote Portlets....879
Language Support for Consuming Remote Portlets....879
Language Support Rules for Producing Remote Portlets....879
Appendix L: Overriding the Default Breadcrumb Behavior....881
Understanding the Default Breadcrumb Behavior....881
Overriding Default Breadcrumbs Using Content Reference Attributes....883
Appendix M: PeopleTools Web Libraries....885
Understanding Web Libraries....885
PeopleTools Web Libraries....885

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft Applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

PeopleSoft Hosted Documentation

You access the PeopleSoft Online Help on Oracle's PeopleSoft Hosted Documentation website, which enables you to access the full help website and context-sensitive help directly from an Oracle hosted server. The hosted documentation is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support, because that documentation is now incorporated into the hosted website content. The Hosted Documentation website is available in English only.

Note: Only the most current release of hosted documentation is updated regularly. After a new release is posted, previous releases remain available but are no longer updated.

Locally Installed Help

If your organization has firewall restrictions that prevent you from using the Hosted Documentation website, you can install the PeopleSoft Online Help locally. If you install the help locally, you have more control over which documents users can access and you can include links to your organization's custom documentation on help pages.

In addition, if you locally install the PeopleSoft Online Help, you can use any search engine for full-text searching. Your installation documentation includes instructions about how to set up Oracle Secure Enterprise Search for full-text searching.

See *PeopleTools Installation* for your database platform, "Installing PeopleSoft Online Help." If you do not use Secure Enterprise Search, see the documentation for your chosen search engine.

Note: Before users can access the search engine on a locally installed help website, you must enable the Search portlet and link. Click the Help link on any page in the PeopleSoft Online Help for instructions.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The two main types of common help are:

- Application Fundamentals
- Using PeopleSoft Applications

Most product families provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product family. Whether you are implementing a single application, some combination of applications within the product family, or the entire product family, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

In addition, the *PeopleTools: Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft Applications.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

Typographical Convention	***Description***
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.
. . . (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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Typographical Convention	*Description*
& (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.
⇒	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region and Industry Identifiers

Information that applies only to a specific 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 region-specific heading: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in the PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Industry Identifiers

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

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

Using and Managing the PeopleSoft Online Help

Click the Help link in the universal navigation header of any page in the PeopleSoft Online Help to see information on the following topics:

- What's new in the PeopleSoft Online Help.
- PeopleSoft Online Help acessibility.
- Accessing, navigating, and searching the PeopleSoft Online Help.
- Managing a locally installed PeopleSoft Online Help website.

PeopleTools Related Links

Oracle's PeopleSoft PeopleTools 8.54 Documentation Home Page (Doc ID 1664613.1)

"PeopleTools Product/Feature PeopleBook Index" (PeopleTools 8.54: Getting Started with PeopleTools)

PeopleSoft Information Portal

My Oracle Support

PeopleSoft Training from Oracle University

PeopleSoft Video Feature Overviews on YouTube

Contact Us

Send us your suggestions Please include release numbers for the PeopleTools and applications that you are using.

Follow Us

Get the latest PeopleSoft updates on Facebook.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Follow PeopleSoft on Twitter@PeopleSoft_Info.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Part I

Getting Started with Portal Technology

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 1

Getting Started with Portal Technology

Portal Technology Overview

In PeopleTools, portal technology consists of PeopleSoft Pure Internet Architecture and the various systems, code, and standards used for creating and managing portals.

PeopleSoft Pure Internet Architecture enables internet application deployment through a browser, and enables you to take advantage of PeopleSoft intranet solutions, internet solutions, and integration technologies. PeopleSoft Pure Internet Architecture runs seamlessly in portals created and managed by PeopleTools portal technology.

In PeopleTools, portals are built on top of PeopleSoft Pure Internet Architecture and enable you to easily access and administer content from a variety of providers, such as PeopleSoft FSCM, CRM, and HCM applications; other Oracle applications; and many third party content suppliers. PeopleTools portals enable you to combine content from these multiple sources and deliver the result to users in a unified, consistent, and intuitive interface.

The main elements of the PeopleTools portal technology are a *portal servlet* and an *application server*. These two elements work together to provide common portal processing features such as page assembly, content search and management, navigation, and homepage or dashboard personalization.

The terms *portal technology, portal technologies, PeopleTools portal,* and *portal* are used interchangeably throughout these topics to refer to the core portal technology delivered with PeopleTools.

Note: Portal packs and PeopleSoft Interaction Hub are separate applications and provide their own documentation.

For information related to licensing terms and restrictions for PeopleTools portal technology, refer to the *Licensing Notes for Oracle's PeopleTools* document on My Oracle Support.

See My Oracle Support.

Portal Technology Implementation

To implement PeopleTools portals, you need to:

- Design and configure the portal environment.
- Install PeopleSoft Pure Internet Architecture.
- Configure your web profile.
- Configure application security.
- Set up the portal.
- (Optional) Set up JSR 168.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Configure performance features, such as caching.

Note: These implementation activities assume that you have installed your required databases, application servers, and web servers.

See the *PeopleTools Installation* for your database platform found on My Oracle Support.

Related Links

"PeopleTools Overview" (PeopleTools 8.54: Getting Started with PeopleTools)

Designing and Configuring the Portal Environment

In the designing phase of your implementation, take advantage of all Oracle resources for PeopleSoft applications including installation guides, certification requirements, red papers, and product documentation.

Build the physical network and server architecture that is necessary to support your implementation. Ensure that you consider the various facets of network management and security, including creating subnets with appropriate routers and switches; creating firewalls, proxy servers, and reverse proxy servers, and using Secure Sockets Layer/Transport Layer Security.

During implementation, you should research and understand how to:

Step	***Reference***
Size servers appropriately	See your hardware configuration guides and PeopleSoft minimum requirements.
Configure software and hardware to optimize speed and performance	See your network administration guides.
Set up optimal load balancing	See your server configuration guides and industry best practices.

Installing PeopleSoft Pure Internet Architecture

To install one or more PeopleTools portal sites, perform this step:

Step	***Reference***
Run the PeopleSoft Pure Internet Architecture setup program for your database.	See the *PeopleTools 8.53 Installation Guide* My Oracle Support.

Configuring Your Web Profile

During PeopleSoft Pure Internet Architecture setup, you'll be prompted to select one of the web profiles delivered with your PeopleSoft application or to specify a new web profile if no existing profile meets your needs. To configure your selected web profile, you perform the following step:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Step	Reference
Configure your selected web profile.	See Configuring Web Profiles. See "Understanding WebLogic" (PeopleTools 8.54: System and Server Administration). See "Understanding WebSphere Application Server within Your PeopleSoft Implementation" (PeopleTools 8.54: System and Server Administration).

Configuring Application Security

Applications come with roles and permissions that you can configure or use as delivered. Roles define the permission lists that are assigned to groups of people. A permission list defines a group of securable objects. A comprehensive role and permission list design is necessary to best manage which users can access each piece of content in the portal.

A powerful feature of PeopleTools security is the dynamic role. User membership to these roles is defined programmatically instead of by manually updating a membership list. Dynamic roles can simplify user role assignments significantly.

To secure access to content, you perform the following steps:

Step	Reference
Configure permission lists, roles, and user profiles.	See Configuring Web Profiles.
(Optional) Set up single sign-on.	See "Understanding Single Signon" (PeopleTools 8.54: Security Administration). See Implementing Single Signon Functionality. See "Configuring the LDAP Directory" (PeopleTools 8.54: Security Administration).

Setting Up the Portal

To set up the portal, you perform the following steps:

Step	Reference
Manage portal definitions.	See Administering Portal Definitions.
Define folders and set folder security.	See Administering Folders.
Define content references, create related links, and set content reference security.	See Administering Content References.
Add templates and set template security.	See Managing Portal Objects.
Create, configure, and secure homepages and dashboards.	See Understanding Homepages and Dashboard Pages.

(Optional) Setting Up JSR 168

JSR 168 is an emerging standard for portals running in a Java 2 Platform, Enterprise Edition architecture. It specifies content definitions much like pagelets, which present portal content defined according to the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

JSR 168 standard. Portlets are Java-based web components that you can register in the PeopleSoft portal. To set up JSR 168, perform the following step:

Step	*Reference*
Implement JSR 168 portlets.	See Describing the Portlet Entity Registry.

Configuring Performance Features

Properly configuring caching options significantly increases portal performance. To administer caching features for the portal, perform the following steps:

Step	*Reference*
Configure server-based caching.	See Administering Server-Based Caching.
Configure browser-based caching.	See Administering Browser-Based Caching.
Configure web server-based navigation caching.	See Administering Web Server-Based Navigation Caching.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 2

Understanding PeopleSoft Pure Internet Architecture

PeopleSoft Pure Internet Architecture Fundamentals

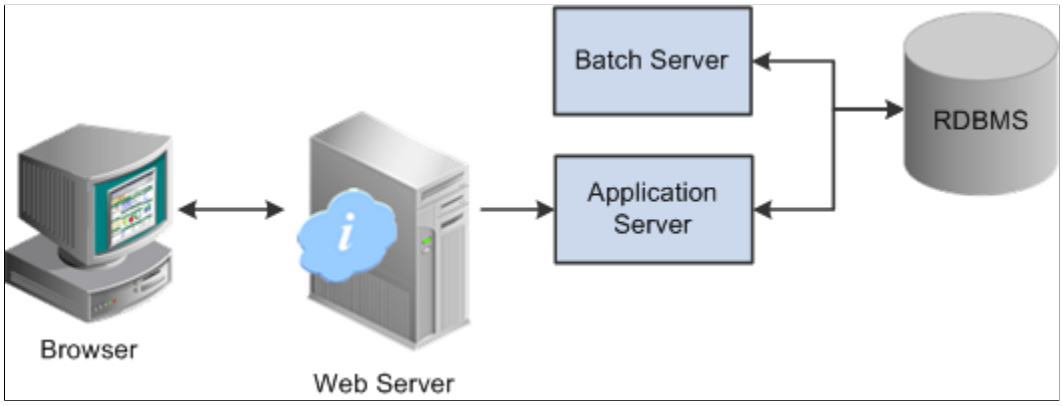
Your PeopleSoft application runs on PeopleSoft Pure Internet Architecture, which requires a variety of software and hardware elements:

- A relational database management system (RDBMS) server.
- An application server.
- A batch server.
- A web server.
- A web browser.

You need to understand the role of each element before you can decide which configuration will work best for your implementation of PeopleSoft Pure Internet Architecture.

Image: High-level overview of PeopleSoft Pure Internet Architecture relationships

The following diagram illustrates, at a high level, the physical relationship between the following PeopleSoft Pure Internet Architecture elements: the browser interacts with the web server; the web server interacts with the application server; and the application server interacts with the database server. In addition, the batch server also interacts with the database server.



Configuring PeopleSoft Pure Internet Architecture is not just about enabling internet application deployment through a browser. PeopleSoft Pure Internet Architecture enables you to take advantage of all PeopleSoft intranet and internet solutions, as well as PeopleSoft integration technologies such as PeopleSoft Integration Broker.

See the *PeopleTools 8.53 Installation Guides* at My Oracle Support.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Database Server

The database server houses a database engine and your PeopleSoft application database, which includes all the application's object definitions, system tables, application tables, and data. The database server must be running one of the PeopleSoft-supported RDBMS and operating system combinations.

A single database server can have multiple application servers connecting to it. The database server simultaneously handles the application server connections, development environment connections, and batch programs running against it.

Note: Using the PeopleTools development environment in Microsoft Windows, you can connect directly to the database, or indirectly through an application server.

The PeopleSoft database is the repository for all information managed by your PeopleSoft application. Both application data and PeopleSoft metadata are stored and maintained in the database. PeopleSoft Application Designer enables you to define and maintain this metadata, which the system uses to drive the runtime architecture. The application server runs business logic based on the metadata.

Use PeopleSoft Application Designer to define, configure, and modify PeopleSoft applications. You can create dozens of different types of application objects, such as fields, records, pages, and messages. When an application developer saves an application object, PeopleSoft Application Designer saves this definition to the metadata repository in the PeopleSoft database.

At runtime, the application server fetches the most recent application object definitions from the metadata repository, compiles and caches the application object into memory, and runs the business rules based on the definitions.

Application Servers

This section discusses:

- Application servers.
- Domains.
- Listeners, handlers, and queues.
- PeopleSoft server processes.
- Services.
- Oracle products.
- Database connectivity.

Application Servers

The application server is the core of PeopleSoft Pure Internet Architecture; it runs business logic and issues SQL to the database server. An application server consists of numerous PeopleSoft services and server processes. Just as different elements make up the physical environment in which an application server operates, such as database servers and web servers, a variety of elements operate on the application server, enabling it to respond effectively to a multitude of transaction requests and handle transaction processing, system scaling, browser requests, and so on.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

An application server maintains the SQL connection to the database for browser requests and the PeopleTools development environment in Microsoft Windows.

You should have a general knowledge of how an application server operates before you attempt to configure and tune it.

Related Links

Configuration and Deployment Options

Domains

An application server domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. You manage each domain with a separate configuration file, and you configure each application server domain to connect to a single database. A single application server machine can support multiple application server domains running on it. You configure an application server domain using the PSADMIN utility located in the *PS_HOME*\appserv directory on the application server.

A one-to-one or a many-to-one relationship can exist between application server domains and a database. In the simplest case, you configure a single application server domain to connect to a single PeopleSoft database. In a more sophisticated environment, you can configure multiple application server domains, with each domain connecting to the same PeopleSoft database. The opposite is not true; a single application server domain cannot be used to connect to multiple PeopleSoft databases.

For example, suppose that you have installed three databases, HRDMO1, HRDMO2, and HRDMO3, and you want to enable browser requests to each database. In this case, you must configure at least three application server domains, one for each database. As demand increases, you may need to configure multiple application server domains per database, for redundancy, fail-over, and performance reasons.

You can configure multiple application server domains under a single PeopleSoft configuration home directory, or *PS_CFG_HOME*. In this context, *PS_CFG_HOME* refers to the PeopleSoft high-level directory on the application server, not the file server. *PS_CFG_HOME* is the directory in which you installed the PeopleSoft application server configuration files when using the Server Transfer program.

PSADMIN creates a directory beneath *PS_CFG_HOME*\appserv for each application server domain that you configure. Using the previous HRDMO example, suppose that you decided to name the application server domains the same name as the database to which they connect. In this case, PSADMIN creates subdirectories \HRDMO1, \HRDMO2, and \HRDMO3 beneath the *PS_CFG_HOME*\appserv directory on the application server.

Starting an application server domain starts the set of server processes associated with that domain, such as PSAPPSRV, PSQCKSRV, and PSSAMSRV. Each server process establishes a persistent connection to a PeopleSoft database, and this connection acts as a generic SQL pipeline that the server process uses to send and receive SQL.

Each server process uses the same SQL connection to facilitate requests from multiple sources. From the RDBMS perspective, each server process within a domain represents a connected user.

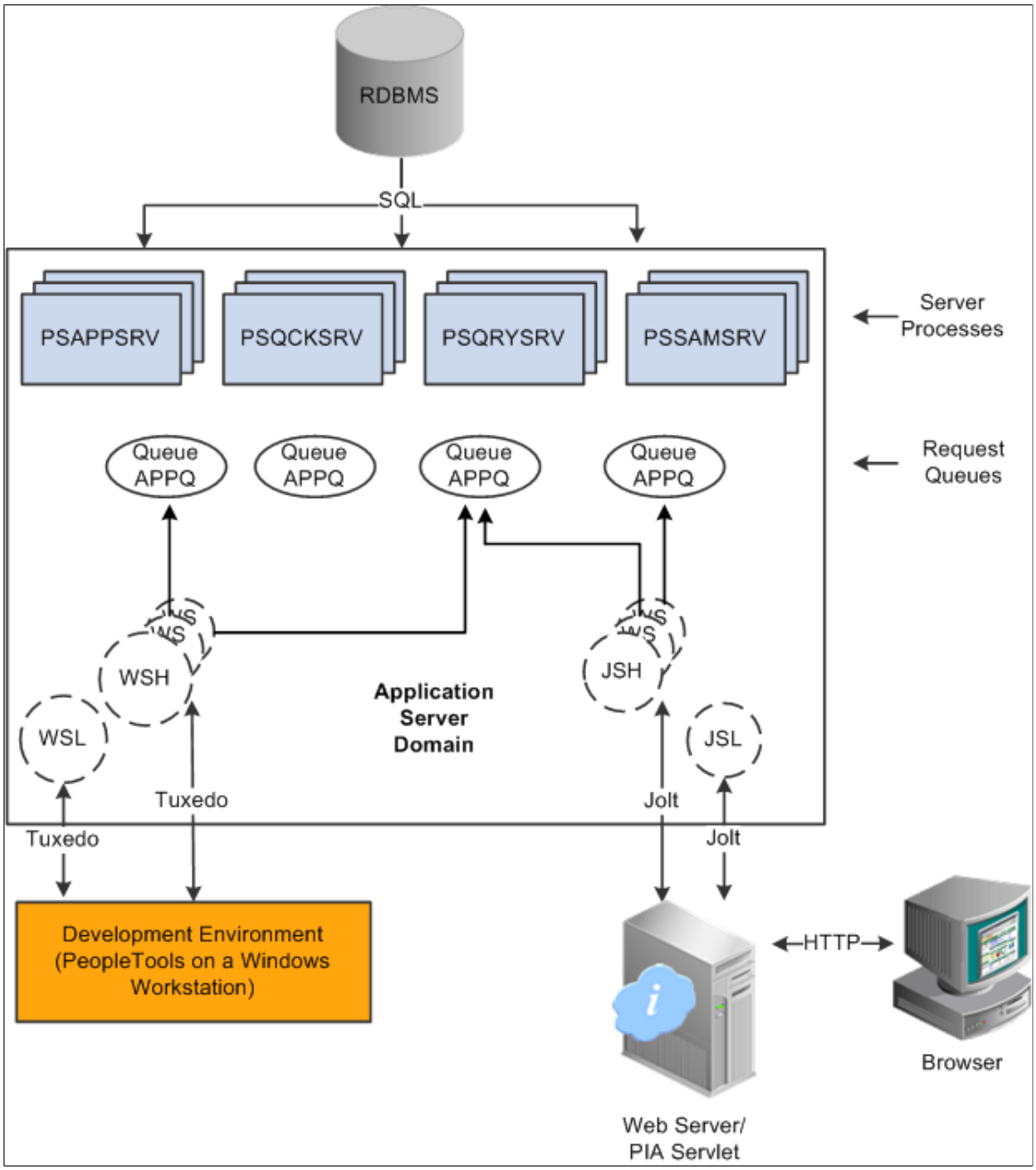
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Listeners, Handlers, and Queues

Image: Application server components and server processes

Listeners, handlers, and queues provide the basis of the application server functionality. Understanding the role of each element is essential when you configure and tune your application server. For instance, although you need to know how to configure multiple Java server handlers, you also need to know why and when to perform this task.

The following diagram illustrates application server processes:



For simplicity, the diagram does not depict every server process that runs on the application server. For example, the optimization server process, PSOPTENG, is not pictured, and integration server processes were also omitted.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: When PeopleSoft architecture mechanics is discussed, the term *service* is overused. The following statement may help to clarify this term: An application server domain calls server processes, such as PSAPPSRV, which in turn invoke services, such as MgrGetObject, on the database.

The following table describes each component depicted in the previous diagram:

Item	*Description*
Workstation listener (WSL)	The workstation listener monitors Oracle Tuxedo ports for initial connection requests sent from the PeopleTools development environment. After the workstation listener accepts a connection from a workstation, it directs the request to a workstation handler. From that point, the Microsoft Windows workstation interacts with the workstation handler to which it is assigned.
Workstation handler (WSH)	The workstation handler processes the requests that it receives from the workstation listener. A unique port number identifies a workstation handler. The port numbers for the workstation handler are selected (internally by Oracle Tuxedo) from a specified range of numbers. You can configure multiple workstation handlers to take care of demand increases; new processes are created as other processes become overloaded.
Oracle Jolt server listener (JSL)	The Oracle Jolt server listener applies only to browser requests. The Oracle Jolt server listener monitors the Oracle Jolt port for connection requests sent from the browser through the web server. After the Oracle Jolt server listener accepts a connection, it directs the request to an Oracle Jolt server handler. From that point, the browser interacts with the Oracle Jolt server handler. This is analogous to the relationship between the workstation server listener and workstation server handler.
Oracle Jolt server handler (JSH)	The Oracle Jolt server handler applies only to browser requests. The Oracle Jolt server handler processes the requests that it receives from the Oracle Java server listener. The port numbers for the Oracle Jolt server handler are selected internally by Oracle Tuxedo in sequential order.
Request queues	Each type of server process has a service request queue that it shares with other servers of the same type (as in PSAPPSRV on APPQ and PSQCKSRV on QCKQ). The workstation handler and Oracle Jolt server handler insert requests into the appropriate queue, and then the individual server processes complete each request in the order that it appears.
Server processes	The server processes act as the heart of the application server domain. They maintain the SQL connection and make sure that each transaction request gets processed on the database and that the results are returned to the appropriate origin.

PeopleSoft Server Processes

Multiple server processes run in an application server domain. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service, such as MgrGetObject.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Server processes invoke services to perform application logic and issue SQL to the RDBMS. Each application server process, such as PSAPPSRV, PSQCKSRV, PSQRYSRV, PSSAMSRV, or PSOPTENG, establishes and maintains its own connection to the database.

The server process waits for the service to finish, then returns information to the device that initiated the request, such as a browser. While a server process waits for a service to finish, other transaction requests wait in a queue until the current service finishes. A service may take a fraction of a second to finish or several seconds, depending on the type and complexity of the service. When the service finishes, the server process is then available to process the next request in the corresponding queue.

You need to configure only those server processes that your implementation requires per domain. The minimum server processes that a domain requires are PSAPPSRV and PSSAMSRV.

You can configure multiple instances of the same server processes to start when you start the application server domain. This helps you handle predicted workloads. Furthermore, Oracle Tuxedo can dynamically generate incremental server processes to handle increasing numbers of transaction requests. The capability to configure multiple server processes and generate incremental server processes contributes to the application server's scalability.

The following list describes the possible server processes included in an application server domain. Depending on the configuration options that you choose, not all of the server processes will necessarily be a part of every domain.

- PSAPPSRV

  This process performs functional requests, such as building and loading components (which were known as panel groups in previous releases). It also provides the memory and disk-caching feature for PeopleTools objects on the application server. PSAPPSRV is required to be running in any domain.

- PSQCKSRV

  This process performs quick, read-only SQL requests. This is an optional process designed to improve performance by reducing the workload of PSAPPSRV.

- PSQRYSRV

  This process is designed to handle any query run by PeopleSoft Query. This is an optional process designed to improve performance by reducing the workload of PSAPPSRV.

- PSSAMSRV

  This SQL application manager process handles the conversational SQL that is mainly associated with PeopleSoft Application Designer. This process is required to be running on any domain.

- PSOPTENG

  This optimization engine process provides optimization services in PeopleSoft Optimization Framework. You need to configure this process in a server domain only if you want to use the optimization plug-in delivered with PeopleSoft applications.

The following set of server processes is used for application messaging. (Your messaging domain must also contain PSAPPSRV and PSSAMSRV, the required server processes.)

- PSMSGDSP

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- PSMSGHND
- PSPUBDSP
- PSPUBHND
- PSSUBDSP
- PSSUBHND

Note: You can examine servers by using the ps -ef command in UNIX or Task Manager in Microsoft Windows. The PeopleSoft configuration utility, PSADMIN, also offers a monitoring utility.

Services

When a PeopleSoft application sends a request to the application server, it sends a service name and a set of parameters, such as MgrGetObject and its parameters. Oracle Tuxedo then queues the transaction request to a specific server process that is designed to handle certain services.

When a server process starts, it advertises to the system the predefined services it handles. You can see the association between the many services and server processes by reviewing the PSAPPSRV.UBB file.

Related Links

"Using PSADMIN Configuration Files" (PeopleTools 8.54: System and Server Administration)

Oracle Middleware

PeopleSoft software uses Oracle Tuxedo, a middleware framework and transaction monitor, to manage database transactions. PeopleSoft software also uses Oracle Jolt, a Java API and class library, as the layer that facilitates communication between the PeopleSoft servlets on the web server and the application server. Both Oracle Tuxedo and Jolt are required.

Although we document the Oracle components with respect to their function within the context of a PeopleSoft environment, we do not duplicate the documentation provided by Oracle. You should become familiar with the Oracle documentation that we ship along with our products. The Oracle documentation provides an extensive error catalog that serves as an invaluable source of information when you are troubleshooting.

Note: Oracle Tuxedo doesn’t actually perform processing on the application server; it schedules PeopleSoft server processes to perform the transactions.

See the *PeopleTools 8.53 Installation Guides* at My Oracle Support.

Database Connectivity

Application servers require database connectivity software to be installed locally to maintain the SQL connection with the RDBMS. You must install the required connectivity software and associated utilities for your RDBMS.

After the application server establishes a connection to the database, any device that initiates a transaction request through the application server takes advantage of the application server's direct connection to the database and therefore requires no connectivity software.

Batch Server Environment

This section discusses:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Batch server environment.
- Batch server support.
- PeopleSoft Process Scheduler and the application server.

Batch Server Environment

The batch server environment is where PeopleSoft Process Scheduler is installed and configured, and it is the location where many of your batch programs, such as Application Engine programs, run. In most situations, this is also where COBOL and Structured Query Report (SQR) executables are installed.

In a multiserver environment, you can decide where your site batch server environment resides. In PeopleSoft software, you can install the batch server on a separate server, or you can install the batch server on either the application server or the database server.

Related Links

"PeopleSoft Process Scheduler" (PeopleTools 8.54: Process Scheduler)

Batch Server Support

You can install PeopleSoft Process Scheduler on any supported application server, database server, or batch server, but you need to choose a location that’s supported in the PeopleSoft environment. At least two options are available for each database environment.

If you install PeopleSoft Process Scheduler on a machine that is a supported database machine but not a supported application server machine, you can still configure and administer PeopleSoft Process Scheduler using PSADMIN, the PeopleTools server administration utility. However, you won't be able to access application server options from PSADMIN.

PeopleSoft Process Scheduler and the Application Server

PeopleSoft uses PSADMIN to configure and administer both the application server and PeopleSoft Process Scheduler server. The PeopleSoft Process Scheduler setup procedure in PSADMIN provides a menu-driven interface to configure PeopleSoft Process Scheduler parameters and administer the Process Scheduler server agent.

Even though the application server and PeopleSoft Process Scheduler have PSADMIN as a common interface and share the directories under *PS_HOME* on the application server, they are separate entities. For instance, you start, configure, and shut down the application server and the PeopleSoft Process Scheduler server separately.

The application server uses Oracle Tuxedo to schedule transaction requests and send transaction results, and it issues SQL requests to the database server using the persistent connections of a collection of PeopleSoft application server processes.

In contrast, PeopleSoft Process Scheduler is a separate facility that's designed to poll the PeopleSoft database table, PSPRCSRQST, for inserted requests and initiate Application Engine, COBOL, SQR, and other batch processes.

Web Server

A Java-enabled web server is required to support browser transaction requests and PeopleSoft application messaging technology. You install on the web server a collection of PeopleSoft Java servlets designed to handle a wide range of PeopleSoft transactions.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

This section discusses:

- Server software elements.
- PeopleSoft servlets.
- Oracle Jolt.

Server Software Elements

During the PeopleSoft installation, a variety of PeopleSoft Java servlets are installed on the web server, so you must have a supported servlet engine installed.

The following software runs on the PeopleSoft Pure Internet Architecture web server:

- Web services.

  Web services manage the web server software, such as Oracle WebLogic or WebSphere.

- Servlet engine.

  The servlet engine is the environment in which servlets run. This component is tied to the web services software, but in some cases you install it separately.

- Java servlets.

  Java is a platform-independent programming language used widely for web-based programs. Servlets are Java programs that run on the web server. The Java executables are required for the servlet engine to operate.

PeopleSoft Servlets

The following PeopleSoft servlets reside on the web server:

- Portal servlet.

  The portal servlet manages all of the requests and formatting when users access PeopleSoft applications through the PeopleTools portal. It manages content searches, content access, and homepage personalization.

- Integration gateway servlet.

  This servlet transmits publish/subscribe messages between message nodes. The gateway handles PeopleSoft-to-PeopleSoft messages, PeopleSoft-to-third-party messages, and third-party-to-PeopleSoft messages.

- Report repository servlet.

  This servlet enables users to easily access and distribute the output of batch reports, such as Crystal and SQR reports, that are run through PeopleSoft Process Scheduler over the internet. This servlet retrieves the report output in the report repository and serves it to the browser.

Oracle Jolt

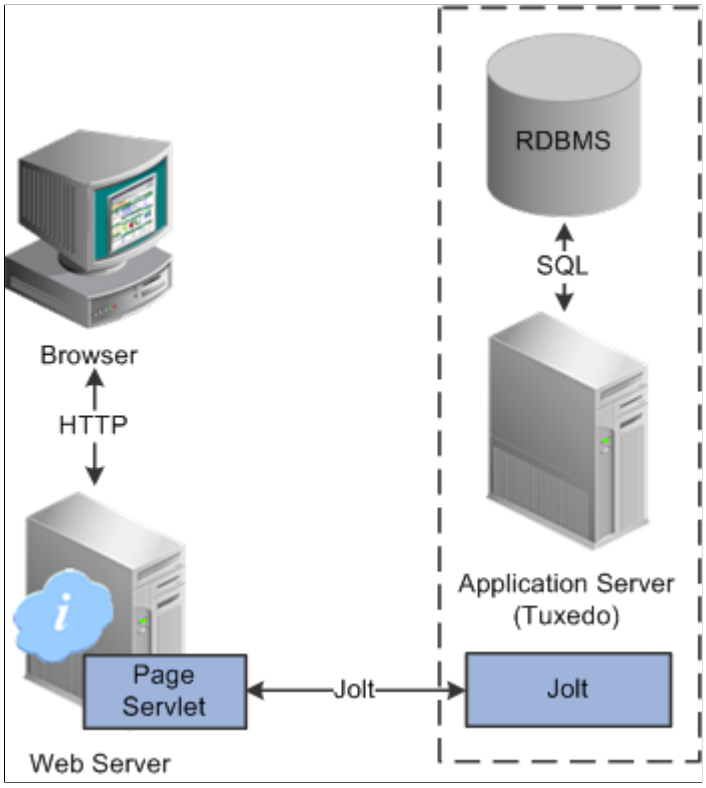
The PeopleSoft servlets on the web server transmit requests and data through a connection to Oracle Jolt, which runs on the application server. Oracle Jolt extends Oracle Tuxedo capabilities to the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

communication layer between the web-based environment and the C++ environments. You configure the servlets to direct requests from the web server to a predefined Oracle Jolt port on the application server.

Image: Oracle Jolt in relation to other components

Oracle Jolt must coexist with Tuxedo on the same application server machine. Oracle Jolt can't function without Tuxedo. The following diagram shows the relationship between PeopleSoft Pure Internet Architecture components—specifically, Jolt on the application server and servlets on the web server.



Web browsers don't connect directly to the application server. Instead, they send HTTP requests to the portal servlet running on the web server. The web server translates the HTTP request into an Oracle Jolt request that is sent to a specified Jolt port. Then the application server itself, leveraging Oracle Tuxedo, runs the appropriate SQL against the database.

Web Browser

The web browser is the primary means by which users and administrators access PeopleSoft applications and administrative tools.

You must make sure that a supported browser is installed on each workstation. You do not need to install other software on the workstation running the browser, such as applets or connectivity software. The system sends pure HTML to the browser.

A web browser uses the HTTP protocol. The browser sends a request to the web server, which forwards the request to the application server. A servlet installed on the web server facilities all browser connections.

The browser does not download any applets to complete a transaction. The application server sends only the following items to the browser:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- HTML
- XML
- JavaScript
- Cookies

Because the browser processes only this basic internet content, the client workstation is not burdened with unnecessary processing responsibility.

PeopleSoft Pure Internet Architecture leverages web browser cookies to store a unique access token for each user when the user is initially authenticated. When the user connects to another PeopleSoft system, the token in the browser cookie is used to reauthenticate the user and bypass the sign-in process. The browser cookie is an in-memory cookie and is never written to disk. The cookie is also encrypted to prevent snooping and uses a checksum to prevent tampering.

See the *Browser Compatibility Guide* at My Oracle Support.

Note: With PeopleSoft Pure Internet Architecture, no client software installation is necessary. Most processing occurs at the server level. You install the PeopleTools development applications in a Microsoft Windows environment. Development applications, such as PeopleSoft Application Designer and Data Mover, are usually intended for application developers and system administrators who need to access data other than application data. These users require workstations running Microsoft Windows.

Web Browser Cookies Setting

For you to use PeopleSoft Pure Internet Architecture, the Internet option to allow session cookies must be enabled.

To enable session cookies:

1. From your browser menu, select Tools, Internet Options, Security, Custom Level.
2. Scroll down to the Cookies – Allow per-session cookies (not stored) option.
3. Select Enable.
4. Click OK to save the setting.
5. Click OK again to close the dialog window.

Note: The internet option to allow cookies that are stored on your computer is optional.

Configuration and Deployment Options

This section discusses:

- Configuration options.
- Deployment options.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuration Options

You can configure your environment to support either a physical or a logical application server configuration. In some cases, the PeopleSoft standard installation procedure recommends one or the other depending on the combination of database and operating system at your site.

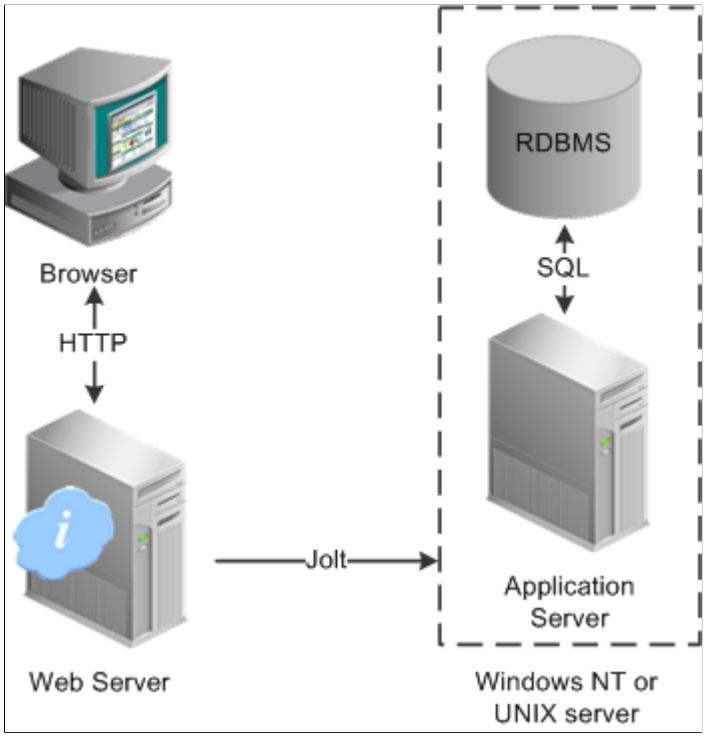
Generally, your application server should be on the same physical machine as the database server. This produces a logical separation between the application server and database server. If the application server and database server don't reside on the same machine, then the application server and the database server should be connected to the same high-performance backbone network. This ensures optimum performance.

Logical Application Server Configuration

A logical application server environment is one in which one or more servers share the same physical machine. The servers are logically, but not physically, separate.

Image: Logical application server configuration

The following diagram depicts a logical configuration with two server machines—one for the web server and the other for the application and database servers:



The dashed line surrounding the application server and the database server represents one physical machine. In this case, a logical application server configuration is possible only when both the database server and the application server are supported on a particular operating system. This shows that certain PeopleSoft Pure Internet Architecture elements can share the same machine.

Although this diagram depicts the application server and the database server sharing the same machine, the web server could also reside on the same machine with both the application server and the database server. The only requirement is that each component be supported by the underlying operating system. If all servers are located on the same machine, however, you should consider security issues. If you're deploying PeopleSoft applications to the internet, you will most likely want your web server outside of your network firewall and not on the same machine as the database server.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

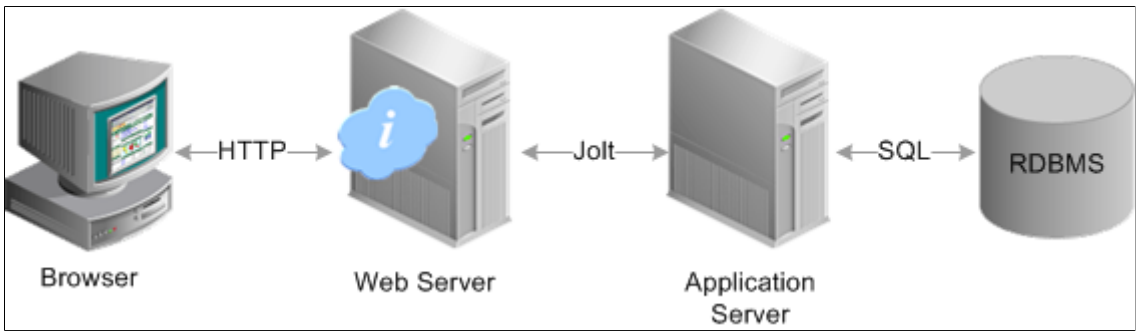
Note: For development, testing, or training purposes, you might want to have all PeopleSoft Pure Internet Architecture elements on the same Microsoft Windows or UNIX machine.

Physical Application Server Configuration

A physical application server configuration is one in which each component resides on a separate machine.

Image: Physical application server configuration

The following diagram depicts a physical configuration in which the application server is physically separate from the database server and the web server:



Deployment Options

A variety of user deployment options exist.

PeopleTools Portal

You can use the PeopleTools portal to deploy PeopleSoft applications to a browser. The portal enables you to integrate PeopleSoft content with content from other data sources. The PeopleSoft portal can be standalone, or you can integrate it with any application portal that you may already be using.

PeopleSoft portal technology consists of the portal servlet and an application server. These two components work together to provide common portal processing, such as page assembly, search ability, content management, navigation, and homepage personalization. You can combine content from a wide variety of data sources and deliver the result to users in a unified, simple-to-use interface.

Development Environment

Although the majority of users connect by using browsers, application developers and system administrators, who require access to PeopleTools, must use Microsoft Windows workstations. PeopleSoft Application Designer and various other PeopleTools applications are not accessible through a browser.

With the PeopleTools development environment in Microsoft Windows, you can connect directly to the database (two-tier), or you can connect through an application server (three-tier).

Integration Technologies

Although PeopleSoft integration solutions don't deploy a user interface, they do enable you to share information with third-party systems and other PeopleSoft databases. To take advantage of these integration solutions, you must configure PeopleSoft Pure Internet Architecture. Oracle provides the following integration technologies:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- PeopleSoft Integration Broker.

  This middleware technology facilitates synchronous and asynchronous messaging among internal systems and with trading partners, while managing message structure, message format, and transport disparities.

- PeopleSoft Component Interfaces.

  This object-oriented, request/reply, component architecture enables third-party applications to synchronously invoke PeopleSoft business logic.

- PeopleSoft Business Interlinks.

  This plug-in framework enables PeopleSoft applications to easily invoke third-party application programming interfaces over the internet.

Related Links

"Introduction to PeopleSoft Integration Broker" (PeopleTools 8.54: Integration Broker)

"Understanding Component Interfaces" (PeopleTools 8.54: Component Interfaces)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 3

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Understanding Portal Technology

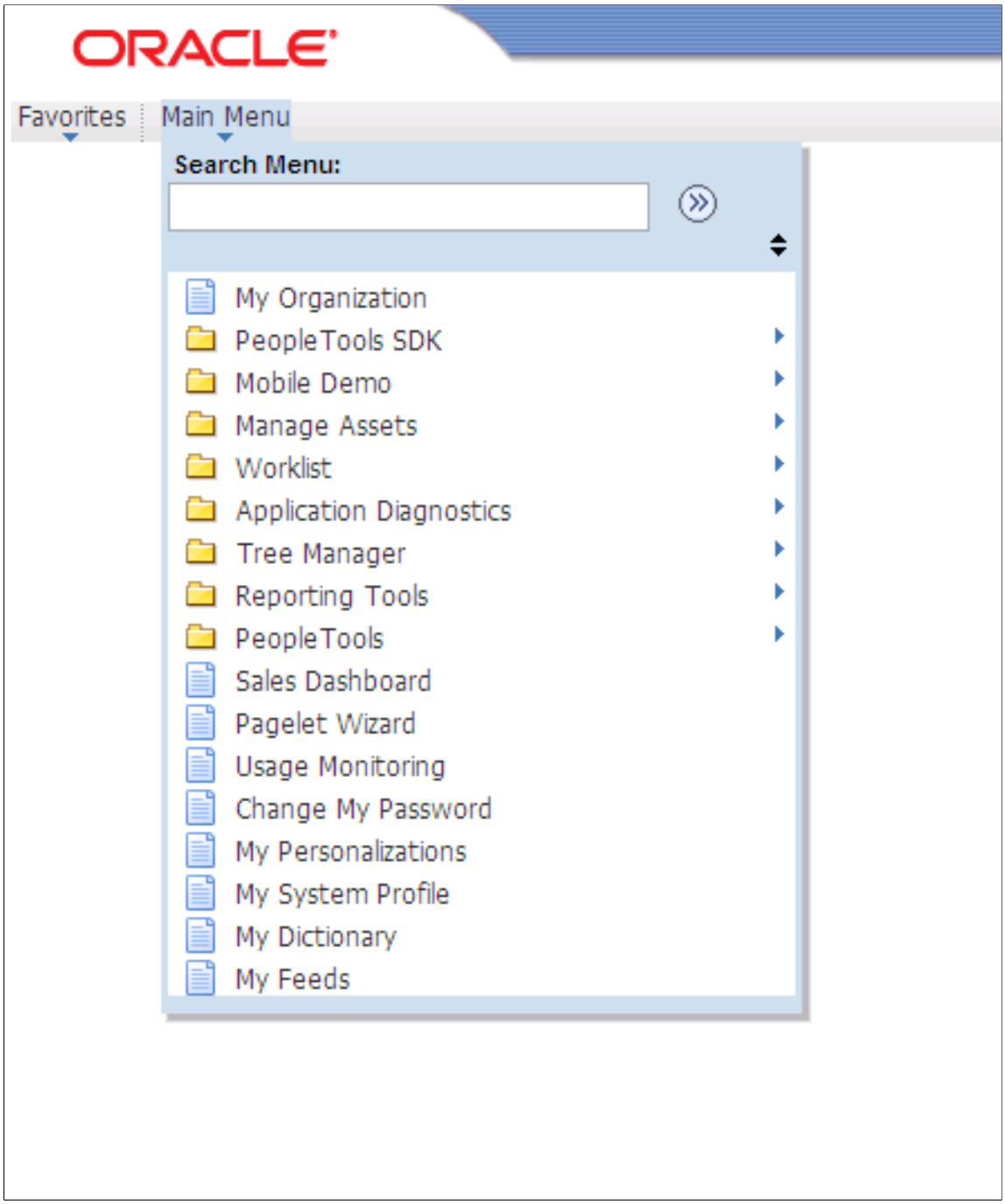
Portal Architecture

PeopleSoft portal technology provides basic web-based navigation for application pages.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Image: Example homepage with drop-down navigation and search

The following is a screenshot of homepage with drop-down navigation and search which contains the drop-down menu navigation and the portal menu search box, as shown in the following example:



The main features of the portal architecture are:

- Templates and template pagelets.
- Ability to register PeopleSoft transactions.
- The portal servlet.
- Navigation.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Favorites.
- Related information.
- Search.
- Single-sign on support for PeopleSoft applications.

Node Definitions

A node is a source of HTML content. A node is a URL string that defines the entire database and server. It is used when the portal servlet attempts to retrieve content, whether internal PeopleSoft content or external references, and assemble pages.

The use of nodes simplifies PeopleCode programming because long URLs can be referred to by the appropriate node name. Some node names, such as HRMS, EPM, SA, FDM, and CRM, are preset in your portal. You can add additional nodes.

Note: Node names can consist of any combination of letters, digits, and underscores, but they must not contain any spaces or special characters.

Related Links

"Node Class" (PeopleTools 8.54: PeopleCode API Reference)

"Understanding Nodes" (PeopleTools 8.54: Integration Broker Administration)

Portal Templates and Template Pagelets

Developers create portal templates in PeopleSoft Application Designer. At runtime, a template is transformed into a web page by PeopleTools portal technology. Each template is made from various template pagelets.

Each template (and each template pagelet) is designed individually and stored as an HTML object in the application database. Oracle delivers a default portal template for each installed PeopleSoft database—such as HRMS, ERP, and so on.

In addition to template pagelets, portal templates also contain special PeopleSoft tags that indicate where template pagelets are to be inserted in the template. These XML tags specify one or more regions of a page, the insertion point of the target page, and any other template pagelets that provide HTML for the other regions.

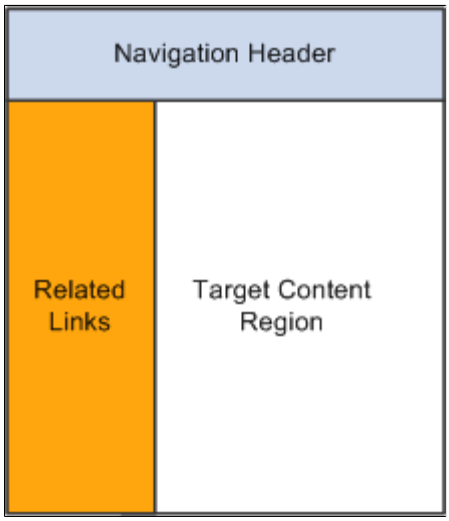
Each portal template reserves space for a target page, which contains the specific HTML page that a user requested. For example, if a user is running a PeopleSoft Financial application, the page that the user

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

is currently accessing appears in the target region, in addition to the other template pagelets displayed elsewhere on the page. The target region is typically the largest area of the template.

Image: Three template pagelets comprising a portal template

The template in the following example is composed of three separate template pagelets: one for the navigation header, one for related links, and one for the target content region. At runtime, the target content region is filled by the HTML returned by the target page, as are the other template pagelet regions.



Navigation

You can navigate in an application by using the menu and the universal navigation header.

Menu navigation is provided through a drop-down menu that enables you to move through the portal registry visually, using folders. Using the drop-down menu, you can view the registry by drilling down into the folders, accessing the favorites list, or using the search field.

The universal navigation header includes several links.

Search	Enter a term or phrase in the Search field and click the Go button to search for a registered content reference within the system.
Home	Click to return to the homepage.
Worklist	Click to access the Worklist page, which provides summary information about all items on your personal worklist. This page also provides links enabling you to view additional details about the work, perform the indicated work, and reassign work items. This is an optional setting that your system administrator enables; it might not appear to all users. See "Sending and Receiving Notifications" (PeopleTools 8.54: Applications User's Guide).
MultiChannel Console	Click to access and manage tasks assigned to you as an agent in PeopleSoft MultiChannel Framework. This is an optional setting

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	that your system administrator enables. If you do not have access to this feature, the system displays a warning message to that effect. See "Configuring PeopleSoft MCF Clusters" (PeopleTools 8.54: MultiChannel Framework) "Managing Tasks with the MultiChannel Console" (PeopleTools 8.54: MultiChannel Framework)
Performance Trace	Click to monitor the performance of a business process that you typically complete. This is an optional setting that your system administrator enables; it might not appear to all users.
Add to Favorites	Click to add the current page to your favorites list for easy accessibility.
Sign out	Click to terminate the user session and return to the Sign In page.

Related Links

"Understanding Verity Search Indexes" (PeopleTools 8.54: System and Server Administration)
"Understanding PeopleSoft Navigator" (PeopleTools 8.54: Applications User's Guide)
"Using the Universal Navigation Header" (PeopleTools 8.54: Applications User's Guide)

Portal Registry

Each portal is defined by a portal registry. The portal registry is a set of dedicated PeopleSoft database tables that store every content reference, typically a URL, that is available through the portal. A portal registry has a tree-like structure in which content references are organized, classified, and registered. A portal registry contains folders and content references.

Content references are definitions that have been entered or registered in the portal registry. Content references fall into four categories: homepages , target content, templates, and pagelets. Although each type of content reference differs slightly from the other types, every content reference includes information, such a name, label, creator, effective date, and expiration date.

Content references include a URL, which points to portal content. The content can be PeopleSoft components, iScripts, external web pages, or other external content. The URL can point to any website that responds to HTTP or HTTPS requests with an HTML response—any static or dynamic web page. One example of a content reference is a URL that points to a PeopleSoft application component. Other examples include static HTML pages on an intranet site or dynamic pages created by a reporting system. Access to content references is controlled by basic PeopleTools security through the assignment of permission lists and roles.

In the portal registry, folders organize content references into a multilevel hierarchy. Each folder can contain content references and other folders. Folders and content references share many of the same properties.

Every portal registry contains a root folder and standard PeopleTools-related folders including My Favorites, Portal Objects, Reporting Tools, PeopleTools, Tools - Hidden, and many others. The Portal Objects folder includes the following folders:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Homepage
- Navigation Collections
- Pagelets
- Template Pagelets
- Templates
- WSRP Produced Portlets

In addition to these standard folders, there are multiple folders per PeopleSoft application. These application folders contain the subfolders and content references associated with each PeopleSoft application you've licensed. All application pages in PeopleSoft product line databases are registered, secured, and described in the portal registry.

The hierarchical structure of the registry enables a portal administrator to create a navigation system, in which content references can be registered and managed for all portal users. PeopleTools provides a set of portal administration pages for this purpose. Additionally, a registry API is provided for accessing each portal registry from PeopleCode, Component Object Model (COM), or C programs.

A portal registry can reside in an existing PeopleSoft application database or in an independent PeopleSoft database. A single database can support multiple portal registries, and therefore multiple portals, but only one portal registry is associated with any given portal. To improve performance, a copy of the registry is also stored on the application server in cache files; this is how the portal servlet accesses the registry at runtime.

Maintaining the Portal Registry

Several tools exist to access and maintain the portal registry:

- Use the portal administration pages to manage portal and node definitions, folders, content references, templates, pagelets, navigation collections, and portlets.
- Use other portal administration pages to view menu security and structure.
- Use the portal registry API to programmatically maintain objects in the portal registry.
- Use the component registration wizard in PeopleSoft Application Designer to register and assign permissions to component-based content references.
- Use the Menu Import utility to import custom menu group definitions.
- Use the security synchronization process to reinstate the correct security relationships between objects in the portal registry after you copy a project that contains portal objects.

Related Links

Importing Menu Groups into the Portal Registry

"Understanding the Portal Registry" (PeopleTools 8.54: PeopleCode API Reference)

"Understanding the Registration Wizard" (PeopleTools 8.54: Application Designer Developer's Guide)

"Using the Portal Registry API" (PeopleTools 8.54: PeopleCode API Reference)

"Adding Node Definitions" (PeopleTools 8.54: Integration Broker Administration)

Node Definitions

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying the Node Type
"Node Class" (PeopleTools 8.54: PeopleCode API Reference)

Portal Servlets

This section discusses:

- Portal servlets.
- Page assembly for page-based templates.

Portal Servlets

A portal servlet is a Java servlet that runs on the portal web server. It intercepts a user request for an HTML page, retrieves the requested page, envelopes the page with additional content, and then sends the resulting page to the user's browser. It enables the portal to integrate content from a wide variety of sources and present the content on a single page in a coherent, consistent fashion.

The portal servlet performs these functions:

- Retrieves user-requested content.

  The content can be either target content, such as a PeopleSoft application page, for display in the large target region of the browser, or content for the smaller-sized pagelets.

  Note: Portal servlets do not accept HTTP PUT requests from clients. They accept only HTTP GET or POST data from clients. If the portal servlet is sent a PUT request, a 405 error page appears.

  See World Wide Web Consortium, Hypertext Transfer Protocol — HTTP/1.1.

- Provides a consistent user interface.

  The portal servlet checks properties associated with each content reference, including the name of a portal template. When a user accesses content through the portal, the portal servlet wraps the target page with the portal template specified in the content reference. This template provides a consistent user interface.

- Ensures that PeopleSoft-specific tags are processed correctly.

  Developers create portal pages using a template-based layout system. In addition to traditional HTML tags, templates can contain PeopleSoft-specific tags that a normal browser cannot interpret. The portal servlet can interpret these PeopleSoft-specific tags when constructing templates, and can interpret any other HTML content, at runtime. The portal servlet then sends the resulting page to a browser as a single HTML document.

The portal servlet's behavior changes slightly depending on whether it's handling page-based or frame-based templates.

Page-Based Template Handling

For page-based templates, the portal servlet assembles pages for the browser and ensures that all URL references in the HTML on the assembled pages are referenced back to the portal servlet itself.

The portal servlet receives user requests for content and then constructs a single, complete HTML page for the user's browser by combining the requested content with the appropriate portal template. This

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

process is called *page assembly.* The portal servlet uses a template to incorporate the contents of the requested page into the context of the site (headers, navigation, and so on). Each content reference can be associated with a template in the portal registry. The template tells the portal servlet what URLs to combine and how to arrange them in the user's browser.

Note: The portal servlet uses relative URLs. The portal servlet performs URL rewriting (proxying) only when the URL is generated by an external system or a PeopleTools release prior to 8.4.

Pages that the portal servlet assembles for the browser might point to numerous other web pages from different sources on the internet. Because the user's request goes through the portal servlet, the servlet must ensure that requests for content can be fulfilled during the page assembly process. In some cases, each URL in the HTML document assembled by the portal servlet must be rewritten to reference the portal servlet, not the originally requested URL. This process of redirecting URLs so that they point to the portal servlet is called *proxying.*

See Proxying and URL Conversion.

Frame-Based Template Handling

For a frame-based template, the portal servlet updates the src tags in the frameset with the target content before sending the page to the browser.

The portal servlet inserts a URL into each frame in the src tag and sends the page to the browser, rather than retrieving documents for the browser, as it does with page-based templates.

iFrame-Based Template Handling

For an iframe-based template, the portal servlet updates the src tags for each inline frame (iframe) element in the document with the target content before sending the page to the browser. The portal servlet inserts a URL into each frame in the src tag and sends the page along to the browser, rather than retrieving documents for the browser, as it does with page-based templates.

Portal Servlet Setup

A portal servlet must be set up properly on a web server before the portal can service user requests. The web server and portal servlet environment should be configured and tested during the installation of PeopleSoft applications.

See *PeopleTools Installation* for your database platform.

Page Assembly for Page-Based Templates

When a user clicks a link on a page-based template, the portal servlet receives the user's request, retrieves the content, properly formats it using a predefined portal template, and sends the assembled page back to the user's browser. The portal servlet also ensures that the user has the proper security permissions and is allowed to view the page.

The following processing steps occur during the page-assembly process:

1. The browser sends a request for content to the portal web server.

   The browser sends an HTTP request for target content; the request includes the target URL. The portal web server invokes the portal servlet. If the request includes a URL from a PeopleTools release prior to 8.4, the portal servlet then identifies the target content based on the query string parameter that was inserted there when the referring page was proxied.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. The portal servlet checks the portal registry to determine whether a content reference exists for the target URL.

   If a content reference exists for the target URL and the user does not have access to the content reference, the portal servlet responds with an error message. If the user does have access, the portal servlet continues its processing. If the content reference has been registered with a frame template, the portal servlet constructs the template and returns it as the response to the browser. The browser gets the content for each frame in the usual way. If the content reference has been registered as having no template, the servlet sends a redirect response to the browser for the original content. Otherwise, it goes on to the next step.

3. The portal servlet retrieves the appropriate template.

   If a registered content reference was found in the previous step, then the template associated with that content reference is used to wrap that content. If no template is associated, the portal servlet uses the template for the node associated with the content reference. If no node is associated with the content reference, then the default template for the portal is used. If no default portal template exists, the content appears without any template.

4. The portal servlet issues HTTP requests for content.

   The portal servlet issues an HTTP request for the requested content to the appropriate web server and receives an HTML document in return. It also issues an HTTP request for each pagelet used in the template.

5. The portal servlet merges content from previous versions of PeopleTools 8.4 from all HTML documents that it retrieved into the template HTML, and sends a single, complete HTML document back to the user's browser.

   The template HTML may contain special PeopleSoft tags, including the Target tag and the Pagelet tag. Each Target tag is replaced with whatever content is retrieved when the request for target content is fulfilled. Each Pagelet tag is replaced with the proxied content specified by that tag.

   If a style sheet is associated with the template, it is used. If not, the style sheet associated with the target content is used. Style sheets included in template pagelets are used if they are present. Additionally, the portal servlet ensures that cookies and headers returned in the responses for the template pagelets are merged into the main response.

Portal URL Formats

This section discusses:

- Basic portal URL format.
- URL format for PeopleSoft Pure Internet Architecture content types.
- URL format for unwrapped PeopleSoft Pure Internet Architecture content.
- Pagelet URLs.
- System URLs.
- Proxy architecture and relative URLs.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Basic Portal URL Format

URLs provide the addresses for content so that it can be located and correctly identified. The portal servlet needs three pieces of information to present a page to the user. These are integral parts of a PeopleSoft portal URL:

- The name of the portal through which content is being accessed.
- The node that hosts the content.
- The type and ID of the content.

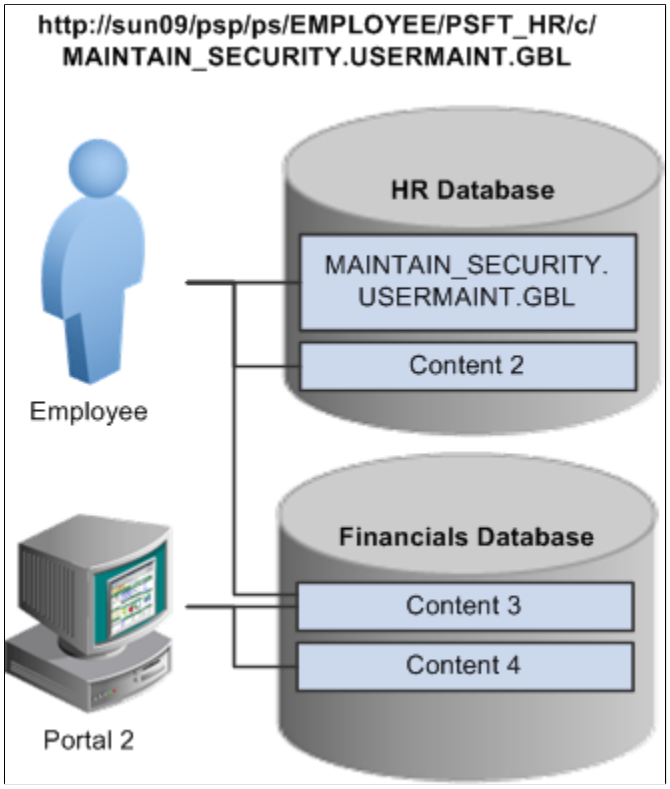
The portal servlet uses the node name in the URL to determine the location of the servlet for that node (stored as part of the node object). If the content is hosted by the local database, the portal servlet communicates with the content servlet directly (Java method calls), not through HTTP (using the portal content servlet URL).

The ID format is different for each content type. Components are identified by menu, component, and market; scripts are identified by the program name, and so on. The portal registry determines the content reference for this content, and for the template and the pagelets that appear around with it.

Image: URL structure

The following diagram shows how the URL lists the required information from left to right: portal (EMPLOYEE), node (PSFT_HR), content type (c), content ID (MAINTAIN_SECURITY.USERMAINT.GBL) for this example URL:

```
http://sun09/psp/ps/EMPLOYEE/PSFT_HR/c/MAINTAIN_SECURITY.USERMAINT.GBL
```



This is consistent with the logical organization of content in portals and databases. At the left side of the figure, portals point to nodes and, ultimately, to content within those nodes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Here's what a portal URL looks like:

http://server/servlet_name/sitename/portalname/nodename/content_type/content_id?content_parm

The following table describes the different sections of the URL:

Section	***Description***
http://server/	Scheme (HTTP / HTTPS) and web server name. **Important!** The server name and port in this section must not exceed 30 characters in length. For example, *http://mybiglongservername.peoplesoft.com:8080,* not including the *http://* prefix, is 39 characters—nine characters too long.
servlet name/	The name of the physical servlet that the web server invokes to handle the request.
sitename/	The site name specified during PeopleSoft Pure Internet Architecture setup. This enables you to set up multiple sites on one physical web server. The site name is ultimately mapped by the web server to the appropriate configuration.properties file. **Important!** The site name can include underscores (_), but an underscore cannot be followed by a numeric character or the string newwin (for example, *my_site_3* or *my_newwin_site)*.
portalname/	Name of the portal to use for this request. The portal definition contains metadata that describes how to present the content (template, pagelets, and so on).
nodename/	Name of the node that contains the content for this request.
content_type/	Type of the content for this request.
content_id	The identification of the content. The ID and type ensure that the correct content is retrieved.
?content parm	Query string parameters (name value pairs) for the content.

URL Format for PeopleSoft Pure Internet Architecture Content Types

This table lists the URL formats for each PeopleSoft Pure Internet Architecture content type:

Content Type	***URL Format***	***Example***
Component	/c/*menu.component.market*/?Page=*page*&Action=*action* &*key_id=key_value*. . .	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/c/ MAINTAIN_SECURITY.USERMAINT.GBL?page=view&view=narrow
Script	/s/*recordname*. *fieldname*. *event*. *function*/?*parm_id=parm_value*. . .	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/s/ WEBLIB_Portal.PORTAL_HEADER.FieldChange.Iscript_DoSomething

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Content Type	*URL Format*	*Example*
External	/e/?url=*URL*	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/e/?url=http%2f%3a%3awww.peoplesoft.com
Homepage	/h/?tab=*homepage_tab*	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/h/?tab=HR homepage tab **Note:** This homepage URL tells the portal servlet to display the specified tab of the current user's homepage.
Query	/q/*query*	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/q/my_query
Worklist	/w/*worklist*	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/w/my_worklist
Navigation	/n/*business_process_map*	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/n/my_BusProcMap
File	/f/*filename*	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/f/myfile.html **Note:** The file URL for file content tells the servlet to retrieve the named file from the database and return it to the browser.
Disconnect	/?disconnect=y	http://sun09/psc/ps/EMPLOYEE/PSFT_HR/s/WEBLIB_QETEST.ISCRIPT1.FieldFormula.IScript_WhoAmI/?disconnect=y&postDataBin=y This is an internal flag used by xmllink to process a request and to invalidate the HTTP session afterwards. In a normal browser, the HTTP session is maintained so that the subsequent requests are not creating new HTTP sessions.

URL Format for Unwrapped PeopleSoft Pure Internet Architecture Content

PeopleSoft Pure Internet Architecture content is accessible with no template wrapping by means of the content servlet. This enables portals to implement a proxied architecture and enables you to include PeopleSoft Pure Internet Architecture content in other portal products and websites.

URLs for unwrapped PeopleSoft Pure Internet Architecture content are similar to URLs for wrapped content. Unwrapped PeopleSoft Pure Internet Architecture content has not gone through the portal servlet template process. In the URL, the content servlet is specified rather than the portal servlet. The following table shows sample URLs for a component and an iScript.

Note: The content servlet ignores the portal and node name, but they are still necessary as placeholders. Omitting them causes a runtime error because the psc servlet checks that the URL contains a portal and node name, even though it doesn't use them.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Unwrapped Content Type	*URL Example*
Component	http://sun09/psc/ps/EMPLOYEE/PSFT_HR/c/E_PRO.CheckOut.GBL
Script	http://sun09/psc/ps/EMPLOYEE/PSFT_HRm/s/WEBLIB_Portal.PORTAL_HEADERFieldChange.Iscrip_DoSomething

Pagelet URLs

Pagelets are snippets of HTML content that appear in one section of a template. Unlike target content, they are referenced by name within the template. This sample HTML refers to a pagelet:

```
<Pagelet Name="UniversalNavigation">
<Source Node ="LOCAL_NODE" Pagelet="MyPagelet"/>
</Pagelet>
```

When the servlet resolves this tag, it generates the URL using:

- The name of the portal containing this template.
- The node name specified in the Source tag (for example, Node = “SomeNode”).
- Content type and content name specified in the pagelet definition for the specified pagelet.

The resolved URL for this example is (assume this template is in the Employee portal):

http://sun09/psc/ps/EMPLOYEE/PSFT_HR/s/WEBLIB_TEST.ISCRIPT1.FieldFormula.IScript_WhoAmI

System URLs

System URLs do not have content or query strings; instead, they contain system commands, such as Login or Expire. System URLs can be issued to both the content and portal servlets. This table lists system URLs:

URL	*Description*
http://sun09/psp/ps/?cmd=expire	Closes the current session and returns the expire page. **Note:** The system detects whether the user is still active during the cmd=expire process on the server. The system ignores the session after detecting that the user is still active so that ongoing transactions in the active window are not disturbed. However, if another session is open simultaneously that *is* inactive, the inactive session will be closed.
http://sun09/psp/ps/?cmd=logout	Closes the current session and returns to the sign-in page.
http://sun09/psp/ps/?cmd=login	Tells the servlet to return to the sign-in page.

Proxy Architecture and Relative URLs

A relative URL is written to an HTML document without some portion of the scheme, server, or path. When the browser downloads the document containing the relative URL, it makes the relative URL absolute by adding the scheme, server, and path of the downloaded document. Relative URLs simplify moving static documents around on web servers because you don’t have to change the URLs embedded within the documents that you move.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

One portal servlet can proxy content from several other content servlets. The portal servlet acts as an intermediary in the conversation between the browser and the various content services, relaying HTTP requests and responses from the content servlet to the browser.

The portal servlet acts as a reverse proxy server by ensuring that all URL references on portal pages point back to the portal servlet itself. The portal servlet does this by rewriting all content retrieved through the portal to contain relative URLs in appropriate URL formats.

To increase performance, you can include the custom header UsesPortalRelativeURL with the value *True* to indicate that the URL is already set in the correct format. All content from databases using PeopleTools 8.42 and later generates URLs with the correct format and uses this command in the header.

Examples of Relative PeopleSoft URLs

For example, assume that the MAINTAIN_SECURITY.USERMAINT_SELF.GBL component is in the PSFT_HR node, and it is being accessed by the EMPLOYEE portal.

If you wanted a navigation iScript within the PSFT_HR node to construct a link to the MAINTAIN_SECURITY.USERMAINT_SELF.GBL component, you would add the following HTML to the response:

```
<a href="../../../EMPLOYEE/PSFT_HR/c/MAINTAIN_SECURITY.USERMAINT_SELF.GBL" . . . >
```

When this HTML is downloaded to the browser, the absolute URL includes the scheme, server, and servlet directory of the proxying portal servlet, even though the iScript may have run on a content servlet on a different web server. The absolute URL continues with the portal, node, service type, and component name, as specified by the iScript.

Here is what the final URL looks like:

http://sun0server/psp/ps/EMPLOYEE/PSFT_HR/c/MAINTAIN_SECURITY.USERMAINT.GBL

Note: The content services always specify the portal, node, and content type (with the "../../..) even if those values are the same as the current page.

Now assume that you want the navigation iScript to create a link to the MAINTAIN_SECURITY.USERMAINT_SELF.GBL component in the HRMS node. Also assume that the component is being accessed by the employee portal. The navigation iScript adds the following HTML to the response:

```
<a href="../../../EMPLOYEE/HRMS/c/MAINTAIN_SECURITY.USERMAINT_SELF.GBL" . . .>
```

The absolute URL looks like this:

http://sunserver/psp/ps/EMPLOYEE/HRMS/c/MAINTAIN_SECURITY.USERMAINT.GBL

The URL correctly points to the appropriate content without any HTML parsing or URL rewriting.

Finally, assume that you want a navigation iScript running within the EMPLOYEE portal to construct a link to the MAINTAIN_SECURITY.USERMAINT_SELF.GBL component within the e_benefits portal. To construct this link, the iScript generates the following HTML:

```
<a href="../../../E_BENEFITS/HRMS/c/MAINTAIN_SECURITY.USERMAINT_SELF.GBL" . . . >
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: The HREF tag with a relative URL can be used to change a portal or node only if the HTML is being accessed through an HTML template. It won't work with a frame template because the base URI of the frame points to the content servlet, which ignores the portal and node names. Use the PeopleCode transfer function to specify a target portal and node.

Related Links

Portal Technology Overview

Page-Based Template Proxying

This section discusses:

- Proxying and URL conversion.
- Anchor tag conversions.
- Form tag conversions.
- JavaScript conversions.

The following discussion applies only to content that is not in the simple URL format.

See Portal URL Formats.

Proxying and URL Conversion

When processing page-based templates, the portal servlet uses a process called *proxying* to help ensure that users always stay within the context of the portal and to ensure that familiar portal features, such as the universal navigation header, do not disappear when a user clicks a link.

When users sign in to a PeopleSoft portal, they sign in to a web server on which the portal servlet is running. The portal servlet processes all the HTML that isn't in the simple URL format, converting all URL references to point to the portal web server rather than the original URL. The original URL is still necessary to retrieve the requested content; it is stored in the new URL in the URL query string parameter. The portal servlet proxies all links and all form actions in this manner. Additionally, it converts any relative URLs into absolute URLs.

As an example, imagine that a user requests a page from an external website through a proxied link in the portal. The request arrives at the portal web server, invoking the portal servlet. The portal servlet then programmatically retrieves the page from the web server associated with the requested page. It proxies all the links on the retrieved response and sends the page (the contents of the HTTP response) back to the browser, formatted as the user would expect within the portal.

Converting Relative URLs to Absolute URLs

The use of relative URLs is common in web page design for external content. They are often used when a web page includes links to content on the same web server that the page is on. This works fine when a browser communicates directly with a web server because no ambiguity exists about where the relative URL points. However, because the portal servlet—and the proxying process—is placed between the browser and the target page, relative URLs become relative to the portal web server instead of the original target server. To prevent this from occurring and causing broken links, part of the proxying process includes the conversion of all relative URLs to absolute URLs. The following examples show the original, relative version of an HTML tag and the rewritten absolute version created by the portal servlet.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The following example shows a relative tag:

```
<IMG src="/image/cache/image.gif" lowsrc="/image/cache/image2.gif">
```

The following example shows an absolute tag:

```
<IMG src="http://originalserver/image/cache/image.gif"
 lowsrc="http://originalserver/image/cache/image2.gif">
```

Anchor Tag Conversions

The portal servlet rewrites all anchor tags so that their SRC attributes direct the browser's request to the portal web server instead of the server that created the URL. After rewriting the anchor tag, the portal servlet can determine whether the target URL should be wrapped with a template.

Old Anchor Tag

The following example shows an old anchor tag:

```
<a src=http://server/targetpage.html?Action=New>
```

New Anchor Tag

The following example shows a new anchor tag:

```
<a src=http://portalserver/psp/ps/EMPLOYEE/NODE/e/?url=http%3a%2f%2fserver%2ftarget⇒
page.html%3fAction%3dNew>
```

Form Tag Conversions

Similar to anchor tags, all Form tags must have their Action attributes rewritten. However, with Form tags, the original URL is captured in a hidden form field instead of a query string parameter.

Old Form Tag

The following example shows an old Form tag:

```
<form action=http://server/targetpage.html>
```

New Form Tag

The following example shows a new Form tag:

```
<form action=http://portalserver/psp/ps/EMPLOYEE/NODE/e>
 <input type=hidden name=URL>
  http://server/targetpage.html
 </input>
```

JavaScript Conversions

The portal servlet ensures that URL references contained in JavaScript are rewritten to point to the portal servlet instead of their original reference.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Part II

Working with PeopleTools Portals

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 4

Working with Navigation Pages and Navigation Collections

Understanding Navigation Pages

Navigation pages provide an alternative to menu navigation. Three types of navigation pages are available:

- Standard navigation pages.

  You access a standard navigation page by clicking the icon for a folder in the drop-down menu. Standard navigation pages are generated automatically based on the folder and content reference sequence numbers defined in the portal registry.

- Custom navigation pages.

  You access a custom navigation page by clicking a link (a content reference) in the drop-down menu. A custom navigation pages is generated by publishing a navigation collection. Some custom navigation pages are delivered with PeopleSoft applications to facilitate commonly used tasks and business processes. Authorized users can edit delivered navigation collections, as well as create new ones tailored to suit the business process needs of your organization. These navigation collections and generated navigation pages can be configured to incorporate the use of icons to further increase the ability to intuitively navigate tasks.

  See Creating and Maintaining Navigation Collections

- SmartNavigation pages.

  You access a SmartNavigation page by clicking a link (a content reference) in the drop-down menu. SmartNavigation pages are generated dynamically based on hierarchical data and appear not as folders and links, but as an organization chart, which users can traverse to access transactions. The pages and processes that you use to configure SmartNavigation pages are not the same as those that you use to configure standard and custom navigation pages.

  See Administering SmartNavigation Folders.

The following sections further describe standard navigation pages, custom navigation pages, and navigation page display caching.

Standard Navigation Pages

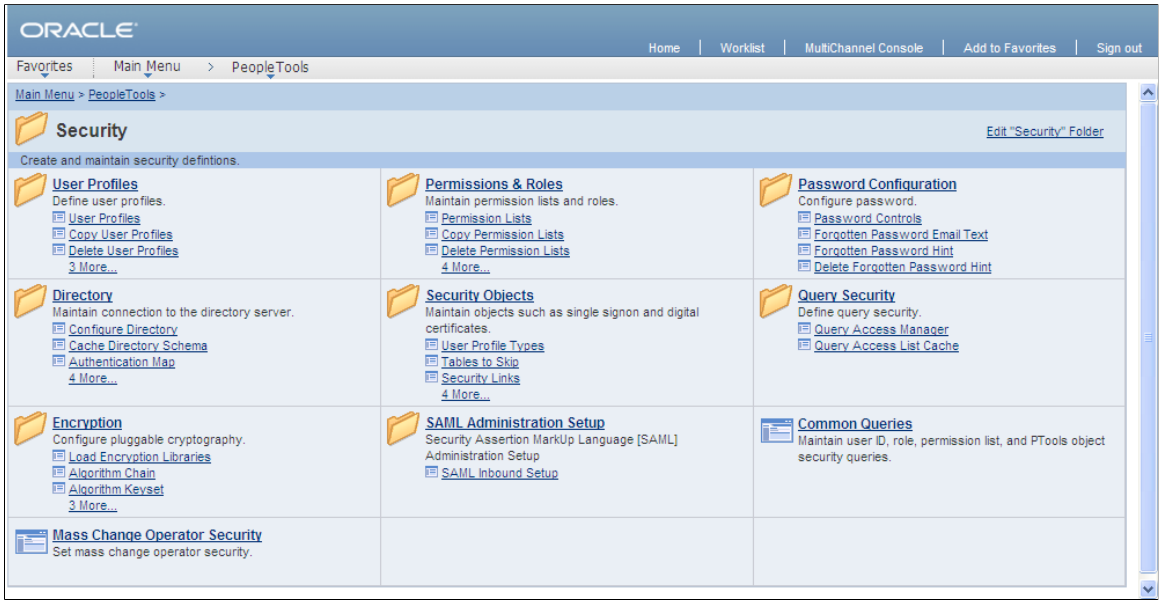
You enable the display of standard navigation pages at the registry level on the General Settings page. You can then disable the display of standard navigation pages for specific folders in the registry on the Folder Administration page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

You access standard navigation pages by clicking a folder icon in the menu navigation. Items contained within the folder, including content references and other folders, are arranged in columns on the page.

Image: Standard navigation page

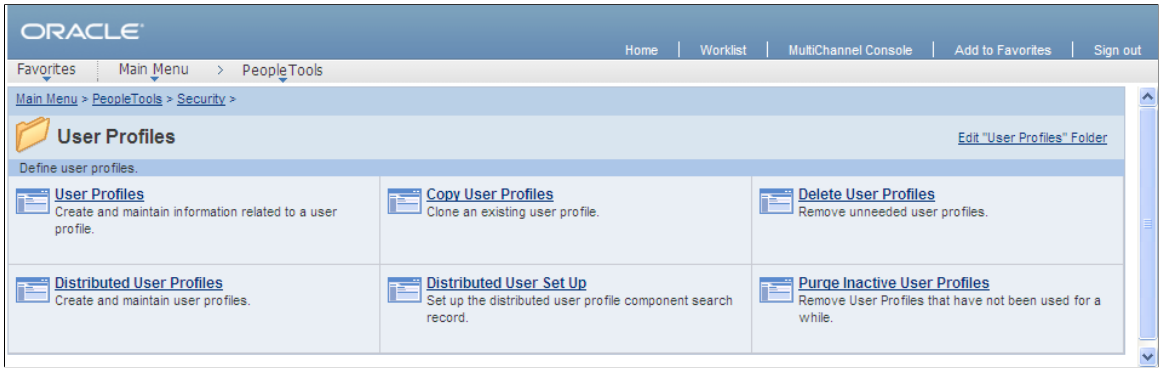
The following example illustrates a standard navigation page for the PeopleTools, Security folder.



From a navigation page, you can click any content reference link to directly access the component without drilling down through the folder hierarchy. When you click a folder link on a standard navigation page, you access another standard navigation page that reflects the contents of the selected subfolder.

Image: Standard navigation page for the User Profiles folder

For example, when you click the User Profiles link from the Security standard navigation page, the standard navigation page for the User Profiles folders appears as shown here:



Standard Navigation Page Display

To enable the display of standard navigation pages:

1. Select PeopleTools, Portal, General Settings.
2. Select the Is Folder Navigation Enabled check box.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

3. Enter *PT_PTPP_SCFNAV_BASEPAGE_SCR* (the delivered Base Navigation Page object) in the Folder Navigation Object Name field.

   This Base Navigation Page object is designed to display standard navigation pages as they are documented in these topics.

Note: To enable the display of standard navigation pages for the registry in a portal other than the one you are signed in to, select PeopleTools, Portal, Portal Definitions then click Edit next to the portal that you want to change to access the Portal Definitions - Edit Portal page.

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

Certain aspects of standard and custom navigation page display are dependent on system and registry options. See Setting Up Navigation Collection Options.

See Administering Folders, Administering Portal Definitions.

Folder and Link Sequence on Standard Navigation Pages

The sequence in which folders and links appear on standard navigation pages is defined in the portal registry. Although the contents of a standard navigation page correspond to the content of the selected menu folder, the sequence of elements on the page may not be the same as the sequence in which they are displayed in the menu.

In the drop-down menu, folders are always displayed before content references. The folders appear as a group in the order determined by the sequence numbers defined in the portal registry; similarly, the content references appear as a group in the order determined by the sequence numbers defined in the portal registry.

Unlike the drop-down menu, standard navigation pages do not group folders first followed by content references. Sequence numbers alone determine the order of items on a standard navigation page. However, if a folder and content reference have the same sequence number, the folder is displayed before the content reference on a standard navigation page.

See Administering Folders, Administering Content References.

Custom Navigation Pages

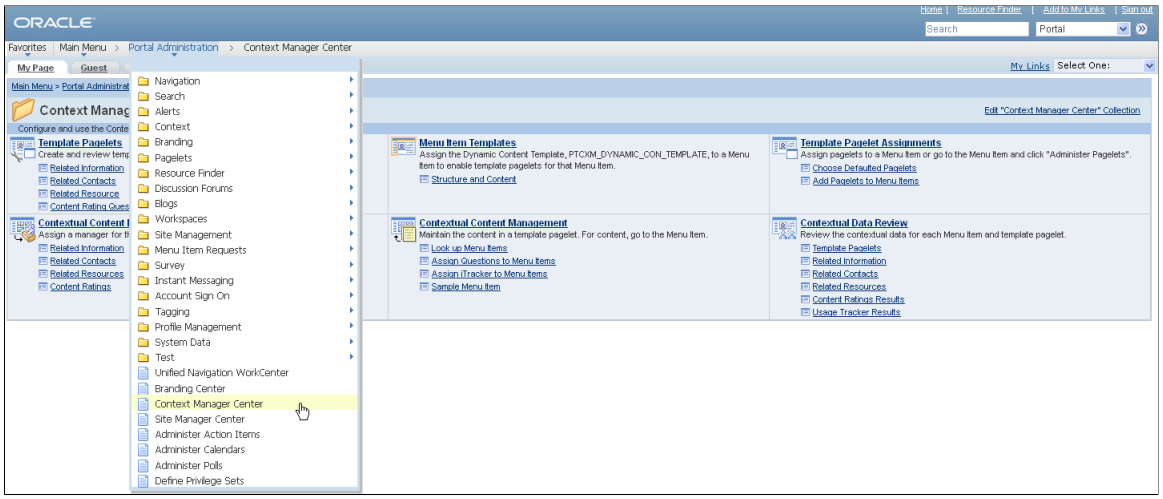
While standard navigation pages automatically display only the contents of a selected menu folder, custom navigation pages can be defined to display any content references and folders available in the menu navigation, regardless of their location.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Unlike standard navigation pages, you access custom navigation pages using links in the menu navigation instead of folder icons.

Image: Context Manager Center custom navigation page from PeopleSoft Interaction Hub

For example, in PeopleSoft Interaction Hub, selecting the Context Manager Center link in the menu navigation displays the Context Manager Center custom navigation page as shown here:



Similar to standard navigation pages, 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 component to which the content reference points.

See Creating or Editing a Navigation Collection.

Folder and Link Sequence on Custom Navigation Pages

The sequence in which folders and links appear on custom navigation pages is defined in the navigation collection. Once the folder and link sequences for a navigation collection are registered, their sequence numbers are propagated to the portal registry. However, any modifications to navigation collections (and the generated custom navigation pages) *must* be made in the navigation collection definition. Do not modify custom navigation pages in the portal registry.

See Editing a Folder, Editing a Link.

Navigation Page Display Caching

You can enable the caching of displayed navigation pages. Enabling caching improves performance because the next time the same navigation page is accessed from any computer via the same web server with the same cache designation (role, user, or public), the navigation page is retrieved from the web server cache. To clear the web server cache memory, stop and restart the web server or wait until the cache time elapses.

Methods for enabling caching depend on the method that you use to access navigation pages:

- To enable caching for a custom navigation page accessed by clicking a content reference link, select the Enable Caching option on the Publish Collection page when you publish the custom navigation page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

On 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.

- To enable caching for all folders or navigation paths, select the Enable Drill Down Cache option on the System Options page.

  Enter a value in the Cache Enabled Value field (*role,max-age=30,* for example).

- To enable caching for a standard navigation page that you access by means of menu navigation, edit the content reference for the Base Navigation Page object (PT_PTPP_SCFNAV_BASEPAGE_SCR) in the portal registry.

  The Base Navigation Page content reference is located in the Tools - Hidden folder. In the Content Reference Attributes group box, enter the *PSCACHECONTROL* attribute name and an attribute value, such as *role,max-age=30.*

See Setting Up Navigation Collection Options, Publishing Navigation Collections, Administering Content References.

Understanding Navigation Collections

This section discusses:

- Navigation collections
- Navigation collection images

Navigation Collections

Navigation collections provide 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.

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 references needed to construct a URL at runtime to redirect output to the original content reference or folder reference. These references include the portal link, folder reference link, or content reference link.

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

After being defined, a navigation collection can be published as a custom navigation page or as a navigation pagelet for use on a homepage, dashboard page, or WorkCenter page.

When you publish navigation collections, 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 choose one of these two options for setting security access to published navigation collections:

- A security synchronization process that will apply content reference and folder security as it is defined in the portal registry to the content reference definition for published navigation pages or pagelets.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- A manual process that entails you configuring the security access to published navigation pages and pagelets.

Note: Public access is allowed by default.

Navigation collections enable 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 that their user communities need to access on a frequent basis. They should also know where to find links to this content, and whether the content is accessed by portal navigation or external URLs.

Navigation collections can contain these elements:

- Links to existing, nonhidden content references in the portal registry.
- Links to existing, nonhidden folders in the portal registry.

  The navigation collection will contain all contents of the menu folder and its subfolders.
- Links to remote, component-based content references.

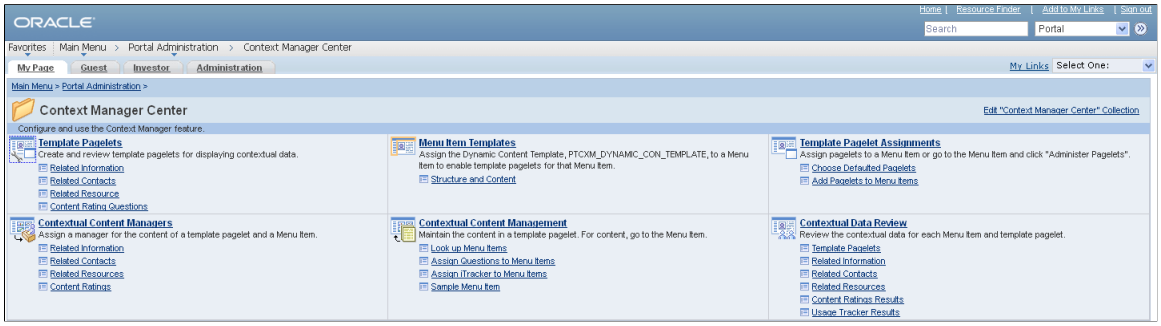
  Note: Remote links are supported only on PeopleSoft Interaction Hub systems with unified navigation configured to include the remote content provider system.

- Custom (user-defined) folders that can be used to group combinations of any of these elements, including other user-defined folders.

Navigation Collection Published as a Navigation Page

Image: Navigation collection published as a custom navigation page

The following example shows a navigation collection that is published as a custom navigation page:



Top-level folders in the navigation collection appear as bold and linked section headings that organize the folders and content references they contain, as shown in these examples. However, the appearance of your published navigation collections depends on the assigned style sheet. The assigned style sheet must use style classes contained in the PTPP_SCSTYLEDEF style class definition.

Click any folder link to access another navigation page that contains any folders and content references nested in the parent folder.

Note: Click the Edit <collection name> Collection link in the upper right corner of the navigation page to access the navigation collection definition on the Maintain Collection page. This link appears only for users who can access the navigation collection definition component.

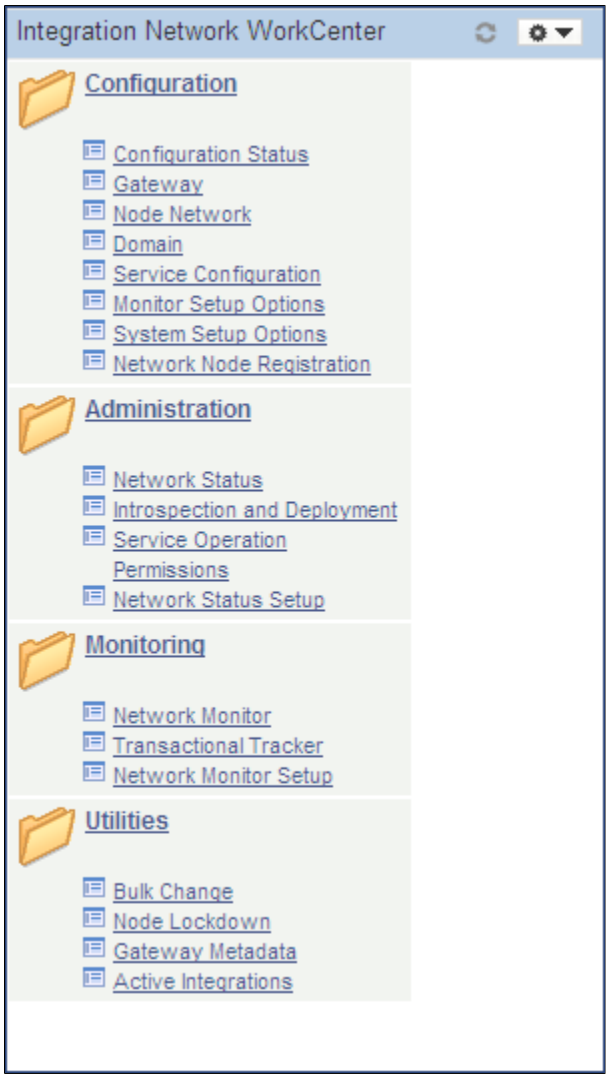
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Creating and Maintaining Navigation Collections.

Navigation Collection Published as a Navigation Pagelet

Image: Navigation collection published as a navigation pagelet

The following example shows a navigation collection that is published as a navigation pagelet:



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

Related Links

Publishing Navigation Collections

Understanding Portal Utility Processes

Navigation Collection Images

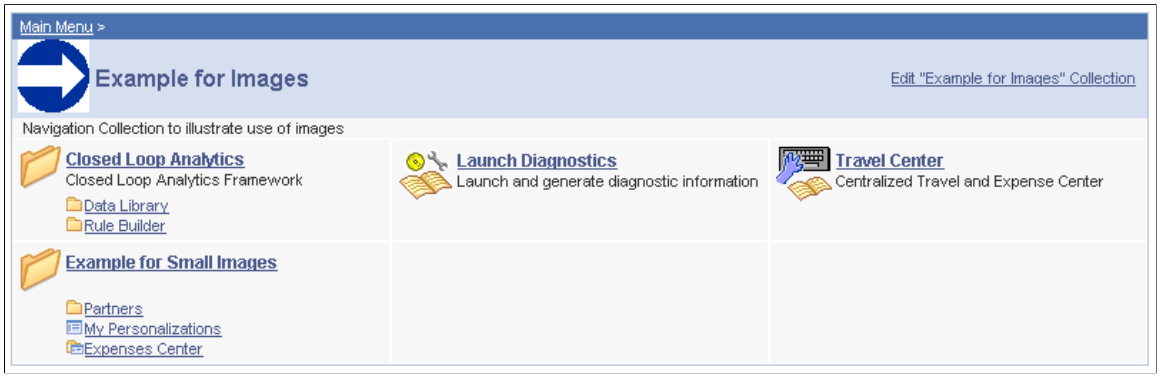
The images that you define for your navigation collections are published and displayed on navigation collection pages. You can associate images with navigation collections themselves, as well as with

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

navigation collection folders and content references. You can define these images at the system, registry, source, and individual navigation collection levels.

Image: Example of navigation collection images

The following navigation collection illustrates the different ways in which images can be used:



In the sample navigation collection, suppose that the following images have been defined for the different image types:

This image was defined as the collection image for this specific navigation collection.

This image was 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 appear for folders at the top level of a navigation collection publication.

This image was defined as the content image in the navigation collection definition. Large content images appear 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 appear for navigation collections at the top level of a navigation collection publication.

This image was defined as the small folder image. Small folder images appear for folders nested within top-level folders on a navigation collection publication.

This image was defined as the small content image. Small content images appear for content references nested within top-level folders of a navigation collection publication.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

This image was defined as the small collection image. Small collection images appear 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 a small image 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 appear until the element is accessed as a top-level 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 appears with its unique large image.

Common Elements Used on Navigation Pages

Breadcrumbs

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

See Setting Up Navigation Collection Options.

Edit <folder name> Folder

The Edit <folder name> Folder link appears 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 Defining Folders.

Edit <Navigation Collection name> Collection

The Edit <Navigation Collection name> Collection link appears 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 Creating and Maintaining Navigation Collections.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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

See "Using Spell Check" (PeopleTools 8.54: Applications User's Guide).

Setting Up Navigation Collection Options

This section provides an overview of navigation collection options, lists common elements used in this section, and discusses how to:

- Define Portal Utilities system options.
- Define Portal Utilities registry options.

Note: These options are used by navigation collections, Feeds, and Pagelet Wizard pagelets.

Understanding Navigation Collection Options

You define and override navigation collection options at different levels. 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.

PeopleTools delivers a set of system-level defaults. You can change these values on the System Options page or override these values at successive levels in the default hierarchy. This table lists the navigation collection options and the levels at which they are set up or overriden:

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	None available	None available	None available
Show breadcrumbs	X	X	None available	None available
Show main menu breadcrumb	X	X	None available	None available
Registry node name	X	None available	None available	None available
Large (parent) image	X	X	X (with attribute name PTPP_IMAGE)	X
Small (child) image	X	X	None available	None available
Style sheet	X	X	None available	X
Show images	X	X	None available	X
Maximum child links	X	X	None available	X

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Options	*System (default)*	*Registry (overrides system default)*	*Source Reference (overrides system and registry values)*	*Collection Reference (overrides system, registry, and source reference values)*
Maximum columns	X	X	None available	None available
Owner ID	X	None available	None available	X

You define and set navigation collection options on these pages:

- Set system options on the System Options page.
- Set registry options on the Registry Options page.
- Set source reference options through the Structure and Content pages.
- Set navigation collection reference options on the Maintain Collections page.

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.

Related Links

Defining Portal Utilities System Options
Defining Portal Utilities Registry Options
Creating and Maintaining Navigation Collections
Understanding Portal Administration
Managing General Portal Settings

Common Elements Used in This Section

Show Breadcrumbs

Select to indicate that you want navigation path to appear on published navigation pages.

These breadcrumbs reflect the path taken by users as they navigate through left-navigation menu folders and navigation page folders. Displaying breadcrumbs facilitates user navigation because the left-navigation menu is not updated to reflect the user's navigation through navigation pages.

The first breadcrumb in the Main Menu collection points to the delivered PT_PTPP_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 displayed depend on the left-navigation menu folders that the user opened to access the navigation collection page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Subsequent breadcrumbs displayed depend on the folder path that the user accesses on the navigation collection page.

Show Main Menu Breadcrumb

If you 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 deselect this option, the first entry in the breadcrumb path depends on the level-1 left-navigation folder that the user opens to access the navigation collection page. The Main Menu link does not appear as the first link in the breadcrumbs.

Main Menu Collection Name

This field appears when the Show Main Menu Breadcrumb option is selected. The default value is *Main Menu,* which designates that the first breadcrumb displayed accesses the Main Menu page. You can choose to replace the first breadcrumb Main Menu link with a link that accesses an alternate navigation collection that 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) that you want to use to generate navigation collection pages. Cascading style sheets are defined in PeopleSoft Application Designer.

This setting applies to both navigation pages and navigation pagelets.

Note: You must use the same styleclass names that are found on the stylesheet PTPP_STYLEDEFN.

See "Understanding Style Sheets and Style Classes" (PeopleTools 8.54: Application Designer Developer's Guide)."Creating Standard and Sub Style Sheets" (PeopleTools 8.54: Application Designer Developer's Guide)

Maximum Columns

Enter the maximum number of columns to display on a navigation page.

Navigation 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 only to navigation pages.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See "Personalizing Your Homepage" (PeopleTools 8.54: Applications User's Guide).

Maximum Child Links

Enter the maximum number of links to display 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 jumps to a page from which you can access all links stored in the navigation collection folder.

For example, if you set Maximum Child Links to *4* and the navigation collection folder contains five links, three links appear 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 appears.

This setting applies to both navigation pages and navigation pagelets.

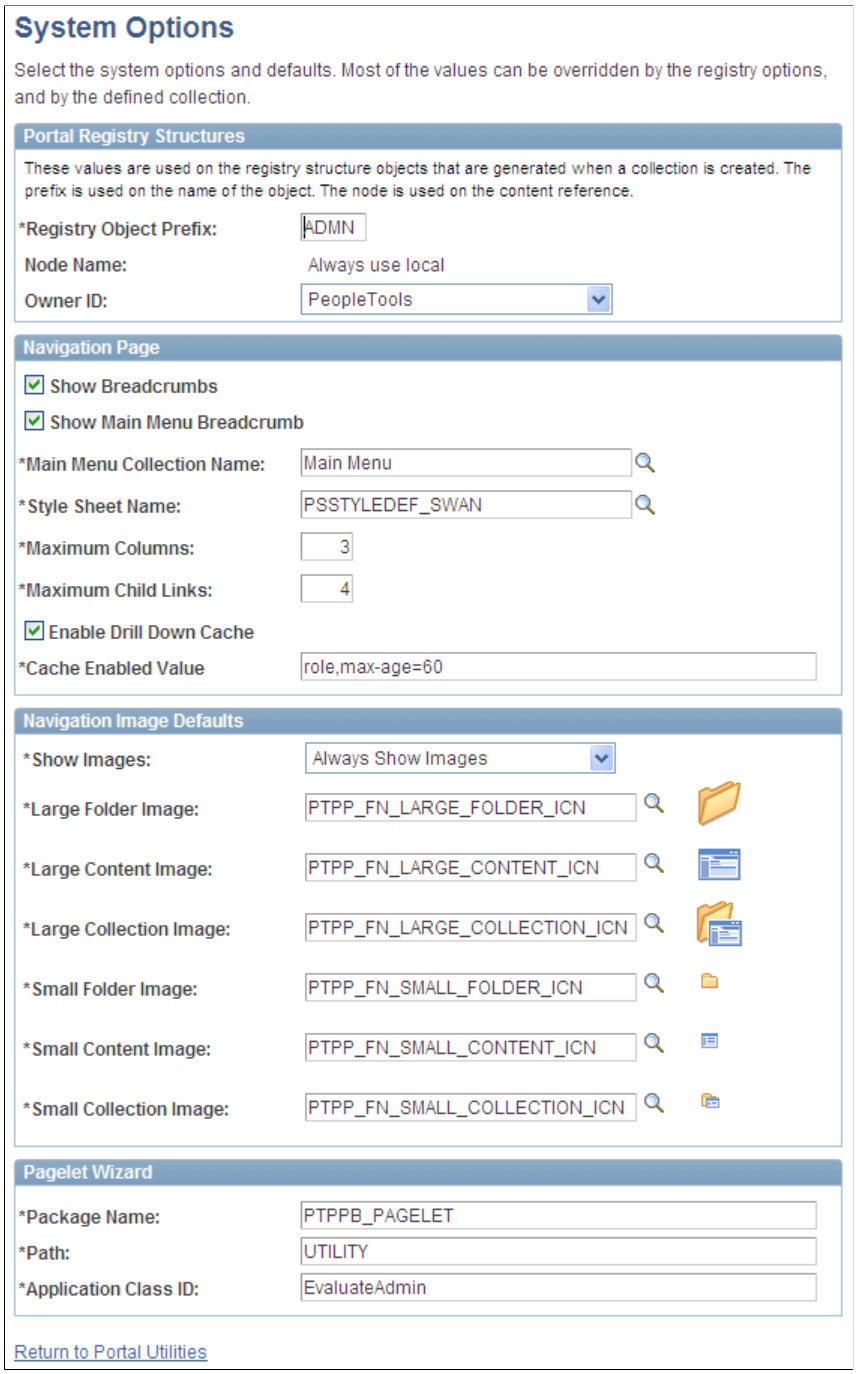
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Portal Utilities System Options

Access the System Options page. (Select PeopleTools, Portal, Portal Utilities, System Options.)

Image: System Options page

The following screenshot illustrates how to manage values in the System Options page. Most of the values can be overridden by the registry options and by the defined collection.



PeopleTools delivers a set of system-level defaults. You can change the default values on this page or override the defaults at successive levels in the default hierarchy. However, you cannot delete the default values because system-level default values are required.

See Understanding Navigation Collection Options.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Portal Registry Structures

Registry Object Prefix

Enter the value that the system prepends to registry structure object names (content and folder references) that are created when you publish navigation collections, template pagelets, and feed definitions.

For example, the object name of a published navigation pagelet has this format: `<prefix>_SC_PGT_<object label>`. The object name of a published navigation page has this format: `<prefix>_SC_SP_<object 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 listed default node names:

PeopleSoft Enterprise Financials/SCM: *ERP*

PeopleSoft Enterprise CRM: *CRM*

PeopleSoft Enterprise HCM: *HRMS*

PeopleSoft Interaction Hub: *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 of the portal registry tables.

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

Enable Drill Down Cache

Select to enable caching for navigation pages that you access by selecting a folder on a navigation page. You must also enter a value in the Cache Enabled Value field on this page.

See Navigation Page Display Caching.

Cache Enabled Value

Enter the value 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

Select one of these values:

Always Show Images: Select if you always want images to appear on navigation pages and pagelets.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Never Show Images: Select if you never want images to appear on navigation pages and pagelets.

Only Show Defined Images: Select if you want only specifically defined images to appear on navigation pages and pagelets. If you select this option, only images defined as part of the navigation collection or defined on the source reference are displayed on the navigation collection's publications. System-level 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 page or pagelet.

Large Content Image Select the default image that you want to display for top-level content references on a navigation page or pagelet.

Large Collection Image Select the default image that you want to display for top-level navigation collection references on a navigation page or pagelet.

Small Folder Image Select the default image that you want to display for a folder within a folder on a Navigation page or pagelet.

Small Content Image Select the default image that you want to display for a content reference within a folder on a Navigation page or pagelet.

Small Collection Image Select the default image that you want to display for a navigation collection within a folder on a Navigation page or pagelet.

Pagelet Wizard

Package Name Specify the application package name to use for the display format.

Path Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method.

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.

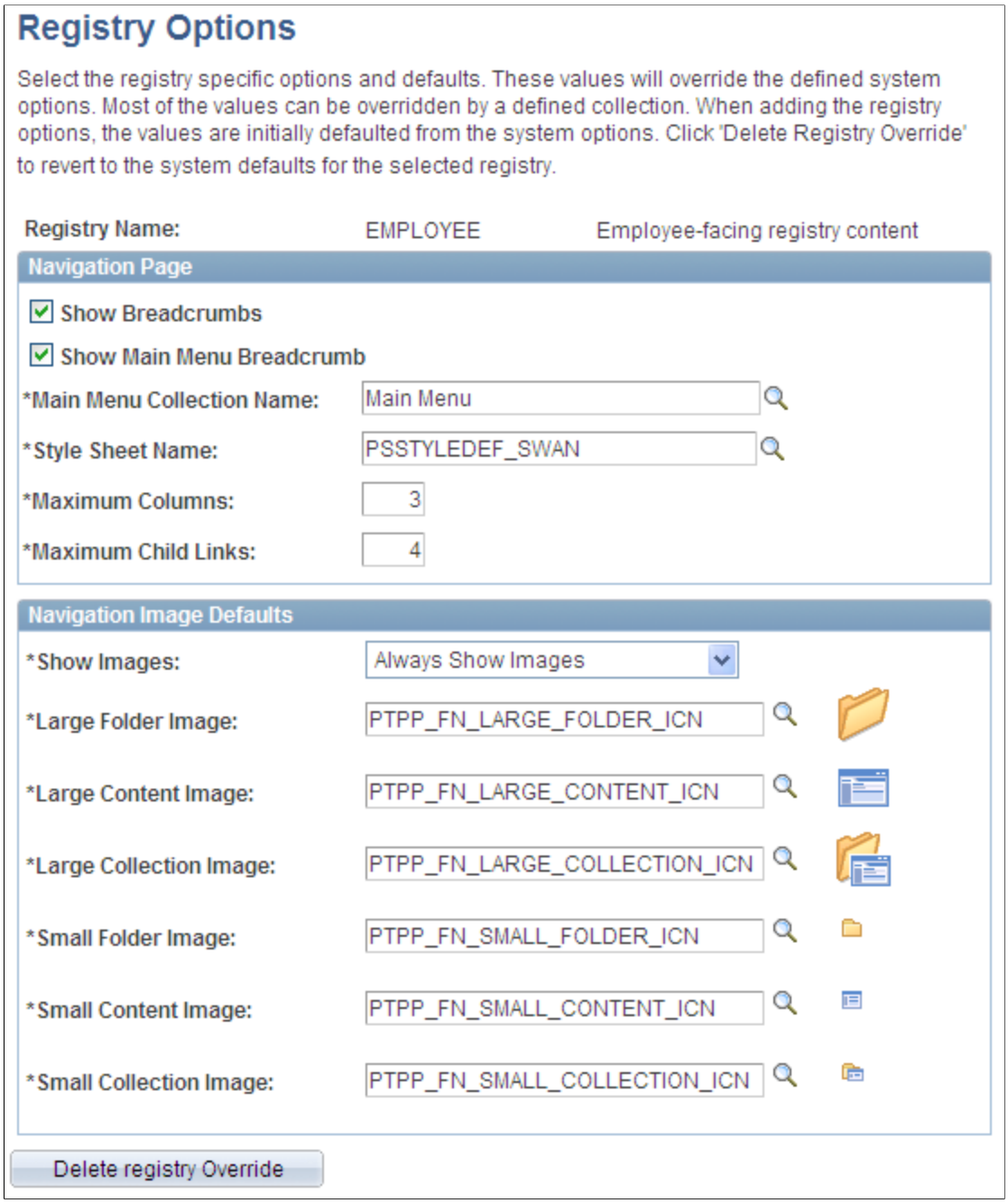
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Portal Utilities Registry Options

Access the Registry Options page. (Select PeopleTools, Portal, Portal Utilities, Registry Options.)

Image: Registry Options page

The following screenshot illustrates how to manage values in the Registry Options page. Most of the values in this page can be overridden by a defined collection.



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

See Understanding Navigation Collection Options.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Registry Name

Select the portal registry for which you want to define navigation collection options that will override system-level defaults.

Navigation Image Defaults

Show Images

Always Show Images: Select if you always want images to appear on navigation pages and pagelets accessed by means of the selected portal registry.

Never Show Images: Select if you never want images to appear on navigation pages and pagelets accessed by means of the selected portal registry.

Only Show Defined Images: Select if you want only specifically defined images to appear on navigation pages and pagelets accessed by means of 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 pages and pagelets. System-level 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 page or pagelet accessed by means of the selected portal registry.

Large Content Image

Select the image that you want to display for top-level content references on a Navigation page or pagelet accessed by means of the selected portal registry.

Large Collection Image

Select the image that you want to display for top-level navigation collection references on a Navigation page or pagelet accessed by means of the selected portal registry.

Small Folder Image

Select the image that you want to display for a folder within a folder on a Navigation page or pagelet accessed by means of 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 page or pagelet accessed by means of 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 page or pagelet accessed by means of 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating and Maintaining Navigation Collections

This section discusses how to:

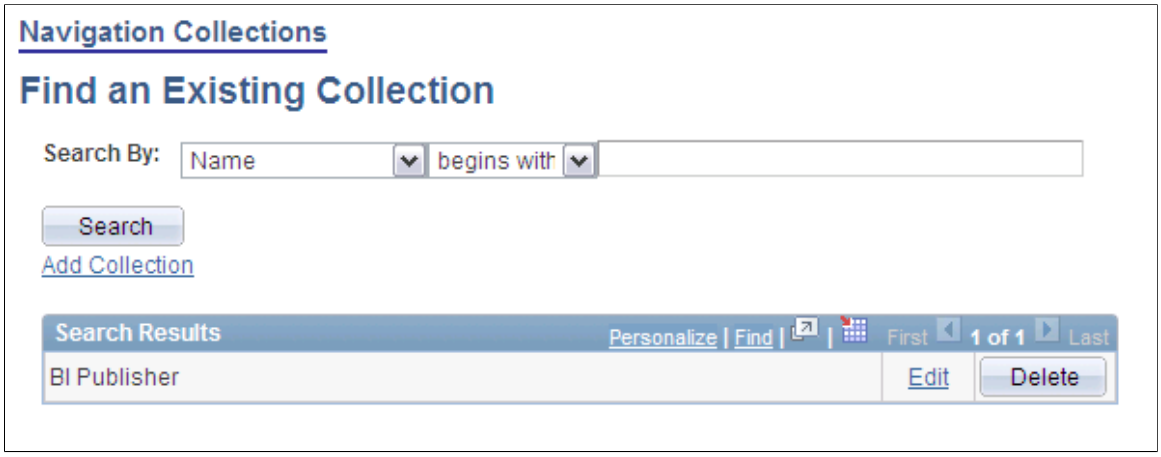
- Search for a navigation collection.
- Create or edit a navigation collection.
- Add a folder.
- Select a source folder.
- Select navigation collection images.
- Edit a folder.
- Add a link.
- Select a local link.
- Select a remote link.
- Edit a link.

Finding Existing Navigation Collections

Access the Find an Existing Collection page. (Select PeopleTools, Portal, Portal Utilities, Navigation Collections.)

Image: Find an Existing Collection page

This example illustrates the fields and controls on the Find an Existing Collection page. You can find definitions for the fields and controls later on this page.



Search by

Select one of these values:

Description: Select to conduct your search based on navigation collection description text.

Name: Select to conduct your search based on navigation collection name text.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 appear for the Main Menu navigation collection. While you can edit this navigation collection, you should not delete it because 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.

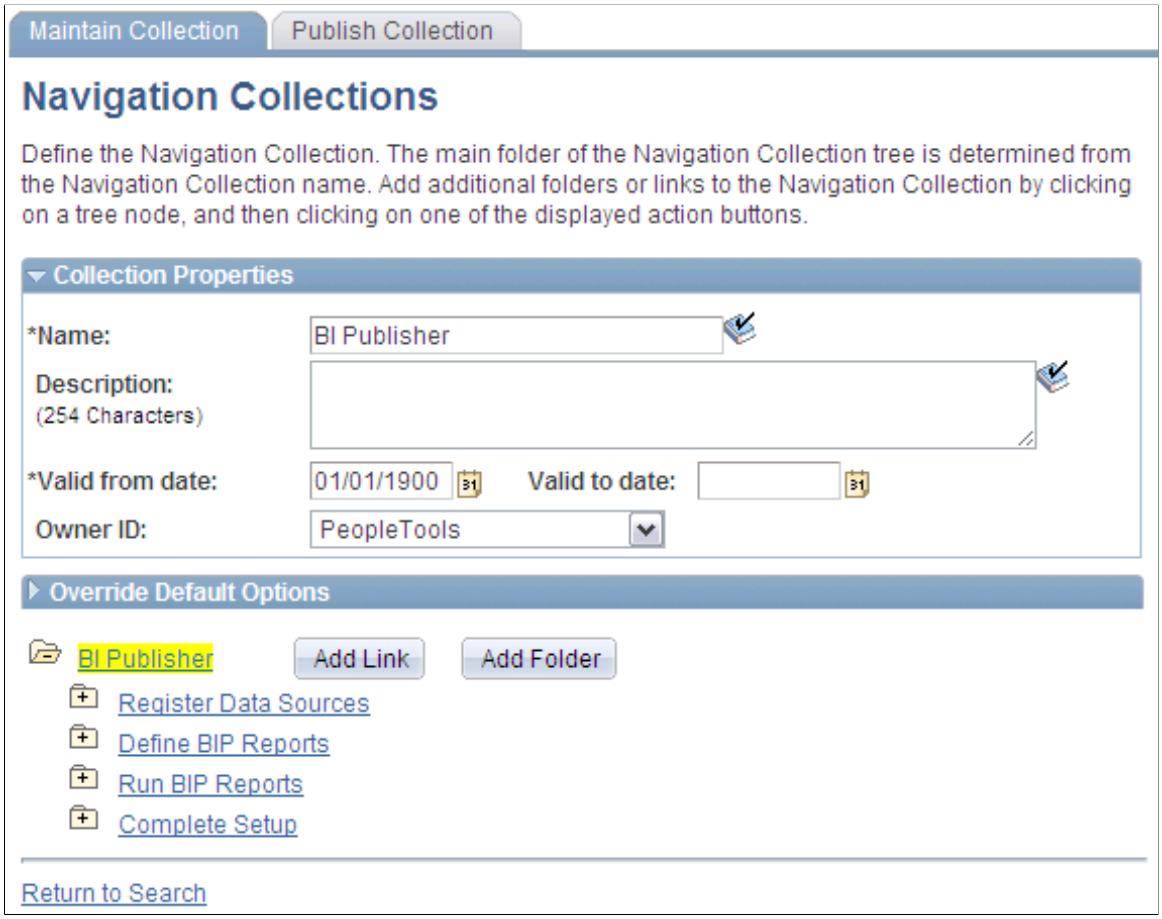
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating or Editing a Navigation Collection

Access the Navigation Collections page. (Click the Edit link for an existing navigation collection on the Find an Existing Collection page).

Image: Navigation Collections page

This example illustrates the fields and controls on the Navigation Collections page. You can find definitions for the fields and controls later on this 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 the navigation collection. This text appears as the label on the published navigation page or pagelet.

Description

Enter a description of the navigation collection. This text can appear in several locations.

If the navigation collection is published as a navigation page, the description text appears when the mouse cursor is moved

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

over the menu item link that is used to access the page. The description text also appears on the navigation page.

If the navigation collection is published as a navigation pagelet, the description text appears on the navigation pagelet.

Valid from date/Valid to date

Enter a range of dates during which this navigation collection is available. This information is copied to the portal registry when you register the navigation collection using the options on the Publish Collection page.

Oracle recommends 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 identify navigation collections created by your organization when you are 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 appears, but can be overridden.

See Setting Up Navigation Collection Options.

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 that you are creating. Leave fields clear if you don't want to override any default values.

See Setting Up Navigation Collection Options.

Style Sheet Name

Select the style sheet for this navigation collection.

Maximum Child Links

Enter the maximum number of child links to display.

Note: If more links are available for this collection, a More link appears.

Show Images

Select one of these values:

Always Show Images: Select to always display images on pages and pagelets for this navigation collection.

Never Show Images: Select if you don't want images to appear on pages and pagelets for this navigation collection.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Only Show Defined Images: Select to display only specifically defined images on pages and pagelets 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 pages and pagelets for this navigation collection. System-level and registry-level default images are not displayed.

Note: If you select the *Only Show Defined Images* option, small images do not appear on your navigation collection pages or pagelets because small images can be defined only as system-level and registry-level defaults.

Collection Image

Select the large image that you want to display as the top-level image for this navigation collection.

The navigation collection that you define is displayed as a tree hierarchy of nodes at the bottom of the page.

The Name value that you entered for the navigation collection appears 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 that 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 appears 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 that 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 appears 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 appears when you select a content reference link in the navigation collection hierarchy.

Add Folder

Click to access the Add Folder page. This option appears 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 that you create appears directly off the root node. Using

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 appears 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 appears when you select any folder other than the root note folder in the navigation collection hierarchy. This deletes the folder only from the navigation collection.

Designates a folder for which content is user-defined. These folders are added on the Add Folder page by means of the *User Defined Folder* folder type.

Click to expand the node and access options that enable you to edit or delete the folder.

Alternatively, select the associated folder name link to display options for editing or deleting the folder, but not expand folder contents.

Designates a folder for which content is menu-based. These folders are added on the Add Folder page by means of 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 for editing 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Adding a Folder

Access the Add Folder page. (Click the Add Folder button on the Maintain Collection page.)

Image: Add Folder page (user-defined folder)

This example illustrates the fields and controls on the Add Folder page (user-defined folder). You can find definitions for the fields and controls later on this page.

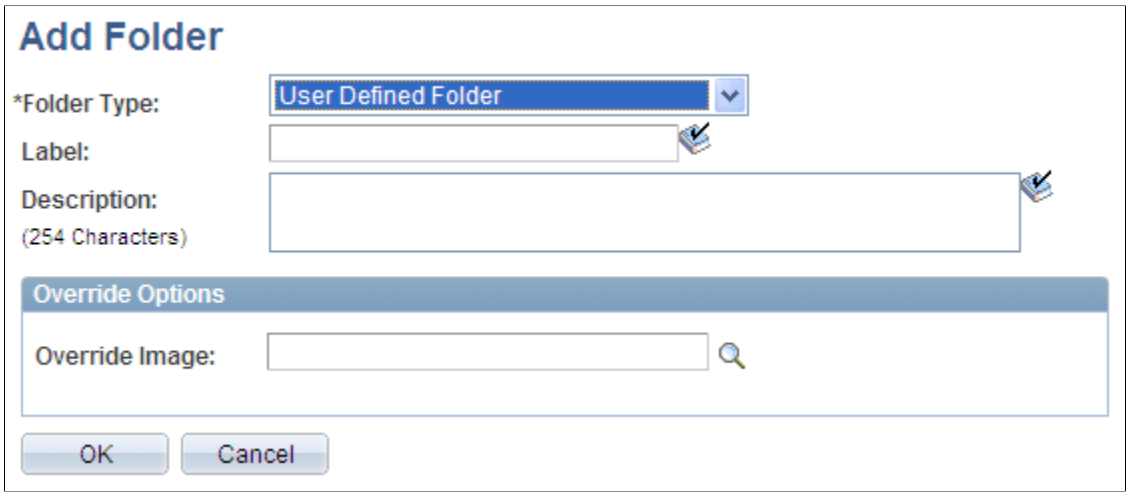
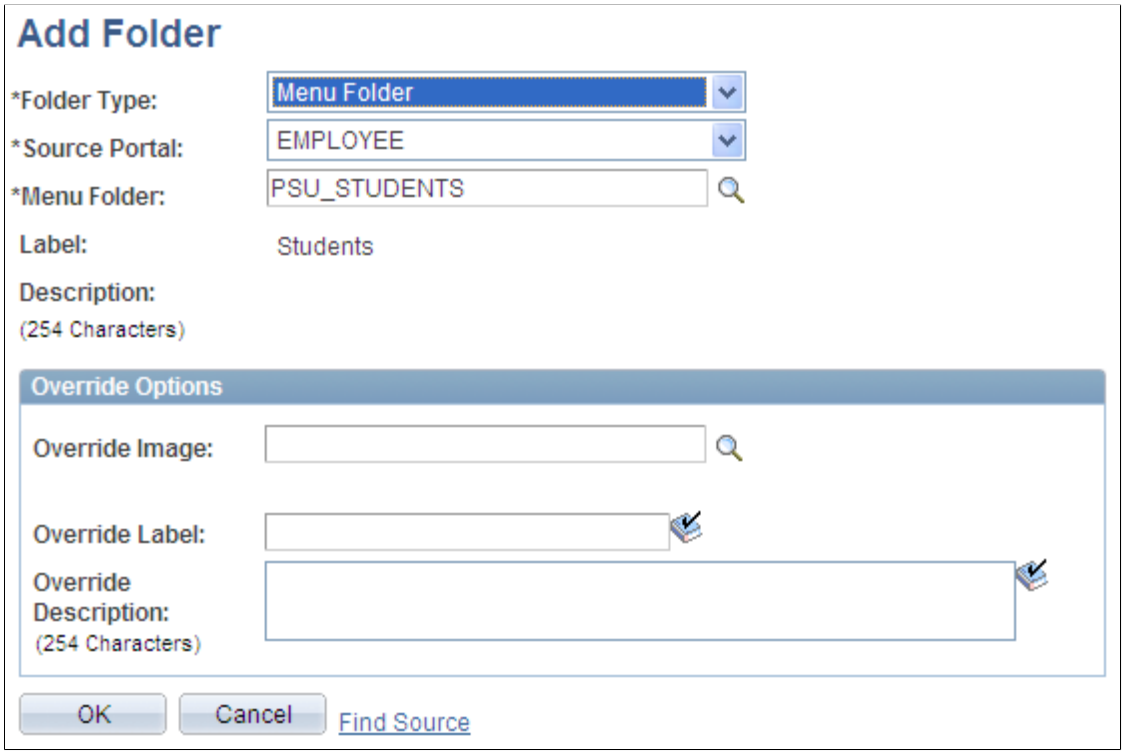


Image: Add Folder page (menu folder)

This example illustrates the fields and controls on the Add Folder page (menu folder). You can find definitions for the fields and controls later on this page.



Folder Type Select one of these values:

Menu Folder: Creates a reference to an existing folder in the portal registry. The reference automatically inherits all of the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 date and to date for the folder.

Source Portal

This field appears when the Folder Type value is set to *Menu Folder*.

Select the portal registry from which you want to select a folder. 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 in which a navigation collection will be copied between portals, and the folder reference that you are defining should automatically refer to the current portal.

See Understanding Portal Administration.

Source Folder

This field appears 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 that you want to reference.

Label

If you are adding a menu-based folder, this field is populated with the label on the source, which 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 page or pagelet.

Description

If you are adding a menu-based folder, this field is populated with the description on the source, which can be overridden in the Override Description field.

This text appears when the mouse cursor is moved over a folder link and as descriptive text beneath the folder link on the navigation collection page or pagelet.

Override Options

Override Image

Select the image that 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 overrides the default. This image appears as the large image for the folder on the navigation collection publication.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

When defined for folders nested within folders, the image appears 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 Setting Up Navigation Collection Options, Navigation Collection Images.

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 appears for users who can access the Structure and Content pages 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 source reference security.

See Administering Folders.

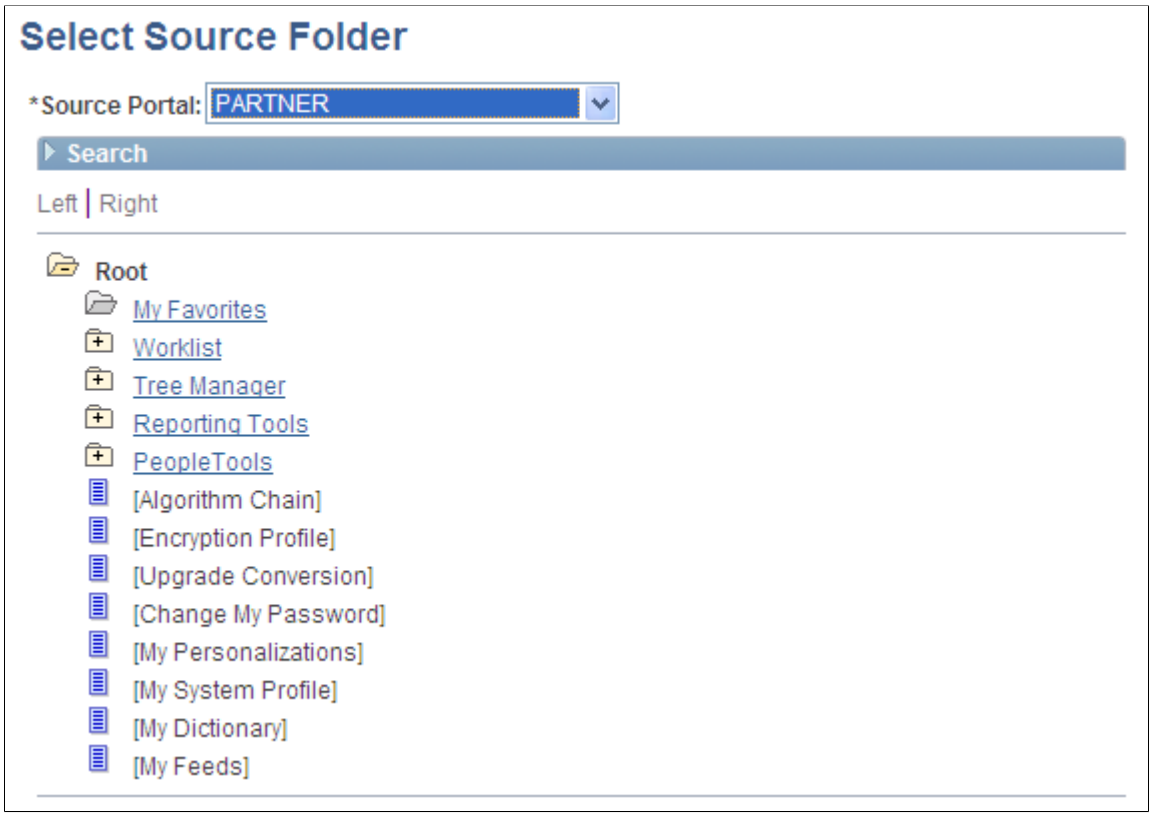
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting a Source Folder

Access the Select Source Folder page. (Click the Browse Portal Registry icon on the Add Folder page.)

Image: Select Source Folder page

This example illustrates the fields and controls on the Select Source Folder page. You can find definitions for the fields and controls later on this 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 in which a navigation collection will be moved between portals, and the folder reference that you are defining should automatically refer to the current portal.

See Administering Portal Definitions.

Search

Search by

Select one of these values:

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* results, but may return results with *resource.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Label: Select to conduct your search based on folder label text.

Alternatively, 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.

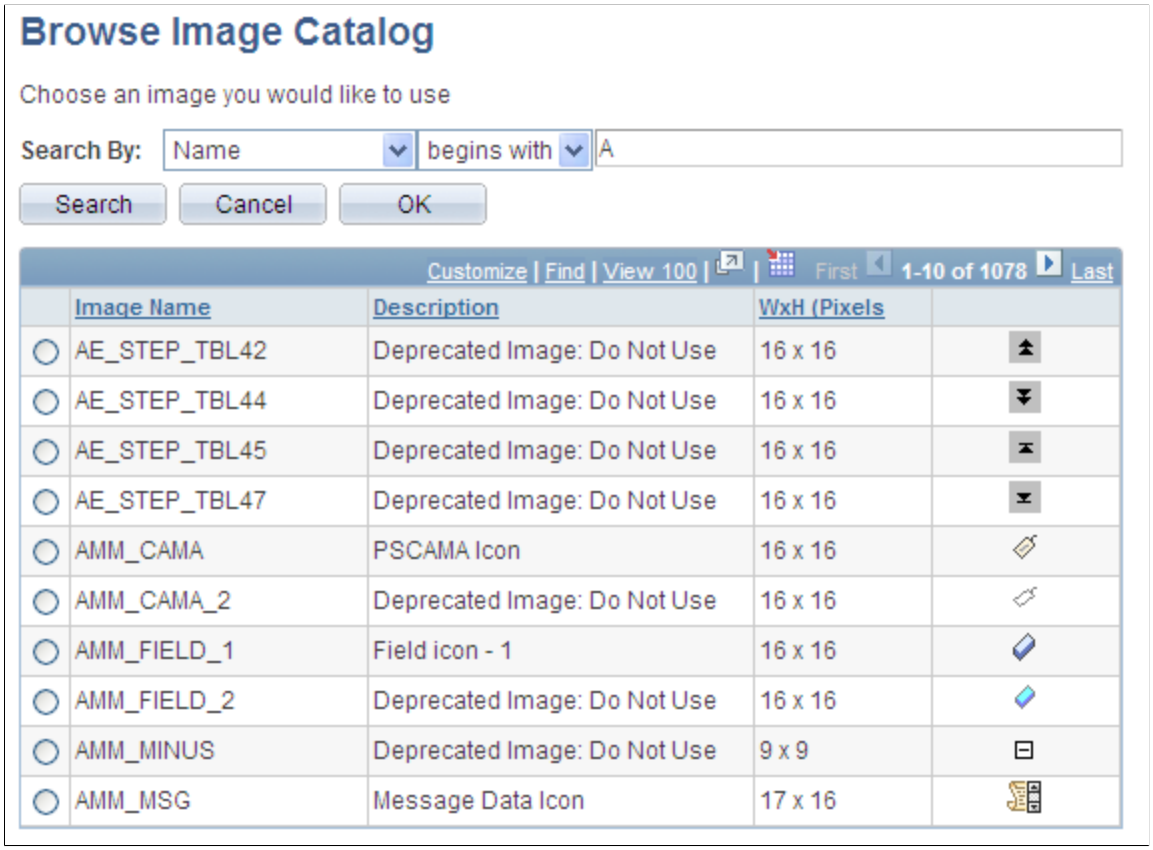
Once you have located your desired folder, click the folder link in the hierarchy to select the folder and return to your originating page.

Selecting Navigation Collection Images

Access the Browse Image Catalog page. (Click the Browse Image Catalog button on the Add Folder page.)

Image: Browse Image Catalog page

This example illustrates the fields and controls on the Browse Image Catalog page. You can find definitions for the fields and controls later on this page.



Search by Select one of these values:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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* results, but may return results with *resource.*

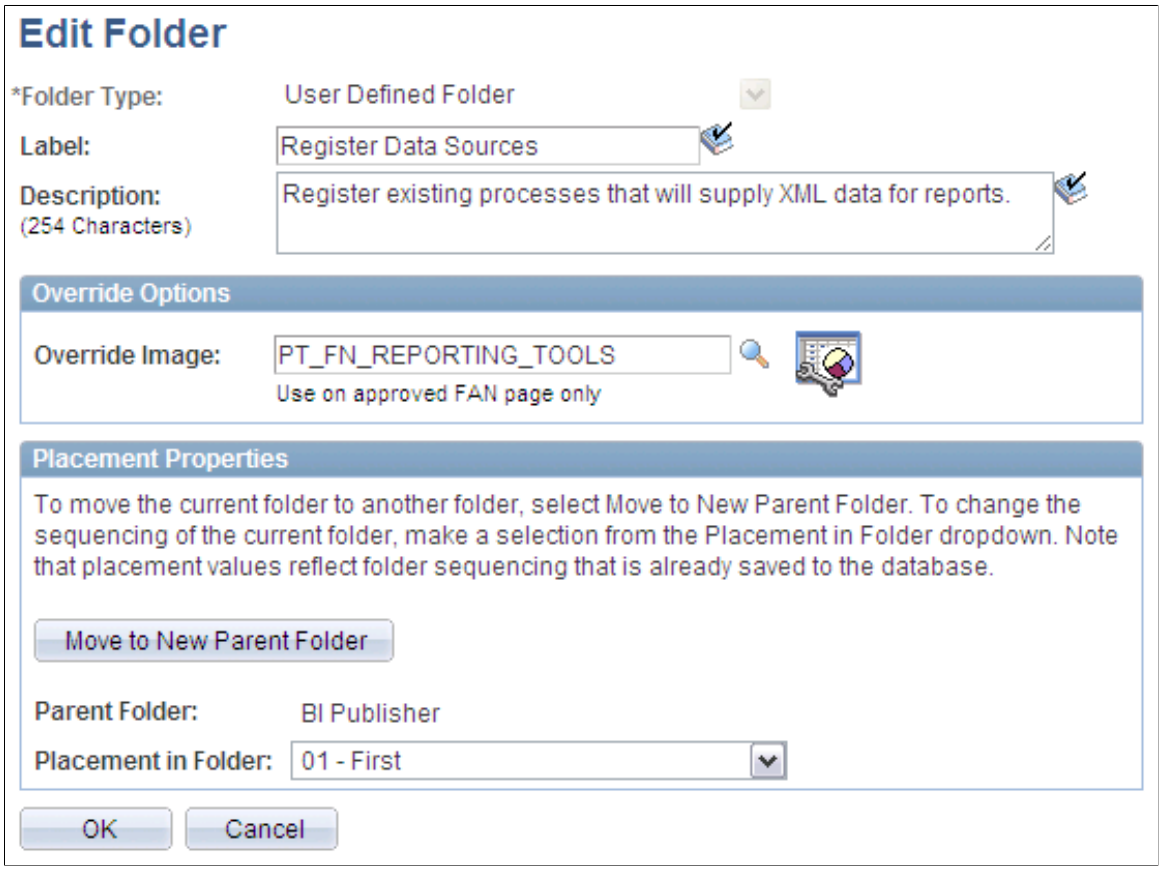
Once you have located the desired image, select the radio button for the image and click OK.

Editing a Folder

Access the Edit Folder page. (Click the Edit Folder button on the Maintain Collection page.)

Image: Edit Folder page

This example illustrates the fields and controls on the Edit Folder page. You can find definitions for the fields and controls later on this page.



With the exception of the Placement Properties group box, which is discussed subsequently, all other options on this page are available on and documented as a part of the Add Folder page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Adding a Folder.

Placement Properties

Move to New Parent Folder	Click to move the folder that you are editing to a folder other than the folder designated in the Parent Folder field. You can change the placement of a folder reference only after you have initially saved the navigation collection.
Parent Folder	Displays the parent folder of the folder that you are editing.
Placement in Folder	Select a value to change the placement of the folder within the parent folder. Available placement values reflect saved folder sequencing.

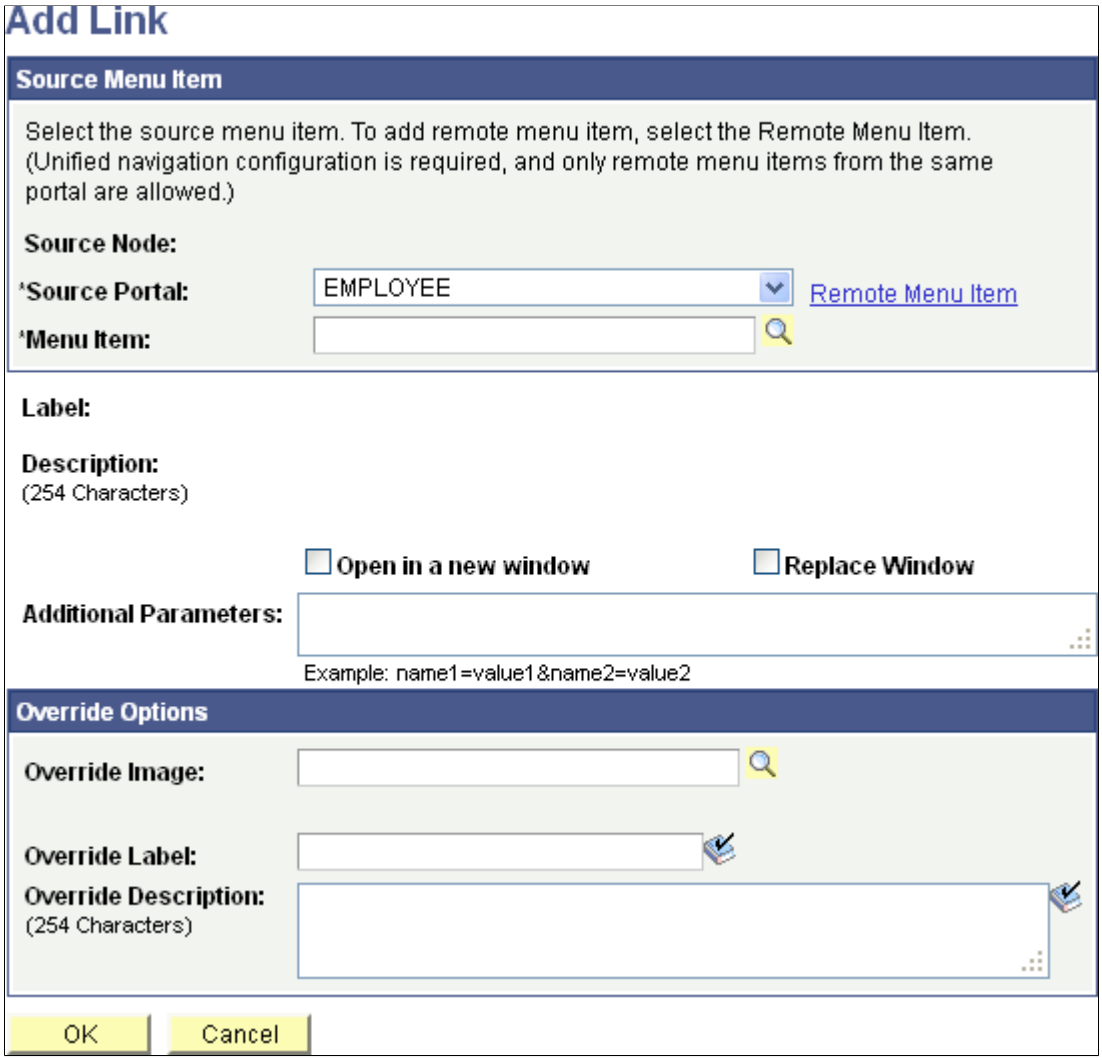
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Adding a Link

Access the Add Link page. (Click the Add Link button on the Maintain Collection page.)

Image: Add Link page

This example illustrates the fields and controls on the Add Link page. You can find definitions for the fields and controls later on this page.



Source Node

Displays the name of the remote source node if a remote menu item has been selected.

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 in which a navigation collection will be copied between portals, and the content reference that you are defining should automatically refer to the current portal.

See Administering Portal Definitions.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: For remote menu items, the Source Portal defaults to the current portal and is read only.

Menu Item

Select a local content reference to add this link 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 that you want to add.

See Selecting a Local Link.

Note: You can use this option to add a published navigation page to the navigation collection that you are creating.

Remote Menu Item

Select a remote component-based content reference to add to the navigation collection from the Select a Content Reference page.

See Selecting a Remote Link.

Label

Displays the existing menu content reference label text, but you can override this default using 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

Displays the existing description text, but you can override the default using the Override Description field. This description text appears when the mouse cursor is moved over the content reference link.

Open in a new window

Select to open the page in a new browser window. This option and Replace Window are mutually exclusive.

Replace Window

Select to replace the current browser window, including navigation and target area, with the new content. Replace window updates the template and navigation to display the template and navigation of the target content reference. This option and Open in new window are mutually exclusive.

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 eliminates the need 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Override Options

Override Image

Select the image to display as the large image for the content reference. If a default content image has been defined at the system or registry level, the selected image overrides the default. This large image appears for the content reference on the navigation collection publication.

See Understanding Navigation Collection Options.

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 appears for users who can access the Structure and Content pages 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 date and from date, and content provider for the selected source content reference.

See Administering Content References.

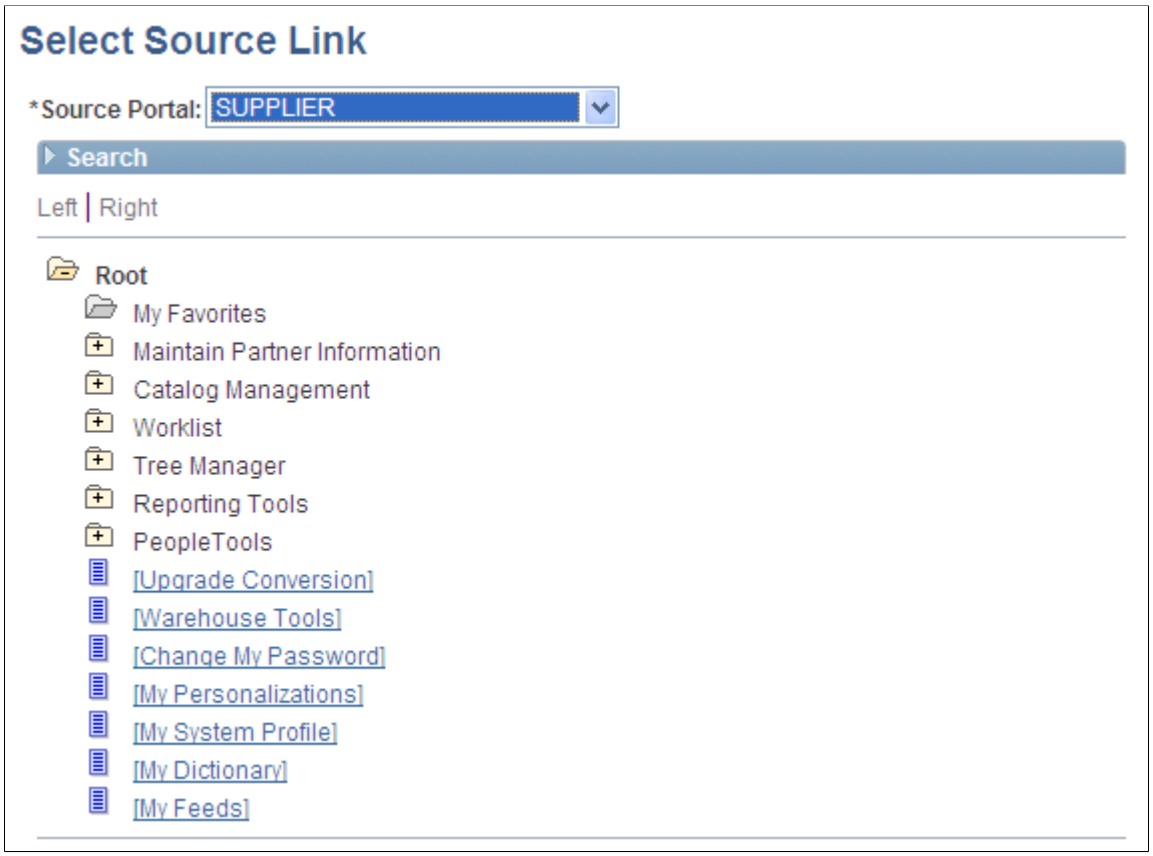
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting a Local Link

Access the Select Source Link page. (Click the Browse Portal Registry button on the Add Link page.)

Image: Select Source Link page

This example illustrates the fields and controls on the Select Source Link page. You can find definitions for the fields and controls later on this 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 in which a navigation collection will be moved between portals, and the content reference that you are defining should automatically refer to the current portal.

See Administering Portal Definitions.

Search

Search by

Select one of these values:

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* results, but may return results with *resource.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Label: Select to conduct your search based on content reference label text.

Alternatively, 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.

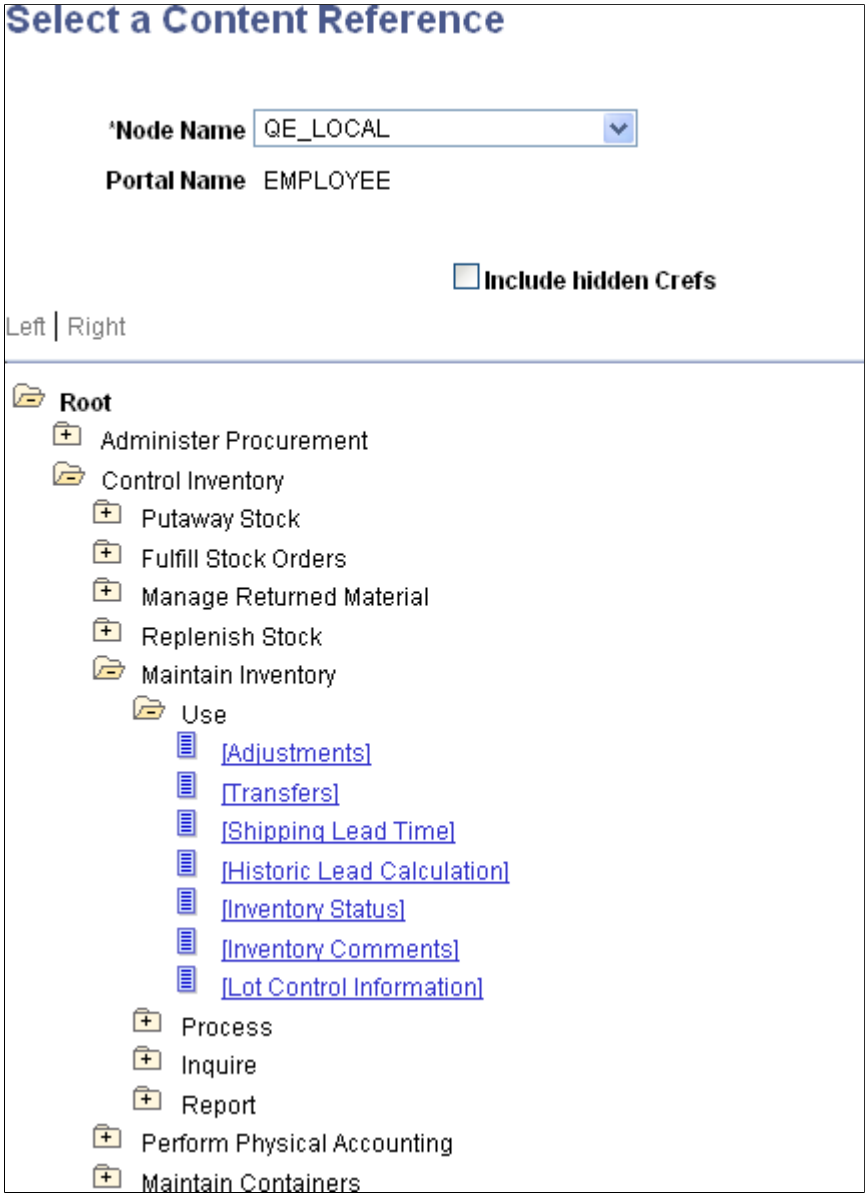
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting a Remote Link

Access the Select a Content Reference page. (Click the Remote Menu Item link on the Add Link page.)

Image: Select a Content Reference page

This example illustrates the fields and controls on the Select a Content Reference page:



Important! In order to select a remote link, unified navigation must be configured between the local PeopleSoft Interaction Hub system and the remote content provider system.

To select a remote link:

1. Select the remote node.
2. Navigate the remote folder and menu item hierarchy.
3. Select the remote, component-based content reference as the remote link.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To configure unified navigation on the PeopleSoft Interaction Hub system, see *PeopleSoft Interaction Hub: Portal and Site Administration*, "Administering Unified Navigation in PeopleSoft Applications Portal."

Editing a Link

Access the Edit Link page. (Click the Edit Link button on the Maintain Collection page.)

Image: Edit Link page

This example illustrates the fields and controls on the Edit Link page. You can find definitions for the fields and controls later on this page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

With the exception of the Placement Properties group box, which is discussed subsequently, all other options on this page are available on and documented as a part of the Add Link page.

See Adding a Link.

Placement Properties

Move to New Parent Folder	Click to move the content reference that 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 that you are editing.
Placement in Folder	Select a value 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

This section discusses how to publish navigation collections.

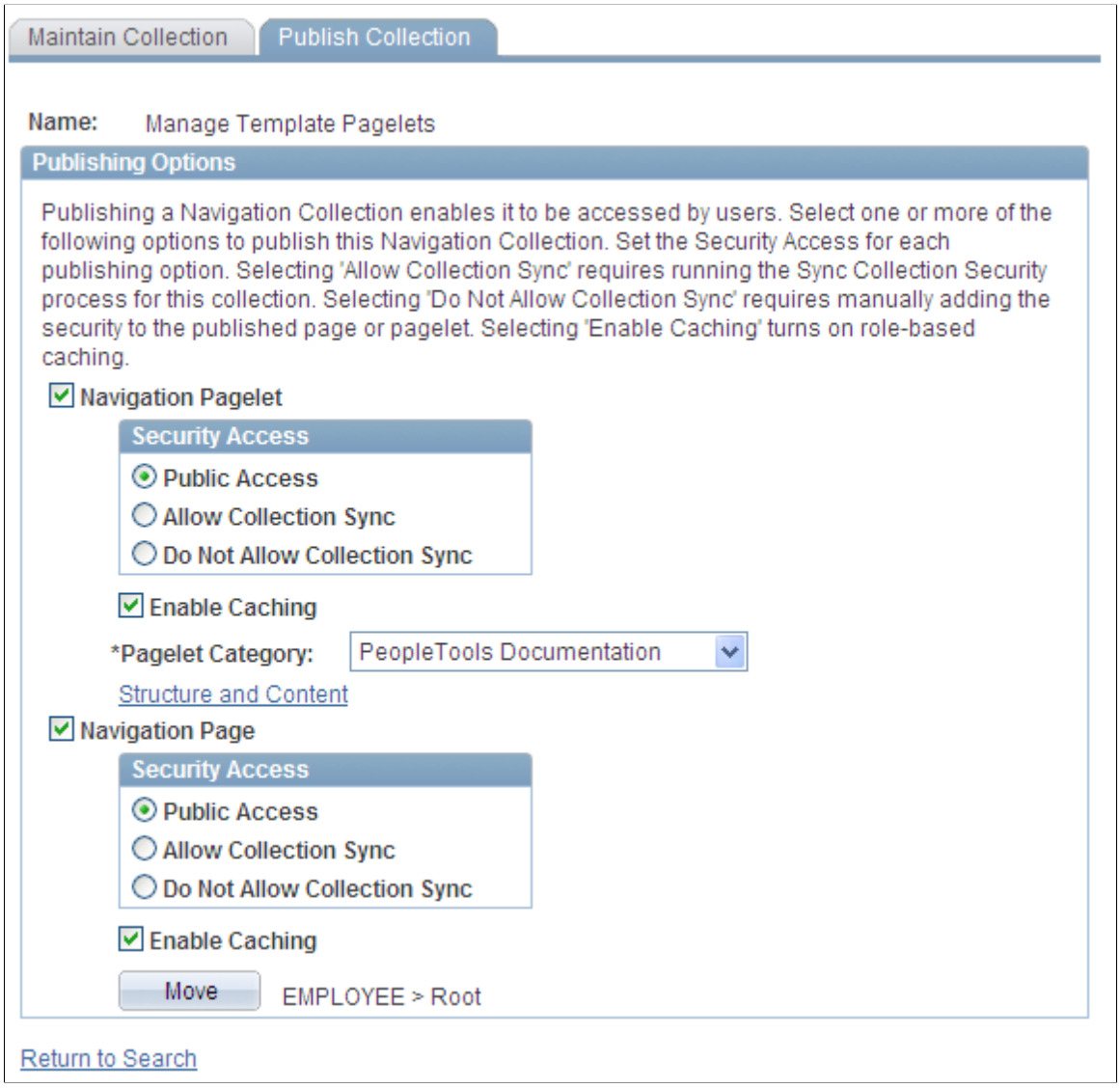
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Publishing a Navigation Collection

Access the Publish Collection page. (Select PeopleTools, Portal, Portal Utilities, Navigation Collections, Publish Collection.)

Image: Publish Collection page

This example illustrates the fields and controls on the Publish Collection page. You can find definitions for the fields and controls later on this page.



Name Displays the name of the navigation collection defined on the navigation 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.

Security Access Select one of these values:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Public Access: Select to give all users access to the pagelet. This option is selected by default.

Allow Collection Sync (allow collection synchronization): Select to make the pagelet eligible for processing by the Sync Collection Security process.

Do Not Allow Collection Sync (do not allow collection synchronization): Select if you do not want the pagelet security to be updated by the Sync Collection Security process. Select this option to ensure that security that you manually define for the pagelet cannot be overridden by the process.

See Understanding Portal Utility Processes.

Enable Caching

Select to have role-based caching enabled for this published pagelet. Selecting this option adds portal caching attributes to the published navigation pagelet.

See Understanding Portal Caching.

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 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 the results of the Sync Collection Security (synchronize collection security) Application Engine process (PTPP_SCSYNC) and change content reference label text or manually update the security.

See Understanding Portal Utility Processes, Administering Content References.

Navigation Page

Select to publish the navigation collection as a navigation page, which can be accessed from the menu.

Security Access

Select one of these values:

Public Access: Select to give all users access to the page. This option is selected by default.

Allow Collection Sync (allow collection synchronization:. Select to make the page eligible for processing by the Sync Collection Security process.

Do Not Allow Collection Sync (do not allow collection synchronization): Select if you do not want the page security

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	updated by the Sync Collection Security process. Select this option to ensure that any security that you manually define for the page is not overridden by the process. See Understanding Portal Utility Processes.
Enable Caching	Select to have role-based caching enabled for this published page. Selecting this option adds portal caching attributes to the published navigation page. See Understanding Portal Caching.
Move	Click to access the Select Source Folder page, where you can select the parent folder in which you want the navigation page to appear in the left navigation menu. See Selecting a Source Folder.
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 the results of the Sync Collection Security process and change navigation page label text. See Understanding Portal Utility Processes, Administering Content References.

Publishing Navigation Collections as Pagelet Wizard Pagelets

Navigation collections can be published directly as navigation pagelets, as Pagelet Wizard pagelets, or both. Because Pagelet Wizard provides many options for controlling the look of the pagelet, Oracle recommends that you use Pagelet Wizard to publish pagelets from navigation collections. Some of the publishing options available through Pagelet Wizard but not when directly publishing the pagelet include:

- At step 3, designating the pagelet as configurable (administrators), personalizable (users), or both.
- At step 4, indicating that you intend to use a custom XSL template for transforming the output format for the pagelet.
- At step 5:
  - Specifying a custom XSL template.
  - Displaying or hiding the top level title, description, or both.
  - Displaying or hiding folder descriptions.
  - Specifying the image type for folders.
  - Specifying the list type for links.
  - Adding pagelet headers and footers.
- At step 6:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Indicating whether the pagelet will be published as a homepage pagelet, template pagelet, embedded pagelet, or marked for WSRP production.
- Displaying or hiding the pagelet title bar and borders.

Related Links

Selecting a Navigation Collection Data Source
Specifying Navigation Collection Data Source Parameters
Step 4: Selecting a Pagelet Display Format
Step 5: Specifying Pagelet Display Options
Step 6: Specifying Pagelet Publishing Options

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 5

Administering Portals

Understanding Portal Administration

Every component delivered with PeopleSoft applications is registered as a content reference in the portal registry . Common administrative tasks include adding, deleting, and renaming these definitions in the portal. If you have portal administrator access, you can use the administration pages to manage the registry, including folders, content references, and security.

Additionally, to provide programmatic access to the portal registry, PeopleTools delivers a portal registry API. This API provides the same kind of registry management capability as the administration pages and is fully described in the PeopleCode documentation. Developers can also register components by using the Registration Wizard, which is part of PeopleSoft Application Designer.

You use the administration pages to:

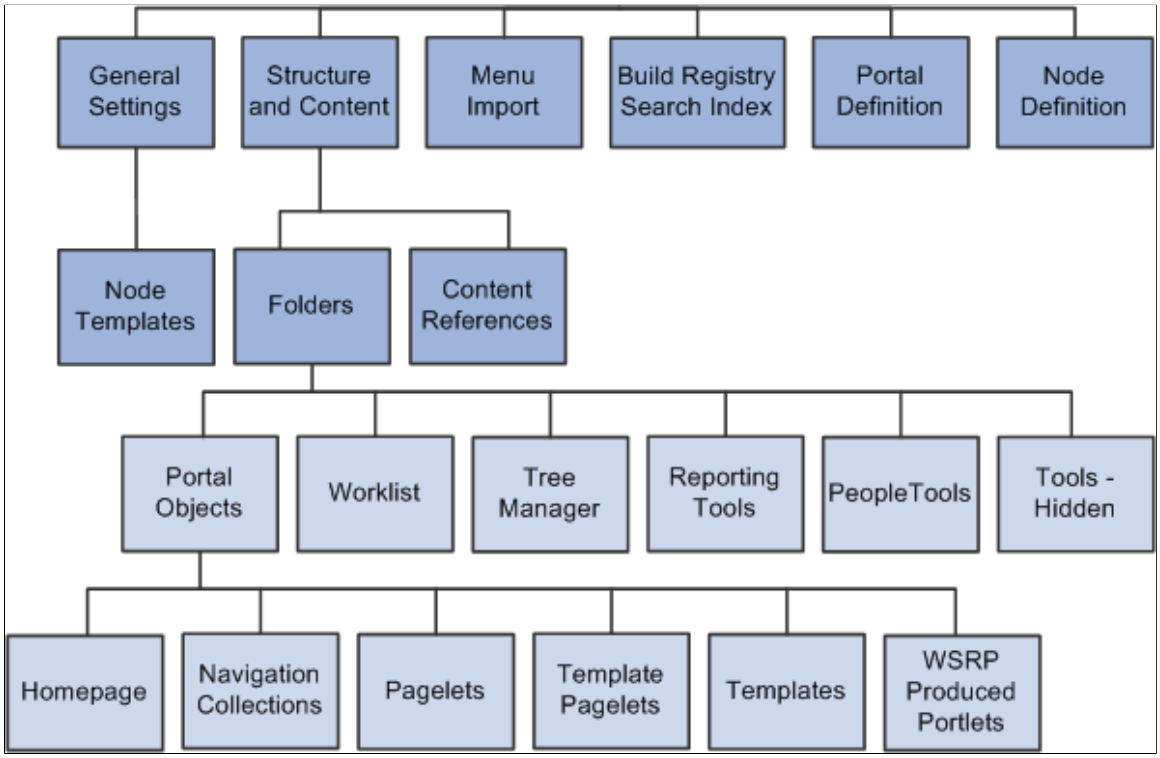
- Manage the portal registry, or the structure and content of a portal, which includes:
  - Templates.
  - Navigation collections.
  - Folders.
  - Content references.
  - Pagelets.
  - Homepages and dashboard pages.
- Manage general portal settings, which include node templates, the portal template, navigation options, portal search options, and so on.
- Add, edit, and delete local and remote portal definitions.
- Add, edit, and delete local and remote node definitions.
- Populate a portal registry with custom menus by importing menu definitions.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Generate a Verity search index or collection that reflects the content of the portal registry.

Image: Administration pages under the PeopleTools > Portal folder

The following diagram shows the organizational structure of some of the portal administration pages under the PeopleTools > Portal folder. Note the location of the special Portal Objects folder. This folder contains folders and content references for all homepages, templates, and pagelets associated with a given portal.



Note: To view the portal administration pages, you must have the appropriate security permissions.

Related Links

"Understanding Nodes" (PeopleTools 8.54: Integration Broker Administration)

"Using the Portal Registry API" (PeopleTools 8.54: PeopleCode API Reference)

Understanding Folder Navigation

PeopleSoft applications integrate folders into the menu, or navigational, structure. While most forms of navigation in PeopleSoft applications use folders, the term *folder navigation* has a specific meaning for PeopleTools portals. Folder navigation is the default method of navigation delivered in PeopleSoft applications and refers to a specific response by the portal when the user clicks a folder, which is rather than expanding and collapsing menu items, the portal generates navigation pages in the target content area.

Navigation Pages

Note these points about the characteristics of navigation pages:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Navigation pages provide users with an organized, graphical approach to traversing the menu structure.
- Navigation pages, like menus, reflect the structure of the portal registry. Each navigation page corresponds to a folder in the portal registry and provides access to any subordinate folders and content references.
- Users click folders to navigate through the menu structure to other navigation pages.
- Breadcrumb links appear in the header of the navigation page to indicate position in the hierarchy of folders. Click a breadcrumb link to return a higher-level navigation page.
- Navigation pages provide access to access transaction pages when users click a content reference link.
- The character and appearance of navigation pages is controlled by a default folder navigation object (page) that is standard with every PeopleTools implementation.
- You can modify the appearance of any navigation page by changing its default folder navigation object.
- The folders and content references that appear on navigation pages vary by user according to his or her assigned permissions.

Image: Example navigation page for the Security folder

The following screenshot is an example of navigation page for security folder, it uses the default folder navigation object, shows the Security folder when folder navigation is enabled:

Traversing Menus Without Engaging Navigation Pages

If you use one of these methods to navigate the menu structure, the portal will not generate navigation pages.

- *For drop-down navigation:* Click the folder name. This action causes the menus to cascade sideward revealing subfolders and content references.
- *For left navigation:* Click the triangle icon to the left of the menu name. This action causes the menus to expand downward revealing submenus and content references.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Important! You enable folder navigation at the portal definition and general portal settings levels; folder navigation must be enabled at both levels for the feature to work. When you enable folder navigation at the site level, you can navigate all folders by using this method; however, you can disable or override folder navigation at the individual folder level.

Note: If folder navigation is *disabled,* then navigation pages do not appear when the user clicks a folder. Instead, the item expands to display any subordinate menu items. The user continues to expand menu items by clicking them until he or she clicks a content reference. No content appears in the target area until the user clicks the content reference.

Related Links

Adding and Editing Portal Definitions
Managing General Settings for Portals
Understanding Navigation Pages
"Understanding Permission Lists" (PeopleTools 8.54: Security Administration)
"Understanding Roles" (PeopleTools 8.54: Security Administration)

Common Elements Used in This Chapter

Chart Navigation Object Name

At the portal definition and general settings levels, specify the portal object that creates the general appearance and character of SmartNavigation folders. The system default value is *PT_PORTAL_DEFAULT_CHART_NAV,* which is a hidden iScript-based content reference that renders the basic SmartNavigation page.

At the individual folder level, specify an object that overrides the system-level chart navigation object.

If you don't specify a value at the folder level, the portal uses the system-level chart navigation object.

Copy object

Click to access the Copy Object to a Different Portal Registry page, which enables you to copy an existing folder to a different portal registry. This button appears only when you're editing a folder.

Default Chart Navigation Page

Select to replace the standard navigation page with chart-based SmartNavigation page. This field is disabled by default. To enable this field, select a value in the Data Source field.

If you select this check box, then the application uses *PT_ABN_ORGCHART* component as the default PeopleTools chart navigation page. If you deselect this check box, then you must enter your own value in the Folder Navigation Object Name field.

Folder Navigation Object Name

At the portal definition and general settings levels, specify the content reference that the system launches when the user clicks a folder icon. The system default value is *PT_PTPP_SCFNAV*

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

_BASEPAGE_SCR, which is a hidden iScript-based content reference that renders the standard navigation page.

At the individual folder level, specify an object that overrides the system-level default.

If you don't specify a value at the folder level, the portal uses the system-level folder navigation object.

Important! You must use the definitions associated with the *PT_ABN_ORGCHART* component as the basis of any custom SmartNavigation folder object.

Note: If you select a content reference that's hidden from portal navigation, the folder navigation occurs, but the menu name that you click remains highlighted.

Hide From Portal Navigation

Select to make this folder invisible to users.

Is Folder Navigation Disabled

Select to override the system-level default setting and disable folder navigation for individual folders.

Note: If you disable and then re-enable folder navigation at the system level, this field is automatically cleared.

Is Folder Navigation Enabled

Select to enable folder navigation for all folders in the current portal registry or selected portal definition.

When folder navigation is enabled, if the user clicks the folder name in the menu structure, the navigation page or a content reference that you specify in the Folder Navigation Object Name field appears in the target area.

Label

Enter the text to appear as the link for this folder.

Long Description

Enter text to appear when the cursor rests over the link for this folder. If you leave this field blank, the label appears.

Name

Enter a name for the folder.

Note: The name can consist of letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.

(Object) Owner ID

Select the owner of the portal object such as portal, folder, or content reference. The owner ID helps determine the team that created or modified the object. You can define unique owner IDs for your organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer. Do not use PeopleSoft owner IDs.

Parent Folder

The name of the current parent folder. This field is display-only.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Product (Optional) Portal-aware applications can use this field for group processing. All PeopleTools objects have the product name *PT.*

Select New Parent Folder Click to copy the folder to a new location within the current portal registry. This button appears only when you're editing a folder.

Sequence number Enter a sequence number to define where this folder appears in its parent folder, and therefore where its associated menu entry appears within its parent menu. If you leave the sequence number blank, the folder is added to the top of the list of folders and is displayed in alphabetical order. You can change the sequence only one item at a time, and each time you do so, you must save your changes to see the result.

Valid from date and Valid to date Enter a valid from date. The valid to date is optional and is omitted by default. These dates can be used for application-specific processing by portal-aware applications and are used by the portal navigation to determine visibility.

Administering Portal Definitions

This section lists common elements used in this section and discusses how to:

- Define portals.
- Add and edit portal definitions.
- Change default portals.

Common Elements Used in This Section

Hosted by this node The node where the portal is defined. This is either a remote or local node. The node must have been defined previously on the Maintain Node Definitions page.

Portal Name The name to appear as the portal label.

This value is used for the browser window title when a long description has not been entered.

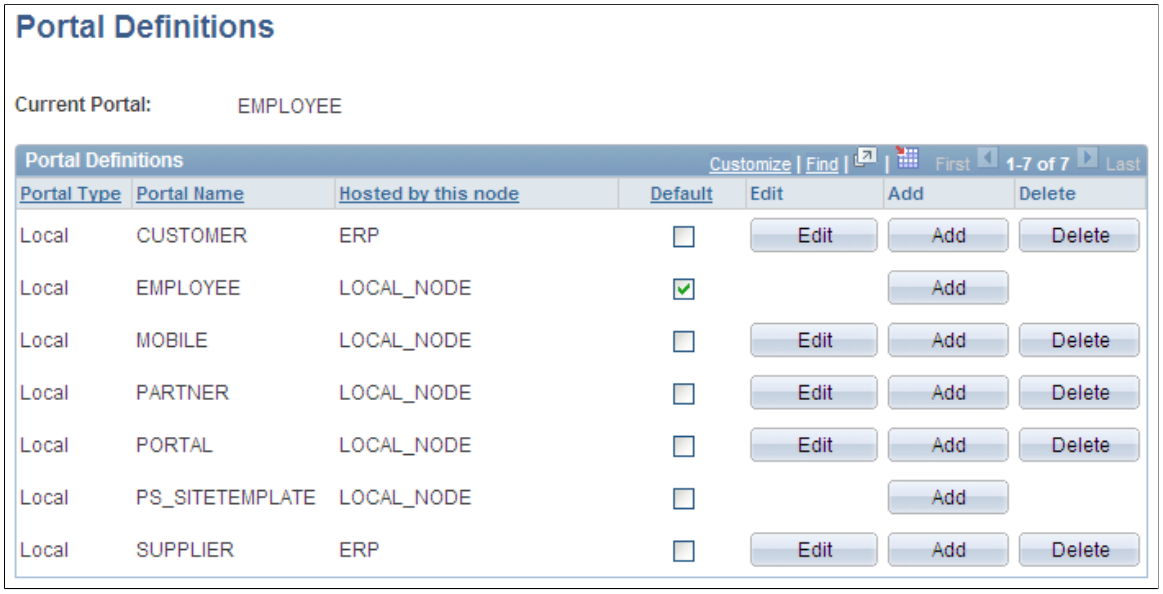
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Portals

Access the Portal Definitions page (PeopleTools, Portal, Portal Definitions).

Image: Portal Definitions page

The following screenshot describes the different portal definitions on the Portal definitions page.



You can have multiple portals on a database or create one portal on multiple databases. You can define portals from other databases and associate them with a node. This supports transfers between portals and redirected requests for a portal not supported by the local web server.

Portal Type — Displays the attribute of the selected node. Possible types are *Local* and *Remote.*

Default — Select to use this as the default node. Only local nodes can be the default. If you change the default, be sure to click the Save button.

Edit — Click to update the portal definition.

> **Note:** This button does not appear for the site template or the local node that is currently in use. To modify the current portal, use the General Settings page.

See Managing General Settings for Portals.

Add — Click to add a new portal definition. The new portal is listed in alphabetical order.

Delete — Click to delete a portal definition. You cannot delete the site template or node that is currently in use.

Related Links

Managing General Portal Settings

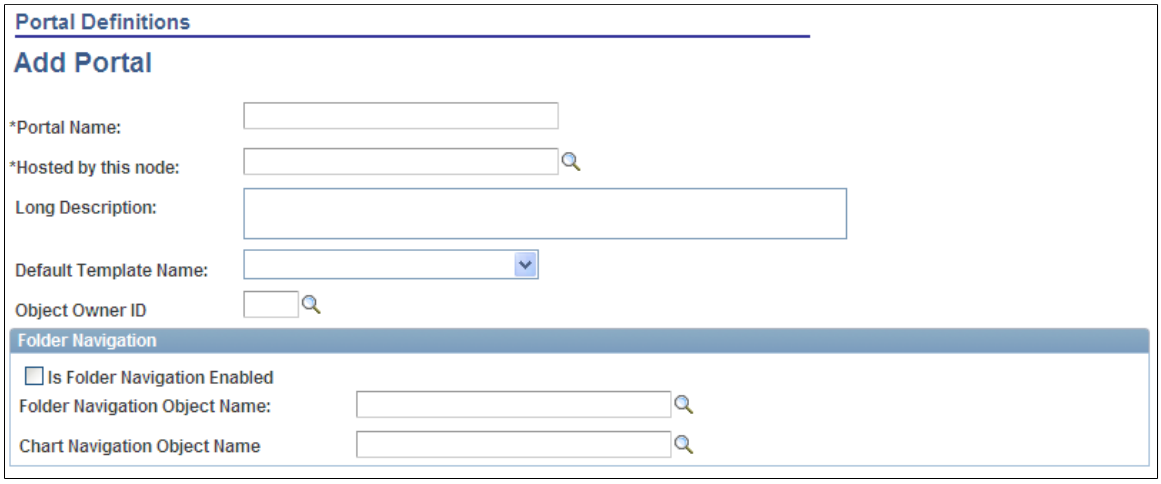
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Adding and Editing Portal Definitions

Access the Portal Definitions - Add Portal page (Click the Add button on the Portal Definitions page).

Image: Add Portal page

The following screenshot illustrates how to add and edit portal definitions on the Add Portal page.



Note: The Edit Portal page contains the same fields as the Add Portal page (Click the Edit button on the Portal Definitions page).

Long Description	Enter a description to be used by screen readers if accessibility mode is enabled. This description is also used as the browser window title. If you don't enter a description in this field, the portal name is used as the browser window title.
Default Template Name	Select the template to use for this new portal.

Folder Navigation

Use these fields to enable folder navigation and define the objects associated with folder navigation for portals other than the one you are currently signed in to. To enable folder navigation for the current portal, use the General Settings page.

Important! Folder navigation is not enabled unless you select the Is Folder Navigation Enabled check box on both the Portal Definition and General Settings pages. Is Folder Navigation Enabled is selected by default in the standard PeopleTools portals.

See Managing General Settings for Portals.

Changing Default Portals

You can change the default portal in two ways:

- Change the default portal in the node definition.
- Insert the new default portal name into the URL.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Changing the Default Portal in the Portal Definition

To change the default portal:

1. In the current default portal, select PeopleTools, Portal, Node Definitions.
2. Select the default local node and then click the Portal tab.
3. Select the Default check box for the portal that you want to open.
4. Save.
5. Sign out.
6. Shut down and restart the portal web server.
7. Sign in.

   When you sign in, the new default portal is invoked.

Changing the Default Portal by Inserting the New Portal Name in the URL

You can also change the default portal by changing the URL. To change the URL, insert the new portal name instead of the existing portal name. For example, to change to the supplier portal from the employee portal, make the following substitution:

URL before the change: http://localhost/psp/ps/*EMPLOYEE*/QE_LOCAL/c/UTILITIES.PORTAL_DEFINITIONS.GBL

URL after the change:http://localhost/psp/ps/*SUPPLIER*/QE_LOCAL/c/UTILITIES.PORTAL_DEFINITIONS.GBL

Note: This type of portal change is valid until the browser session is closed.

Administering Folders

This section provides an overview of folder administration and discusses how to:

- Define folders.
- Set folder permissions.

Understanding Folder Administration

In the portal registry, folders organize content references into a multilevel hierarchy. A folder can contain content references and other folders. When you create folder definitions, you explicitly configure the properties, attributes, and position of the folder in the hierarchy. The portal uses your configurations to create the menu structure, which remains static unless you modify structure in the portal registry.

Organization of the Portal Registry

You use the Structure and Content component to access the portal registry, where you create and manage folders and content references. When you click the Structure and Content link, the application opens the portal registry at the root level. The initial set of folders, content references, and links that you see correspond to the high level folders and links that appear directly under Main Menu or in the Menu pagelet.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The Structure and Content component enables you to organize and manage your folders and content references; all folders and content references that share the same Structure and Content page are at the same level hierarchically and share the same parent folder. You click the Edit link to access the properties of a folder. You click the folder label link to navigate one level lower in the hierarchy where any child folders and content references that exist appear.

As you navigate through the hierarchy of folders, breadcrumbs appear at the top of the page directly beneath the Main Menu navigation bar. The breadcrumbs appear to visually indicate your current position in hierarchy of the portal registry. Click a link in the breadcrumbs to return to the folder. In this sample Structure and Content page, the breadcrumbs indicate that you are viewing elements that appear in the PeopleTools, Security folder:

Image: Structure and Content page showing the folders and content references that appear in the Security folder

The following screenshot illustrates the Structure and Content page showing the folders and content references that appear in the Security folder.

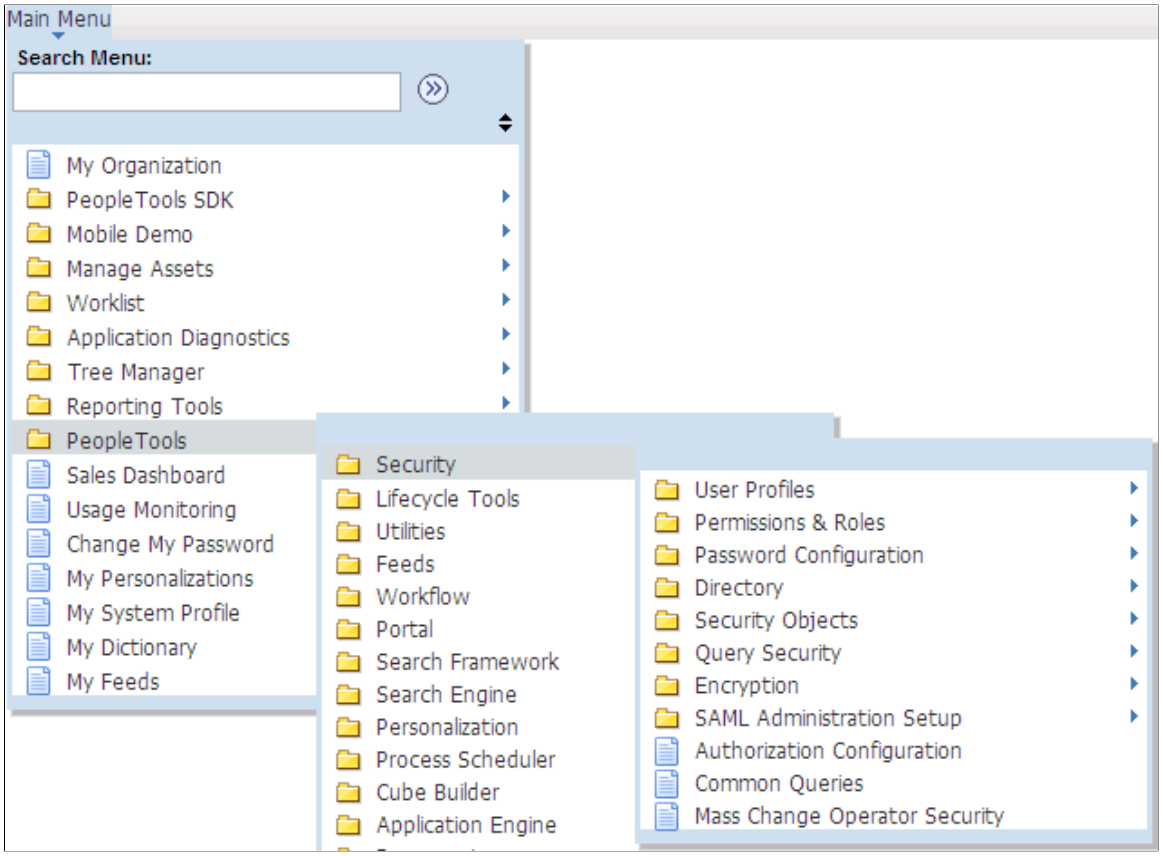
The portal uses the hierarchy of folders in the portal registry to create the menu structure of the PeopleSoft application. Menus open to reveal the folders and content references in the same general sequence and location as they appear in the portal registry. In this sample drop-down menu, notice that

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

the organization of folders and content references in the PeopleTools, Security folder correspond to the folders and links located in the Security folder in the portal registry:

Image: Menu folder and content reference organization in the Security folder

This example illustrates how the folder and content reference organization for the same Security folder is rendered in the drop-down menu .



Folders can appear at any level in the menu hierarchy in both the left navigation Menu pagelet and the Main Menu drop-down navigation. Folders cascade open when users click the folder label and generate navigation pages when you click the folder icon. Most PeopleSoft application folders are standard navigation folders; however, you can configure certain folder properties to create SmartNavigation folders.

Related Links

Administering Folders

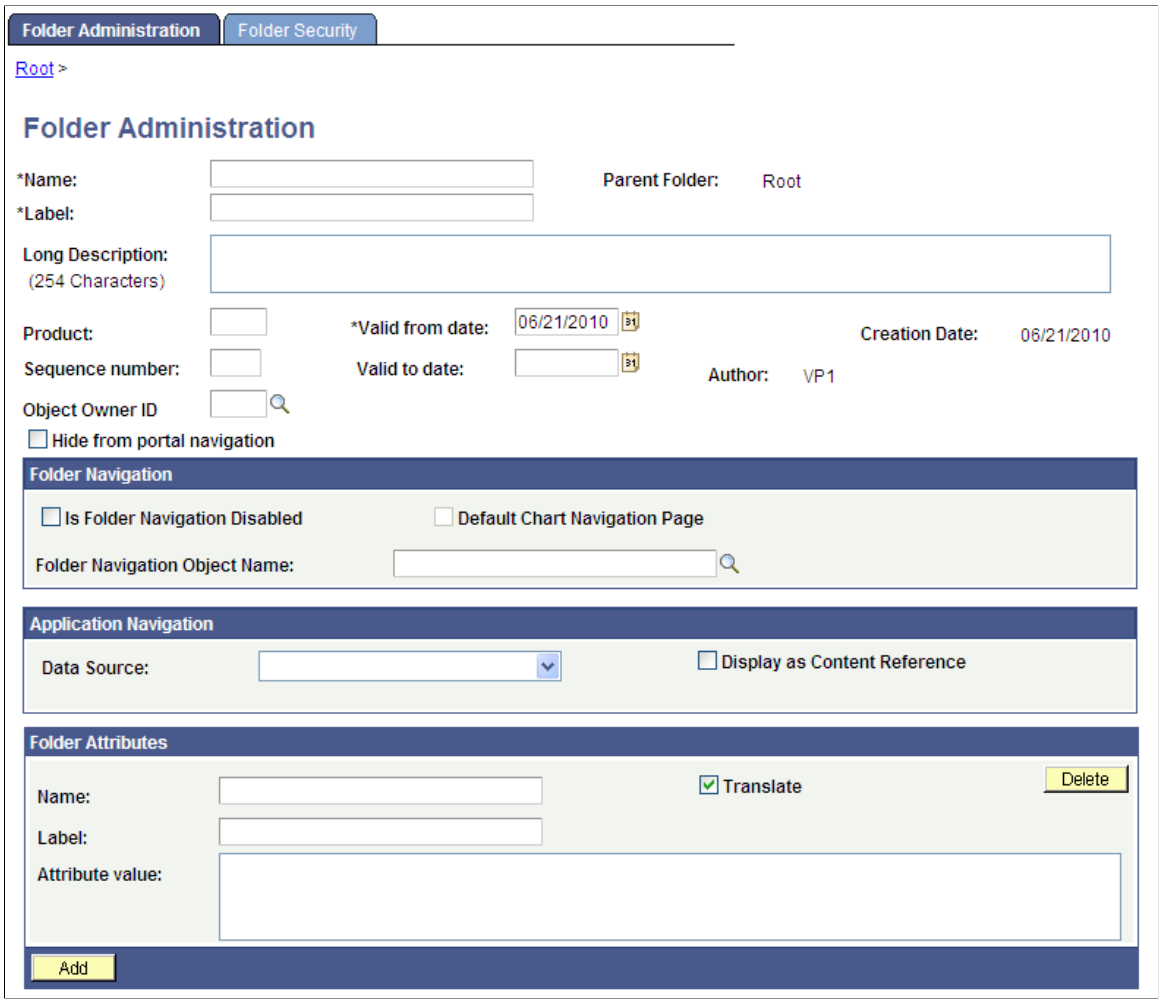
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Folders

Access the Folder Administration page (Select PeopleTools, Portal, Structure and Content Click the Add Folder link).

Image: Folder Administration page

The following screenshot illustrates how to define folders on the Folder Administration page.



1. Enter values in the Folder Administration fields.
2. (Optional) Enter values in the Folder Attributes fields.
3. If this folder is a SmartNavigation folder, enter values in the Application Navigation fields.

   See Defining SmartNavigation Folders.
4. Set the folder security.

   See Setting Folder Permissions.
5. Save the folder.

Folder Administration

This table describes the primary fields that appear on the Folder Administration page:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Name	Enter a unique identifier for the folder. **Note:** The name can consist of letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.
Label	Enter the text to appear as the link for this folder.
Copy object	Click to access the Copy Object to a Different Portal Registry page, which enables you to copy an existing folder to a different portal registry. This button appears only when you're editing a folder.
Parent Folder	The name of the current parent folder. This field is display-only.
Select New Parent Folder	Click to copy the folder to a new location within the current portal registry. This button appears only when you're editing a folder.
Long Description	Enter text to appear when the cursor moves over the link for this folder. If you leave this field blank, the label appears.
Product	(Optional) Portal-aware applications can use this field for group processing. All PeopleTools objects have the product name *PT*.
Sequence number	Enter a sequence number to define where this folder appears in its parent folder, and therefore where its associated menu entry appears within its parent menu. If you omit the Sequence number, the folder is added to the top of the list of folders and is displayed in alphabetical order. You can change the sequence only one item at a time, and each time you do so, you must save your changes to see the result.
Valid from date and Valid to date	Enter a valid from date. The valid to date is optional and is omitted by default. These dates can be used for application-specific processing by portal-aware applications and are used by the portal navigation to determine visibility.

Application Navigation

Use these fields to create SmartNavigation folders.

See Defining SmartNavigation Folders.

Folder Attributes

Folder attributes can be used by portal-aware applications for application-specific processing that adds functionality to folders.

1. Specify the attribute name.
2. (Optional) Specify the attribute label.
3. Enter the attribute value.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. Select the Translate check box to make an attribute label and attribute value translatable.

This table describes the fields that you use to define folder attributes.

Name	Enter the attribute name. The attribute must be an attribute that is recognized by the portal. If you enter an attribute, then you must also enter an attribute value.
Label	Enter text to describe the attribute. This field is required only if the Translate check box is selected.
Attribute Value	Enter the value that the portal should use in conjunction with the attribute.
Translate	Select to translate the label.

Related Links

"Using Menu Pagelets" (PeopleTools 8.54: Applications User's Guide)

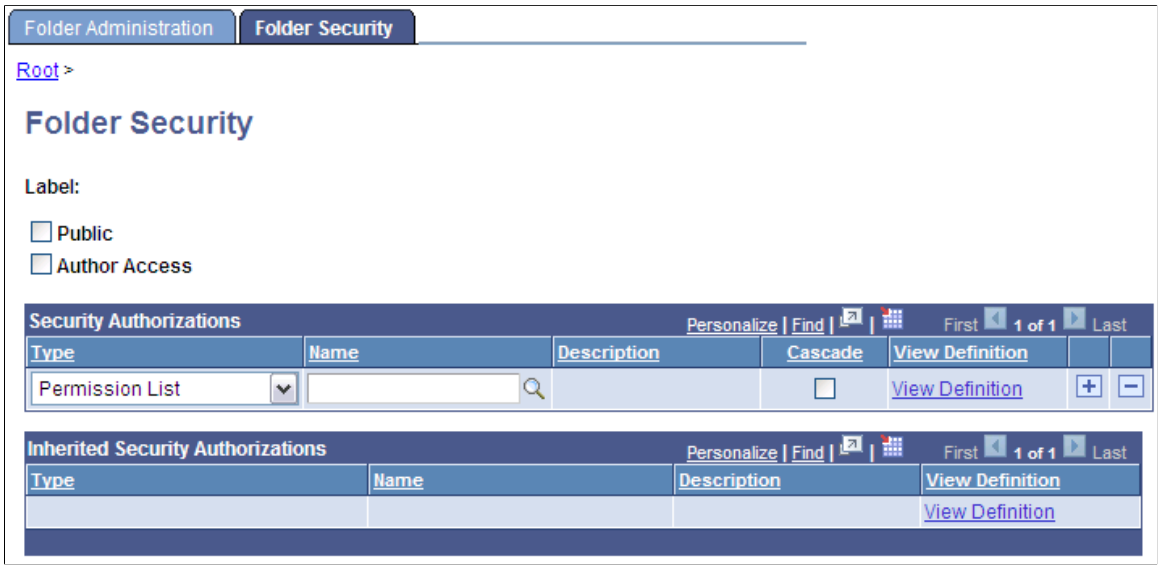
Administering Content References

Setting Folder Permissions

Access the Folder Security page (Select PeopleTools, Portal, Structure and Content. Navigate the portal registry to the appropriate folder, click the Edit link, and then select the Folder Security tab.)

Image: Folder Security page

The following screenshot illustrates how to set folder permissions on the Folder Security page.



Public	Select to make the folder available to all users. When this check box is selected, all other fields except Label are hidden.
Author Access	Select to restrict folder access to the creator of the folder, regardless of how other permissions are set. This setting is useful for troubleshooting and testing purposes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Security Authorizations

You can base noninherited folder security on roles or permission lists. You can specify any combination of roles and permission lists to provide folder security.

Type	Select from these types: *Permission List:* Enables access for users based on permission list membership. You then specify the permission lists that should have access to this folder. *Role:* Enables access for users based on role membership. You then specify the roles that should have access to this folder.
Name	Select the permission list or role that provides access to this folder.
Cascade	Select to indicate that the permission list or role of the current row should also apply to any folder or content reference that is subordinate to this folder.
View Definition	Click to access the security administration component for the permission list or role specified on the current row.

Inherited Security Authorizations

Displays authorizations that have been inherited from parent folders. The fields in this grid have the same purpose as the corresponding fields in the Security Authorizations grid. You can change inherited authorizations by modifying the Cascade setting in the Security Authorizations grid of the parent folder.

Related Links

"Understanding Permission Lists" (PeopleTools 8.54: Security Administration)

"Understanding Roles" (PeopleTools 8.54: Security Administration)

Administering SmartNavigation Folders

This section provides overviews of SmartNavigation folders; rowset, tree, and dynamic hierarchy data sources; and discusses how to:

- Define SmartNavigation folders.
- Define rowset-based SmartNavigation folders.
- Define tree-based SmartNavigation folders
- Define dynamic hierarchy folders.

Important! This topic contains code samples to guide you in using the built-in PeopleCode functions that generate SmartNavigation folders and content references. The code samples exist for illustrative purposes only; never use code samples in a production environment.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Understanding SmartNavigation Folders

SmartNavigation is a form of folder navigation, but unlike standard navigation, with SmartNavigation, you define only the root folder in the portal registry. All subordinate SmartNavigation folders and content references are generated dynamically when the user navigates to a SmartNavigation menu item. The character and appearance of SmartNavigation is also quite different from standard folder navigation because of additional folder properties that enable the portal to access and use your application data.

Note these points about SmartNavigation folders:

- When you configure SmartNavigation folders, you explicitly define the root folder only; all subordinate menu items, which reflect the data values from the data source, are generated dynamically when the user clicks the initial SmartNavigation menu item.
- The menu items in SmartNavigation menus reflect a hierarchical set of application data, which you specify as the data source for the SmartNavigation folder.
- Two data source types exist for SmartNavigation folders: rowsets and trees.
- Each SmartNavigation menu item and content reference corresponds to a data element: a row in a rowset or node in a tree.
- SmartNavigation menu hierarchy reflects the parent-child relationships that exists between rows.
- You differentiate SmartNavigation folders and content references from standard folders and content references by the images that appear beside the link. The default SmartNavigation folder and content reference images look like stylized representations of multiple entities or a single entity, respectively. You can replace these images with your own.

  See Understanding SmartNavigation Icons.
- SmartNavigation folders can appear as content references instead of folders.
- You can configure SmartNavigation to take advantage of related content services, which appear in an Actions folder in the drop-down menu hierarchy.

Important! SmartNavigation folders appear in the drop-down navigation only; they do not appear in left navigation (the Menu pagelet).

SmartNavigation Pages

When you click a SmartNavigation folder, the portal generates a navigation page; this response is comparable to standard folder navigation. The SmartNavigation page provides a characteristically different navigation experience than the standard navigation page.

Note these points about SmartNavigation pages:

- SmartNavigation pages display an organizational chart rather than folders and links.
- The character and appearance of SmartNavigation pages are controlled by PT_ABN_ORGCHART, the default chart navigation component that is standard with every PeopleTools implementation. You can clone this component and use the copy to create a different character and appearance to your SmartNavigation page or to extend and enhance the functionality of the chart.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The chart on SmartNavigation pages reflects both the content and the structure of your data. Each chart node corresponds to a data element: a row in a rowset or a node in a tree. Chart hierarchy reflects the parent-child relationships that exists between rows.
- The nodes that appear on navigation pages vary by user based on the data permission security model that is specific to your application and your implementation.
- You associate nodes with details components so that when users click the Details link, the transaction page appears.
- As is the case with standard navigation pages, breadcrumb links appear at the top of the navigation page to indicate position in the hierarchy. Click a breadcrumb link to access a higher-level node.
- You can configure SmartNavigation pages to take advantage of related content services so that related actions are accessible from each chart node. Users do not have to navigate back through a menu; the links are available in the Actions drop-down list box on the node.

Image: Example SmartNavigation page showing an organization chart

Notice the organization chart and the navigation path (breadcrumbs) on this example SmartNavigation page, which clearly indicates the hierarchical nature of this type of navigation :

See Understanding Assignment of Related Content and Related Actions.

Understanding Data Sources: Rowsets

A rowset is a data structure that represents hierarchically organized records. A scroll area and all of its subordinate scroll areas are an example of a rowset. You create PeopleCode application packages or iScripts to programmatically populate rowset objects with your application data, which the portal uses when it builds SmartNavigation folders.

Rowset Characteristics

Note these points about the rowsets that you use as SmartNavigation folder data sources:

- The rowset must be a standalone rowset because the component buffers do not contain this data.
- The rowset structure must include these subrecords in this order:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

1. PTORGNODE_SBR
2. PT_ABNNDURL_SBR
3. PT_ABNNDDTL_SBR
4. PTORGBOXFLD_SBR

- Because SmartNavigation pages display up to three levels of the organizational chart, the rowset should contain the relevant data required to display:
  - The requested node, which is determined by a user click or mouse over event.
  - The parent of the requested node.
  - The children of the requested node.
- The rowset should contain the data that the portal requires to build the hierarchy of folders in the breadcrumb trail. The breadcrumb trail should being with the defined SmartNavigation folder and be followed by the rowset root node. Next, each child node should appear in the breadcrumb trail in hierarchical order until the trail ends with either the focus node or content reference.
- The rowset should contain data that relevant to the current user only.
- For each row in the rowset, set a details component. The details component information must include these elements:
  - *MenuName*- the menu definition that contains the component.
  - *BarName*- the menu bar beneath which the component appears.
  - *ItemName* - the component name.
  - *PageName*- the page that appears to the user when the component is accessed.
  - *ActionSpecifier*- the action mode of the component. Use the single character identifier, such as U, A, and so on.
  - *SearchRecordName* - the component search record.
  - *NumberOfKeys* - the number of keys. (Use a three-digit format; for example, use 001 for one key, 010 for ten keys, and 100 for one hundred keys.)
  - *KeyFieldName* - the key field name.
  - *KeyFieldValue* - the key value.
- The details component information must be assigned to the PT_ABN_CHART_DFLD2 field value as a string that takes the following format:

Note: A comma followed by a space delimits the elements of the string.

MenuName.*MenuName,* BarName.*BarName,* ItemName.*ItemName,* Page.*PageName,* Action.*ActionSpecifier,* Record.*SearchRecordName,* KeyFieldNumber.*NumberOfKeys,*

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

FieldName.*KeyFieldName,* FieldValue.*'KeyFieldValue',* FieldName.*KeyFieldName,* FieldValue.*'KeyFieldValue'*

Important! If you deviate from the format then SmartNavigation cannot parse the details component information string.

Application Package PeopleCode Sample: Rowset

This items in this list step through a sample PeopleCode program for use as the data source of a rowset-based SmartNavigation folder. The next topic explains the PeopleCode that you see in this list.

Important! This code sample exists for illustrative purposes only. Use this sample to guide your programming; never use code samples in a production environment.

---

1. Declare the method and variables.

```
method QE_ABNRS_M

 Local Request &req = %Request;
 Local Rowset &rs_DataSource, &rs_DatabaseData;
 Local Rowset &rs_ChartRowset, &rs_DbReqNode;

 rem create related actions rowset for SmartNavigation;
 Local Rowset &rs_RelatedActions;

 rem added for app pkg;
 Local array of array of string &reqParams;
 Local string &dataKey;
 Local string &szDtlCompData;
 Local integer &i;
```

2. Get the SmartNavigation request parameters and specify the initial node of the rowset.

```
 rem Get request parameters and set initial node
 rem Binu's Org;
 &reqParams = GetABNReqParameters("BINU01");
```

3. Create the stand alone rowset data source.

```
 rem begin: create the data source rowset;

 rem retrieve complete data source;
 rem &rs_DatabaseData = CreateRowset(Record.MY_ORG_NODES);
 rem &dataKey = "PT_ORG_001";
 rem &rs_DatabaseData.Fill("Where MY_ORG_KYFLD=:1", &dataKey);
 rem &rs_DataSource = CreateRowset(Record.MY_ORG_NODE_DR);
 rem &rs_DatabaseData.CopyTo(&rs_DataSource, Record.MY_ORG_NODES, Record.MY_OR⇒
G_NODE_DR);

 rem retrieve subset of data source
 &rs_DbReqNode = CreateRowset(Record.MY_ORG_NODES);
 &rs_DatabaseData = CreateRowset(Record.MY_ORG_NODES);
 &dataKey = "PT_ORG_001";
 &rs_DbReqNode.Fill("Where  MY_ORG_KYFLD=:1 AND PTCHART_NODE=:2", &dataKey, Ge⇒
tABNNode(&reqParams));
 &rs_DatabaseData.Fill("Where  MY_ORG_KYFLD=:1 AND (PTCHART_NODE=:2 OR PTPAREN⇒
T_CHART_ND=:2 OR PTCHART_NODE=:3)", &dataKey, GetABNNode(&reqParams), &rs_DbRe⇒
qNode(1).MY_ORG_NODES.PTPARENT_CHART_ND.Value);
 &rs_DataSource = CreateRowset(Record.MY_ORG_NODE_DR);
 &rs_DatabaseData.CopyTo(&rs_DataSource, Record.MY_ORG_NODES, Record.MY_ORG_NO⇒
DE_DR);
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. Set the details component

```
 rem set details componenent for all nodes;
 &szDtlCompData = "MenuName.MY_MENU, BarName.USE, ItemName.EMPLOYEE, Page.EMPL⇒
OYEE, Action.U, Record.EMPLOYEE, KeyFieldNumber.001, FieldName.EMPLID, FieldVa⇒
lue.";

 For &i = 1 To &rs_DatabaseData.ActiveRowCount

  &rs_DataSource(&i).QE_ABNRS_DS.PT_ABN_CHART_DFLD2.Value = &szDtlCompData | &⇒
rs_DataSource(&i).QE_ABNRS_DS.PTCHART_NODE.Value;

 End-For;

 rem end: create the data source;
```

5. Get the chart rowset.

```
 rem get the chart rowset returns the chart rowset;
 &rs_ChartRowset = GetABNChartRowSet();
```

6. Get the SmartNavigation related actions rowset.

```
 rem  gets  related action rowset from the chart component;
 &rs_RelatedActions = GetABNRelActnRowSet();
```

7. Generate the HTML for the SmartNavigation menu.

```
 rem generate the html fragment for the menu;
 rem GenHTMLMenu(&rs_DataSource, GetABNNode(&reqParams), GetABNInitialNode(&re⇒
qParams));
```

8. Load the chart rowset.

```
 rem load the chart rowset from the data source rowset;
 LoadABN(&rs_DataSource, &rs_ChartRowset, &rs_RelatedActions, GetABNNode(&reqP⇒
arams), GetABNInitialNode(&reqParams));


end-method;
```

Application Package PeopleCode Sample Explanation: Rowset

Each item in this list explains the corresponding list item in the application package PeopleCode sample program.

1. Declare the function and variables.

2. Get the SmartNavigation request parameters and specify the initial node of the rowset.

   - Retrieves a two dimensional array, or array of arrays, of name/value pairs.
   - The name/value pairs are request parameters needed by SmartNavigation functionality.
   - The input parameter is the identifier of the initial node .

3. Create the stand alone rowset data source.

   - The rowset structure is based on the record MY_ORG_NODES, which includes the necessary subrecords.
   - This table describes the variables that are used to store data in the functions:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Parameter	*Description*
&rs_dbReqNode	Rowset that contains the data for the requested node.
&rs_DatabaseData	Rowset that contains the data for the requested node, parent node, child nodes.
&rs_dataKey	Variable that contains the high level key.
&rs_DataSource	Rowset that contains the entire data set that is used by SmartNavigation chart.

- Retrieve the data for the requested node (&rs_DbReqNode.Fill).
- Retrieve the data for the parent and child nodes of the requested node (&rs_DatabaseData.Fill).
- Copy the data from all nodes to the chart rowset (&rs_DatabaseData.CopyTo)

4. Set the details component.
   a. Set the name of the details component.
   b. Loop through all of the nodes and insert the component string into the PT_ABN_CHART_DFLD2 field.

5. Get the chart rowset.
   - This function returns Null if triggered by a mouse over event; returns the rowset if triggered by an on click event at a menu element or the first link of a chart node.
   - Upon retrieval of the related actions rowset, the chart rowset is flushed. So, this function be called one time only in the iScript program.
   - Retrieval of the related action rowset allows developers to access and modify the chart rowset beyond the PeopleTools' functionality.

6. Get the SmartNavigation related actions rowset.
   - This function returns Null if triggered by an on click event of the menu item (folder) label; returns the rowset if triggered by an on click event of the menu item (folder) image or the first link of a chart node.
   - Upon retrieval of the related actions rowset, the chart rowset is flushed. So, this function be called one time only in the iScript program.
   - Retrieval of the related action rowset allows developers to access and modify the chart rowset beyond the PeopleTools' functionality.

7. Generate the HTML for the SmartNavigation menu.
8. Load the chart rowset from the data source rowset.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- This table describes the parameters for the function:

Parameter	*Description*
&rs_DataSource	Data source rowset retrieved in item 4 in this list.
&rs_ChartRowset	Chart rowset retrieved in item 6 in this list.
&rs_RelatedActions	Related actions rowset retrieved in item 7 in this list.
&reqParams	The requested parameters retrieved in item 2 in this list.
GetABNNode(&reqParams)	The requested node based on the user on click event of a menu item (folder) image or the on click event of the first link of a chart node.
GetABNInitialNode(&reqParams)	The initial node set in item 2 in this list

- End the method.

Related Links

"GetABNChartRowSet" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNInitialNode" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNNode" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNRelActnRowSet" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNReqParameters" (PeopleTools 8.54: PeopleCode Language Reference)
"GenHTMLMenu" (PeopleTools 8.54: PeopleCode Language Reference)
"LoadABN" (PeopleTools 8.54: PeopleCode Language Reference)

Understanding Data Sources: Trees

Trees are hierarchically-organized structures that represent a group of summarization rules for a particular database field. For example, a tree can specify how your manufacturing locations should be summarized for reporting purposes. Or a tree can show the reporting relationships within an organization by specifying how the individual department should be summarized into territories, territories into regions, and regions into countries. Similarly, a tree can categorize items in a catalog.

Tree Characteristics

Note these points about trees that you use as SmartNavigation folder data sources:

- Trees must be valid and active. Inactive, draft, and unaudited trees render unexpected results.
- SmartNavigation passes the values of several tree-specific fields to the application via URL. Certain characters are inappropriate for use in a URL and should be avoided. When you set up trees, do not use any of the following characters in the Tree Name, SetID, Set Control Value, and Tree Branch fields: pound (#). percent (%). dollar ($). ampersand (&). plus (+). comma (,). forward slash/virgule (/). colon (:). semi-colon (;). equals (=). question mark (?). at symbol (@). space ( ), quotation marks("), less than symbol (<), greater than symbol (>), left curly brace ({), right curly brace (}),

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

vertical bar/pipe (|), backslash (\), caret (^), tilde (~), left square bracket ([), right square bracket (]), and grave accent (`).

- The root node of the tree appears as the initial node of the SmartNavigation page unless you use PeopleCode to specify another initial node.
- For tree data sources displayed in a SmartNavigation chart that uses the default PT_ABN_ORGCHART component, the tree detail content is displayed in a modal window.

iScript PeopleCode Sample: Tree

This items in this list step through a sample PeopleCode program for use as the data source of a tree-based SmartNavigation folder. The next topic explains the PeopleCode that you see in this list.

Important! This code sample exists for illustrative purposes only. Use this sample to guide your programming; never use code samples in a production environment.

---

1. Declare the function and variables.

```
Function IScript_PT_ABN()

 Local Request &req = %Request;
 Local string &szLI;
 Local ApiObject &Session = %Session;
 Local ApiObject &MyTree, &MyNode, &ChildNode, &ChildLeaf;
 Local Rowset &rs, &rars;
```

2. Retrieve the chart rowset.

```
 rem get the chart rowset;
 &rs = GetABNChartRowSet();
```

3. Retrieve the related actions rowset from the chart component.

```
 rem retrieve the related action rowset from the chart component;
 &rars = GetABNRelActnRowSet();
```

4. Get the SmartNavigation request parameters and specify the initial node of the tree.

```
 rem get the request parameters and specify the initial node or not;
 &reqParams = GetABNReqParameters("00001");
```

5. Create an empty tree from session object and open the tree based on the request parameters.

```
 /* get an empty tree from the session object */
 &MyTree = &Session.GetTree();

 /* open the tree based on the request parameters */
 &TreeReturn = &MyTree.Open(GetABNTreeSetid(&reqParams), ˙ ˘GetABNTreeUserKey(⇒
&reqParams), GetABNTreeName(&reqParams), GetABNTreeEffdt(&reqParams), "", True⇒
);
```

6. Retrieve the requested tree node.

```
 /* get the requested tree node */
 &MyNode = &MyTree.FindNode(GetABNNode(&reqParams), "");
```

7. Load the requested node into the chart rowset.

```
 rem load the requested node into the chart and track the starting node;
 &MyNode.LoadABNChart(&rs, &rars, True, GetABNInitialNode(&reqParams));
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

8. Generate the related actions of the requested tree node.

```
rem generate the requested node's related actions;
&szLI = &szLI | &MyNode.GenRelatedActions();
```

9. Generate the SmartNavigation folder elements to display in the menu (based on some optional criteria).

```
/* begin-filter
based on some criteria, display only the child nodes and leaves of the
requested tree node that meet the criteria */

&nNodeOrder = &MyNode.allChildCount;
If &MyNode.HasChildNodes Then
 /* get the first child node */
 &ChildNode = &MyNode.FirstChildNode;

 /* generate the <li> tag that the portal consumes for this node */
 &szLI = &szLI | &ChildNode.GenABNMenuElement(GetABNInitialNode(&reqParams));
```

10. Load the child node into the chart rowset buffer.

```
 &ChildNode.LoadABNChart(&rs, &rars, False, GetABNInitialNode(&reqParams));
End-If;
If &MyNode.HasChildLeaves Then
  If &MyNode.ChildLeafCount >= 3 Then
   &MaxCount = 3;
  Else
   &MaxCount = &MyNode.ChildLeafCount;
  End-If;

  For &i = 1 To &MaxCount
   If &i = 1 Then
    /* get the first child leaf */
    &ChildLeaf = &MyNode.FirstChildLeaf;
   Else
    /* get the next child leaf */
    &ChildLeaf = &ChildLeaf.NextSib;
   End-If;
```

11. Generate the SmartNavigation content reference elements to display in the menu.

```
   /* generate the <li> tag that is consumed by the portal for this leaf */
   &szLI = &szLI | &ChildLeaf.GenABNMenuElement();
```

12. Load a child node of the requested tree node and its related actions into the chart buffer.

```
   rem load the  child leaf of requested node into the chart;
   &ChildLeaf.LoadABNChart(&rs, &rars);
  End-For;

 End-If;

/* end-filter */
```

13. Generate the entire HTML that the portal consumes to create the SmartNavigation menu.

```
 /* generate the html fragment required by the portal to build menu structure⇒
*/
 GenHTMLMenu(&szLI);
```

14. Close the tree and end the function.

```
 &MyTree.Close();
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
End-Function;
```

iScript PeopleCode Sample Explanation: Tree

Each item in this list explains the corresponding list item in the sample Tree PeopleCode program.

1. Declare Function and variables.
2. Retrieve the chart rowset.
   - This function returns Null if triggered by an on click event of the menu item label; returns the rowset if triggered by an on click event of the menu item image or the first link of a chart node.
   - The rowset must be retrieved because the tree API doesn't have access, or know of the existence of the panel processor.
   - The rowset must be passed to the tree api so that the tree API can properly load the chart rowset.
   - Upon retrieval of the chart rowset, the chart rowset is flushed. So, function should be called one time only in the iScript program.
   - Retrieval of the chart rowset enables developers to access and modify the chart rowset beyond the PeopleTools' functionality.
   - If you use a chart component other than the PeopleTools-delivered chart component, the component buffer structure must match the PeopleTools-delivered template so that this function can find the correct rowset. The chart rowset must be at level 1 and the first scroll at level 1.
3. Retrieve the related actions rowset from the chart component buffer.
   - This function returns Null if triggered by a mouse over event; returns the rowset if triggered by an on click event at a menu element or the first link of a chart node.
   - Upon retrieval of the related actions rowset, the chart rowset is flushed. So, this function be called one time only in the iScript program.
   - Retrieval of the related action rowset allows developers to access and modify the chart rowset beyond the PeopleTools' functionality.
4. Get the SmartNavigation request parameters and specify the initial node of the tree.
   - Retrieves a two dimensional array, or array of arrays, of name/value pairs.
   - The name/value pairs are request parameters needed by SmartNavigation functionality.
   - The input parameter is the initial node identifier.
   - If no input parameter is passed, the initial node is the root node of the tree.
5. Create an empty tree from session object and open the tree based on the request parameters.
   - Create an empty tree session object.
   - Open the tree using the request parameters. This table describes the tree parameters:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Tree Parameter	*Description*
GetABNTreeSetid(&reqParams)	Retrieves the SetId key value from the input parameter
GetABNTreeUserKey(&reqParams)	Retrieves the tree user key value from the input parameter
GetABNTreeUserKey(&reqParams)	Retrieves the SetId key value from the input parameter
GetABNTreeEffdt(&reqParams)	Retrieves the SetId key value from the input parameter

Note: For the preceding table, &reqParams is the output of item 4 in this list.

6. Retrieve the requested tree node.
   - The requested node is the menu element upon which the user is focused during the mouse over or click event.
   - The SmartNavigation function retrieves the node identifier from the input parameter, which is the output of item two in this list.
7. Load the requested node into the chart rowset.
   - Load the requested node from item six in this list into the chart rowset buffer.
   - If necessary, load the parent node into the chart buffer.
   - Determine the initial node by retrieving its identifier from request parameters set in item three in this list.
   - This table describes the function parameters:

Parameter	*Description*
&rs	Chart component rowset retrieved from item 2 in this list.
&rars	Related action rowset retrieved by item 3 in this list.
Boolean	Boolean which, if true, directs the function to load the node's parent so that the correct number of chart levels are displayed.
GetABNInitialNode(&reqParams)	Retrieves the identifier tree initial node from the input parameter. The value of the input parameter is the output of item three in this list. You must keep track of the starting node so that PeopleTools can navigate correctly.

8. Generate the related actions elements to display in the menu.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Generate the <li> element for related actions. In the SmartNavigation menu, these elements appear as the folder and subordinate actions for the requested tree node.
- The output of the function is concatenated to the current string of <li> elements, &szLI, to create a list of <li> elements.
- The list of <li> elements is used to generate an HTML fragment that the portal consumes.

9. Generate the SmartNavigation folder elements to display in the menu (based on some optional criteria).
   - Filter the nodes that you want to display and order them
   - Generate an <li> element for a tree node. In the SmartNavigation menu, this element appears as a folder.
   - The output of the function is concatenated to the current string of <li> elements, &szLI, to create a list of <li> elements.
   - The list of <li> elements is used to generate an HTML fragment that the portal consumes.
   - This table describes the function parameters:

Parameter	***Description***
&rs	Chart component rowset retrieved from item 2 in this list.
&nNodeOrder	The display order of the leaf in the chart.
&szLI	The list of items that appear in the menu.
GetABNInitialNode(&reqParams)	Retrieves the identifier tree initial node from the input parameter. The value of the input parameter is the output of item three in this list.

10. Load the child node into the chart rowset buffer.
    - Load the child node from item 2 in this list into the chart buffer.
    - Load the related actions rowset from item 3 in this list into the chart buffer.
    - Determine the initial node by retrieving its identifier from request parameters set in item 4 in this list.
    - This table describes the function parameters:

Parameter	***Description***
&rs	Chart component rowset retrieved from item 2 in this list.
&rars	Related actions rowset retrieved from item 3 in this list.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Parameter	*Description*
Boolean	Boolean which, if false, directs the function to load the calling node into the chart buffer. Because the requested node is loaded in the buffer already, select false.
GetABNInitialNode(&reqParams)	Retrieves the identifier of the initial node from the input parameter. The value of the input parameter is the output of item 3 in this list.

11. Generate the SmartNavigation content reference elements to display in the menu.
    - Generate an <li> element for a tree leaf. In the SmartNavigation menu, this element appears as a content reference.
    - The output of the function is concatenated to the current string of <li> elements, &szLI, to create a list of <li> elements.
    - The list of <li> elements is used to generate an HTML fragment that the portal consumes.
12. Load a child leaf into the chart rowset buffer.
    - Load the child leaf from item 2 in this list into the chart buffer.
    - Load the related actions of the child leaf from item 3 in this list into the chart buffer.
13. Generate the HTML that creates the SmartNavigation menu.

    Generate the HTML fragment using the list of <li> elements from items 9, 10, and 11 in this list.
14. Close the tree and end the function.

Related Links

"GetABNChartRowSet" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeEffdt" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeName" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeSetid" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeUserKey" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNInitialNode" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNNode" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNRelActnRowSet" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNReqParameters" (PeopleTools 8.54: PeopleCode Language Reference)
"GenHTMLMenu" (PeopleTools 8.54: PeopleCode Language Reference)
"LoadABN" (PeopleTools 8.54: PeopleCode Language Reference)

Understanding Data Sources: Dynamic Hierarchy

A dynamic hierarchy is a data source that enables you to group SmartNavigation folders within a folder. You define the dynamic hierarchy by creating an application package PeopleCode program that you reference from the defined dynamic hierarchy folder. All of the properties of the subordinate

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

SmartNavigation folders, such as data source , must be defined programmatically in an application package PeopleCode program. Then, the PeopleCode generates the enclosing SmartNavigation folder elements of each tree or rowset so that groups of SmartNavigation folders appear under one menu.

With other SmartNavigation folders, permission lists and roles determine whether the SmartNavigation folder and the underlying details component are visible and accessible to users. Dynamic hierarchy folders, through the use of application package PeopleCode, enable you to add another element to the security mechanism for the programmatically defined SmartNavigation folders. For example, you could use a dynamic hierarchy folder to display project data for projects that are assigned to the current user only.

Dynamic Hierarchy Characteristics

Note these points about the dynamic hierarchy data source:

- The dynamic hierarchy folder must specify an application package PeopleCode program.
- The application package PeopleCode must generate a string of <li> elements, each of which a programmatically-defined SmartNavigation folder that will appear as a subordinate menu item in the dynamic hierarchy folder.
- These subordinate menu items are logical SmartNavigation folders; they are not defined in the portal registry.
- The application package PeopleCode then generates an HTML fragment that the portal consumes; this HTML creates the SmartNavigation menu structures..

Application Package PeopleCode Sample: Dynamic Hierarchy

This items in this list step through a sample PeopleCode program for use as the data source of a dynamic hierarchy folder. The next topic explains the PeopleCode that you see in this list.

Important! This code sample exists for illustrative purposes only. Use this sample to guide your programming; never use code samples in a production environment.

---

1. Declare the method within the appropriate class.

```
class TEST_ABN_DYN
 method GenerateDynFlyout();
end-class;


method GenerateDynFlyout
```

2. Declare the variables.

```
 Local string &flderList;
 Local string &ds_t, &cref_t, &id_t, &label_t, &portal, &node, &chart, &pcode_⇒
t, &tree_name, &tree_setid, &tree_userkey, &tree_effdt, &tree_branch;
 Local string &ds_r, &cref_r, &id_r, &label_r, &pcode_r;
```

3. Provide the values for the tree-based SmartNavigation folder as variable assignments.

```
 rem tree information;
 &ds_t = "t";
 &cref_t = "f";
 &label_t = "New Dynamic ABN Tree";
 &chart = "PT_ABN_CHART.PT_ABN_ORGCHART.GBL";
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
 &pcode_t = "QE_ABNTREE.qe_abntree.QE_ABN_TREE_AP";
 &tree_name = "QE_PERS_DATA";
 &tree_setid = "QEDM1";
 &tree_userkey = "";
 &tree_effdt = "1997/05/05";
 &tree_branch = "";
```

4. Create the list of tree-based SmartNavigation folders.

```
 rem create the <li> element for the tree-based SmartNav folder;
 &flderList = GenDynABNElement(&ds_t, &cref_t, &fldr_id, &label_t, &chart, &pc⇒
ode_t, &tree_name, &tree_setid, &tree_userkey, &tree_effdt, &tree_branch);
```

5. Provide the values for the rowset-based SmartNavigation folder.

```
 rem rowset information;
 &ds_r = "r";
 &cref_r = "f";
 &label_r = "Dynamic ABN Rowset";
 &chart = "PT_ABN_CHART.PT_ABN_ORGCHART.GBL
 &pcode_r = "QE_ABN_RS_APKG.QE_ABN_RS_1:QE_ABNRS_C1.QE_ABNRS_M";
```

6. Create the list of rowset-based SmartNavigation folders.

```
 rem create the <li> element for the smartnav rowset folder and append;
 &flderList = &flderList | GenDynABNElement(&ds_r, &cref_r, &label_r, &chart, ⇒
&pcode_r);
```

7. Generate the HTML that the portal uses to construct the menu navigation.

```
 rem this function already exists;
 GenHTMLMenu(&flderList);

end-method;
```

Application Package PeopleCode Sample Explanation: Dynamic Hierarchy

Each item in this list explains the corresponding list item in the Application Package PeopleCode Sample: Dynamic Hierarchy:

1. Declare the method in the appropriate class.
2. Declare the variables.
   - Declare the necessary local variables.
   - All variables in the sample program are strings.
3. Provide the values for the tree-based SmartNavigation folder as variable assignments.

   This table describes the tree parameters; you use these parameters in the next step.

Parameter	***Description***
&ds_t = "*data source type*"	Data source type: t, tree; r, rowset
&cref_t ="*flag value*"	Display as content reference flag: true, content reference; false, folder

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Parameter	Description
&fldr_id = " *folder ID*"	Folder ID
&label_t = "*menu folder label*"	Folder label
&chart = "*chart component*"	Chart component
&pcode_t = "*PeopleCode location*"	PeopleCode ID: (*package*:*path*:*method*) or (*weblib_record.event.function_name*)
&tree_name = "*tree name*"	Tree name
&tree_setid = "*setID value*"	Tree setID
&tree_userkey = "*key value*"	Tree user key
&tree_effdt = "*EFFDT value*"	Tree effective date
&tree_branch = "*branch value*"	Tree branch

4. Create the list of tree-based SmartNavigation folders.
   - This function creates the tree-based SmartNavigation folders based on the parameter values you provide.
   - The parameters are the same as the Application Navigation properties that you supply on the Folder Administration page to create a SmartNavigation folder.

     See Defining SmartNavigation Folders.
   - The output is a string composed of <li> elements for the tree-based SmartNavigation folders.

5. Provide the values for the rowset-based SmartNavigation folder as variable assignments.

   This table describes the rowset parameters; you use these parameters in the next step.

Parameter	Description
&ds_r = "*data source type*"	Data source type: t, tree; r, rowset
&cref_r ="*flag value*"	Display as content reference flag: true, content reference; false, folder
&fldr_id = " *folder ID*"	Folder ID
&label_r = "*menu folder label*"	Folder label
&chart = "*chart component*"	Chart component

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Parameter	Description
&pcode_r = "*PeopleCode location*"	PeopleCode ID: (*package*:*path*:*method*) or (*weblib_record.event.function_name*)

6. Create the list of rowset-based SmartNavigation folders.
   - This function creates the rowset-based SmartNavigation folders based on the parameter values you provide.
   - The parameters are the same as the Application Navigation properties that you supply on the Folder Administration page to create a SmartNavigation folder.

     See Defining SmartNavigation Folders.
   - The output is a string composed of <li> elements of the tree-based SmartNavigation elements from item 4 in this list concatenated with the string composed of <li> elements for the rowset-based SmartNavigation folders.

7. Generate the HTML that the portal uses to construct the menu navigation.
   - The dynamic hierarchy folder becomes the top level menu and each programmatically-defined SmartNavigation folder appears as a subordinate menu item in the list of folders.
   - The input parameter is the list of <li> elements.
   - The output is an HTML fragment.

8. Load the chart rowset.
   - Load the chart rowset from the data source rowset.
   - Load the related actions rowset.
   - Set the initial chart node.
   - End the method.

Related Links

Understanding SmartNavigation Icons
"GenDynABNElement" (PeopleTools 8.54: PeopleCode Language Reference)
"GenHTMLMenu" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNChartRowSet" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeEffdt" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeName" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeSetid" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNTreeUserKey" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNInitialNode" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNNode" (PeopleTools 8.54: PeopleCode Language Reference)
"GetABNRelActnRowSet" (PeopleTools 8.54: PeopleCode Language Reference)

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

"GetABNReqParameters" (PeopleTools 8.54: PeopleCode Language Reference)
"LoadABN" (PeopleTools 8.54: PeopleCode Language Reference)

Common Elements Used In This Section

This section discusses the fields that identify the different data sources that you use with SmartNavigation folders.

Application Navigation

This table describes the fields that appear in the Application Navigation group box:

Data Source

Specify the type of data source that the system uses to generate the SmartNavigation folder and page elements.

Select *Dynamic Hierarchy, Rowset,* or *Tree.*

When you select a data source, the system enables and selects Default Chart Navigation Pagecheck box, and displays the PeopleCode Type field.

Display as Content Reference

Select to replace the SmartNavigation folder with a content reference link. If you select this check box, no subordinate menu items appear when the user clicks the link. The system immediately opens the SmartNavigation page to the initial node in the target area.

This field is enabled for rowset and tree-based folders. This field is hidden for dynamic hierarchy folders.

Note: This property applies to the root-level SmartNavigation folder only. System-generated drop-down navigation subfolders are not affected by this setting.

Node Name

Select the node where the content resides.

PeopleCode Type

This field appears when you specify a data source.

Select *Application Package* to enable the Application Package Parameters group box.

Select *iScript* to enable the iScript Parameters group box

Note: Use the group box that corresponds to the PeopleCode type to identify the PeopleCode program.

Style Sheet

Enter the style sheet definition that replaces the standard folder and content reference icons in the SmartNavigation menus.

Application Package Parameters

The Application Package Parameters group box appears when any of these conditions are met:

- *Dynamic Hierarchy* is selected as a data source.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Rowset* is selected as the data source and *Application Package* is selected as the PeopleCode type.
- *Tree* is selected as the data source and *Application Package* is selected as the PeopleCode type.

This table describes the fields that appear in the Application Package Parameters group box

Application Package	Enter the PeopleCode application class name where the application class method is defined.
Application Package Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method. Beginning and ending colons are *not* required. For example, the path could be *sub_package:sub_package:app_class* or simply *app_class* if no sub application packages exist.
Application Package Method	Enter the name of the specific method that builds the rowset that populates the navigation page in the target area.

iScript Parameters

The iScript Parameters group box appears when any one of these conditions are met:

- *Tree* as a data source and *iScript* as the PeopleCode type.
- *Rowset* as the data source and *iScript* as the PeopleCode type.

This table describes the fields appear in the iScript Parameters group box:

Record (Table) Name	Specify the record definition that contains the iScript that defines the rowset.
Field Name	Specify the record field that contains the iScript that defines the rowset.
PeopleCode Event Name	Specify the event that triggers the iScript.
PeopleCode Function Name	Specify the function in the iScript that contains the business logic.

Tree Properties

The Tree Properties group box appears when you select *Tree* as the data source.

This table describes the fields that appear in the Tree Properties group box:

Tree Name	Select the name of the tree that the portal is to use to render the menu.
SetID	Enter the SetID for the tree.
Set Control Value	Enter the set control value for the tree.
Effective Date	(Optional) Enter the effective date for the tree.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

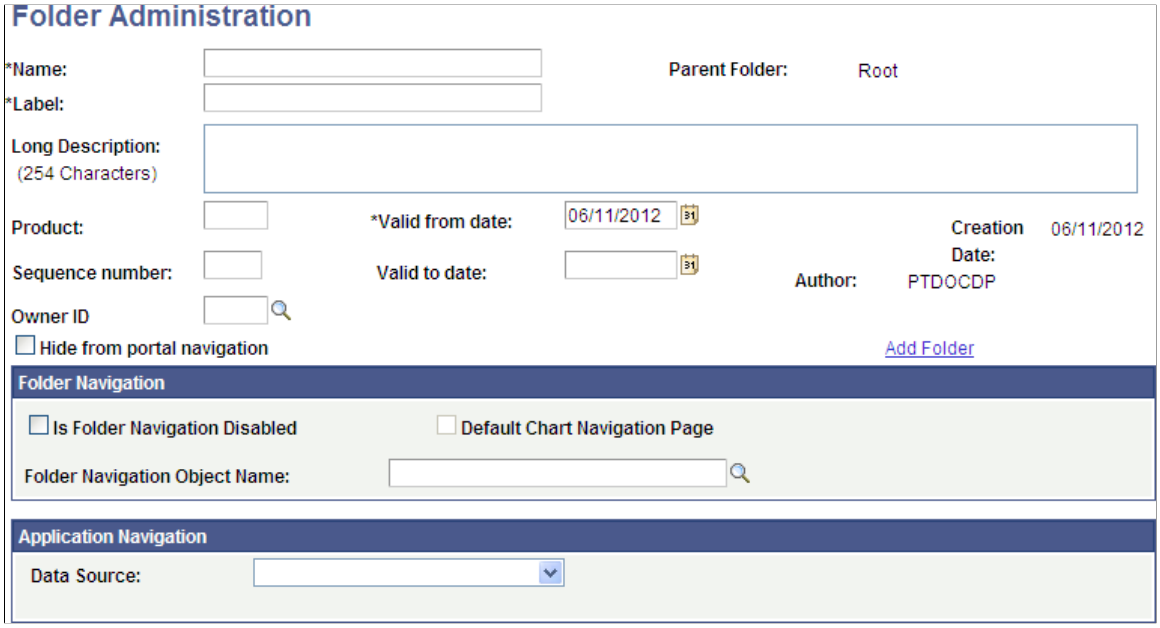
Tree Branch (Optional) Enter the tree branch for the tree.

Defining SmartNavigation Folders

Access the Folder Administration page (Select PeopleTools, Portal, Structure and Content. Click the Add Folder link).

Image: Folder Administration page without a SmartNavigation data source

In this example, the Folder Administration page shows basic folder attributes prior to selecting a SmartNavigation data source.



To configure SmartNavigation folders:

1. Enter a name, label and other identifying information.
2. In the Application Navigation group box, select a data source and define the data source parameters.

   See Defining Dynamic Hierarchy Folders.

   See Defining Rowset-Based SmartNavigation Folders.

   See Defining Tree-Based SmartNavigation Folders.
3. (Optional) To use a nondefault chart navigation page, deselect the Default Chart Navigation Page check box and enter a value in the Folder Navigation Object Name field.

   Important! If the node is a *remote* content site, then you must clear the Default Chart Navigation Page field and then enter a *remote* content reference in the Folder Navigation Object field. Otherwise, the portal cannot generate the correct URL for the SmartNavigation folder. You also must enable single signon.

   ---
4. Set folder permissions.

   See Setting Folder Permissions.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

5. Save the folder.

Related Links

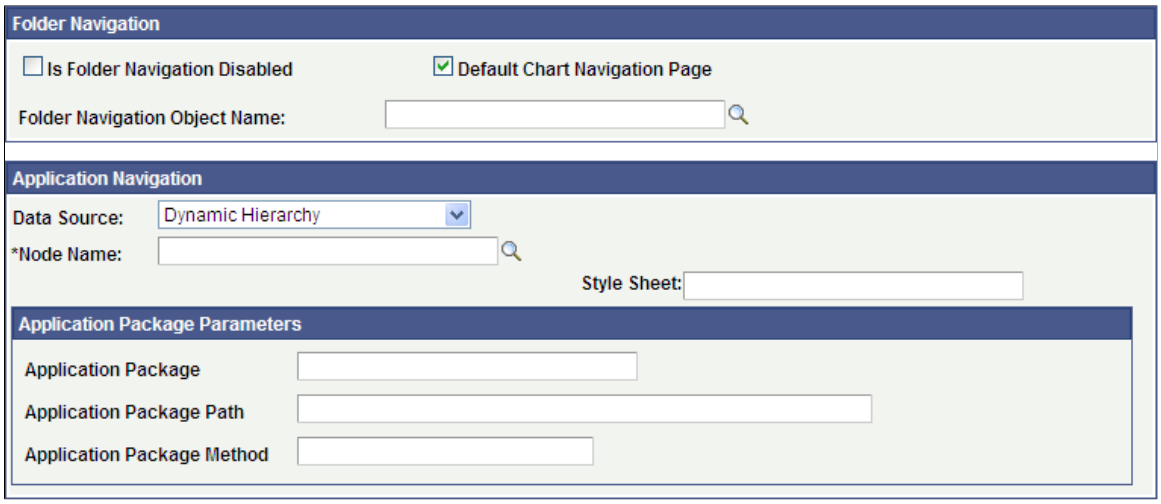
"Understanding Single Signon" (PeopleTools 8.54: Security Administration)

Defining Dynamic Hierarchy Folders

Access the Folder Administration page (Select PeopleTools, Portal, Structure and Content Click the Add Folder link).

Image: Folder Administration page showing fields necessary to create a dynamic hierarchy folder

In this example, the Folder Administration page shows SmartNavigation folder options for a dynamic hierarchy data source.



To define dynamic hierarchy folders:

1. Enter a name, label, and any other identifying information for the folder.
2. Select *Dynamic Hierarchy* as the data source.
3. Specify the content site node.
4. Enter the parameters of the application package PeopleCode program that serves as the data source for this folder.

   See Understanding Data Sources: Dynamic Hierarchy.

5. (Optional) Enter the style sheet that provides alternate images for SmartNavigation menu items.

   See Understanding SmartNavigation Icons.

6. (Optional) Specify any necessary folder attributes.
7. Set folder permissions.

   See Setting Folder Permissions.

8. Save the folder definition.

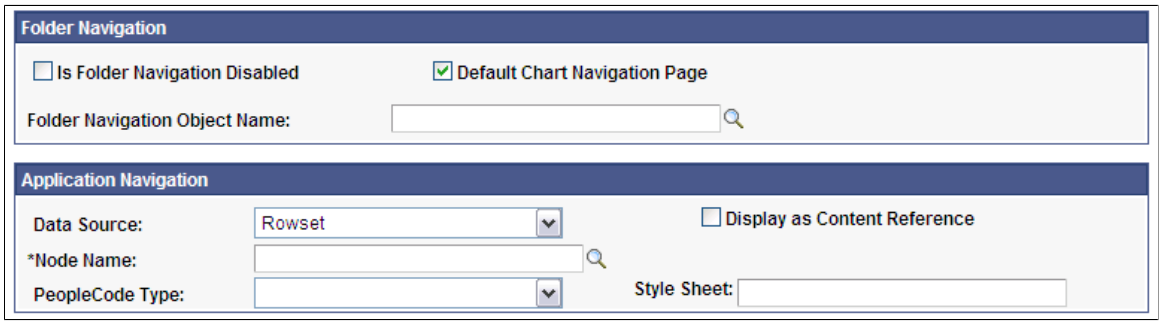
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Rowset-Based SmartNavigation Folders

Access the Folder Administration page (Select PeopleTools, Portal, Structure and Content Click the Add Folder link).

Image: Folder Administration page showing fields necessary to create a rowset-based SmartNavigation folder

In the following screenshot, the folder administration page shows SmartNavigation folder options for a rowset data source.



To define rowset-based SmartNavigation folders:

1. Enter a name, label, and any other identifying information for the folder.
2. Select *Rowset* as the data source.
3. Specify the content site node.
4. In the PeopleCode Type field, select which type of PeopleCode program defines the data source. Select *Application Package* or *iScript.*
5. Enter either the iScript or application package parameters that define the rowset.

   Note: If you use an iScript, you must grant permissions to the web library and iScript for users who will access the SmartNavigation folder.

   See "Setting Web Library Permissions" (PeopleTools 8.54: Security Administration).
6. (Optional) Select Display as Content Reference so that the SmartNavigation folder appears and functions as a content reference instead of a folder.
7. (Optional) Deselect the Default Chart Navigation Page check box and select a custom folder navigation object.

   Complete this step only if you want to replace the SmartNavigation chart page with a custom navigation page.
8. (Optional) Specify any necessary folder attributes.
9. Set folder permissions.

   See Setting Folder Permissions.
10. Save the folder definition.

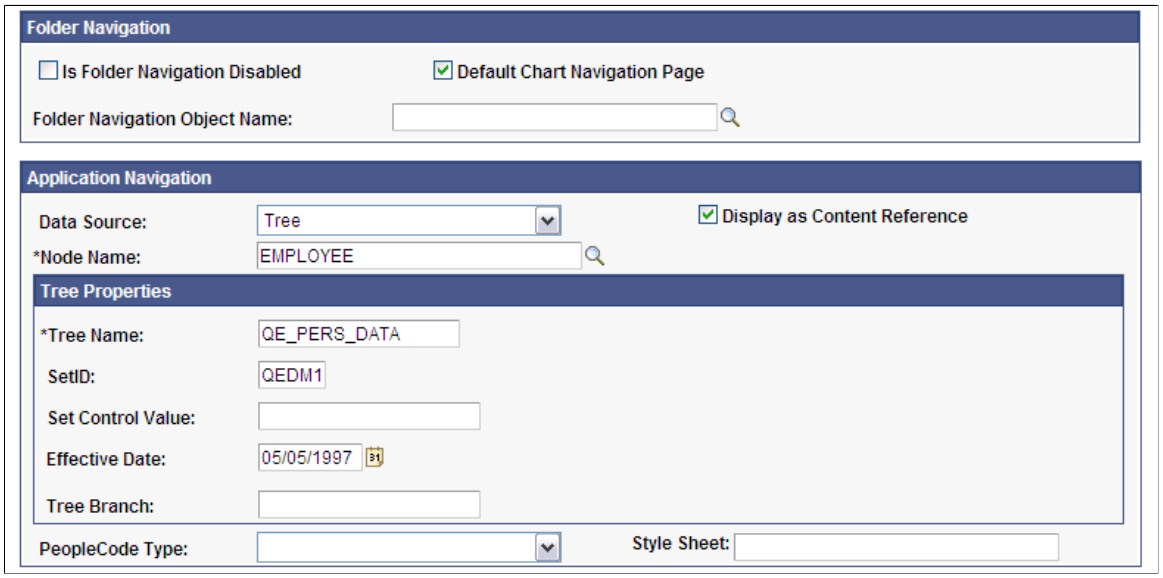
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Tree-Based SmartNavigation Folders

Access the Folder Administration page (Select PeopleTools, Portal, Structure and Content Click the Add Folder link).

Image: Folder Administration page showing fields necessary to create a tree-based SmartNavigation folder

In the following screenshot, the folder administration page shows SmartNavigation folder options for a tree data source.



1. Enter a name, label, and any other identifying information for the folder.
2. Select *Tree* as the data source.
3. Specify the content site node.
4. In the PeopleCode Type field, select which type of PeopleCode program defines the data source. Select *Application Package* or *iScript.*
5. Enter either the iScript or application package parameters that define the rowset.

   Note: If you use an iScript, you must grant permissions to the web library and iScript for users who will access the SmartNavigation folder.

   See "Setting Web Library Permissions" (PeopleTools 8.54: Security Administration).
6. (Optional) Select Display as Content Reference so that the SmartNavigation folder appears and functions as a content reference instead of a folder.
7. (Optional) Deselect the Default Chart Navigation Page check box and select a custom folder navigation object.

   Complete this step only if you want to replace the SmartNavigation chart page with a custom navigation page.
8. (Optional) Specify any necessary folder attributes.
9. Set folder permissions.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Setting Folder Permissions.

10. Save the folder definition.

Administering Content References

This section discusses how to:

- Define content references.
- Create related links.
- Set content reference permissions.
- Review content reference permissions.
- Test content references.
- Set fluid attributes for content references.
- Define content reference links.
- Set content reference link security.

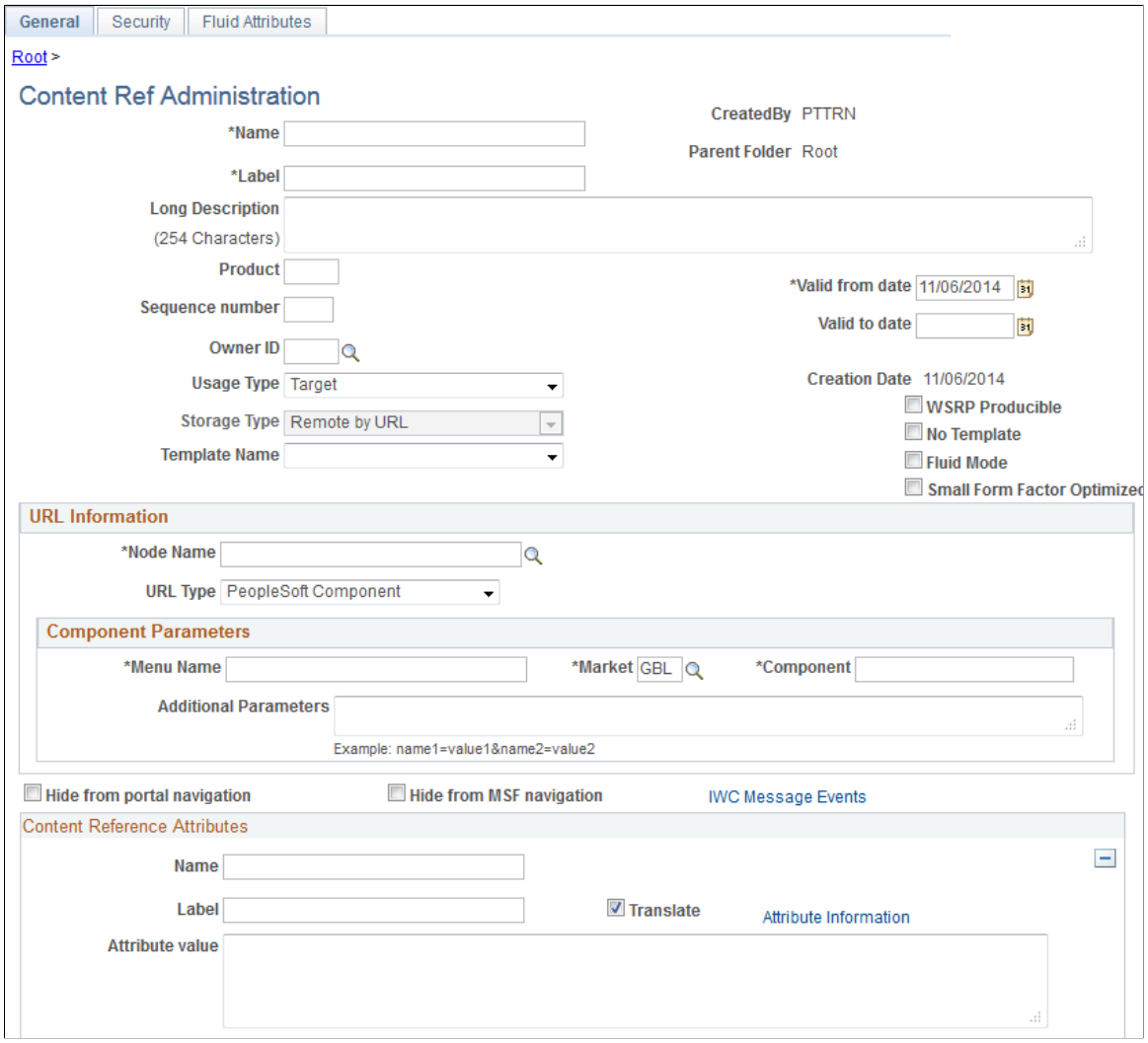
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Content References

Access the Content Ref Administration page (PeopleTools, Portal, Structure and Content. Navigate the portal registry hierarchy until you are in the parent folder for the content reference; then click the link for an existing content reference or click the Add Content Reference link.)

Image: Content Ref Administration page

This example illustrates the fields and controls on the Content Ref Administration page. You can find definitions for the fields and controls later on this page.



Note: Many of the fields described in this section appear only when you enter certain field values.

Many of these fields are similar to fields on the Folder Administration page.

When you save a content reference that points to a PeopleSoft component or script on the default local node, the security settings for the content reference are inherited automatically from the menu or script security settings. Any permission lists or roles that you enter on the Content Reference Security page before saving the content reference are deleted and replaced by those of the component or script. These new security settings are also propagated up the folder hierarchy. This ensures that the content reference is visible and all folders are viewable, enabling users to navigate to the component.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Warning! Do not add internal component-based PeopleSoft content references using the Content Ref Administration page. Use the registration wizard instead so that information is consistent between PeopleSoft Application Designer and the portal. Any changes made to the menu definition in PeopleSoft Application Designer and not made to the portal content reference definition break the content reference. The portal navigation system uses the portal registry, not the internal location of menu definitions in PeopleSoft Application Designer.

See "Understanding the Registration Wizard" (PeopleTools 8.54: Application Designer Developer's Guide).

This table describes the fields that appear on the Content Ref Administration page:

Usage Type

Select from these usage types:

- *Frame template.*
- *Homepage tab.*
- *HTML template.*
- *Inline frame template.*
- *Pagelet.*

  Important! The portal recognizes pagelets that are registered as portal objects in the Pagelets folder only. This restriction is not enforced by the portal registry; you can create a pagelet that is not recognized.
  The name that you specify for a new pagelet can consist of letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.

- *Target.*

  This is the default value.

Based on the value that you select, the application shows, hides, disables, and enables the other fields on the page so that only the fields that are relevant to the usage type are available. For example, if you select a usage type of *Pagelet,* then Pagelet Attributes group box appears in which you can specify the default column, the refresh time, and other properties that apply specifically to pagelets.

For a typical PeopleSoft application page, set the usage type to *Target* (the default value) and the URL type to *PeopleSoft Component.* With a usage type of target, the content will populate the target area on the page.

Storage Type

This field is enabled when you selected a usage type of *Frame template* or *HTML template.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	Select *Local (in HTML Catalog)* or the default value, *Remote by URL.*
HTML Area	This text area is enabled when you select a usage type of *HTML template* or *Frame Template* and *Local (in HTML Catalog)* as the storage type. An HTML template uses the HTML that is entered into the HTML area of a content reference that is located in the Templates folder, which you find within the Portal Objects folder. Bind variables cannot be used within the HTML area. The HTML area includes any HTML that is required for the template itself, plus tags that specify the URLs for template pagelets.
Template Name	Select the template for this page. If you leave this field blank, the portal servlet automatically uses the default template of the node at runtime. The node default template is set up on the General Settings page. If a node default template has not been specified, it uses the portal default template.
WSRP Producible	This option appears only for usage types of *Target* and *Pagelet.* Select this option for PeopleSoft content that you intend to expose for remote WSRP portals to consume.
No Template	Select to have the target content (your application page) not wrapped with a portal template at runtime; by viewing your page, the user is effectively taken outside of the portal environment (this is not usually desirable). When this check box is selected, the Template Name field is hidden.
Fluid Mode	Select this option for this content reference to be displayed in fluid mode. This is required for fluid rendering.
Small Form Factor Optimized	Select if this content reference has been designed specifically for optimum display and performance on a smartphone. If selected, this content reference will be available on small form factor devices.
Create Content Reference Link	Click to create or edit a content reference link.
Add Content Reference	Click to add a content reference.
Test Content Reference	Click to test a content reference. See Creating Related Links.
Hide from portal navigation	Select to hide the content reference from menu navigation.
Hide from MSF Navigation	MSF is not currently supported.
IWC Message Events	Click to access the Interwindow Communications page. See Publishing and Subscribing to IWC Messages.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

URL Information

The fields that appear in this group box are based on the selection that you make in the URL Type field.

Node Name

Select the node for this page. Select *Always use local* to associate the page with the default local node. The default local node is designated on the Portal Definitions page.

URL Type

Select from these URL types:

- *Non-PeopleSoft URL:* Enter the URL in the Portal URL field that appears.

  This usage type can also be used in conjunction with a defined node. The result is the concatenation of the selected node's URI text and the portal URL that you entered.

- *PeopleSoft Component:* The Component Parameters region appears. Select the menu name, market, and component name to use for this content reference.

- *PeopleSoft Generic URL:* Enter the URL in the Portal URL field that appears.

  This usage type can also be used in conjunction with a defined node. The result is the concatenation of the selected node's URI text and the portal URL that you entered.

- *PeopleSoft Homepage Tab:* No longer used. Do not select this option.

- *PeopleSoft Script:* The iScript Parameters group box appears. Select the record (table) and field names, PeopleCode event name, and PeopleCode function name that specifies the iScript to use.

- *PeopleSoft Worklist URL:* Enter the URL in the Portal URL field. Use the following format:

  ```
  w/WORKLIST?
  ICAction=ICViewWorklist&Menu=Worklist&Market=GBL&PanelGroupN
  ```

- *Remote Portlet:* Select the producer and the portlet that you want to register manually. Only producers that have been registered are available for selection. Values for the fields Label and Long Description are supplied by default to the portlet's title and description. You can change these values. Click Producer Details to display the Display Producers page and view remote producers that have been defined in the system.

  See Using the WSRP Consumption Wizard (Consumer).

Note: The availability of these options depends on the selected usage type.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Additional Parameters Enter query string parameters to be appended onto a PeopleSoft component or PeopleSoft script. For example, enter *emplid=8001*.

Pagelet Attributes

The Pagelet Attributes group box enables you to manage pagelet behavior and appearance.

See Using Attributes to Enhance Pagelets.

Content Reference Attributes

Content reference attributes provide a framework to specify and store free-form information about a content reference. For example, you can use content reference attributes to specify and store keywords, help information, related content properties, and related links for content references.

In addition, portal-aware applications can use content reference attribute information for application-specific processing. Portal-aware applications are those that:

- Know about the portal registry and API.
- Know how to get information from a content reference.
- Contain code that can access the properties of content references and do something meaningful with the information.

Name Enter an object name to identify the attribute programmatically. This name is not displayed to the user.

Note: The name can consist of letters, digits, and underscores (_) only. It cannot contain any spaces and cannot begin with a digit.

Translate Select to specify that the label and attribute values are translated.

Attribute Information Click to create related links. You can create related links only after entering *RELLINK* in the Name field.

Label Enter a descriptive name for the attribute to display to the user. The label is required only if Translate is selected.

Attribute value Enter information about the attribute that the portal uses to apply the attribute to the content reference.

Links

This grid appears only if content reference links that connect to this content reference exist. Click Link Definition to view the definition of each listed content reference link.

Available Content Reference Attributes

This table describes the content reference attributes that your PeopleTools portal recognizes:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Content Reference Attribute Name	***Description***
BCUPDATELABEL	Use this attribute to override the default label in the breadcrumbs. See Overriding Default Breadcrumbs Using Content Reference Attributes.
BCUPDATETYPE	Use this attribute to override the default breadcrumb update behavior. See Overriding Default Breadcrumbs Using Content Reference Attributes.
KEYWORD KEYWORD_*LNG*	Use these attributes to specify a keyword or phrase that you want to associate with this content reference. Specific portal features, such as the Pagelet Search page, make use of these keywords when searching for definitions. **Note:** However, while the KEYWORD attribute is also included in Verity-based search indexes, it is not included in indexes generated by the PeopleSoft Search Framework.
Name of a default homepage object or image	Use this attribute to override a default HTML object or image used for homepage components, such as tabs and pagelets, by providing the name of an alternative HTML object or image. **Note:** You can override only a specified set of homepage objects and images.
IMAGE_BIND_12 to IMAGE_BIND_26	Use these attributes (numbered from 12 to 26) to insert additional custom images on a homepage tab.
NAVNEWWIN	Use this attribute to specify that the component or page always opens in a new browser window. **Note:** This attribute applies to links on navigation collection template pagelets that appear in the pagelet area of a WorkCenter page.
NAVTOPWIN	Use this attribute to specify that the component or page opens in the current browser window and replaces the current page completely. **Note:** This option affects only the link behavior when it is rendered in a navigation collection template pagelet inside the pagelet area of a WorkCenter page.
PSCACHECONTROL	Use this attribute to implement caching for individual pagelets and targets based on user, role, or application.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Content Reference Attribute Name	*Description*
PORTAL_BANNER_PGLT	Use this attribute to create a banner pagelet, which can appear at the top of a homepage or dashboard page and spans the width of the page. **Note:** To use a pagelet as a banner pagelet, it has to be added to a homepage tab or dashboard page as a required-fixed pagelet in the first column. See Maintaining Pagelet Branding Attributes.
PORTAL_ENCODING_OVERRIDE	Use this attribute to override the character set encoding that is used by the content reference.
PORTAL_HP_COMPONENT	Use this attribute to hide pagelet attributes such as the title bar, the borders, or both. See Maintaining Pagelet Branding Attributes.
PSTIMEOUT	Use this attribute to specify how long the portal should wait for a pagelet to load before it considers the pagelet unavailable.
PT_PORTAL_PGLT_HDR_IMG	Use this attribute to add an icon to the header of the pagelet. The attribute value should be the name of the image definition that is stored in the image catalog using PeopleSoft Application Designer.
PTPPB_PAGELET_HELP_LINK	Use this attribute to specify a URL that you want to associate with the help link for a pagelet. **Note:** This attribute takes priority over a PeopleSoft Help ID if both properties are set.
RCF_SERVICE_PARAMETER	Use this attribute to provide related content service parameters in name/label pairs for the content reference. The presence of this attribute changes the default behavior of component-based URLs and supplants all parameters with those that you enter in the Value field. See Defining Related Content Services.
RELLINK	Use this attribute to create related links for the content reference.

Content Reference Type Parameters

This table summarizes the valid combinations of usage type, storage type, and URL type or key HTML content supported for content references:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Usage Type	*Storage Type*	*URL Type or Key HTML Content*
Frame template	Remote by URL	PeopleSoft Script
Frame template	Remote by URL	Non-PeopleSoft URL
Frame template	Local (in HTML Catalog)	Target or IClientComponent element
Target	Remote by URL	PeopleSoft Component
HTML template	Local (in HTML Catalog)	Target, Pagelet, or Source element
HTML template	Remote by URL	PeopleSoft Script
HTML template	Remote by URL	Non-PeopleSoft URL
Homepage tab	Local (in HTML Catalog)	PeopleSoft Homepage tab
Inline Frame template	Local (in HTML Catalog)	Target or Pagelet component
Inline Frame template	Remote by URL	PeopleSoft Script
Inline Frame template	Remote by URL	Non-PeopleSoft URL
Pagelet	Remote by URL	Non-PeopleSoft URL
Pagelet	Remote by URL	PeopleSoft Component
Pagelet	Remote by URL	PeopleSoft Script
Pagelet	Remote by URL	PeopleSoft Generic URL
Pagelet	Remote by URL	PeopleSoft Worklist URL
Pagelet	Remote by URL	Remote Portlet
Target	Remote by URL	PeopleSoft Generic URL
Target	Remote by URL	PeopleSoft Script
Target	Remote by URL	PeopleSoft Worklist URL
Target	Remote by URL	Non-PeopleSoft URL
Target	Remote by URL	Remote Portlet

Related Links

Defining Folders

Understanding Portal Templates

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Managing General Portal Settings
Configuring Pagelet Time-out Settings
Using Predefined Homepage HTML Definitions
Administering Server-Based Caching
Overriding Page Encoding
Maintaining Pagelet Branding Attributes
Searching for and Importing Pagelets

Creating Related Links

You can create related links for a portal page using the RELLINK content reference attribute. You can create related links to external sites, to internal sites, and to other portal pages.

Note: Related links can be implemented with your choice of template on the content reference. Using the PeopleTools Related Links template displays the related links for the content reference, but not the portal menu navigation.

To create related links:

1. Define the RELLINK content reference attribute.
2. Create related links groups.
3. Create related links for each group.
4. Test the related links.

Image: Related links for a content reference

In this sample page screenshot, you see the results of using the related links template and adding the RELLINK attribute on a given content reference:

This table describes the page elements shown in the sample page:

Page Element	*Description*
Related Links	RELLINK content reference attribute with the label Related Links.
External Sites	A related link group defined with the label External Sites.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Page Element	*Description*
Site #1, Site #2, Site #3	Related links for the External Sites group.
PeopleSoft Sites	A related link group defined with the label PeopleSoft Sites.
Batch rules, Developers' site, and User maintenance	Related links for the PeopleSoft Sites group.

Defining the RELLINK Content Reference Attribute

To define a related link content reference attribute:

1. Access the Content Ref Administration page (PeopleTools, Portal, Structure and Content. Navigate to the appropriate content reference; then click the Edit link.)
2. Enter *RELATEDLINKS_TEMPLATE.* in the Template Name field.
3. In the Name field of the Content Reference Attributes region, enter *RELLINK*.

   After you enter this value and move out of the field, the Attribute Information link changes to Edit Related Links.
4. Verify that the Translate check box is selected.
5. Enter a label for the related links attribute.

Note: You do not need to enter any information in the Attribute Value field. After you create the related link group and the related links, the system populates this field with all defined attributes as a string.

Creating Related Link Groups

Access the Related Links Group (PORTAL_RLNK_ADM) page.

To create a related link group:

1. In the Related Link Groups Label field, enter a name.

   This name is the heading under which you place related links.
2. To create additional related link groups, click Add and repeat step 1.

Creating Related Links

To create a link:

1. In the Related Links region on the Related Links Group (PORTAL_RLNK_ADM) page, enter a name in the Related Link URL Label field.

   This name appears as a link under the group name.
2. If you're linking to an internal site, select a node name from the list.

   If you're linking to an external site, leave this field blank.
3. In the Related Link URL field, enter URL information for the link.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

If the link is to an internal site, enter the URL or URI. The URL or URI can be to an internal site, to a portal, and so on. If you specified a value for the Portal URI Text field on the node definition, you can enter a specific document name as the target. The name is appended to the URI text for a full URL.

If the link is to an external site, enter the URL.

4. To define additional links for the group, click Add and repeat steps 2 and 3.

5. Click OK to save the entry and return to the Content Ref Administration page.

   The Attribute Value field is populated with an XML-type formatted string. This string contains all the related link data and is stored in the PORTAL_ATTR_VAL field of the PSPRSMATTRVAL record for the content reference specified in the PORTAL_OBJNAME field. The following example shows a generated XML-type formatted string:

```
<GRPLBL>Worklist Pages
  <URLLBL>Worklist Details</URLLBL>
  <CP>Portal</CP>  <URL>ICType=Panel&Menu=WORKLIST&Market=GBL&PanelGroupName=W⇒
ORKLIST_DETAILS
  </URL>
  <URLLBL>Worklist Monitor</URLLBL>
  <CP>Portal</CP>  <URL>ICType=Panel&Menu=WORKFLOW_ADMINISTRATOR&Market=GBL&Pa⇒
nelGroupName=WF_MONITOR_01
  </URL>
 </GRPLBL>

 <GRPLBL>Tasks
  <URLLBL>Task List Summary</URLLBL>
  <CP>Portal</CP>
  <URL>ICType=Panel&Menu=PORTAL_COMPONENTS&Market=GBL&PanelGroupName=EO_PE_TAS⇒
K_LIST  </URL>
</GRPLBL>
```

Testing Related Links

You should test related links to view how the page appears to users, ensure that the labels are correct, and verify that the links function properly. To test related links, click Test Content Reference on the Content Ref Administration page just above the URL Information region.

Note that:

- If the portal cache is enabled on the web server, the related link template does not appear for a content reference page accessed from the menu until the cache expires.

- Related link URLs are displayed only when a user has security access to the URLs.

  To grant security access to a related link URL, register the related link URL as a separate content reference and associate permission lists with it.

  If the related link is not separately registered, then it is treated as having public access.

  If a related link group contains no URLs to which the user has security access, the related link group is hidden. If the user does not have access to any of the related links associated with the target page, then the related links section of the template is hidden.

If the target page does not display any related links, ensure that:

- The application and portal databases are registered as content providers.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Also check for data-entry errors.

- The templates are registered in PeopleSoft Application Designer with the correct names.

  If the templates are registered, check for data-entry errors.

- The correct related link template is specified for the target page.

  Also ensure that the value of the PORTAL_TEMPL_NAME field in the PSPRSMDEFN record displays a correct related link template

- The RELLINK string was saved on the target page in the content registry.

  Check whether an XML string appears in the Attribute Value field on the Content Ref Administration page. If not, then delete and reenter the RELLINK attribute for all the related link information.

- You assigned security in the registry to the user for the content references listed as related link URLs.

  If not, assign security using the Content Reference Security page for registered related link URLs.

Setting Content Reference Permissions

Access the Content Reference Security page (Select PeopleTools, Portal, Structure and Content. Click the Edit link for a content reference; then select the Security tab.)

Image: Content Reference Security page

This example illustrates the fields and controls on the Content Reference Security page. You can find definitions for the fields and controls later on this page.

This page is similar to the Folder Security page. You can specify the permission lists and roles that grant security access to the content reference.

Certain URL types inherit the security permission list and role authorizations from the referenced object. For these URL types, you can view, but not modify the security by using the Content Reference Security page. All fields except Public and Author Access are display-only. This is the list of URL types that inherit security authorization:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *PeopleSoft Component.*
- *Worklist.*
- *PeopleSoft Script.*

When you save a content reference that points to a PeopleSoft component or script on the default local node, the security settings for the content reference are inherited automatically from the menu or script security settings. Any permission lists or roles that you entered on the Content Reference Security page before saving the content reference are deleted and replaced by those of the component or script. These new security settings are also propagated up the folder hierarchy. This ensures that the content reference is visible and all folders are viewable, enabling users to navigate to the component.

Note: If you copy, move, or add a content reference to a folder, the security settings of the content reference are propagated up the folder hierarchy. This ensures that the content reference is visible and all folders are viewable, and it enables users to navigate to the content reference. If you remove the content reference from that folder, the security settings that were applied when it was added to the folder are also removed from the folder hierarchy, without disturbing any settings still required for other content references remaining in that folder.

Note: Content reference links do not inherit cascaded security from folders.

Related Links

Setting Folder Permissions

"Understanding the Registration Wizard" (PeopleTools 8.54: Application Designer Developer's Guide)

"Understanding Permission Lists" (PeopleTools 8.54: Security Administration)

"Understanding Roles" (PeopleTools 8.54: Security Administration)

Reviewing Content Reference Permissions

You can use automated queries to discover the content references to which a given user, permission list, or role has access. Each query provides a different degree of detail about the properties of the content references.

Reviewing Content Reference Security by User

Access the User ID Queries page (PeopleTools, Security, User Profiles, User Profiles, User ID Queries).

This page contains several general-purpose security queries, along with the following links to content reference access queries for the selected user, which are described on the page:

- User ID's Content Reference Access.
- User ID's Content Reference (includes Portal) Access.
- User ID's Content Reference (includes Menu, Component and Market) Access.
- User ID's Content Reference (includes Portal, Menu, Component and Market) Access.

When you click one of the query links, a page appears in a new window, containing a grid with the information described for each content reference.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Reviewing Content Reference Security by Permission List

Access the Permission List Queries page (PeopleTools, Security, Permissions & Roles, Permission Lists, Permission List Queries).

This page is similar to the User ID Queries page, with the following equivalent links to content reference access queries for the selected permission list:

- Permission List's Content Reference Access.
- Permission List's Content Reference (includes Portal) Access.
- Permission List's Content Reference (includes Menu, Component and Market) Access.
- Permission List's Content Reference (includes Portal, Menu, Component and Market) Access.

When you click one of the query links, a page appears in a new window, containing a grid with the information described for each content reference.

Reviewing Content Reference Security by Role

Access the Role Queries page (PeopleTools, Security, Permissions & Roles, Roles, Role Queries).

This page is similar to the User ID Queries page, with the following equivalent links to content reference access queries for the selected role:

- Role's Content Reference Access.
- Role's Content Reference (includes Portal) Access.
- Role's Content Reference (includes Menu, Component and Market) Access.
- Role's Content Reference (includes Portal, Menu, Component and Market) Access.

When you click one of the query links, a page appears in a new window containing a grid with the information described for each content reference.

Testing Content References

After creating or modifying a content reference and setting up permissions, test the content reference.

1. Click Home in the portal navigation header
2. Use the Menu pagelet or the Main Menu to navigate to the content reference.
3. Click the content reference.

You can also click the Test Content Reference on the Content Ref Administration page.

Note: The Test Content Reference link appears on the Content Ref Administration page when the usage type of the content reference type is set to *Target* only.

Setting Fluid Attributes for Content References

Access the Fluid Attributes page (Select PeopleTools, Portal, Structure and Content. Click the Edit link for a content reference; then select the Fluid Attributes tab.)

Use the Fluid Attributes page to configure the behavior and appearance of tiles. See "Configuring Tile Options" (PeopleTools 8.54: Fluid User Interface Developer’s Guide) for more information.

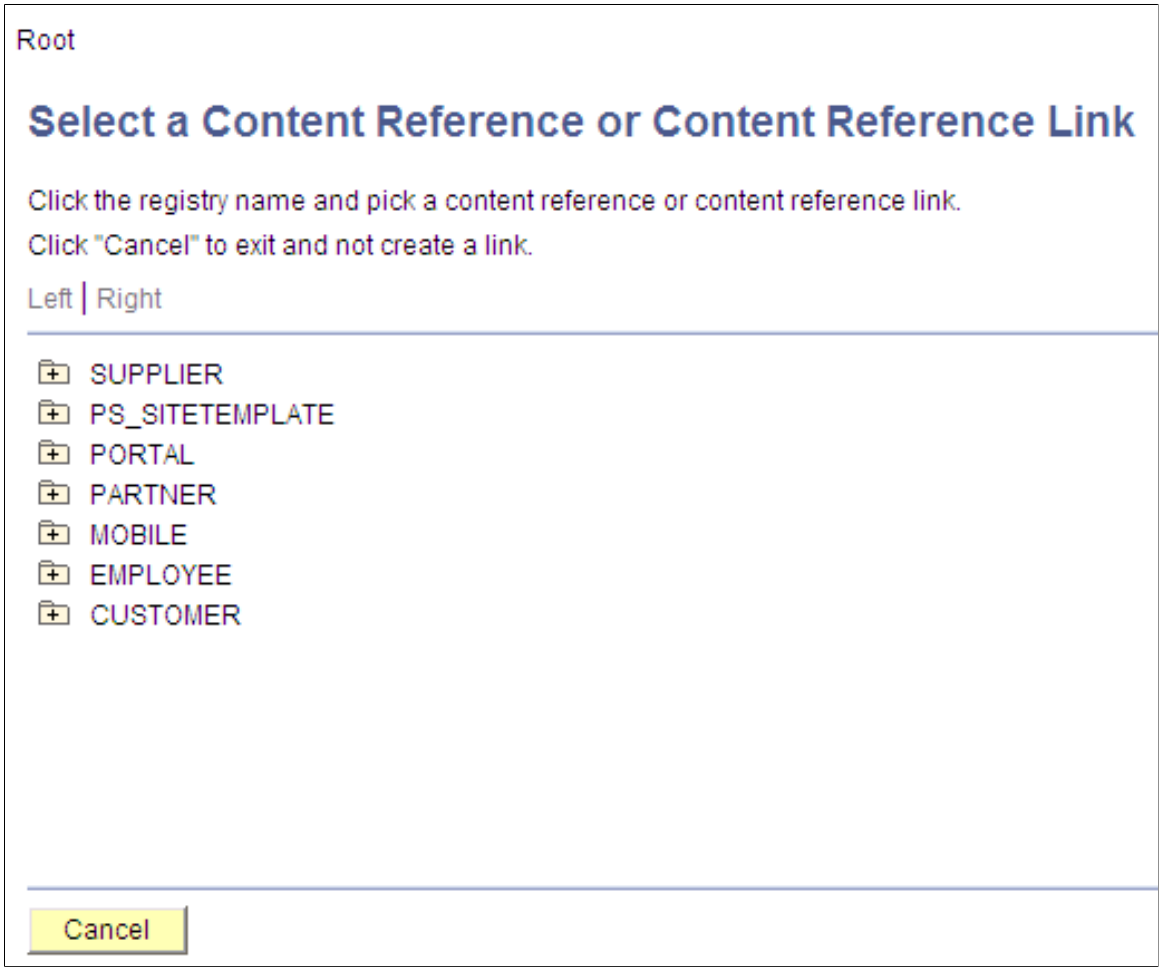
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Content Reference Links

Access the Select a Content Reference or Content Reference Link page (Select PeopleTools, Portal, Structure and Content. Navigate the portal registry hierarchy into the parent folder in which you want the link to appear. Click the Add Content Reference Link link that appears after the Content References group box.)

Image: Select a Content Reference or Content Reference Link page

The following screenshot illustrates how to define content reference links on the Select a Content Reference or Content Reference Link page.

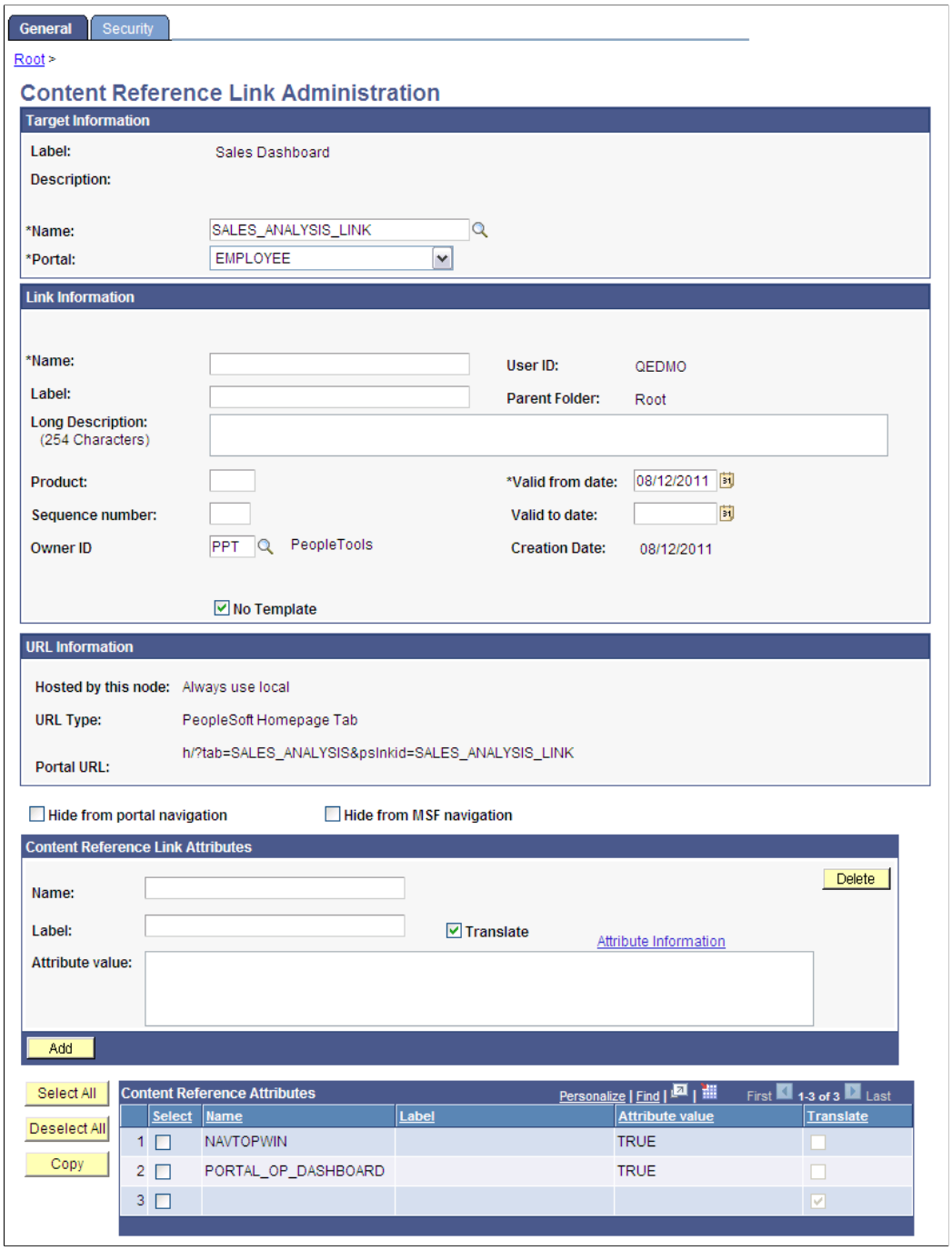


 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Click the folder icons to navigate the tree structure until you find the target of the link. Click the link and the Content Reference Link Administration page appears.

Image: Content Reference Link Administration page

The following screenshot illustrates selecting content reference or content reference link page on the Content Reference Link Administration page.



PeopleSoft portals store content references based on their URLs, so you can't register two different content references that specify the same URL. Content reference links overcome this limitation by enabling you to define more than one menu item that invokes a given URL. You define a content reference link like any other content reference, but instead of specifying a URL, you specify an existing

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

content reference as its destination. At runtime, clicking the content reference link on the portal menu connects to the destination content reference by URL.

Note: To the user, a content reference link looks and works the same as a content reference on the portal menu.

A content reference link can point to:

- A content reference of usage type *Target.*
- Another content reference link.

You can define a content reference link in the same locations that you can define a content reference of usage type *Target.* Multiple content reference links can connect to a single destination content reference.

By connecting one content reference link to another content reference link, you can create a chain of redirection, which always ends in a content reference of usage type *Target.* You don't need to know what the ultimate destination is when you define your new content reference link, and you can duplicate the property settings of the content reference link to which it connects rather than those of the ultimate destination. However, Oracle recommends that you limit your use of chains of content reference links because they impose processing overhead on your portal when selected.

Warning! If you delete a destination content reference or content reference link, all content reference links that connect to it are also deleted. This behavior propagates back through any content reference link chain.

Target Information

Name

Select the destination content reference to which you want this content reference link to connect.

Portal

Select the portal that contains the destination content reference to which you want this content reference link to connect. The *Always use local* value refers to the portal where the content reference link is defined, so moving or copying it to a different portal changes the resolution of this value.

Important! If you change this value, you must also select a new link content reference.

Link Information

Name

Enter a unique object name for this content reference link. This field is required.

Test Content Reference Link

Click to test this content reference link. Clicking this button should produce the same result as clicking the content reference link from the portal menu.

If you omit values in following fields, their values are inherited from the destination content reference:

- Label.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Long Description.
- Product.
- Sequence number.
- Object owner identifier.
- No Template.
- Template Name.
- Valid from date.
- Valid to date.

URL Information

These fields are display-only and reflect the URL information of the content reference that you are linking to.

See “URL Information” in Defining Content References.

Content Reference Attributes

If the destination content reference has defined content reference attributes, they appear in this grid. You can copy them from the grid to the Content Reference Link Attributes group box using the following page elements:

Select	Select to designate a content reference attribute for copying to the Content Reference Link Attributes group box.
Select All	Click to designate all content reference attributes for copying to the Content Reference Link Attributes group box.
Copy	Click to copy the selected content reference attributes to the Content Reference Link Attributes group box.

Content Reference Link Attributes

Content reference link attributes and content reference attributes share the same page fields.

You can define content reference link attributes directly in this group box, or you can copy an attribute from the destination content reference using the Content Reference Attributes grid, and modify it to suit your purposes.

At runtime, the content reference link attributes defined here are combined with the content reference attributes defined at the link destination, and they are applied together.

Important! If a content reference link attribute defined here has the same name as a content reference attribute defined at the link destination, but a different attribute value, the value defined here takes precedence and is applied at runtime.
Content reference link attributes are independent and separate from the content reference attributes at the link destination. Modifying one has no effect on the other.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Setting Content Reference Link Permissions

Access the Content Reference Link Security page (Click the Security tab on the Content Reference Link Administration page.)

The fields on this page are disabled because content reference links inherit the security settings of the destination object. The inherited security settings propagate upward through the folder hierarchy from the content reference link.

Related Links

Setting Content Reference Permissions

Managing Folders and Content References

This section discusses how to:

- Copy folders and content references.
- Move folders and content references.
- Delete folders and content references.
- Synchronize portal object security.
- Register components in multiple portals.
- Register components multiple times.

Copying Folders and Content References

Access the Copy Object to a Different Portal Registry page (Select PeopleTools, Portal, Structure and Content, Portal Objects, Content Ref Administration. Click the Copy Object button).

To copy an object:

1. Navigate to the portal registry to which you want to copy the object.

   Click the portal name to copy the object to root level. To expand the folders and navigate further into the registry structure, click the folder icon (not the folder name).

   Important! Clicking the folder name starts the copying process.

2. (Optional) If you are copying a folder, the Recurse subfolders? check box appears. Select this check box to copy the folder and all the subfolders and content references within it.
3. Click the folder name to copy the object into that folder.
4. Click the Yes - Copy button to proceed.

Warning! If you attempt to copy an object to a location that already contains an object of the same name and type, the newly copied object replaces the existing object. For example, if you copy a folder and subfolders to another portal where a subfolder by that name already exists, the existing subfolder is deleted and replaced with the subfolder from the source portal. No warning message appears.

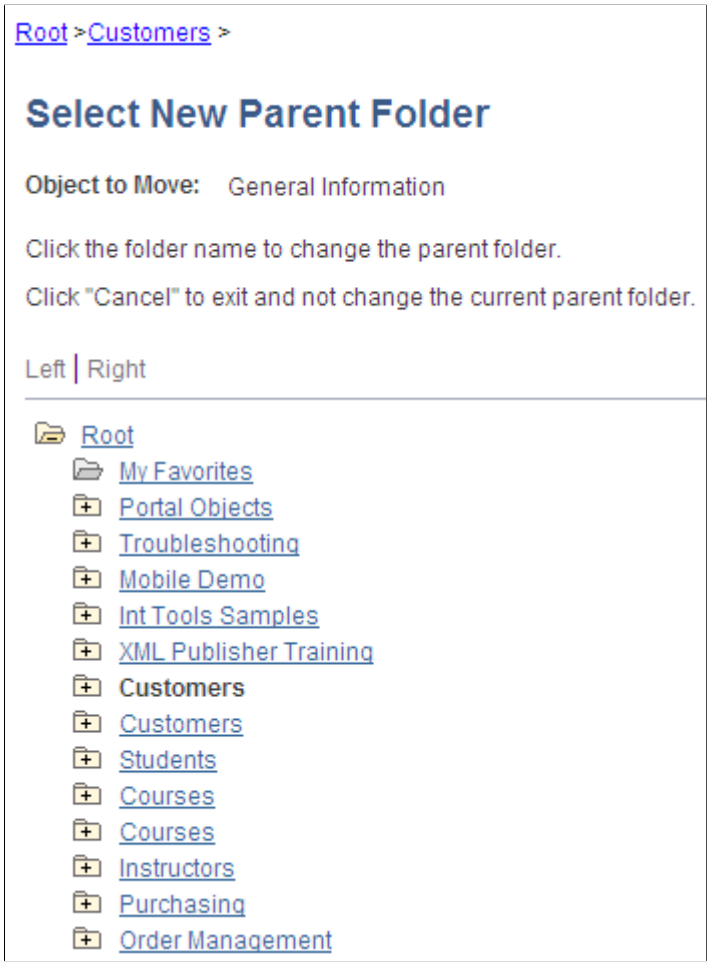
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Moving Folders and Content References

Access the Select New Parent Folder page (PeopleTools, Portal, Structure and Content, Portal Objects, Content Ref Administration Click the New Parent Folder button).

Image: Select New Parent Folder page

The following screenshot illustrates moving folders and content references on the Select New Parent Folder page.



The current parent folder is shown in bold type, and the folder tree is expanded just enough to show the location of the current parent folder. All other folders are shown collapsed. You can expand the tree to show content references, which are displayed as tree leaves and appear in normal text.

To move a folder or content reference to a new parent folder:

1. Navigate to the new parent folder by clicking the appropriate folder icons with plus signs.

   Note: When navigating, do not click the folder name unless it's the name of the new parent folder that you want to select. Once you click a folder name, as opposed to the icon, the folder name that you selected becomes the new parent folder.

2. Select the new parent folder by clicking the folder name.

3. Click Save.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: When you move a content reference that points to a component or script at the default local node, its security settings are cascaded up to its new parents all the way up to the root folder. In addition, when you move a folder, its security settings (not including security inherited from a subordinate object) are cascaded up to its new parent, all the way up to the root folder. This ensures that the folder or content reference is accessible in its new location.

Deleting Folders and Content References

To delete a folder or content reference:

1. Select PeopleTools, Portal, Structure and Content.
2. In the portal registry, navigate to the folder or content reference that you want to delete.
3. Click the Delete button next to the object that you want to delete.

   All child folders and child content references are deleted when you delete the selected folder. Make sure that no critical dependencies exist for these objects before deleting the parent folder.

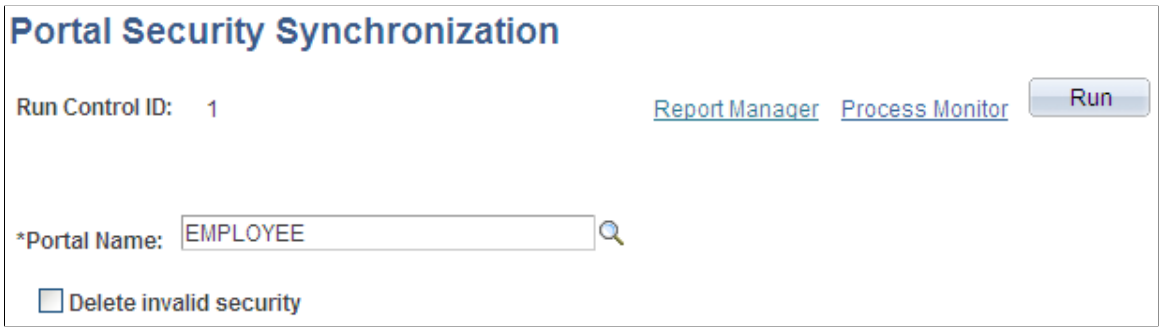
Warning! Do not delete folders or content references that are delivered with PeopleTools. These default objects have special significance to your portal. Changing them could make your portal unusable.
When you delete a content reference or a content reference link, all content reference links that connect to it are also deleted. This behavior propagates back through any content reference link chain.

Synchronizing Portal Object Security

Access the Portal Security Synchronization page (PeopleTools, Portal, Portal Security Sync).

Image: Portal Security Synchronization page

The following screenshot illustrates synchronizing portal object security on the Portal Security Synchronization page.



The hierarchical relationships and dependencies between objects in the portal registry determine what security settings each object must have. The portal won't work correctly if these security relationships aren't maintained. Here are some examples of these relationships:

- A folder that is not public or hidden must have at least the level of access that its immediate child objects (folders, content references, and content reference links) have.
- A content reference link must have exactly the same level of access as the object (content reference or content reference link) to which it links.
- A content reference that represents a PeopleSoft component or iScript must have exactly the same level of access as the object that it represents.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Portal object security settings can become unsynchronized when you move portal objects from one database to another using the Project Copy feature in PeopleSoft Application Designer. When you merge projects this way, if the projects contain any portal objects with identical names, the security settings of the portal objects in the last project copied overwrite the security settings of portal objects copied earlier. Also, when a copied portal object doesn't overwrite an existing object, it changes the structure of the resulting portal registry hierarchy.

Use the Portal Security Synchronization page to reinstate the correct security relationships between objects in the portal registry after you copy a project that contains portal objects.

This page enables you to run the Application Engine program PORTAL_CSS from the standard menu navigation. You can use this feature on local portals only, not on portals on remote databases. In addition, from any local portal, you can run this program against another local portal.

To synchronize portal object security, specify a local portal in the Portal Name field and click Run. The portal objects are synchronized in the following way:

1. The security settings of each content reference are compared to the component or iScript that it represents, and updated to match.

2. The security settings of each content reference link are compared to the content reference or content reference link to which it connects, and updated to match.

3. The security settings of each content reference and content reference link are propagated to its parent folder, in addition to the parent folder's existing settings.

   None of the parent folder's existing security access is reduced.

   Note: The settings aren't propagated if the parent folder is public or hidden.

4. The security settings of each folder are propagated to its parent folder, in addition to the parent folder's existing settings.

   None of the parent folder's existing security access is reduced.

   Note: The settings aren't propagated if the parent folder is public or hidden.

The following warning messages may appear when security synchronization cannot occur:

- No permissions in PSAUTHITEM for object %1

- Cref %1 points to Menu: %2, Component: %3 which doesn't exist.

  Note: When this message appears, you need to delete the invalid content reference .

The security administrator and portal administrator are responsible for determining why an object has no security or why an object does not exist.

Delete Invalid Security Option

When you move portal objects from one database to another, roles and permission lists assigned to folders and content references on the source database may not exist on the target database and therefore become invalid. Select the Delete Invalid Security check box on the Portal Security Sychronization page to remove nonexisting roles and permission lists from folders and content references.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: When the Delete Invalid Security option is selected, the PORTAL_CSS process runs more slowly because it checks every role and permission list on every portal registry structure. When a nonexisting role or permission list is found on the portal registry structure, it is removed from that portal registry structure.

Registering Components in Multiple Portals

After you use Registration Wizard to register a new component in a portal, you may need to register the component in other portals. You can register components in multiple portals in two ways:

- Use Registration Wizard again.

  Select the second option: Add this component to a portal registry. Then reenter the content reference label, long description, and sequence number (if not 1) to match the other portal's entry.

- Copy the component to other portals.

  When you have copied your component to another portal, open that portal to verify that it was copied as anticipated.

See Copying Folders and Content References.

Related Links

Changing Default Portals

Registering Components Multiple Times Within a Portal

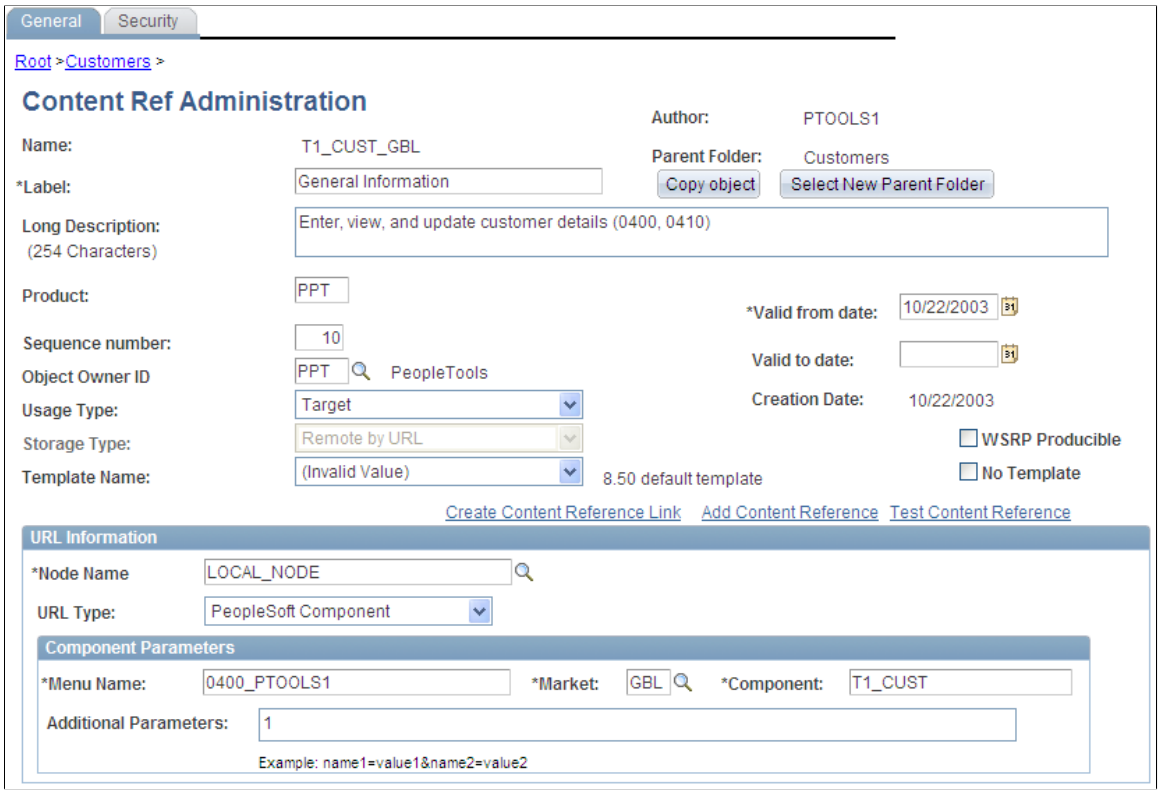
The portal registry does not allow a URL to be registered more than once. When you define a content reference, the component name is part of the URL. If you must register the same component URL in more

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

than one location, add a unique character to the parameter field. (Be sure that you really must register a URL twice because this creates upgrade and maintenance concerns).

Image: Registering a component the second time

The following screenshot illustrates an example showing that an additional numeric character *1* is added to the Additional Parameter field to allow a component to be registered a second time:



Related Links

Registering Components in Multiple Portals

Managing General Portal Settings

This section discusses how to manage general portal settings.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Managing General Settings for Portals

Access the General Settings page (PeopleTools, Portal, General Settings).

Image: General Settings page

The following screenshot illustrates the fields and controls on the General Settings page.

Use this page to manage the settings for the portal that you're signed in to. To modify other portals, use the Portal Definitions page.

Note: You must sign out of the portal and sign in again before your changes take effect.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Default Template Name Select the template to use to wrap all registered content that has not been associated with another template in this portal, as well as all unregistered content.

Warning! If you change the default template, make sure that the new default is able to handle any arbitrary page.

See Administering Portal Definitions.

Folder Navigation

You enter the default folder navigation values for the current portal on this page. Enabling folder navigation at the portal level enables folder navigation for all standard folders and SmartNavigation in the portal. You can subsequently disable folder navigation at the individual folder level

Folder navigation is not enabled unless you select the Is Folder Navigation Enabled check box on both the Portal Definition and General Settings pages. The Is Folder Navigation Enabled check box is selected by default in the standard PeopleTools portals.

See Adding and Editing Portal Definitions.

Note: If you disable and then re-enable folder navigation at this level, folder navigation is re-enabled for *all* folders. All overrides at the individual folder level are cleared.

Portal Search Options

Use the fields in this group box to enable system-level search options.

Display search in navigation Select to display the search at the top of the menu pagelet or main menu.

Display search in header Select to display the search field in the universal navigation header. This search is limited to the current application. You can display either the standard search or the global search in the universal navigation header. The two options are mutually exclusive.

Display global search in header Select to display the portal-wide search field in the universal navigation header. This search spans all application portal sites. You can display either the standard search or the global search in the universal navigation header. The two options are mutually exclusive.

Auto Suggest Limit Enter the maximum number of rows to appear in the drop-down autocomplete list for edit and prompt fields that have autocomplete enabled at the record definition level.

The default value is *50.*

Search Results Page

Enter the names of the menu definition, component definition, and market of the Search Results page. This group box is disabled when the Display global search in header check box is selected.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation Options

Navigation Type

Select *Left* to enable menu navigation that uses the Menu pagelet, which appears at the side of the page.

Select *Drop-down* to enable menu navigation that uses the navigation header, which appears at the top of the page.

Important! Many PeopleTools portal-related features, such as SmartNavigation, persistent search, and the My Favorites and Recently Used menus are available only when you use drop-down navigation type.

Favorites Options

Display My Favorites folder

Select to enable the application user to save favorite components and pages in the My Favorites folder, which appears in the Favorites drop-down menu. This option is available only when you select *Drop-down* as your navigation type.

Display Recently Used

Select to enable the portal to store and display the five most recently visited components as a folder under the Favorites menu. This option is available only when you select *Drop-down* as your navigation type.

Menu Text Options

Display menu text in header

Select to show the menu title in the header rather than in the left frame above the menu.

Display menu in nav pagelet

Select to show the menu title in the pagelet header.

Note: In PeopleTools, the menu title is displayed in the header. If both Display menu text in header and Display menu in nav pagelet are selected, the menu title appears twice.

Menu title position

The number of pixels from the top of the frame that the menu expand/collapse bar appears in a frame template. The default is *47*.

Disable Drop-down Menu Sort

Enter the PeopleSoft language codes of the languages for which you want to remove the drop-down menu sort functionality at the system level. The drop-down menu sort is disabled for these languages by default: Japanese, Korean, Thai, Simplified Chinese, and Traditional Chinese.

Enter PeopleSoft language codes in a comma-separated list. Select PeopleTools, Utilities, International, Languages to see all PeopleSoft language codes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Header Frame Options

This group box sets the number of header frame rows if you are using the PeopleTools default template. The value entered for this field determines the size of the header frame. The default value for the PeopleTools portal is *65* and the default value for the PeopleSoft Interaction Hub is *100.*

Node Templates

Configure node-specific template information.

Node Name	Select from the list of nodes defined for this portal. If you add a new node name, you must also specify the default template for that node.
Default Template Name	Select the default template for all content references using this node. The selected template is applied only to content references for which the No Template check box is deselected but no template is selected.

Related Links

"Using Menu Pagelets" (PeopleTools 8.54: Applications User's Guide)

Managing Portal Objects

The Portal Objects folder is of special significance to the portal. This section discusses how to:

- View Portal Objects folder settings.
- Manage templates.
- Add and edit templates.
- Delete templates.

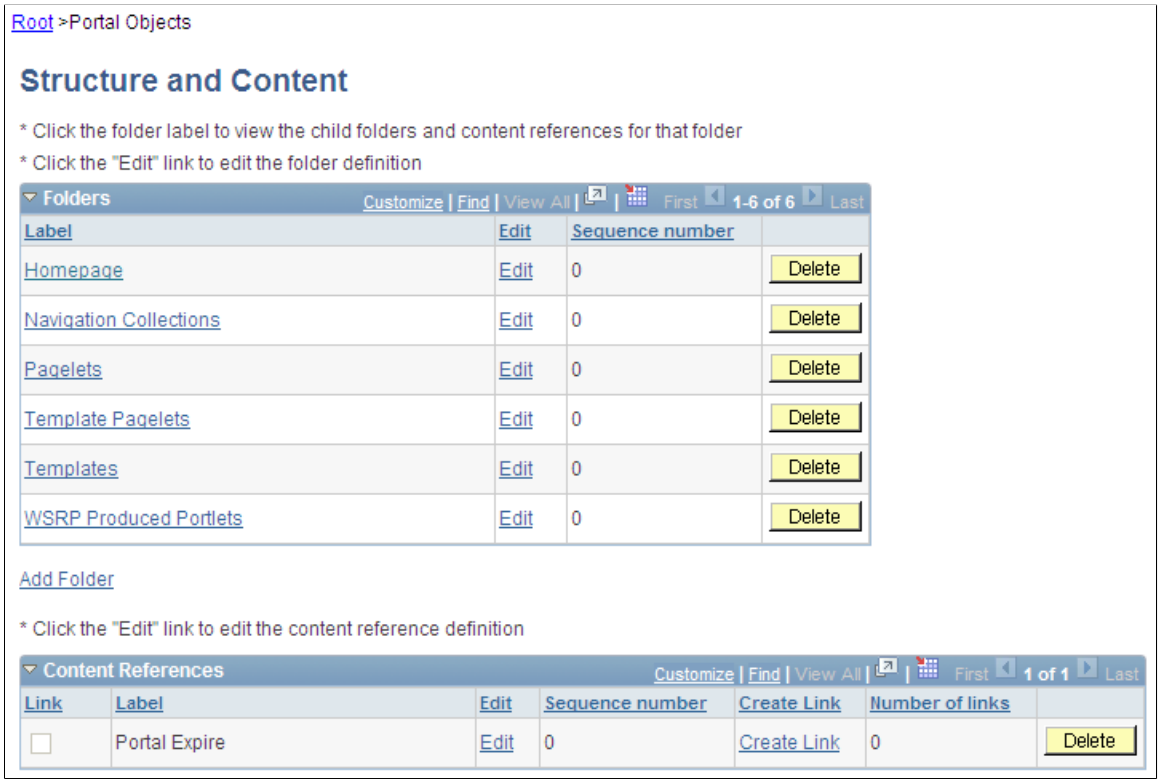
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Viewing Portal Objects Folder Settings

Access the Structure and Content page (Select PeopleTools, Portal, Structure and Content. In the Folders group box, click the Edit link for Portal Objects).

Image: Structure and Content page for the Portal Objects folder

The following screenshot illustrates viewing portal objects folder settings in the Structure and Content page for the Portal Objects folder.



Understanding Templates

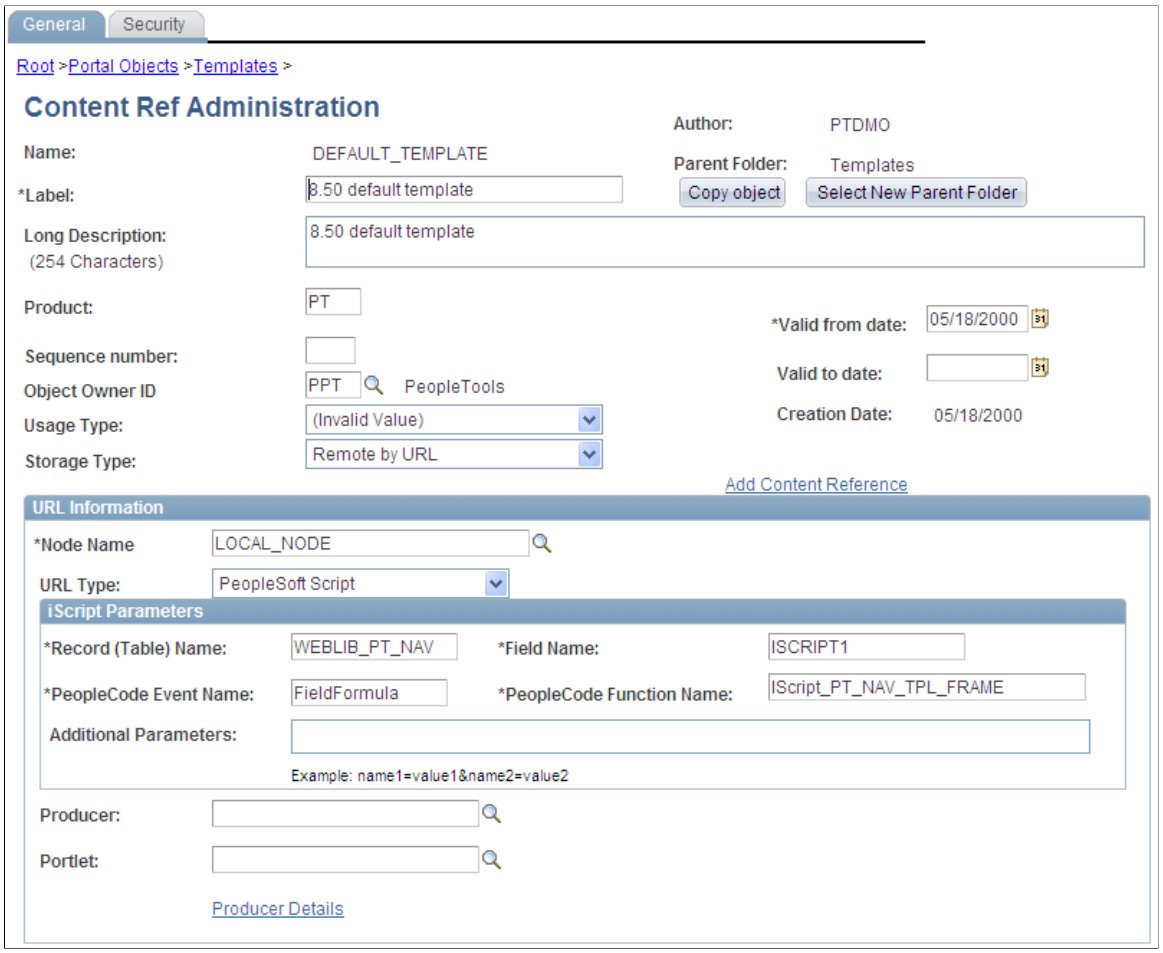
Templates are stored in the HTML catalog and are edited through the portal administration pages. For the portal to process templates properly, they must be referenced in the Templates folder within the Portal Objects folder. Storing templates in the HTML catalog enables them to be upgraded like other objects.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Templates are stored in the HTML catalog according to the following naming convention: Template names always begin with the literal *PR_*. Here's an example name: PR*_PORTALNAME_CONTENT REFERENCE ID*

Image: Example of a content reference definition for a portal template

The following screenshot illustrates example of content reference definition for a portal template:



Templates can be larger than the maximum size permitted by the HTML catalog. Such templates are automatically divided into several files in the HTML catalog. In this case, a letter is appended to each HTML object that makes up the template, as shown here:

PR_EMPLOYEE_123456789_A

PR_EMPLOYEE_123456789_B

Related Links

Understanding Portal Templates

Adding and Editing Templates

Access the Content Ref Administration page (Select PeopleTools, Portal, Structure and Content. Then click these links in succession: Portal Objects, Templates).

To add a new template, click Add Content Reference in the Content References region at the bottom of the Content Ref Administration page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To edit an existing template, click the Edit link next to the template that you want to edit in the Content References region at the bottom of the Content Ref Administration page.

See Defining Content References.

The fields for adding a dynamic template are the same as for any content reference, except for additional iScript parameter fields. These fields are enabled only if the storage type is set to *Remote by URL.*

If the URL Type field is set to *PeopleSoft Script,* the iScript Parameters group box appears. Select the record name, PeopleCode event name, field name, and PeopleCode function name that stores the iScript that the content reference accesses to build the template.

Related Links

Defining Folders

Defining Content References

Deleting Templates

You delete templates the same way that you delete folders and content references.

See Deleting Folders and Content References.

Note: Do not delete the DEFAULT_TEMPLATE portal template. This template has special significance to your portal and must always be available. Deleting this template could make your portal inoperative. Oracle recommends that you not delete any template unless you are fully aware of how it's used within the portal.

Reviewing Menu Item Information

This section lists common elements used in the section and discusses how to:

- View menu item details.
- View menu folder structure information.
- View menu security details.

Common Elements Used in This Section

Menu Item Displays the menu item ID. The menu item label is displayed to the right.

Content Provider Name Displays the database node that contains the PeopleSoft page.

Product (Optional) Displays text that is used to tag menu items for inquiry.

Note: Each PeopleSoft product line has a naming standard. For example, PAPP is always used for PeopleSoft Interaction Hub.

Menu Path Displays the navigation path for this menu item.

Hidden If this check box is selected, then the menu item is hidden from portal menu navigation. However, you can still access the

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	content by using a link that is incorporated on a target page or pagelet.
Is Public	If this check box is selected, the menu item is available for all users. No data appears in the Accessible Permissions, Accessible Roles, and Accessible Users scroll areas.
Auth Access (author access)	If this check box is selected, the author can access the menu item regardless of how other permissions are set.

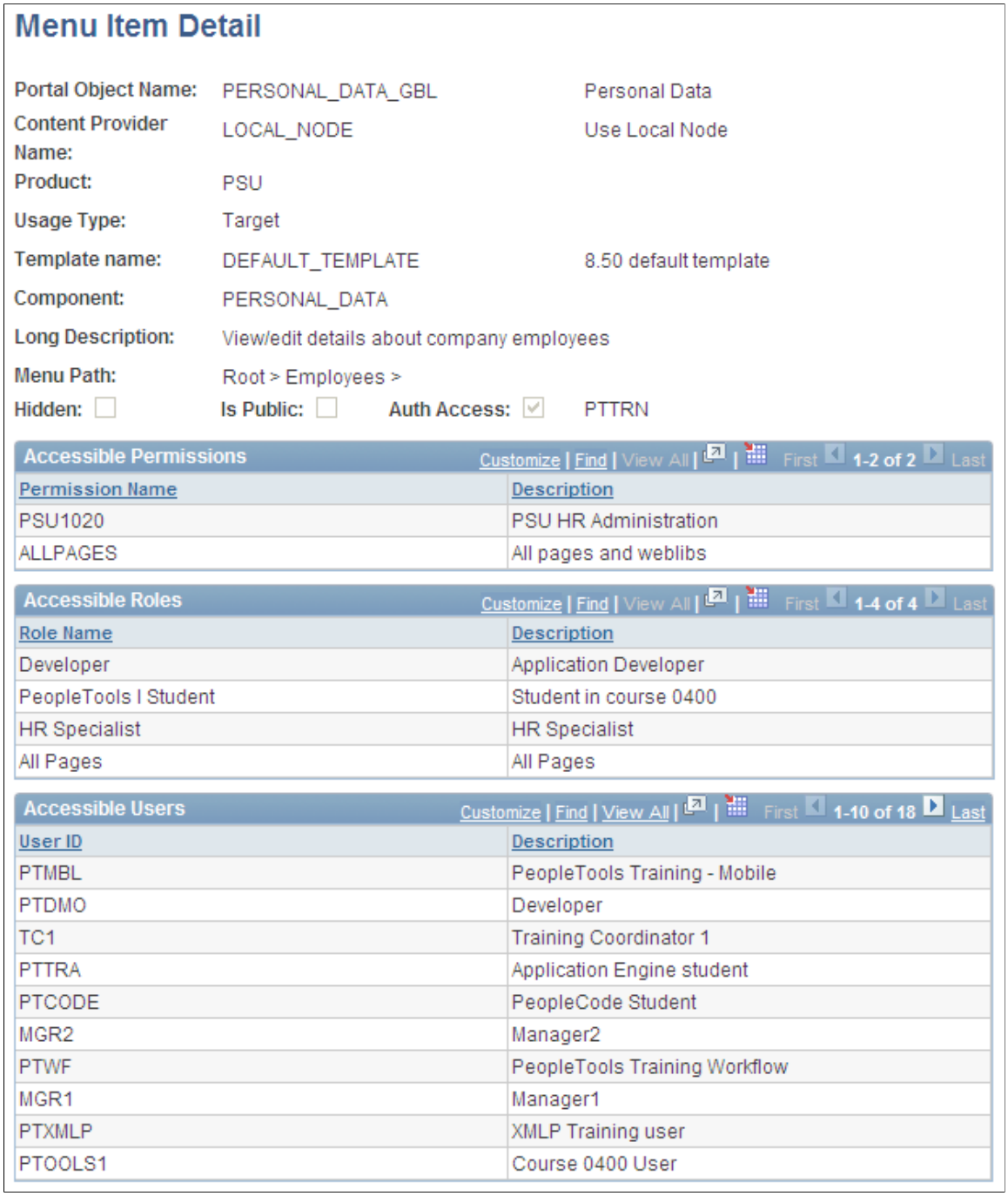
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Viewing Menu Item Details

Access the Menu Item Detail page (PeopleTools, Portal, View Menu Item Detail).

Image: Menu Item Detail page

The following screenshot illustrates viewing menu item details.



Menu Item Detail

Portal Object Name:	PERSONAL_DATA_GBL	Personal Data
Content Provider Name:	LOCAL_NODE	Use Local Node
Product:	PSU	
Usage Type:	Target	
Template name:	DEFAULT_TEMPLATE	8.50 default template
Component:	PERSONAL_DATA	
Long Description:	View/edit details about company employees	
Menu Path:	Root > Employees >	
Hidden: ☐	Is Public: ☐	Auth Access: ☑ PTTRN

Accessible Permissions — Customize | Find | View All | First 1-2 of 2 Last

Permission Name	Description
PSU1020	PSU HR Administration
ALLPAGES	All pages and weblibs

Accessible Roles — Customize | Find | View All | First 1-4 of 4 Last

Role Name	Description
Developer	Application Developer
PeopleTools I Student	Student in course 0400
HR Specialist	HR Specialist
All Pages	All Pages

Accessible Users — Customize | Find | View All | First 1-10 of 18 Last

User ID	Description
PTMBL	PeopleTools Training - Mobile
PTDMO	Developer
TC1	Training Coordinator 1
PTTRA	Application Engine student
PTCODE	PeopleCode Student
MGR2	Manager2
PTWF	PeopleTools Training Workflow
MGR1	Manager1
PTXMLP	XMLP Training user
PTOOLS1	Course 0400 User

Use this page to review the portal registry information for a menu item. The details that appear include the exact users and the roles and permission lists that allow access to a given menu item.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Usage Type	Displays the type of object that is registered in the portal as a content reference. Possible values are *Target, Pagelet, Frame Template, HTML Template,* and *Homepage Tab.*
Template Name	Displays the template that controls how content appears on the page. The default template is used except for special circumstances.
Component, iScript, and URL Type	Information that appears depends on the URL type that is entered on the Content Ref Administration page. Possible URL type values are *Non-PeopleSoft URL, PeopleSoft Component, PeopleSoft Generic URL, PeopleSoft Script, PeopleSoft Worklist URL,* and *Remote Portlet.* The availability of these values depends on the selected usage type.
Long Description	Displays the text that appears when you move the cursor over the menu navigation link.

Note: The Accessible Permissions, Accessible Roles, and Accessible Users scroll areas are populated only if the access type is set to *permission list.*

See Defining Content References.

Accessible Permissions

This section displays the permission list name and description for all permission lists that can access this menu item.

Note: If the component or iScript is using a local node with no extra parameters, the permission lists are automatically populated based on the menu object component security.

Accessible Roles

Displays the role name and description for all roles that can access this menu item.

Accessible Users

Displays the user ID and description for all users that can access this menu item.

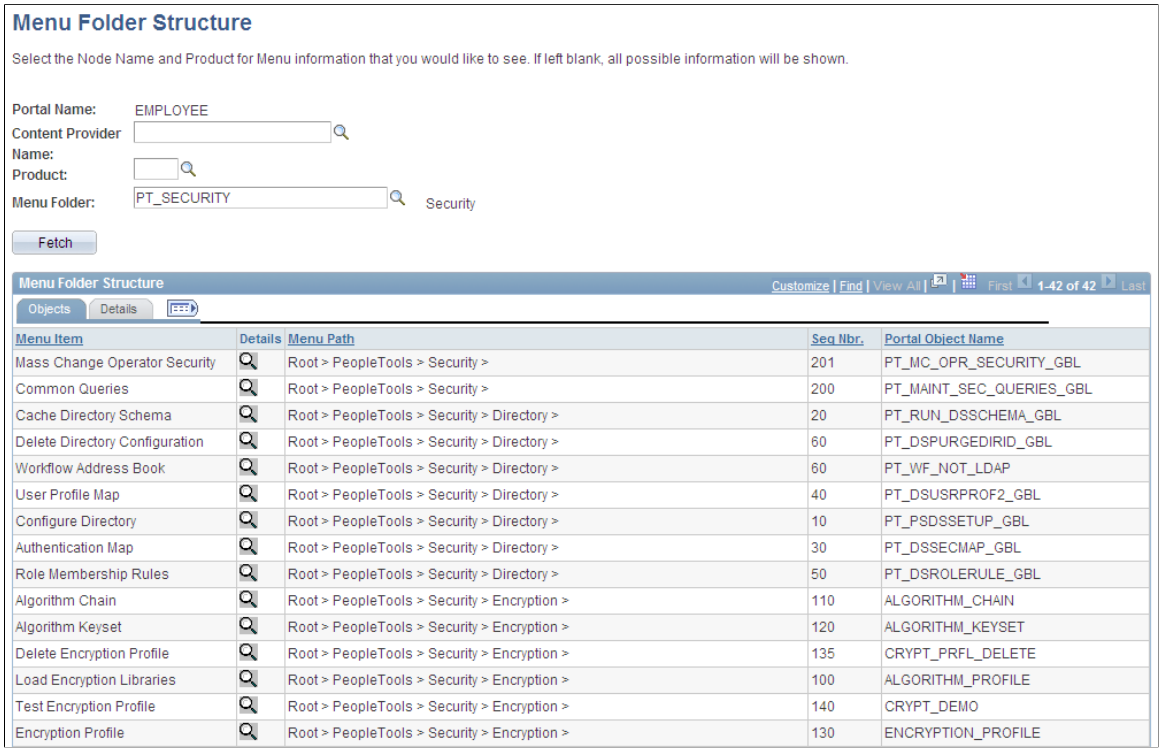
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Viewing Menu Folder Structure Information

Access the Menu Folder Structure page (PeopleTools, Portal, View Menu Folder Structure).

Image: Menu Folder Structure page: Objects tab

The following screenshot illustrates viewing information of menu folder structure on the Objects tab.



Menu Item	Details	Menu Path	Seq Nbr.	Portal Object Name
Mass Change Operator Security		Root > PeopleTools > Security >	201	PT_MC_OPR_SECURITY_GBL
Common Queries		Root > PeopleTools > Security >	200	PT_MAINT_SEC_QUERIES_GBL
Cache Directory Schema		Root > PeopleTools > Security > Directory >	20	PT_RUN_DSSCHEMA_GBL
Delete Directory Configuration		Root > PeopleTools > Security > Directory >	60	PT_DSPURGEDIRID_GBL
Workflow Address Book		Root > PeopleTools > Security > Directory >	60	PT_WF_NOT_LDAP
User Profile Map		Root > PeopleTools > Security > Directory >	40	PT_DSUSRPROF2_GBL
Configure Directory		Root > PeopleTools > Security > Directory >	10	PT_PSDSSETUP_GBL
Authentication Map		Root > PeopleTools > Security > Directory >	30	PT_DSSECMAP_GBL
Role Membership Rules		Root > PeopleTools > Security > Directory >	50	PT_DSROLERULE_GBL
Algorithm Chain		Root > PeopleTools > Security > Encryption >	110	ALGORITHM_CHAIN
Algorithm Keyset		Root > PeopleTools > Security > Encryption >	120	ALGORITHM_KEYSET
Delete Encryption Profile		Root > PeopleTools > Security > Encryption >	135	CRYPT_PRFL_DELETE
Load Encryption Libraries		Root > PeopleTools > Security > Encryption >	100	ALGORITHM_PROFILE
Test Encryption Profile		Root > PeopleTools > Security > Encryption >	140	CRYPT_DEMO
Encryption Profile		Root > PeopleTools > Security > Encryption >	130	ENCRYPTION_PROFILE

Use this page to view a list of content references for a selected portal registry folder. You see the contents of one or several menu folders.

Portal Name — Displays the name of the portal that you are currently using.

Content Provider, Name, Product, and Menu Folder — Select a combination of criteria for filtering the results. If any of the fields are left blank, all possible information for those fields are shown.

Objects Tab

Select the Objects tab.

Menu Item — Displays the menu item description.

Click to view more detail information on the Menu Item Detail page.

Seq Nbr (sequence number) — Displays where the menu item is located in its menu folder. If this field is blank, the menu item is added to the top of the list of content references within its folder and displayed in alphabetical order.

Portal Object Name — Displays the menu item ID.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Details Tab

Select the Details tab.

Image: Menu Folder Structure page: Details tab

The following screenshot displays content of details tab in the Menu Folder Structure page.

Menu Folder Structure — Customize | Find | View All | First 1-42 of 42 Last

Objects | Details

Menu Item	Product	Content Provider Name	Component	iScript	Hidden	Is Public	Auth Access	User ID
Mass Change Operator Security	PT	LOCAL_NODE	MC_OPR_SECURITY		☐	☐	☐	
Common Queries	PT	LOCAL_NODE	MAINT_SEC_QUERIES		☐	☐	☐	
Cache Directory Schema	PT	LOCAL_NODE	RUN_DSSCHEMA		☐	☐	☐	
Delete Directory Configuration	PT	LOCAL_NODE	DSPURGEDIRID		☐	☐	☐	
Workflow Address Book	PT	LOCAL_NODE	PT_WF_NOT_DSCFG		☐	☐	☐	
User Profile Map	PT	LOCAL_NODE	DSUSRPROF2		☐	☐	☐	
Configure Directory	PT	LOCAL_NODE	PSDSSETUP		☐	☐	☐	
Authentication Map	PT	LOCAL_NODE	DSSECMAP		☐	☐	☐	
Role Membership Rules	PT	LOCAL_NODE	DSROLERULE		☐	☐	☐	
Algorithm Chain	PT	LOCAL_NODE	ALGORITHM_CHAIN		☐	☐	☑	QEDMO
Algorithm Keyset	PT	LOCAL_NODE	CRYPT_KEYSET		☐	☐	☑	QEDMO
Delete Encryption Profile	pt	LOCAL_NODE	CRYPT_PRFL_DELETE		☐	☐	☑	QEDMO
Load Encryption Libraries	PT	LOCAL_NODE	ALGORITHM_PRFL		☐	☐	☑	QEDMO
Test Encryption Profile	PT	LOCAL_NODE	CRYPT_DEMO		☐	☐	☑	QEDMO
Encryption Profile	PT	LOCAL_NODE	ENCRYPTION_PRFL		☐	☐	☑	QEDMO

User ID If the Auth Access check box is selected, this field displays the author's user ID.

Viewing Menu Security Details

Access the Menu Security page (PeopleTools, Portal, View Menu Security).

Image: Menu Security page: Objects tab

The following screenshot illustrates viewing menu security details on the Objects tab of the Menu Security page.

Menu Security

Select the level of Object security and Folder visibility for Menu information that you would like to see. Please note that inquiries which return a large amount of rows may take a minute or two.

Object Security
- User — User ID: QEDMO
- Role
- Permission List

Folder Visibility
- Show All
- Choose Top Level — Menu Item: PORTAL_BASE_DATA
- Show Top Level Only

Fetch

Menu Security — Customize | Find | View All | First 1-39 of 39 Last

Objects | Details

Menu Item	Details	Menu Path	Seq Nbr.	Portal Object Name
Portal Expire		Root > Portal Objects >		PORTAL_EXPIRE
My Page		Root > Portal Objects > Homepage > Tabs >		DEFAULT
Design Helper		Root > Portal Objects > Navigation Collections > XML Publisher > Complete Setup >	18	ADMN_S2006011316411063588366 34
Report Category		Root > Portal Objects > Navigation Collections > XML Publisher > Complete Setup >	17	ADMN_S2006011316402166881313 51
Content Library		Root > Portal Objects > Navigation Collections > XML Publisher > Define XMLP Reports >	9	ADMN_S2006011316372999789422 28
Report Definition		Root > Portal Objects > Navigation Collections > XML Publisher > Define XMLP Reports >	5	ADMN_S2006011316370944352549 82
Translations		Root > Portal Objects > Navigation Collections > XML Publisher > Define XMLP Reports >	10	ADMN_S2006011316374481691946 16
Data Source		Root > Portal Objects > Navigation Collections > XML Publisher > Register Data Sources >	4	ADMN_S2006011316350943449201 94

Use this page to view a list of folders or content references that a user ID, role, or permission list can access.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: The Seq Nbr (sequence number) column is hidden in this illustration, and because of the large number of objects, not all are shown.

Object Security

Select the level of object security for the menu information that you want to see.

User, Role, and Permission List Select a type of object security, and then select an object.

You can view the complete menu folder structure that is available in the navigation (or hidden, but accessible from another page) for the selected security value.

Warning! To prevent unexpected errors and inordinately long retrieval times, do *not* select the *ALLPAGES* permission list in conjunction with the *Show All* folder option. Always narrow your folder options if you select the *ALLPAGES* permission list.

Folder Visibility

Select the level of visibility for the menu information that you want to see.

Show All Select to view all objects.

Note: To avoid unexpected errors and inordinately long retrieval times, Oracle recommends that you not select this option if you select the *ALLPAGES* permission list in the Object Security section.

Choose Top Level Select a level-one folder. The Menu Security grid displays all content references and their complete paths below this level that are accessible by the selected object security level.

Show Top Level Only Select to display only the level-one folders that are accessible by the object security type.

Permission Lists

If either the Role object security option or the Show Top Level Only folder visibility option is selected, then this scroll area appears, displaying the permission lists that are associated with each menu item.

Publishing Pagelets to Other Portals

This section discusses how to:

- Publish a pagelet.
- Publish pagelets to multiple portals.

Note: The Publish Pagelets component can be used with any pagelets including Pagelet Wizard pagelets.

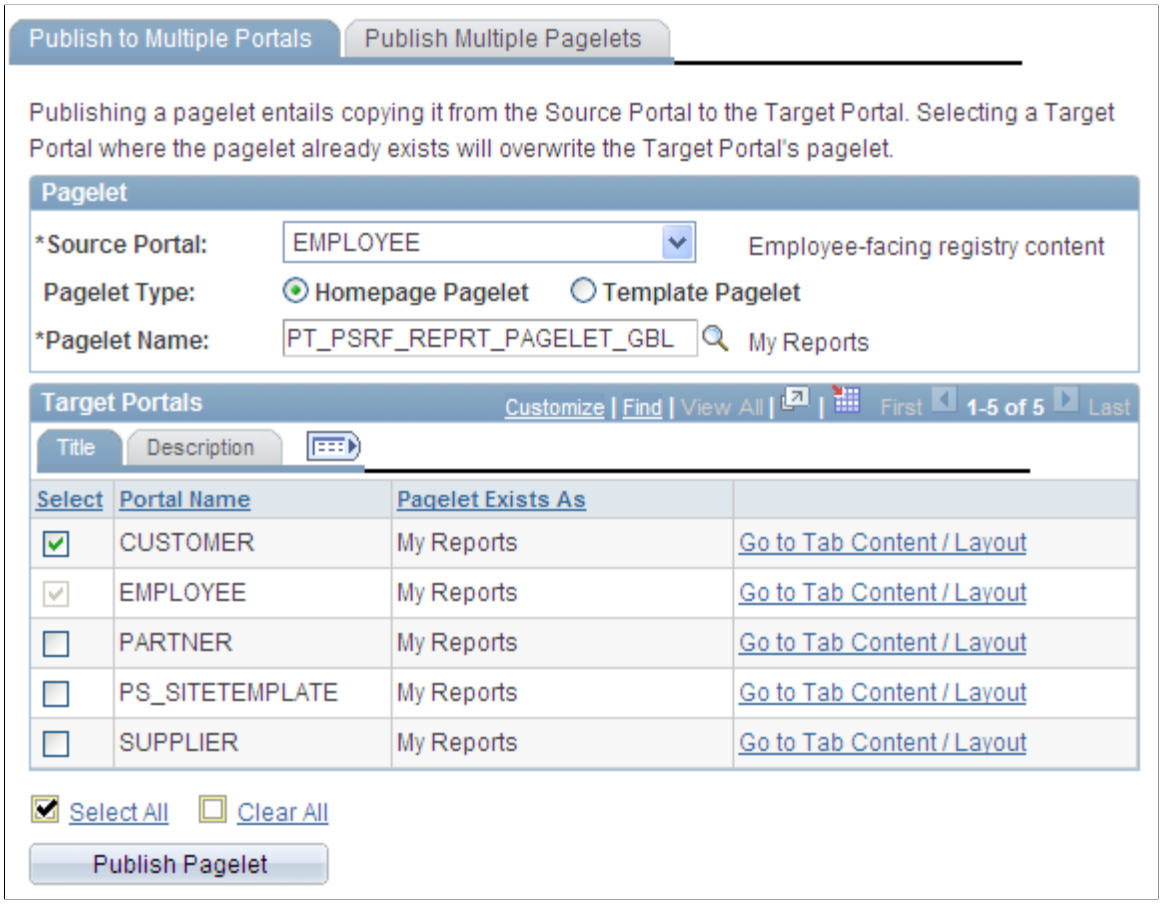
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Publishing Pagelets to Multiple Portals

Access the Publish to Multiple Portals page. (Select PeopleTools, Portal, Portal Utilities, Publish Pagelets, Publish to Multiple Portals.)

Image: Publish to Multiple Portals page

The following screenshot illustrates how to publish pagelets by accessing Publish to Multiple Portals page.



Pagelet

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 signed in.

Pagelet Type

Select the type of pagelet that you want to copy. Options are:

*Homepage Pagelet:*A homepage pagelet is a version of a pagelet that is available for use on the portal homepage and can be configured by the end user.

*Template Pagelet:*A template pagelet is a version of a pagelet that can be added to a PeopleSoft template. These pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

when Context Manager or WorkCenter pages are used to display the template pagelet.

Pagelet Name Select a pagelet that exists in the selected Source Portal registry. This is the pagelet content reference that you want to publish to one or more other portal registries that reside in the database where you are currently signed 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 that you 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 Technology News pagelet is selected for copying from the EMPLOYEE source portal to the CUSTOMER target portal. The Pagelet Exists As field shows that this pagelet also already exists in the CUSTOMER target portal. If you copy the pagelet to the CUSTOMER portal, the pagelet definition in the CUSTOMER portal is overwritten by the pagelet definition in the EMPLOYEE source portal.

Go to Tab Content/Layout Click to display the Tab Content and Tab Layout pages where you can specify display settings for the pagelet.

Publish Pagelet Select to copy the pagelet definition from the source portal and publish it in the selected target portals.

Content reference are created in the registries of all selected target portals by means of 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 generated dynamically.

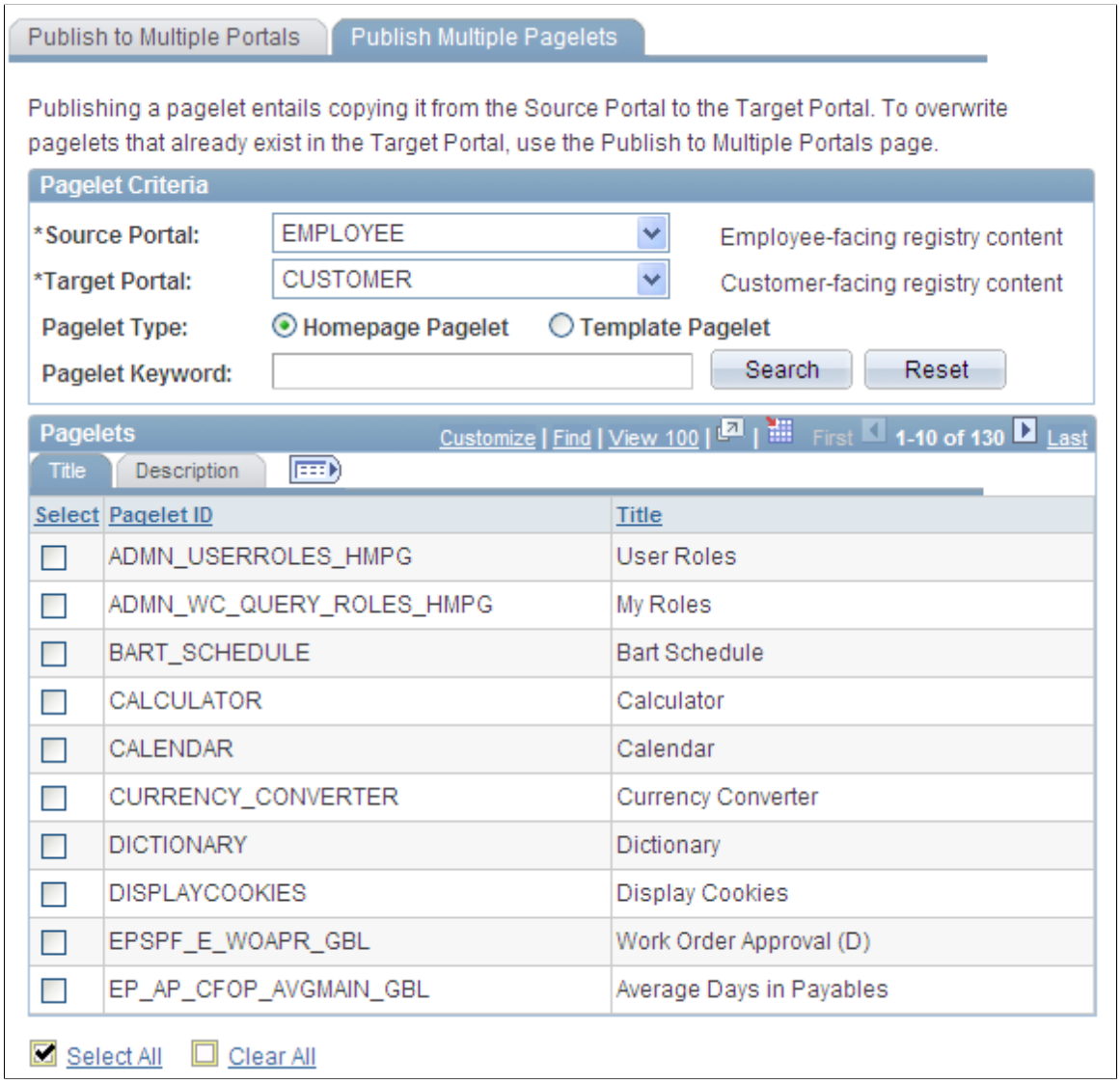
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Publishing Multiple Pagelets to a Portal

Access the Publish Multiple Pagelets page. (Select PeopleTools, Portal, Portal Utilities, Publish Pagelets, Publish Multiple Pagelets.)

Image: Publish Multiple Pagelets page

The following screenshot illustrates how to publish pagelets by accessing the Publish Multiple Pagelets to a Portal 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 signed 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 signed in.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 that you want to copy. Options are:

Homepage Pagelet: A homepage pagelet is a version of a pagelet that is available for use on the portal homepage and can be configured by the end user.

Template Pagelet: A template pagelet is a version of a pagelet that can be added to a PeopleSoft template. These pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful when Context Manager or WorkCenter pages are used to display the 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 to any selected target portals.

The portal registry entries for the content references are created in the target portal registry by means of 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 generated dynamically.

Running Portal Utility Processes

This section provides an overview of portal utility processes and discusses how to:

- Run the Synchronize Collection Security process.
- Run the Delete Empty Folders process.
- Run the Clean Portal Project process.

Understanding Portal Utility Processes

This section discusses:

- The Synchronize Collection Security process.
- The Delete Empty Folders process.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The Clean Portal Project process.

The Synchronize Collection Security Process

The Synchronize Collection Security (PTPP_SCSYNC) Application Engine process updates security for published custom navigation pages and navigation pagelets by adding permission lists, roles, or both from the content references and folders in the navigation collection to the content references for the published navigation page or pagelet.

The Synchronize Collection Security process runs for navigation collection publications according to the Security Access options that are set for the publication on the Publish Collection page. The process runs for navigation collection publications that are published with the *Allow Collection Sync* option selected. The process is not run for navigation collection publications that are published with the *Do Not Allow Collection Sync* option selected. The process also removes extraneous security settings for navigation collections that were published with the *Public Access* option selected.

Run this process when you initially publish or change a navigation page or pagelet with the *Allow Collection Sync* option selected or after security updates to the items within the collection.

See Publishing Navigation Collections.

The Delete Empty Folders Process

The Delete Empty Folders Application Engine process (PTPP_FD_DEL) deletes portal registry folder references that do not contain child folders or content. The process does not delete empty navigation collection folder placeholders or the My Favorites folder. The user ID running the process must have been assigned the security role of Portal Administrator.

Run this process 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. You should also run this process after copying a *deleted* portal project.

The Clean Portal Project Process

The Clean Portal Project Application Engine process (PTPP_CPPROJ) removes unnecessary and potentially harmful data from a portal project definition before you copy it to the PeopleSoft Interaction Hub. Run this process as a part of an upgrade or implementation when you want to load navigation from a content provider into the PeopleSoft Interaction Hub.

The Clean Portal Project process removes common objects from the content provider portal project which, because they are common, also exist in the PeopleSoft Interaction Hub. Run this process on the portal project before copying it to the PeopleSoft Interaction Hub so that you don't override these common objects configured for use with the PeopleSoft Interaction Hub with the common objects that were configured in the content provider.

The Clean Portal Project process ensures that you copy only the navigation objects that you think are necessary, such as registry structures, templates, and permissions.

Warning! Navigation collections cannot be rendered in the PeopleSoft Interaction Hub if the content provider registry is hosted by a nonlocal 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 nonlocal node. If you want to render a content provider navigation collection page or pagelet in the PeopleSoft Interaction Hub, the content provider's registry must be hosted by a local node in the content provider database.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Administering Portal DefinitionsUnderstanding Portal Administration.

Running the Synchronize Collection Security Process

Access the Synchronize Collection Security page. (Select PeopleTools, Portal, Portal Utilities, Sync Collection Security.)

Image: Synchronize Collection Security page

This example illustrates the fields and controls on the Synchronize Collection Security page. You can find definitions for the fields and controls later on this page.

Request Parameters

Portal Name

Select the portal for which you want to synchronize security for published navigation collections.

Collection Name

Select the navigation collection for which you want to synchronize security. Only navigation collections created for the selected portal are available for selection.

Leave this field blank if you want to synchronize security for all 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.

Related Links

Publishing Navigation Collections

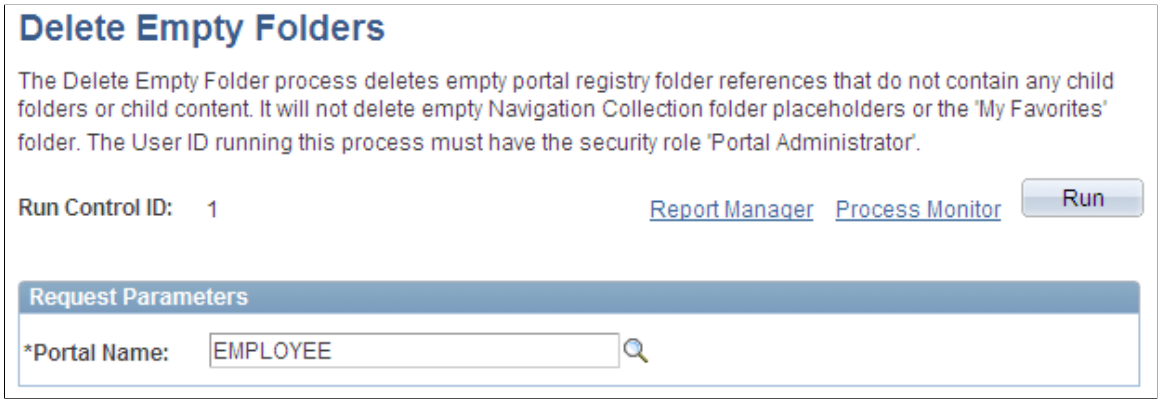
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Running the Delete Empty Folders Process

Access the Delete Empty Folders page. (Select PeopleTools, Portal, Portal Utilities, Delete Empty Folders.)

Image: Delete Empty Folders page

This example illustrates the fields and controls on the Delete Empty Folders page. You can find definitions for the fields and controls later on this page.



Request Parameters

Portal Name Select the portal for which you want to delete portal registry folder references that do not contain child folders or content.

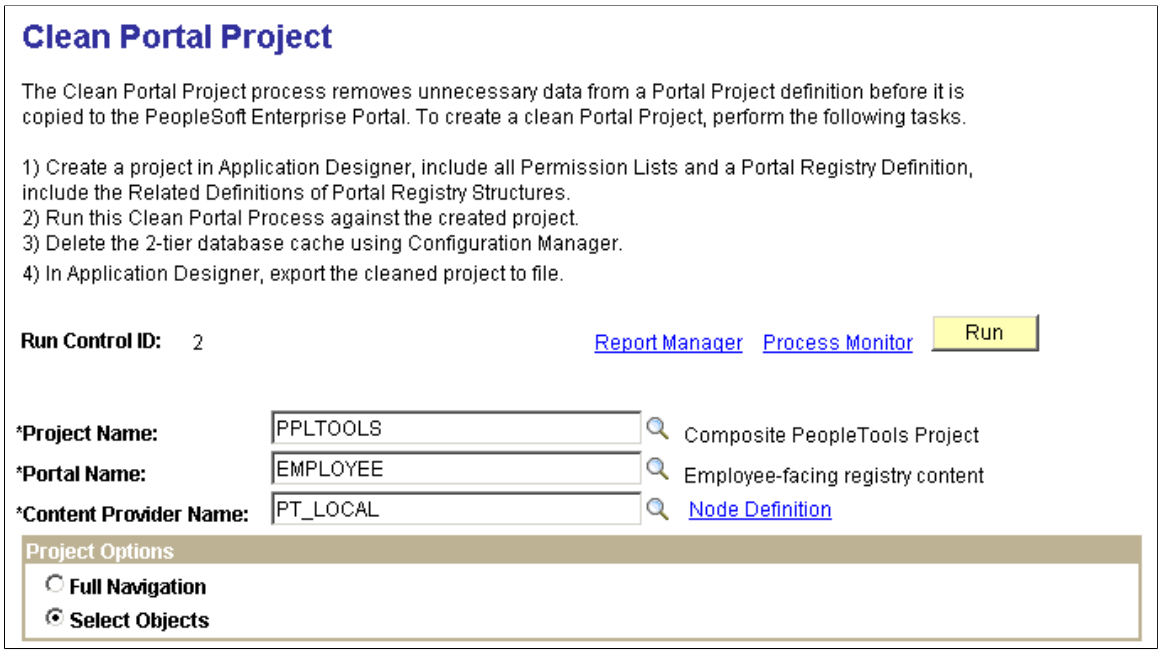
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Running the Clean Portal Project Process

Access the Clean Portal Project page. (Select PeopleTools, Portal, Portal Utilities, Clean Portal Project.)

Image: Clean Portal Project page (1 of 2)

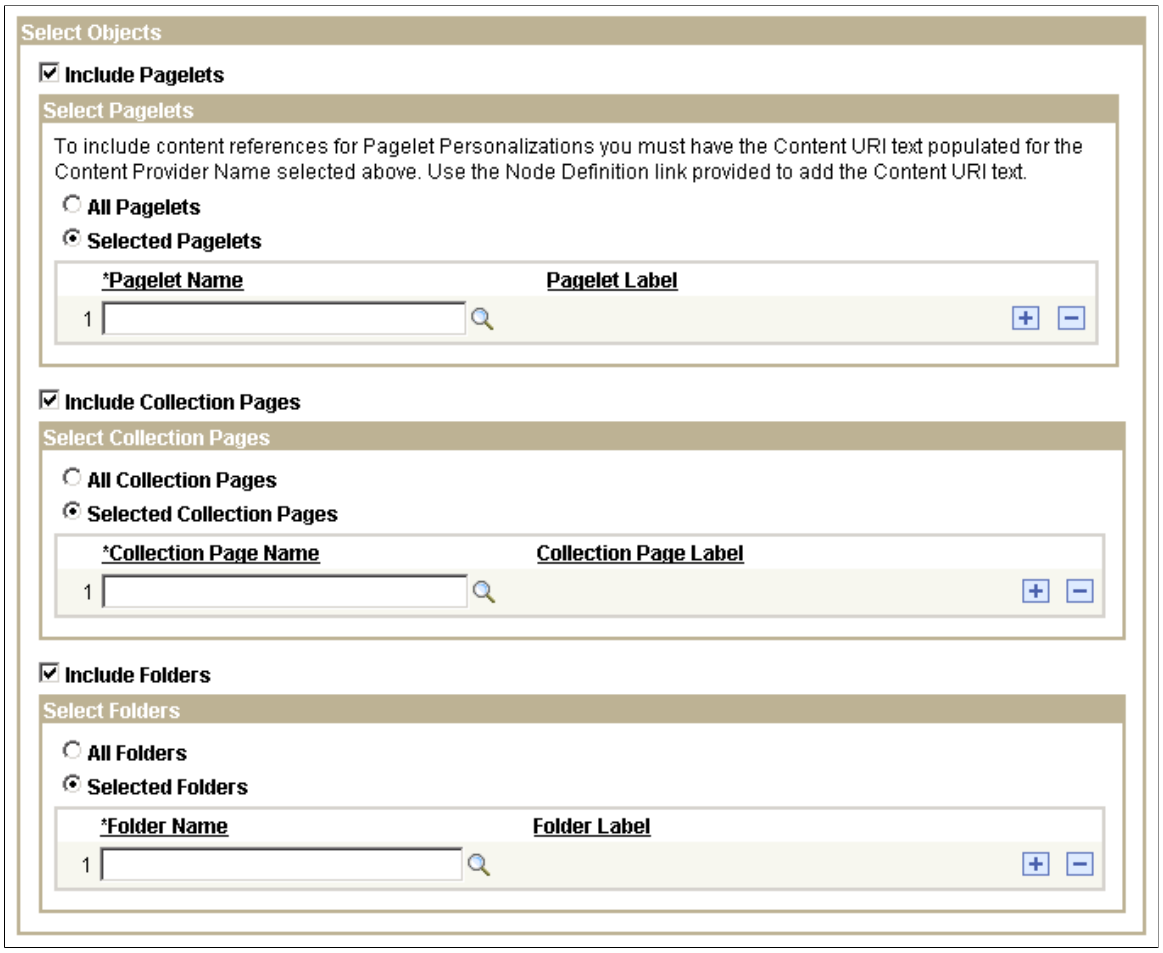
This example illustrates the fields and controls on the Clean Portal Project page. You can find definitions for the fields and controls later on this page.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Image: Clean Portal Project page - Select Objects group box (2 of 2)

This example illustrates the fields and controls on the Select Objects group box of the Clean Portal Project page. You can find definitions for the fields and controls later on this page.



Project Name

Select the portal project that you want to clean for export into the PeopleSoft Interaction Hub. Projects available for selection are those in the database that contain at least one registry structure.

Portal Name

Select the portal that contains the registry structures that you want to copy. Available portal names are derived from the portal project that you selected, as well as from 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 that you want to include in your portal project. Available node names are derived from the portal project that you selected.

Node Definition

Select to access the Node Definitions - Portal Content page. If you select the Include Pagelets option, 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 that you copy over to the PeopleSoft Interaction Hub. Access the Node Definitions - Portal Content page to enter the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

content URI text for the content provider node name selected on this page.

When you click this link, you are prompted to save or cancel any changes that you made in the component, and you are then transferred to the Node Definitions - Portal Content page.

See 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 appears, enabling you to select the type of navigation that you want to include in your portal project.

Use this option if you don't want to copy and maintain the full content provider navigation structure.

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 appears, enabling you to specify which pagelets you want to be processed.

The actual published navigation collection pagelet content references are included for copying to the PeopleSoft Interaction Hub. The navigation collection definitions are not included.

Pagelet personalization pages are included in the portal project if the content provider node name contains the correct URL. Click 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 appears, enabling you to specify which navigation collection pages you want to be processed. The actual published navigation collection pages are included for copying to the PeopleSoft Interaction Hub. 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 appears, enabling you to specify which folders you want to be processed.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting a folder for inclusion in the process also includes all of its children and parents for copying into the PeopleSoft Interaction Hub.

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 be processed.

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 be processed.

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 be processed.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 6

Using PeopleTools Branding Features

Understanding PeopleTools Branding

This topic provides an overview of PeopleTools branding and discusses:

- Branding definitions.
- Style sheets and JavaScript objects.
- Branding process overview.

Overview of PeopleTools Branding

PeopleTools provides branding features that are powerful and flexible, allowing you to manage the look and feel of any PeopleSoft application. With PeopleTools, many common branding tasks, including managing definitions and objects associated with the overall site style are performed online. In addition, other tasks such as maintaining style sheets, images, and other objects can be performed either online or by using Application Designer.

With PeopleTools branding, you can apply branding definitions to:

- The entire system.
- Individual portals.
- Users by role or permission list.
- Individual classic components.
- Fluid components by node.

The key branding definition is the theme, which can be applied system wide, to specific portals, and by specific user attribute. A theme definition includes:

- Homepage header, footer (optional), and target page header (optional).
- Default theme style sheet for classic components (optional) if necessary to override the style sheet of the theme family.
- List of additional skins for classic components (optional).
- Mapped theme style sheets for fluid components by node (optional).
- Global theme style sheet for fluid components if necessary to override the style sheet of the theme family (optional).

At the system level, branding settings include:

- Selection of the default theme family (which includes the default theme and compatible default style sheet for classic components)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Selection of a navigation collection for custom homepage tabs (optional).
- Additional style sheets and JavaScript objects for all classic components (optional).
- A default style sheet for all fluid components (optional).
- The branding application package and class.

At the portal registry level, additional settings can be specified, some of which override the system settings:

- Default theme.
- Default skin style sheet.
- Default custom tabs collection (and a default homepage tab).
- Default fluid homepage.

At the user attribute level, additional settings can be specified, some of which override the system settings and the portal registry settings:

- Theme.
- Skin style sheet.
- Custom tabs collection (and a default homepage tab).
- Default fluid homepage.

In addition to these system, portal, and user attribute settings, different branding can be applied to individual components. In Application Designer, the component's developer can specify style sheets and JavaScript objects as component properties. Through the browser, the portal administrator can specify additional custom style sheets and JavaScript objects as component properties. (These capabilities are in addition to existing features in Application Designer that allow the component's developer to apply a style sheet to each page definition, and to apply style classes to fields and field labels.)

Branding Definitions

PeopleTools allows you to maintain the following definitions used with branding:

- Branding system options for system-wide settings.
- Branding objects: HTML, JavaScript, style sheets, and images. (Definitions for these branding object types can also be created and maintained in Application Designer.)
- Pagelet branding, which allows you to modify display attributes of individual pagelets.
- Component branding, which allows you to add custom style sheets and JavaScript objects to individual components. These custom objects are in addition to any objects specified by the component's developer in Application Designer.
- Header and footer definitions, which are incorporated in theme definitions.
- Theme definitions, which include headers, a footer, and default theme style sheets for classic and fluid components.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Theme assignments, which allow you to specify themes and other options at the portal level and by user attribute.
- Branding system element types, which allow you to define the elements that can be used to construct header and footer definitions.
- Branding user attribute types, which allow you to define the user attributes that can be used for theme assignments.
- Navigation collections designed for use as custom homepage tabs.

Style Sheets and JavaScript Objects

Levels at Which Style Sheets Can Be Applied

Style sheets are associated with different definitions throughout the system. Certain definitions require a style sheet, whereas one or more style sheets are optional for others. Moreover, different style sheet types are allowed with different definitions.

The following table summarizes the locations where style sheets can be attached for *classic components*:

Level	***#***	***Comments***
System default	1	Standard or free form. Practically speaking, the system default style sheet should be a standard style sheet that is compatible with the selected theme family. The system style sheet should be comprehensive and define both layout and style aspects of all user interface elements. Typically, you would select one of the three delivered layout style sheets: • PSSTYLEDEF • PSSTYLEDEF_SWAN • PSSTYLEDEF_TANGERINE
Additional system style sheets	0 – *n*	Standard or free form.
Component	0 – *n*	Standard or free form. Component-level style sheets can be added by the component's developer, by the portal administrator, or both.
Page	0 or 1	Standard or free form. Page-level style sheets can be specified by the component's developer only.
Theme	0 or 1	Free form only.
Skin	0 – *n*	Free form only. While multiple skins can be associated with a theme definition, only one skin (or no skins) can be in effect per assignment (portal or user attribute).

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The following table summarizes the locations where style sheets can be attached for *fluid components*:

Level	***#***	***Comments***
System default*	0 or 1	Standard or free form. If no style sheet is defined at the system level, PSSTYLEDEF_FMODE is used by default.
Component* or page*	0 – *n*	Standard or free form. Component-level and page-level style sheets can be added by the component's developer through PeopleCode.
Theme (global override)	0 or 1	Standard or free form.

* Theme definitions include the optional capability to associate mapped theme style sheets to base style sheets on a per node basis. You can create 0 – *n* mappings. If a base style sheet with a mapped theme style sheet is specified at the system, component, or page level, then the mapped theme style sheet will be loaded directly beneath the base style sheet at the specified level.

Loading Sequence

At run time, the generated HTML includes style sheets and JavaScript objects in a specified sequence. Style sheets (or more specifically, the specific style classes defined in those style sheets) override any style class definitions of the same name loaded earlier in the sequence. Therefore, you can use theme and skin style sheets to overlay your branding look on top of styles defined at the system or component level.

Classic Components

For classic components, style sheets and JavaScript objects are loaded in the following order:

1. PSSTYLEDEF_REQ (automatically loaded by default for all classic components).*
2. System default style sheet.*
3. Additional system-level style sheets in the order defined.
4. Additional system-level JavaScript objects in the order defined.
5. Component-level style sheets in the component developer list in the order defined.
6. Component-level style sheets in the portal administrator list in the order defined.
7. Component-level JavaScript objects in the developer list in the order defined.
8. Component-level JavaScript objects in the portal administrator list in the order defined.
9. Page-level style sheet.
10. Theme default style sheet.
11. Skin style sheet.

* A required style sheet that is always loaded. All other style sheets are optional.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Fluid Components

For fluid components, style sheets and JavaScript objects are loaded in the following order:

1. System default style sheet* (and its mapped theme style sheet if specified).
2. Component-level or page-level style sheets (and their mapped theme style sheets if specified) added by the component developer through PeopleCode.
3. Component-level JavaScript objects added by the component developer through PeopleCode.
4. Theme-level global override style sheet.

* A required style sheet that is always loaded. All other style sheets are optional.

Branding Process Overview

Use the following process when developing and applying your own branding:

1. Determine your branding requirements in collaboration with your marketing organization and your portal branding administrator.
2. Create new objects (images, style sheets, HTML objects, and JavaScript) as required to fulfill the branding requirements.

   See Understanding Branding Objects for more information.
3. (Optional) Create any new branding element type definitions to support your requirements.

   See for Creating a New Branding Element more information.
4. Create new header and footer definitions combining delivered and new branding elements. Typically, it is easier to clone an existing definition and modify it to suit your needs.

   See Defining Headers and Footers for more information.
5. Create a branding theme definition using your new header, footer, and theme style sheet definitions.

   See Assembling Branding Themes for more information.
6. (Optional) Generate mapped theme style sheets for fluid components.

   See Generating Theme Style Sheets for Fluid Components for more information.
7. (Optional) Create a navigation collection for custom homepage tabs.

   See Creating and Maintaining Navigation Collections for Custom Homepage Tabs for more information.
8. Specify branding system-wide settings.

   See Configuring Branding System Options for more information.
9. (Optional) Create any new user attribute type definitions to support your requirements.

   See Creating a New User Attribute Type for more information.
10. (Optional) Make any portal- or user attribute-specific theme and branding assignments.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Assigning Branding Themes for more information.

11. (Optional) Apply any component-specific branding to classic components.

    See Branding Classic Components for more information.

12. (Optional) Apply an pagelet-specific branding to pagelets.

    See Maintaining Pagelet Branding Attributes for more information.

Creating System Definitions and Objects for Branding

This topic provides an overview of branding objects and discusses how to:

- Create and maintain HTML objects.
- Create and maintain JavaScript objects.
- Create and maintain style sheet objects.
- Add and edit branding objects.
- Upload and maintain images.
- Add and edit images.
- Define element types.
- Define user attribute types.

Understanding Branding Objects

PeopleTools branding allows you to create and maintain branding objects such as HTML, JavaScript, free form style sheets, and images without requiring access to Application Designer. These objects can then be used in places such as branding system definitions, headers or footers, custom layout objects for pagelets and homepage tabs, and in other definitions.

While you can create and maintain HTML, JavaScript, and free form style sheet objects, and images in Application Designer, these Application Designer objects *are not* accessible to be viewed or maintained through portal branding. But, if you enter the same name as an existing Application Designer object and if you have edit permissions on the managed object, then you will be prompted as to whether you want to overwrite the existing object through portal branding. However, objects that you create through portal branding can be viewed and updated in Application Designer.

Note: You can use Application Designer to migrate these portal branding objects from a test environment to a production environment.

Related Links

"Understanding Projects" (PeopleTools 8.54: Application Designer Developer's Guide)

Creating and Maintaining HTML Objects

Use the HTML Objects page to upload and maintain HTML objects used for branding.

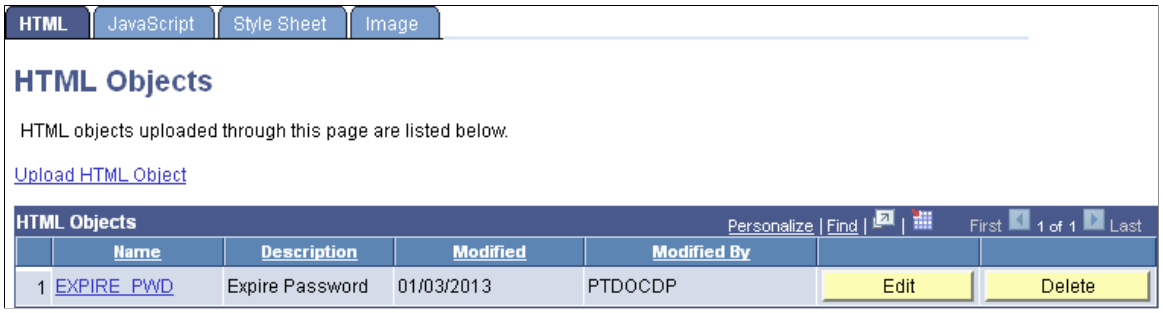
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

PeopleTools, Portal, Branding, Branding Objects

Image: HTML Objects page

This example illustrates the fields and controls on the HTML Objects page. You can find definitions for the fields and controls later on this page.



Upload HTML Object Click the link to create a new HTML object.

See Adding and Editing Branding Objects.

Name Click the object name link for an existing HTML object to view the object details including the HTML code.

Description Displays the object's description.

Edit Click the Edit button to modify the description or stored HTML code.

See Adding and Editing Branding Objects.

Delete Click the Delete button to delete the object from the database.

Note: The link to view the HTML object is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

HTML objects created in Application Designer cannot be viewed or edited in portal branding. However, HTML objects created in portal branding can be viewed and modified in Application Designer.

Related Links

"Understanding HTML Definitions" (PeopleTools 8.54: Application Designer Developer's Guide)

Creating and Maintaining JavaScript Objects

Use the JavaScript objects page to to upload and maintain JavaScript objects used for branding.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

1. PeopleTools, Portal, Branding, Branding Objects.
2. Select the JavaScript tab.

Image: JavaScript objects page

This example illustrates the fields and controls on the JavaScript objects page. You can find definitions for the fields and controls later on this page.

Upload JavaScript Object Click the link to create a new JavaScript object.

Note: JavaScript is stored as an HTML object in the database.

See Adding and Editing Branding Objects.

Name Click the object name link for an existing JavaScript object to view the object details including the JavaScript code.

Description Displays the object's description.

Edit Click the Edit button to modify the description or stored JavaScript code.

See Adding and Editing Branding Objects.

Delete Click the Delete button to delete the object from the database.

Note: The link to view the JavaScript object is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

HTML objects created in Application Designer cannot be viewed or edited in portal branding. However, JavaScript objects created in portal branding (and stored as HTML objects) can be viewed and modified in Application Designer.

Related Links

"Understanding HTML Definitions" (PeopleTools 8.54: Application Designer Developer's Guide)

"Working with JavaScript in HTML Definitions" (PeopleTools 8.54: Application Designer Developer's Guide)

Creating and Maintaining Style Sheet Objects

Use the Style Sheet objects page to to upload and maintain free form style sheet objects used for branding.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

1. PeopleTools, Portal, Branding, Branding Objects.
2. Select the Style Sheet tab.

Image: Style Sheet objects page

This example illustrates the fields and controls on the Style Sheet objects page. You can find definitions for the fields and controls later on this page.

Upload Style Sheet Object Click the link to create a new free form style sheet object.

See Adding and Editing Branding Objects.

Name Click the object name link for an existing free form style sheet object to view the object details including style definitions.

Description Displays the object's description.

Edit Click the Edit button to modify the description or stored style definitions.

See Adding and Editing Branding Objects.

Delete Click the Delete button to delete the object from the database.

Note: The link to view the style sheet is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

Style sheet objects created in Application Designer cannot be viewed or edited in portal branding. However, free form style sheet objects created in portal branding can be viewed and modified in Application Designer.

Related Links

"Style Sheet Types" (PeopleTools 8.54: Application Designer Developer's Guide)

"Working With Free Form Sub Style Sheets" (PeopleTools 8.54: Application Designer Developer's Guide)

Adding and Editing Branding Objects

Use the Add/Edit Branding Object page to add or edit a branding object.

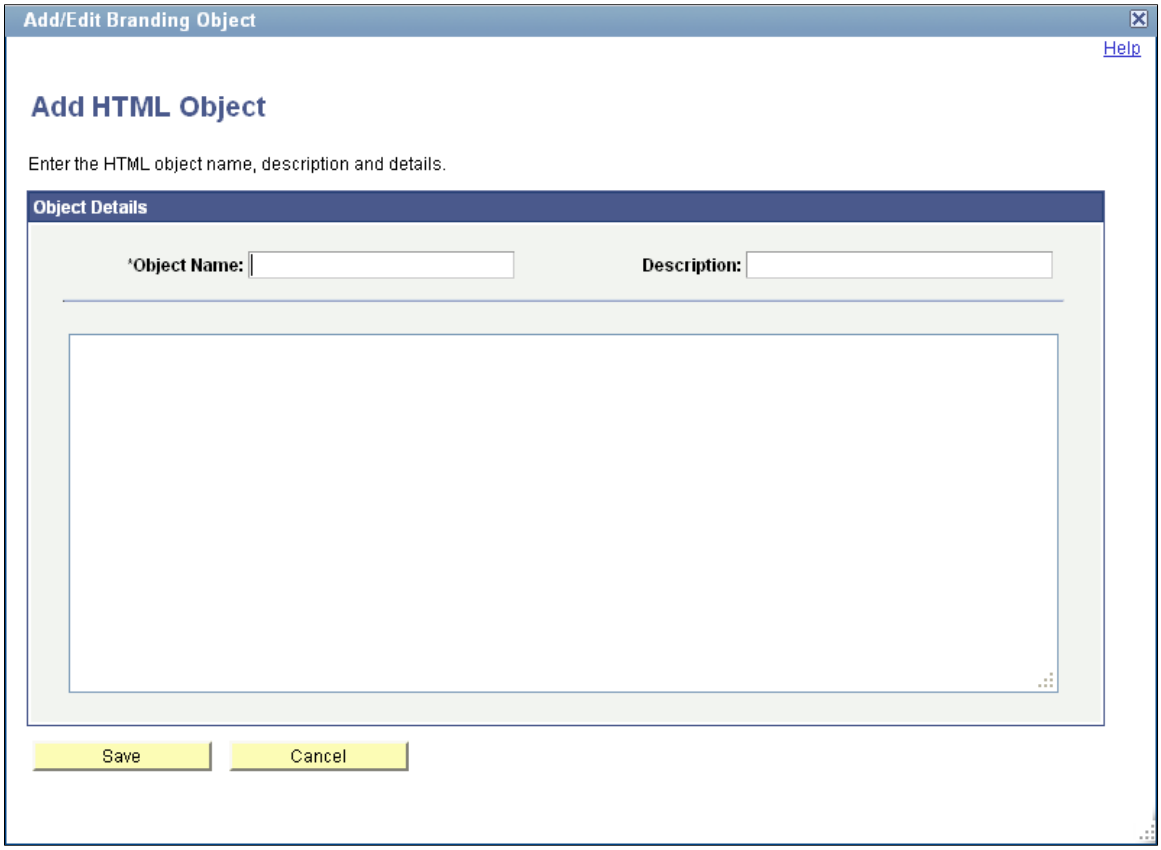
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

On the HTML Objects page, JavaScript objects page, or Style Sheet objects page, either click the Upload Object link or click the Edit button for an existing object.

Image: Add/Edit Branding Object page

This example illustrates the fields and controls on the Add/Edit Branding Object page (for an HTML object). You can find definitions for the fields and controls later on this page.



Object Name

Enter an ID for the new object.

Important! If you specify the name of an existing object of this type whether it was created in portal branding or through Application Designer, you will be prompted to overwrite the existing object after you click the Save button.
However, if you have no permissions or read-only permissions on a Application Designer object, a duplicate ID error message will be displayed, and you will need to enter a different object ID.

Warning! Just as you must not update PeopleTools-delivered objects in Application Designer, do not use portal branding to update these PeopleTools-delivered objects.

Description

Enter a descriptive name for the object.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

(unlabeled) Enter the HTML, JavaScript, or free form style sheet code.

Uploading and Maintaining Images

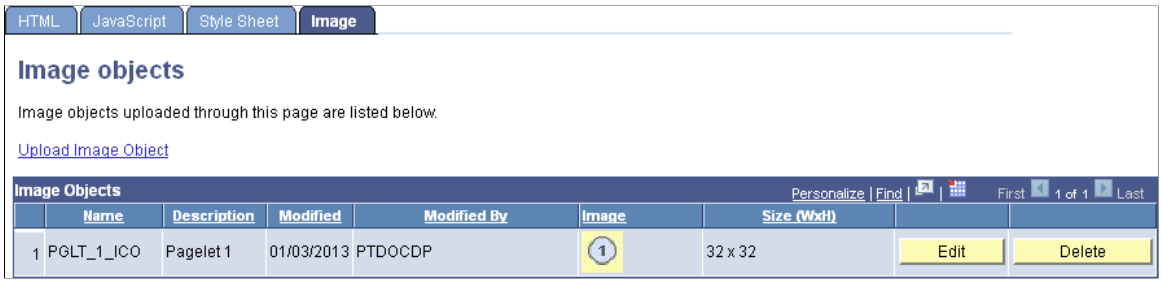
Use the Image objects page to upload and maintain image objects used for branding.

Navigation

1. PeopleTools, Portal, Branding, Branding Objects.
2. Select the Image tab.

Image: Image objects page

This example illustrates the fields and controls on the Image objects page. You can find definitions for the fields and controls later on this page.



Upload Image Object Click the link to upload a new image file.

See Adding and Editing Images.

Name Displays the image object's name.

Description Displays the object's description.

Edit Click the Edit button to modify the image including uploading a new image file.

See Adding and Editing Images.

Delete Click the Delete button to delete the object from the database.

Note: The link to view the image is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

Image objects created in Application Designer cannot be viewed or edited in portal branding. However, image objects created in portal branding can be viewed and modified in Application Designer.

Related Links

"Understanding Image Definitions" (PeopleTools 8.54: Application Designer Developer's Guide)

Adding and Editing Images

Use the Add/Edit Image page to add or edit an image.

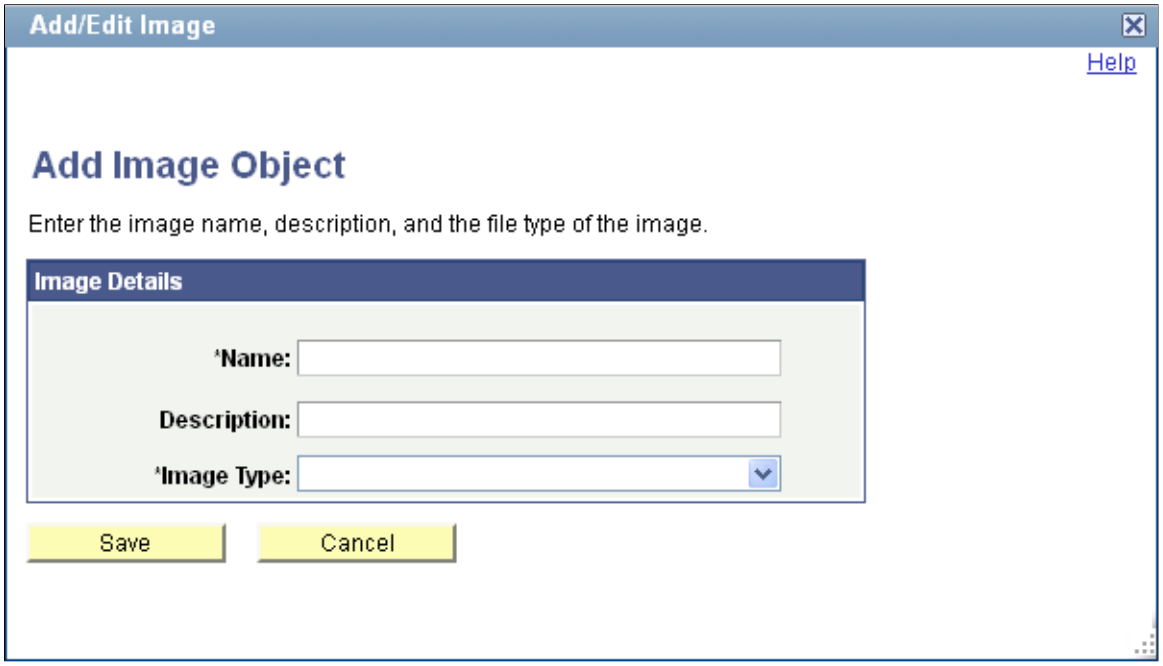
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

On the Image objects page, either click the Upload Image Object link or click the Edit button for an existing image. Alternatively, on the Pagelet Branding page, click the Upload new image link.

Image: Add/Edit Image page

This example illustrates the fields and controls on the Add/Edit Image page. You can find definitions for the fields and controls later on this page.



Name

Enter an ID for the new image object.

Important! If you specify the name of an existing image object whether it was created in portal branding or through Application Designer, you will be prompted to overwrite the existing object after you click the Save button.

Description

Enter a descriptive name for the object.

Important! Some database platforms, such as DB2 UDB for OS/390 and z/OS, limit the file size to 32 KB.

Image Type

Select one of the following image formats to access the File Attachment dialog box to browse and select the file to upload:

- BMP Files [*.bmp]
- DIB Files [*.dib]
- GIF Files [*.gif]
- JPG Files [*.jpg]
- PNG Files [*.png]

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- SVG [*.svg]

Note: The File Attachment dialog box does not filter your browsing by the selected file type. However, if you attempt to upload a file that does not match the specified format, then an error message will be generated and the upload will not occur.

Defining Element Types

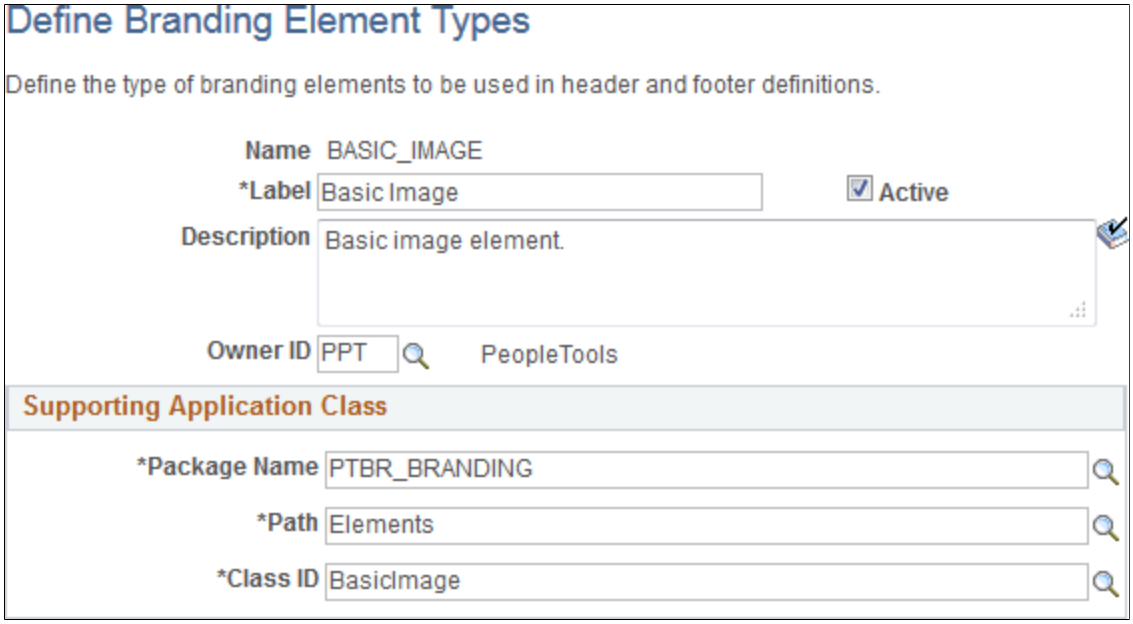
Use the Define Branding Element Types page to create and maintain the types of branding elements to be used in header and footer definitions. Each branding element type is mapped to a supporting PeopleCode application class definition.

Navigation

PeopleTools, Portal, Branding, System Data, Define Element Types

Image: Define Branding Element Types Page

This example illustrates the fields and controls on the Define Branding Element Types page.



Name Displays the unique identifier for this element type.

Label Enter a short name for the element type, which will be displayed when an element of this type is selected to be used in a header or footer definition.

Active This check box is selected by default.

Deselect this check box to make the element type inactive. No new elements of inactive types can be added to header or footer definitions. Moreover, existing elements of inactive element types in existing header or footer definitions are ignored.

Description Enter a description for the element type.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Owner ID Select which product, feature, or application owns this element type definition.

Package Name Select the name of the application package that contains the application class that defines this element type.

Path Select the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Class ID Select the name of the application class that defines this element type.

Related Links

Creating a New Branding Element

Defining Headers and Footers

Defining User Attribute Types

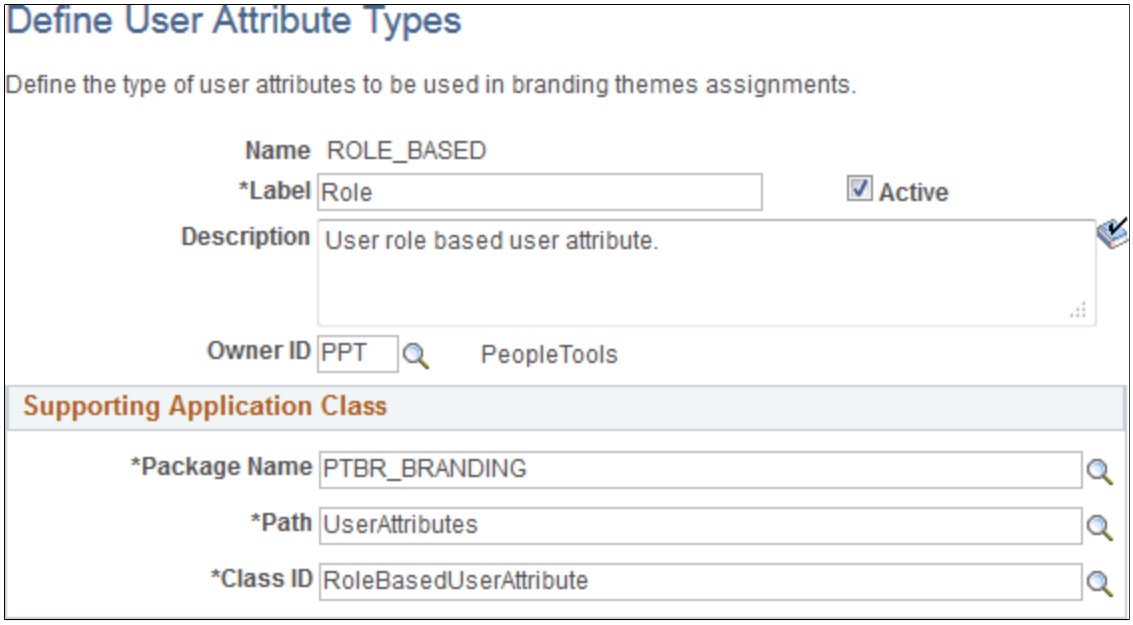
Use the Define User Attribute Types page to create and maintain the types of user attributes to be used in branding theme assignments. Each user attribute type is mapped to a supporting PeopleCode application class definition.

Navigation

PeopleTools, Portal, Branding, System Data, Define User Attribute Types

Image: Define User Attribute Types Page

This example illustrates the fields and controls on the Define User Attribute Types page.



Name Displays the unique identifier for this user attribute type.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Label	Enter a short name for the user attribute type, which will be displayed when this type is selected to be used in a theme assignment.
Active	This check box is selected by default. Deselect this check box to make the user attribute type inactive. An inactive user attribute type cannot be selected for use in a branding theme assignment. Moreover, existing theme assignments with inactive user attribute types are ignored.
Description	Enter a description for the user attribute type.
Owner ID	Select which product, feature, or application owns this user attribute type definition.
Package Name	Select the name of the application package that contains the application class that defines this user attribute type.
Path	Select the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class ID	Select the name of the application class that defines this user attribute type.

Related Links

Creating a New User Attribute Type
Assigning Branding Themes

Administering System Branding

This topic discusses how to:

- Configure branding system options.
- Define headers and footers.
- Assemble branding themes.
- Assign branding themes.

Configuring Branding System Options

Use the Branding System Options page to specify system-wide branding settings including the branding theme type (family), the branding theme, the default style sheet for classic components, and the default style sheet for fluid components.

Important! Set the Theme Style Type field first. When the value of this field is changed, compatible values are automatically specified for both the Default Branding Theme field and the Default Style Sheet Name field.

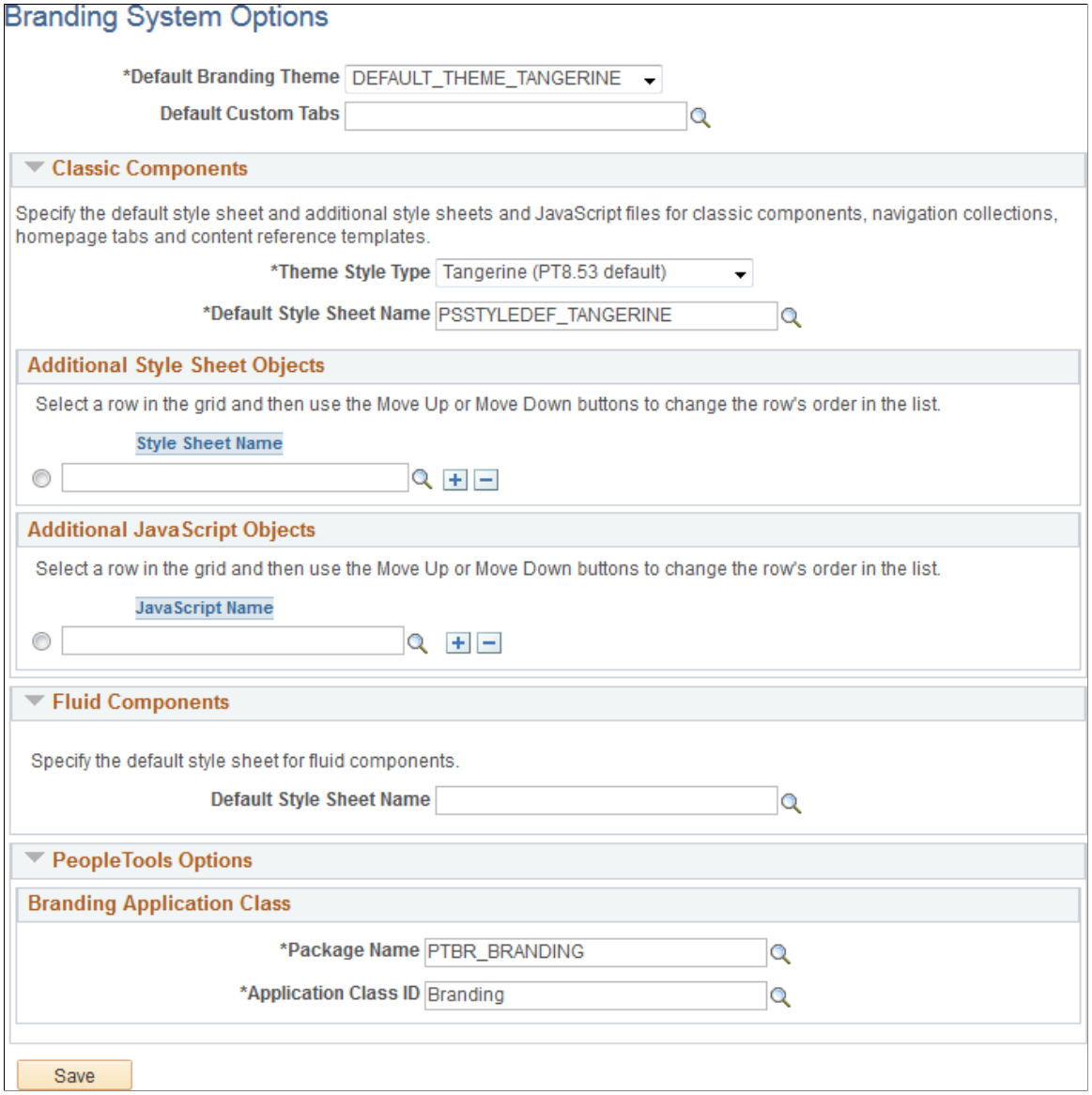
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

PeopleTools, Portal, Branding, Branding System Options

Image: Branding System Options Page

This example illustrates the fields and controls on the Branding System Options page.



Default Branding Theme Select a default, system-wide branding theme.

Warning! The branding theme selected must be compatible with the theme family selected in the Theme Style Type field. Otherwise, unexpected and undesirable results can occur. Therefore, specify the Theme Style Type field first; then retain the automatically selected value of this field or select another theme that is compatible within the selected theme family. See the following section, “Compatible Branding Themes,” for more information on compatible branding themes.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Default Custom Tabs Select a default, system-wide navigation collection that will be used to render custom homepage tabs.

Compatible Branding Themes

Use the following table to determine which branding theme is compatible with a specified theme style family (type).

Theme Style Family	*Compatible Branding Themes*	*Description*
Tangerine (PT8.53 default)	DEFAULT_THEME_ TANGERINE	The default theme for PeopleSoft 9.2 applications (and above) on PeopleTools 8.53 (and above). **Note:** PeopleSoft Interaction Hub 9.1 Rev 2 is the equivalent of a PeopleSoft 9.2 application in regard to default themes and style sheets.
Tangerine (PT8.53 default)	DEFAULT_THEME_ TANGERINE_ALT	An alternate Tangerine theme that includes a portal header that is compatible with the look of PeopleSoft fluid applications.
Swan (PT8.52 default)	DEFAULT_THEME_SWAN	The default theme for PeopleSoft 9.1 applications on PeopleTools 8.50 to 8.52.
Classic (DarkBlue)	DEFAULT_THEME_CLASSIC	The default theme for PeopleSoft 9.0 applications (and below) on PeopleTools releases prior to 8.50.

Classic Components

Theme Style Type Use this field to specify the theme style family (type).

Important! Select this field first. When the value of this field is changed, compatible values are automatically specified for both the Default Branding Theme field and the Default Style Sheet Name field.

Default Style Sheet Name Select a default, system-wide style sheet for classic components.

The value selected here will also be visible on the PeopleTools Options page. Moreover, if the previous value of the default style sheet was also used on the System Options page or the Registry Options page for one or more portal registries, the value selected here is updated to each of those pages.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Warning! The style sheet selected should be compatible with the theme family selected in the Theme Style Type field. Otherwise, unexpected and undesirable results can occur. Therefore, specify the Theme Style Type field first; then retain the automatically selected value of this field or select another style sheet that is compatible within the selected theme family.

Style Sheet Name (Additional Style Sheet Objects)

Specify zero, one, or more than one style sheet to be loaded with all classic components.

JavaScript Name (Additional JavaScript Objects)

Specify zero, one, or more than one JavaScript object to be loaded with all classic components.

Important! JavaScript objects are stored as HTML objects in the database. The lookup does not differentiate between standard HTML objects and JavaScript objects. Therefore, it is your responsibility to ensure that you have selected valid JavaScript to be loaded with a component.

Move Up
Move Down

Select the row designating a style sheet or a JavaScript object, and then use the Move Up and Move Down buttons to change the order in which the objects are loaded with all classic components.

Fluid Components

Default Style Sheet Name

Select a default, system-wide style sheet for fluid components.

Note: If no style sheet is specified, then PSSTYLEDEF _FMODE is used as the default style sheet for all fluid components.

The value selected here will also be visible on the PeopleTools Options page.

PeopleTools Options

The branding application class generates the portal headers, footers, and other aspects of the branding theme. The default branding application class is PTBR_BRANDING:Branding. If this delivered application class does not fully support your requirements, you can create your own by extending the delivered application class.

Package Name

Select the name of the application package that contains the application class that defines system-wide branding functions. The default value is: PTBR_BRANDING.

The value selected here will also be visible on the PeopleTools Options page.

Application Class ID

Select the name of the application class that defines system-wide branding functions. The default value is: Branding.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The value selected here will also be visible on the PeopleTools Options page.

Related Links

Assembling Branding Themes

Creating and Maintaining Navigation Collections for Custom Homepage Tabs

"PeopleTools Options" (PeopleTools 8.54: System and Server Administration)

Defining Portal Utilities System Options

Defining Portal Utilities Registry Options

Defining Headers and Footers

Use the Define Headers and Footers page to create and maintain header and footer definitions by adding elements to a tree structure. The tree structure is represented as:

- Folders – These are container elements such as the root element (created automatically with the Root Element type) and other elements added as the Basic Container element type. Folders can contain children, which are leaves and other folders.
- Leaves – These are other elements added with other element types including basic types (Basic Link, Basic Text, Basic Image, and others) as well as specialized element types (Home Link, Sign Out Link, Dashboard Label, and others).

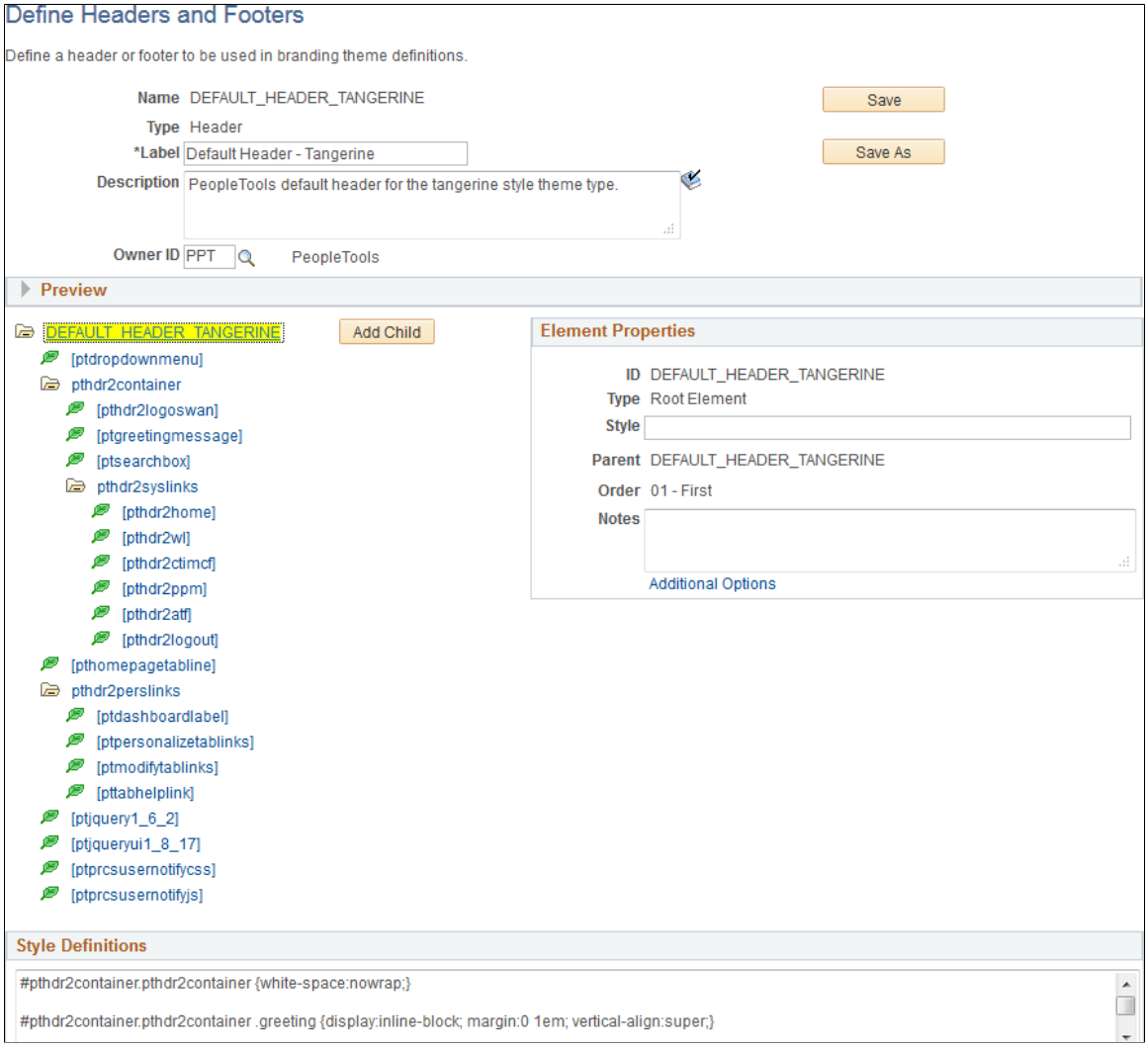
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

PeopleTools, Portal, Branding, Define Headers and Footers

Image: Define Headers and Footers Page

This example illustrates the fields and controls on the Define Headers and Footers page.



Name Displays the ID for the header or footer that was specified when the definition was created.

Type Displays the type of the definition:

- Footer
- Header

Label Enter a label for the header or footer definition.

Description Enter a description for the header or footer definition.

Owner ID Select which product, feature, or application owns this header or footer definition.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

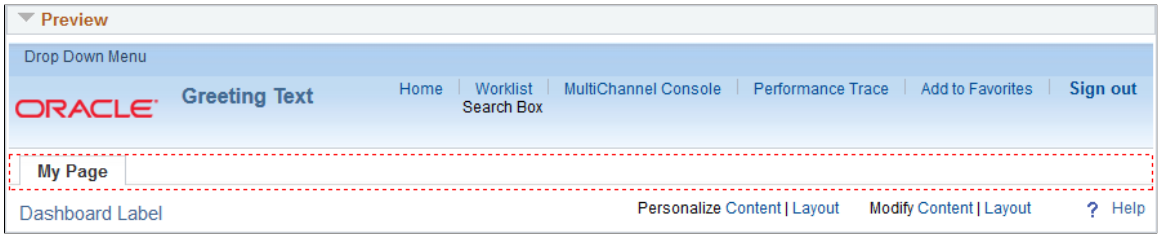
Save	Click the Save button to save changes to the current definition.
Save As	Click the Save As button to save the current definition with a new ID.
Delete	At the definition level, a Delete button will be displayed only for those definitions that are not included in a theme definition. Use this Delete button to delete the header or footer definition.
Add Child (in tree structure)	Select the root element or a container element and then click the Add Child button to add a child element.
Delete (in tree structure)	Select any element except the root element and then click Delete to delete the element (and any children).
Style Definitions	Enter any additional free form style class definitions in this long edit box.

Preview Section

Expand the Preview section to review the look of the current header or footer definition. The preview is updated after changes are made to any of the elements or style classes used in the definition. Tab out of any changed field or element to see the updated preview.

Image: Preview Section

This example illustrates the preview of a header definition.



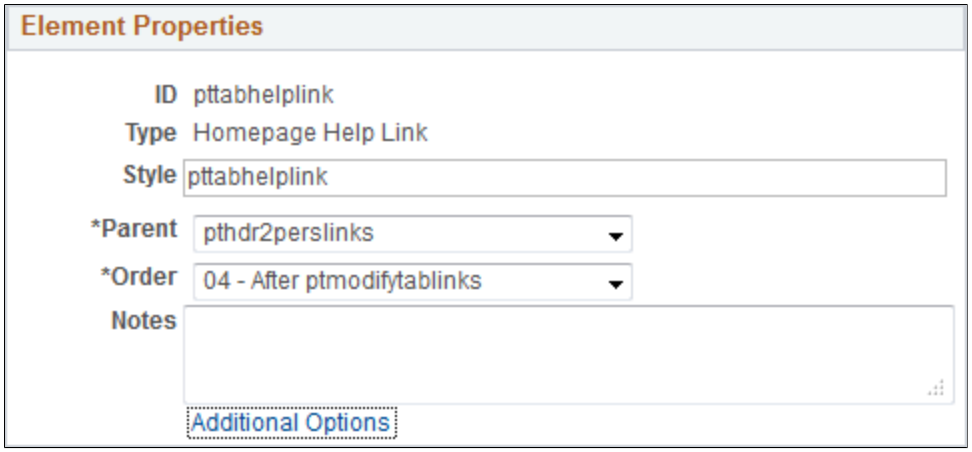
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Element Properties

Use the Element Properties group box to review and set properties of the element selected in the tree.

Image: Element Properties Group Box

This example illustrates the fields and controls on the Element Properties group box.



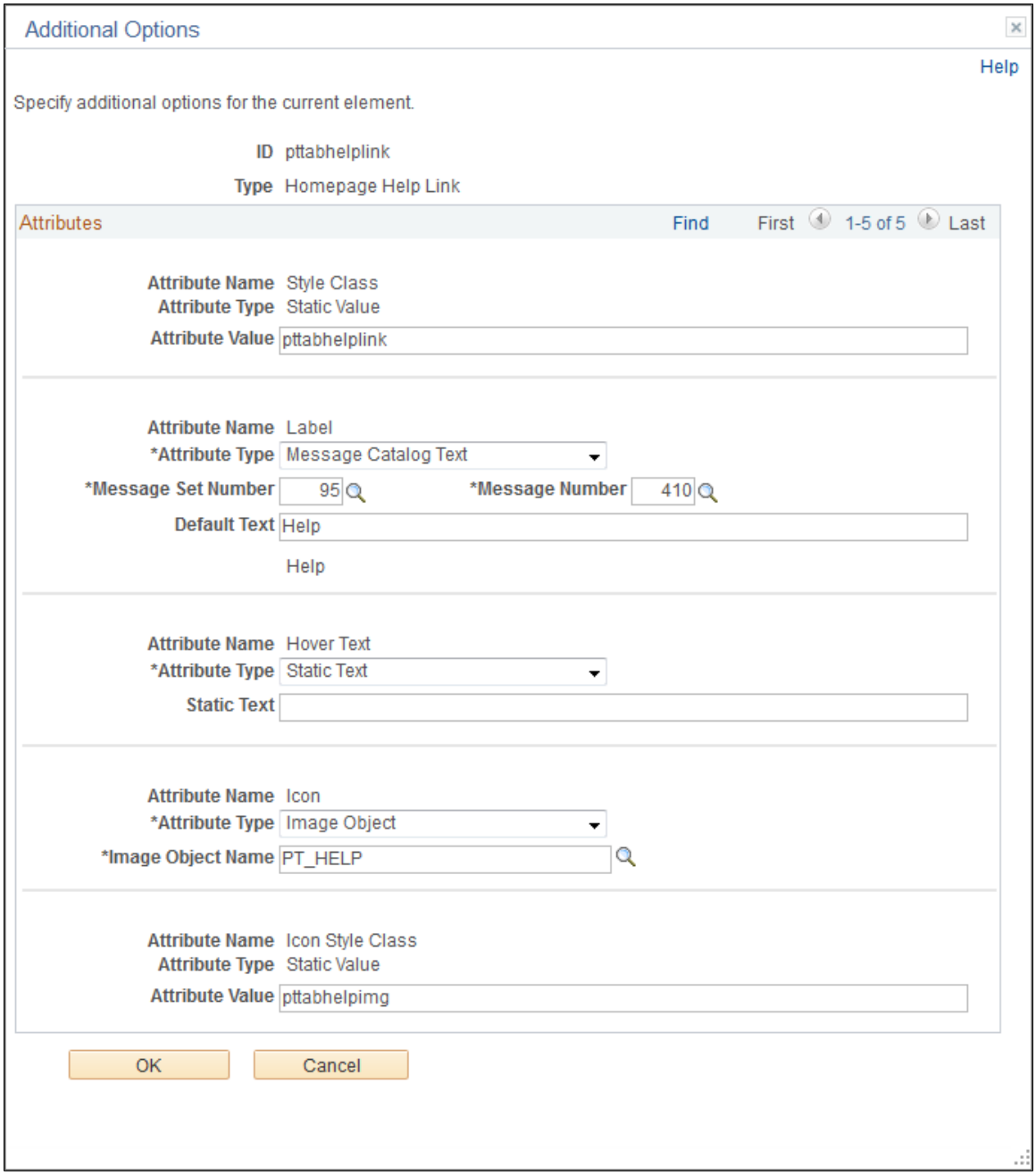
ID	Displays the ID of the element that was specified when the element was created.
Type	Displays the element type that was selected when the element was created. Valid element types are defined on the Define Element Types page.
Style	Enter one or more free form style class names separated by spaces to specify additional formatting for this element.
Parent	Select a new parent container for this element by container ID. **Note:** All container elements are listed except children of the current element.
Order	Select a new placement for this element among its sibling elements within the parent container.
Notes	Enter any optional notes about this element.
Additional Options	Click the Additional Options link to view the Additional Options page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Additional Options Page

Image: Additional Options Page

This example illustrates some of the fields and controls that can be found on the Additional Options page.



The fields displayed on the Additional Options page are controlled by the application class definition that supports this element type and will vary depending on the element type.

ID	Displays the ID of the element.
Type	Displays the element type.
Attribute Name	Displays the name of the attribute.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Attribute Type

Displays the attribute type. For some attributes, this value is not read-only and can be changed.

Depending on the attribute type, additional fields are displayed showing the attribute value, static text, message set and number, default text, an object ID, and others.

Related Links

Defining Element Types

Creating a New Branding Element

Assembling Branding Themes

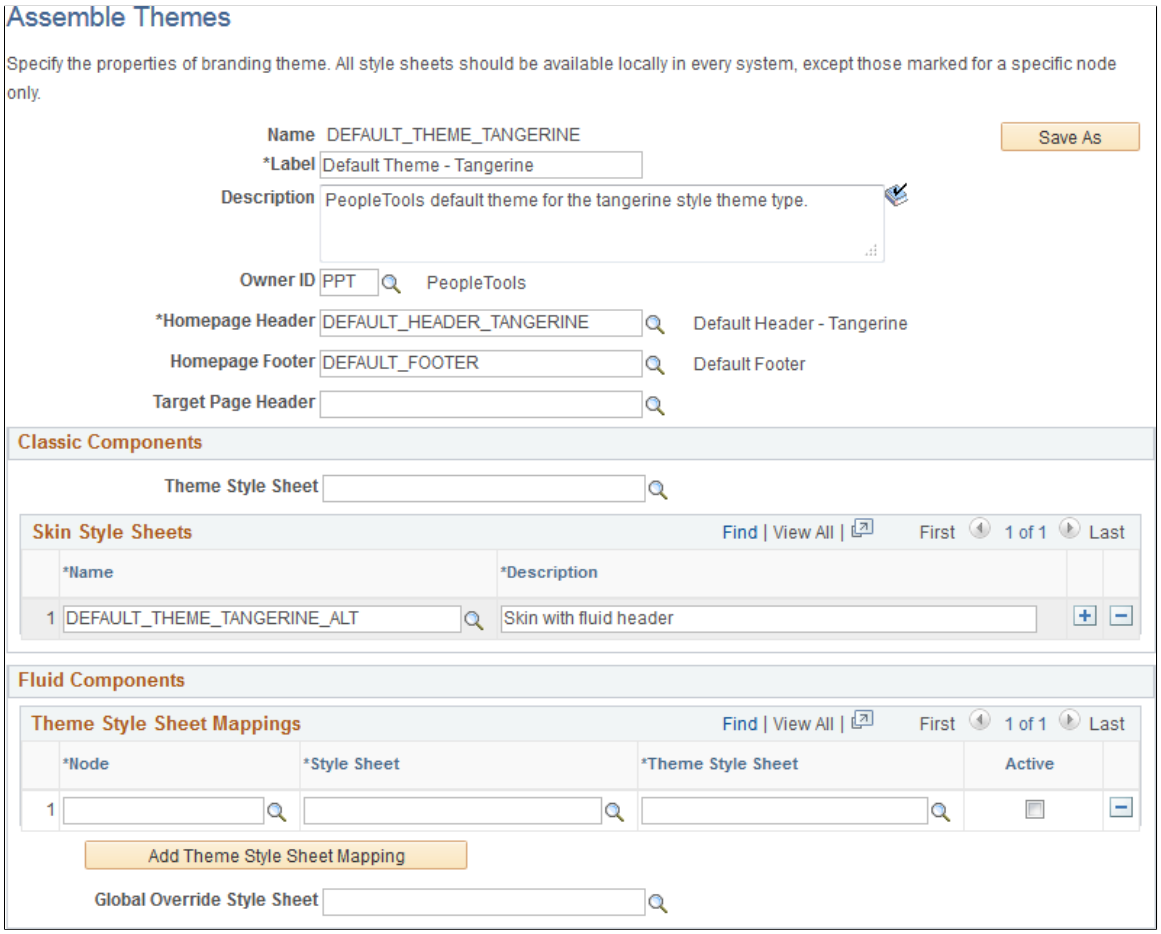
Use the Assemble Themes page to create and maintain the definitions of branding themes.

Navigation

PeopleTools, Portal, Branding, Assemble Themes

Image: Assemble Themes Page

This example illustrates the fields and controls on the Assemble Themes page.



Name

Displays the ID of the theme that was specified when the theme was created.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Save As Click the Save As button to save the current theme definition with a new ID.

Delete Click the Delete button to delete the this theme definition.

Note: The Delete button is displayed only for those theme definitions that are not assigned at the system, portal, or user attribute level.

Label Enter a label for the theme definition.

Description Enter a description for the theme definition.

Owner ID Select which product, feature, or application owns this theme definition.

Homepage Header Select a header definition to be used for homepages.

Homepage Footer Select a footer definition to be used for homepages.

Target Page Header Select a header definition to be used for all target pages.

Note: If no header is specified, then the header defined for homepages will also be used on target pages.

Classic Components

Note: For a cluster of applications managed by PeopleSoft Interaction Hub, the theme style sheet and any skin style sheets must exist in all systems.

Theme Style Sheet Select an override style sheet for the theme for classic components.

Note: Theme style sheets can only be free form style sheets.

Name (Skin Style Sheets) Identify one or more free form style sheets that can be made available to use with this theme as skin style sheets. The skin style sheets identified here are made available for use on the Assign Branding Themes page.

Description Enter a description of the skin.

Fluid Components

Use the Fluid Components group box to register a base style sheet and theme style sheet pair for a specific node. Alternatively, you can use the Register Fluid Theme Style Sheet Mappings page to perform this registration, which will ensure that the pair is mapped correctly. See Generating Theme Style Sheets for Fluid Components. The rows in the Theme Style Sheet Mappings grid specify a mapping of a theme style sheet to a base style sheet only; the grid does not specify that this base style sheet is in use. The base style sheet is assigned at the system level on the Branding System Options page, or at the component or page

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

level through PeopleCode. Then, when the base style sheet is loaded by the system into the generated HTML, any mapped theme style sheet is loaded immediately after it.

Note: For a cluster of applications managed by PeopleSoft Interaction Hub, the global override style sheet and any mapped them style sheets for a node specification of *ANY* must exist in all systems.

Node	Select the node to which the style sheet and theme style sheet is to be applied. You can select a specific node name, or one of the following values: • *ANY* – Applies to all nodes in all systems. (This setting is applicable to a cluster of applications managed by PeopleSoft Interaction Hub.) • *LOCAL_NODE* – Applies to all nodes of the current system only. **Note:** The list of specific nodes includes non-local PIA nodes that have a defined content URI.
Style Sheet	Select the base style sheet for fluid components for the selected node. **Note:** The lookup lists style sheets on the current system. For remote nodes, you must ensure that the style sheet you select also exists in the remote system.
Theme Style Sheet	Select a mapped style sheet that has already been generated from the selected base style sheet. **Note:** The lookup lists style sheets on the current system. For remote nodes, you must ensure that the style sheet you select also exists in the remote system.
Active	Select this option to make the mapped pair active.
Add Theme Style Sheet Mapping	Click this button to add another row to the grid.
Global Override Style Sheet	Select an override style sheet for the theme for all fluid components on all nodes.

Related Links

Assigning Branding Themes

Generating Theme Style Sheets for Fluid Components

Assigning Branding Themes

Use the Assign Branding Themes page to specify portal-specific branding options and defaults to override the system default settings. In addition, use this page to assign different themes, skins, custom tabs, and fluid homepages by user attribute.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

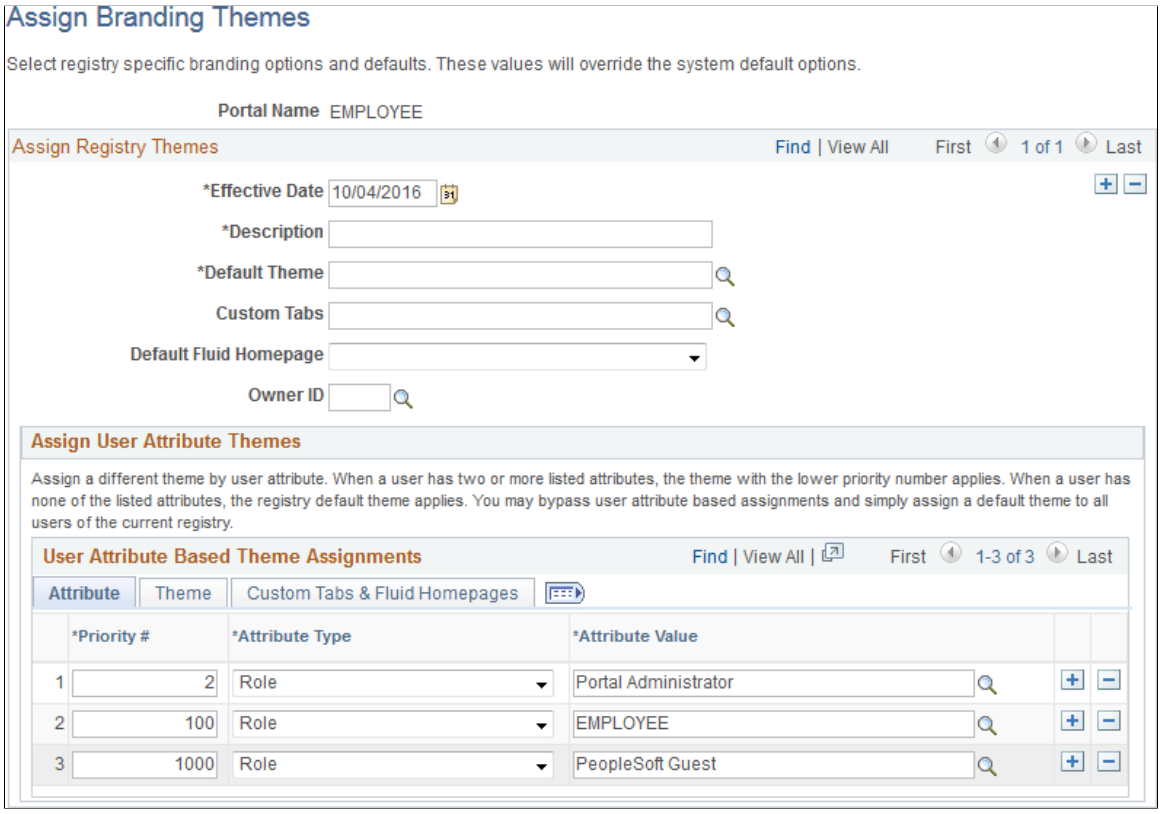
Note: Once a portal-specific definition has been created, it cannot be entirely deleted. However, you can use effective dates to control whether a definition is currently in effect.

Navigation

PeopleTools, Portal, Branding, Assign Themes

Image: Assign Branding Themes Page

This example illustrates the fields and controls on the Assign Branding Themes page.



Portal Name Displays the name of the current portal registry.

Note: You can edit theme assignments for the current portal only. Therefore, sign into a different portal to modify its theme assignments.

Effective Date Specify the effective date for this definition.

Label Enter a label for the theme definition.

Default Theme Select the default theme for this portal.

Default Skin Select a default skin for all users who sign into this portal.

Note: The list of available skins is specified on the Assemble Theme Templates page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Custom Tabs Select the default custom tabs navigation collection for this portal.

Default Custom Tab Select a default classic homepage tab from the custom tabs navigation collection for all users who sign into this portal.

Default Fluid Homepage Select a default fluid homepage for all users who sign into this portal.

Note: If the Fluid Homepage field on the Web Profile Configuration - Look and Feel page specifies both a component for rendering fluid homepage tabs and a specific homepage, then any value set on this page will be overridden. See "Web Profile Configuration - Look and Feel Page" (PeopleTools 8.54: Fluid User Interface Developer's Guide) for more information.

Owner ID Select which product, feature, or application owns this definition.

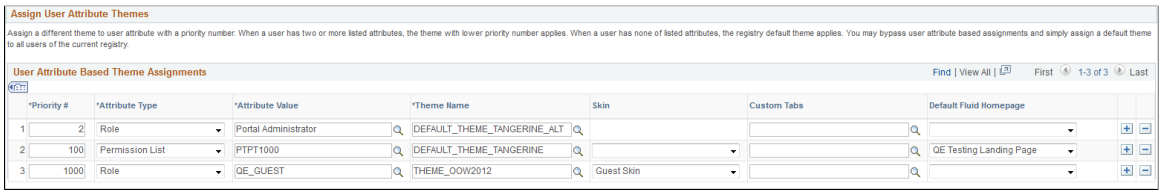
User Attribute Based Theme Assignments

Use the User Attribute Based Theme Assignments grid to make theme assignments by user attribute. In PeopleTools, two user attribute types are delivered: by role or by permission list. You can use this grid to override the portal settings for the theme, skin, custom tabs collection, or default fluid homepage.

When you assign branding themes by user attribute, you also assign a priority sequence number to each attribute. The priority number dictates each attribute's precedence in determining which theme will be displayed for each user. If a user has two or more attributes with a branding theme assignment, the attribute with the higher priority applies with *1* being the highest priority, *2* being the next highest priority, and so on. If a user has none of the attributes used in theme assignments, the portal-wide settings apply.

Image: User Attribute Based Theme Assignments Grid

This example illustrates the fields and controls on the User Attribute Based Theme Assignments grid. In this example, the three tabs of this grid have been expanded and the tabs themselves are not visible.



Priority # Specify the priority for this attribute with *1* being the highest priority, *2* being the next highest priority, and so on.

Attribute Type Specify the attribute type. The PeopleTools-delivered values are:

- *Role*
- *Permission List*

Attribute Value Select a value for the selected attribute type.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Theme Name	Select the theme for this attribute.
Custom Tabs	Select a custom tabs navigation collection for this attribute.
Default Custom Tab	Select a default classic homepage tab from the custom tabs navigation collection for this attribute.
Default Fluid Homepage	Select a default fluid homepage for this attribute.

Note: If the Fluid Homepage field on the Web Profile Configuration - Look and Feel page specifies both a component for rendering fluid homepage tabs and a specific homepage, then any value set on this page will be overridden. See "Web Profile Configuration - Look and Feel Page" (PeopleTools 8.54: Fluid User Interface Developer's Guide) for more information.

Related Links

"Managing Fluid Homepages" (PeopleTools 8.54: Fluid User Interface Developer's Guide)
Assembling Branding Themes
Creating and Maintaining Navigation Collections for Custom Homepage Tabs

Applying Branding to Other Objects

This topic discusses how to:

- Brand classic components.
- Maintain pagelet branding attributes.

Branding Classic Components

Important! The Component Branding page can be used with classic components only. If you attempt to add a value for a fluid component, an error message is displayed.

Use the Component Branding page to specify custom style sheet objects and JavaScript objects in addition to those specified by the component's developer in Application Designer.

Important! If the current component has an associated component interface definition, then modifying the component definition by adding, deleting, or reordering style sheets or JavaScript objects will invalidate any associated component interface definition. A warning message will be issued when you save such a component on the Component Branding page. Therefore, you must also open and update the associated component interface definitions in Application Designer.

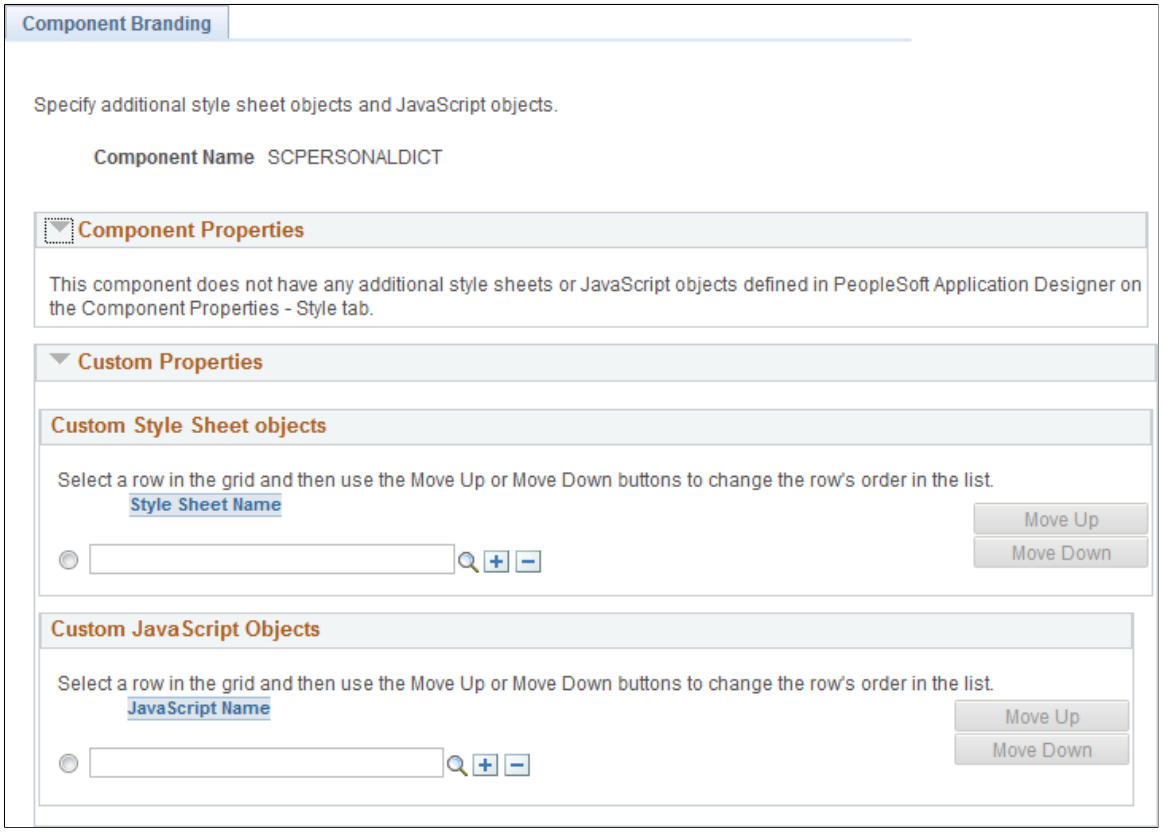
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

PeopleTools, Portal, Branding, Component Branding

Image: Component Branding Page

This example illustrates the fields and controls on the Component Branding page.



Component Name Displays the name of the component.

Component Properties The Component Properties group box displays the additional style sheet objects and JavaScript objects specified by the component's developer in Application Designer. These values are display-only, and cannot be modified here.

Custom Properties The Custom Properties group box displays the additional, custom style sheet objects and JavaScript objects specified by the portal administrator through the use of this Component Branding page.

Style Sheet Name Specify zero, one, or more custom style sheets to be loaded for this component.

JavaScript Name Specify zero, one, or more custom JavaScript objects to be loaded for this component.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Important! JavaScript objects are stored as HTML objects in the database. The lookup does not differentiate between standard HTML objects and JavaScript objects. Therefore, it is your responsibility to ensure that you have selected valid JavaScript to be loaded with a component.

Move Up
Move Down

Select the row designating a style sheet or a JavaScript object, and then use the Move Up and Move Down buttons to change the order in which the objects are loaded with the component.

Maintaining Pagelet Branding Attributes

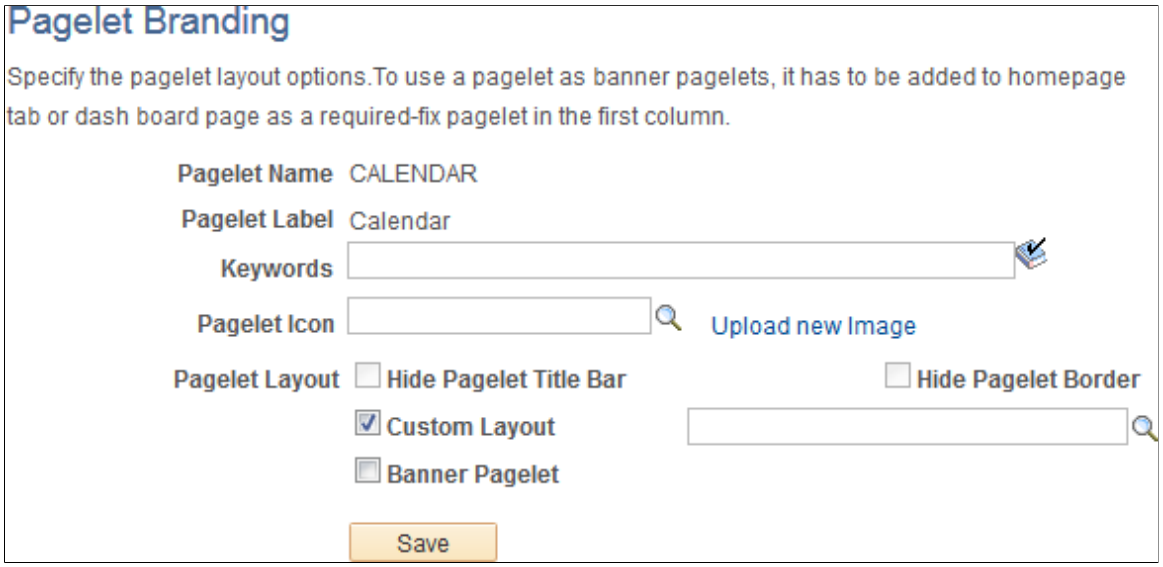
Use the Pagelet Branding page to specify pagelet title bar, border, and layout options.

Navigation

PeopleTools, Portal, Branding, Pagelet Branding

Image: Pagelet Branding Page

This example illustrates the fields and controls on the Pagelet Branding page.



Warning! The Pagelet Icon option is currently incompatible with either the Hide Pagelet Title Bar option or the Hide Pagelet Border option. Do not specify these options together for the same pagelet.

Note: If a pagelet is already in use on a homepage tab or dashboard page, clear the application server cache so that these branding changes become visible to all users. Alternatively, for testing purposes, you can remove the pagelet from the homepage or dashboard page and then add it back to view the changes on a per-user basis.

Pagelet Name Displays the ID of the selected pagelet.

Pagelet Label Displays the label of the selected pagelet.

Keywords Enter a keyword or keyword phrase to be stored as a content reference attribute.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Searching for and Importing Pagelets for more information on how keywords are used to search for pagelet definitions.

Note: Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language).

Pagelet Icon

Displays the ID of the pagelet icon if one has been selected for this pagelet. If an icon has been selected, then the image itself displays below this field.

Note: Oracle recommends image sizes from 16x16 pixels to 32x32 pixels for use as pagelet icons.

Click the Look Up Pagelet Icon button to select an image from the database. Clear the pagelet icon field to display a pagelet without an icon.

The PT_PORTAL_PGLT_HDR_IMG content reference attribute is created to store the image ID.

Upload new image

Click this link to access the Add Image Object page to upload a new image file.

See Adding and Editing Images.

Hide Pagelet Title Bar

Select this option to hide the pagelet title bar. Hiding the title bar hides the pagelet's label and all user interaction buttons to personalize, minimize, and remove the pagelet, or to get pagelet help.

Note: When this option is selected for a pagelet, users will not be able to use drag-and-drop to reposition the pagelet. The Personalize Layout page must be used instead.

The PORTAL_HP_COMPONENT content reference attribute is created to store this setting.

Hide Pagelet Border

Select this option to hide the pagelet's borders.

The PORTAL_HP_COMPONENT content reference attribute is created to store this setting.

Custom Layout

Select the Custom Layout check box to enable selection of a custom HTML layout object for this pagelet. Once this check box is selected, an additional unlabeled field and a lookup button are revealed allowing you to select an HTML object from the database.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: When you select the Custom Layout check box, both the Hide Pagelet Title Bar and Hide Pagelet Border check boxes are cleared (if selected) and disabled. If this check box is itself cleared, then these check boxes are re-enabled; however, their former settings are not restored.

Banner Pagelet

Select this option to identify this pagelet as one that can be used as a banner pagelet.

Note: To use a pagelet as a banner pagelet, it has to be added to a homepage tab or dashboard page as a required-fixed pagelet in the first column.

The PORTAL_BANNER_PGLT content reference attribute is created to store this setting.

Related Links

Defining Content References

Additional Branding Tasks

This topic provides an overview of navigation collection used to create custom homepage tabs and discusses how to:

- Create and maintain navigation collections for custom homepage tabs.
- Generate theme style sheets for fluid components.
- Create new theme style sheets and skins.
- Use additional style sheets to inject your custom branding.
- Create a new branding element.
- Create a new user attribute type.

Understanding Navigation Collections Used to Create Custom Homepage Tabs

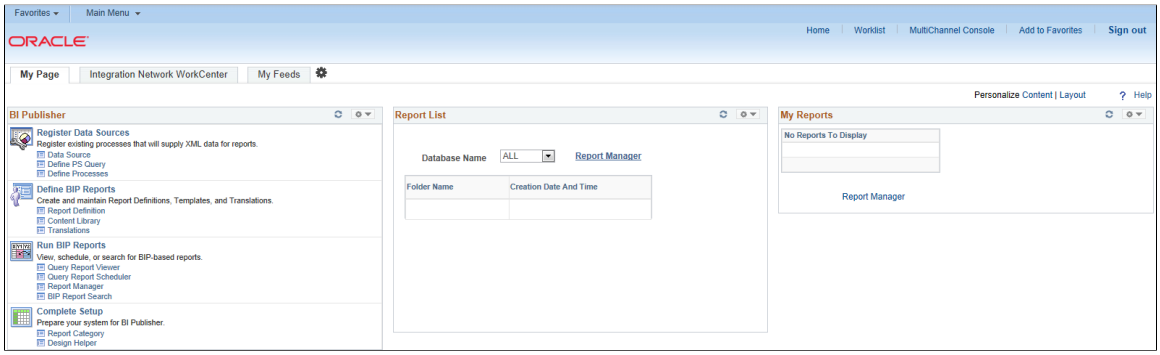
PeopleTools allows you to use specially designed navigation collections as definitions for custom homepage tabs. These navigation collections allow you to create links to local or remote content references that are then rendered as homepage tabs, and which appear after standard homepage tabs beneath the portal header. Because simple content references can be added to these collections, you can now designate any content reference or WorkCenter as a homepage tab. The requirement for creating such a “custom tabs” navigation collection is that the links must reside in the root folder of the navigation collection; links in subfolders will not be displayed as homepage tabs.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Once a navigation collection has been created, it can be designated for use as homepage tabs in the Custom Tabs field on the Branding System Options page or on the Assign Branding Themes page.

Image: Homepage with custom homepage tabs from a navigation collection

The following example displays a homepage that includes two custom homepage tabs. Integration Network WorkCenter and My Feeds have been defined as custom homepage tabs through the use of a navigation collection. These tabs are displayed after the default homepage tab: My Page.



Note: To avoid forcing users to scroll horizontally on the homepage, Oracle recommends that you configure and display no more than seven standard or custom homepage tabs.

Using a navigation collection to designate homepage tabs also provides you with the following benefits with respect to remote content:

- Security is checked in real time against definitions in the content provider system.
- You do not need to manually import content references or security from the content provider system
- If the remote system is down or has not been configured, the homepage tab is not displayed.

Note: Remote items require additional processing whenever a user signs in or the cache expires to check security on the remote content provider system. In addition, the Navigation Collections component only supports adding remote component-based content references. To add other remote items to a navigation collection, you must manually create the content reference through the Structure and Content component.

Note: PeopleSoft Interaction Hub 9.1 Rev 2 (and above) is delivered with navigation collection named Custom Tabs, which includes 12 pre-defined links as sample data. See the documentation for PeopleSoft Interaction Hub for more information on this Custom Tabs navigation collection.

Related Links

Administering System Branding

Assigning Branding Themes

Creating and Maintaining Navigation Collections for Custom Homepage Tabs

Use the Navigation Collections page to create and maintain links in any custom tabs navigation collection.

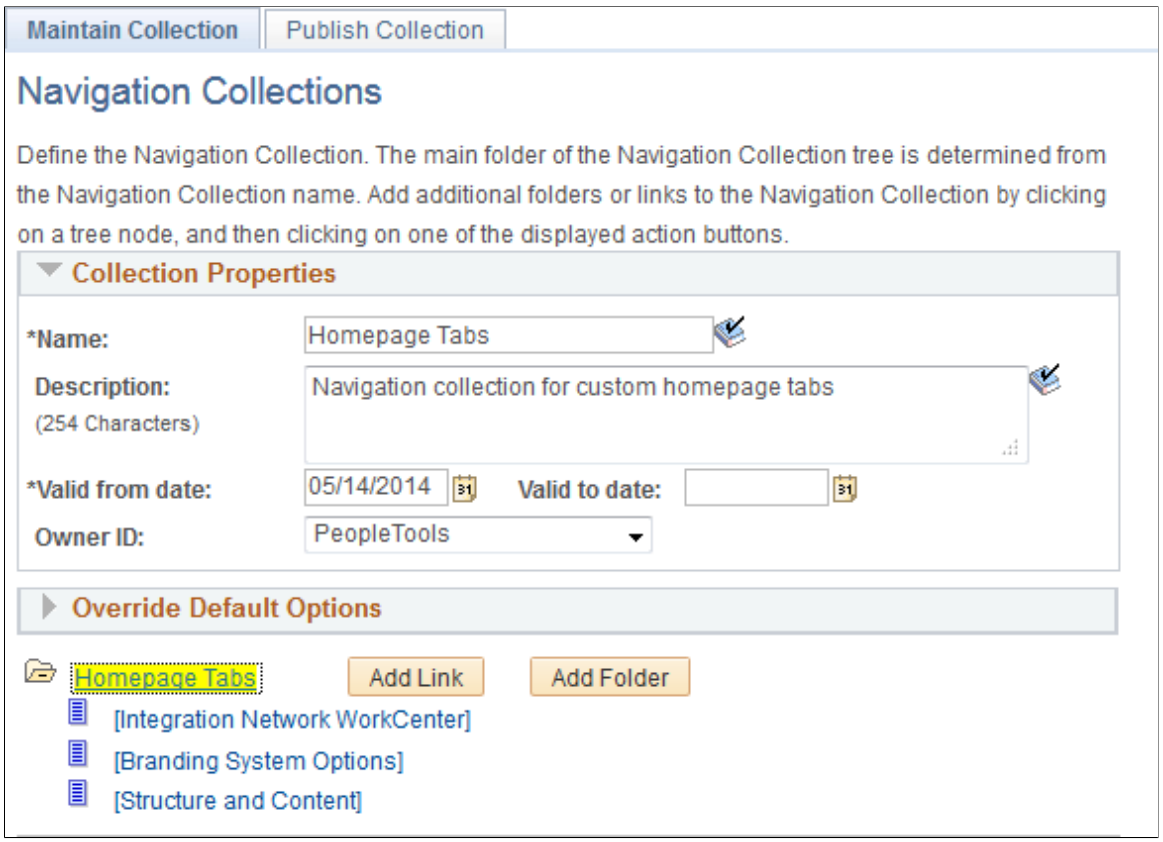
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

1. PeopleTools, Portal, Portal Utilities, Navigation Collections
2. Do one of the following:
   - Click the Add Collection link to create a new navigation collection.
   - Click the Search button and select an existing navigation collection to maintain.

Image: Navigation Collections page for a collection to be used as custom homepage tabs

This example illustrates the fields and controls on the Navigation Collections page for a navigation collection that will be used to display custom homepage tabs.



To serve as a custom homepage tab, a link must reside in the root folder of the navigation collection, and not in a subfolder.

To maintain a custom tabs navigation collection:

- Click the Add Link button to define a new local or remote link.
- Select an existing link and click the Edit Link button to edit an existing link.
- Select an existing link and click the Delete Link button to delete an existing link.

Important! Do not edit the properties for the Custom Tabs navigation collection delivered with PeopleSoft Interaction Hub.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Name	Enter a name for the navigation collection.
Description	Enter a description for this navigation collection.
Valid from date	Select the date on which you want this navigation collection to become active.
Valid to date	Select the date on which you want this navigation collection to become inactive.
Owner ID	Set the default owner ID for new definitions you create. Alternatively, leave the option blank, which is the default, to allow users to select the owner ID they want. The Owner ID helps identify which applications team last made a change to a definition.
Add Link	Click to access the Add Link page to define a link for a new custom tab. On the Add Link page, you can create a link to a local menu item or to a remote component based content reference. See Adding a Link, Selecting a Local Link, and Selecting a Remote Link for more information on creating local and remote links in navigation collections. **Note:** Do not select the Open in a new window check box. If you need to create a remote link to a non-component based content reference, then you will need to edit the custom tabs navigation collection through Structure and Content. See "Creating a Content Reference to a Remote, Non-Component-Based Item" for more information.
Add Folder	Click to add a subfolder to the navigation collection. **Important!** Links in subfolders are not supported for custom tabs navigation collection. Do not create a subfolder and links within the subfolder.
Edit Link	Click to access the Edit Link page for a link.
Delete Link	Click to access the Confirmation page to delete or cancel the deletion.

Maintaining Content References in a Custom Tabs Navigation Collection

You can use the Structure and Content component to maintain content references in a custom tabs navigation collection. This will allow you to:

- Disable a homepage tab by selecting the hide from portal navigation attribute.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Enable the display of a homepage tab (for example, one of the tabs delivered as sample data) by deselecting the hide from portal navigation attribute.
- Manually create a content reference to a remote non-component-based item.

To access a custom tabs navigation collection in the Structure and Content component:

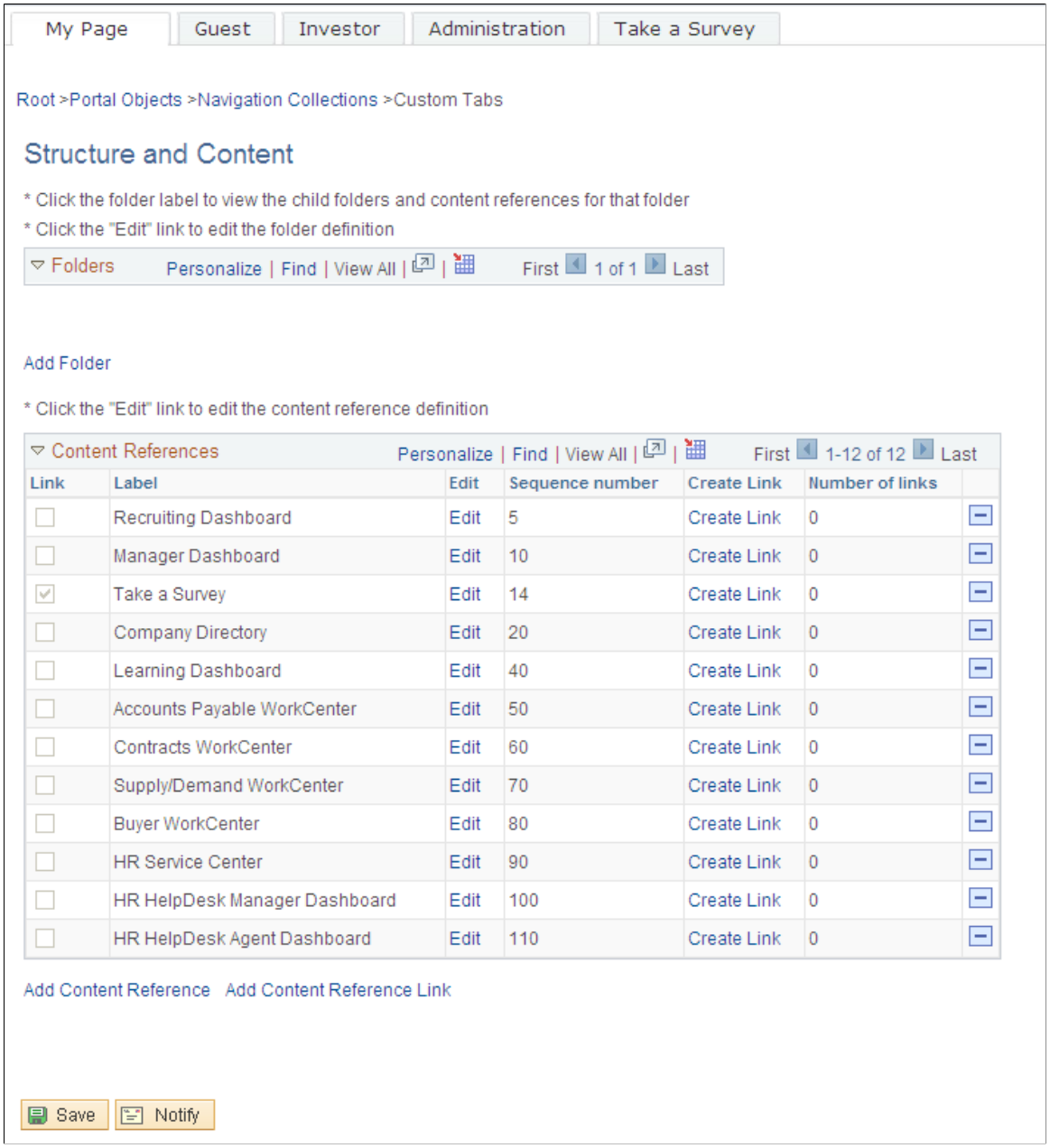
1. Select PeopleTools, Portal, Structure and Content.
2. Click the Portal Objects link.
3. Click the Navigation Collections link.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. Click the link for the custom tabs navigation collection.

Image: Structure and Content page for a custom tabs navigation collection

This example illustrates the fields and controls on the Structure and Content page for a custom tabs navigation collection.



To disable a custom homepage tab, click the Edit link for that content reference. Then, select the Hide from portal navigation option and save the definition.

To enable a custom homepage tab, click the Edit link for that content reference. Then, deselect the Hide from portal navigation option and save the definition.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating a Content Reference to a Remote, Non-Component-Based Item

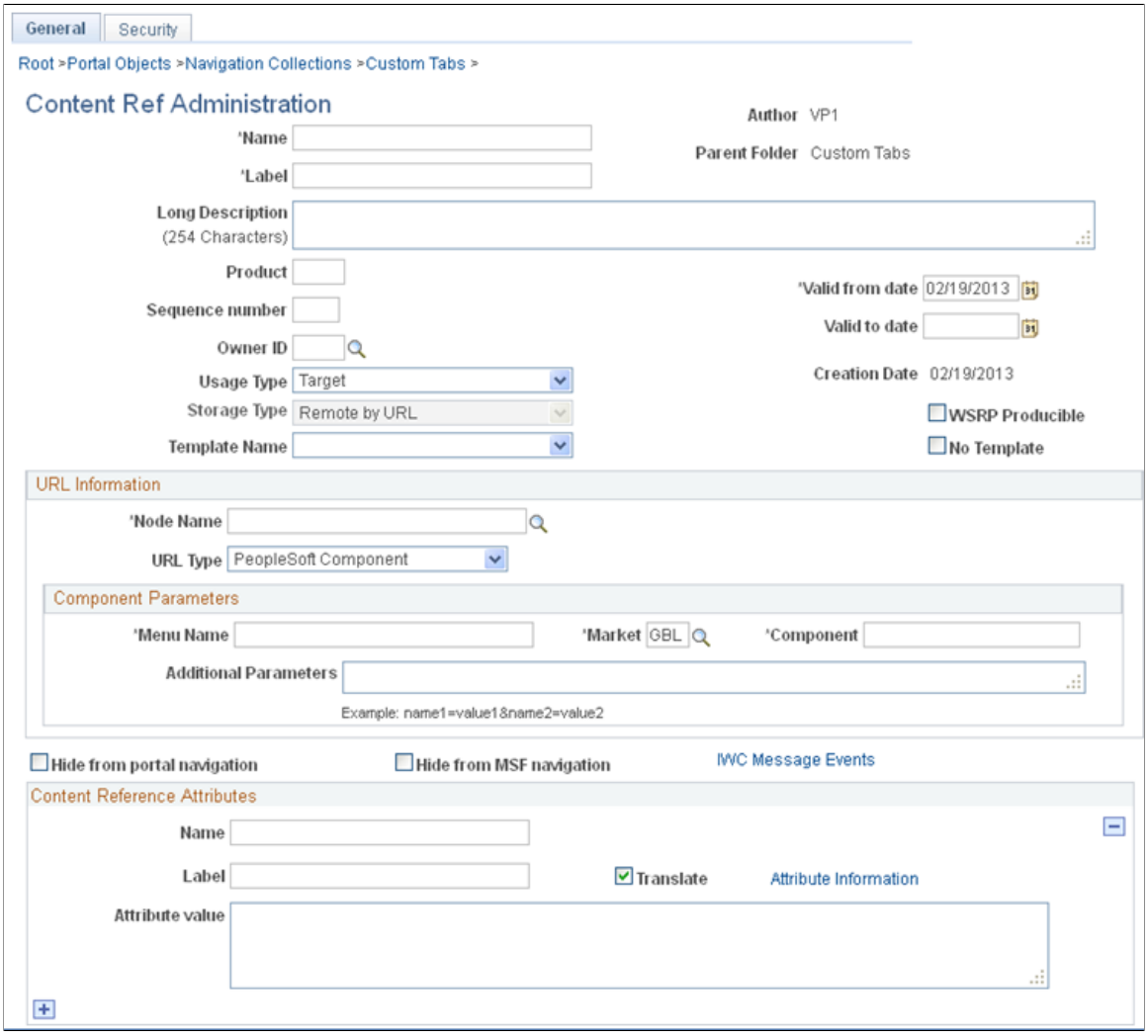
Note: Remote, non-component based content (such as a remote dashboard) must be accessible through unified navigation from the PeopleSoft Interaction Hub system before it can be designated as a homepage tab.

To manually create a content reference to a remote, non-component-based item such as a remote dashboard to be used as a custom homepage tab:

1. On the Structure and Content page click the link labeled Add Content Reference.

   Image: Content Ref Administration page

   This example illustrates the fields and controls on the Content Ref Administration page.



2. Specify the node name of the portal host node, such as HRMS, ERP, or LOCAL_NODE.

   Note: For a remote dashboard, the Node Name field can specify LOCAL_NODE or can be left blank.

3. In the URL Type field, select one of the following:

   - PeopleSoft Generic URL – For remote content from a PeopleSoft application.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Non-PeopleSoft URL – For remote content from a non-PeopleSoft system, such as an external web site.

4. In the Portal URL field, enter a partial or full URL depending on the URL type:

   - For a PeopleSoft Generic URL, navigate to the remote item through the menu from the local system. Copy the full URL. Then, remove the server, port, servlet, site, portal, and node from the URL, leaving you with a partial URL string.

     For example, when navigating using unified navigation from a local PeopleSoft Interaction Hub system, a partial URL string would be similar to the following:

     ```
     h/?tab=REMOTEUNIFIEDDASHBOARD&unifieddashboard=y&remotedburl=http%3a%2f%2⇒
     fserver%3aport%2fpsp%2fps%2fEMPLOYEE%2fHRMS%2fh%2f%3ftab%3dHC_HRS_RECRUIT⇒
     ING_HOMEPAGE_TAB%26pslnkid%3dHC_HRS_RECRUITING_HOMEPAGE_LNK&uninavpath=Re⇒
     cruiting%20Dashboard{HC_HRS_RECRUITING_HOMEPAGE_TAB}
     ```

     For example, a partial URL string for a remote SmartNavigation folder would be similar to the following:

     ```
     c/ROLE_EMPLOYEE.HRCD_CO_DIRECTORY.GBL?NAVSTACK=Clear&pt_fname=HC_COMPANY_⇒
     DIRECTORY&isleaf=false&abnpcode=WEBLIB_HRCD.ISCRIPT1.FieldFormula.IScript⇒
     _setFolderFromTree&abnds=t&TREE_NAME=COMPANY_DIRECTORY&TREE_SETID=&TREE_U⇒
     SERKEY=&TREE_EFFDT=&TREE_BRANCH=&abnnode=&cmd=uninav&unabnfldr=false&unin⇒
     avabnpcode=
     ```

   - For a Non-PeopleSoft URL, copy the full URL from the browser and paste it into the Portal URL field. For example:

     ```
     http://server.example.com/content/my_info.html
     ```

     Note: Alternatively, you can use the Node Definitions page to create a node definition for the non-PeopleSoft system, which would include the definition of the Content URI Text field—for example, `http://server.example.com/`. If you identify that non-PeopleSoft node through the NODE attribute as indicated in step 5, then you can specify a partial URL—for example, `content/my_info.html`. See Defining Portal Nodes for more information.

5. Define the following four content reference attributes to link the content reference to a remote menu item and enable real-time security checks:

Name	***Attribute Value***
CREFID	The name of the remote CREF as defined in the content provider system.
LOCATION	REMOTE
NODE	The portal host node name of the remote node.
PORTAL	The name of remote portal. This must be the same name as the current portal in the local system—for example, EMPLOYEE.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

6. Save the content reference definition.

Note: The Custom Tabs navigation collection delivered with PeopleSoft Interaction Hub 9.1 Rev 2 (and above) includes definitions for several remote, non-component based items that can be used as examples, including the Recruiting Dashboard and the Manager Dashboard, among others.

Related Links

Understanding Navigation Collections Used to Create Custom Homepage Tabs

Generating Theme Style Sheets for Fluid Components

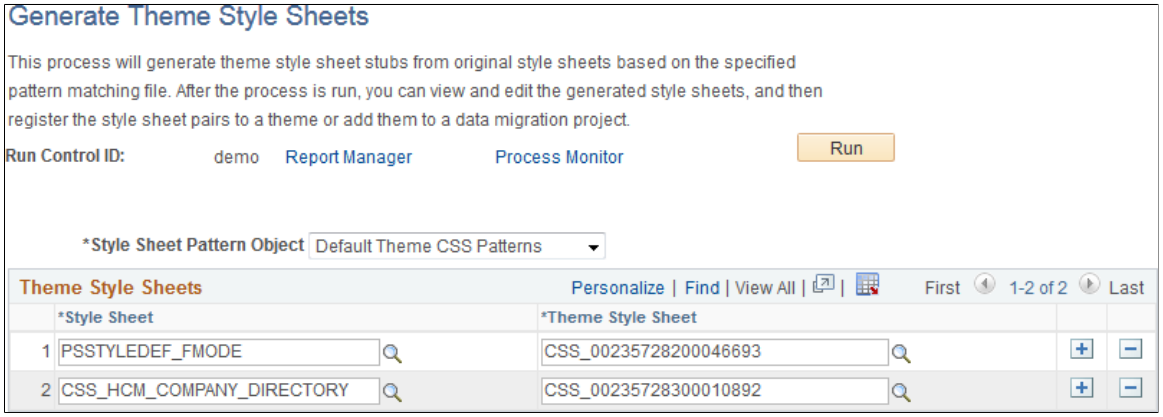
Use the Generate Theme Style Sheets page to create and maintain theme style sheet stubs for fluid components from original style sheets based on the specified pattern matching file.

Navigation

PeopleTools, Portal, Branding, Generate Theme Style Sheets

Image: Generate Theme Style Sheets page (Before Process Execution)

This example illustrates the fields and controls on the Generate Theme Style Sheets page before process execution.



Style Sheet Pattern Object

Select a style sheet pattern object. Currently, PeopleTools delivers one style sheet pattern object, Default Theme CSS Patterns, which will leave color attributes only in the generated style sheets.

Note: By definition, a style sheet pattern object is an HTML definition that begins with the PTBR_CSSPTN_ prefix.

Style Sheet

Select an original style sheet from which the mapped theme style sheet will be generated.

When the run control ID is first created, the system default style sheet for fluid components defined on the Branding System Options page will be added to the list. If none is defined, the PSSTYLEDEF_FMODE style sheet is added to the list.

Theme Style Sheet

Note: A unique, random name is generated initially when a style sheet is selected.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

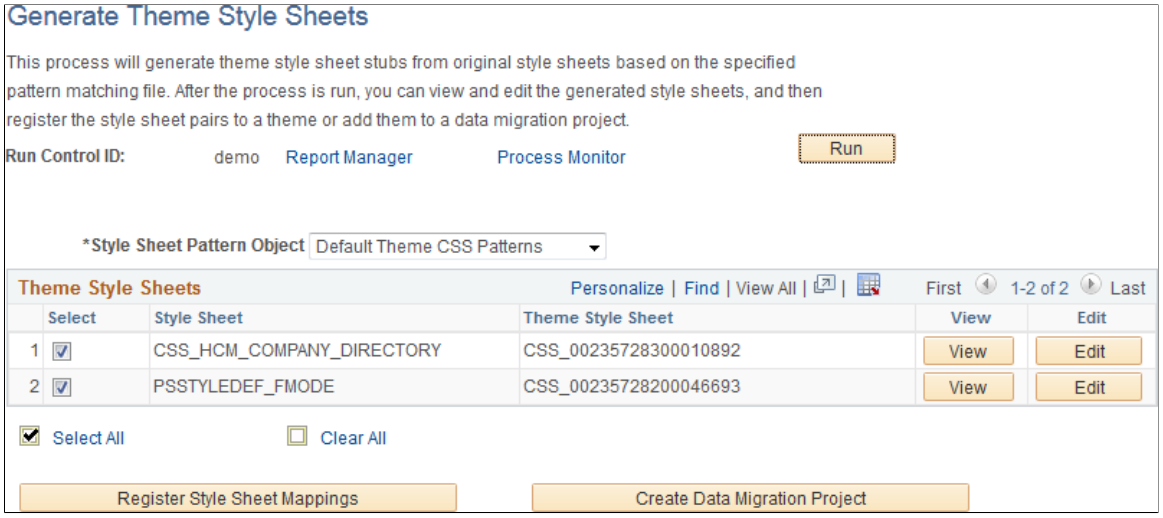
Enter a unique name for the mapped theme style sheet to be generated.

Note: You can use the prompt to look up style sheet objects already defined in the current system to avoid using a non-unique name.

Click the Run button to execute the process. After process execution, additional options are displayed on this page.

Image: Generate Theme Style Sheets Page (After Process Execution)

This example illustrates the fields and controls on the Generate Theme Style Sheets page after process execution.



Select
Select All
Clear All

Use the Select, Select All, and Clear All check boxes to select or deselect items for further processing.

View

Click the View button to view the content of the generated theme style sheet.

Note: This button is displayed only for users who have read permissions on the style sheet object.

Edit

Click the Edit button to edit the content of the generated theme style sheet. See Adding and Editing Branding Objects for more information.

Note: This button is displayed only for users who have both read and write permissions on the style sheet object.

Register Style Sheet Mappings

Click this button to register the selected style sheet pairs to a theme definition. At least one base style sheet/theme style sheet pair must be selected.

Create Data Migration Project

Click this button to add the selected theme style sheets and all images they use to a Data Migration Workbench project for

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

migration to a different database. At least one base style sheet/ theme style sheet pair must be selected.

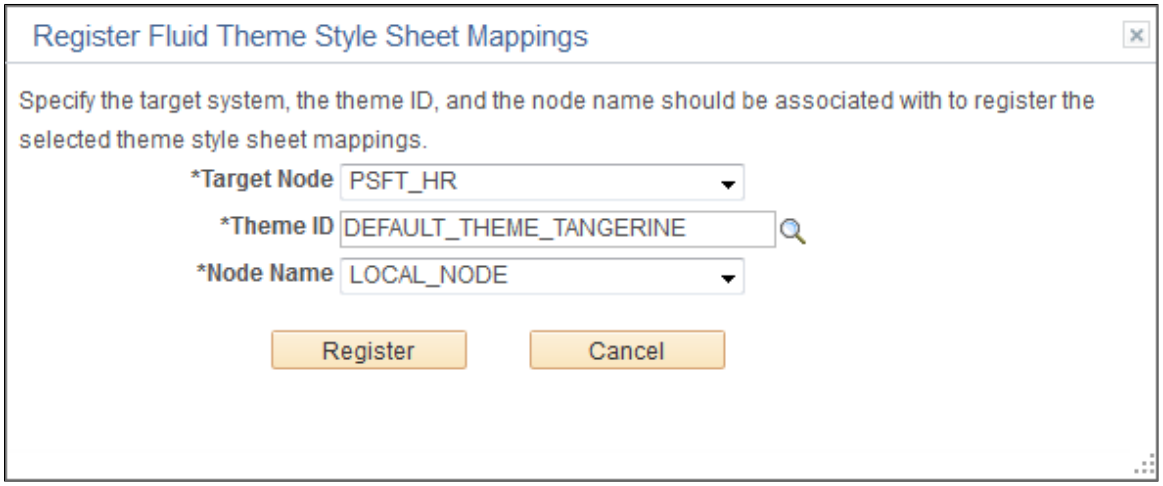
Note: A Data Migration Project Workbench project is also referred to as an Application Data Set (ADS).

Register Fluid Theme Style Sheet Mappings Page

Use the Register Fluid Theme Style Sheet Mappings page to register a mapped theme style sheet to the specified theme definition. This page provides an alternative to manually selecting the style sheet pair on the Assemble Theme Templates page.

Image: Register Fluid Theme Style Sheet Mappings Page

This example illustrates the fields and controls on the Register Fluid Theme Style Sheet Mappings page.



Target Node Select the default local node to identify the *target system.*

Theme ID Select the theme definition.

Node Name Select the node to which the style sheet and theme style sheet is to be applied. You can select a specific node name, or one of the following values:

- *ANY* – Applies to all nodes in all systems. (This setting is applicable to a cluster of applications managed by PeopleSoft Interaction Hub.)
- *LOCAL_NODE* – Applies to all nodes of the current system only.

Note: The list of specific nodes includes non-local PIA nodes that have a defined content URI.

Create Data Migration Project Page

Use the Create Data Migration Project page to add the selected theme style sheets and all images they use to a Data Migration Workbench project. You can use this project to migrate all of the theme style sheets

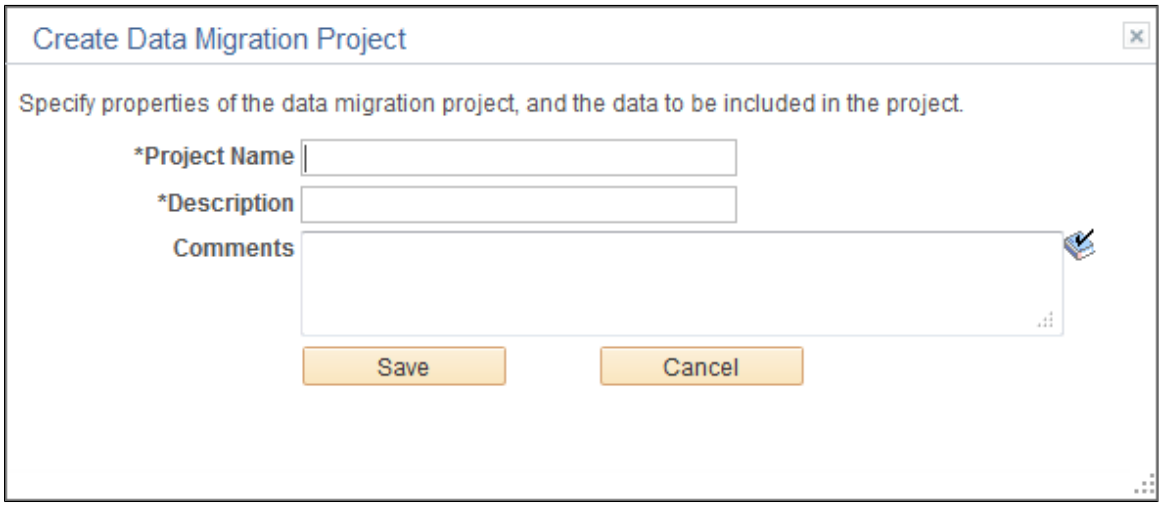
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

and images to a different system—for example, if you generate the theme style sheets on one system and then must distribute and register these style sheets on other systems in a cluster.

Once the project is created, you will be given the opportunity to open the project in Data Migration Workbench.

Image: Create Data Migration Project Page

This example illustrates the fields and controls on the Create Data Migration Project page.



Project Name	Enter a name for the data migration project.
Description	Enter a description for the data migration project.
Comments	Enter any additional comments that will appear with the project in Data Migration Workbench.

Related Links

Assembling Branding Themes

"Defining Data Migration Project" (PeopleTools 8.54: Data Migration Workbench)

Creating New Theme Style Sheets and Skins

For classic components, to create a new theme style sheet or skin:

1. In Application Designer, clone the DEFAULT_THEME_TANGERINE_ALT free form style sheet.

   The DEFAULT_THEME_TANGERINE_ALT contains style class definitions that override the style class definitions for the portal header found in PSSTYLEDEF_TANGERINE. The style class overrides in DEFAULT_THEME_TANGERINE_ALT are grouped as follows:

   - Header container
   - Logo
   - Header links
   - NavBar links
   - Global search widget

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Hover menu
- Menu icons

2. Modify one or more these override style classes in the new style sheet. Alternatively, you can delete style class overrides or an entire section to retain the settings from the PSSTYLEDEF_TANGERINE style sheet. See "Changing Font and Color Settings in Free Form Sub Style Sheets" (PeopleTools 8.54: Application Designer Developer's Guide).

3. Add additional override style class definitions to modify the look of pagelets or other items on your site. The following example provides free form style sheet overrides for pagelets:

```
/********************
 * Pagelets
 ********************/
#ptpglts .pthpli {
  margin-bottom: 10px;
}

#ptcol0 .PTPAGELETHEADER,
#ptpglts .PTPAGELETHEADER {
  background: url(%Image(MY_THEME_PAGELET_HEADER_BG)) repeat-x scroll 0px 0px ⇒
#78A4CB;
  border-color: #78A4CB;
  border-radius: 4px 4px 0 0;
  padding: 4px;
}

#ptcol0 .PTPAGELETHEADER .ptpgltlabel,
#ptpglts .PTPAGELETHEADER .ptpgltlabel {
  color: #333333;
  font-size: 12pt;
  font-weight: bold;
  font-family: Tahoma,Arial,Helvetica,Geneva,sans-serif;
}

#ptcol0 .PTPAGELETBODY,
#ptpglts .PTPAGELETBODY {
  border-color: #78A4CB;
  border-radius: 0 0 4px 4px;
  border-style: none solid solid solid;
  border-width: 1px;
  background-color: #FFFFFF;
}

/* Banner pagelets */
#ptcol0 .PTPAGELET {
  padding-%AlignStart: 4px;
  padding-%AlignEnd: 4px;
}

#ptcol0 .ptbannerpglt2 .PTPAGELETBODY {
  border-radius: 4px;
  border-top-style: solid;
}

#ptcol0 .ptbannerpglt3 .PTPAGELETBODY {
  border-radius: 4px;
  border-style: none;
}
```

4. To use the new style sheet:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Create a new theme template and select the new style sheet as the theme template for classic components.
- Modify an existing theme template and add the new style sheet as an available skin for classic components.

See Assembling Branding Themes.

For fluid components, to create a new theme style sheet:

1. Generate a mapped theme style sheet. See Generating Theme Style Sheets for Fluid Components.
2. Edit the new style sheet to modify one or more style class definitions.
3. Map the new theme style sheet to its base style sheet for one or more nodes for use with fluid components.
4. Assign the base style sheet as the default system style sheet for fluid components, or assign the base style sheet at the component or page level through PeopleCode.

Using Additional Style Sheets to Inject Your Custom Branding

To use additional style sheets to inject your custom branding:

1. Develop custom style sheets, custom JavaScript objects, or both to implement your custom branding. See Creating and Maintaining Style Sheet Objects and Creating and Maintaining JavaScript Objects.
2. Use these custom definitions by:
   - Applying them to classic components system-wide on the Branding System Objects page. See Configuring Branding System Options.
   - Applying them to individual components on the Component Branding page. See Branding Classic Components.

Example

The following provides and example of how you can use a free form style sheet to modify the display certain page elements and to invoke custom JavaScript to provide additional formatting:

1. Upload a new free form style sheet object. For example, upload the following as MY_BRANDING_DEMO:

```
/* Page Titles */
.PAPAGETITLE, .PSSRCHTITLE, .PSQRYTITLE {
  color: #FF0000;
}

/* Grid Border */
.PSLEVEL1GRIDWBO, .PSLEVEL2GRIDWBO, .PSLEVEL3GRIDWBO {
  border-color: #FF0000;
}

/* Group Box Border */
.PSGROUPBOXWBO {
  border-color: #FF0000;
}

/* Highlighter Active Field */
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
.FIELD_HIGHLIGHTER {
  border-color: #FF0000;
  background-color: #D2D2D2;
}
```

2. Upload a new JavaScript object. For example, upload the following as FIELD_HIGHLIGHTER:

```
(function($) {
  $(document).ready(function(){
    $(document).delegate("input[type='text'], input[type='password'], " +
        "select, textarea", "focus", function(e) {
      // Add a style class when activated
      $(this).addClass("FIELD_HIGHLIGHTER");
    }).delegate("input[type='text'], input[type='password'], " +
        "select, textarea", "blur", function(e) {
      // Remove a style class when not active
      $(this).removeClass("FIELD_HIGHLIGHTER");
    });
  });
})(jQuery);
```

3. To inject this custom branding system wide, select PeopleTools, Portal, Branding, Branding System Options.

4. In the Additional Style Sheet Objects group box, add the MY_BRANDING_DEMO style sheet.

5. In the Additional Javascript Objects group box, add this JavaScript object first: PT_JQUERY_1_6_2_JS.

6. Add a new row and add your new JavaScript as a second JavaScript object: FIELD_HIGHLIGHTER.

7. Save your changes.

8. Navigate to any page that includes a page title, group box, and grid. For example, select the User Profile - Roles page, or the Assemble Theme Templates page. Search and select value.

   Notice the style changes to the page title, group box borders, and grid borders introduced by the additional style sheet.

9. Place the cursor in any edit box or long edit box.

   Notice the highlighting changes to the field introduced by the additional JavaScript object, which calls jQuery to perform the dynamic highlighting.

Creating a New Branding Element

To create a new branding element:

1. Create an application class definition that extends the PTBR_BRANDING:Elements:BaseElement class.

   Use the PTBR_BRANDING:Elements:Basic* class definitions as examples.

2. Implement the following methods in the new class:

   - The class constructor to initialize the object. If the class includes additional attributes, ensure that the constructor also sets an initial value for these attributes.

     MyElement(*ID* as string)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- clone to create an exact copy of the current object.

  ```
  clone() Returns
  PTBR_BRANDING:Elements:BaseElement
  ```

- getHTML to return an HTML code fragment representing the object.

  ```
  getHTML(preview as boolean)
  Returns string
  ```

3. Optionally, implement the getStyleDefinitions method to return style definitions for the object as an in-line style definition block or an external style sheet link:

   ```
   getStyleDefinitions() Returns string
   ```

4. If the element includes additional attributes, implement the following:

   - A get property of type `PTBR_BRANDING:Attributes:attr_type`. The value of *attr_type* will depend on the type of attribute (text, image, HTML, and so on)—for example, `PTBR_BRANDING:Attributes:TextAttribute`.

   - getAttributeByID to return the additional attribute by ID.

     ```
     getAttributeByID(ID as string, create as boolean) Returns
     PTBR_BRANDING:Attributes:BaseAttribute
     ```

5. Create a new element type definition.

   See Defining Element Types.

Example

The following example provides the implementation of the PTBR_BRANDING:Elements:BasicText class.

```
import PTPP_PORTAL:UTILITY:Collection;
import PTBR_BRANDING:Attributes:BaseAttribute;
import PTBR_BRANDING:Attributes:TextAttribute;
import PTBR_BRANDING:Elements:BaseElement;



/**
  * BasicText Class
  */
class BasicText extends PTBR_BRANDING:Elements:BaseElement
   /* --- Properties --- */
   property PTBR_BRANDING:Attributes:TextAttribute Text get;


   /* --- Methods --- */
   method BasicText(&pId As string);
   method clone() Returns PTBR_BRANDING:Elements:BaseElement;

   method getAttributeByID(&pId As string, &pCreate As boolean) Returns PTBR_BRANDI⇒
NG:Attributes:BaseAttribute;

   method getHTML(&pPreview As boolean) Returns string;


private
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
   /* --- Private Properties --- */
   Constant &cstText = ".text";


   /* --- Private Methods --- */


end-class;



/**
  * Constructor
  *
  * @param pId ID of the object.
  *
  */
method BasicText
   /+ &pId as String +/

   %Super = create PTBR_BRANDING:Elements:BaseElement(&pId);

   %This.setElementType("BasicText");
   rem %this.setAdvancedOptionsPage (Page.);

   /* Initialize attributes */
   %This.StyleClass = "PSTEXT";
   %This.Text.setText("");

end-method;



/**
  * Make an exact copy of this object.
  * NOTE: Only create the object here, do not copy any class properties.
  *       If there are new class properties, override the copyProperties
  *       method and do it there.  Should invoke the %Super.copyProperties
  *       in that method though.
  *
  * @return BaseElement New object exactly matching this object.
  *
  */
method clone
   /+ Returns PTBR_BRANDING:Elements:BaseElement +/
   /+ Extends/implements PTBR_BRANDING:Elements:BaseElement.clone +/

   Local PTBR_BRANDING:Elements:BasicText &newObj = create PTBR_BRANDING:Elements:B⇒
asicText(%This.ID);

   /* Call the copy properties function */
   %This.copyProperties(&newObj);

   Return &newObj;

end-method;



/**
  * Get Attribute By ID
  *
  * @param pId     ID of the attribute.
  * @param pCreate True - create the attribute when not exist.
  *
  * @return BaseAttribute - Attribute object
  *
  */
method getAttributeByID
   /+ &pId as String, +/
   /+ &pCreate as Boolean +/
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
   /+ Returns PTBR_BRANDING:Attributes:BaseAttribute +/
   /+ Extends/implements PTBR_BRANDING:Elements:BaseElement.getAttributeByID +/

   Local PTBR_BRANDING:Attributes:BaseAttribute &getAttributeByID = %Super.getAttri⇒
buteByID(&pId, &pCreate);
   Local PTBR_BRANDING:Attributes:TextAttribute &TextAttribute;

   If ((&getAttributeByID = Null) And
         &pCreate) Then

      Evaluate &pId
      When &cstText
         &TextAttribute = create PTBR_BRANDING:Attributes:TextAttribute(&pId);
         &TextAttribute.Label = MsgGetText(219, 7509, "Text");
         &TextAttribute.setText("");
         &TextAttribute.IsVisible = True;
         &getAttributeByID = &TextAttribute;
         Break;

      End-Evaluate;

      If (&getAttributeByID <> Null) Then
         %This.Attributes.insert(&getAttributeByID);
      End-If;

   End-If;

   Return &getAttributeByID;

end-method;



/**
  * Get HTML of the element
  *
  * @param pPreview  True - preview mode.
  *
  * @return String - HTML
  *
  */
method getHTML
   /+ &pPreview as Boolean +/
   /+ Returns String +/
   /+ Extends/implements PTBR_BRANDING:Elements:BaseElement.getHTML +/

   Local string &getHTML = "";

   &getHTML = GetHTMLText(HTML.PTBR_ELM_CONTAINER_SPAN, EscapeHTML(%This.ID), Escap⇒
eHTML(%This.StyleClass), EscapeHTML(%This.Text.getRuntimeValue()));

   Return &getHTML;

end-method;



/****************************
*   Get/Set method pairs
****************************/

/**
  * Text Property
  *
  */
get Text
   /+ Returns PTBR_BRANDING:Attributes:TextAttribute +/

   Return (%This.getAttributeByID(&cstText, True) As PTBR_BRANDING:Attributes:TextA⇒
ttribute);
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
end-get;
```

Creating a New User Attribute Type

To create a new user attribute type:

1. Create an application class definition that extends the PTBR_BRANDING:UserAttributes:BaseUserAttribute class.

   Use the PTBR_BRANDING:UserAttributes:* class definitions as examples.

2. Implement the following methods in the new class:

   - The class constructor to initialize the object.

     ```
     MyUserAttribute(ID as string)
     ```

   - getPromptViewName to return the record definition to be used as the search prompt view. If no search prompt view exists, return an empty string.

     ```
     getPromptViewName() Returns string
     ```

   - validateValue to return a Boolean value indicating whether the user has the specified attribute.

     ```
     validateValue(value as string) Returns boolean
     ```

3. Create a new user attribute type definition.

   See Defining User Attribute Types.

Example

The following example provides the implementation of the PTBR_BRANDING:UserAttributes:RoleBasedUserAttribute class.

```
import PTBR_BRANDING:UserAttributes:BaseUserAttribute;



/**
  * RoleBasedUserAttribute Class
  */
class RoleBasedUserAttribute extends PTBR_BRANDING:UserAttributes:BaseUserAttribute
   /* --- Properties --- */


   /* --- Methods --- */
   method RoleBasedUserAttribute(&pId As string);

   method getPromptViewName() Returns string;
   method validateValue(&pValue As string) Returns boolean;


end-class;



/**
  * Constructor
  *
  * @param pId ID of the object.
  *
  */
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
method RoleBasedUserAttribute
   /+ &pId as String +/

   %Super = create PTBR_BRANDING:UserAttributes:BaseUserAttribute(&pId);

   %This.setUserAttributeType("RoleBasedUserAttribute");

end-method;



/**
  * Return the role prompt view name
  *
  * @return string - the prompt view name
  *
  */
method getPromptViewName
   /+ Returns String +/
   /+ Extends/implements PTBR_BRANDING:UserAttributes:BaseUserAttribute.getPromptVi⇒
ewName +/

   Return Record.PSROLEDEFN_SRCH;

end-method;



/**
  * Validate whether the current has this role
  *
  * @return boolean - the result
  *
  */
method validateValue
   /+ &pValue as String +/
   /+ Returns Boolean +/
   /+ Extends/implements PTBR_BRANDING:UserAttributes:BaseUserAttribute.validateVal⇒
ue +/

   Return IsUserInRole(&pValue);

end-method;
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

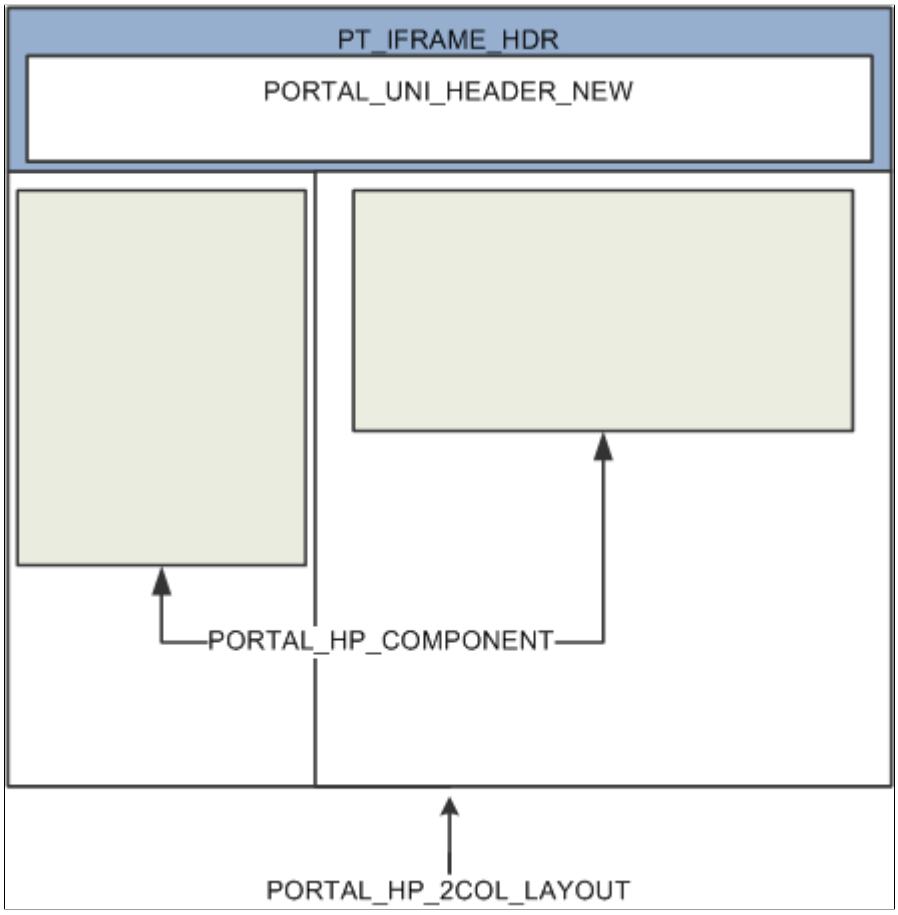
Chapter 7

Modifying the Portal Interface

Using Predefined Template HTML Definitions

Image: Predefined HTML definitions that comprise portal templates

The following example shows the arrangement of some of the HTML definitions that make up a template. The PORTAL_HP_USER_TEMPLATE HTML definition controls the overall page. The PORTAL_UNI_HEADER_NEW HTML definition controls the universal navigation header. The PORTAL_HP_COMPONENT HTML definition controls the appearance of each pagelet and appears twice in this two-column template.



This table describes the HTML definitions:

HTML Definition	*Description*
PORTAL_HP_2COL_LAYOUT	This is the HTML definition for a two-column user homepage. The default column widths are set to 33 percent for column one and 66 percent for column two.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

HTML Definition	*Description*
PORTAL_HP_3COL_LAYOUT	This is the HTML definition for a three-column user homepage. The default column width is set to 33 percent for each column.
PORTAL_HP_ACTIONBAR	Use this HTML definition to create the pagelet action bar. The default action bar buttons are Remove and Minimize/Restore.
PORTAL_HP_ACTIONBARIMAGE	Use this HTML definition to create the Customize image button for the pagelet action bar. The image button is added only if you've defined an advanced personalization page.

PORTAL_HP_COMPONENT	Use this HTML definition to create all homepage pagelets. Modify it to change a pagelet's appearance.
	
	Do not move, delete, or change the first or last line of this HTML definition:
	
	`<!-- Begin Pagelet=%BIND(:6) -->.`
	
	`<!-- End Pagelet=%BIND(:6) -->`
	
	These special lines must exist exactly as shown. You must also ensure that all bind variables exactly match the bind variables that are used in the associated PeopleCode.

PORTAL_HP_USER_TEMPLATE	Each user homepage is generated from this HTML definition. Modify this definition to add default components that are displayed to all users.
PORTAL_UNI_HEADER_NEW	This HTML definition controls the universal header for the portal.

With the exception of the PORTAL_UNI_HEADER_NEW HTML definition, all homepage HTML definitions are built in the BuildStaticHPTemplate function.

This function is located in FUNCLIB_PORTAL.TEMPLATE_FUNC.FieldFormula.

The homepage modification functionality (minimize, maximize, and remove) is located in WEBLIB_PORTAL.PORTAL_HOMEPAGE.FieldFormula.

The iScripts are IScript_HPCompRemove, IScript_HPCompMinimize, and IScript_HPCompExpand.

Using Predefined Homepage HTML Definitions

This section discusses how to:

- Construct homepages.
- Construct headers.
- Construct pagelets.
- Adapt homepage and pagelet definitions.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Constructing Homepages

Image: Graphical representation of the HTML definitions that comprise a homepage

This example shows the layout that is used for displaying a homepage with a header and pagelets. The PORTAL_HP_USER_TEMPLATE object serves as a container for the other HTML definitions found on the homepage.

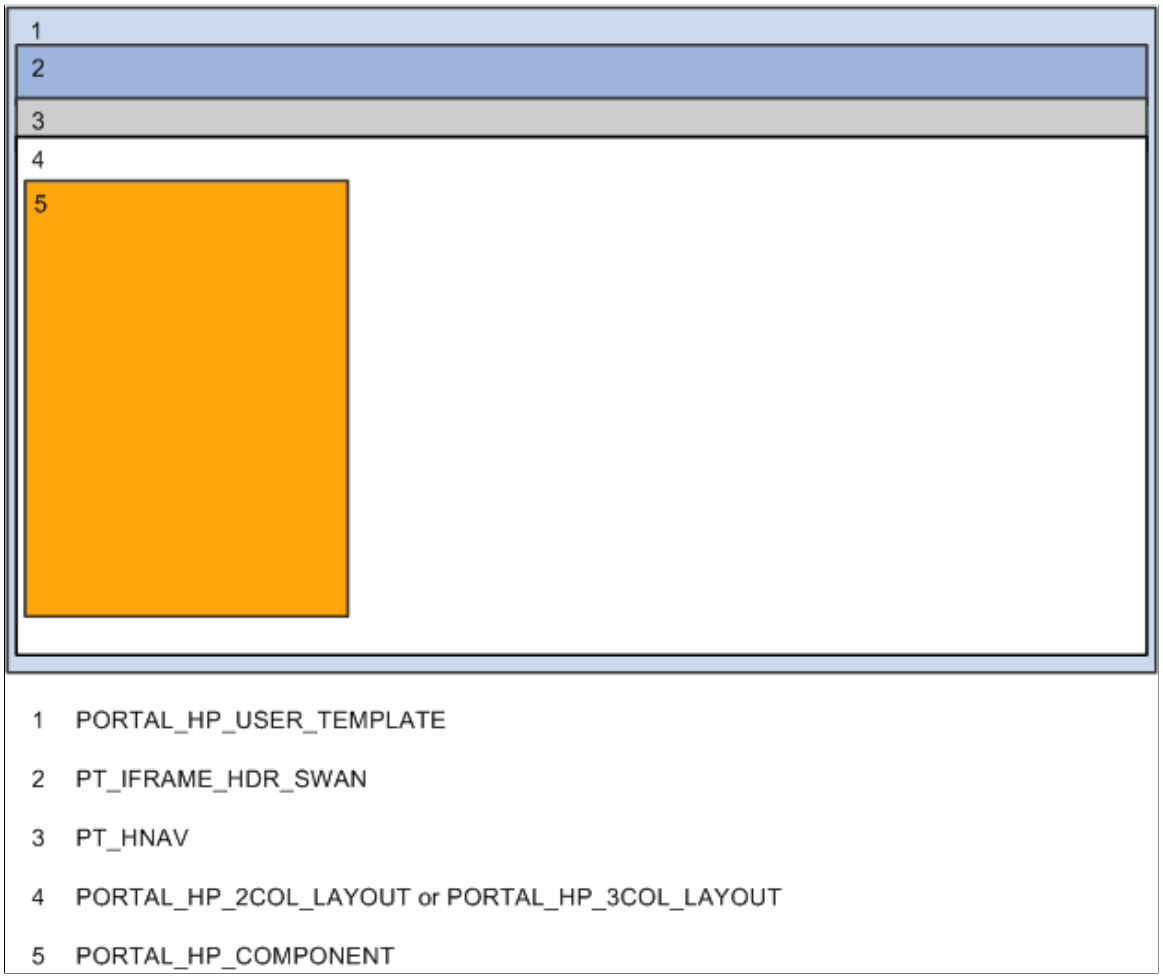


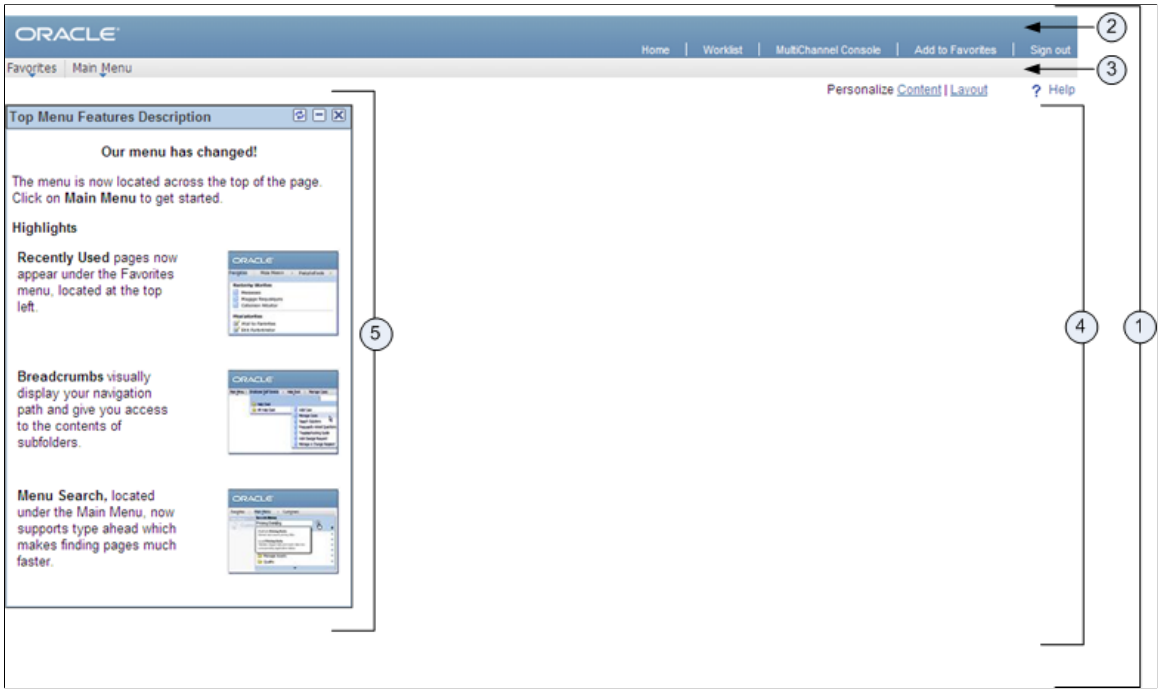
Image: PeopleSoft homepage showing the composite HTML definitions

This example shows an actual PeopleSoft homepage with numbers specifying these HTML definitions:

1. PORTAL_HP_USER_TEMPLATE
2. PORTAL_UNI_HEADER_NEW
3. PT_HNAV
4. PORTAL_HP_2COL_LAYOUT or PORTAL_HP_3COL_LAYOUT

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

5. PORTAL_HP_COMPONENT



This table describes the HTML definitions that the system uses to construct homepages:

HTML Definition	***Description***
PORTAL_HP_USER_TEMPLATE	This is the HTML definition for the homepage tab. This definition contains the iScript reference to the universal header, a bind variable for the two-column or three-column layout HTML, and binds for the Oracle image.
PORTAL_UNI_HEADER_NEW	This HTML definition contains all the HTML for the header portion of a homepage tab. The HTML definition is set up in WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula. GetUniHeaderHTML.
PT_IFRAME_HDR	This HTML definition contains all of the HTML for the iframe header.
PT_HNAV	This HTML definition contains the HTML for the drop-down navigation header.
PORTAL_HP_2COL_LAYOUT	This HTML definition contains all the pagelet HTML for the two-column layout (columns 1 and 2).
PORTAL_HP_2COL_LAYOUT_ACCESS	This HTML definition contains all the pagelet HTML for the two-column layout (columns 1 and 2) when the homepage is in Accessibility mode.
PORTAL_HP_3COL_LAYOUT	This HTML definition contains all the pagelet HTML for the three-column layout (columns 1, 2 , and 3).
PORTAL_HP_3COL_LAYOUT_ACCESS	This HTML definition contains all the pagelet HTML for the three-column layout (columns 1, 2 , and 3 when the homepage is in Accessibility mode.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

HTML Definition	*Description*
PORTAL_HP_COMPONENT	This is the HTML definition for pagelets.

Constructing Headers

This table describes the HTML definitions that the system uses to construct headers:

HTML Definition	*Description*
PORTAL_HP_TABS	This is the HTML definition for all tabs. Use this HTML definition when more than one system tab exists for a user. This HTML definition is set up in WEBLIB_PORTAL. PORTAL_HOMEPAGE.FieldFormula.GetHPTabHTML().
PORTAL_HP_INACTIVE_TAB	This is the HTML definition for the inactive tab. The inactive tab name is linked. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HOMEPAGE.FieldFormula. GetHPTabHTML().
PORTAL_HP_ACTIVE_TAB	This is the HTML definition for the active tab. Only one active tab can exist. The tab name is static text. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_ HOMEPAGE.FieldFormula.GetHPTabHTML().
PORTAL_HP_PERSONALIZE_LINKS_TD	This is the HTML definition for the Personalize Content and Personalize Layout links. This HTML definition is used only when more than one system tab exists for a user. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_ HOMEPAGE.FieldFormula.GetHPTabHTML().
PORTAL_HP_PERSONALIZE_LINKS	This is the HTML definition for the Personalize Content and Personalize Layout links. This HTML definition is used only when one system tab exists for a user. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HOMEPAGE. FieldFormula.GetHPTabHTML().
PORTAL_HEADER_ADDTOFAVORITES	This is the HTML definition for the Add To Favorites link. This HTML definition is set up in WEBLIB_PORTAL. PORTAL_HEADER.FieldFormula.GetUniHeaderHTML().
PORTAL_HEADER__ADDTOFAVSFORM	This is the HTML definition for the Add to Favorites form. This form contains the PortalActualURL and PortalContentURL hidden fields. The Add to Favorites page uses these hidden fields to construct the content reference to the page to which you are adding a favorite (PortalContentURL) and a link back to the actual page (PortalActualURL). This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula. GetUniHeaderHTML().
PORTAL_HEADER_WL	This is the HTML definition for the View Worklist link. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_ HEADER.FieldFormula.GetUniHeaderHTML().

Constructing Pagelets

This table describes the HTML definitions that the system uses to construct pagelets:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

HTML Definition	*Description*
PORTAL_HP_ACTIONBAR	This is the HTML definition for the pagelet action bar. For a personalized tab, the pagelet action bar always includes a Maximize image button. It may also contain up to three more image buttons (Remove, Edit, and Help). This HTML definition is not populated for a pagelet on a system tab.
PORTAL_HP_ACTIONBAR_EDIT	This is the HTML definition for the Edit image button on the pagelet action bar.
PORTAL_HP_ACTIONBAR_HELP	This is the HTML definition for the Help image button on the pagelet action bar.
PORTAL_HP_ACTIONBAR_REFRESH	This is the HTML definition for the Refresh image button on the pagelet action bar.
PORTAL_HP_ACTIONBAR_REMOVE	This is the HTML definition for the Remove image button on the pagelet action bar.
PORTAL_HP_COMPONENT	This is the HTML definition for pagelets.
PT_HP2_BANNER	This is the HTML definition for the banner pagelet.
PT_HP2_PGLTHNDL	This is the HTML definition for the pagelet label when the pagelet cannot be dragged.
PT_HP2_PGLTHNDL_DRAG	This is the HTML definition for the pagelet label when the pagelet can be dragged.
PT_HP2_PGLT_HDR_IMG	This is the HTML definition for the pagelet header image.

Adapting Homepage and Pagelet Definitions

This section discusses how to:

- Build portal homepage tabs by using fixed-name HTML and image definitions.
- Add images to homepage tabs.

Building Portal Homepage Tabs by Using Fixed-Name HTML and Image Definitions

PeopleSoft applications use fixed-name HTML and image definitions to build portal homepage tabs. You can override default HTML definitions and images that are used for homepage elements, such as tabs and pagelets, by adding attributes on the Content Ref Administration page.

Access the Content Ref Administration page (Select PeopleTools, Portal, Structure and Content. Click the Portal Objects link.)

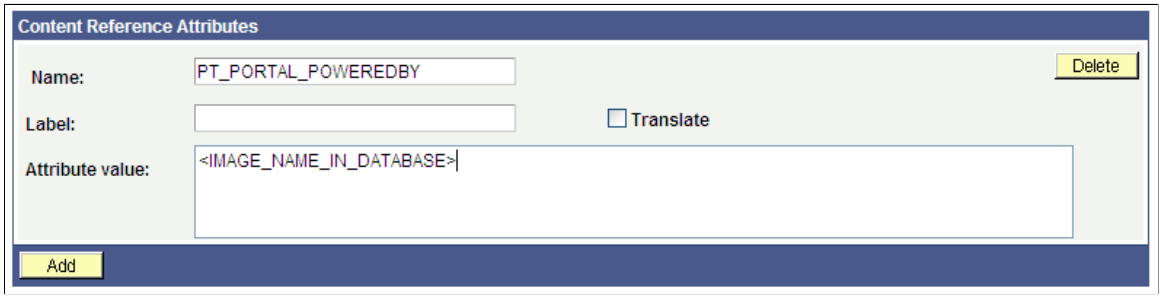
To work with homepages, click the Homepages link, then the Tabs link. Click a homepage tab label.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To work with pagelets, click the Pagelets link, then a pagelet group name link. Click a pagelet label.

Image: Content Reference Attributes group box of the Content Ref Administration page

The following screenshot illustrates building portal homepage tabs using fixed-name HTML and image definitions.



To override a default fixed-name HTML or image definition:

1. In the Content Reference group box at the bottom of the page, click the Edit link for a content reference.
2. Scroll to the Content Reference Attributes group box for a content reference.
3. In the Name field, enter the name of the default definition that you want to change.
4. Deselect the Translate check box.

   Note: You must deselect the Translate check box for the override to be successful.

5. In the Attribute Value field, specify the new definition value.

   Note: The value that you specify can consist of letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.

6. Click Save.

You can omit the Label field value.

As shown in the example, to replace the Powered by PeopleSoft image on the homepage tab, add a nontranslatable attribute named PT_PORTAL_POWEREDBY with a new image definition name in the database.

This table shows a list of default definitions that you can change for a tab:

HTML or Image Definition	***Value***
PT_PORTAL_POWEREDBY	*Image name in database.*
PT_PORTAL_IC_REFRESH	*Image name in database.*
PT_PORTAL_IC_EDIT	*Image name in database.*
PT_PORTAL_IC_CLOSE	*Image name in database.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

HTML or Image Definition	*Value*
PT_PORTAL_IC_COLLAPSE	*Image name in database.*
PT_PORTAL_IC_EXPAND	*Image name in database.*
PT_PGLT_HELP	*Image name in database.*
PORTAL_HP_COMPONENT	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR_REFRESH	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR_REMOVE	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR_EDIT	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR_HELP	*HTML definition name in database.*
PORTAL_HP_2COL_LAYOUT	*HTML definition name in database.*
PORTAL_HP_3COL_LAYOUT	*HTML definition name in database.*
PORTAL_HP_USER_TEMPLATE	*HTML definition name in database.*

Adding Images to Homepage Tabs

You can specify up to 15 extra images for a homepage by using %BIND variables on PORTAL_HP_COMPONENT and PORTAL_HP_ACTIONBAR_xxx HTML definitions. Homepage generation logic replaces %BIND12 to %BIND26 on the homepage tab by the images that are specified as attributes.

Note: Bind variables for images and all HTML components (tab and pagelet) should be nontranslatable. To make an item nontranslatable, deselect the Translate check box.

The following %BIND variables can be replaced by the Homepage Generation service (specify bind variables in PeopleSoft Application Designer):

Bind Variable	*Definition Replaced*
%BIND(:1)	The PT_PORTAL_POWEREDBY image (Powered By PeopleSoft image).
%BIND(:2)	The result of a function call to %Request.Full URI.
%BIND(:3)	The result of a function call to %Request.GetHelpURL('PERSHOEPAGE').
%BIND(:4)	The result of a function call to refreshonExpire.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Bind Variable	*Definition Replaced*
%BIND(:5)	The PT_PGLT_HELP image.
%BIND(:6)	The result of a function call to %Request.GetHelpURL(&Pglt.HelpID).
%BIND(:7)	The PT_PORTAL_IC_EDIT image (edit image for a pagelet).
%BIND(:8)	The result of a function call to SetRefreshPage.
%BIND(:9)	The PT_PORTAL_IC_CLOSE image.
%BIND(:10)	The PT_PORTAL_IC_COLLAPSE image.
%BIND(:11)	The PT_PORTAL_IC_EXPAND image.
%BIND(:12) to %BIND(:26)	The value of the tab attribute IMAGE_BIND_*X,* where *X* ranges from 12 to 26.
%BIND(:28)	The PT_PORTAL_IC_REFRESH image.

This code sample shows how to specify custom %BIND images to display custom images for the PORTAL_HP_COMPONENT HTML definition:

```
<!-- Begin Pagelet=%BIND(:5) -->
<!-- PageletState=MAX -->
<TABLE id="" CLASS="PTPAGELET" WIDTH="100%" CELLPADDING="0" CELLSPACING="0" BORDER=⇒
"1">
<TR>
     <TD>
          <!--Image 12 will be displayed here.-->
          <IMG SRC="%BIND(:12)" NAME="anything you like">
          <!--Image 13 will be displayed here.-->
          <IMG SRC="%BIND(:13)" NAME="anything you like">
          <!--Image 14 will be displayed here.-->
          <IMG SRC="%BIND(:14)" NAME="anything you like">
          <!--Image 26 will be displayed here.-->
          <IMG SRC="%BIND(:26)" NAME="anything you like">
     </TD>
</TR>
   <TR>
      <TD CLASS="PTPAGELETHEADER">
         %BIND(:1)
         %BIND(:2)</nobr>
      </TD>
   </TR>
   <TR>
      <TD CLASS="PTPAGELETBODY" WIDTH="100%">
         <Pagelet Name="%BIND(:2)">
            <Source Pagelet="%BIND(:4)" href="%BIND(:3)" />
         </Pagelet>
      </TD>
   </TR>
</TABLE>
<BR>
<!-- End Pagelet=%BIND(:5) -->
```

The HTML definitions PORTAL_HP_2COL_LAYOUT and PORTAL_HP_3COL_LAYOUT HTML can be modified similarly to display extra images.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The Homepage Generation service replaces the image that is specified by the attribute IMAGE_BIND_12 in place of bind variable %BIND(:12), and so on.

Warning! Pagelet HTML is parsed by the portal servlet. Be sure that the HTML is properly formatted and that none of the matching tags are missing. Improperly formatted HTML and missing tags may result in unexpected web server behavior.

Using Custom Portal Headers

The following list presents custom headers that you can use with the portal:

- PortalServletURI
- PortalURI
- PortalActualURL
- PortalContentURL
- PortalContentProvider
- PortalRegistryName
- PortalTargetFrame
- PortalPagelet

These are valid parameters that you can pass with the PeopleCode GetParameter method. PortalContentURL retrieves the registered URL, while PortalActualURL retrieves the entire URL, including any additional query string parameters.

Changing Link Colors

Use the PSSTYLEDEF style sheet to change link colors.

To change the link color:

1. Open the PSSTYLEDEF style sheet in PeopleSoft Application Designer.
2. Locate and open the PSHYPERLINK class.
   a. Modify the attributes as appropriate.
   b. On the Properties page, modify the pseudo classes.
   c. Specify default font attributes for <A> and <TD> in your own style sheet or in an embedded style sheet at the top of PORTAL_UNI_HEADER_NEW.

      This applies font attributes to text that has no class associated with it. For example:

```
<STYLE TYPE="text/css">
 td { font-family:arial; font-size:9pt; }
   a:visited { font-family:arial; font-size:9pt; color: #5c93ae; text-dec⇒
oration:underline }
   a:link { font-family:arial; font-size:9pt; color: #5c93ae; text-decora⇒
tion: underline }
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
   a:hover { font-family:arial; font-size:9pt; color: #5c93ae; text-decor⇒
ation: underline }
   a:active { font-family:arial; font-size:9pt; color: #5c93ae; text-deco⇒
ration: underline }
</STYLE>
```

Note: Oracle recommends that you do not alter the delivered PSSTYLESHEET style sheet definition. Make a copy of PSSTYLESHEET and modify the copy.

Related Links

"Understanding Style Sheets and Style Classes" (PeopleTools 8.54: Application Designer Developer's Guide)

Applying Changes

To immediately view changes that you make to a page, click the browser Refresh button. This reloads the updated page, instead of waiting for the old page to expire and the new page to be loaded.

These functions, which are found in FUNCLIB_PORTAL.TEMPLATE_FUNC FieldFormula, are also available to force a refresh:

Function	***Description***
`ForceRefreshPage(&url)`	Refreshes the default tab for the current user.
`&myPage = URLEncode("?tab=DEFAULT&user=" | %userid);`	Forces the page that is identified by the parameter to refresh. The parameter is the escaped query string of the page that is being refreshed, including the question mark.
`ForceRefreshPage(&myPage);`	Refreshes the default tab for the current user.
`ForceRefreshHomePageTab(&tab)`	Refreshes the homepage tab for the current user with the tab name that is passed in.
`ForceRefreshHomePage()`	Refreshes the default homepage tab of the current user.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 8

Working with Portal Templates

Understanding Portal Templates

The PeopleSoft portal integrates content from numerous sources and presents the merged content on a single web page in a coherent, consistent fashion that keeps users within the portal framework. A portal template tells the portal servlet what content to place on the assembled page and where each piece of content should be placed.

A portal template is HTML code. However, in addition to standard HTML tags, the portal template can include PeopleSoft tags that are used for giving instructions to the portal servlet during page assembly. A normal browser cannot interpret these special tags. The portal servlet follows instructions in the PeopleSoft tags and then removes them before passing the final page back to the user's browser.

The content in a template falls into four categories:

- HTML for the template itself.
- HTML for the main target content.
- HTML for the related target content.
- Additional HTML, such as for the navigation header.

Applying Template Types

This section provides an overview of template types and discusses how to:

- Use page-based static templates.
- Use frame-based static templates.
- Use frame-based dynamic templates.
- Use iframe-based dynamic templates.

Understanding Template Types

Each portal template is either page-based, frame-based, or inline frame-based (iframe-based), and each template is either static or dynamic. Template types come in five possible combinations:

- Page-based static templates.

  These are content references with a usage type of *HTML template* and a storage type of *Local (in HTML Catalog)*.

- Page-based dynamic templates.

  These are content references with a usage type of *HTML template* and a storage type of *Remote by URL*.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Frame-based static templates.

  These are content references with a usage type of *Frame template* and a storage type of *Local (in HTML Catalog)*.

- Frame-based dynamic templates.

  These are content references with a usage type of *Frame template* and a storage type of *Remote by URL*.

- iFrame-based dynamic templates.

  These are content references with a usage type of *Inline frame template* and a storage type of *Remote by URL*.

A page-based template uses HTML tables to generate a page. Content is placed in each table cell that is defined by the template. The portal servlet uses the page-assembly process to retrieve documents for each cell. The assembled page is then sent as a single unit to the user's browser.

A frame-based template uses frames, rather than HTML tables, to divide the page into parts. The portal servlet constructs the appropriate URL for each frame (<src> tag in the frameset) and sends the entire frameset to the browser. The browser then retrieves the content for each frame.

An iframe-based template uses HTML with embedded inline frames to divide the page into parts. The portal servlet constructs the appropriate URL for each section of the page and sends the HTML to the browser. The browser then retrieves the content for each section and each inline frame.

A static template is based on HTML code that is entered into the HTML area of a template content reference (a content reference that exists in the Portal Objects\Templates folder of the portal registry). Bind variables cannot be used within the HTML area. The HTML includes any HTML that is required for the template itself, plus tags that specify the URLs for template pagelets. You can view and update the contents of a static template on the portal administration pages.

A dynamic template is retrieved from the web in real time, as part of the page-assembly process, rather than stored in the database with the content reference. A dynamic template can use bind variables. You cannot view the HTML contents of the dynamic template directly on the portal administration pages. Instead, the portal administration pages enable you to identify the URL that points to the resource that provides the template content. If the dynamic template is implemented by an iScript, you can use PeopleSoft Application Designer to navigate to the record and field where the iScript exists. The content provided by the URL is the HTML that is used for the dynamic template at runtime.

See Managing Portal Objects.

Using Page-Based Static Templates

This static template example combines a universal navigation header with target content. It is based on HTML tables, not frames.

```
<html>
<head>
</head>
<body>
<table>
<tr><td>
<Pagelet Name="UniversalNavigation">
    <SOURCE Node="LOCAL_NODE" href="s/WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula.
   IScript_UniHeader" />
</Pagelet>
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
</td></tr>
<tr><td>
<Target Name="TargetContent"/>
</td></tr>
</table>
</body>
</html>
```

This template is composed of HTML for three items:

- The template itself.

  Some HTML is required for the overall template. The code for this template is represented in the preceding example by all the nonemphasized text. This HTML code remains on the assembled page that is sent to the user's browser after the page-assembly process.

- A template tag for the universal navigation header pagelet.

  This is the first block of HTML code that is emphasized in the preceding example. At runtime, the portal servlet replaces the Pagelet tag with the navigation header in the template for the final assembled page.

- A template tag for target content.

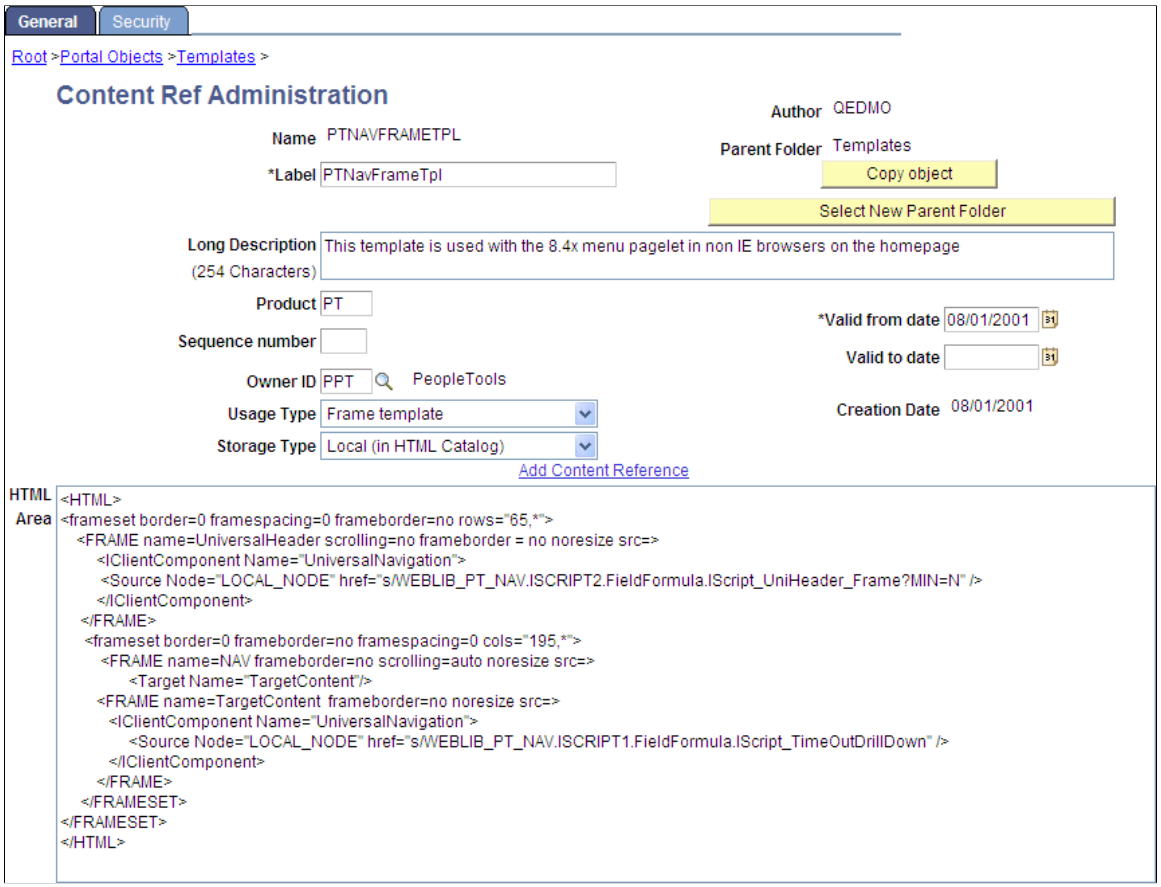
  At runtime, the portal servlet replaces the Target tag—the second block of emphasized HTML code in the previous example—with whatever content the user requested by clicking a link or button.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Using Frame-Based Static Templates

Image: Content reference definition for a frame-based static template

The following screenshot is an example of content reference definition for a frame-based static template.



The Content Ref Administration page shows how the HTML exists within the context of the whole content reference. Note that the usage type is set to *Frame template.* In the previous example of a page-based template, which uses HTML tables, the usage type would have been set to *HTML template.*

Using Frame-Based Dynamic Templates

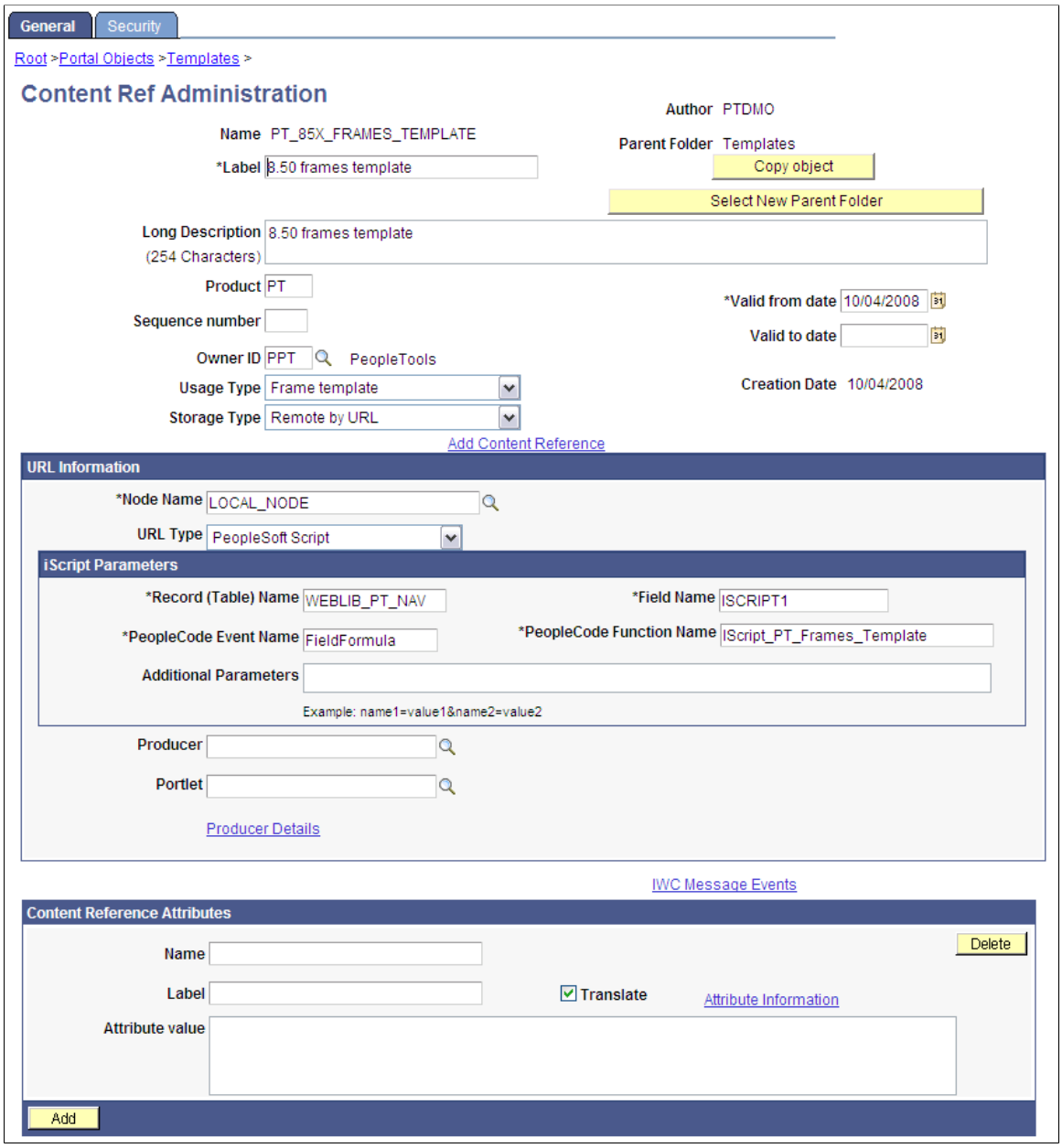
All dynamic frame-based templates are retrieved from a URL rather than from an HTML document. Most commonly, the URL for dynamic templates is an iScript URL. As with static templates, the template content reference must be defined as a template on the portal administration pages. However, instead of

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

including specific HTML content, the dynamic template references an iScript. The iScript is associated with a specified field in a specified record.

Image: Defining a frame-based dynamic template

The following screenshot illustrates a frame-based dynamic template named PT_85X_FRAMES_TEMPLATE in the Portal Objects, Templates folder:



The content reference storage type is set to *Remote by URL*. This setting is required for dynamic templates. Additionally, no HTML area displays the associated HTML code for the template, as you would expect with a static template. Instead, when the URL type for this example is set to *PeopleSoft Script,* an iScript parameters area appears on the page, in which you can define the record name, field

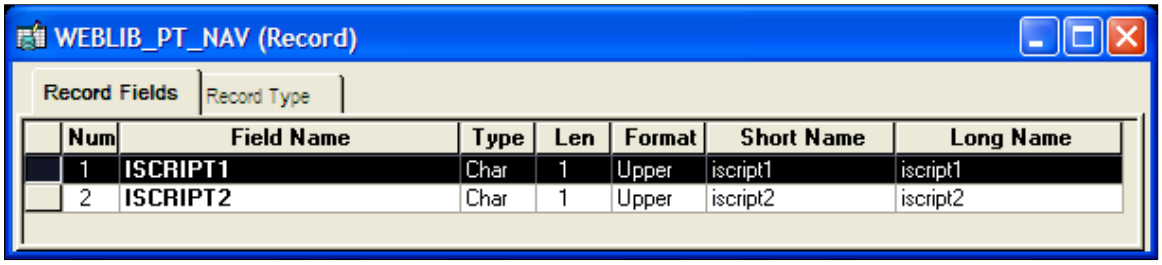
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

name, PeopleCode event name, and PeopleCode function name that specify the iScript to use for the template.

Image: Opening the WEBLIB_PT_NAV record associated with a dynamic template

In this example, an iScript that dynamically generates the template is located in the WEBLIB_PT_NAV record in the ISCRIPT1 field, as shown in PeopleSoft Application Designer in this example:

The following screenshot focuses on a iScript that dynamically generates the template.

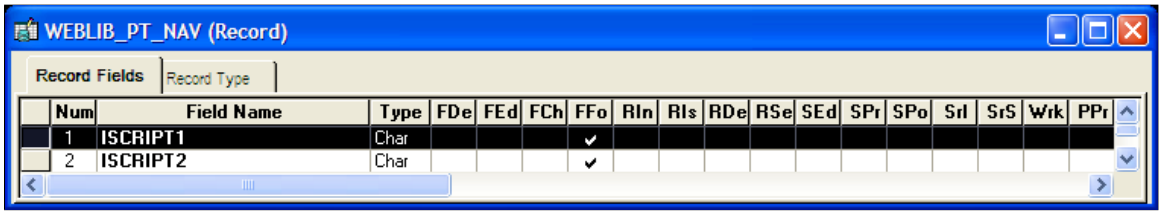


Additionally, the iScript function is associated with the FieldFormula PeopleCode event. The PeopleCode function name is IScript_PT_Frames_Template.

You can select the PeopleCode Display view to see which fields have PeopleCode programs defined for which PeopleCode events. By convention, function libraries have PeopleCode defined for the FieldFormula event.

Image: Selecting the PeopleCode Display view

The following screenshot illustrates which fields have PeopleCode programs defined for which PeopleCode events.



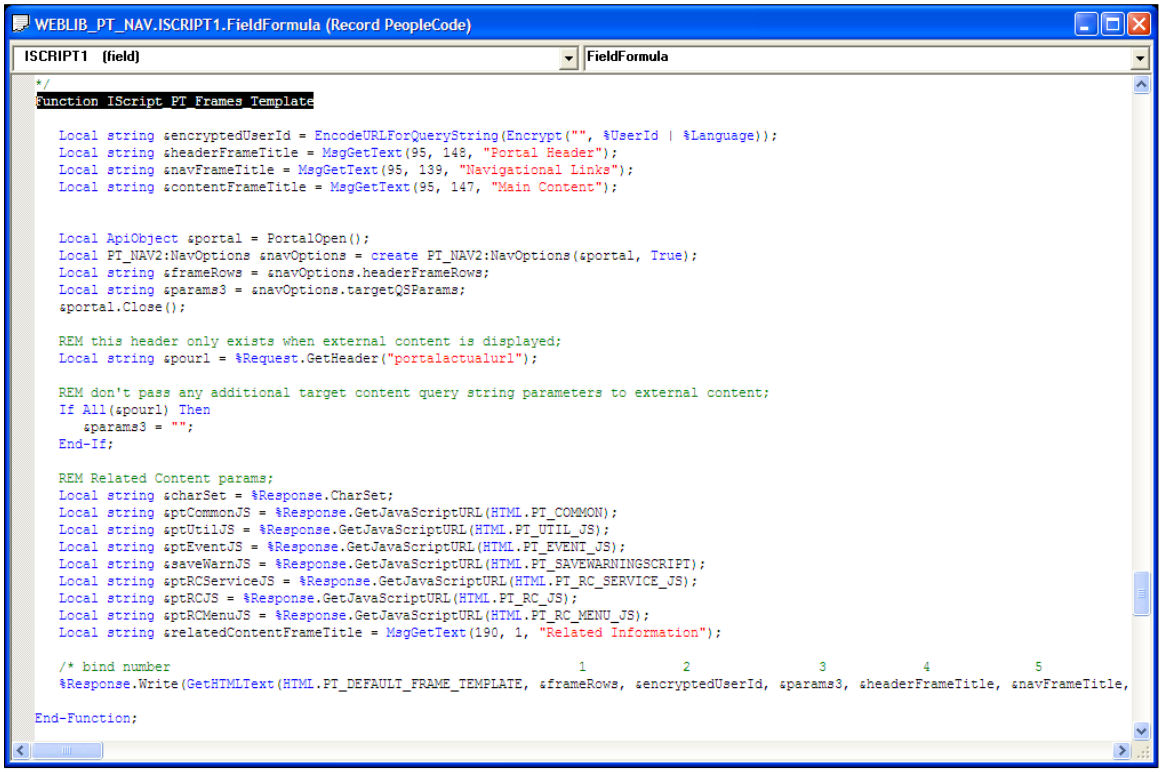
Once you open the appropriate record and field in PeopleSoft Application Designer, you can view the FieldFormula PeopleCode. The iScript is referenced by the PeopleCode function name in the template's definition. An easy way to find the exact reference is to copy the PeopleCode function name from the

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

template definition and paste it into the Find dialog box of the PeopleCode editor once you open the corresponding record and field.

Image: The iScript upon which the dynamic template is based

The following screenshot illustrates the code of the iScript that is referenced by the portal template.



The following code calls an HTML definition called PT_DEFAULT_FRAME_TEMPLATE from the HTML catalog and passes the definition bind variables:

```
%Response.Write(GetHTMLText(HTML.PT_DEFAULT_FRAME_TEMPLATE, &frameRows, &encryptedU⇒
serId, &params3, &headerFrameTitle,
     &navFrameTitle, &contentFrameTitle, &relatedContentFrameTitle, &charSet, &ptCo⇒
mmonJS, &ptUtilJS, &ptEventJS, &ptRCMenuJS,
```

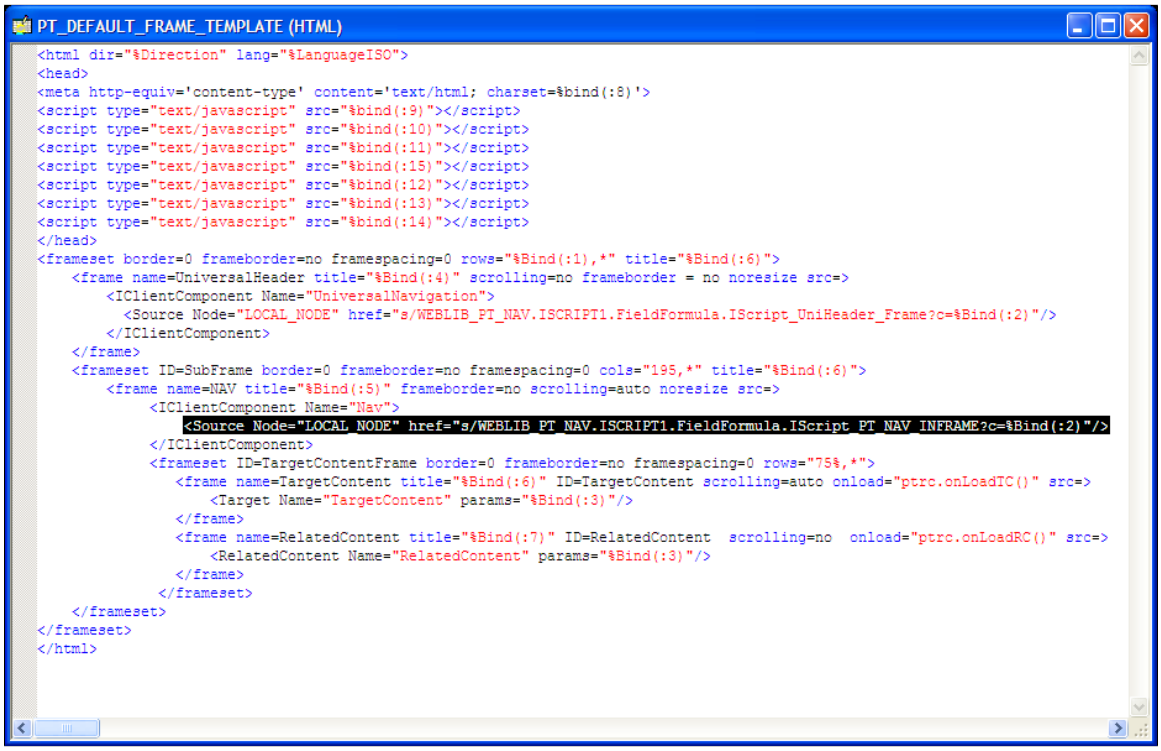
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
&ptRCServiceJS, &ptRCJS, &saveWarnJS));
```

Image: Using bind variables with a dynamic template

The resulting HTML code forms the basis for the dynamic portal template at runtime. You can open the HTML definition PT_DEFAULT_FRAME_TEMPLATE in PeopleSoft Application Designer, as shown in this example, and see where the bind variable is passed in (note that this is a frame-based template, as frame tags are present):

The following screenshot is an example of frame-based dynamic template.



Related Links

Administering Content References

Using iFrame-Based Dynamic Templates

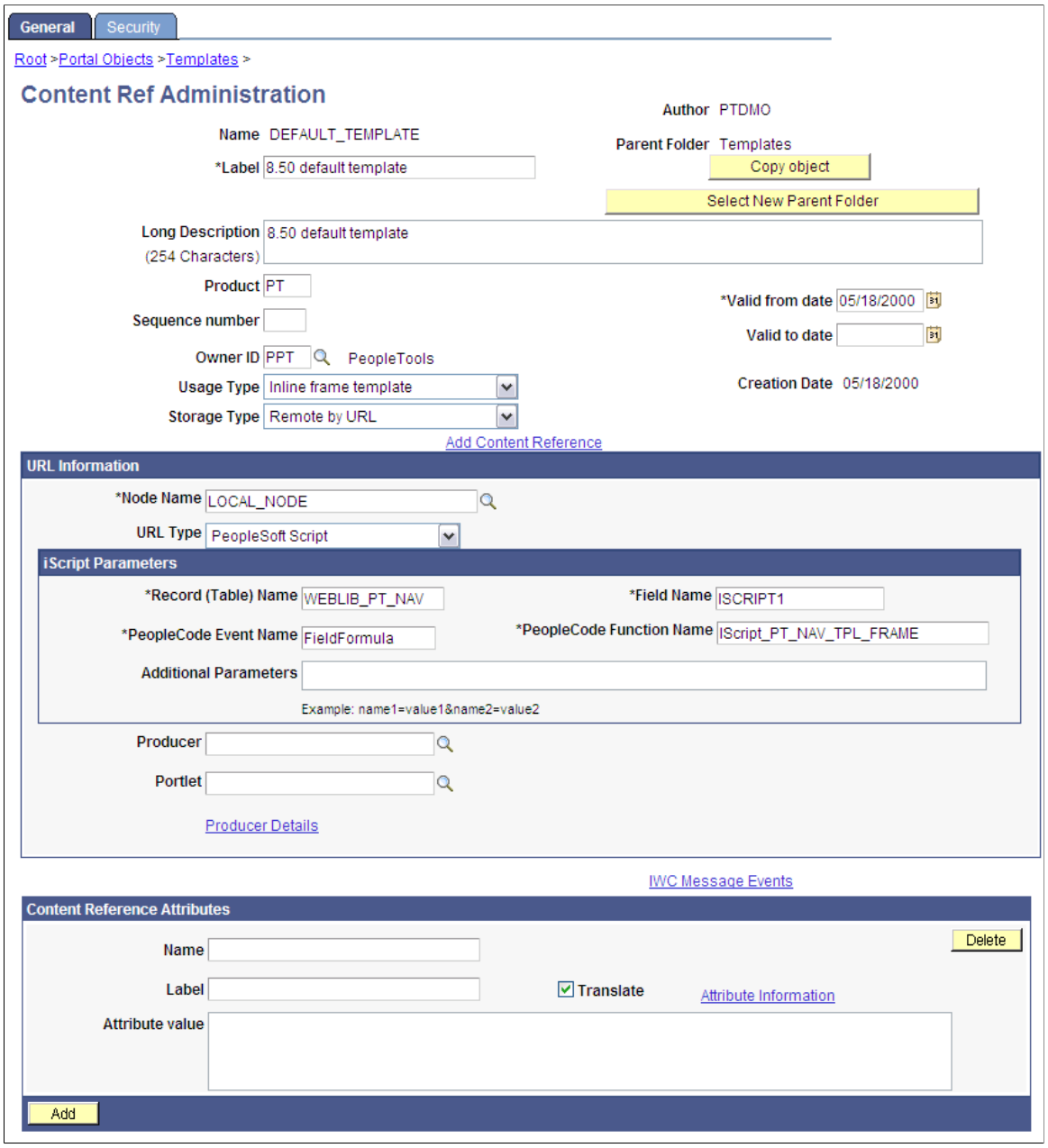
All dynamic iframe-based templates are retrieved from a URL rather than from an HTML document. Most commonly, the URL for dynamic templates is an iScript URL. As with other templates, the template content reference must be defined as a template on the portal administration pages. However, instead of including specific HTML content, the dynamic template references an iScript. The iScript is associated with a specified field in a specified record.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Important! HTML templates are deprecated as of PeopleTools 8.50. PeopleSoft applications require iframe templates to function as designed. Features that are unsupported in HTML templates experience significant or total loss of functionality; such features include:
• Modal windows used for prompts, secondary pages, and grid zoom.
• Main Menu drop-down navigation.
• Related content.
• WorkCenter pages.
• Global search header.

Image: Defining an iframe-based dynamic template on the Content Ref Administration page

The following screenshot illustrates an iframe-based dynamic template named DEFAULT_TEMPLATE in the Portal Objects, Templates folder:



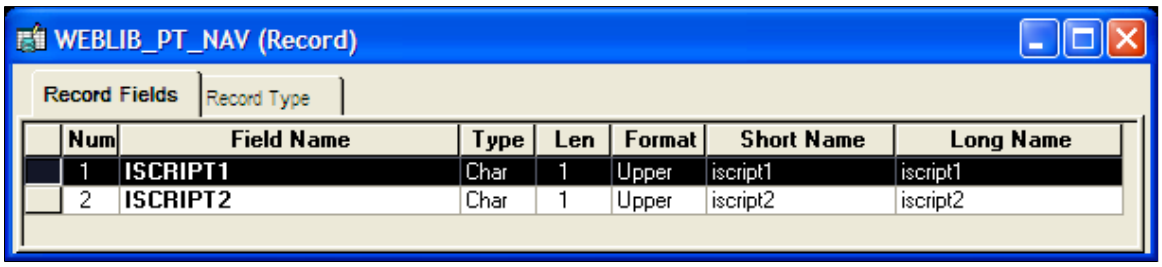
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The content reference storage type is set to *Remote by URL.* This setting is required for dynamic templates. Additionally, no HTML area displays the associated HTML code for the template, as you would expect with a static template. Instead, when the URL type for this example is set to *PeopleSoft Script,* an iScript parameters area appears on the page, in which you can define the record name, field name, PeopleCode event name, and PeopleCode function name that specify the iScript to use for the template.

The iScript that dynamically generates the template is located in the WEBLIB_PT_NAV record in the ISCRIPT1 field, as shown in this example:

Image: Opening the WEBLIB_PT_NAV record associated with a dynamic template

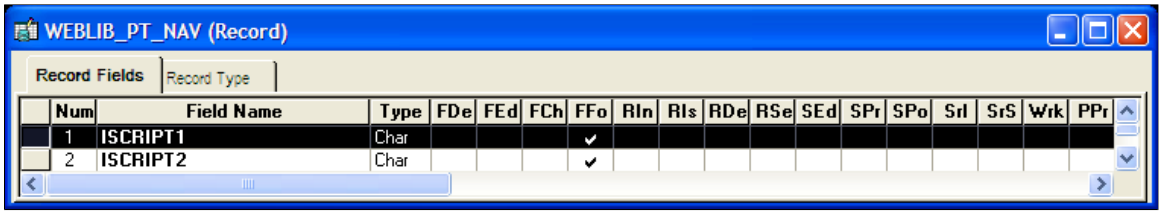
The following screenshot illustrates a WEBLIB_PT_NAV record associated with a dynamic template.



Additionally, the iScript function is associated with the FieldFormula PeopleCode event. The PeopleCode function name is IScript_PT_Frames_Template.

Image: Selecting the PeopleCode Display view

You can select the PeopleCode Display view to see which fields have PeopleCode programs defined for which PeopleCode events. By convention, function libraries have PeopleCode defined for the FieldFormula event.



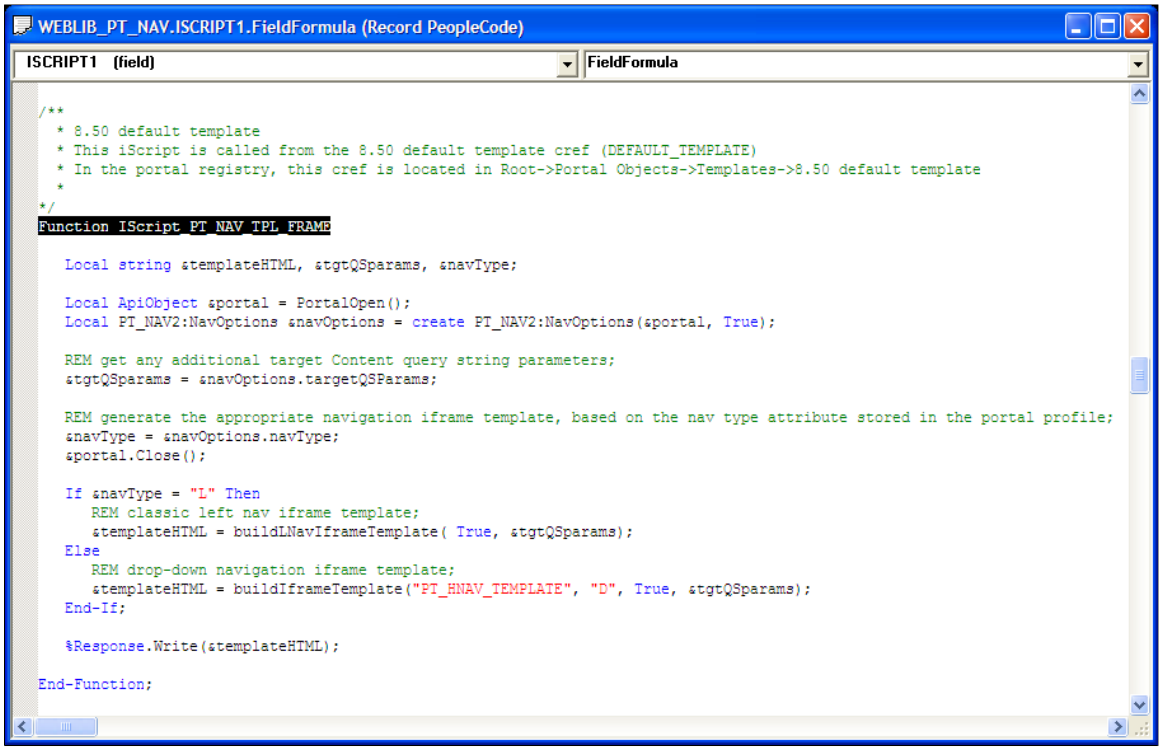
Once you open the appropriate record and field in PeopleSoft Application Designer, you can view the FieldFormula PeopleCode. The iScript is referenced by the PeopleCode function name in the template's definition. An easy way to find the exact reference is to copy the PeopleCode function name from the

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

template definition and paste it into the Find dialog box of the PeopleCode editor once you open the corresponding record and field.

Image: The iScript upon which the dynamic iframe template is based

The following screenshot illustrates the e code of the iScript that is referenced by the portal template, as shown in this example:



The following code calls the PT_IFRAME_TEMPLATE HTML definition from the HTML catalog and passes the definition bind variables:

```
Return GetHTMLText(HTML.PT_IFRAME_TEMPLATE, &charSet, &requiredToolsSS, &sysDefa⇒
ultSS, &domainJS, &expireJS, &ptCommonJS, &ptEventJS, &ptUtilJS, &ptAjaxJS, &crossD⇒
omainJS, &saveWarnJS, &iframeHdrJS, &iframeJS, &rcJS, &navHeader.HTML, &nav2JS, &pt⇒
PopupJS, &params1, &encryptedUserId, &rcMenuJS, &rcHTML1, &tgtIframeTitle, &rcServi⇒
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
ceJS, &cssOverride);
```

Image: PT_IFRAME_TEMPLATE HTML definition showing inline frames

The resulting HTML code forms the basis for the dynamic portal template at runtime. You can open the PT_IFRAME_TEMPLATE HTML definition in PeopleSoft Application Designer, as shown in this example, and see where the bind variable is passed in (note that this is an iframe-based template, as `<iframe>` tags are present):

The following screenshot illustrates an example of iframe-based template (HTML) that contains inline frames.

PT_IFRAME_TEMPLATE (HTML)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html dir="%Direction" lang="%LanguageISO">
<head>
<meta http-equiv="content-type" content="text/html; charset=%bind(:1)">
<title></title>
<link rel="stylesheet" href="%bind(:2)" type="text/css">
<link rel="stylesheet" href="%bind(:3)" type="text/css">
%bind(:4)
%bind(:5)
<script type="text/javascript" src="%bind(:6)"></script>
<script type="text/javascript" src="%bind(:7)"></script>
<script type="text/javascript" src="%bind(:8)"></script>
<script type="text/javascript" src="%bind(:9)"></script>
<script type="text/javascript" src="%bind(:10)"></script>
<script type="text/javascript" src="%bind(:11)"></script>
<script type="text/javascript" src="%bind(:17)"></script>
<script type="text/javascript" src="%bind(:12)"></script>
<script type="text/javascript" src="%bind(:23)"></script>
<script type="text/javascript" src="%bind(:20)"></script>
<script type="text/javascript" src="%bind(:14)"></script>
%bind(:24)
</head>
<body class="PSPAGE" id="ptifrmtemplate">
  <div id="ptifrmhdr">
    <pagelet name="UniversalNavigation">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_UniHeader_Frame?c=%Bind(:19)" />
    </pagelet>
  </div>
  <div id="ptifrmsbarcollexp" class="PTPAGELETHEADER">
    %bind(:15)
  </div>
  <div id="ptifrmcontent">

    <div id="ptifrmsidebar">
      <div id="ptifrmnav" class="PTPAGELET PTNAVBACKGROUND">
        <pagelet name="nav">
          <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_PT_NAV_INFRAME?c=%Bind(:19)" />
        </pagelet>
        <div id="ptnav2block"> </div>
      </div>
      <script type="text/javascript" src="%bind(:16)"></script>
    </div>

    <div id="ptifrmtarget" class="ptifrmtgtstd">
      <iframe id="ptifrmtgtframe" name="TargetContent" title="%bind(:22)" frameborder="0"
            scrolling="auto" onload="ptrc.onLoadTC()" src=><target Name="target" params="%bind(:18)"/>
      </iframe>
    </div>
    %bind(:21)
    <script type="text/javascript" src="%bind(:13)"></script>
  </div>
</body>
</html>
```

Inline frame-based, or iframe-based, templates enable you to embed a frame within an HTML document rather than as a frameset. You can use the iframe `<source>` attribute to load content into the template. PeopleSoft applications use this attribute for target content only. The header and navigation content are in the HTML document itself as you see in the preceding PT_IFRAME_TEMPLATE HTML definition:

Note the following points about the standard iframe template:

- The Usage Type is set to *Inline frame template;* this is a requirement for *all* iframe-based templates.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The iframe HTML definition supports only the <pagelet> and <target> meta-tags.

  Important! The IClientComponent meta-tag is not supported.

- Using inline frames enables you to click the navigation and see only the target content change. The entire page does not refresh. This approach allows for much more flexibility and facilitates the header and navigation working together.
- When you navigate to different content within the iframe template, the URL in the browser address bar may not be up to date.
- If you create your own iframe-based template, include the !DOCTYPE at the top of the code.

Developing Portal Templates

This section discusses:

- Template pagelets based on pages.
- Inheritance of style classes.
- Template pagelet tags.
- Considerations for non-PeopleSoft content providers.
- Considerations for frame-based templates.
- Partial path nodes.
- Pagelet names in templates.
- Considerations for forms and templates.
- Error message display.

Template Pagelets Based on Pages

Template pagelets can be provided by any URL, but usually they are based on either a page or an iScript. If you are using template pagelets based on pages, add the query string parameter *&target= PanelGroupName* at the end of the query string parameter list for the template pagelet in the template HTML. This prevents PeopleSoft Pure Internet Architecture from assuming the same default name, Main, for each page. Failure to specify the target parameter results in a JavaScript error when the portal servlet tries to load the template. If more than one form exists with the name Main on the same page, the JavaScript on that page cannot determine which Main component to use. In the template HTML, add the target parameter to the source specification of each page-based template pagelet that is not meant to be the target content.

Here is an example:

```
<tr>
<td><Pagelet Name="Related Links">
     <Source Node="LOCAL_NODE" href="c/PORTAL_ADMIN.PORTAL_RLNK_1COL.GBL?target=
        relatedlinks"/>
    </Pagelet>
</td>
</tr>
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Inheritance of Style Classes

When the portal servlet assembles portal pages based on PeopleSoft Application Designer pages, the assembled pages inherit the style classes that are defined for them in PeopleSoft Application Designer. Because you can develop a template that includes pagelets built with different style sheets in PeopleSoft Application Designer, you may have conflicting style classes in your template.

For example, if you have a style class named Big on page X and you also have a style class named Big on page Y, conflicting styles may be on the resulting page. When conflicting style classes exist, the style sheet that is associated with the page that is used for the target content always takes precedence over competing style sheets in the template.

Template Pagelet Tags

Template HTML can contain three PeopleSoft-specific tags that are used as portal servlet directives to assemble content on the page: Pagelet tags, Source tags, and Target tags.

Pagelet

Note the following attribute, element, and content information for the Pagelet tag:

Attributes	Name. Used to identify the component in a comment in the final assembled page.
Elements	Source tag.
Contents	None.

Source

Note the following attribute, element, and content information for the Source tag:

Attributes	Product tag. Used to identify which PeopleSoft Pure Internet Architecture web server to route to. The Product tag value should be the name of the registered content provider for the content.
Elements	None.
Contents	Query string to access the page or iScript implementing the component.

Contents must be escaped. Ampersands must be written as *&*. Use the IClientComponent tag to specify content that is not specific to the PeopleSoft Pure Internet Architecture. If a pagelet is implemented by a web server that is not a registered content provider, the entire URL of the pagelet can be specified in the Source tag contents, and the product attributes can be omitted.

Target

Note the following attribute, element, and content information for the Target tag.

Attributes	Name. Used to identify the component in a comment on the final assembled page.
Elements	None.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Contents None.

Considerations for Non-PeopleSoft Content Providers

To determine which template to apply to requested content, the portal servlet invokes the portal registry API to look up the content reference that is associated with the target content URL.

Warning! The information in this section applies to any content that is generated by a system for which the PeopleTools release is earlier than 8.4, or any time the portal cannot determine from the submitted URL what content reference is being requested.

The same URL can supply different content pages depending on the context. For example, with PeopleSoft application pages, the URL for submit buttons before PeopleTools 8.4 always looks like this: http://server/servlets/iclientservlet/peoplesoft8.

Because of this, the target content can specify its registered URL by using a custom response header, PortalRegisteredURL. When the target content supplies this response header, the portal servlet uses it to identify the URL that is registered in the portal, rather than using the actual URL that is used to get the content. All PeopleSoft Pure Internet Architecture technologies, such as pages, iScripts, and queries, supply this URL.

The value of this header must be the string with which the page should be registered. You must register the content with the identical string that it will return as this header.

If content does not return the header information, the portal servlet identifies the content by the URL in the request that is submitted to the portal.

You can override the value of the PortalRegisteredURL response header in a PeopleSoft Pure Internet Architecture script or page by adding the header to the response yourself, like this:

```
%Response.SetHeader("PortalRegisteredURL", &myURL);
```

You can do this to register the content with additional parameters.

Considerations for Frame-Based Templates

In a frame-based template, all relative URL references refer back to the psc servlet. To construct a URL reference to a page that is assembled by the portal servlet, use either the GenerateXXXXPortalURL or GenerateXXXXPortalRelURL PeopleCode functions.

Partial Path Nodes

URLs that are not registered but partially match a registered node have the template of the associated node applied. This helps you to manage unregistered URLs and to ensure that they achieve a consistent appearance within the portal. Most pages have the template formatting that you and portal users expect, rather than resorting to the default template when an exact node cannot be found.

Note: When the content reference isn't registered, it doesn't use the template defined for the node. Instead, it uses the default template. Therefore, unregistered URLs use the default template, not the template assigned to the node.

This table shows examples of nodes, the URI, and sample URLs that would use the template that is specified by the node:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Node	*URI*	*Sample Match*
Yahoo	http://www.yahoo.com	http://www.yahoo.com/business/webleaders/peoplesoft.html
Yahoo Sports	http://www.yahoo.com/sports	http://www.yahoo.com/sports/nba/heat/teamreport.html
Yahoo NFL	http://www.yahoo.com/sports/nfl	http://www.yahoo.com/sports/nfl/dolphins/teamreport.html

Pagelet Names in Templates

You can specify a pagelet name in the template rather than specifying a URL. This is the syntax:

```
<Pagelet Name="My Pagelet">
<Source Product="Portal"  Pagelet="MY_PAGELET" />
</Pagelet>
```

If the portal servlet sees the Pagelet attribute in the Source tag, it looks up that pagelet name in the registry and uses the associated URL there.

Considerations for Forms and Templates

When the application posts data from a form in an HTML template, post data is lost if the action of the form is in a frame template. To solve this problem, make the frame source for the target a special request to the portal to repost the form data. The HTML response to that request includes a form containing the posted field and value pairs in hidden input fields and a Script tag that submits that form. When the browser receives this request, the JavaScript immediately submits the form to the original URL, causing the original post request to occur.

To make the frame source for the target a special portal request:

1. Construct the frameset.

   The source of the target frame is the portal, with the query string cmd=framerepost&key=xxx (or something similar).

2. Store the post data on the session object as a PropertySet (name and value pairs) for later retrieval, with a key.

   The key must uniquely identify the post data. The URL to post to can be kept in the PropertySet or put on the query string.

3. When the portal receives a frame repost request, it uses the incoming key to look up the stored PropertySet, then uses PropertySet to construct the HTML response with the name and value pairs in hidden input fields.

   The form also has a Submit button. The HTML response contains an inline JavaScript (after the form) that submits the form.

See the PeopleTools installation guide for your database platform.

Combining Forms and Frames

If you include a form on a page that's been assembled by the portal servlet, and that page's target is displayed in a frame (either in a frame template or in a nontemplate frame), then the form action cannot

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

be Post. The portal servlet retrieves the page by using Get—not Post—regardless of the form action. Therefore, form data is lost if the form action was originally Post. Making the form action Get resolves this problem.

Error Message Display

When a pagelet cannot be retrieved, an error message appears.

You can disable this error message. If the error is not displayed, then the pagelet disappears from the template without a message to the user.

To control the display of this error message, use the DisplayError attribute of the IClientComponent tag:

```
<IClientComponent Name="Example" DisplayError=false>
   <Source Product="">http://www.InvalidServer/InvalidPage.html</Source>
</IClientComponent>
```

If the portal servlet finds DisplayError equal to False, it does not display the error but adds the empty pagelet comment tags. If DisplayError is equal to True, or if the attribute is not found, the error message is displayed.

This feature works best when the HTML for the template can flow around the empty content properly.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 9

Using Portal Caching Features

Understanding Portal Caching

Information on your PeopleSoft portal that changes frequently must be regularly updated, but for best efficiency, information that remains static should be cached for quick retrieval. Information that changes less frequently should be updated on a more deliberate schedule. The composition of changing and static information varies depending on the user, role, application, portal, web server, information structure, and information content involved.

Caching Configuration Options

When properly configured, caching significantly boosts portal performance. PeopleSoft portal technology provides a wide range of caching options. These options target different elements of the portal environment, use a variety of criteria for applying caching, and have varying degrees of scope. Not all caching options apply to all circumstances.

You can target portal caching to a specific pagelet, template, portal, or web server. You can also apply caching based on the content reference type or browser type. In some cases, you can specify the scope of caching to be by user, by role, or by application.

Some types of caching take place on the portal web server, and others are applied at the browser without requiring individual browser configuration. You can make various caching settings in the Web Profile Configuration component (WEB_PROFILE), on a personalization page, or in a template HTML element.

With server caching, the cached data is stored on the portal web server. When a user's browser requests a piece of information from the portal, the server transmits the currently available data. The currency of the data provided depends on whether the server has refreshed the data since the last request. This is controlled by the server-based caching criteria that you defined for that data, such as a time interval or a specified number of requests.

Note: When browser caching is enabled, you can manually refresh the browser, which reloads the entire browser HTML content from the portal server. This doesn't update the information that's cached on the server, however.

Default Homepage Caching

The default homepage is automatically cached on the server only when you select the Allow Public Access check box in the on the Web Profile Configuration - Security page. Caching the default homepage according to these rules protects the server from having to regenerate the same homepage repeatedly. This is useful for guest sign-in scenarios.

Related Links

"Cache Settings" (PeopleTools 8.54: System and Server Administration)

Configuring Portal Security

Cookie Rules

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Administering Server-Based Caching

This section provides an overview of caching scope and discusses how to:

- Implement target content caching.
- Implement pagelet caching.
- Implement metadata caching.
- Implement proxied JavaScript caching.

Understanding Caching Scope

You can define caching for server-based portal content with one of three levels of scope:

- Private scope.

  This is also known as the *user* scope. The portal maintains the cached content independently for each user. This scope is useful when the content that is displayed, such as a 401K balance, is specific to the user who's signed in.

- Role scope.

  Each user is assigned one or more roles. The portal maintains a single cache of the content for all users with a given combination of roles. This scope is useful when the content that is displayed, such as the menu structure, is specific to the role set of the user who’s signed in. If your user population has many unique role combinations, your caching memory requirements may be nearly the same as private scope caching.

- Public scope.

  This is also known as the *application* scope. The portal maintains a single cache of the content for all users who are using a given application. This scope is useful when all users who have access to this content, such as company news articles, bulletins, or messages, see the same thing. Because only one cache object is managed in memory for your entire user population, this is the least memory-intensive scoping option.

Implementing Target Content Caching

You can cache the content of an individual target by specifying several caching parameters in the HTML of each locally defined template that invokes the target. You must also enable target content caching for the portal.

Important! Target content is cached in memory on the web server. If you use this feature excessively, the cache could consume all available memory. Use target content caching only to cache content that is static or doesn't require constant updating and that is accessed frequently by many users.

Specifying Caching Parameters

You apply target content caching by inserting a Cache element inside a TargetContent element in the HTML of a locally defined template.

The Cache element has these attributes:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Attribute	*Description*
Scope	Use one of these values: • *user:* The portal maintains the cached target content independently for each user. • *role:* The portal maintains a single cache of the target content for all users in a role. • *application:* The portal maintains a single cache of the target content for all users who are using an application.
Interval	Specify the number of seconds that should pass before the portal refreshes the target content cache.

This example shows caching that is defined with the application scope and an interval of 1200 seconds:

```
<TargetContent Name="MyContent">
    <Cache Scope="application" Interval="1200">ANY</Cache>
</TargetContent>
```

Note: The Cache element must not be empty, but its content can be anything.

Enabling Target Content Caching

Any Cache elements that you insert in template HTML are ignored by the portal unless you enable target content caching globally for the portal. Select the Cache Target Content check box in the On the Web Server group box of the Web Profile Configuration - Caching page.

Related Links

Understanding Portal Templates
Applying Template Types
Developing Portal Templates
Configuring Caching

Implementing Pagelet Caching

If you're developing a pagelet and you know that its content will remain static for all users, you can specify the application or public scope. If you know that the pagelet's content will remain static for all users with the same role, you can specify the role scope. However, if the pagelet content is static only for individual users, you must specify the user or private scope. If you're not the pagelet developer, you probably won't know when the pagelet content will change, so to be sure you must specify the user scope.

Important! Pagelet content is cached in memory on the web server. If you use this feature excessively, the cache could consume all available memory. Use pagelet caching only to cache content that is static or doesn't require constant updating, and is accessed frequently by many users.

You can cache the content of an individual pagelet by specifying caching parameters in one of these locations:

- The HTML of any locally defined template that invokes the pagelet.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- A content reference attribute in the pagelet definition.

Note: If you specify caching parameters for a pagelet in both locations, the caching parameters in the template take precedence, and the caching parameters in the content reference attribute are ignored.

The presence of the caching parameters is sufficient to enable caching for the associated pagelet; no additional setting is required. To disable pagelet caching, delete the caching parameters.

Specifying Pagelet Caching in the Template

You apply pagelet caching in a template by inserting a Cache element inside a Pagelet element or IClientComponent element in a locally defined template.

The Cache element has these attributes:

Attribute	***Description***
Scope	Use one of these values: • *user:* The portal maintains the cached pagelet independently for each user. **Note:** In a high-volume environment, caching pagelets with the user scope requires a separate web server cache of each pagelet for every user, which can result in significant memory problems. Because of this, caching pagelets with the role or application scope whenever possible is strongly recommended. • *role:* The portal maintains a single cache of the pagelet for all users in a role. • *application:* The portal maintains a single cached version of the pagelet for each user. The cache manager clones a copy of the meta pagelet and stores a copy for each user. For example, if 1000 users have signed in and they all have this pagelet on their homepage, 1000 copies of the pagelet will be stored in cache. Each of these pagelet copies has the associated time to live (TTL) variable. Each time the system builds the user's homepage, it reviews the TTL to determine whether the time has come to refresh the pagelet. When a TTL time expires, that copy of the meta pagelet is invalidated and refreshed from the application server. When each user's copy expires, his or her copy will be updated with the latest version.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Attribute	*Description*
Interval	Specify the number of seconds that should pass before the cache expires and is subject to an automatic refresh. This value depends on the frequency with which the data changes, and the need for data to be 100 percent accurate. For example, a company news article rarely changes once it's been published. You might want to set the refresh period for the article pagelet to be *86400* seconds so that it's cached for a 24-hour period.

This example shows caching that is defined in an IClientComponent element with the role scope and an interval of 1800 seconds:

```
<IClientComponent Name="Header">
    <Cache Scope="role" Interval="1800">ANY</Cache>
    <Source Product="Portal">ICType=Script...</Source>
</IClientComponent
```

Note: The Cache element must not be empty, but its content can be anything.

Specifying Pagelet Caching in the Content Reference

You apply pagelet caching in a template by defining a special content reference attribute on the pagelet definition.

To implement caching for a pagelet:

1. Open the content reference administration page of the pagelet that you want to cache.

   select the folder containing the pagelet, and edit its content reference.

   a. Select PeopleTools, Portal, Structure and Content.

   b. Click the Portal Object link.

   c. Click the Pagelets link.

   d. Click the link for the pagelet folder.

   e. Click the Edit link for the pagelet.

2. Add a new content reference attribute.

3. Specify the attribute name and label.

   - In the Name field, enter *PSCACHECONTROL.*
   - In the Label field, enter *PSCache-Control.*

4. Specify the attribute value with this syntax:

   `cache_scope,max-age=`*`interval`*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Parameter	*Description*
cache_scope	Use one of these values: • *private:* The portal maintains the cached pagelet independently for each user. **Note:** In a high-volume environment, caching pagelets with the private scope requires a separate web server cache of each pagelet for each user, which can result in significant memory usage. Because of this, caching pagelets with the role or public scope wherever possible is strongly recommended. • *role:* The portal maintains a single cache of the pagelet for all users in a role. • *public:* The portal maintains a single cache of the pagelet for all users who are using an application.
interval	Specify the number of minutes that should pass before the cache expires and is subject to an automatic refresh.

For example, consider a PeopleSoft Interaction Hub environment in which users can personalize their company news pagelet. The articles don't change frequently, and users don't need them to be current; therefore, you might specify private caching with a three-hour (180 minute) expiration:

```
private,max-age=180
```

5. Save the content reference.

   The next time a user signs in and displays the pagelet, the caching that you specified takes effect.

Managing the Caching of Pagelets

You can enable users to personalize their homepage pagelets. When a user personalizes a pagelet, the result should be visible only to that user, and therefore should be cached only for that user. To achieve this goal, as soon as the user attempts to personalize the pagelet, regardless of the caching scope that you originally specified, the portal automatically changes the caching scope to apply on a private basis and maintains the cached pagelet independently for that user.

Important! This automatic scope change applies only to the user who is personalizing the pagelet and to the portal on which the pagelet is personalized. On other portals, and for other users, the originally defined caching scope for that pagelet still applies. If the personalizations aren't stored as portal-specific data, then the personalizations for a pagelet that was originally defined with a wider caching scope can become visible to others who are accessing the same site through another portal.
Therefore, when developing personalizable pagelets, you need to ensure that stored pagelet personalization metadata is keyed by both user ID and portal name so that you can populate the pagelet independently for each portal. This prevents user personalizations on one portal from being cached and visible to a larger audience on another portal.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Disabling Manual Pagelet Refresh

When you implement pagelet caching, a refresh button appears by default in the pagelet header. You can prevent users from refreshing the pagelet manually by hiding the refresh button, which you do by selecting a check box on the pagelet's content reference administration page.

Related Links

Administering Content References
Managing Pagelet Attributes
Understanding Caching Scope

Implementing Metadata Caching

Metadata caching is governed by portal-wide properties that control caching for these portal metadata objects as a group:

- The portal registry.
- Node definitions.
- Content references.
- Locally defined templates.

Portal metadata is likely to change far less frequently than pagelets or target content, so caching it on the server is an easy way to improve portal performance. Caching improves system performance by decreasing service requests from the web server to the application server.

Note: Metadata caching always applies with the application scope—the portal maintains a single metadata cache for all users who are using an application.

You implement metadata caching for a portal by setting these properties in the On the Web Server group box of the Web Profile Configuration - Caching page:

Cache Portal Objects

Select to enable metadata caching.

Metadata caching is enabled by default. You must restart the web server or wait for the stale interval to expire when you change:

- The default portal for a database.
- The default local node for a database.
- The remote node to local or a local node to remote.
- Templates that are associated with a content reference, node, or portal.
- HTML within a template.

Note: When you add or delete a content reference or folder, this change shows up in the menu structure immediately because it's rendered within the application server, not the portal servlet.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Cache Stale Interval

Specify the number of seconds that should pass before the portal refreshes the metadata cache.

The default value of this property is *86400* (24 hours). With this setting, if you change the default local node in the database, the change doesn't take effect until the following day.

Cache Purge All Hit Count

Set to the maximum total number of HTTP requests that the web server should receive for objects in the metadata caches of all portals. When the web server receives this number of hits, it purges all metadata caches, forcing the portals to refresh them.

This setting enables you to manage application server memory consumption. If the web server is consuming too much memory, you can lower this value. This causes the web server to reclaim the memory that is used for cached portal objects more often. The cost of this is increased CPU usage on both the web and the application server when the web server deletes the cached objects and then retrieves them again from the application server.

The default value of this property is *1000.* Set to *–1* to disable the feature.

Note: The timing properties—the cache stale interval and the cache purge all hit count—compete for effect. The first timing property to reach the specified interval or number of hits takes precedence. When the metadata cache on a portal is refreshed, the cache stale interval and cache purge all hit count counters for all portals on the same web server are reset.

Related Links

Configuring Caching

Implementing Proxied JavaScript Caching

In addition to target content, pagelets, and metadata, you can cache proxied JavaScripts on the server. This is recommended for production systems; you might want to disable this caching during development.

You enable caching of proxied JavaScripts for a portal by selecting the Cache Proxied JavaScripts check box in the On the Web Server group box of the Web Profile Configuration - Caching page.

Related Links

Configuring Caching

Administering Browser-Based Caching

This section provides an overview of browser-based caching and discusses how to:

- Implement homepage caching.
- Control navigation pagelet caching.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Implement PeopleSoft page caching.

Understanding Browser-Based Caching

A modern web browser can cache the information that it receives from a web server in several ways. Its preference settings determine how much storage space to use for caching, as well as the criteria by which it should refresh its cache.

How the Portal Controls Browser-Based Caching

If the browser is accessing a straight HTML page, it caches the page as a whole. If it's accessing a frameset, it caches the individual frames in the frameset so that each frame's content can be refreshed independently.

Rather than depend on users to enable and set the timing criteria for caching, PeopleSoft portal technology takes advantage of a feature of HTTP. The portal web server can specify a caching interval in the HTTP response header of the HTML page or frame content that it delivers. The browser applies this value to determine when to stop using its cached copy of the data and request the most recent version from the server.

Note: Even when the browser requests a fresh copy of an HTML page or frame content, the portal web server controls whether the content it sends is cached data or fresh data based on its own caching settings.

When to Disable Browser-Based Caching

The caching system increases system response performance in typical production environments in which users access the portal web site by using computers that have been assigned specifically to them or computers that they own. However, some environments don't conform to this scenario.

You should consider disabling browser-based caching in these environments:

- Any environment in which multiple users access the same computer.

  These include kiosk-type environments, as well as corporate hoteling environments, in which multiple users access the same machine. Users might access the same machine within minutes of each other before the browser cache for the previous user times out. Security is enforced in this situation, but the new user may be confused by an inability to access certain items.

- A development or testing environment.

  When content references are created or changed for a portal during development or testing, the new data might not appear to be immediately available due to caching.

- Any environment in which you are using web server-based navigation caching.

Related Links

Configuring Caching

Implementing Homepage Caching

In a PeopleSoft portal, a homepage is a combination of portal content elements that are assembled by the portal and delivered as a single HTML page, which is cached on each user’s browser.

Based on the configuration that you specify, the portal enables or disables homepage caching by using the homepage's HTTP header. If caching is enabled, the header also conveys the time interval before the web server is accessed again to get a fresh homepage. The browser does not access the web server

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

after it initially receives the homepage until the specified interval elapses. You can also specify whether a particular browser model and version should use or ignore the caching information in the header.

In any case, if a user clicks the browser's Refresh button, the homepage is accessed from the web server again, overwriting the homepage that is cached on the browser and restarting the interval counter. If any of the content is cached on the web server, the browser is refreshed from the server cache.

Enabling Homepage Caching

You enable homepage caching for a site by selecting the Cache Homepage check box in the On the Browser group box of the Web Profile Configuration - Caching page. When this check box is selected (the default value), two related page elements become editable:

- Homepage Stale Interval.
- Browsers.

When the Cache Homepage check box is selected, the following occurs:

- All non-component-based pagelets will be cached on the browser by default unless the pagelet has a PORTAL_NOBROWSERCACHE content reference attribute set to *Y*.
- Component-based pagelets will not be cached on the browser by default unless the pagelet has a PORTAL_NOBROWSERCACHE content reference attribute set to *N*.

Important! Do not enable homepage caching for any component-based pagelet that uses interactive processing mode.

When this check box is deselected, any PORTAL_NOBROWSERCACHE content reference attributes are ignored.

Specifying the Homepage Stale Interval

Specify the homepage caching interval for a site by entering a value in seconds in the Homepage Stale Interval field in the On the Browser group box of the Web Profile Configuration - Caching page. When the specified interval elapses, the browser sends a new request for the homepage to the portal web server the next time a user accesses the homepage. The default value for this field is *1200*.

Disabling Caching for Selected Browsers

Because browsers don't all process HTML in exactly the same way, you might need to disable homepage caching for selected browser versions. This can be useful if you have one or two supported browsers and want to disable caching for nonstandard browsers that could pose an administration problem.

To disable homepage caching for a browser version, add a row to the Browsers grid of the Web Profile Configuration - Caching page, enter the browser's user agent ID, and select the Cache Home Page check box for that row. Homepage caching is enabled for all browser versions except those with this check box deselected, regardless of whether they're listed in the grid.

Note: As a convenience, several well-known browser types are included as sample rows in the configuration. All of the sample rows have homepage caching enabled by default.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Links

Configuring Caching

Controlling Navigation Pagelet Caching

When you use the portal menu to access target content other than your homepage, the portal delivers that content within a frameset. The portal header, menu, and target content regions are independent and can be managed separately by the browser. The menu region contains the portal's navigation pagelet, which can be cached on the browser with the user scope.

Without caching, every time the user clicks a menu folder, the browser requests a new copy of the navigation pagelet from the portal web server, which can significantly affect performance. With caching, the browser saves all of the menu states that are generated by user activity and can immediately restore them on demand.

Users manage navigation pagelet caching by using the Time page held in cache personalization option to specify a caching interval.

To specify a caching interval:

1. In PeopleSoft Pure Internet Architecture, select My Personalizations.
2. On the Personalizations page, click the Personalize Option button for general options.
3. Enter an override value in minutes for the Time page held in cache option and click OK.

   The default value of this option is *900* minutes. To disable navigation pagelet caching, specify *0* minutes. The maximum value can be *525600* minutes, which is one year.

Note: For the new setting to take effect, you must delete the browser cache.
With caching in effect, any changes in the menu structure while the user is signed in to the portal aren't available until the user signs out and signs in again, or until the caching interval elapses.

Implementing PeopleSoft Page Caching

PeopleSoft pages are application pages that appear in the target content region of the portal frameset.

Being able to control PeopleSoft page caching on the browser is useful for situations in which PeopleSoft applications are deployed to kiosk workstations where multiple users access the applications. Disabling caching means that users can't click the browser's Back button to view another person's transaction or to view any other sensitive data.

Enabling PeopleSoft Page Caching

You enable PeopleSoft page caching for a site by selecting the Cache Generated HTML check box in the On the Browser group box of the Web Profile Configuration - Caching page. PeopleSoft page caching is enabled by default.

With PeopleSoft page caching disabled, when users click the Back button they receive a *data missing* message in Netscape Navigator and a *page expired* message in Microsoft Internet Explorer.

Note: The side effect of turning off caching completely is degraded performance. For each new page, the system calls the database. However, PeopleTools offers a compromise related to browser caching in the form of navigation pagelet caching, which doesn't expose any sensitive information.

See Controlling Navigation Pagelet Caching.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Supported States

When the browser caches a PeopleSoft page, it stores the state of that page, including any unsaved changes. With the page state in the cache, users clicking the browser's Back button see the cached page in the state that it was in when they last viewed it. The browser caches as many page states as its own settings for storing temporary data allow.

The portal also maintains its own cache of page states for each browser window. You can specify how many of these states the portal should consider valid and support for further interaction. This setting, combined with the number of states that the browser maintains, determines whether users can continue to work with previously accessed page states.

For example, if you specify that the portal should support two states, and the user clicks the Back button three times, any activity that requires refreshing the displayed page from the web server (such as field validation) fails, producing a page unavailable message in the browser.

You specify how many states the portal should support by entering a number in the Number of States Supported field in the On the Browser group box of the Web Profile Configuration - Caching page. The default value of this field is *5* states.

Related Links

Configuring Portal Security
Configuring Caching

Administering Web Server-Based Navigation Caching

This section provides an overview and discusses how to implement web server-based navigation caching.

Understanding Web Server-Based Navigation Caching

Web server-based navigation caching considerably boosts performance by automatically storing cached data on the web server when users select items from the navigation menu. This form of caching saves unnecessary calls to the application server.

The portal maintains a single cache for all users with a given combination of roles. If your user population has limited role combinations, your caching memory requirement is significantly reduced.

See Understanding Caching Scope.

Implementing Web Server-Based Navigation Caching

Implement web server-based navigation caching by selecting the Cache Menu check box in the On the Web Server group box of the Web Profile Configuration - Caching page.

The cached data is stored in the following folder and cannot be moved:

\\PIA_HOME\webserv\peoplesoft\applications\peoplesoft\PORTAL\psftcache

You must delete the cache periodically to keep the folder at a manageable size.

To prevent conflicts, disable browser-based caching and homepage caching.

Disabling Browser-Based Caching

To disable browser-based caching:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

1. Select PeopleTools, Personalization, Personalization Options.
2. Select *PeopleTools (PPTL)*.
3. Remove the *METAXP* personalization definition.
4. Click Save.

Disabling Homepage Caching

To disable homepage caching:

1. Select PeopleTools, Web Profile, Web Profile Configuration.
2. Select the appropriate web profile.
3. Access the Browsing grid on the Caching page.
4. Deselect the Cache Homepage check box for the appropriate browser.
5. Click Save.

Related Links

Configuring Caching

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 10

Configuring the Portal Environment

Understanding the Authentication Domain

The portal *authentication domain* is the domain in which the portal is running and across which the single signon authentication token is valid. It's specified as a web server property and is used extensively throughout the PeopleSoft Pure Internet Architecture and portal runtime systems. An authentication domain is expressed as a string that completes the domain portion of a URL, for example, .peoplesoft.com.

Note: The leading period is required. The correct string is, for example, .peoplesoft.com, and not peoplesoft.com.

The authentication domain supports the following functionality:

- Cross-frame JavaScript updates between the PeopleSoft Pure Internet Architecture and the portal.

  Failure to set the authentication domain correctly for the portal and PeopleSoft Pure Internet Architecture applications causes JavaScript security errors to appear in the browser when PeopleSoft Pure Internet Architecture pages are accessed through a portal frame-based template. (The default template through which all PeopleSoft Pure Internet Architecture pages are displayed is frame-based.) The authentication domain must be set for both the portal web server and other PeopleSoft content web servers.

- PeopleCode global variable sharing between components on the homepage and components within a frame.

  Failure to set the authentication domain correctly for the portal and PeopleSoft Pure Internet Architecture applications that use different web servers causes a new, incompatible session to be created on the PeopleSoft Pure Internet Architecture web server when the user accesses a PeopleSoft Pure Internet Architecture component through a frame-based template.

- Single signon among PeopleSoft applications.

  Failure to specify the authentication domain correctly prevents the PeopleSoft authentication cookie from being passed to the target PeopleSoft application and forces the target system to reauthenticate the user.

- Cookie sharing between the portal and third-party web applications.

  If cookies need to be shared between web applications, then each web application must be accessed over a common domain name.

  To share cookies, specify the authentication domain as the Cookies Passed to Server (forwarding domain) property in the portal's web profile. You specify this property on the web Profile Configuration - Cookie Rules page.

  See Configuring Cookie Rules.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Base-Level and Extended Authentication Domains

You can define the portal authentication domain as a *base-level* authentication domain and as an *extended* authentication domain.

You define the base-level authentication domain during the PeopleSoft Pure Internet Architecture setup. This domain is stored as part of your web server configuration. It enables PeopleCode global variable sharing, which is required for initial access to the portal. The portal uses the base-level domain if you don't define an extended authentication domain.

Important! You must specify a base-level authentication domain during the setup of every PeopleSoft application with which your portal interacts. The authentication domain must be identical and be stored on the web server of each application.

See the *PeopleTools 8.53 Installation Guides* at My Oracle Support.

You can define an optional extended authentication domain in your portal web profile. An extended authentication domain overrides, but must be compatible with, the base-level authentication domain. For example, if you entered *.customer.com* during the PeopleSoft Pure Internet Architecture setup, only values such as *.enterprise.customer.com* and *.individual.customer.com* would be valid.

Note: If you defined a base-level or extended authentication domain, you must use it in all URLs that you specify in your portal. For example, if your authentication domain is .mydomain.com, then instead of using the URL http://mymachine:8080/pshome/signon.html, you must use the URL http://mymachine.mydomain.com:8080/pshome/signon.html.

You specify the extended authentication domain on General tab of the Web Profile Configuration page.

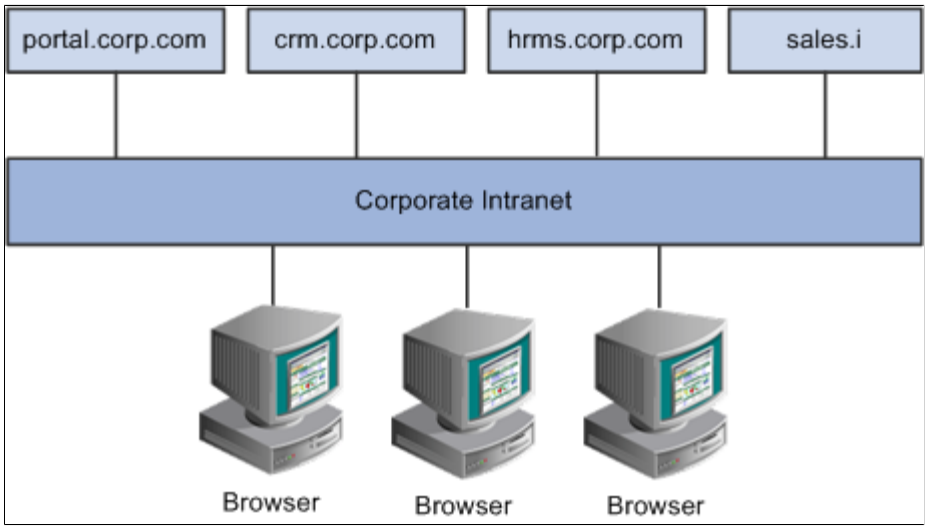
See Configuring General Portal Properties.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

An Example of Multiple Applications on a Portal

Image: Example of the portal system and several PeopleSoft applications within the same authentication domain

In the following example, the CRM and HRMS web profiles need to be defined with Domain Name Server (DNS) names that include the same authentication domain as the DNS name of the portal web server. They also each need the Authentication Domain property in their web profiles set to this value.



Web servers that don't have the same server domain as the portal (such as sales.i) can still be used to serve content to the portal. However, cookies set by the portal are not forwarded to these servers. The sales.i server in the example can provide pages and applications to the portal, but it cannot host a PeopleSoft application that supports single signon functionality with the portal.

Related Links

"Understanding WebLogic" (PeopleTools 8.54: System and Server Administration)

"Understanding WebSphere Application Server within Your PeopleSoft Implementation" (PeopleTools 8.54: System and Server Administration)

Configuring Web Profiles

This section provides an overview of web profiles and discusses how to:

- Configure general portal properties.
- Configure portal security.
- Configure virtual addressing.
- Configure cookie rules.
- Configure caching.
- Configure trace and debug options.
- Configure appearance and character.
- Configure custom properties.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- View web profile history.

Important! The PeopleSoft portal configuration environment has moved to a PeopleSoft Pure Internet Architecture interface, the web profile. Web profiles are discussed in this topic. A complete listing that maps PeopleTools release 8.43 and earlier portal settings to current web profile settings is documented in an appendix.

Related Links

Understanding Changes in Portal Configuration Settings

Understanding Web Profiles

A web profile is a specific configuration of portal-related properties that control or affect the functionality of your PeopleTools portal. These predefined web profiles provide several configurations for use with your PeopleTools portal:

- *DEV*

  This web profile provides basic portal functionality for development, including trace and debug settings that are appropriate for development.

  Important! Never run your production system with the DEV web profile. This profile is for development environments and is not tuned for performance.

- *KIOSK*

  This web profile uses the same settings as the PROD web profile, except that public user access is enabled for the Guest user, and all options for storing caching or persistent cookies on the browser are disabled.

- *PROD*

  This web profile uses the settings that are most commonly needed in a production environment that authenticates users.

- *TEST*

  This web profile uses the same settings as the DEV web profile, except that fewer trace and debug properties are enabled.

Additional web profiles might be included as part of PeopleSoft Interaction Hub or an application portal pack.

You can modify the property settings of any web profile via the Web Profile Configuration component, which includes pages for these types of properties:

- General
- Portal security
- Virtual addressing
- Cookie rules
- Caching

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Trace and debug options
- Appearance and character
- Custom properties.

Specifying an Initial Web Profile

As you perform the PeopleSoft Pure Internet Architecture setup procedure for your system, you're prompted to specify which web profile to apply to your portal. You're also prompted for an existing user ID and password. This information is saved in encrypted form which the portal servlet can use to gain secure access to the web profile.

The default web profile is PROD.

The default web profile user ID and password are both PTWEBSERVER.

The PTWEBSERVER account provides the portal servlet with minimal security access, sufficient only to launch the portal environment, but without access to any pages or other PeopleSoft Pure Internet Architecture objects. This account uses the PTPT1500 permission list, which is set to never time out, and provides necessary access 24 hours a day, seven days a week.

Note: Your PeopleSoft application might be delivered with the account that you specify here locked. You must unlock it before you can access and configure the web profile, or you will receive an error message about incorrect site configuration on the sign-in page.
You can unlock the web profile account by deselecting the Account Locked Out check box on the User Profile - General page. You can alternatively issue the following SQL command against your database (this example uses the PTWEBSERVER account):

```
update PSOPRDEFN set ACCTLOCK=0 where OPRID='PTWEBSERVER'
```

See "Specifying User Profile Attributes" (PeopleTools 8.54: Security Administration).

You can specify any of the delivered profiles, or you can enter a different profile name. In that case, the portal is set up to use a profile by that name. If a profile by the name that you specify doesn't exist in the database, internal default settings are used until you sign in to the portal and create a profile with that name. The portal then automatically configures itself according to those settings.

As long as no profile exists by the name that you specified during the PeopleSoft Pure Internet Architecture setup, the internal default settings remain in effect, which is indicated on the site sign-in page. When the site is in this mode, every browser request triggers an attempt to load the named profile. Therefore, you shouldn't use the site for extended periods like this.

Once you complete the PeopleSoft Pure Internet Architecture setup, you can then use the Web Profile Configuration component to modify the properties to reflect your settings.

Note: The default internal settings are not the same as the DEV profile. Don't modify any of the delivered profiles so that you always have access to unmodified reference versions. You can make a copy of any profile and modify it, or you can define a new profile instead.

Copying and Deleting Web Profiles

To copy a web profile:

1. Select PeopleTools, Web Profile, Copy Web Profile.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. Enter or select a web profile name.
3. On the Web Profile Save As page, enter a new name for the web profile in the To edit box.

To delete a web profile:

1. Select PeopleTools, Web Profile, Delete Web Profile.
2. Enter or select a web profile name.
3. On the Delete Web Profile page, click Delete Web Profile.

Changing the Web Profile After the PeopleSoft Pure Internet Architecture Setup

After the PeopleSoft Pure Internet Architecture setup, to select a different web profile, you must edit the text file that stores this information, called configuration.properties. Each PeopleSoft portal that you set up has its own copy of configuration.properties, located in this directory:

PIA_HOME\webserv*web_server*\applications\peoplesoft\PORTAL\WEB-INF\psftdocs*sitename*

The configuration.properties file includes these properties:

- WebProfile

  Change the value of this property to the name of the web profile that you want to apply to the portal, for example, WebProfile=MYPROFILE.

- WebUserId and WebPassword

  Change these values to specify a different user ID and password for the portal servlet to use to access the web profile. You must encrypt the password by using PSCipher.

  To encrypt the user ID or password:

  1. Run the PSCipher.bat file:

     See "Using the PSCipher Utility" (PeopleTools 8.54: Security Administration).

     Note: On a UNIX machine, change the PSCipher.sh script file permissions so that you can run the program.

  2. Copy the encrypted string and paste it into the configuration.properties file replacing the existing value, for example:

     WebPassword=VOBN5KcQZMg=

     Important! Make sure that the entire encrypted string, (including all symbol characters), is copied.

After you save the configuration.properties file, restart your web server and the new profile takes effect.

Alternatively, you can use PSADMIN to specify the web profile that is used. You must restart your web server for the new profile to take effect. For more information about using PSADMIN, see "Administering a PIA Site" (PeopleTools 8.54: System and Server Administration)

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Links

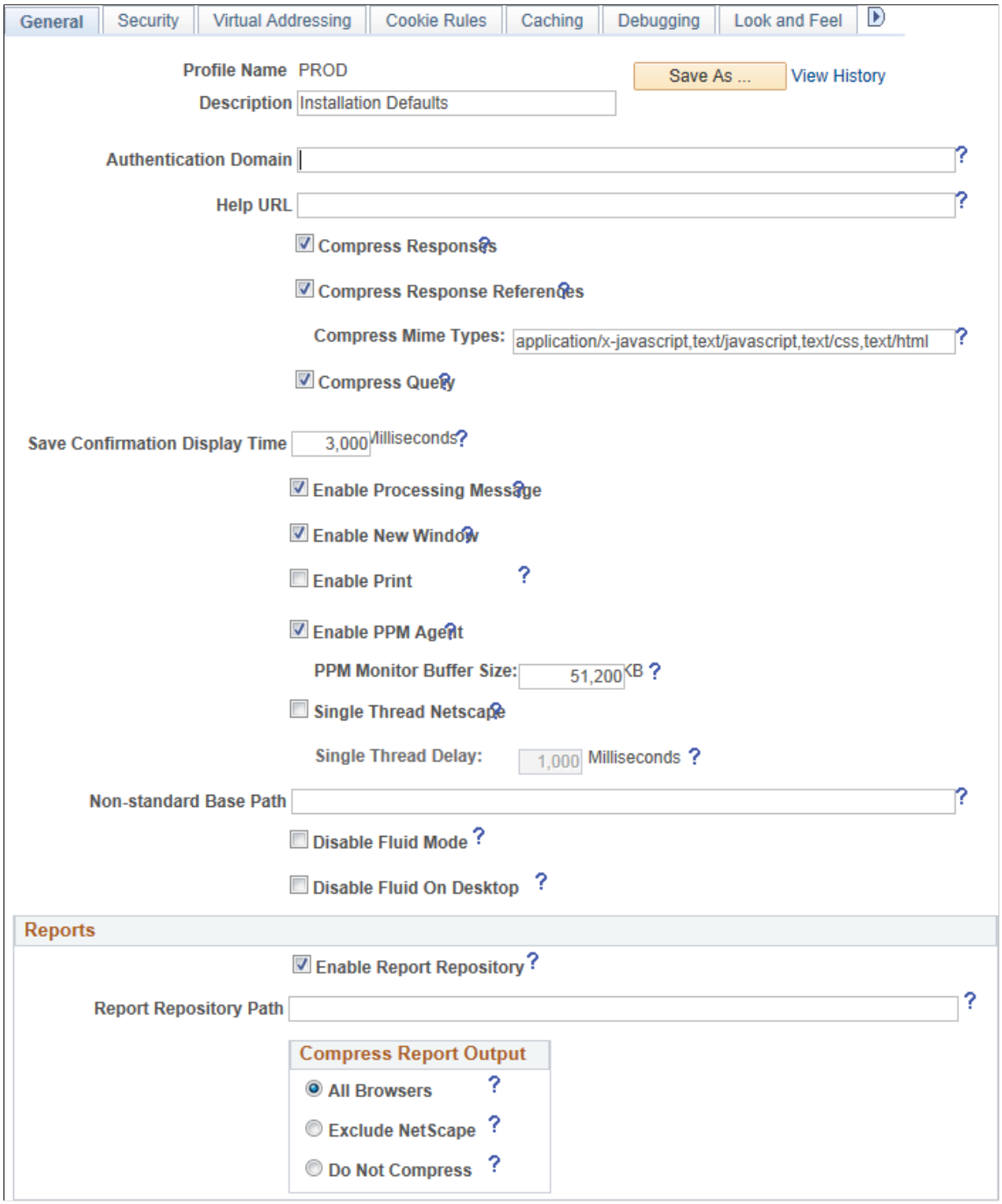
"Using the integrationGateway.properties File" (PeopleTools 8.54: Integration Broker Administration)

Configuring General Portal Properties

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration).

Image: Web Profile Configuration - General page

This example illustrates the fields and controls on the Web Profile Configuration - General page. You can find definitions for the fields and controls later on this page.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Save As

Click to save a copy of the current web profile under a new name. On the Save Web Profile As page, enter a new profile name and click OK.

Use this button to modify one of the profiles that is delivered with your PeopleSoft application.

View History

Click to access the Web Profile History search page in a new browser window.

Authentication Domain

Enter the name of the extended authentication domain in which the portal is running, starting with a leading period. This value overrides, but must be compatible with, the base-level authentication domain. For example, if you entered *.customer.com* during the PeopleSoft Pure Internet Architecture setup, only values such as *.enterprise.customer.com* and *.individual.customer.com* are valid.

An authentication domain is required for a variety of portal functions. For example, if a cookie is shared on web servers srv1.oracle.com and srv2.oracle.com, you must specify an authentication domain of *.oracle.com.*

This field requires a value if you specified a default addressing server name on the Web Profile Configuration - Virtual Addressing page. You must qualify that server name with this domain name.

The default value of the authentication domain is the one that you specified during the PeopleSoft Pure Internet Architecture setup. That value is stored as the web server session cookie domain and kept in an internal web server file.

For Oracle WebLogic, that file is weblogic.xml. The preferred method for changing that value is to rerun the website setup.

For WebSphere, that file is deployment.xml. This is located under PIA_HOME\webserv\<profile-name>\config\cells\<profile-name>NodeCell\applications\<application.name>.ear\deployments\<application-name>\deployment.xml

---

Note: The value that you enter in this field is automatically converted to lowercase.

Note: You must enter your authentication domain to enable all functionality associated with PeopleSoft Related Content Framework.

Help URL

Specify the URL that points to an installation of PeopleSoft online help. When users click the Help link, the appropriate context-sensitive PeopleSoft documentation is displayed. To remove the Help link from all application pages, omit this value.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

You can point to either a local installation of PeopleSoft online help or to an installation on Oracle's hosted documentation website:

- To point to a local installation of PeopleSoft online help, construct the URL similar to the following:

  ```
  http://webserver:port/help_folder/help.html?
  ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%
  ```

  For example:

  ```
  http://myserver.example.com:8080/help-PB/help.
  html?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%
  ```

- To point to an installation on Oracle's hosted documentation website, construct a URL similar to the following:

  ```
  http://www.oracle.com/pls/topic/lookup?id=
  %CONTEXT_ID%&ctx=library_cd
  ```

  For example:

  ```
  http://www.oracle.com/pls/topic/lookup?id=
  %CONTEXT_ID%&ctx=hcm92pbr6
  ```

  Note: When using Oracle's hosted documentation website, you can refer to more than one library—for example, to the library for PeopleSoft FSCM in addition to PeopleTools. For more information on current library codes, see PeopleSoft Online Library and refer to the attachment in the section titled "Setting Up Context-Sensitive Help."

Note: This setting applies only to the Help link displayed on PeopleSoft pages accessed through a browser. Use the PeopleTools Options page to configure F1 help for Application Designer and other Windows client applications. See "PeopleTools Options" (PeopleTools 8.54: System and Server Administration) for more information.

Compress Responses

Select to enable compression in the communication between the web server and the browser. Gzip and Compress protocols are supported.

This check box is selected by default.

Compress Response References

Select to enable compression of cache files that are delivered from the web server to the user's browser. Only cache files with the Multipurpose Internet Mail Extensions (MIME) types that are specified in the Compress MIME Types text box are compressed. Gzip and Compress protocols are supported.

This check box is deselected by default.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Compress MIME Types

Specify the MIME types of the cache files to be compressed as a comma-separated string. This field is available only if you selected the Compress Response References check box.

The default value is *application/x-javascript,text/javascript,text/css,text/html.*

Compress Query

This property applies to browser requests in which the content type section of the URL is /q/, indicating a query. Select to enable compression of query responses to the browser.

This check box is selected by default.

Save Confirmation Display Time

Specify in milliseconds how long the save_confirm image should appear for the user if the save confirm personalization option is enabled.

The default value is *3000* milliseconds (3 seconds).

Enable Processing Message

Select to enable processing notification while the system processes a request.

This check box is selected by default.

Enable New Window

Enables or disables the New Window link in the pagebar site wide:

- Select to enable the New Window link in the pagebar for components that have the pagebar enabled and that have the New Window option selection.
- Deselect to disable the New Window link site wide.

This check box is selected by default.

Enable Print

Enables or disables the Print link in the pagebar site wide:

- Select to enable the Print link link in the pagebar for components that have the pagebar enabled.
- Deselect to disable the Print link site wide.

This check box is deselected by default.

Enable PPM Agent (enable PeopleSoft Performance Monitor agent)

Select to activate the PeopleSoft Performance Monitor agent for this portal. The performance monitoring system has *agents* and *monitors.* An agent captures units of data (PMUs), and a monitor views and analyzes agent flows. This check box controls whether the agent is active on sites that use this profile.

This check box is selected by default.

PPM Monitor Buffer Size (PeopleSoft Performance Monitor buffer size)

Specify the maximum size, in kilobytes (KB), of the Monitor Server buffer. Data is discarded once this limit is reached. This field is available only when Enable PPM Agent is selected.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The default value is *51200* KB (50 megabytes).

Single Thread Netscape

Select to indicate that requests from a Netscape browser should be single-threaded to prevent crashes.

Older versions of Netscape browsers had problems with multithreading responses and would crash. Select this check box if your users are using older versions of Netscape browsers and are experiencing browser crashes.

This check box is deselected by default. When selected, the Single Thread Delay option appears.

Single Thread Delay

Specify a delay, in milliseconds, for single-threaded Netscape requests. This field is available only when Single Thread Netscape is selected.

The default value is *1000* milliseconds (one second).

Non-standard Base Path

In the rare circumstance that you receive an error message that the base physical path has not been set, you must enter the location of the directory that contains the signon.html file for your application.

This field is blank by default.

Disable Fluid Mode
Disable Fluid On Desktop

These options specify fluid mode operation. See "Configuring Fluid Mode Web Profile Settings" (PeopleTools 8.54: Fluid User Interface Developer's Guide) for more information.

See Viewing Web Profile History, Understanding the Authentication Domain, Configuring Caching, Configuring Virtual Addressing, and "Working with Performance Monitor Web Profile Properties" (PeopleTools 8.54: Performance Monitor), "Setting Internet Properties" (PeopleTools 8.54: Application Designer Developer's Guide), and "Using the Pagebar" (PeopleTools 8.54: Applications User's Guide).

See *PeopleTools Installation* for your database platform.

Reports

Enable Report Repository

Select to enable users to view the report repository. When this check box is deselected, no report files can be viewed by users.

This check box is selected by default.

Report Repository Path

(Optional) Specify a drive and directory path to indicate where the site that is served by this web profile should look for generated reports.

This value overrides the location that is specified during the PeopleSoft Pure Internet Architecture setup; you can leave this field blank to use the original value.

Note: Changing the report repository path prevents you from viewing existing reports unless the old subdirectory structure is moved to the new location. Be careful when editing this field.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: (For Microsoft Windows only) If the report repository resides on a different machine from the web server, specify the report repository path in UNC format instead of mapped drive format. For example, if the report repository is on MachineX in folderY, the Report Repository Path should be specified as:
//MachineX/folderY/psreports
or
\\\\MachineX\\folderY\\psreports
If MachineX/folderY is mapped to the Y: drive, do not specify the Report Repository Path as Y:\psreports.

Compress Report Output

Specify how reports should be compressed:

All Browsers: Reports are always compressed. This is the default value.

Exclude Netscape: Reports are compressed, except for users who are using Netscape Navigator.

Do Not Compress: Reports are never compressed.

Related Links

"Request Class Properties" (PeopleTools 8.54: PeopleCode API Reference)

"Understanding WebLogic" (PeopleTools 8.54: System and Server Administration)

"Understanding WebSphere Application Server within Your PeopleSoft Implementation" (PeopleTools 8.54: System and Server Administration)

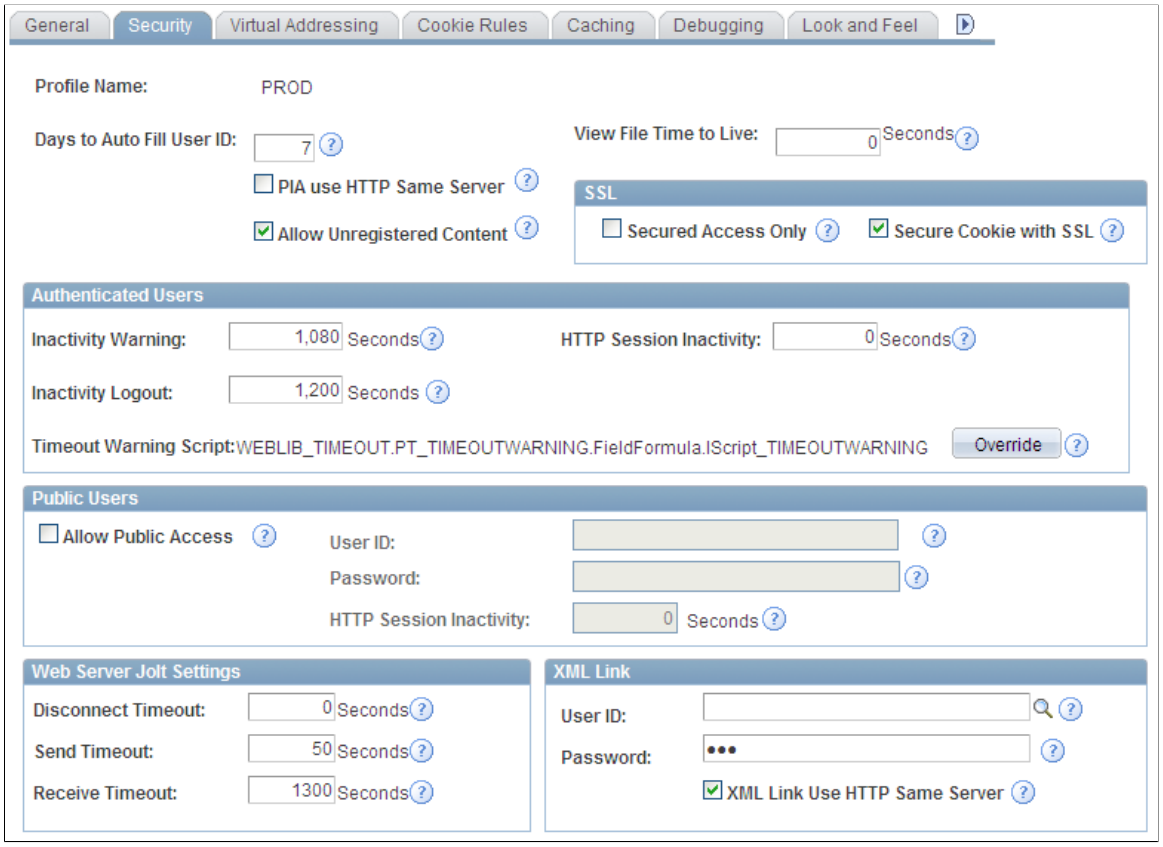
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Portal Security

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Security tab.)

Image: Web Profile Configuration - Security page

This example illustrates the fields and controls on the Web Profile Configuration - Security page. You can find definitions for the fields and controls later on this page.



Days to Autofill User ID

If this field is populated, the system caches the user ID and automatically inserts it on the sign-in page. This a convenience for users. This feature is implemented through the use of a stored cookie on the browser.

The default value is *7* days. Specify a value of *0* to disable this feature. Use *0* in a public area or kiosk situation, or if your security policy doesn't allow cookies stored on the browser.

View File Time to Live

Specify, in seconds, how long the portal should wait after sending a file attachment to a user's browser before removing that file from the web server storage.

If this value is *0,* then the amount of time that the file remains on the web server is always greater than or equal to one minute. The specific amount of time beyond one minute depends on the size of the file.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

If the value is *greater than 0,* then the amount of time that the file remains on the web server depends, approximately, only on the value of that property in seconds.

The default value is *0.*

PIA use HTTP Same Server (PeopleSoft Pure Internet Architecture use HTTP same server)

Select to specify that the portal should use the HTTP protocol instead of HTTPS for requests that are issued by the portal for content that is hosted on the same server as the portal servlet.

Setting this property is necessary when the portal web server is behind an SSL accelerator or when SSL is terminated on a device in front of the portal web server, such as a reverse proxy server, or between different sites that are on the same web server.

You can also use this property to improve the performance of homepage pagelets that are provided by the PeopleSoft Pure Internet Architecture servlet that run on the same web server as the portal, and from which the web server receives SSL requests. That is, SSL has not been terminated by a device in front of the web server.

You must also specify the default addressing protocol and port on the Web Profile Configuration - Virtual Addressing page.

See Using SSL Accelerators.

Allow Unregistered Content

Select to instruct the portal to serve both registered and unregistered content.

The portal generally allows any external content links to be wrapped with the portal header and navigation frames. Deselecting this option prevents explicitly registered content references from being displayed in the portal.

This check box is selected by default.

Note: If Allow Unregistered Content is not selected, the portal won't wrap a content link if it is not registered in the portal registry or if it isn't coming from a registered node. The portal checks whether the content link is from a registered node by checking whether the content link starts with the URI specified in the Content URI Text or the Portal URI Text in the Node Definition page, on the Portal tab. For security purposes, the portal checks only against the Content URI Text and Portal URI Text if it contains text that is longer than 12 characters.

SSL (Secure Sockets Layer)

Secured Access Only

Select to enforce SSL if the entire website requires the SSL protocol. This prevents users from using non-SSL protocols to access any link within this website or application.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

If only some pages require SSL access, deselect this check box (the default setting).

Secure Cookie with SSL

Select to prevent the single sign on token from traveling over an insecure network. If you select this check box and the scheme of the current request is HTTPS (an SSL server), the system sets the secure attribute of the single sign on cookie (PS_TOKEN) to True.

PeopleSoft single sign on functionality also applies at the web server level. For example, suppose that you have two web servers: server X and server Y. Web server X is an SSL site, and web server Y is not. In these situations, many sites want server Y to trust the authentication token, PS_TOKEN, issued by server X. This requires that you set the PS_TOKEN to be secure.

If the PS_TOKEN is not secure, when a user signs in through server Y, the browser sends PS_TOKEN to server Y over the unencrypted, non-SSL link. This is typical behavior for browsers when dealing with cookies that aren't secure. Potentially, in this situation, a hacker could obtain this token from the clear network and use it to sign in to the SSL-secure server X.

Another important use of this property relates specifically to the PeopleSoft portal. When the portal uses a proxy server to relay content with an HTML template, it should forward only PS_TOKEN cookies that are marked secure over SSL connections.

Note: By selecting this check box, you effectively disable single sign on functionality with any non-SSL servers.

If, at your site, you want users to sign in to an HTTPS server, and then want to use single signon functionality with HTTP servers, deselect this check box to enable single signon functionality between HTTPS and HTTP servers.

Note: Before you deselect this check box, make sure that you are aware of all the security implications, such as the security of the HTTPS server being compromised.

This check box is selected by default.

Authenticated Users

When single signon has been set up, the portal web server and any content provider web servers should have the same values for the following fields:

- Inactivity Warning
- Inactivity Logout

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- HTTP Session Inactivity

Inactivity Warning

Specify the number of seconds that the portal should wait before warning users that the browser session will expire. Users can continue with the current session by clicking the OK button in the warning message, which resets the inactivity timer. If a user doesn't respond within the inactivity logout interval, the session ends and the expired connection page appears.

The default value is *1080* seconds (18 minutes).

HTTP Session Inactivity

Specify the number of seconds of inactivity after which the HTTP session times out for authenticated users. When the interval passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session. From the user experience, the user may be put in a search page or home page because the previous session state is lost at this point. This property is unaffected by any permission list setting.

When this property isn't specified, the HTTP session time-out interval is the same value as the authenticated user inactivity logout property that is specified on this page.

This property is similar to the public users HTTP Session Inactivity property that is specified on this page.

Use this property with caution. Don't use this property if you don't understand its usage.

Inactivity Logout

Specify the value in seconds of the inactivity time-out interval that applies to PeopleSoft applications to which a user is signed in. When the interval passes with no user activity, the user's browser displays the page that is specified by the Expire Page - Page field on the Web Profile Configuration - Look and Feel page. The web server applies this value in JavaScript.

Note: The inactivity logout can be overridden by a permission list inactivity time-out setting.

The default value is *1200* seconds.

Note: Depending on the application implementation, authenticated users might also experience an HTTP session inactivity time-out, which by default is the same as the inactivity logout value that you specify here.
However, you can override this behavior by using the HTTP Session Inactivity property on this page. This property independently controls the HTTP session inactivity time-out. The HTTP Session Inactivity property is unaffected by any permission list setting.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Timeout Warning Script

Displays the name of the time-out warning script that is currently in effect. The default value is *WEBLIB_TIMEOUT.PT_TIMEOUTWARNING.FieldFormula.IScript_TIMEOUTWARNING.*

Override

Click to change the time-out warning script. The Override Timeout Warning Script page appears, with these fields:

- Record (Table) Name: The default value is *WEBLIB_TIMEOUT.*
- Field Name: The default value is *PT_TIMEOUTWARNING.*
- PeopleCode Event Name: The default value is *FieldFormula.*
- PeopleCode Function Name: The default value is *IScript_TIMEOUTWARNING.*

The values of these fields are concatenated to produce the value in the Timeout Warning Script field.

See "Defining Permissions" (PeopleTools 8.54: Security Administration).

Public Users

Public users are PeopleSoft application users who are not required to go through a PeopleSoft sign in page. As a result, public users are neither identified nor authenticated by their own user ID and password. Public users also are not forwarded to the sign in page after any period of inactivity.

Because they access sites without credentials, public users should be limited to informational websites where sensitive data is not accessible.

You should enable public user access under these conditions:

- When you provide users with direct links to nonsecure pages.

  Note: If a user clicks a direct link that accesses a page for which the public user ID is not authorized, the system displays the sign-in page prompting the user to supply her PeopleSoft credentials.

- When you use an external authentication method.

  Important! If you are using an external authentication method in conjunction with the PeopleCode *SwitchUser* function, then the Inactivity Warning and Inactivity Logout values specified for authenticated users are restored.

  See "SwitchUser" (PeopleTools 8.54: PeopleCode Language Reference).

- When you want crawlers to search your PeopleSoft site and need to provide access to the portal.

Allow Public Access

Select so that the system bypasses the user sign-in page when users click a direct link to a page. The system uses the values that you specify in the User ID and Password fields for authentication. This check box is deselected by default.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

When you select this check box, the system enables the User ID, Password, and HTTP Session Inactivity fields.

User ID

Enter the account that the system uses to authenticate public users. This field is enabled when you select the Allow Public Access check box.

Note: This is *not* the same as the WebUserId property in the configuration.properties file, which you specify during the PeopleSoft Pure Internet Architecture installation for accessing the web profile.

Password

Enter the password for the account that you entered in the User ID field. This field is enabled when you select the Allow Public Access check box

---

Note: This is *not* the same as the WebPassword property in the configuration.properties file, which you specified during the PeopleSoft Pure Internet Architecture installation for accessing the web profile.

HTTP Session Inactivity

Specify the value in seconds of the inactivity period that applies to public users.

When the period passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session.

This field is enabled when you select the Allow Public Access check box.

Note: If you are using an external authentication method in conjunction with the SetAuthenticationResult() function in Signon PeopleCode or the SwitchUser() function in Non-Signon PeopleCode, you are logged out if you are inactive.

The default value is *1200* seconds.

See "SwitchUser" (PeopleTools 8.54: PeopleCode Language Reference).

Web Server Jolt Settings

Disconnect Timeout

Specify the amount of time to wait while disconnecting the Oracle Jolt connection. A value of *0* means no limit.

The default value is *0* seconds.

Send Timeout

Specify the maximum number of seconds that the servlet allows for a request to be sent to the application server. This setting does not indicate a maximum amount of time for the service to

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

finish; it indicates only the maximum amount of time to send the request to the application server.

The default value is *50* seconds.

Receive Timeout

Specify the maximum number of seconds that the servlet should wait for a response from the application server.

Make sure that the value of this field is greater than your application server service time-outs, such as the Service Timeout setting for PSAPPSRV that appears in the PSAPPSRV.CFG configuration file on the application server.

The default value is *600* seconds.

XML Link

User ID

Enter the user account that the system uses to authenticate users for XML Link technology.

Password

Enter the password for the account name that you entered in the User ID field.

XML Link Use HTTP Same Server

Select to indicate that the portal should use the HTTP protocol instead of HTTPS for requests that are issued by the xmllink servlet for content that is hosted on the same server as the xmllink servlet.

This check box is selected by default.

Related Links

Improving Same-Server Performance Under SSL

"Understanding WebLogic" (PeopleTools 8.54: System and Server Administration)

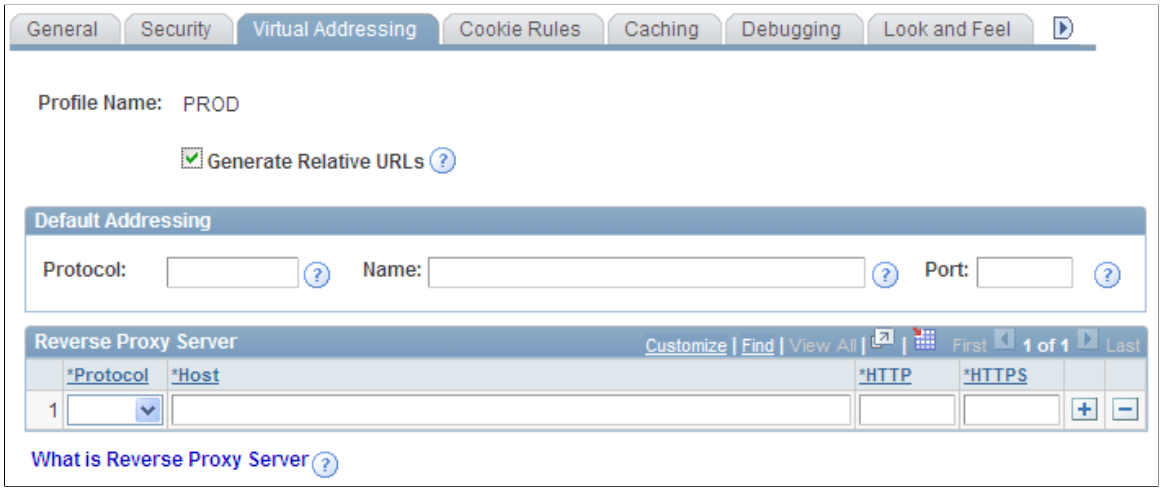
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Virtual Addressing

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Virtual Addressing tab.)

Image: Web Profile Configuration - Virtual Addressing page

This example illustrates the fields and controls on the Web Profile Configuration - Virtual Addressing page. You can find definitions for the fields and controls later on this page.



Generate Relative URLs

Select to generate relative URLs. This setting is for use with proxy server implementations.

This check box is selected by default.

Default Addressing

Protocol

Specify the default protocol—*HTTP* or *HTTPS*—to override the protocol that is used by the PeopleSoft Pure Internet Architecture and the portal to construct URL references. When this property is omitted, the protocol of the incoming request to the PeopleSoft Pure Internet Architecture servlet is used.

You must set this property when the browser makes an SSL connection and SSL is terminated before the PeopleSoft Pure Internet Architecture servlet, using a device such as an SSL accelerator or a reverse proxy server, or when you select the PIA use HTTP Same Server property on the Web Profile Configuration - Security page. In this situation, the protocol of the request to the PeopleSoft Pure Internet Architecture servlet is HTTP, but the PeopleSoft Pure Internet Architecture must generate URL references with an HTTPS protocol.

Name

Enter a default server name to override the server name that is used by the PeopleSoft Pure Internet Architecture and the portal to construct URL references. When this property is omitted, the server name of the incoming request to the PeopleSoft Pure Internet Architecture servlet is used to construct URL references.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

You must set this property when the server on which the PeopleSoft Pure Internet Architecture servlet resides is different from the server to which the browser is connected. This can occur when a reverse proxy server or load balancer is configured in front of the web server. In these situations, the PeopleSoft Pure Internet Architecture must generate URL references with the server that is used by the browser for its initial connection.

Note: If you specified an authentication domain on the Web Profile Configuration - General page, it must be part of the server name that you specify here. The value that you enter in this field is automatically converted to lowercase.

Port

Enter a default port to override the port that is used by the PeopleSoft Pure Internet Architecture and the portal to construct URL references. When this property is omitted, the port of the incoming request to the PeopleSoft Pure Internet Architecture servlet is used to construct links on a PeopleSoft Pure Internet Architecture or portal page.

You must set this property when the port through which the PeopleSoft Pure Internet Architecture servlet is accessed is different from the port to which the browser is connected. This can occur when an SSL accelerator is used or when a reverse proxy server is configured in front of the web server, and the reverse proxy server or load balancer listens on a different port from the web server, or when you select the PeopleSoft Pure Internet Architecture use HTTP Same Server property on the Web Profile Configuration - Security page. In these situations, the PeopleSoft Pure Internet Architecture must generate URL references with the port that is used by the browser for its initial connection.

Reverse Proxy Server List

Use this grid to specify the reverse proxy servers through which the portal can expect to retrieve content. External content that is retrieved from these sources and contains relative references is rewritten by the portal to contain relative references instead of absolute references to preserve reverse proxy server requirements.

For each reverse proxy server on the list, you specify:

- The protocol to use.
- The server's host or machine name.
- The server's HTTP port number.
- The server's HTTPS port number.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Cookie Rules

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Cookie Rules tab.)

Image: Web Profile Configuration - Cookie Rules page

This example illustrates the fields and controls on the Web Profile Configuration - Cookie Rules page. You can find definitions for the fields and controls later on this page.

This page defines rules that determine how the portal passes cookies to servers in the same domain.

Oracle WebLogic and IBM WebSphere web servers use browser cookies (containing the server path and domain) to establish session identity. These cookies have a default name that's used to retrieve the cookie on each request to the web server. In an environment in which multiple web servers are in use with the portal, you need to define unique session cookie names between web servers to prevent one cookie from overwriting another cookie of the same name set by a different web server. The path or domain of the cookie must be different to prevent overwriting.

Note: Use the first three fields to specify patterns to match, where the asterisk (*) is treated as a wildcard, matching zero or more characters.
The portal knows its own domain (from the authentication domain), and the first default rule instructs it to pass portal servlet cookies to any server in the same domain, over both secure and insecure connections. All other cookies can continue to be passed to any server on any domain.

Cookie Pattern

Specify the name pattern for cookies to which this rule applies. For example, to specify all cookies whose names start with *HR_* , enter the value *HR_**.

Cookies Passed to Server

Specify the name pattern for the servers to which the specified cookies should be passed. For example, to specify all servers in the domain .hr.peoplesoft.com, enter the value **.hr.peoplesoft.com*. You can use a substitution parameter %AuthTokenDomain to represent the authentication domain currently in effect.

Cookies Not Passed to Server

Specify the name pattern for the servers to which the specified cookies should not be passed. You can use a substitution parameter %AuthTokenDomain to represent the authentication domain currently in effect.

Note: This field takes precedence over the Cookies Passed to Server field.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Delete Cookie on Logout

Select to indicate that the cookie should be deleted when the user signs out of the portal.

Note: This check box is deselected for the first default rule (matching all cookies), but any cookies that are generated by PeopleSoft with, for example, the PSJSESSIONID prefix, are deleted when the user signs out. If you're running multiple PeopleSoft portals and want cookies to be maintained after the user signs out, update this page to specify which cookies to preserve or delete.

Warning! Oracle recommends that you use extreme caution when setting all cookies to be deleted on sign-out. Cookie pattern "*" means all cookies. Some cookies must be kept. For example, the Arrow Point cookie (ARPT) is used by some load balancers to ensure sticky http(s) sessions.

Proxied

Select to indicate that the cookie should be proxied in which the cookie is retrieved through an HTML template. The cookie isn't set directly into the browser without proxying. This setting is useful for applications where a particular cookie name is not unique across the authentication domain.

Secure

Select to indicate that the cookie should be passed only over a secure connection.

This table shows how cookie names are set in WebSphere for the Portal and pspc (PeopleSoft Portlet Container) web modules:

WAR Name	***WebSphere Session Cookie Name***
Portal	%MACHINE_NAME%-%WEB_SERVER_HTTP_PORT%-PORTAL-WASPSJSESSIONID
pspc	%MACHINE_NAME%-%WEB_SERVER_HTTP_PORT%-PORTLET-WASPSJSESSIONID

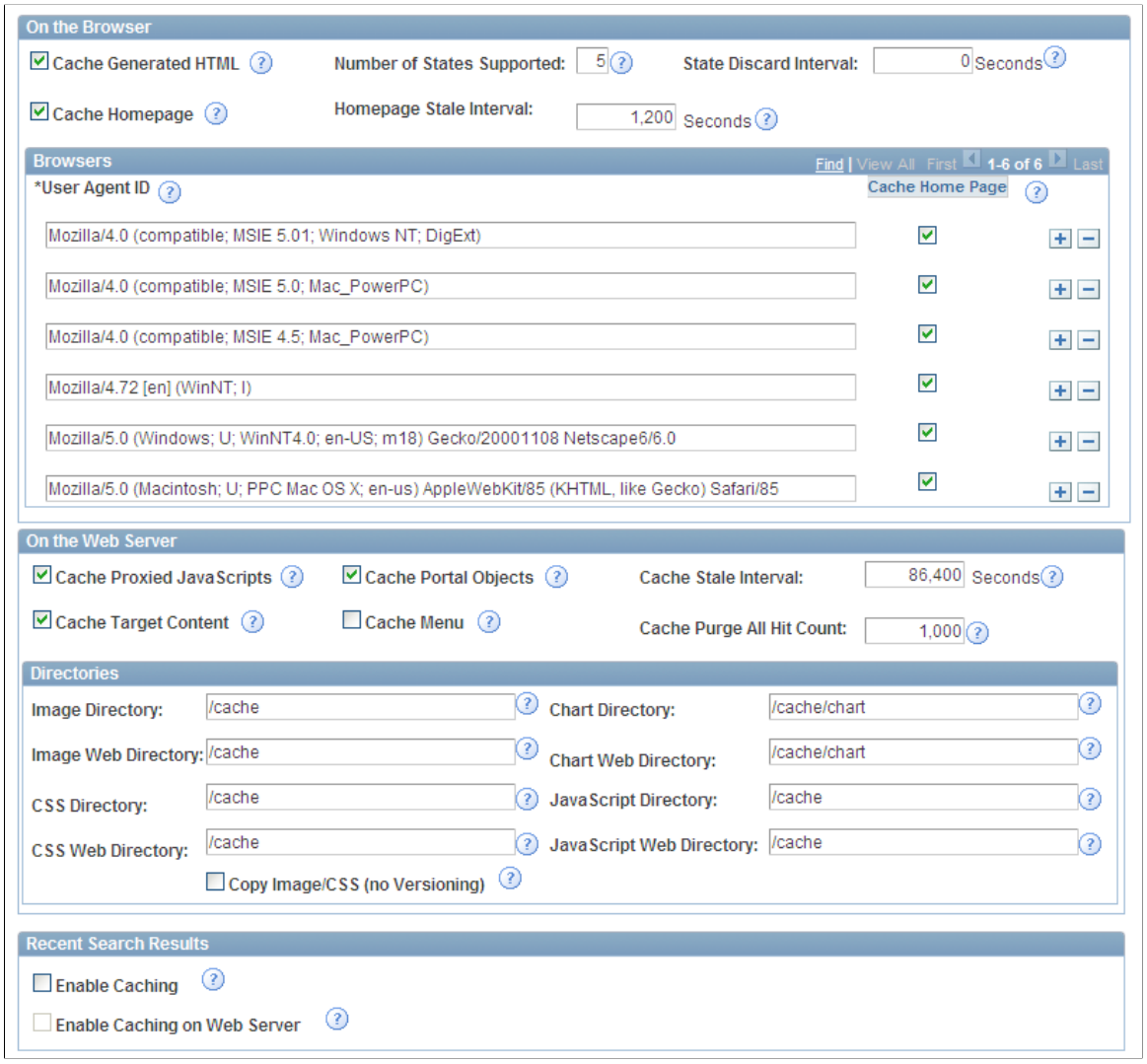
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Caching

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Caching tab.)

Image: Web Profile Configuration - Caching page

This example illustrates the fields and controls on the Web Profile Configuration - Caching page. You can find definitions for the fields and controls later on this page.



On the Browser

Cache Generated HTML

Select to indicate that PeopleSoft application pages should be cached by the user's browser. For security reasons, deselect this check box in kiosk environments.

This check box is selected by default.

Note: When you use menu navigation, the back button works even if this option is deselected because menu navigation does not generate the No-Cache directive.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Number of States Supported

Specify how many browser states the portal should support when the user accesses PeopleSoft application pages.

Note: If you have applications that make numerous server requests, you may want to increase this value. This increases the virtual machine's memory requirements, so be prepared to allocate more memory accordingly.

The default value is *5* states.

State Discard Interval

When a user signs out (either explicitly or because of a session timeout), the system normally clears all application states for the user's session. This property enables clearing of application states for individual windows instead. If your users have a long session time-out and many browser windows open, use this property to force the portal web server to release cached states for windows that a user has closed.

Every time a user clicks the New Window link on a portal page, or when transfer PeopleCode opens a new window, the portal web server creates and maintains a state cache for that new browser window. Even if the user closes the window, the web server continues to maintain the cache during the interval that is specified by the Inactivity Logout field on the Web Profile Configuration - Security page. If users open many application browser windows simultaneously and then close most of them while still continuing to work actively in one or two remaining windows, you may be using more web server memory than is required and unnecessarily degrading its performance.

You can improve performance by using this property to specify an independent time-out interval that applies to individual browser windows, which forces the portal web server to release cached states for windows that users have closed.

When a user clicks the New Window link, and each time user activity in a window produces a request for data from the web server, the portal applies a current timestamp to its corresponding state cache. The next time the user clicks the New Window link, the portal examines the timestamp for each state cache. For any timestamp that's older than the status block time-out, the portal web server discards the associated window state cache on the assumption that the window must have been closed.

Note: The portal performs this comparison only when the user clicks the New Window link.

Enter a value that represents the number of seconds for the inactivity time-out per window. This time-out should be as short as possible, but still long enough to avoid discarding the state cache of any window that's still in use. A good starting value is the same value as the inactivity logout. If you don't specify this

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

property, or if you enter *0,*status blocks are tracked per session —all status blocks are deleted when the user signs out or when the inactivity logout interval expires.

Note: A potential inconvenience with this property is that users might be inactive in a window without closing it, then return to it after the interval that you specify here has expired. In that case, they're presented with the application's search page.

Cache Homepage

Select to apply homepage caching on the browser. When this check box is selected, the Homepage Stale Interval field and the Browsers grid become available.

This check box is selected by default.

When the Cache Homepage check box is selected, the following occurs:

- All non-component-based pagelets will be cached on the browser by default unless the pagelet has a PORTAL_ NOBROWSERCACHE content reference attribute set to *Y.*
- Component-based pagelets will not be cached on the browser by default unless the pagelet has a PORTAL_ NOBROWSERCACHE content reference attribute set to *N.*

  Important! Do not enable homepage caching for any component-based pagelet that uses interactive processing mode.

When this check box is deselected, any PORTAL_ NOBROWSERCACHE content reference attributes are ignored.

Homepage Stale Interval

Enter the number of seconds that the browser should wait before requesting an updated homepage from the portal server.

The default value is *1200* seconds (20 minutes).

See Implementing Homepage Caching, Implementing PeopleSoft Page Caching.

Browsers

Use this grid to identify the browsers that you don't want to cache the homepage. For each browser make, model, and version, supply the identifying user agent ID and deselect the Cache Home Page check box. To re-enable caching for a listed browser, select the browser's Cache Home Page check box.

Any browser that is not listed caches the homepage if the global Cache Homepage check box is selected.

Note: If the global Cache Homepage check box is deselected, you can't enable homepage caching for individual browsers.

See Implementing Homepage Caching.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

On the Web Server

Cache Proxied JavaScripts

Select to enable caching of proxied JavaScripts on the portal server. This check box is selected by default.

Cache Portal Objects

Select to enable metadata caching. This check box is selected by default.

Cache Stale Interval

Enter the number of seconds that should pass before the portal refreshes the metadata cache. The default value of this property is *86400* seconds (24 hours).

Note: This property competes for effect with Cache Purge All Hit Count.

Cache Target Content

Select to cache all target content HTML that has an appropriately defined Cache element. This check box is selected by default.

Cache Menu

Select to enable web server-based navigation caching.

For all profiles, including those that are newly created, the default value is *False.*

Cache Purge All Hit Count

Specify the maximum total number of HTTP requests that the web server should receive for metadata objects before it purges the metadata cache, forcing the portal to refresh the cache.

The default value is *1000* requests. Set this property to *0* to disable the feature.

Note: This property competes for effect with the cache stale interval.

See Implementing Target Content Caching, Implementing Metadata Caching, Implementing Proxied JavaScript Caching, Administering Web Server-Based Navigation Caching.

Directories

Image Directory

Enter the image file cache directory.

The default value is */cache.*

Image Web Directory

Enter the image file web cache directory.

The default value is */cache.*

CSS Directory (cascading style sheet directory)

Enter the cascading style sheet (CSS) cache directory.

The default value is */cache.*

CSS Web Directory

Enter the CSS web cache directory.

The default value of this field is */cache.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Copy Image/CSS (No Versioning) Select to have the system write a copy of the image and CSS cache with no version number. This check box is provided in case an external reference to the PeopleSoft stylesheet is needed.

Warning! Enable this setting with care. In a production environment, it can double the number of files that are stored in the site's cache directory. Unless a clear need exists to use a custom solution to access style sheets and the like, this check box should be disabled.

This check box is deselected by default.

Chart Directory Enter the directory of the cache for the chart image file.

The default value is */cache/chart.*

Chart Web Directory Enter the directory of the web cache for the chart image file.

The default value is */cache/chart.*

JavaScript Directory Enter the Java Script cache directory.

The default value is */cache.*

JavaScript Web Directory Enter the JavaScript web cache directory.

The default value is */cache.*

Recent Search Results

Enable Caching Select to temporarily store recent (persistent) search results on the browser and web server. Deselecting this setting suppresses all search persistence property settings that you configure in PeopleSoft Application Designer definitions. The default value is *unchecked.*

Enable Caching on Web Browser Select to temporarily store recent (persistent) search results on the web server. deselecting this settingt suppresses all search persistence property settings that you configure in PeopleSoft Application Designer definitions. This option applies only when you use either Internet Explorer 7 or Safari 3 as your browser. This option does not affect Internet Explorer 8, Firefox 3.5, or Safari 4. The default value is *unchecked.*

See "Setting Record Field Use Properties" (PeopleTools 8.54: Application Designer Developer's Guide).

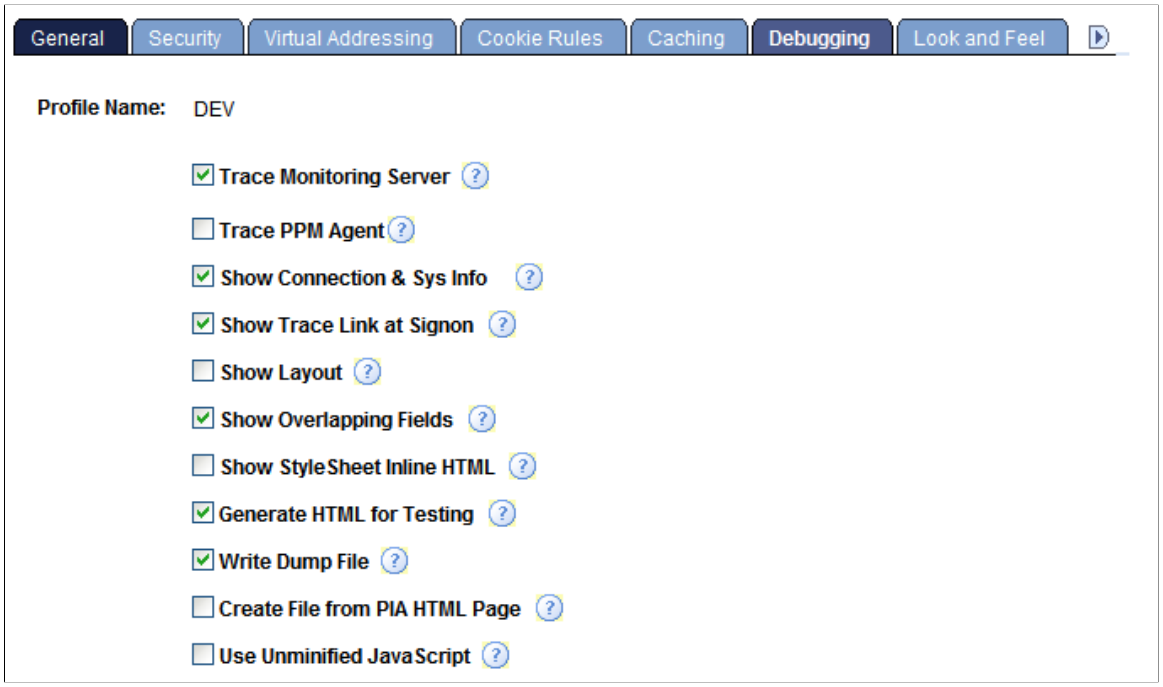
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Trace and Debug Options

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Debugging tab.)

Image: Web Profile Configuration - Debugging page

This example illustrates the fields and controls on the Web Profile Configuration - Debugging page. You can find definitions for the fields and controls later on this page.



Trace Monitoring Server

Select to have the PeopleSoft Performance Monitor and PPMI servlets write debug information to the web server log. This check box is deselected by default.

Note: This feature is not an agent trace. For an agent trace, use Trace PPM Agent.

Trace PPM Agent

Select to enable PeopleSoft Performance Monitor tracing on performance agents.

This check box is deselected by default.

Show Connection Information

Select to have the application generate a system information page, which includes the browser, OS, PeopleTools release, application release, service pack, page definition name, component definition name, menu definition name, user ID, database name, database type, and application server address when you press Ctrl+J. The default is deselected.

Note: This information is useful for orientation and troubleshooting purposes, but might not be suitable for end users.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Show Trace Link at Signon

Select to display a URL link at sign-in. The link opens a page for setting trace parameters.

Show Layout

Select to apply border and color attributes in a table layout for pages. This enables developers to see the position of PeopleSoft Application Designer objects in HTML.

Show Overlapping Fields

Select to include comments in generated HTML pages that may help in diagnosing page layout problems, such as fields overlapping other fields.

Show StyleSheet Inline HTML

Select to insert the page's stylesheet into its generated HTML.

Show JavaScript Inline HTML

Select to display all the JavaScript functions that are used for processing in the generated HTML page.

Generate HTML for Testing

Select to alter the generated HTML to assist with testing and troubleshooting. For example, this option provides additional white space and comments on the page to aid readability. Also, it includes additional name attributes for reference from SQA robot scripts.

Note: Selecting this option may cause some pages or pagelets not to appear correctly. You can also set this in the application server configuration file with the TracePIA option. To ensure that all pages appear correctly, verify that TracePIA isn't enabled in the configuration file.

Write Dump File

Select to have the system write a log file to the web server if an Oracle Jolt exception error occurs.

Create File from PIA HTML Page (create file from PeopleSoft Pure Internet Architecture HTML page)

Select to view and debug the source HTML that the application server generates. The system saves each generated page as *PS_CFG_HOME*\appserv*domain*\LOGS*client**element**N*.html.

The variables in the name are:

- *Domain:* The name of the application server domain.
- *Client:* The name of the machine or Internet Protocol (IP) address where the browser is running.
- *Element:* The name of the process or object that generates the HTML (query name for query, program name for iScripts, and so on).
- *N:* The state number for the generated page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Warning! Use this tracing feature only for troubleshooting and testing. Enabling this feature generates numerous directories and files on the application server, which significantly affects performance. Use it for short periods, preferably only for a single-user test scenario. Never enable it on a production website.

Use Unminified JavaScript

Select to disable the execution of minified JavaScript for the PeopleSoft Pure Internet Architecture domain.

During runtime, the PeopleSoft system compresses, or minifies, all JavaScript stored in HTML definitions created in Application Designer. This is the default behavior. JavaScript that has been minified has been stripped down to only the essential strings required for runtime execution to save bandwidth.

For more information on minified JavaScript in HTML definitions, see "Working with JavaScript in HTML Definitions" (PeopleTools 8.54: Application Designer Developer's Guide).

See Configuring Custom Properties and "Working with Performance Monitor Web Profile Properties" (PeopleTools 8.54: Performance Monitor).

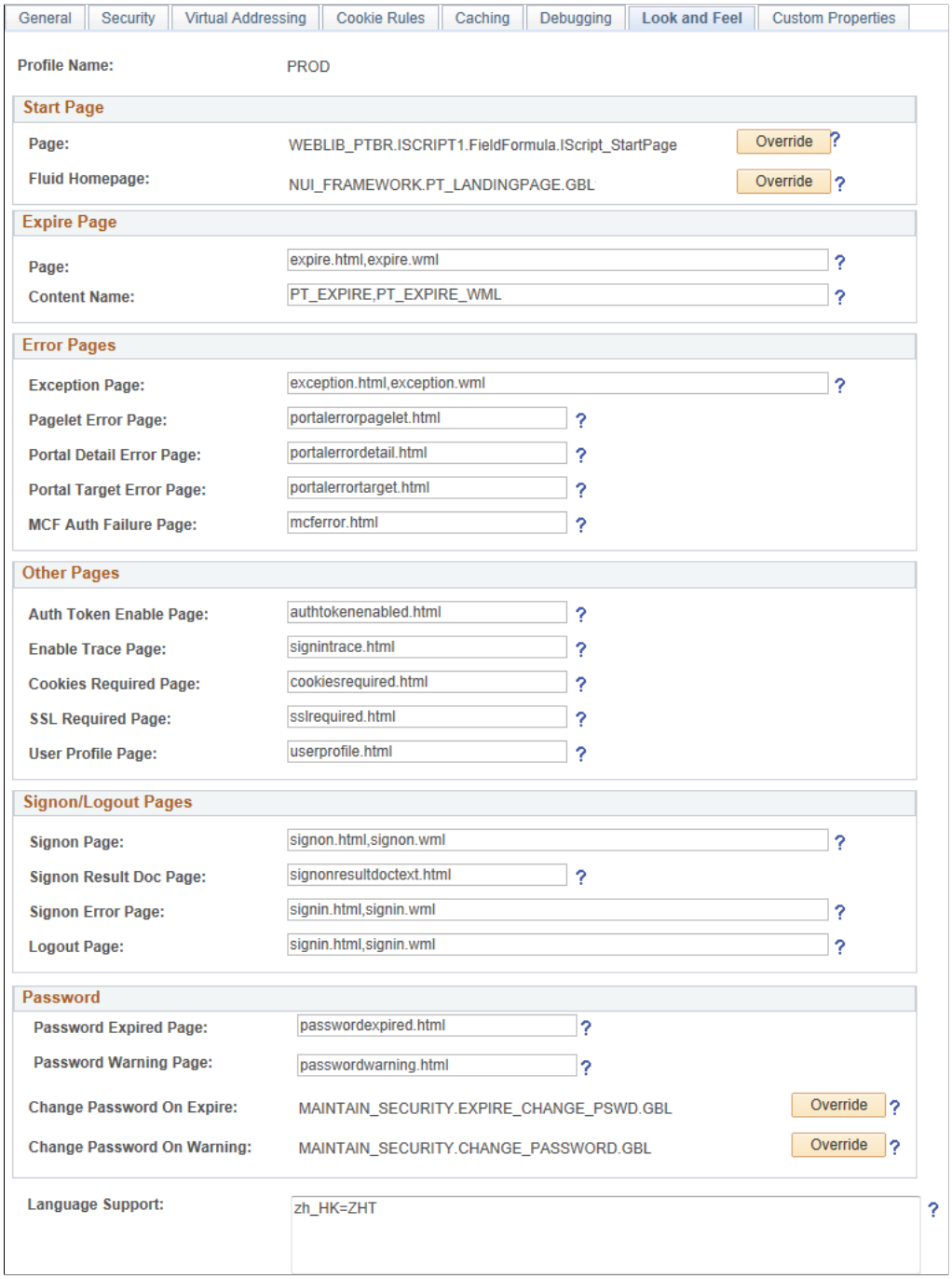
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Appearance and Character

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Look and Feel tab.)

Image: Web Profile Configuration - Look and Feel page

This example illustrates the fields and controls on the Web Profile Configuration - Look and Feel page. You can find definitions for the fields and controls later on this page.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Start Page

Page

Displays the script to which the system redirects users after a successful sign-in. This value references the iScripts that build the PeopleSoft navigation.

The default value is *WEBLIB_PTBR.ISCRIPT1.FieldFormula.IScript_StartPage.*

Override

Click to change the start page script. The Override Start Page Script page is displayed with these fields:

- Record (Table) Name: The default value is *WEBLIB_PTBR.*
- Field Name: The default value is *ISCRIPT1.*
- PeopleCode Event Name: The default value is *FieldFormula.*
- PeopleCode Function Name: The default value is *IScript_StartPage.*

The values of these four fields are concatenated to produce the value of the Page field on the Look and Feel page.

Fluid Homepage

Displays the name of the component used to render fluid homepage tabs. The default value is *NUI_FRAMEWORK.PT_LANDINGPAGE.GBL.*

See "Web Profile Configuration - Look and Feel Page" (PeopleTools 8.54: Fluid User Interface Developer's Guide) for more information.

Expire Page

Page

Displays the HTML page containing text variables that are defined in text.properties. This page appears when user inactivity exceeds the limit that is specified by the Inactivity Logout field in the Authenticated Users region of the Web Profile Configuration - Security page.

Do not change this setting.

The default value is *expire.html,expire.wml.*

Content Name

Enter the content name that is stored in the HTML catalog. It appears when a page has expired due to reaching the limit that is specified by the Number of States Supported field in the On the Browser region of the Web Profile Configuration - Caching page.

Change this value with caution.

The default value is *PT_EXPIRE,PT_EXPIRE_WML.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Configuring Portal Security, Configuring Caching.

Error Pages

Exception Page

Enter the name of the page that is used in Java to handle exceptions.

Change this value with caution.

The default value is *exception.html,exception.wml.*

MCF Auth Failure Page (multichannel framework authorization failure page)

Enter the name of the multichannel framework authorization failure page.

The default value is *mcferror.html.*

Pagelet Error Page

Enter the name of an initial error message page to be displayed in the user's browser when a portal pagelet is unavailable.

The default value is *portalerrorpagelet.html.*

Portal Target Error Page

Enter the name of an initial error message page to be displayed in the user's browser when target content is unavailable.

The default value is *portalerrortarget.html.*

Portal Detail Error Page

Enter the name of an error message page that can be displayed in the user's browser to provide more detail about an error when retrieving portal content.

The default value is *portalerrordetail.html.*

Because your PeopleSoft portal can aggregate its content from many different sources, errors can occur for a variety of reasons. An error may occur if:

- A content server is down.
- An invalid URL is specified.
- A portal node from which content is being requested is inactive.
- A portal node fails to deliver requested content within the time-out interval that you specify by using the PSTIMEOUT content reference attribute.

See Configuring Pagelet Time-out Settings.

For practical purposes, all of these reasons result in the requested content being unavailable.

The values that you specify for the pagelet error page, portal target error page, and portal detail error page are the names of HTML pages that present customizable, translatable error messages. The default pages are delivered with the PeopleSoft system and provide useful error messages without any modification.

You can use the delivered error pages, modify them, or create your own. You can choose additional error message information to be displayed by using bind-type variables of two types in your error pages:

- Numeric message variables

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

These map to strings that are predefined in the following file:

PIA_HOME\webserv\peoplesoft\applications\peoplesoft\PORTAL\WEB-INF\psftdocs*sitename*\text.properties

For example, the numeric variable 5006 maps to the phrase "The portal was unable to retrieve the page you are looking for."

- Session variables

  These map to system information that depends on the current page and activity.

  Session variables have values that are dynamically resolved by the portal engine at runtime. Available session variables include:

  - ErrorTitle
  - ErrorDescription
  - ErrorURL
  - AccessedURL
  - OriginalURL
  - TargetPage
  - StackTrace
  - DetailError

    DetailError is the name of the page that is specified by the Portal Detail Error Page field. This can be used to specify a link target.

To invoke a numeric or session variable, you insert it between the strings `<%=` and `%>` in your HTML. For example: `<%=5009%>` or `<%=ErrorDescription%>`. Examine the delivered error pages for examples of how to use the numeric and session variables within your HTML.

Other Pages

Auth Token Enable Page (authorization token enable page)

Enter the name of the page to be displayed when the site being accessed is configured with an authentication domain (or web server session cookie domain), and the URL from the browser doesn't include the domain. This page contains a link to a sign-in page that produces the correct URL for the site. You can modify the content interface and design, but do not change the internal metatags that generate the correct URL.

The default value is *authtokenenabled.html.*

Enable Trace Page

Enter the name of the page to be used to set trace parameters. If you enable tracing, this page appears before the sign-in page so that you can set the trace parameters and then sign in to the system.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The default value is *signintrace.html.*

Cookies Required Page

Enter the name of a page to be displayed when the browser does not accept cookies. You should configure browsers to accept cookies.

The default value is *cookiesrequired.html.*

SSL Required Page

Displays the name of the page that should appear if the Secured Access Only check box is selected in the SSL region of the Web Profile Configuration - Security page, and the user is unable to proceed without SSL.

Do not change this setting.

The default value is *sslrequired.html.*

User Profile Page

Displays the name of the page that appears when the user clicks the link from a password-expired page.

Do not change this setting.

The default value is *userprofile.html.*

See Configuring General Portal Properties.

Signon/Logout Pages

Signon Page

Displays the name of the page that redirects to the servlet for the sign-in process.

Do not change this setting.

The default value is *signon.html,signon.wml.*

Signon Result Doc Page (signon result document page)

If your portal site is configured with sign-in PeopleCode to additionally validate the user, that code can call a result document if the user authentication failed. You can present the result document to the user in several ways by specifying one of these HTML documents in this field:

signonresultdoctext.html: The result text is wrapped in a standard portal error page. This is the default value.

signonresultdocpage.html: The result text itself is formatted as the error page.

signonresultdocredirect.html: This page redirects the user to your own result document, using the *<%=resultDoc%>* session variable element.

Signon Error Page

Enter the name of the page that should appear when the user makes an error signing in. To customize your sign-in page, clone signin.html as a starting point.

Change this value with caution.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The default value is *signin.html,signin.wml.*

Logout Page

Enter the name of the page that should appear when the user signs out. You may define a custom logout page.

Change this value with caution.

The default value is *signin.html,signin.wml.*

Password

Password Expired Page

Displays the name of the page that should appear when the user password is expired.

Do not change this setting.

The default value is *passwordexpired.html.*

Password Warning Page

Displays the name of the page that should appear when the user's password is about to expire in the number of days specified in PeopleSoft security.

Do not change this setting.

The default value is *passwordwarning.html.*

Change Password on Expire

Displays the Change Password page content ID. The system uses the value in passwordexpired.html to take the user to the Change Password page when a password is expired.

The default value is *MAINTAIN_SECURITY.EXPIRE_CHANGE_PSWD.GBL.*

Override

Click to override the expired password change page. The Override Change Password on Expire Page page appears.

- Menu Name: The field is blank by default.
- Component: The field is blank by default.
- Market: The default value is *Global.*

Change Password On Warning

Displays the content ID for the Change Password page. The system uses the value that you enter in passwordwarning.html to take the user to the Change Password page when a password warning is required.

The default value is *MAINTAIN_SECURITY.CHANGE_PASSWORD.GBL.*

Override

Click to override the Password Change Warning page. The Override Change Password on Warning Page page appears.

- Menu Name: The field is blank by default.
- Component: The field is blank by default.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Market: The default value is *Global*.

(Optional) Language Support

Specify additional mappings from International Organization for Standardization (ISO) locale codes to PeopleSoft language codes, using one line per entry.

The PeopleSoft system uses proprietary codes to represent user languages, and the PeopleSoft Pure Internet Architecture sign-in page appears in the PeopleSoft language corresponding to the user's browser language setting (an ISO locale code). This field enables you to provide mappings from additional ISO locale codes to PeopleSoft language codes to support a wider range of browser-specific language settings.

For example, the entry *zh_HK=ZHT* maps the ISO locale code *zh_HK* (Hong Kong Chinese) to the PeopleSoft language code *ZHT* (traditional Chinese) so that traditional Chinese is the default sign-in language for browsers with the preferred language set to Hong Kong Chinese.

Your PeopleSoft application is delivered with default mappings, shown on the Manage Installed Languages page. The default mappings don't require entries in this field.

Related Links

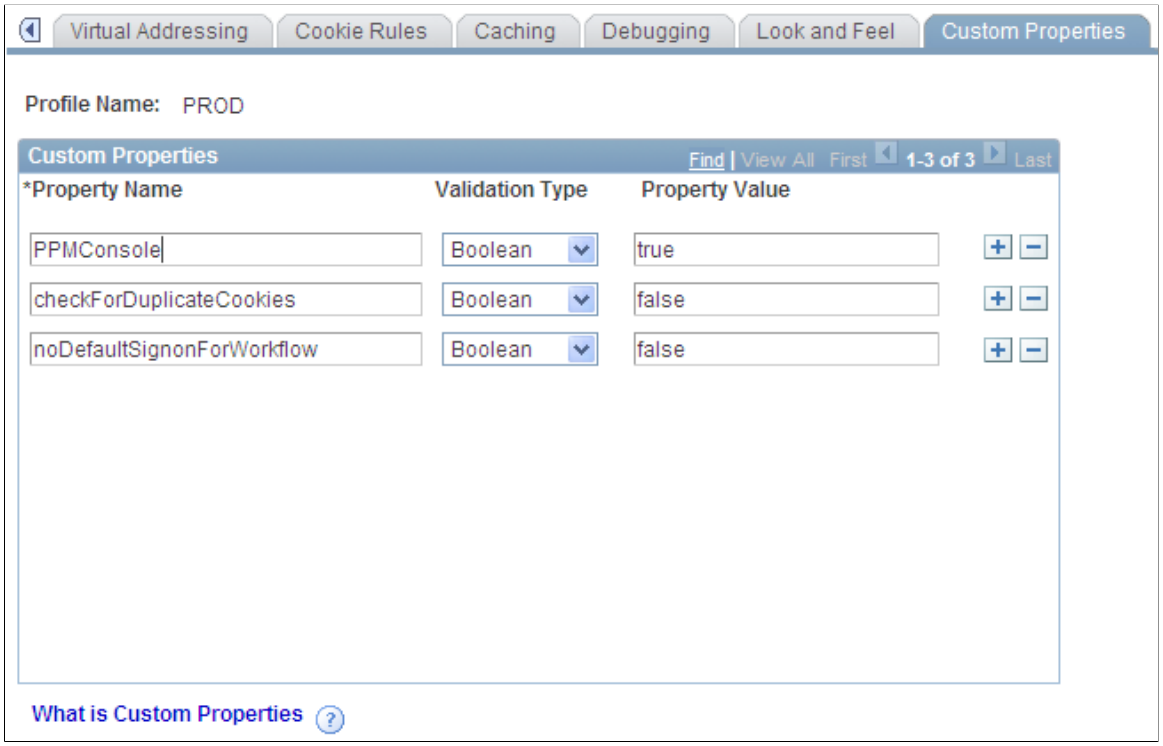
"Understanding the Addition of New Language Definitions" (PeopleTools 8.54: Global Technology)

Configuring Custom Properties

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Custom Properties tab.)

Image: Web Profile Configuration - Custom Properties page

This example illustrates the fields and controls on the Web Profile Configuration - Custom Properties page. You can find definitions for the fields and controls later on this page.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

This page enables the use of web profile properties that have been added since the current release of PeopleTools was shipped, or that are needed only for backward compatibility. For each property, you enter the property name, select its validation type, and provide the appropriate property value. The validation type ensures that the property value that you enter is the correct format. The currently available custom properties are:

AJAXConsole

This property, although not displayed, enables you to launch an interface for viewing diagnostic information related to AJAX parameters that are required by PeopleSoft support personnel, who will provide you with the correct values when necessary.

attachmentExtraWindow

This property enables you to prevent the appearance of a message window when Internet Explorer is used to view certain types of attachments (such as *.doc files) that are not directly displayed by the browser but are instead displayed by the application associated with files of that type. This message window requires explicit user action to dismiss it. However, when Internet Explorer's "Automatic prompting for file downloads" property is set to Disable for the Internet Explorer security zone being used, this message window provides the user with an opportunity (via Internet Explorer's Information Bar) to permit the download of such an attachment to complete rather than having that download silently blocked. The property takes a value of 2 to prevent the appearance of the window. All other values used for this property will allow the appearance of this window.

Note: Therefore, if you set the property value to 2 and use the default Internet Explorer settings, some file attachments won't work.

---

auditPWD

This property enables certain debug and control settings that are required by PeopleSoft support personnel, who will provide you with the correct values when necessary. Never set this property in a production environment unless directed to do so.

checkForDuplicateCookies

Duplicate cookies occur when all of the following are true:

- You have two PeopleSoft applications installed for which you have not implemented single signon functionality.
- The authentication domain that you specify for one application's web server is a subset of the authentication domain that you specify for the other, such as *.mycompany.com* and *.user.mycompany.com*.
- A user attempts to sign in to both applications simultaneously from the same browser.

When these conditions are met, the browser presents multiple cookies to each application, which produces unpredictable browser behavior, such as displaying the sign-in page or a page expiration message.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

For this property, specify a validation type of *Boolean* and enter one of these property values:

True: The portal checks for sets of duplicate cookies. When a duplicate is found, the user is taken back to the sign-in page with this message: “Your browser sent multiple sets of cookies.” You can change the text of this message by editing number 107 in the errors.properties file of the portal site.

False: The portal doesn't check for sets of duplicate cookies. This is the default setting, which applies when this property isn't specified.

To avoid this issue altogether with applications that don't use single signon functionality, make sure that you specify authentication domains that aren't subsets of each other, such as *.user.mycompany.com* and *.corp.mycompany.com*.

EnableAutocomplete

This property, although not displayed, enables AJAX features associated with autocomplete. For this property, specify a validation type of *Boolean* and enter *False* as the property value to disable autocomplete at the system level.

Note: If this property is not set to false, you can control autocomplete through personalizations.

---

mutualAuth

This property, although not displayed, controls whether your environment is set up for mutual authentication. The default value is *True.*

Note: If you are using Lightweight Directory Access Protocol (LDAP) authentication, you must use the default setting.

If you are not using LDAP authentication, a Java exception error caused by a failure to retrieve a certificate property may appear in your webserver log. If this occurs, you must change this value to *False* to disable the authentication from trying to retrieve the certificate property.

PPMConsole

This property, although not displayed, enables you to launch an interface for viewing diagnostic information related to agents and PPMI clients. The default value is *False.*

noDefaultSignon ForWorkflow

Use the Web Profile Custom Property, noDefaultSignonForWorkflow, to control whether a user sees the default signon page after clicking a workflow link. When noDefaultSignonForWorkflow is set to true, the system disables Public Access for workflow notification URLs. By default, noDefaultSignonForWorkflow is set to *False.* See the following table for information related to the behavior of this setting.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: This property applies to worklist URLs, not all workflow URLs. Only the URLs that contain a /w/ use this property.

unauthSessionInterval

Use the Web Profile Custom Property to set session time-out (in seconds) during the authentication process. The default value is *120* seconds.

The purpose is to limit the effectiveness of DOS attacks on failed authentications. To determine the proper time for this property, check the time in seconds that it takes to send an HTTP(S) request from the browser to the web server and multiply the result by 2.

This table describes the behavior of the system based on the noDefaultSignonForWorkflow setting.

Configuration	***User Session***	***User Action***	***System Behavior***
Public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to true.	The user has no session with the system.	The user clicks a workflow notification URL in the email.	The system displays the sign-in page to the user. Once the user signs in to the system with a valid user ID and password, the system takes the user to the workflow notification based on the user profile privileges.
Public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to true.	The user already holds a valid session with the system through the public access.	The user clicks a workflow notification URL in the email.	Because the user already has a valid session, the system takes the user to the workflow notification. If the public access user profile does not have workflow privileges, the system displays an authorization failure page to the user.
Public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to true.	The user is already signed in to the system with a valid user ID and password (other than default sign-in)	The user clicks a workflow notification URL in the email.	The system takes the user to the workflow notification based on the user profile privileges.
When the public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to false.	Any	Any	The system behaves as configured.
When the public user access (default signon) is disabled and the custom web property “noDefaultSignonForWorkflow” is set to false/true.	Any	Any	The system behaves as configured.

Note: If the public user does not have workflow privileges, then close the existing public user session (browser) before clicking the workflow link.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

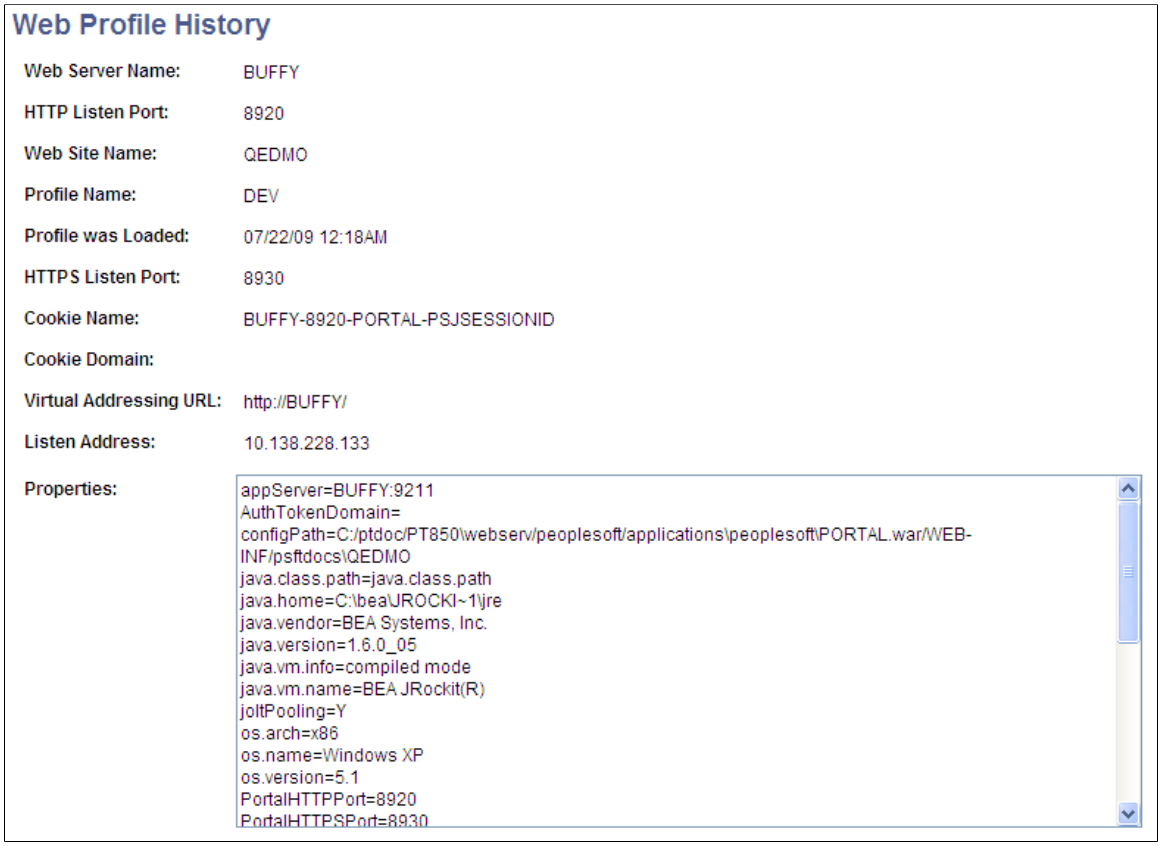
See "PeopleSoft Sign In" (PeopleTools 8.54: Security Administration) and "Viewing Monitor Servlet Diagnostics" (PeopleTools 8.54: Performance Monitor).

Viewing Web Profile History

Access the Web Profile History page. (Select PeopleTools, Web Profile, Web Profile History.)

Image: Web Profile History page

This example illustrates the fields and controls on the Web Profile History page.



Use this page to review the current portal attributes of a web server, website, and web profile, including the web server's HTTP and HTTPS listen ports, the web profile that was last loaded, and the current HTTP session's cookie name and authentication domain.

In addition, the Properties field displays the relevant settings of the web server instance at the time it was loaded by the portal.

Defining Portal Nodes

This section provides an overview of portal nodes and discusses how to:

- Specify the node type.
- Set portal node characteristics.

Understanding Portal Nodes

You can create and maintain a node definition in the Node Definitions component (IB_NODE). A node is defined once but can be used for multiple purposes. Several pages in the Node Definitions component are

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

used only for integration purposes by PeopleSoft Integration Broker. For portal purposes, define a portal node by using the Node Definitions and Portal pages.

Nodes must be blank for external content references.

Note: You cannot delete a node that is in use by a content reference when the Save button is clicked. If the node is in use, the node name is not deleted.

Specifying the Node Type

Access the Node Definitions page. (Select PeopleTools, Portal, Node Definitions..)

Most of the options on the Node Definitions page apply only to Integration Broker nodes, except for the Node Type. A node type designates the URL format for the node.

Node Type

External: Select if this node represents a non-PeopleSoft system, such as an external website, like www.example.com.

ICType: Select if this node represents a PeopleTools database prior to PeopleTools 8.4. This allows compatibility between pre-PeopleTools 8.4 databases and portal databases for PeopleTools 8.4 and later.

PIA (PeopleSoft Pure Internet Architecture): Select if this node represents a PeopleTools database that is using PeopleTools 8.4 or later. This is the default setting.

Related Links

"Understanding Nodes" (PeopleTools 8.54: Integration Broker Administration)

"Configuring Nodes" (PeopleTools 8.54: Integration Broker Administration)

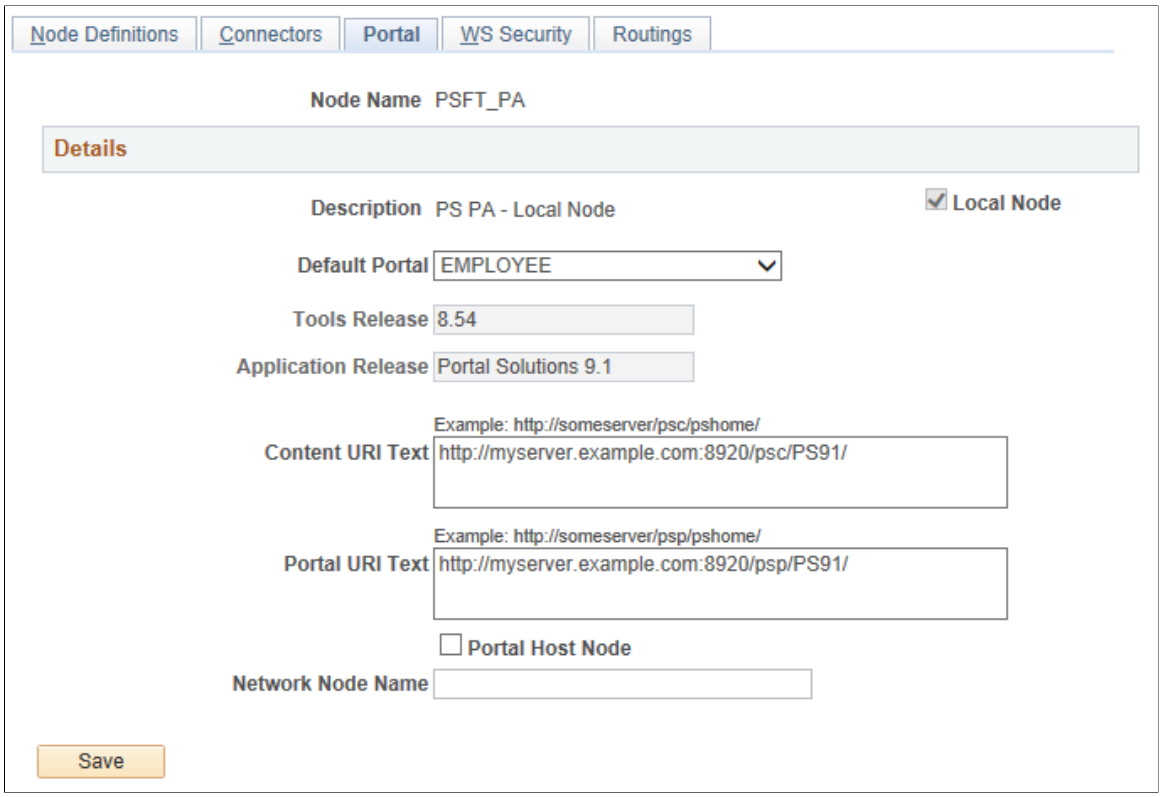
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Setting Portal Node Characteristics

Access the Portal page. (From the Node Definitions page, select the Portal page.)

Image: Node Definitions - Portal page

This example illustrates the fields and controls on the Node Definitions - Portal page. You can find definitions for the fields and controls later on this page.



Node Name

Displays the node name, description, and local node check box status from the Node Definitions page.

Default Portal

For local nodes only, you can specify the portal definition to which this node is associated.

Note: The portal host node for a portal is specified on the Portal Definitions page.

Tools Release

Enter the version of PeopleTools that is running. This is a free-form field. You can enter any information that you want; however, Oracle recommends that you use the same PeopleTools release that is available when you press Ctrl + J.

Important! The Tools Release value is required to support the remote pagelet search feature.

Application Release

Enter the version of the PeopleSoft application that you are running. This is a free-form field. You can enter any information that you want; however, Oracle recommends that you use the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

same application name release that is available when you press Ctrl + J.

Content URI Text
Portal URI Text

Enter the uniform resource identifier (URI) of the pscontent servlet (psc) for the node and the URI text for the portal servlet (psp). For example:

```
http://myserver.example.com:8010/psc/PS/
```

```
http://myserver.example.com:8010/psp/PS/
```

Important! The server name and port in these URIs must not exceed 30 characters in length. For example, *http://mybiglongservername.peoplesoft.com:8080* not including the *http://* prefix, is 39 characters—9 characters too long.

Portal Host Node

Note: The Portal Host Node and the Network Node Name fields are required in a unified navigation system, which for example consists of a PeopleSoft Interaction Hub database and an HCM database. These fields must be set on a PeopleSoft Interaction Hub system and a content provider system. These fields are set so that the unified fluid home page is able to obtain the corresponding IB node for a given portal host node.

Select the Portal Host Node check box:

- For a remote node on the PeopleSoft Interaction Hub system.
- For each local node of the content provider system.

See the product documentation for

PeopleSoft Interaction Hub: Portal and Site Administration

, “Administering Unified Navigation in PeopleSoft Interaction Hub,” Setting Up Single Signon.

Network Node Name

For a remote node on the PeopleSoft Interaction Hub system, enter the default local node of the remote system.

For each local node of the content provider system, enter the default local node of the PeopleSoft Interaction Hub system.

Related Links

Administering Portal Definitions
"Understanding Nodes" (PeopleTools 8.54: Integration Broker Administration)

Implementing Single Signon Functionality

This section provides an overview of single signon functionality and discusses how to sign in to a non-default portal.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Understanding Single Signon Functionality

PeopleSoft PeopleTools supports single signon functionality for use with the PeopleSoft Pure Internet Architecture. Within the context of your PeopleSoft system, *single signon* means that after a user has been authenticated by one PeopleSoft application server, that user can access additional PeopleSoft application servers without entering a user ID or password. Although the user is actually accessing a different application server and database, the user navigates seamlessly through the system.

The single signon authentication service is shared by all PeopleTools web services and involves these steps:

1. The first application server (node) checks the HTTP request for a PeopleSoft authentication cookie.
2. If an authentication cookie is not found, then the server returns a language-specific sign-in page.
3. The user completes the sign-in page, entering a user ID and password.
4. The server authenticates the user based on the user ID and password.
5. A web browser cookie stores a unique access token for the user after the initial authentication.
6. An httpSession object is created for the newly authenticated session.
7. When the user connects to another PeopleSoft application server (node), the second application server uses the token in the browser cookie to reauthenticate the user without an additional sign-in process.

Because the portal integrates content from various data sources and application servers into a unified interface, you need to configure single signon functionality before deploying the portal. Users sign in once and then can navigate freely without encountering numerous sign-in pages.

However, when frame-based content is accessed across different machines, single signon functionality does not work when you access content on another web server. If the web servers are on the same internet domain, such as *.corp.peoplesoft.com,* single signon functionality remains effective and a sign-in page does not appear.

Related Links

"PeopleSoft Sign In" (PeopleTools 8.54: Security Administration)

"Understanding the Delivered External Authentication Solutions" (PeopleTools 8.54: Security Administration)

"Understanding Single Signon" (PeopleTools 8.54: Security Administration)

"Request Class Properties" (PeopleTools 8.54: PeopleCode API Reference)

Signing In to a Non-Default Portal

To sign in to a portal other than the default, use a URL such as: http://portalserver/psp/ps/MY_PORTAL?cmd=login.

To create a sign-in page that signs in to a non-default portal, change signon.html like this:

```
<meta HTTP-EQUIV='Refresh' CONTENT='1; URL=../psp/ps/MY_PORTAL/?cmd=login'>
```

If you are a portal administrator and want to sign in to a database to work with PeopleTools-specific menus, replace the existing portal name with the site template name in the URL to make sure that you are administering your local PeopleTools menu and not inadvertently altering the database. Replace this:

http://../psp/ps/EMPLOYEE/

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Instead, use this:

http://../psp/ps/PS_SITETEMPLATE

Related Links

"Request Class Properties" (PeopleTools 8.54: PeopleCode API Reference)
Using Custom Portal Headers

Redirecting Pages with Sensitive Data

If you have a page that contains sensitive data, and you want to redirect the user to it, you should use the provided template HOMEPAGE_DESIGNER_TEMPLATE. For the content reference for the component, specify this as the template name. You should also use PeopleCode to redirect the user instead of using an HTML area on the page.

The following example shows the type of PeopleCode that you could use:

```
&Res = SwitchUser(QE_SWITCHUSER.QE_SWITCH_USER, QE_SWITCHUSER.QE_SWITCH_USER_PWD, Q⇒
E_SWITCHUSER.QE_SWITCHUSERAUTHT, QE_SWITCHUSER.QE_SWITCHUSERINFO);

If (&Res = True) Then
   Transfer( False, MenuName.PROCESSMONITOR, BarName."INQUIRE", ItemName."PROCESSMO⇒
NITOR", Panel."PMN_PRCSLIST", "U");
Else
   QE_SWITCHUSER.QE_SWITCHRES = "N";
   QE_SWITCHUSER.QE_PPCAUTHTOKEN = %AuthenticationToken;
   QE_SWITCHUSER.QE_RESULTDOC = %ResultDocument;

   Record.QE_SWITCH_SRCH.QE_CUR_USER.Value = %UserId;
End-If;
```

Overriding Page Encoding

You can override the character set encoding for any page that is registered in the portal. You define a content reference attribute called PORTAL_ENCODING_OVERRIDE with a value that specifies the character set to use when processing a page. The portal servlet then overrides all other indicators for encoding, including headers and meta tags.

The standard encoding for PeopleSoft pages is UTF-8. The portal can usually determine the encoding of a page; however, if an external, non-UTF-8 page is not displayed correctly, overriding the page encoding may help.

Related Links

"Understanding Character Sets" (PeopleTools 8.54: Global Technology)

Importing Menu Groups into the Portal Registry

This section provides an overview of menu group importing and discusses how to:

- Import menu groups.
- Select components to include in portal navigation.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Understanding Menu Group Importing

The Menu Import Application Engine process (PORTAL_MENU) imports menu groups from the menu definition into a portal registry.

Important! Only system administrators should perform this process. If the user does not have security to the menus and pages, just a folder titled *Label* gets created.

The process creates folders and content references that reflect the menu definition hierarchy. This table shows the folder hierarchy that is imported:

Level	***Menu Item***
Folder level one	Menu group
Folder level two	Menu name
Folder level three	Bar name
Content reference	Component name

Note: Use the Menu Import process to migrate custom menus only. The navigational menus that are delivered with your PeopleSoft application have already been added to the appropriate portal registry. The Menu Import process has no effect on components that are already registered. However, you can use the process to pick up additional components that are not delivered with the PeopleSoft system.

Before you use the Menu Import process, note these points:

- Object names that are assigned by the Menu Import process are autogenerated.

  If a folder name or content reference name already exists, the system appends a number to the generated object name. Consequently, you may have folder object names such as USE, USE1, USE2, and USE3. As a result, use the registration wizard if you plan to copy the application database registry structures into PeopleSoft Interaction Hub. When you use the registration wizard, you can control the object naming, and you can ensure the uniqueness of object names across databases.

- If a component resides in multiple menus, the process creates a separate content reference for each menu and component combination.

- The navigation hierarchy created by the Menu Import process does not match the delivered PeopleSoft hierarchy.

  To keep the navigation consistent, consider putting all custom menus into a common menu group and generating the registry structures for that menu group. Then, once the registry structures are generated, manually move the content references to an appropriate folder in the existing navigation hierarchy.

  To create a common menu group, open the custom menu in PeopleSoft Application Designer. On the Use tab of the Menu properties dialog box, enter a unique name in the Menu Group field. After you save the menu, it is available for other menus to use.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Using the Menu Import Process with PeopleSoft Interaction Hub

If you run the Menu Import process against multiple application databases and merge multiple registries into PeopleSoft Interaction Hub, bar names (level-three folders) can be overwritten because they frequently have duplicate names. The last folder that is copied takes precedence over all previous folders.

The autogenerated object names are based upon the current database portal registry. The Menu Import process cannot check external databases for existing object names, but it does check the current database.

If you run this process on different databases, the different databases may likely contain identical folder or content reference object names that refer to different objects. This can become problematic if you decide to copy these portal registry structures into PeopleSoft Interaction Hub. The same-named objects overwrite each other. You might need to delete the generated folders and content references to make them unique in the future.

Use the registration wizard to register menus and components into the portal registry. Although with the registration wizard you must register items one by one, you can control folder and object names at the time of the import.

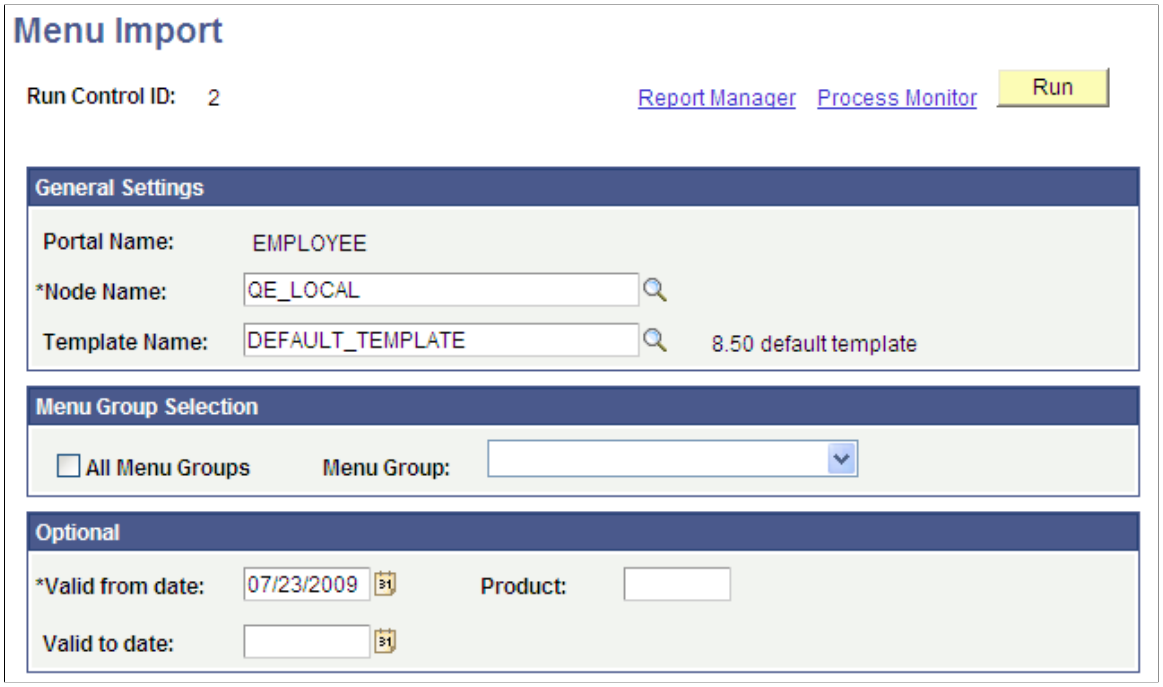
See "Understanding the Registration Wizard" (PeopleTools 8.54: Application Designer Developer's Guide).

Importing Menu Groups

Access the Menu Import page. (Select PeopleTools, Portal, Menu Import.)

Image: Menu Import page

The following screenshot illustrates about importing menu groups on the Menu Import page.



The Menu Import process creates content references for all components that belong to the menu group. This process should be used only to migrate custom menus.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Node Name

Select the node of the content provider where the registry structure resides. The default value is the currently running node.

Note: Never select the local node for an application database. If you copy the generated content references to a PeopleSoft Interaction Hub database, the content reference assumes that the component resides in the PeopleSoft Interaction Hub database, when it does not.

Template Name

(Optional) Specify the name of a template for the portal servlet to use. If you leave this field blank, the servlet uses the template that is associated with the selected node. If a template is not associated with the node, the default template is used for all content references that are created by the import process. The default value is the portal's default template.

Note: Best practice is to not specify a template so that you can perform a mass template change at the content provider node level instead of having to change each content reference template individually.

All Menu Groups

Select to import all menu groups.

Menu Group

Select a menu group to import.

Valid from date and Valid to date

Specify dates to make visible content references and folders that are associated with the menu that you are importing.

In general, use January 1, 1900 for the valid from date and leave the valid to date blank unless you need to limit the time frame.

Product

Enter a unique, 4-character identifier for the menu.

If you perform queries against the database, you can use this identifier to identify items that are associated with the menu.

Run

Click to begin the import process.

PeopleSoft Process Scheduler launches an Application Engine program that imports the selected menu group definitions into the portal registry. This process typically takes several minutes, but can require more time depending on the number of items to be processed.

Permission list settings are initially set to conform with the corresponding permissions set for the menu groups being imported. Once the menu group definitions are imported into the portal registry, no dynamic connection exists between the menu group definition in PeopleSoft Application Designer and the portal registry. If you change the security for the menu or component, the registry security is synchronized as long as the menu or component is in the same database as the registry

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

structure. If you change the menu or component name, you must manually update the registry structure.

Process Monitor Click to view the status of the menu import.

After you run the process, click the Home button in the universal navigation header and verify that the menu group has been imported into the portal registry. The new menu group should now appear along with the other folders at the top level.

Note: If the menu group does not appear, open the folder definition for the menu group and define a label for the folder.

Selecting Components to Include in Portal Navigation

During the menu import process, you can control which components are included in your portal navigation system.

To select components to include in portal navigation:

1. From PeopleSoft Application Designer, open the appropriate component.
2. Select File, Definition Properties and select the Use tab.
3. Select the Include in Navigation check box.

   After a menu import, the component is visible to users with appropriate permissions. (You can still change the permissions for any content reference after the menu import as necessary.)

   If you deselect the Include in Navigation check box, the component is not imported into the portal registry during the Menu Import process. You can also set this option at the menu level.
4. Click OK.

   Changes that you make to the component in PeopleSoft Application Designer are not reflected in the portal until you do another menu import.

Related Links

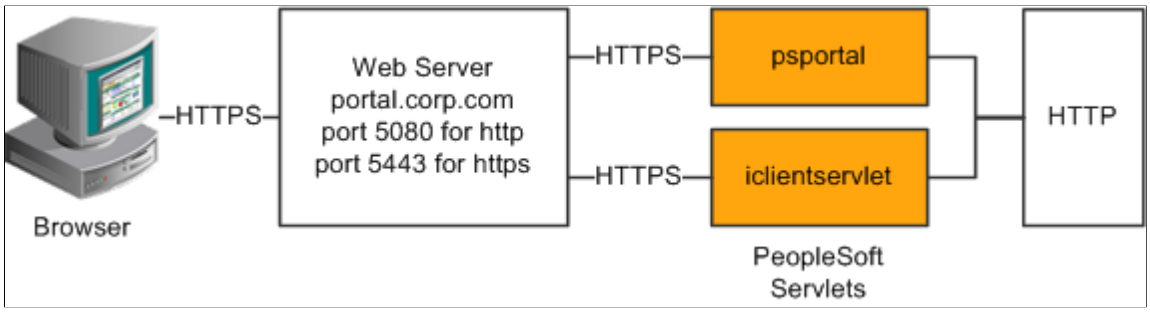
"Using PeopleSoft Application Designer" (PeopleTools 8.54: Application Designer Developer's Guide)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Improving Same-Server Performance Under SSL

Image: Example of the portal running under SSL by using HTTP connections to the PeopleSoft Pure Internet Architecture on the same server

You might run portal and PeopleSoft applications over SSL connections to secure the transmissions between the browser and the web server. However, defining each homepage pagelet as an HTTPS request can cause portal performance to degrade significantly. To achieve secure transmissions between the browser and the server and to achieve acceptable homepage performance, configure the portal to use HTTP connections whenever it needs to talk to a PeopleSoft application that is hosted on the same server as the portal. This diagram shows this setup:



This configuration requires changes to the PeopleTools web configuration properties file. It also requires that a WebServer filter be set up to block non-SSL connections, except those from the portal. The portal does not need to use HTTP or HTTPS to connect to the PeopleSoft Pure Internet Architecture servlet hosting the same website. This configuration is needed only when multiple sites are used on the same web server.

Web Profile Settings

You must tell the portal to use HTTP for all connections that it makes back to the same server on which it is hosted. To do so, modify the web profile configuration for the portal. Change these properties:

Web Profile Page	*Field Name*	*Field Value*
Security	PIA use HTTP Same Server	Selected.
Virtual Addressing	Protocol(in the Default Addressing region of the page)	*HTTPS.*
Virtual Addressing	Port(in the Default Addressing region of the page)	The port on which the web server is listening for HTTPS requests, for example, *5443.* You can leave this field blank if you use the default SSL portal of 443.

Related Links

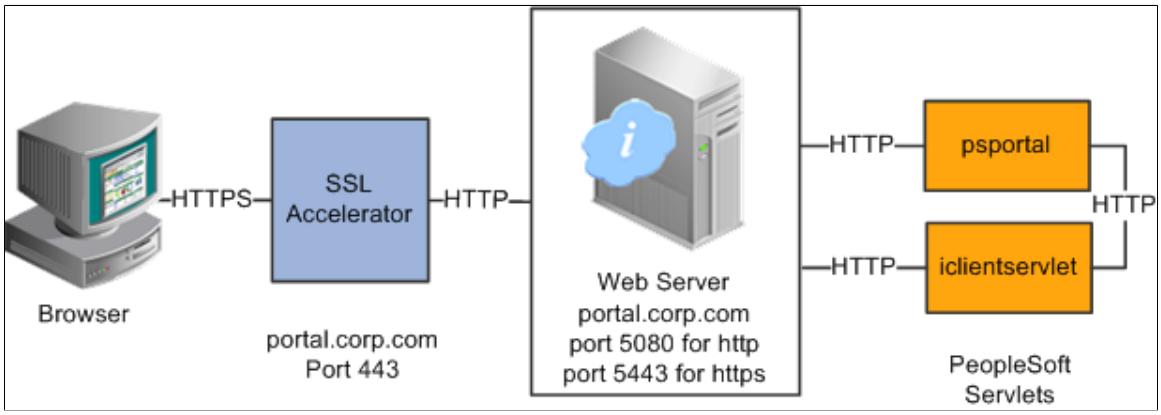
Configuring Web Profiles
Proxy Architecture and Relative URLs

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Using SSL Accelerators

Image: Using an SSL accelerator

You can improve SSL performance by adding an SSL accelerator in front of the web server. You need to have the portal use HTTP for requests to the PeopleSoft Pure Internet Architecture to prevent the portal connection to the PeopleSoft Pure Internet Architecture from going back through the accelerator. This diagram shows this setup:



The configuration is similar to that for configuring a portal running under SSL to use HTTP connections to the PeopleSoft Pure Internet Architecture on the same server; however, you must change the default port to point to the port on which the SSL accelerator is configured to listen.

Important! You must continue to configure Web Server to listen for HTTPS requests, even though the accelerator intercepts all SSL connections. Failure to do so causes the portal to function incorrectly.

Web Profile Settings

Change these properties:

Web Profile Page	*Field Name*	*Field Value*
Security	PIA use HTTP Same Server	Selected. (This might reduce the amount of time it takes to make an HTTP request to the portal.)
Virtual Addressing	Protocol(in the Default Addressing region of the page)	*HTTPS.*
Virtual Addressing	Port(in the Default Addressing region of the page)	The port on which the SSL accelerator is listening for HTTPS requests, for example, *443.*

Related Links

Configuring Web Profiles

"Understanding WebLogic" (PeopleTools 8.54: System and Server Administration)

"Understanding WebSphere Application Server within Your PeopleSoft Implementation" (PeopleTools 8.54: System and Server Administration)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Using Reverse Proxy Servers

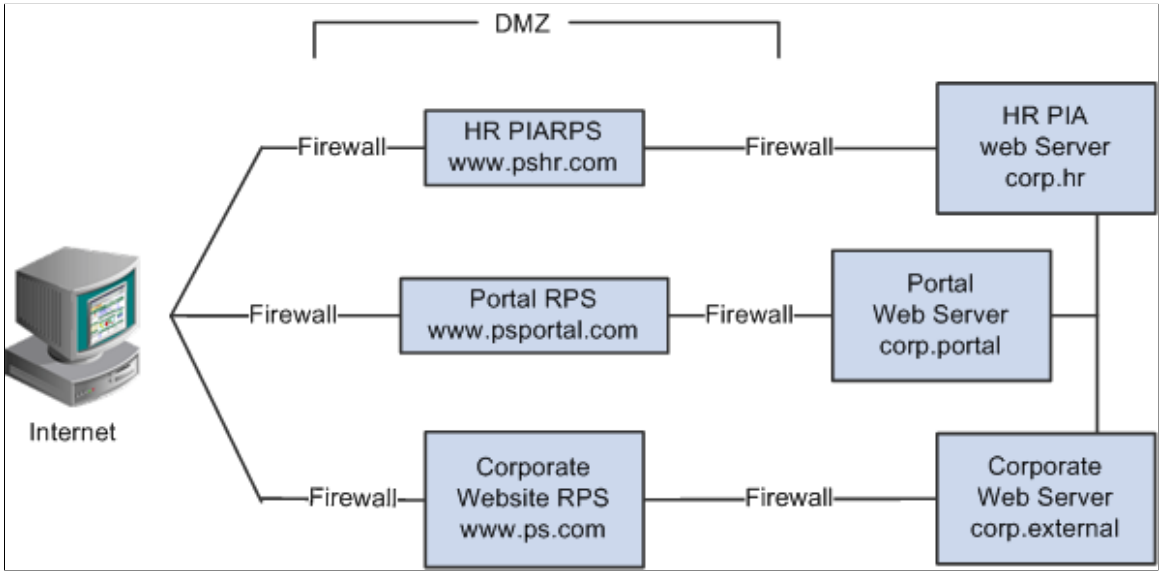
This section provides an overview of reverse proxy servers and discusses how to:

- Make reverse proxy settings.
- Issue requests by using a proxy server.
- Specify hosts that should not be proxied.

Understanding Reverse Proxy Servers

Image: Example of the reverse proxy architecture

A reverse proxy server accepts requests from client browsers on behalf of another web server and then relays those requests to the web server. The reverse proxy server checks web requests and responses to ensure that they adhere to network security policies. For example, a reverse proxy server validates that a command intended for the target web server is a valid command for that server. In this diagram, the HR web server, corp.hr, is behind a reverse proxy server, www.pshr.com:



All internet requests for the HR web server are relayed by www.pshr.com to corp.hr. The site URL routes the client to the proxy as if it were a web server. Replicated content is delivered from the proxy cache to the external client without exposing the origin server or the network residing safely behind the firewall. You can balance the load of an overtaxed web server by installing multiple instances of the web server.

Any external content must come from a node that's registered in the portal database so that the portal servlet can look up the node name based on the URL of the content. The external content HTML is rewritten to use that proxied format.

Related Links

Using Firewalls with Reverse Proxy Servers

Using Reverse Proxy Servers with SSL Accelerators

Making Reverse Proxy Settings

On the Web Profile - Virtual Addressing page, specify the reverse proxy server to use by entering it in the Reverse Proxy Server List grid. Include the protocol, the host name, the HTTP port, and the HTTPS port.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Links

Configuring Virtual Addressing

Issuing Requests by Using a Proxy Server

For you to make a portal issue requests by using a proxy server, the proxy server settings are controlled by these Java system properties:

- http.proxyHost
- http.proxyPort
- https.proxyHost
- https.proxyPort

To use a proxy server for the HTTP protocol, set these properties:

```
http.proxyHost=http_proxy_server_host_name
http.proxyPort=http_proxy_server_port
```

To use a proxy server for the HTTPS protocol, set these properties:

```
https.proxyHost=https_proxy_server_host_name
https.proxyPort=https_proxy_server_port
```

The method for setting the Java system parameters varies by web server (Oracle WebLogic or IBM WebSphere). The properties must be passed to the Java interpreter as command-line directives.

For Oracle WebLogic

Alter the setEnv.cmd or setEnv.sh file located in WebLogic domain's (PIA Install) bin folder. Define the properties on the command line that starts the Java Virtual Machine like this:

```
set PROXY=-Dhttp.proxyHost=proxyhostname -Dhttp.proxyPort=proxy_port -Dhttps.proxyH⇒
ost=proxyhostname -Dhttps.proxyPort=proxy_port
```

Later add the preceding property in startPIA.cmd or startPIA.sh in the Java command line that starts the WebLogic server domain.

```
%JAVA_HOME%\bin\java -ms64m -mx64m
-classpath %JAVA_CLASSPATH% %PROXY%
-Dweblogic.class.path=%WEBLOGIC_CLASSPATH%
-Dweblogic.home=. -Djava.security.manager -Djava.security.policy==.\weblogic.policy⇒
 weblogic.Server

goto finish
```

For IBM WebSphere

All the Java Virtual Machine settings for IBM WebSphere are set in <was_home>\profileTemplates \default\documents\config\templates\servertypes\APPLICATION_SERVER\servers\default.

Note: Here, the profile type is *default* and WebSphere has four types of profiles: cell, default, dmgr, and managed. The preceding path varies based on the type of profile.

Java Virtual Machine properties are embedded within the <jvmEntries> and </jvmEntries> elements.

```
<jvmEntries xmi:id="JavaVirtualMachine_1164791211226" verboseModeClass="false" verb⇒
oseModeGarbageCollection="false" verboseModeJNI="false" runHProf="false" hprofArgum⇒
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
ents="" debugMode="false" debugArgs="-Djava.compiler=NONE -Xdebug -Xnoagent -Xrunjd⇒
wp:transport=dt_socket,server=y,suspend=n,address=7785" genericJvmArguments="-Dprof⇒
ile.name=peoplesoft_auth">
      <systemProperties xmi:id="Property_1164791897101" name="ps_vault" value="c:/p⇒
t849-802-r2-debug/webserv/peoplesoft_auth/installedApps/peoplesoft_authNodeCell/peo⇒
plesoft_auth.ear/psvault" description="Vault location"/>
      <systemProperties xmi:id="Property_1164791897102" name="HttpSessionIdReuse" v⇒
alue="false" description="HttpSessionIdReuse"/>
    </jvmEntries>
```

Specifying Hosts That Should Not Be Proxied

If your portal implementation retrieves content from both an intranet and the internet, you may have a problem retrieving both internal and external content if the proxy server is not aware of the address of the intranet content. You can pevent this situation by specifying that certain hosts (such as your intranet hosts) should not be accessed through the proxy server.

For Oracle WebLogic, specify the hosts not to be proxied in the HTTP_PROXY_NONPROXY_HTTPHOSTS and HTTP_PROXY_NONPROXY_HTTPSHOSTS environment variables with the setEnv.cmd (setEnv.sh in Unix) script. Refer to comments in setEnv for specific syntax.

For IBM WebSphere, specify the hosts not to be proxied in the Java system variables http.nonProxyHosts and https.nonProxyHosts in the IBM WebSphere console:

1. Access the IBM WebSphere administration console.
2. Select Application Servers, *server_name,* Process Definition, Java Virtual Machine, Custom Properties.
3. Click New.
4. Set the property http.nonProxyHosts: List the internal machines for which authentication is not required for routing through the proxy. Separate each machine name in the list with a vertical bar (|).
5. Save the configuration, then sign out and restart IBM WebSphere.

For all the web server platforms, you can list a set of patterns that should not be accessed through proxies, with each one separated by the pipe character (|). This example shows a setting that makes all HTTP connections to the eng domain bypass the proxy settings:

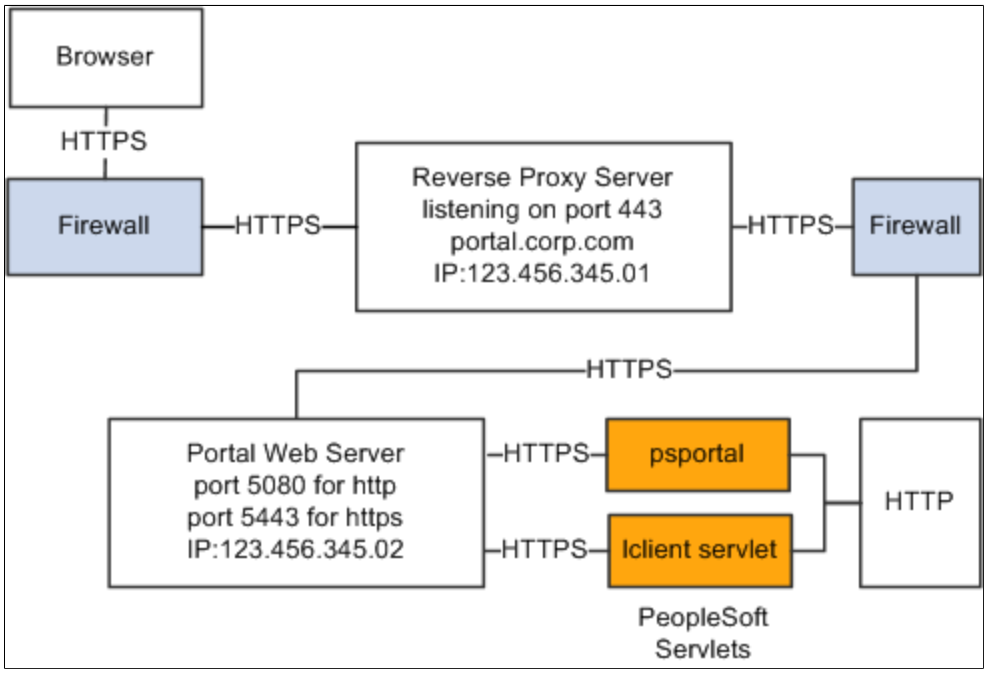
.eng|.eng.peoplesoft.com

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Using Firewalls with Reverse Proxy Servers

Image: Firewall and reverse proxy setup

You can set up a buffer zone in front of the portal web server to prevent unauthorized access to the portal web server and create a more secure environment. A buffer zone is typically configured with a firewall that allows access to a reverse proxy server, which relays incoming requests through a second firewall to the portal web server. This diagram shows this setup:



The configuration is similar to that for configuring a portal running under SSL to use HTTP connections to the PeopleSoft Pure Internet Architecture on the same server; however, you must make these adjustments in the web profile:

Web Profile Page	***Field Name***	***Field Value***
Security	PIA use HTTP Same Server	Selected.
Virtual Addressing	Protocol(in the Default Addressing region of the page)	*HTTPS.*
Virtual Addressing	Name(in the Default Addressing region of the page)	The DNS name of the reverse proxy server, for example, *portal.corp.com.*
Virtual Addressing	Port(in the Default Addressing region of the page)	The port on which the reverse proxy server is listening for HTTPS requests if it's different from the port on which the web server is listening.

Hosts File Setup

The hosts file on the portal web server must have an entry that directs DNS requests for the content provider server name (the reverse proxy server in the previous example) to the portal web server, for example, 123.456.345.02. This enables the portal to make requests for content that is hosted on the same server directly without going back through the reverse proxy server.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

For example:

123.456.345.02 portal.corp.com

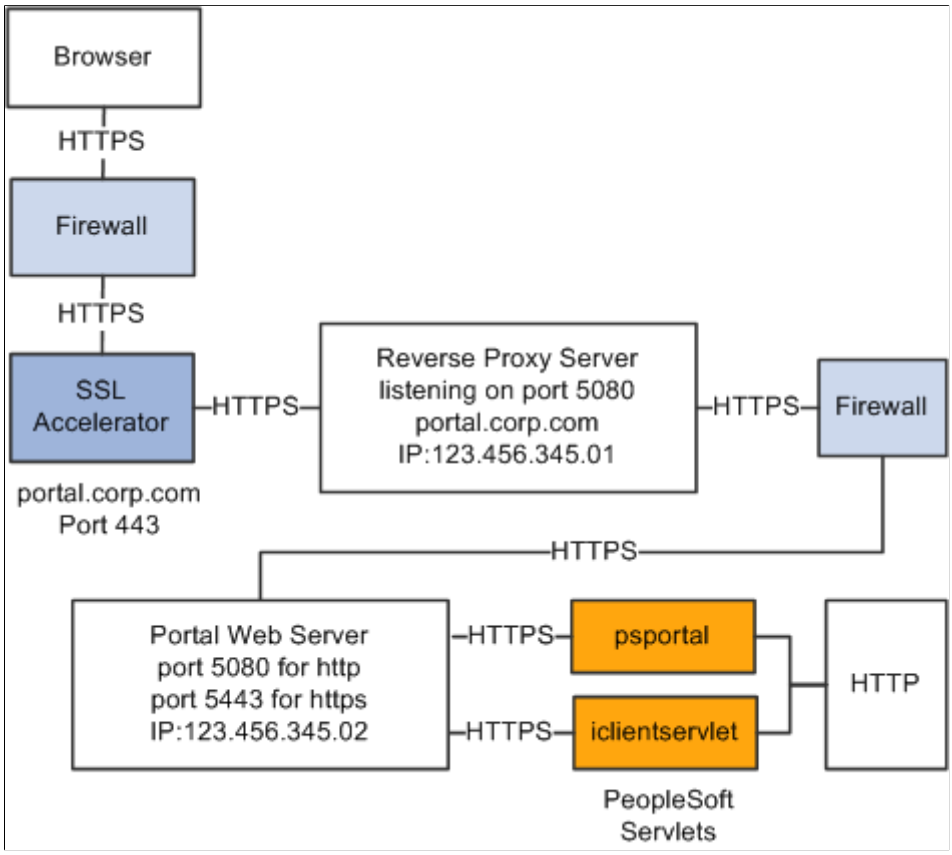
Related Links

Configuring Web Profiles

Using Reverse Proxy Servers with SSL Accelerators

Image: Reverse proxy server with an SSL accelerator

You can combine an SSL accelerator with a reverse proxy server, as illustrated in this diagram:



The configuration is similar to that for configuring a portal running under SSL to use HTTP connections to the PeopleSoft Pure Internet Architecture on the same server; however, you must make these adjustments in the web profile:

Web Profile Page	*Field Name*	*Field Value*
Security	PIA use HTTP Same Server	Selected.
Virtual Addressing	Protocol(in the Default Addressing region of the page)	Set to *HTTPS.*
Virtual Addressing	Name(in the Default Addressing region of the page)	The DNS name of the reverse proxy server, for example, *portal.corp.com.*

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Web Profile Page	*Field Name*	*Field Value*
Virtual Addressing	Port(in the Default Addressing region of the page)	The port on which the SSL accelerator is listening for HTTPS requests, for example, *443.*

Hosts File Setup

The hosts file on the portal web server must have an entry that directs DNS requests for the content provider server name (portal.corp.com) to the portal web server, for example, 123.456.345.02. This enables the portal to make requests for content that is hosted on the same server directly without going back through the reverse proxy server.

For example:

123.456.321.02 portal.corp.com

Related Links

Configuring Web Profiles

"Understanding WebLogic" (PeopleTools 8.54: System and Server Administration)

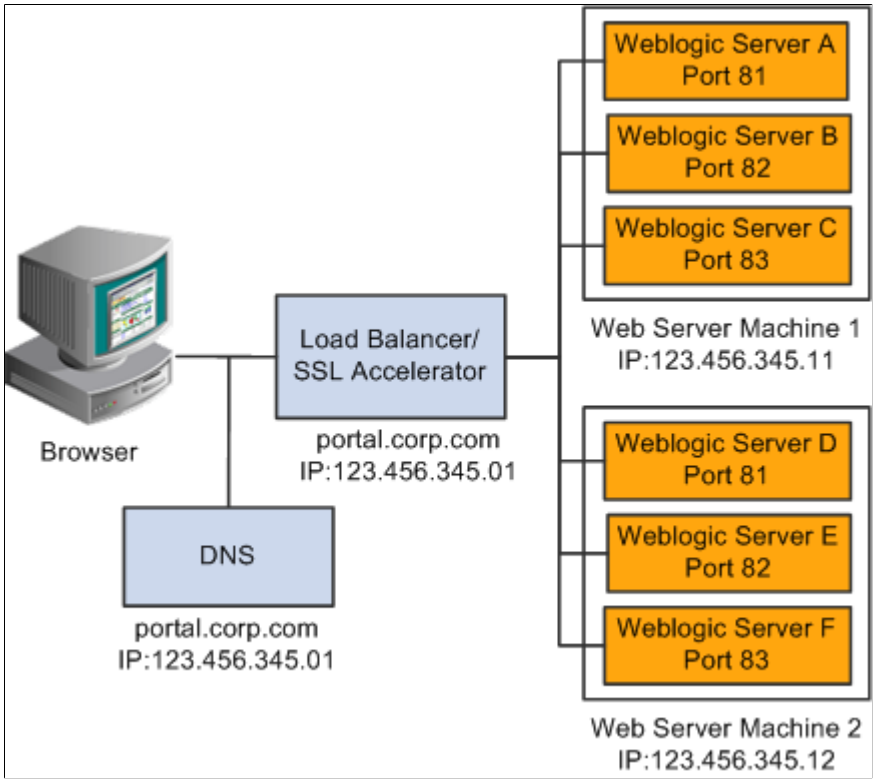
"Understanding WebSphere Application Server within Your PeopleSoft Implementation" (PeopleTools 8.54: System and Server Administration)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Applying Load Balancing

Image: Example of load balancing

You can apply load balancing to a portal by using a hardware load-balancing solution to distribute load across multiple instances of the web server on multiple web server machines. Many of today's hardware load balancers have SSL accelerating options that you should also consider. The number and size of web server machines and the number of web server instances per machine vary and determine load-balancing requirements and hardware selection. This diagram shows a possible load-balancing setup:



The hardware load balancing solution must be configured to enable sticky sessions, which ensures that all requests from a user's browser session are delivered to the same web server session. In the example previous diagram, if the initial request from the browser was directed to Oracle WebLogic Server F, then all subsequent requests from that browser session must be directed to the same instance of Oracle WebLogic.

Keep multiple web server sessions on a single machine to keep the heap size of the Java Virtual Machine manageable and to allow the Java garbage collection to work optimally.

Hosts File Configuration

The hosts file on the portal web server machines in the previous diagram—which we'll call portalws1 and portalws2—must have an entry that directs DNS requests for the portal content provider server name (portal.corp.com) to the portal web server (123.456.345.11 on portalws1). This allows the portal to request content hosted on the same server directly without going back through the load balancer.

This is an example hosts file on portalws1.corp.com:

123.456.345.11 portal.corp.com

123.456.345.11 portalws1.corp.com

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 11

Working with Conditional Navigation

Understanding Conditional Navigation

With the introduction of PeopleSoft Fluid User Interface in PeopleTools, PeopleTools offers administrators the capability to render a page either in classic mode or fluid mode based on conditions and factors. Additionally, you can direct a user to a different content reference, instead of the default content reference, based on conditions and factors. Administrators can configure alternate content reference by specifying conditions. While generating the default content reference, the conditions for an alternate content reference, if specified, are evaluated at runtime.

For example, in an HCM system that is not configured for conditional navigation, when you click the classic Home and Mailing Address content reference (Self Service > Personal Information > Home and Mailing Address) the classic Home and Mailing Address page would be displayed to a user. With conditional navigation, you can configure the classic Home and Mailing Address content reference to point to a fluid equivalent of the page. So, on a browser that supports the fluid interface, you will see the fluid equivalent of the Home and Mailing Address page assuming that the factors and conditions are successfully met. This example is a simple illustration of conditional navigation where a classic content reference is mapped to a fluid content reference.

PeopleSoft supports the following mapping of content references:

- One content reference mapped to one content reference.
- One content reference mapped to multiple content references.
- Multiple content references mapped to one content reference.

See Sample Scenarios of Conditional Navigation

Conditional navigation is dependent on several factors, which are listed here. Some of these factors are also prerequisites that are necessary for using the conditional navigation feature.

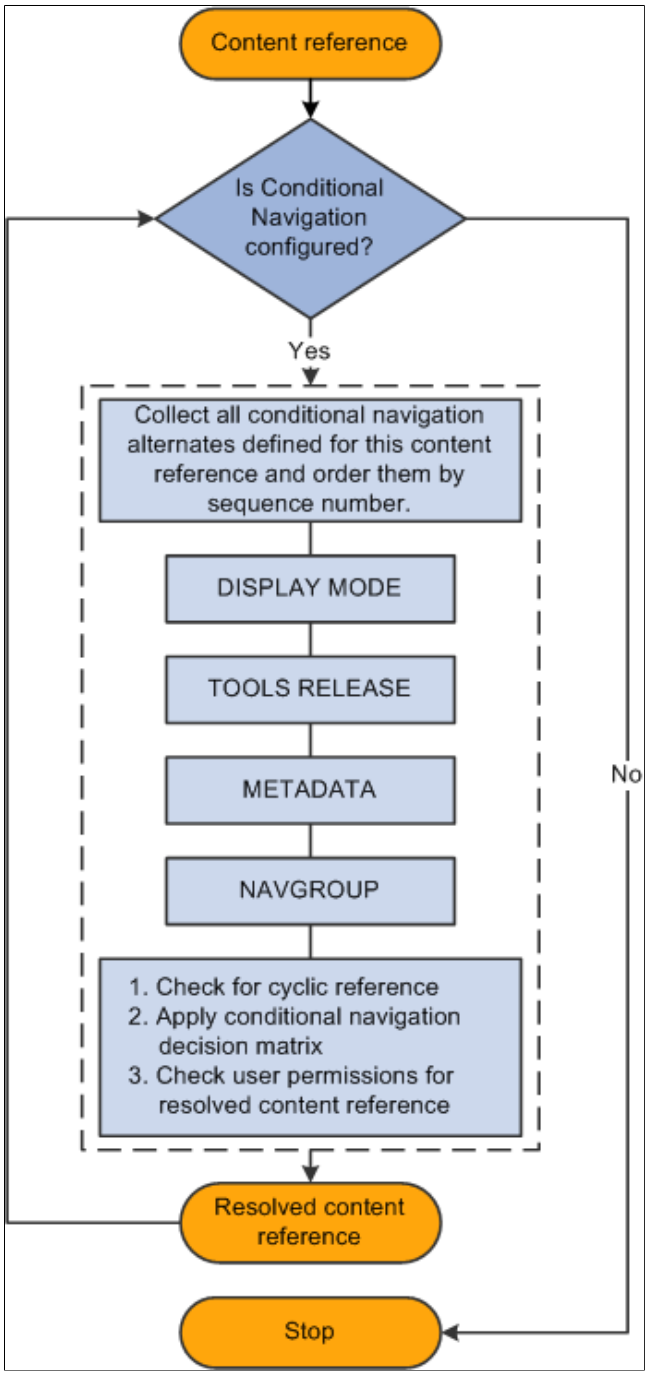
- Content reference settings for classic or fluid mode.
- Web profile settings for classic or fluid mode.
- Browser support for fluid interface.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- User access to the configured alternate content reference.

Image: Conditional navigation process flow

This example illustrates the process flow when conditional navigation is configured.



In a conditional navigation process flow, note the following:

- Multiple conditions or attributes can be configured for a content reference.
- Multiple conditions are evaluated based on their sequence number.
- Conditions with duplicate sequence numbers will not be evaluated.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Upon successful evaluation of conditions, the system encounters a resolved content reference. The first successful condition will stop further evaluation of other conditions for this content reference.
- A resolved content reference may also be configured for conditional navigation, so in this case, the conditions configured in the resolved content reference are also evaluated.
- This process continues till the system does not encounter any other alternate content reference.

Prerequisites

To use the conditional navigation feature, you must ensure that the following prerequisites are in place:

Web Profile Settings for Classic or Fluid Mode

PeopleTools provides options on the Web Profile Configuration page (PeopleTools, Web Profile, Web Profile Configuration, General) to configure the display of fluid pages or classic pages.

Note: The configuration must be completed on the default web profile.

Disable Fluid Mode	Setting this option will disable all fluid pages and content references. Do *not* select this option if you want to use the conditional navigation feature. By default, this option is not selected.
Disable Fluid on Desktop	Setting this option will display only classic pages on desktops and laptops. However, fluid pages will be displayed on tablets and smart phones. By default, this option is selected. Do *not* select this option if you want to use the conditional navigation feature on desktop-based browsers. **Note:** This option is effective only when the Disable Fluid Mode is not set.

Note: If you change any of these configurations, you must restart the Web server for these changes to be effective.

For more information on setting these options, see "Configuring Fluid Mode Web Profile Settings" (PeopleTools 8.54: Fluid User Interface Developer's Guide).

Content Reference Settings for Classic or Fluid Mode

You can specify a content reference as classic or fluid when you create a content reference.

Open the Content Ref Administration page (PeopleTools, Portal, Structure and Content and click the Edit link of a content reference) to view the content reference mode.

For a description of these options, see Administering Content References.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Browser Support for Fluid

Browsers are also a factor that must be considered when you want to view fluid pages or content references because not all browsers support the fluid interface. For a complete list of supported browsers, refer the Certifications tab on My Oracle Support (sign-in required).

Common Terms

The following table lists the common terms used in conditional navigation:

Term	*Description*
Classic display mode	PIA display mode is classic when fluid is disabled on the Web Profile Configuration page or browser is not fluid compatible. See Prerequisites.
Fluid display mode	PIA display mode is fluid when fluid is enabled on the Web Profile Configuration page and browser is fluid compatible. See Prerequisites.

Configuring Conditional Navigation

Configuring conditional navigation involves configuring alternate content references. You use content reference system attributes to configure conditional navigation, and these system attributes are prefixed with CN_. You define the condition in the attribute name field, and you specify the alternate content reference in the attribute value field.

Important! Application developers must not create fluid alternate content references for WorkCenter based pagelets.

You can use the following content reference system attributes:

- CN_DISPLAYMODE.
- CN_TOOLSREL.
- CN_NAVGROUP.
- User defined content reference attribute.

Configuring an Alternate Content Reference Using the CN_DISPLAYMODE Attribute

The CN_DISPLAYMODE_N attribute is used to navigate from a classic content reference to a fluid content reference and vice versa. Configuring this attribute ensures that when you access a content reference configured for display mode the appropriate content reference (classic or fluid) is automatically displayed based on the factors.

Note: When you use the CN_DISPLAYMODE_N attribute, ensure that you configure bi-directional alternate content references. That is, a fluid content reference must be configured to point to a classic content reference and a classic content reference must be configured to point to a fluid content reference.

When you want to specify multiple conditions for a content reference, you can sequence the conditions by replacing N with a number.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

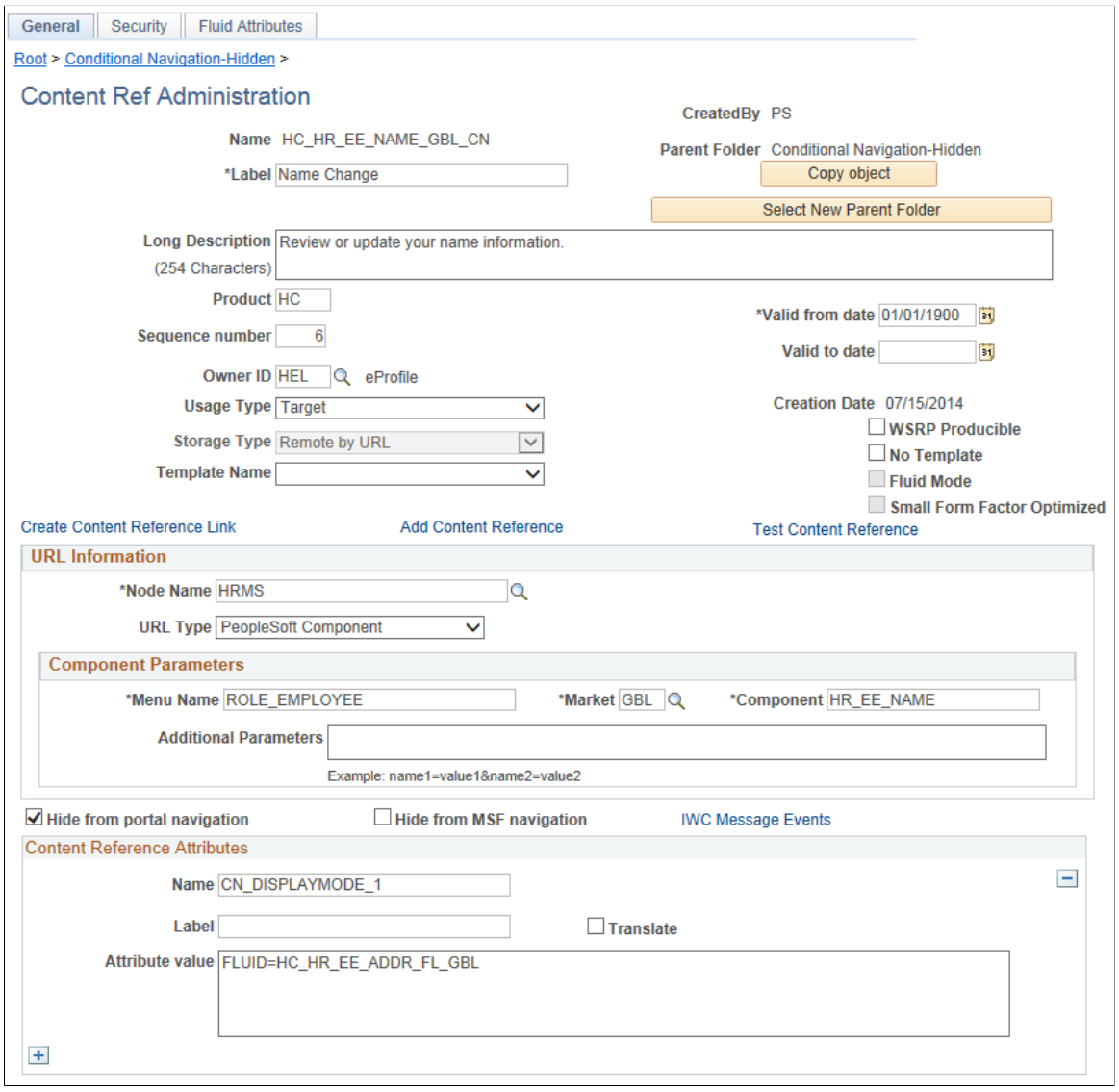
The following table shows the attribute name and attribute-value syntax:

Attribute Name	*Attribute Value*
CN_DISPLAYNODE_N	KEY1=CREF

Use the Content Ref Administration page (PeopleTools, Portal, Structure and Content and click the Edit link of a content reference) to configure an alternate content reference by adding the required attributes in the Content Reference Attributes section.

Image: Content Ref Administration - Content Reference Attributes section

This example illustrates the fields and controls on the Content Ref Administration - Content Reference Attributes section. You can find definitions for the fields and controls later on this page.



This table describes the fields that are associated with conditional navigation. For information on all the fields on the Content Ref Administration page, see Defining Content References.

Additional Parameters Ensure that the Additional Parameters field is blank.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: If you define any additional parameters, you will not be able to use the conditional navigation feature.

See Configuring an Alternate Content Reference.

Content Reference Attributes

Name	Enter an attribute, for example, CN_DISPLAYMODE_1.
Translate	Do *not* select the Translate check box. Conditional navigation does not support the Translate check box.
Attribute value	Enter the alternate content reference. For example, FLUID=HC_HR_EE_ADDR_FL_GBL.

Configuring an Alternate Content Reference

To configure an alternate content reference, do the following:

1. Ensure that the prerequisites for conditional navigation are configured on your system.
2. Open the content reference that you want to configure.

   Check whether additional parameters are defined for the content reference:

   - If the Additional Parameters field is not defined, proceed to step 3 and configure the alternate content reference.
   - If the Additional Parameters field is defined, you cannot use the existing content reference to configure the alternate content reference. You must create a new content reference with the same details as the existing content reference except the Additional Parameters field.

     Note: In cases where you may need to create new content references without the Additional Parameters field being set, PeopleSoft recommends that you create a new folder in Structure and Content at the root level to hold the alternate content references. The folder must have the hidden property set and you may need to use the Hide from portal navigation option if the content reference should not be available for navigation from the main menu.

     See Administering Content References.

3. In the Content Reference Attributes section, enter CN_DISPLAYMODE_1 in the Name field.
4. Optionally, enter a label for the attribute in the Label field.
5. Do not select the Translate check box.
6. Enter an alternate content reference in the Attribute value field.

   For example, FLUID=HC_HR_EE_ADDR_FL_GBL.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: You can add additional content reference attributes. When you add multiple content reference attributes, ensure that you enter sequence numbers for the attributes.

Sample Scenarios of Conditional Navigation

This topic discusses a few scenarios that involve configuring conditional navigation.

Case 1: Configuring Conditional Navigation in a Classic CREF that has an Alternate Fluid CREF

This sample scenario illustrates a case in which you are using the CN_DISPLAYMODE_N attribute to navigate from a classic content reference to the fluid equivalent content reference. You must configure the attribute in the classic and fluid content references as shown in the following example.

Example of CN_DISPLAYMODE_N attribute defined for classic content reference:

Attribute Name	*Attribute Value*
CN_DISPLAYMODE_1	FLUID=HC_HR_EE_ADDR_FL_GBL

Example of CN_DISPLAYMODE_N attribute defined for fluid content reference:

Attribute Name	*Attribute Value*
CN_DISPLAYMODE_1	CLASSIC=HC_HR_EE_ADDR_GBL

Case 2: Accessing Both Classic CREF as well as Fluid CREF from Menu Navigation

This sample scenario illustrates a case in which you want to navigate to both the classic content reference and the fluid content reference from the menu navigation. In conditional navigation, if you have a classic Cref and you configure it to point to a fluid equivalent, the classic CREF is converted to the fluid equivalent in all places including menu navigation. Therefore, menu navigation will no longer display the classic CREF. In this case, if you want to display both the classic and the fluid equivalent CREFs in menu navigation, PeopleSoft recommends that you create a copy of the classic content reference, but you must not configure conditional navigation on this new CREF.

Case 3: Configuring Conditional Navigation in a Classic CREF That has an Alternate Mobile Application Platform (MAP) Page

This sample scenario illustrates a case in which you have a MAP page for the approval process. For example, the MAP CREF name is HC_APPROVAL_PROMOTION_MAP. You want to configure this MAP CREF as an alternate content reference for a classic content reference. To implement this case, you should complete these steps:

Note: In the decision matrix, MAP pages are treated as fluid pages.

1. Open the classic approval content reference.
2. Configure the classic content reference with the MAP CREF as the alternate content reference.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Attribute Name	*Attribute Value*
CN_DISPLAYMODE_1	FLUID=HC_APPROVAL_PROMOTION_MAP

After configuring conditional navigation, when a user clicks the classic approval page, the MAP approval page will be displayed instead of the classic approval page.

Related Links

Mobile Application Platform

Conditional Navigation Decision Matrix

The following decision matrix is for the classic mode versus fluid mode:

PIA Mode	*Input CREF*	*Alternate CREF Configuration*	*Resolved CREF*
Classic	*Classic*	Fluid	*Classic*
Classic	Classic	*Classic*	*Classic*
Classic	Fluid	Fluid	Suppress*
Classic	Fluid	*Classic*	*Classic*
Classic	Fluid	Not configured	Suppress*
Classic	*Classic*	Not configured	*Classic*
Fluid	Classic	*Fluid*	*Fluid*
Fluid	Classic	*Classic*	*Classic*
Fluid	Fluid	*Fluid*	*Fluid*
Fluid	*Fluid*	Classic	*Fluid*
Fluid	*Fluid*	Not configured	*Fluid*
Fluid	*Classic*	Not configured	*Classic*

Consider the following regarding the decision matrix:

- Suppress* is applicable only when invoked by using the ShouldSuppressCREF PeopleCode API.
- No suppression will be done by the runtime/pre processing of conditional navigation logic.
- Suppression is implemented by PeopleTools in Drop Down Menu and Menu Pagelet of the home page and Related Content Framework.
- In case multiple content references resolve to the same alternate content reference, duplicate links are generated, but the duplicate links within the context of a folder will be removed and a user will see only one alternate content reference.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- MAP pages are treated as fluid pages.

Related Links

"ShouldSuppressCREF" (PeopleTools 8.54: PeopleCode Language Reference)

PeopleCode APIs that Support Conditional Navigation

The following PeopleCode APIs support conditional navigation:

Note: Application logic that uses these APIs for URL generation will automatically support conditional navigation and no change in the application logic is required. However, if the URL is created by other means, conditional navigation may not be supported.

- AbsolutePortalURL
- AbsoluteContentURL
- Transfer
- FindCREFByURL
- FindCREFByName
- getabsolutecontenturl
- GenerateComponentContentRelURL
- GenerateComponentContentURL
- GenerateComponentPortalRelURL
- GenerateComponentPortalURL
- GenerateComponentRelativeURL
- FindCRefLinkByName
- ViewURL
- GetURL
- GenerateHomepagePortalURL
- GenerateHomepageRelativeURL

Related Links

"OverrideCNAVDisplayMode" (PeopleTools 8.54: PeopleCode Language Reference)

Additional Content Reference Attributes in Conditional Navigation

In addition to the CN_DISPLAYMODE_N attribute, PeopleSoft supports the following content reference attributes that can be used to configure alternate content references.

You can configure multiple attributes for a content reference, and you specify the order of evaluating attributes by adding a sequence number. For example, in a content reference, you could use CN_DISPLAYMODE_1, CNTOOLSREL_2 and so on.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

CN_TOOLSREL_N

Applications may provide different content references for the same functionality based on the features supported by a PeopleTools version. For example, applications can provide a Change My Password content reference for 8.53 and another content reference for 8.54.

This table describes the attribute and the attribute value syntax:

Attribute Name	*Attribute Value*	*Recommendation*
CN_TOOLSREL_N	KEY1=CREF1[,KEY2=CREF2]*	KEY value should match the output of %TOOLSREL% meta variable. However, if the %TOOLSREL% meta variable returns, for example, 8.54.09, the KEY value of 8.54 will be valid because internally the system recognizes this condition.

Using the example of two implementations of Change My Password where CREF1 points to the 8.53 implementation and CREF2 points to the 8.54 implementation, the CN_TOOLSREL content reference attribute can be configured with the following values:

Attribute Name	*Attribute Value*
CN_TOOLSREL_1	8.53=CREF1, 8.54=CREF2

CN_NAVGROUP

Conditional navigation can also be configured for a multi-instance setup in PeopleSoft Interaction Hub. PeopleSoft enables you to configure multiple instances of the same application. For example, you can create multiple instances of the HRMS application and direct users to the appropriate instance to complete a task.

When you run the AE program for multi-instance, any alternate content reference configuration is automatically copied to the multiple instances.

For more information on the PeopleTools multi-instance feature, see the product documentation for *PeopleSoft Interaction Hub: Portal and Site Administration*, Understanding Multiple Application Instances in Unified Navigation.

User-Defined Content Reference Attribute

In addition to the delivered content reference attributes, PeopleSoft allows you to create user-defined conditional navigation attributes using the system meta variables.

Note: Only the system meta variables must be used when you want to configure a user-defined attribute, but *not* all meta variables can be used.

This table describes the attribute and the attribute value syntax:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Attribute Name	*Attribute Value*	*Recommendation*
CN_METAVAR1[_METAVAR2]*_1	KEY1=CREF1[,KEY2=CREF2]*	At runtime, the resolved Attribute name should match one of the configured values. The Attribute Values should exactly match the output of the system variable resolved Attribute Name.

When you create a conditional navigation attribute using a meta variable, you need to follow these guidelines for the attribute syntax:

- The attribute name must begin with CN_ as a prefix.
- Use a meta variable as a condition that needs to be satisfied, for example, add a system meta variable such as BROWSER.
- [] is optional.
- Use "_" as a separator when you want to include multiple system variables. Internally, the system converts the "_" separator to "%" at runtime.
- * : You can repeat the meta variable multiple times.
- Replace N with a number that is used for sequencing when multiple content reference attributes are used. This value is *required*.

This table shows sample values for a user-defined attribute. In this example, the alternate content reference will be resolved only when the browser is Firefox and the version is 32. For all other browsers and for all Firefox versions except 32, the original content reference will be displayed.

Attribute Name	*Attribute Value*
CN_BROWSER_BROWSERVERSION_1	FIREFOX32.0=PT_PSPMUMTESTINFO_GBL

The following meta variables must not be used to configure an alternate content reference:

- COMPONENT
- PAGE
- RENDERINGMODE
- MENU
- LABELTAG
- FORMNAME
- SUBMITSCRIPTNAME
- USERID
- APPSERVER

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- COPYRIGHT
- DIRECTION
- IMAGE
- BP
- BV
- BB
- JAVASCRIPT
- MESSAGE
- HTMLCONTENT
- URL
- BIND
- ENCODE
- COLS
- DEBUG
- COMPBUFFERSIZE
- LANGUAGEISO

Overriding Conditional Navigation

In certain cases, you may want to override conditional navigation that is configured for a content reference.

You can also override conditional navigation during pre processing and during post processing. The following table describes pre processing and post processing in conditional navigation:

Pre processing	Pre processing refers to the execution of code at the time of page or URL generation at the application server layer. That is, the factors required for conditional navigation and the conditions specified for alternate content references are evaluated at the time of page or URL generation.
Post processing	Post processing refers to the evaluation of factors and conditions for a given content reference at the web server layer. For example, when a user receives an email notification and clicks on the link, the factors and conditions are evaluated at the web server layer to determine whether a classic page or a fluid page should be displayed.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Overriding Conditional Navigation at Component Level

Conditional navigation must not be applicable for configuration pages that are used by administrators, for example, Structure and Content, Tree Manager, Query Drill Down. You can override conditional navigation at component level by using the OverrideCNavDispalyMode() PeopleCode API.

```
OverrideCNavDisplayMode(); //default is true
```

Note: This Peoplecode must be used as the first executable line on the Component PreBuild event.

Overriding Conditional Navigation During Pre Processing

You can use the OverrideCNavDisplayMode() PeopleCode API to suppress the conditional navigation display mode condition. The following code snippet illustrates how you can override conditional navigation for a particular code segment rather than for a component.

In the following code snippet, conditional navigation is not executed while pre processing the links:

```
OverrideCNavDisplayMode(True);
&URL = GenerateComponentPortalURL(%Portal,%Node,MenuName. "ROLE_MANAGER", "GBL", Co⇒
mponent."HR_PROMOTE_APPR","","");
OverrideCNavDisplayMode(False);
SendEmail(&URL);
```

Note: If you do not invoke the OverrideCNavDisplayMode(False) statement, conditional navigation will not be executed for the request session.

Overriding Conditional Navigation During Pre Processing and Post Processing

In cases where you do not want to execute conditional navigation either in pre processing or in post processing, you can use the SKIPCNAV parameter in conjunction with the generated URL.

In the following code snippet, conditional navigation is overridden both in pre processing and in post processing:

```
OverrideCNavDisplayMode(True);
&URL = GenerateComponentPortalURL(%Portal,%Node,MenuName. "ROLE_MANAGER", "GBL", Co⇒
mponent."HR_PROMOTE_APPR","","");
OverrideCNavDisplayMode(False);
&URL = &URL | "?Action=U&BUSINESS_UNIT=US100&REQ_ID=0000000067";
&URL = &URL | "&SKIPCNAV=1";
SendEmail(&URL);
```

In this example, the SKIPCNAV parameter is added as part of the URL, so even if the user clicks on the link the SKIPCNAV parameter will restrict the conditional navigation condition from being executed.

Related Links

"OverrideCNAVDisplayMode" (PeopleTools 8.54: PeopleCode Language Reference)

Working with Other PeopleTools Features in Conditional Navigation

This topic describes how search results, related content, and content from remote folders are handled in the conditional navigation feature.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Working with Search Results in Conditional Navigation

The conditional navigation feature takes into consideration the links that are displayed when a user uses global search. If conditional navigation is configured, the content references displayed on clicking the links on the Search Results page would be as per the conditional navigation guidelines on the search feature.

Consider the following scenario to understand how conditional navigation is implemented for classic display mode and fluid display mode.

- In Classic A content reference, Fluid A is configured as an alternate content reference.

  Classic A is indexed, but not Fluid A.

- Classic B content reference is not configured for any alternate content reference.

  Classic B is indexed.

- Fluid C content reference is not configured for any alternate content reference.

  Fluid C is indexed.

For example, a user searches on the word “Employee” and assume that the search hits are on Classic A, Classic B, and Fluid C.

User Searches in Fluid Display Mode on Any Device

User sees three search results: Fluid A, Classic B, and Fluid C.

Fluid A content reference is displayed rather than Classic A because the user is using fluid display mode.

User Searches in Classic Display Mode on Any Device

Users sees two search results: Classic A and Classic B.

Classic A content reference is displayed rather than Fluid A because the user is using classic display mode.

Working with Related Content in Conditional Navigation

The conditional navigation feature supports the related content features.

Authorization Service

In a system that is not configured for conditional navigation, the related content framework checks for authorization at the level of the input content reference, but in a system that is configured for conditional navigation, the authorization is checked at the level of resolved content reference.

Note: In a system configured for conditional navigation, you must invoke the authorization service for alternate content references.

See "Understanding Developing and Invoking the Security Authorization Service" (PeopleTools 8.54: Security Administration) for invoking authorization service.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Suppressing Duplicate Related Actions

Conditional navigation supports the mapping of multiple content references to a single content reference. In such a scenario, related actions configured for these content references will appear multiple times because of the mapping to a single content reference. In conditional navigation, duplicate related actions are suppressed within the context of a folder.

For example, if related actions are configured for three classic content references — ClassicCREF1, ClassicCREF2, and ClassicCREF3 — and if these three content references are mapped to a single fluid content reference (NUICREF1), the related actions menu would display NUICREF1, NUICREF1, NUICREF1. In fluid display mode, the duplicate related actions are suppressed and the related actions menu will display NUICREF1 only once in the context of a folder.

Passing Service URL Parameters to Alternate Content Reference

When you map multiple classic content references with related actions configured to a single fluid content reference, the Service URL Parameters of the first content reference are passed to the mapped fluid content reference. For example, three classic content references — ClassicCREF1, ClassicCREF2, and ClassicCREF3 — with configured related actions are mapped to a single fluid content reference (NUICREF1), by default, the Service URL Parameters of ClassicCREF1 are passed to NUICREF1.

If you want to pass the Service URL Parameters of ClassicCREF2 or ClassicCREF3 to NUICREF1, complete these steps:

1. Open the service definition that you want to modify, for example, EF_HR_EE_MAR_STATUS. (PeopleTools, Portal, Related Content Service, Define Related Content Service)

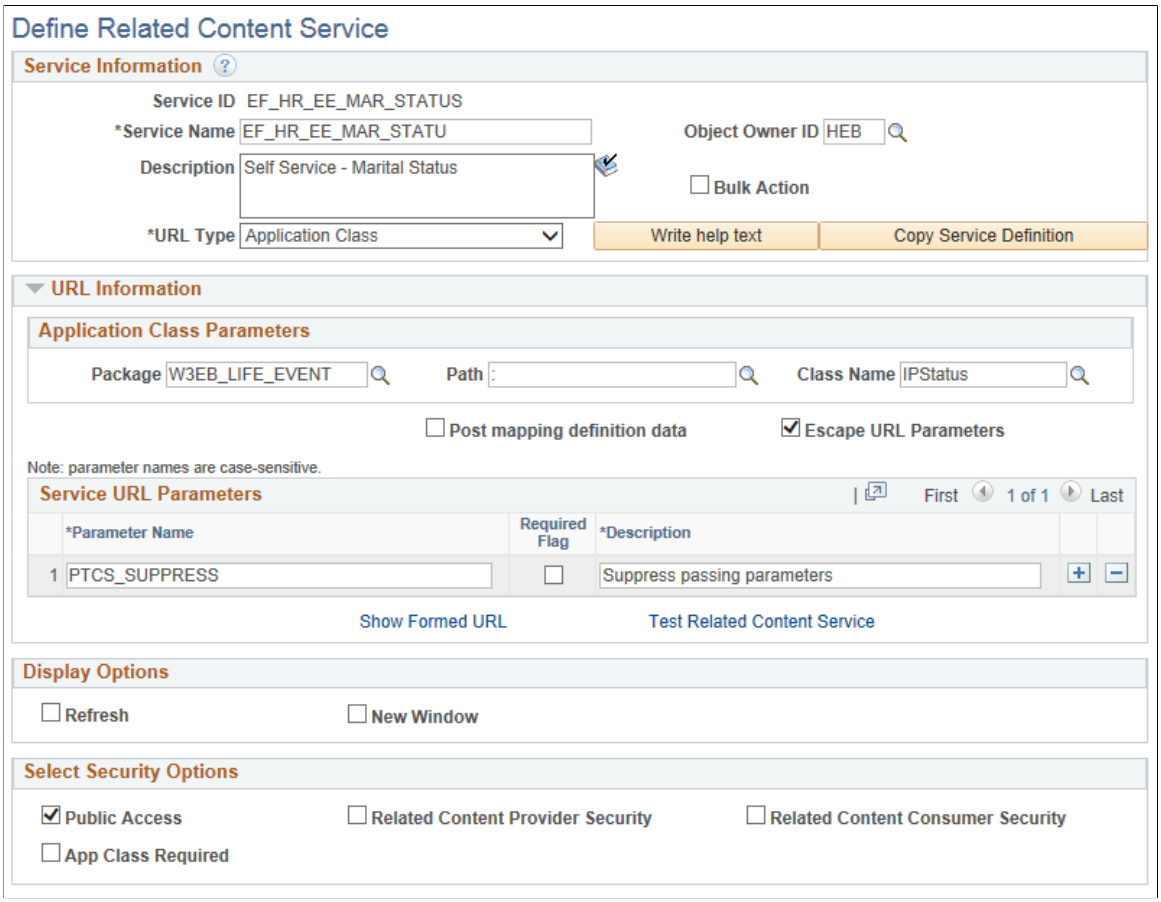
2. In the Service URL Parameters group box, enter *PTCS_SUPPRESS* in the Parameter Name field.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

3. Enter a description.

Image: Define Related Content Service page

This example illustrates adding a service URL parameter.



4. Save the service definition.

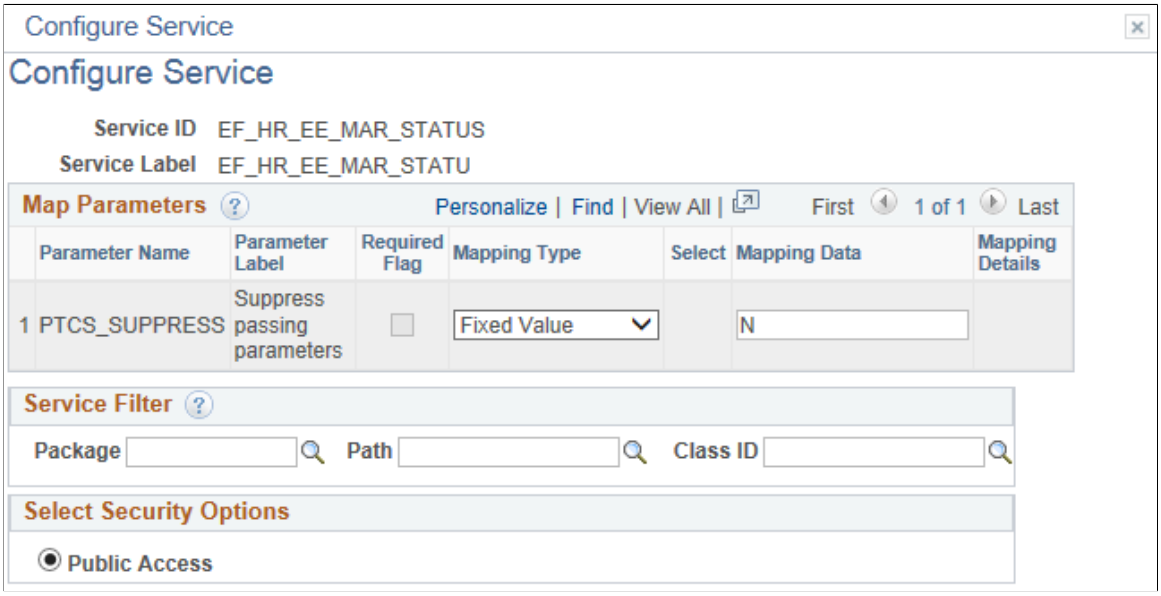
5. Edit the service configuration of the modified service definition. (PeopleTools, Portal, Related Content Service, Manage Related Content Service)

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

6. On the Configure Service page, for the PTCS_SUPPRESS parameter, select *Fixed Value* in the Mapping Type column, and enter *N* in the Mapping Data column.

   Image: Configure Service page

   This example illustrates configuring the PTCS_SUPPRESS service.



7. Save the configuration.

Related Links

Defining Related Content Services
Managing Related Content
Configuring Related Content, Related Actions, and Layout

Accessing Content from Remote Folders in Conditional Navigation

In a system configured for unified navigation, accessing content from remote folders is achieved through an Integration Broker request. Conditional navigation does not support an Integration Broker request. Therefore, when you want to use conditional navigation in a system that is configured for unified navigation, which includes displaying content references from remote folders, an application developer must modify IB SyncRequest to access content from remote folders as shown in the example.

For example, consider a scenario where you want to make an IB SyncRequest to the remote PeopleSoft application node to fetch the application registry content references.

To achieve this, complete the following:

1. Add a cookie with conditional navigation specific information as part of SyncRequest.

   The following code snippet illustrates the code with conditional navigation specific information:

```
&reqMsg = CreateMessage(Operation.PTUN_LANDING_PAGE, %IntBroker_Request);
&reqMsg.SetXmlDoc(&xmlDoc);
&reqMsg.IBInfo.ConnectorOverride = True;

/* Read Conditional Navigation related params */
Local number &formfactor = %Request.BrowserDeviceFormFactor;
Local number &fluidCapable = UIDisplayMode();
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
/*Error - Default back to CLASSIC*/
If (&fluidCapable = 0) Then
    &fluidCapable = 1; /*CLASSIC*/
End-If;

/*  Cookie name can be anything*/
&cookieStr = "PS_CONDNAV" | "=" | &fluidCapable | "," | &formfactor | "; ";
&cookieStr = RTrim(&cookieStr, "; ");
&reqMsg.IBInfo.IBConnectorInfo.Cookies = &cookieStr;
&resMsg = %IntBroker.SyncRequest(&reqMsg, &targetNode);
```

2. Read the cookie (PS_CONDNAV) on the content provider (remote system) system's handler application class code and set the conditional navigation context.

   The following code snippet illustrates the conditional navigation context:

```
Declare Function setCNProp PeopleCode FUNCLIB_PTUN.FUNCLIB FieldFormula;

Local number &nDispMode = 0;
Local number &nDeviceType = 0;
Local string &cookies = &pRequest.IBInfo.IBConnectorInfo.Cookies;

/* Assuming only PS_CONDNAV cookie is present*/
If (&cookies <> "") Then
     /* remove the cookie name out */
     Local array of string &cookieData = Split(&cookies, "=");
     If (&cookieData.Len = 2) Then
            Local array of string &splitCookie = Split(&cookieData [2], ",");
           &nDispMode = Value(LTrim(&splitCookie [1]));
           &nDeviceType = Value(LTrim(&splitCookie [2]));
     End-If;
End-If;

/* Set the CN properties before calling the registry method*/
setCNProp(&nDispMode, &nDeviceType, True);

Local ApiObject &Portal = %Session.GetPortalRegistry();
Local ApiObject &CrefObject = &Portal.FindCrefByName(&CrefName [3]);

/* Reset the CN properties */
setCNProp(&nDispMode, &nDeviceType, False);
```

Note: Before executing FindCrefByName, call the setCNProp method to set the conditional-navigation context.

Troubleshooting Conditional Navigation

To ensure that you have configured conditional navigation correctly on your system, check the following:

- Verify web profile settings:
  - Verify whether web profile settings are set to the correct profile.
  - Verify whether the Disable Fluid Mode and Disable Fluid On Desktop check boxes have correct values and that the web server has been restarted.
- Verify whether attributes are correctly defined on the Content Ref Administration page for content references.
- Verify whether authentication permissions have been configured for alternate content references. If permissions are not configured, conditional navigation will not resolve to the expected content reference and will evaluate the next condition.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

These logs are available in application server logs.

- Verify cyclic references.

  You can configure alternate content references in such a way that it forms a loop. In these cases, conditional navigation identifies cyclic references, breaks the loop, and returns the original content reference.

- Verify whether the OverrideCNavDisplayMode() PeopleCode API is used.

- Verify whether the classic or fluid property for the content reference is selected.

  You may want to check this property because it is possible that the component is marked as fluid, but the content reference is not marked as fluid.

- Use the application server logging to trace conditional navigation. Set LogFence = 4 to trace conditional navigation.

  Logs specific to conditional navigation are prefixed with COND_NAV.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Sample conditional navigation log trace:

Image: Example of conditional navigation trace log

This example illustrates a sample conditional navigation trace log in application server logs.

```
COND_NAV:DisplayMode - Checking for CREF[HC_GP_ABS_MGRSS_BAL_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HGA_SS_BAL_FLU_MSS_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HGA_SS_BAL_FLU_MSS_GBL]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HGA_SS_BAL_FLU_MSS_GBL]

COND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.GP_ABS_MGRSS_BAL.GBL?
PAGE=GP_ABS_SS_BALANCES]
COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/HGA_MANAGER_FL.HGA_SS_BAL_FLU.GBL?
MGR=Y&PAGE=GP_ABS_SS_BALANCES]

COND_NAV:DisplayMode - Checking for CREF[HC_GP_ABS_MGRSS_HIST_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HGA_SS_REQHIST_FLU_MSS_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HGA_SS_REQHIST_FLU_MSS_GBL]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HGA_SS_REQHIST_FLU_MSS_GBL]

COND_NAV:DisplayMode - Checking for CREF[PTSF_GBLSRCH_FLUID_GBL]
COND_NAV:DisplayMode - Alternate CREF[PTSF_GLOBAL_SEARCH_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use Original CREF[PTSF_GBLSRCH_FLUID_GBL]


COND_NAV:DisplayMode - Alternate CREF[PTSF_GLOBAL_SEARCH_GBL] found for [CN_DISPLACOND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.GP_ABS_MGRSS_HIST.GBL?
PAGE=GP_ABS_SS_REQHIST]
COND_NAV:INFO Decision Matrix result - Use Original CREF[PTSF_GBLSRCH_FLUID_GBL]COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/HGA_MANAGER_FL.HGA_SS_REQHIST_FLU.GBL?
MGR=Y&PAGE=GP_ABS_SS_REQHIST]
COND_NAV:DisplayMode - Checking for CREF[HC_GP_ABS_MGRSS_REQ_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HGA_SS_REQ_FLU_MSS_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HGA_SS_REQ_FLU_MSS_GBL]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HGA_SS_REQ_FLU_MSS_GBL]
COND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.GP_ABS_MGRSS_REQ.GBL?
PAGE=GP_ABS_SS_REQUEST]
COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/HGA_MANAGER_FL.HGA_SS_REQ_FLU.GBL?
MGR=Y&PAGE=GP_ABS_SS_REQUEST]
COND_NAV:DisplayMode - Checking for CREF[HC_HR_MGR_FULL_PART_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HR_MSS_CT_FL_GBL_1] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HR_MSS_CT_FL_GBL_1]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HR_MSS_CT_FL_GBL_1]
COND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.HR_MGR_FULL_PART.GBL?
PAGE=HR_MGR_FULL_PART]
COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/EL_MANAGER_FL.HR_MSS_CT_FL.GBL?
TRANS_NAME=HR_FULL_PART_STATUS&PAGE=HR_MGR_FULL_PART]
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Part III

Working with Homepages, Dashboard Pages, and Pagelets

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 12

Administering Homepages and Dashboard Pages

Understanding Homepages and Dashboard Pages

Note: In these topics, the terms *homepage* and *tabbed homepage* are used interchangeably. While Oracle regards the term tabbed homepage as the more accurate term, in this document the terms are equivalent. In addition, the terms *dashboard page* and *dashboard* are equivalent.

Homepages, or tabbed homepages, and dashboard pages are PeopleSoft pages that aggregate and display pagelets that share some common or similar purpose. Homepages and dashboard pages provide quick access to useful information by presenting concise but feature-rich pagelets to the user. These pagelets can vary in function and complexity, enabling you to see not only an overview, but to drill into your data to access more granular and detailed information. Pagelets can enable you to perform routine tasks or provide you with a high-level overview of aspects of your company that are pertinent to you. In addition, pagelets can interact with each other by publishing and subscribing to messages in a process called interwindow communication that refreshes pagelet content based on any passed values.

Note: The homepages and dashboards available to you differs based on the products that you license.

Homepages

Note these points about homepages:

1. Homepages appear immediately after users sign in to PeopleSoft applications or when the user clicks the Home link.
2. PeopleSoft applications can display one or more homepage tabs.
3. If you create multiple homepages, each displays a tab at the top to provide access to each homepage.
4. If you enable personalization, users can personalize homepages. The system stores each user's homepage personalizations in the PSPRUHDEFN (which is updated when a user personalizes the homepage), PSPRUHTAB, and PSPRUHTABPGLT tables.

You manage homepages within the Portal Objects folder in the portal registry, which you access by using the Structure and Content component. Access to this component gives any user unrestricted access to the entire portal registry. Rarely should anyone other than the portal administrator have access to Structure and Content.

Dashboards

Note these points about dashboard pages:

- Dashboard pages are a type of homepage that have been designated as *operational dashboards* through a content reference attribute.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Dashboards do not appear automatically after sign in; instead, you access a dashboard page by clicking a content reference link, which you can configure to appear anywhere in the menu structure.
- PeopleSoft applications can have multiple dashboards.
- If you enable personalization, users can personalize dashboard pages. Similar to homepages, the system stores each user's dashboard page personalizations in the PSPRUHDEFN (which is updated when a user personalizes the dashboard), PSPRUHTAB, and PSPRUHTABPGLT tables.
- WorkCenter dashboards are a special class of dashboards that can be deployed as target content in a WorkCenter.

Unlike homepages, to create and manage dashboards, you do *not* need access to the Structure and Content component. You can use the Manage Dashboard Pages component. No access to the portal registry is necessary. By eliminating the need to access the portal registry to manage dashboard pages, your PeopleSoft application offers increased flexibility of administration and decreased chance of down time due to portal registry mismanagement errors.

Note: When you use the Manage Dashboard Pages component to create new dashboard pages, the system automatically registers the dashboard as a homepage in the portal, places the content reference in the Tab folder, and selects the Operational Dashboard check box. No functionality has been removed from the Structure and Content pages; you can also use them to create and manage homepages that you specify as operational dashboard pages.

Related Links

"Understanding PeopleSoft Browser-Based Applications" (PeopleTools 8.54: Applications User's Guide)

"Personalizing Your Homepage" (PeopleTools 8.54: Applications User's Guide)

Creating and Managing WorkCenter Dashboards

Understanding Interwindow Communication

Managing Homepages

This section discusses how to:

- Create homepages.
- Configure homepages.
- Set homepage permissions.
- Select the pagelets that appear on homepages.
- Arrange pagelets on homepages.
- Rearrange the homepage tab sequence.
- Troubleshoot homepages, dashboards, and WorkCenter pages.

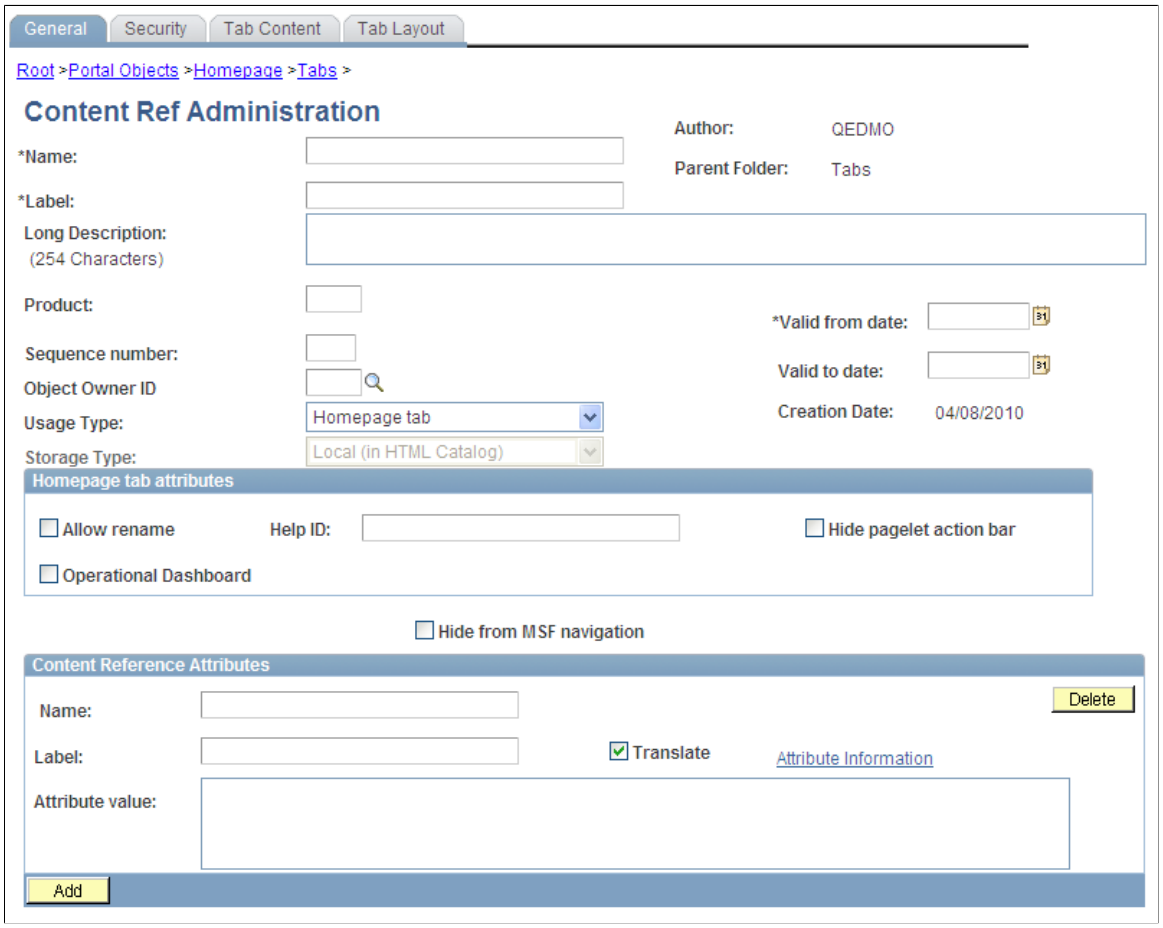
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Registering Homepages in the Portal Registry

Access the Content Ref Administration page. (Select PeopleTools, Portal, Structure and Content. Click the Portal Objects link. Click the Homepage link. Click the Tabs link. Click the Add Content Reference link).

Image: Content Ref Administration page

This example illustrates the fields and controls on the Content Ref Administration page. You can find definitions for the fields and controls later on this page.



To create a homepage entry in the portal registry:

Note: You must create the homepage or dashboard in the Tabs folder.

---

1. Enter a name and label for the homepage or dashboard.

   Note: The label value appears as the homepage tab, dashboard name and content reference link label.

   ---

2. In the Usage Type field, select *Homepage tab.*

   After you select *Homepage tab,* the page refreshes and two additional pages, Tab Content and Tab Layout, are displayed.

3. In the Valid from date field, enter the date that you want the tab to first appear for your users. Optionally, enter a valid to date to remove the tab from use.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. If you want the homepage to be a dashboard page, select the Operational Dashboard check box in the Homepage tab attributes group box.
5. Set the homepage permissions.

   See Setting Homepage Permissions.
6. Define the homepage content and layout.

   See Selecting the Pagelets That Appear on Homepages.

   See Arranging Pagelets on Homepages.
7. Click the Save button.
8. In addition, if you selected the Operational Dashboard check box, create a content reference link to the dashboard in the navigational hierarchy.

   See Defining Content Reference Links.

This table describes the fields that appear on the Content Ref Administration page.

Label	Enter the label that the user sees on the homepage tab, as the dashboard page title, and as the content reference link in the navigation.
Name	Enter a unique name for the tab. The value that you specify in the Name field can consist of up to 50 letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.
Sequence number	Enter a value to designate the absolute position of the homepage when multiple homepages exist for users. The sequence number determines the order in which the homepage tabs appear. The tabs appear with the lowest sequence number first. If any numbers are identical, then the order is alphanumeric. No value is equivalent to specifying *0*. You can change the sequence only one item at a time, and each time you do so, you must save your changes to see the result. **Important!** Therefore, the tab with the lowest sequence number becomes the default homepage.
Usage Type	Select *Homepage tab* to create a tabbed homepage or dashboard. When you select *Homepage tab,* two additional pages appear in the component: the Tab Content and Tab Layout pages.
Valid from date	Enter the date that the tab becomes visible to the end user. You can enter a date to remove the tab from display.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Homepage Tab Attributes

Use the Homepage tab attributes group box of the Content Ref Administration page to enable users to rename the homepage tab label, as well as associate a help topic with the homepage tab, hide the pagelet action bar, and designate a homepage as an operational dashboard.

Allow rename

Select to enable the application user to change the homepage tab label.

Help ID

Enter the help ID for the homepage tab, dashboard, or pagelet. When users click the Help link, the PeopleSoft system opens a new window displaying online help documentation for this item.

Note: You must also configure a help URL on the Web Profile Configuration - General page.

Hide pagelet action bar

Select to hide the pagelet title bars on all pagelets on the homepage tab. This option overrides all properties for pagelets used on this tab.

Operational Dashboard

Select to use the homepage as a dashboard page, which hides the homepage so that it does not appear when the users signs in.

If you select this check box, you must create a content reference link to the dashboard, which you position in the menu navigation as your business rules require.

If you use the Dashboard Administration pages to create dashboards, then the system creates the homepage content reference, selects this check box, and creates the link automatically.

Related Links

Setting Content Reference Permissions

Managing Dashboard Pages

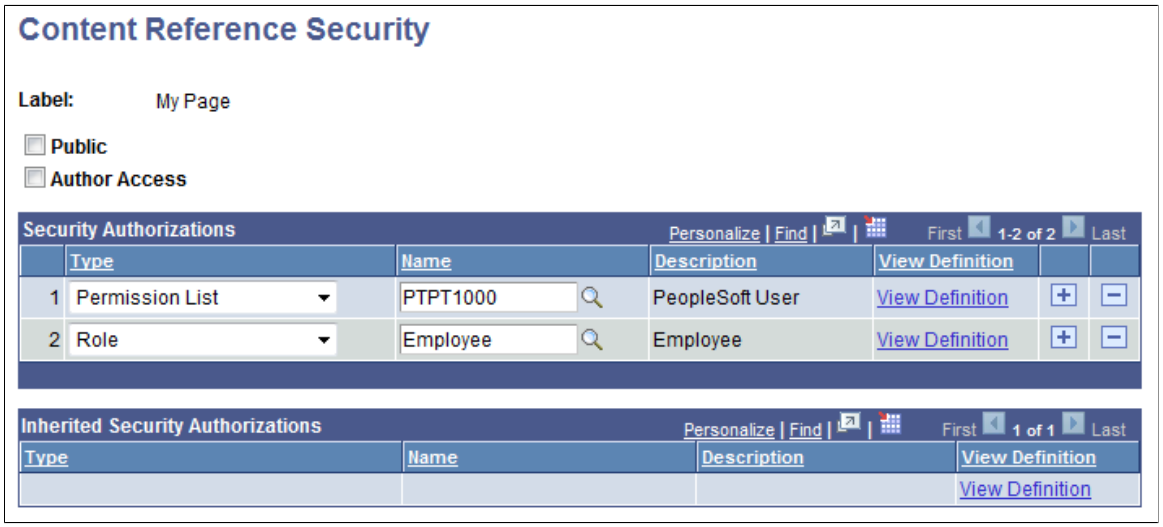
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Setting Homepage Permissions

Access the Content Reference Security page (Select PeopleTools, Portal, Structure and Content. Then click the Portal Objects link, the Homepage link, and the Tab link in that order. Click the Edit link for the homepage tab, then select the Security page).

Image: Content Reference Security page

This example illustrates the fields and controls on the Content Reference Security page. You can find definitions for the fields and controls later on this page.



Public Select to make the homepage or dashboard available to all users. When this check box is selected, all other fields except Label are hidden.

Author Access Select to indicate that you (the currently signed-in user and the author of the homepage) can access the tab in the future, regardless of how other permissions are set.

Security Authorizations

You can base homepage security on roles or permission lists. You can specify any combination of roles and permission lists that allow homepage access.

Type Select from these types:

Permission List: Enables access for users based on permission list membership. You then specify the permission lists that should have access to this folder.

Role: Enables access for users based on role membership. You then specify the roles that should have access to this folder.

Name Select the permission list or role that you want have access to this homepage.

View Definition Click to access the security administration component for the permission list or role specified on the current row.

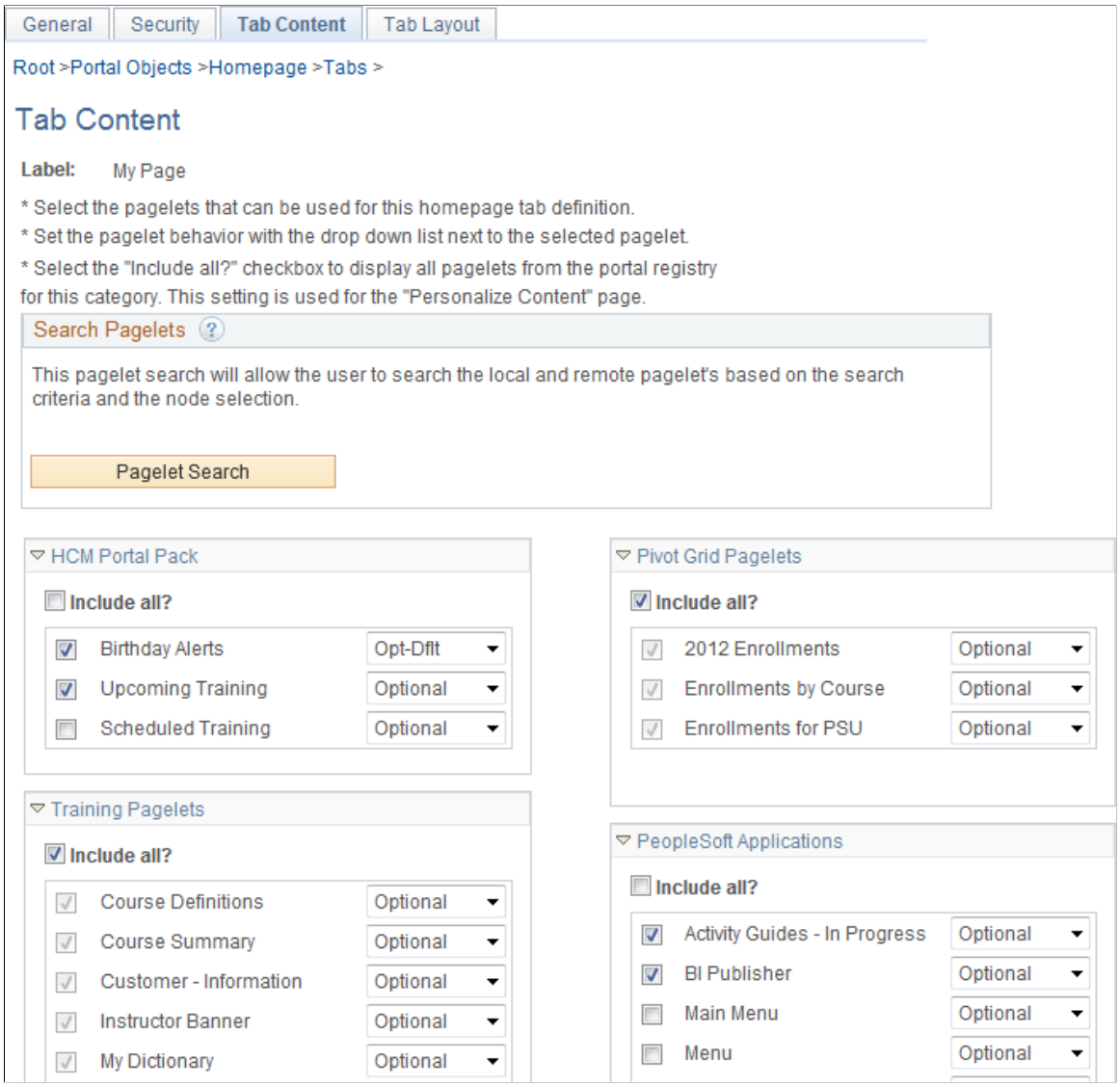
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting the Pagelets That Appear on Homepages

Access the Tab Content page. (Select the Tab Content page.)

Image: Tab Content page

This example illustrates the fields and controls on the Tab Content page. You can find definitions for the fields and controls later on this page.



To select the pagelets that can be displayed on this homepage tab:

1. Expand any collapsed group boxes and then select the check box to make the pagelet available for this homepage tab.

   Select the Include All? check box to select all pagelets in the category.

   Note: The user must have security access to the pagelet component for the pagelet to be properly displayed on the homepage.

   ---

2. In addition, you can click the Pagelet Search button and select a pagelet from the Pagelet Search page to add it to the homepage tab definition.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Searching for and Importing Pagelets for more information on searching for pagelets on local and remote nodes.

3. Use the drop-down list box to select the pagelet behavior. Values are:

***Opt-Dflt* (optional default)**	The pagelet is visible to all users by default for users that have permissions to the pagelet; users can move or delete the pagelet.
Optional	The pagelet is available to all users for users that have permissions to the pagelet, but it is not visible by default; users can add, move, or delete the pagelet.
***Req-Fix* (required fixed)**	The pagelet is visible to all users by default for users that have permissions to the pagelet; however, users cannot move or delete the pagelet.
Required	The pagelet is visible to all users by default for users that have permissions to the pagelet; users can move the pagelet, however, they cannot delete it.

4. Click the Save button.

Important! When you select content for inclusion on any PeopleSoft homepage, only one pagelet or portlet should use the calendar prompt. If more than one pagelet or portlet per homepage uses the calendar prompt, unexpected results might occur.

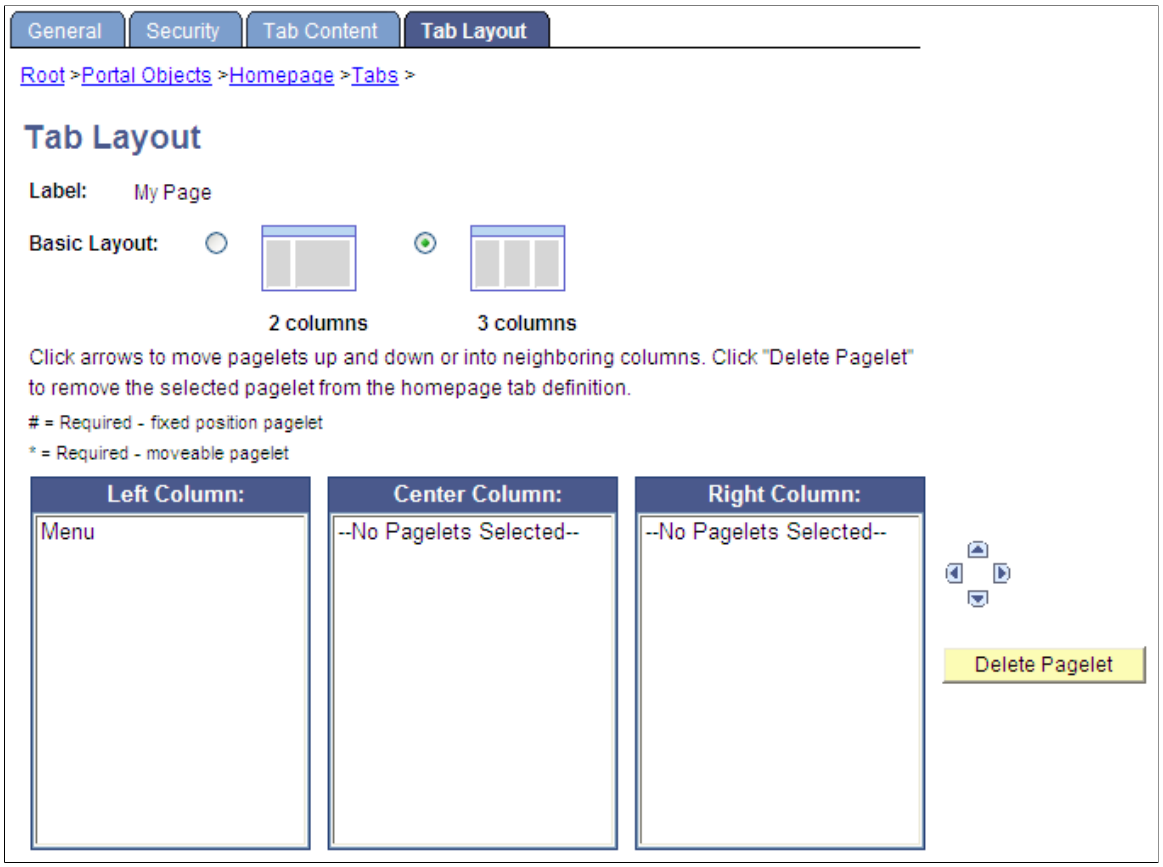
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Arranging Pagelets on Homepages

Access the Tab Layout page. (Select the Tab Layout tab.)

Image: Tab Layout page

This example illustrates the fields and controls on the Tab Layout page. You can find definitions for the fields and controls later on this page.



Label

Displays the label information from the Content Ref Administration page.

Basic Layout

Select to display the pagelets in either two or three columns on the homepage.

To specify the three-column layout, be sure that at least one pagelet does not have the *Req-Fix* behavior option selected on the Tab Content page.

Columns

Displays the pagelets selected on the Tab Content page in the column sections.

If the basic layout is two columns, the pagelets are divided into left and right columns. If the basic layout is three columns, the pagelets are divided into left, center, and right columns.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: A pound sign (#) indicates a pagelet with the *Req-Fix* behavior option selected on the Tab Content page. An asterisk (*) indicates a pagelet with the *Required* behavior option selected on the Tab Content page.
For example, #Signon indicates that the Signon pagelet cannot be moved to a different location on the homepage.

Use the Move icons to position the pagelets. Highlight a pagelet, and then click the directional arrow buttons to move the selected pagelet above a pagelet, below a pagelet, to the next column to the right, or the next column to the left.

Delete Pagelet

Highlight a pagelet and then click to delete it from the homepage.

To arrange the tab layout:

1. Select a basic layout option.
2. Select the pagelet name.
3. Click the directional arrow buttons to position the pagelets in the columns.
4. Click the Save button.

Note: As the portal administrator, you determine the amount of flexibility that you give users to change the layout of the pagelets. The settings that you select on both the Tab Content and Tab Layout pages determine the degree of flexibility. Grant the end user as much or as little flexibility as your organization demands.

Specifying the Homepage Tab Sequence

The sequence number determines the order in which the homepage tabs appear. The tabs appear with the lowest sequence number first. If any numbers are identical, then the order is alphanumeric. If no sequence numbers exist, the default is considered to be zero. For example, if no sequence numbers are entered, the default is zero, and the order is alphanumeric.

Important! Therefore, the tab with the lowest sequence number becomes the default homepage.

To rearrange the tab order:

1. Select PeopleTools, Portal, Structure and Content.
2. Click the Portal Objects link.
3. Click the Homepage link.
4. Click the Tabs link.
5. Click the Edit link for each tab that you want to reorder.
6. Enter a number in the Sequence field.

If caching is turned on for the portal, the tab order change does not take effect until the caching recycles or the web server is restarted.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Troubleshooting Homepages, Dashboards, and WorkCenter Pages

Under most circumstances, PeopleTools portal technology assembles homepages, dashboards, and WorkCenter pages by simultaneously retrieving content for each of the pagelets referenced on the page. As a result, the portal renders the page itself as soon as the user accesses it and immediately begins the deferred loading of individual pagelets.

Pagelet Loading

Image: Deferred pagelet loading on a homepage

Portal homepages, dashboards, and WorkCenter pages defer-load each pagelet individually after you access the page. Each pagelet appears as you see in this example, indicating that it's currently loading its data.

The following example illustrates a pagelet that is loading.

Modifying Default Pagelet Behaviors

In some cases, a pagelet may take awhile to load. Some reasons for this could be the URL for the pagelet is incorrect or has changed, DNS can't resolve the URL, the content server is down or unreachable, the content server is behind a firewall, or the content server is temporarily unavailable.

You can modify unavailable pagelets by specifying a time-out period during which the pagelet must be loaded. To specify a pagelet time-out, add the PSTIMEOUT attribute to the content reference for the pagelet in the portal registry.

See Configuring Pagelet Time-out Settings.

In other cases, when **document.write** or **document.writeln** commands are detected inside a pagelet, the whole page will be reloaded again and all pagelets are then loaded sequentially. In this case, to properly assemble the content of the pagelet in question and the whole page, none of the pagelets can be defer-loaded individually. To avoid reloading the page multiple times and to allow other pagelets to be defer-loaded, you must add a content reference attribute to the pagelet in question.

To specify that a specific pagelet with **document.write** or **document.writeln** commands be loaded in the non-default manner:

1. Select PeopleTools, Portal, Structure and Content, Portal Objects, and navigate to the Content Ref Administration page for the pagelet.
2. In the Content Reference Attributes region of the page, enter *PTPGLTDEFERLOAD* in the Name field.
3. Deselect the Translate option.
4. In the Attribute Value field, enter *FALSE*.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

5. Save the content reference.

Managing Dashboard Pages

This section discusses how to:

- Create and manage dashboard pages.
- Specify dashboard page properties.
- Specify the location of dashboard pages.
- Specify dashboard page security properties
- Specify the location of the dashboard page link.
- Specify the pagelets that appear on dashboard pages.
- Arrange the pagelets on a dashboard page.

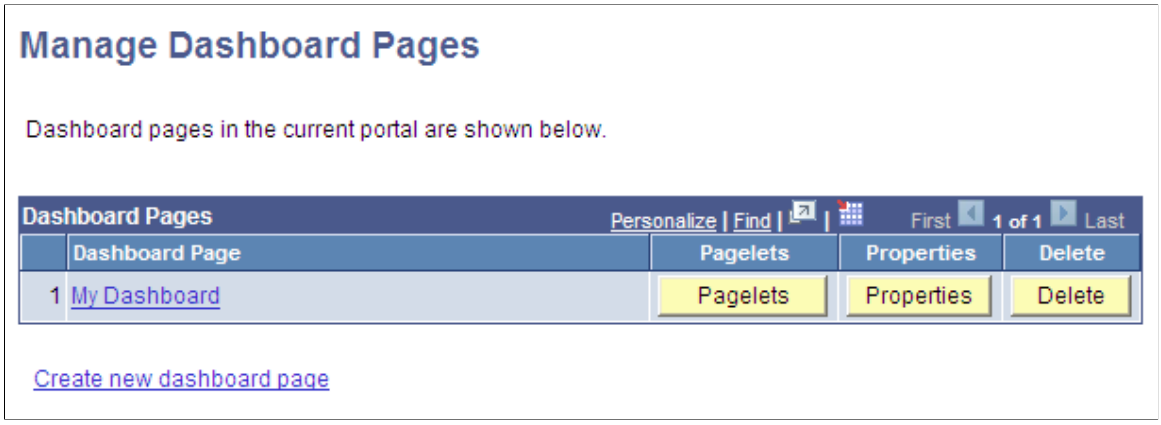
In addition, you can create and manage WorkCenter dashboards, which are maintained separately from other dashboards. See Creating and Managing WorkCenter Dashboards.

Creating and Managing Dashboard Pages

Access the Manage Dashboard Pages page (Select PeopleTools, Portal, Dashboard, Manage Dashboard Pages.)

Image: Manage Dashboard Pages page

This example illustrates the fields and controls on the Manage Dashboard Pages page. You can find definitions for the fields and controls later on this page.



Create new dashboard page	Click to access the Dashboard Page Administration page and configure the properties of new dashboard pages.
Dashboard Page	Displays an active link to the dashboard page. Click the link to access the dashboard page.
Pagelets	Click to configure the content and layout of the dashboard.
Properties	Click to access the Dashboard Page Administration page and specify dashboard navigation and security details.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

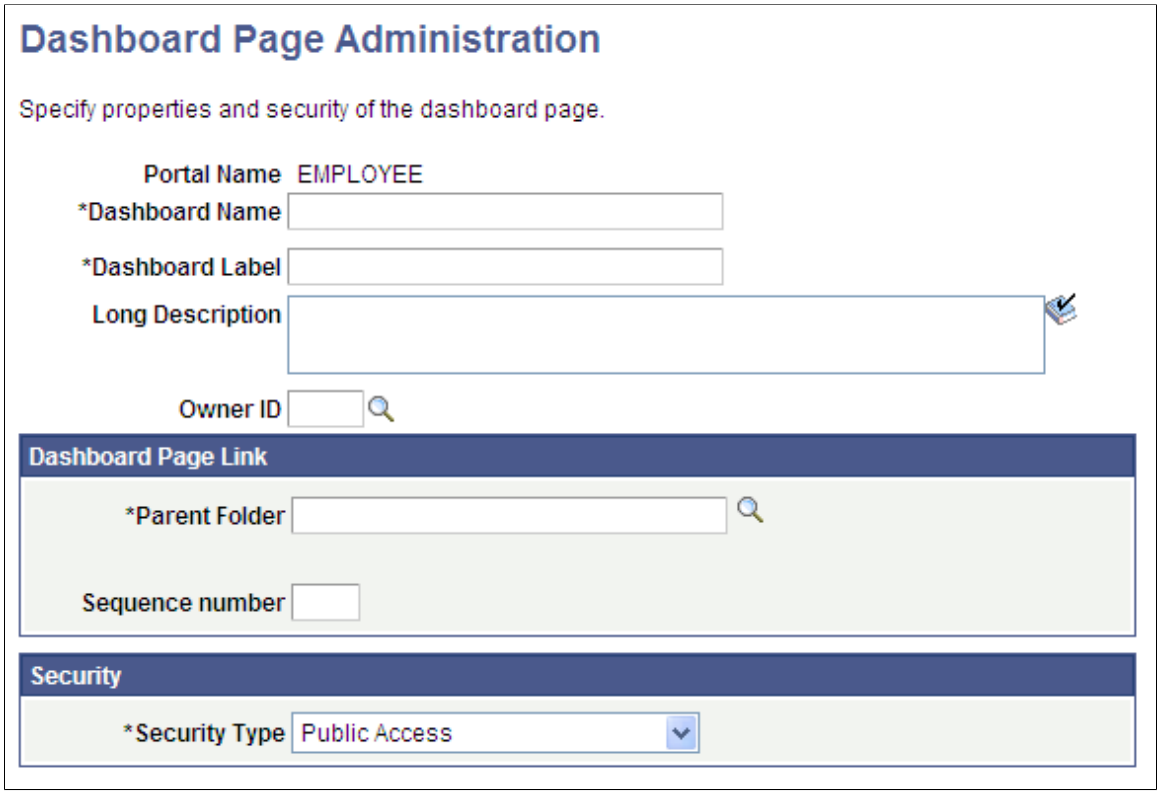
Delete Click to remove the dashboard content reference definition from the portal registry.

Specifying Dashboard Page Properties

Access the Dashboard Page Administration page (Click the Create new dashboard page link. Alternatively, click the Properties button).

Image: Dashboard Page Administration page

This example illustrates the fields and controls on the Dashboard Page Administration page. You can find definitions for the fields and controls later on this page.



Enter the following dashboard page properties:

Dashboard Name Enter a unique identifier for the dashboard. This field is required.

Dashboard Label Enter the label that appears in the navigation. This field is required.

Long Description Enter a description that appears as the tool tip text.

Owner ID Select the organization that owns the object.

Dashboard Page Link

Use these fields to specify the location of the dashboard page link in the menu navigation.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Parent Folder Enter the name of the navigation folder in which you want to place the dashboard page link. Click the look up button to access the Select the Dashboard Link Parent Folder page.

Sequence number Enter a number to specify the order of the dashboard link within the parent folder. The application sorts the items alphabetically if you omit a sequence number.

Specifying Dashboard Page Security Properties

Use the fields in the Security group box to specify whether access to the dashboard page is restricted by permission list and role or unrestricted and available to all users.

Security

Security Type Select from these values:

- *Public Access:* Select to allow all users to view and access the dashboard.
- *Select Security Access:* Select to specify PeopleTools Security access, which is based on permission lists and roles. When you select this option, the Selected Security grid appears.

Selected Security

Insert rows into the Selected Security grid to specify the permission lists and roles that provide access to the dashboard page. You can combine permission lists and roles by inserting multiple rows in the grid.

Click the Insert Row button (+) to add another permission list or role name.

Click the Delete Row button (-) to remove the permission list or role name.

Type For each row, select from these values:

- *Permission List:* Select to secure the dashboard link by permission list.
- *Role:* Select to secure the dashboard link by role.

Name Specify the permission list or role name that provides access to the dashboard page. Click the look up button to see a list of available names.

Save Click to save the configuration and return to the Manage Dashboard Pages page.

Cancel Click to discard any changes and return to the Manage Dashboard Pages page.

Related Links

"Understanding Permission Lists" (PeopleTools 8.54: Security Administration)

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

"Understanding Roles" (PeopleTools 8.54: Security Administration)

Specifying the Location of Dashboard Page Link

Access the Select the Dashboard Link Parent Folder page (Click the look up button for the Parent Folder field).

Image: Select the Dashboard Link Parent Folder page

This example illustrates the fields and controls on the Select the Dashboard Link Parent Folder page. You can find definitions for the fields and controls later on this page.

To select the parent folder of the dashboard page you must navigate the tree until you see the destination folder.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

(collapsed folder)	Click to expand the folder and view the subordinate menu items.
(expanded folder)	Click to collapse the folder and hide the subordinate menu items.
(folder name)	Click to select the folder as the parent of the dashboard page.
Cancel	Click to return to the previous page.

Specifying the Pagelets That Appear on Dashboard Pages

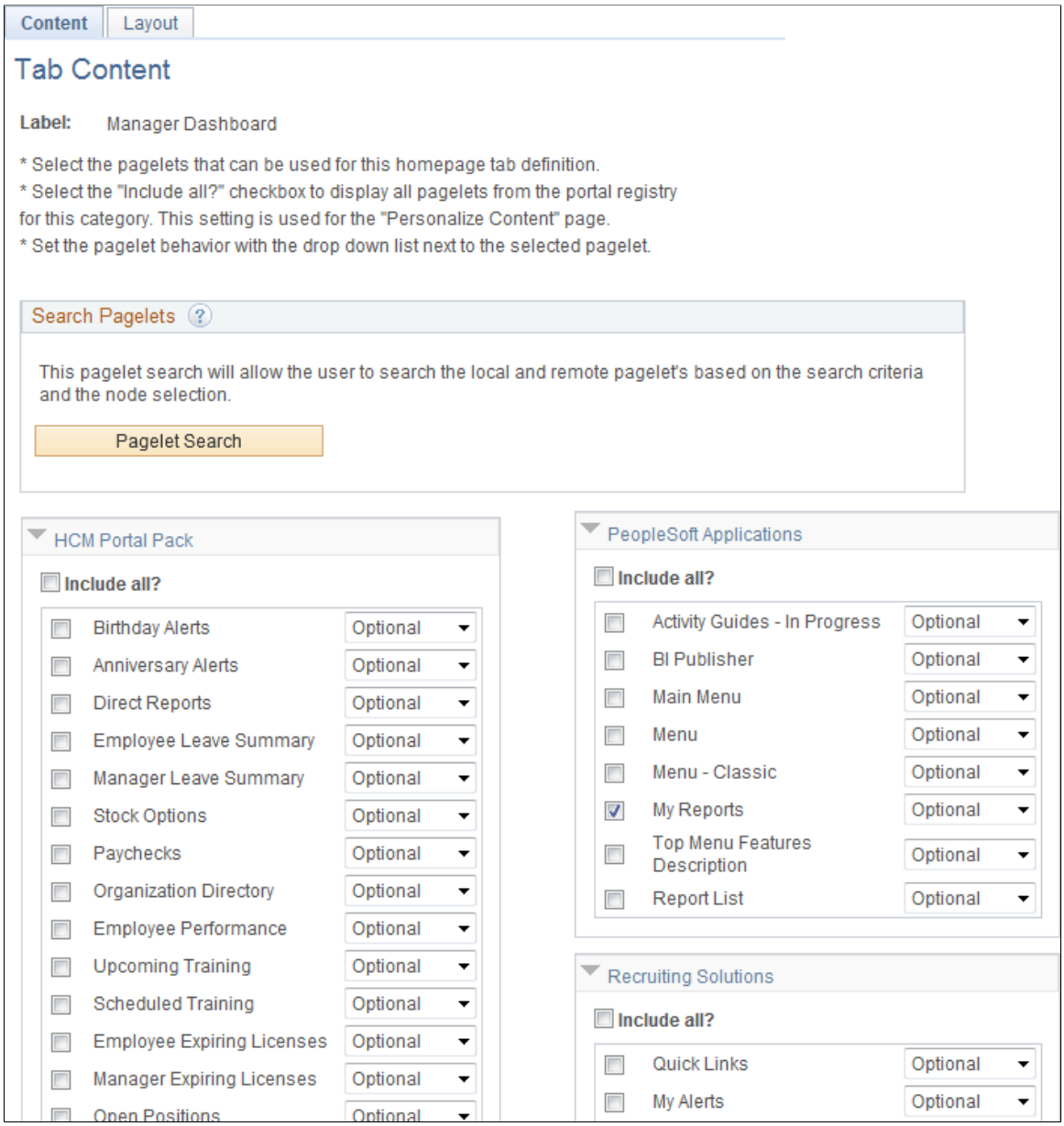
The page that you use to specify which pagelets can appear on dashboard pages is similar to the page that you use for the same purpose on homepages. The process is also the same.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Tab Content page (Select PeopleTools, Portal, Dashboard, Manage Dashboard Pages. Click the Pagelets button.)

Image: Tab Content page

This example illustrates the fields and controls on the Tab Content page. You can find definitions for the fields and controls later on this page.



To select the pagelets that can be displayed on this dashboard:

1. Expand any collapsed group boxes and then select the check box to make the pagelet available for this dashboard page.

   Select the Include All? check box to select all pagelets in the category.

   Note: The user must have security access to the pagelet component for the pagelet to be properly displayed on the dashboard.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. In addition, you can click the Pagelet Search button and select the pagelet from the Pagelet Search page to add to the dashboard definition.

   See Searching for and Importing Pagelets for more information on searching for pagelets on local and remote nodes.

3. Use the drop-down list box to select the pagelet behavior. Values are:

Optional	The pagelet does not appear automatically on the homepage. However, it is available for users when they are personalize their dashboard pages.
Opt-Dflt **(option-default)**	The pagelet appears on all user homepages if the users can access the pagelet. The pagelet can be removed when users personalize the dashboard page.
Req-Fix **(required-fixed)**	The pagelet appears on all user homepages if users can access the pagelet. The placement of the pagelet cannot be changed, and it also cannot be removed.
Required	The pagelet appears on all user homepages if users can access the pagelet. The placement of the pagelet can be changed, but it cannot be removed.

4. Click the Save button.

Important! When you select content for inclusion on any PeopleSoft dashboard page, only one pagelet or portlet should use the calendar prompt. If more than one pagelet or portlet per dashboard page uses the calendar prompt, unexpected results might occur.

Arranging the Pagelets on Dashboard Pages

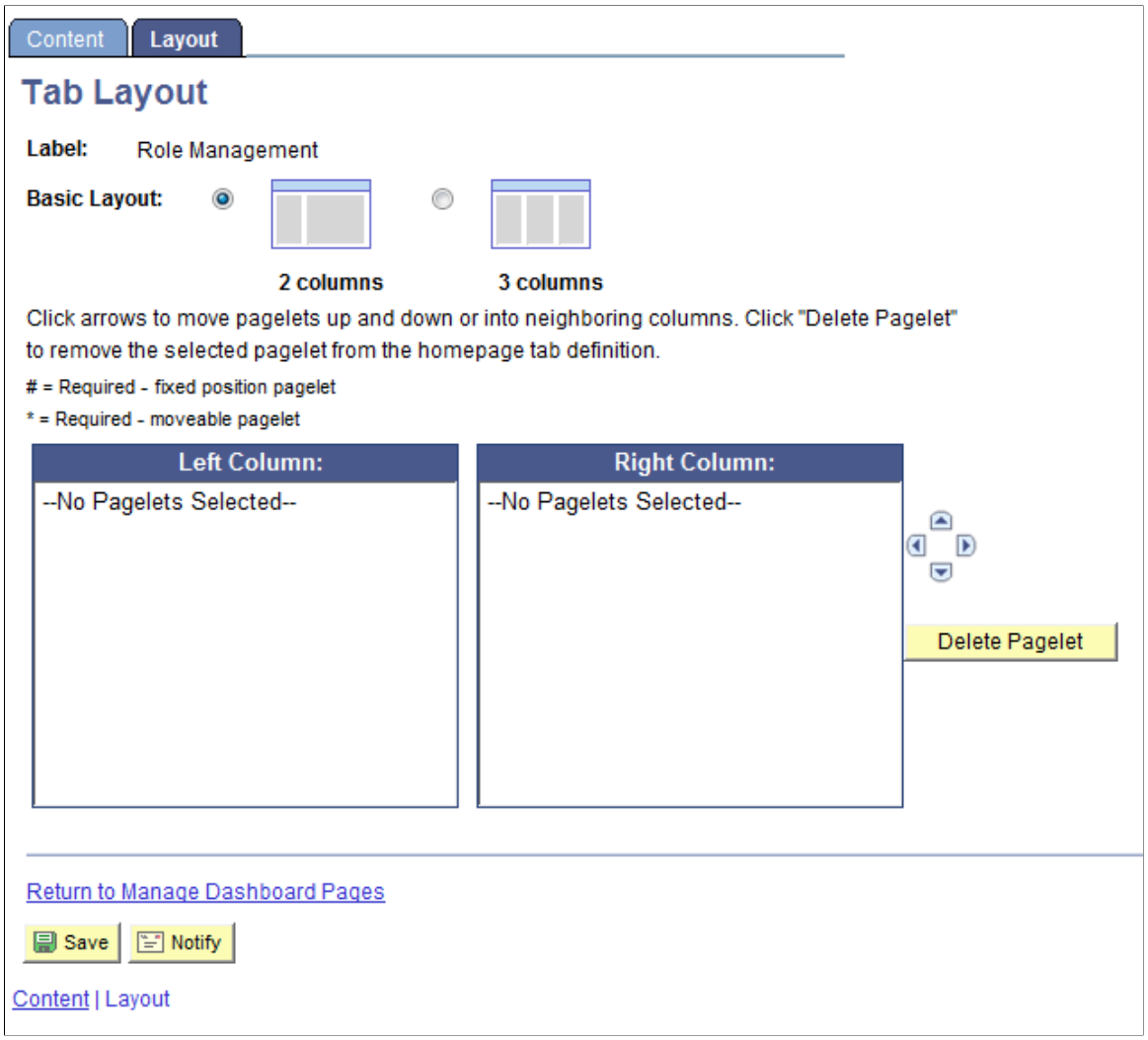
The page that you use to arrange the layout of the pagelets that appear on dashboard pages is similar to the page that you use for the same purpose on homepages. The process is also the same.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Layout page (Select PeopleTools, Portal, Dashboard, Manage Dashboard Pages. Click the Pagelets button.)

Image: Tab Layout page

This example illustrates the fields and controls on the Tab Layout page. You can find definitions for the fields and controls later on this page.

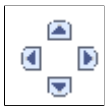


Label	Displays the label information from the Content Ref Administration page.
Basic Layout	Select to display the pagelets in either two or three columns on the dashboard page. To specify the three-column layout, be sure that at least one pagelet does not have the *Req-Fix* behavior option selected on the Content page.
Columns	Displays the pagelets selected on the Content page in the column sections.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

If the basic layout is two columns, the pagelets are divided into left and right columns. If the basic layout is three columns, the pagelets are divided into left, center, and right columns.

Note: A pound sign (#) indicates a pagelet with the *Req-Fix* behavior option selected on the Content page. An asterisk (*) indicates a pagelet with the *Required* behavior option selected on the Content page.
For example, #Signon indicates that the Signon pagelet cannot be moved to a different location on the homepage.



Use the Move icons to position the pagelets. Highlight a pagelet, and then click the directional arrow buttons to move the selected pagelet above a pagelet, below a pagelet, to the next column to the right, or the next column to the left.

Delete Pagelet

Highlight a pagelet and then click to delete it from the homepage.

To arrange the dashboard page layout:

1. Select a basic layout option.
2. Select the pagelet name.
3. Click the directional arrow buttons to position the pagelets in the columns.
4. Click the Save button.

Note: As the portal administrator, you determine the amount of flexibility that you give users to change the layout of the pagelets. The settings that you select on both the Tab Content and Tab Layout pages determine the degree of flexibility. Grant the end user as much or as little flexibility as your organization demands.

Searching for and Importing Pagelets

Pages Used to Search for and Import Pagelets

Page Name	*Definition Name*	*Usage*
Pagelet Search Page	PTUN_PGLTSRCH_SEC	Search for pagelet definitions on local and remote content provider nodes.

Pagelet Search Page

Use the Pagelet Search page (PTUN_PGLTSRCH_SEC) from an invoking definition (homepage tab, dashboard, WorkCenter page, and so on) to search for pagelet definitions on local and remote content provider nodes.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

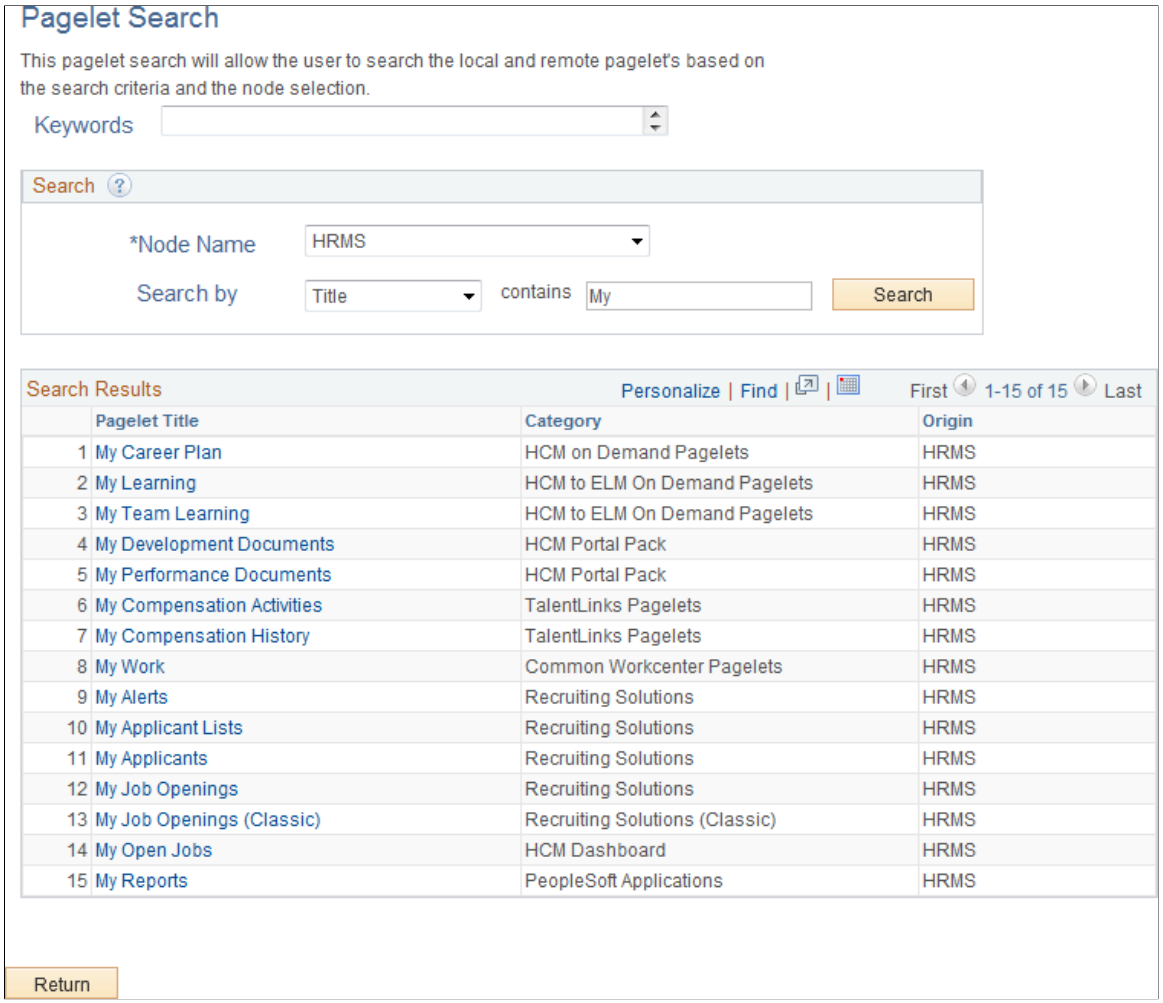
To access the Pagelet Search page:

- Click the Pagelet Search button on the:
  - Dashboard's Tab Content page.
  - Homepage tab's Tab Content page.
  - WorkCenter dashboard's Tab Content page.
- Click the pagelet Select icon button on the:
  - Assign Related Actions page.
  - Assign Related Content page.
  - Configure WorkCenter Page page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Image: Pagelet Search page

This example illustrates the fields and controls on the Pagelet Search page. You can find definitions for the fields and controls later on this page.



Keywords

Displays any keywords defined on the invoking definition as a display-only value—for example, a WorkCenter definition can include keywords.

Note: If keywords are defined on the invoking definition, then this field is populated, the Search by field is set to *Keywords,* the contains field is populated with the same keywords, and the search is automatically executed. You can update the search parameters, such as the contains field, to perform a different search after this initial keyword search.

Search

Node Name

Select the local or remote content provider node.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Important! While this field displays local and remote *default local nodes* and *portal host nodes,* only select portal host nodes (content provider nodes) for pagelet searches.

Search by

Select one of the following:

- *Category* – To search by the title of the pagelet folder.
- *Keyword* – To search by keywords added to the pagelet definition.

  See Maintaining Pagelet Branding Attributes for more information about adding keywords to pagelet definitions.
- *Role* – To search by roles that have access to the pagelet definition.
- *Title* – To search by the pagelet title.

contains

Enter a word, partial word, phrase, or partial phrase as the search term.

Note: Clearing this field and performing a search will return all results displaying valid values for the selected Search by value.

Search

Click the Search button to perform a search.

Return

Click the Return button to return to the invoking page without selecting a pagelet definition.

Search Results

Pagelet Title

Click a pagelet title to select and add that definition to the invoking definition.

Importing Pagelet Definitions

You can search for pagelets in other databases (remote content provider nodes) only after unified navigation has been configured. When you select a remote pagelet on the Pagelet Search page, you import the pagelet definition into the local database. This imported pagelet definition is then available to all features that use pagelets, such as homepage tabs, dashboards, WorkCenters, and so on.

For more information on unified navigation and importing pagelets,see the product documentation for PeopleSoft Interaction Hub:

- For information on configuring unified navigation, see *PeopleSoft Interaction Hub: Portal and Site Administration*, “Administering Unified Navigation in PeopleSoft Interaction Hub.”
- For information on other methodologies on importing pagelets including importing pagelets in bulk, see *PeopleSoft Interaction Hub: Portal and Site Administration*, “Administering Unified Navigation in PeopleSoft Interaction Hub,” Importing Pagelets from Content Provider Systems.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 13

Using Pagelet Wizard to Create and Manage Pagelets

Understanding Pagelet Wizard

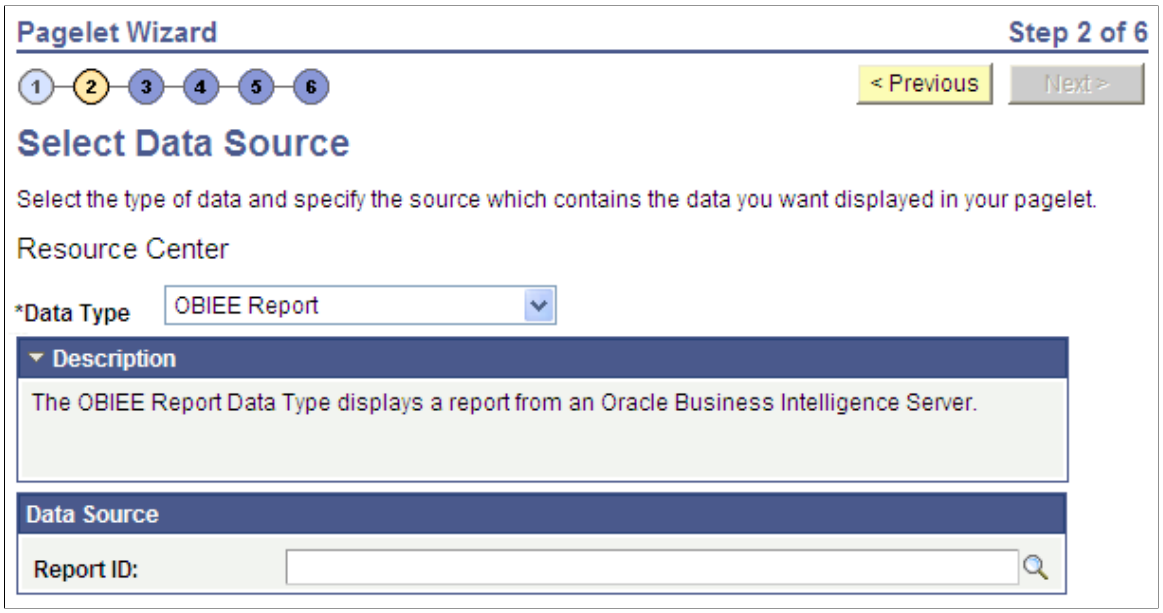
Pagelet Wizard pagelets are small pages that provide display-only snapshots of applicable content. Pagelet Wizard provides a user-friendly, browser-based graphical user interface (GUI) that leads you through the steps involved in creating and publishing a pagelet.

Portal administrators and nontechnical users can use Pagelet Wizard to integrate and transform data from a variety of data sources.

Image: Numbered steps in Pagelet Wizard

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, Pagelet Wizard presents a series of numbered steps. Each step appears in a numbered path at the top of the page to indicate where you are in the pagelet creation process as shown in the following example:

The following screenshot helps us understand numbered steps in the Pagelet wizard.



This labeled path assists you in navigating forward and back through the steps in the wizard, which 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.

Pagelet Wizard leads you through the following steps that are typically used to create a pagelet:

1. Specify pagelet information.
2. Select data source.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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.

Related Links

Step 2: Selecting a Pagelet Data Source

Understanding Pagelet Wizard Data Sources

Pagelet Wizard supports the creation of pagelets using the following data sources:

Activity guides

Use to create activity guide pagelets for deployment in WorkCenters. The pagelet can be generated from an activity guide template and instance, from an activity guide template only, or without a template producing a generic activity guide pagelet.

Free text

Use to present any free-form HTML or text in a pagelet. Free text data source pagelets are compatible with the passthru and custom display formats.

Select this data type instead of the HTML data type when you want to control all aspects of the HTML code, which will allow you to control use of cascading style sheets (CSS), JavaScript, and embedded media such as Adobe Flash.

Important! Free-form HTML can expose your system to potential security vulnerabilities. It is imperative that you understand the security risks before you allow the use of this data type on your system. You can use the Define Data Types page to inactivate or activate a Pagelet Wizard data type. See Defining Data Types.

HTML

Use to present simple formatted HTML in a pagelet. HTML data source pagelets are compatible with the passthru and custom display formats.

Because the HTML data type relies on the rich text editor (RTE) for text entry and formatting, complex HTML code that specifies custom CSS, JavaScript, or embedded media might not appear correctly in a pagelet after RTE post-processing of the entered data. In these circumstances, use the Free Text data type instead.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

IB connectors (Integration Broker connectors)	Use to interact directly with connectors registered to PeopleSoft Integration Broker and secured using Define IB Connector Security within Pagelet Wizard. This data type should be used only when you do not want to use the Integration Broker data type.
Integration Broker	Use to create pagelets that use PeopleSoft Integration Broker to retrieve data from external sources. This middleware technology facilitates synchronous and asynchronous messaging among internal systems and with trading partners, while managing message structure, message format, and transport disparities.
Navigation collections	Use to create 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.
OBIEE reports (Oracle Business Intelligence Enterprise Edition reports)	Use to create pagelets from existing reports on the OBIEE server.
PeopleSoft queries	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, and custom display formats.
Pivot grids	Use to create pagelets from existing pivot grid instances, which use interactive pivot tables and charts that display application data that you can pivot, slice, and dice across multiple dimensions.
Rowsets	Use to access a rowset record for which the data is supplied at runtime. This data source is geared toward supplying data to embedded pagelets. Rowset data source pagelets are compatible with the table, list, chart, and custom display formats. You designate a record as the rowset data source when you design your pagelet using Pagelet Wizard. When the pagelet is run for display, the data populated for the rowset must contain that record in the first level of the rowset. If the runtime rowset does not contain the record specified in Pagelet Wizard, no data appears in the pagelet.
Search records	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,

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	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.
URLs	The URL data type retrieves data from external sources such as the internet. This includes sources such as Real Simple Syndication (RSS) news feeds. Use to display appropriately sized dynamic HTML pages on Pagelet Wizard pagelets. URL data-source pagelets are compatible with the custom and passthru display formats.

Note: The Service Oriented Architecture Protocol (SOAP) data type used in previous releases is now included in the functionality of the IB data type. As of PeopleTools 8.50, SOAP services are consumed using Integration Broker, so use the IB data type to create pagelets going forward.
You can define your own data types on the Define Data Types page.
When you use PeopleSoft Interaction Hub, Pagelet Wizard includes other data sources that are specific to PeopleSoft Interaction Hub installations.

Related Links

Defining Display Formats
Defining Data Types
Step 2: Selecting a Pagelet Data Source

Understanding 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)

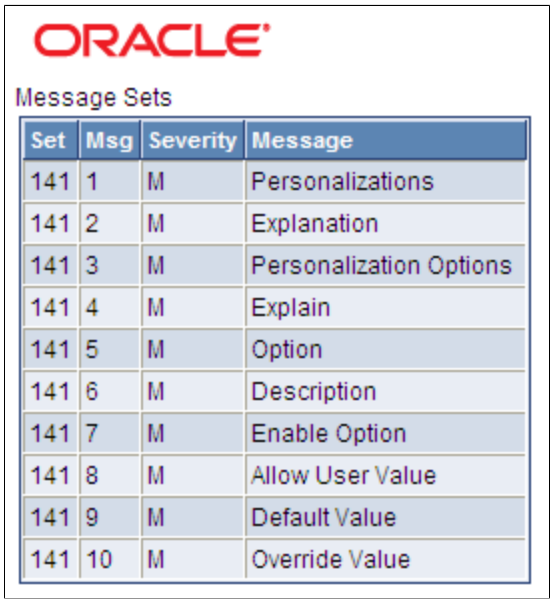
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Table Display Format

Image: Table pagelet example

The table display format is appropriate for creating a pagelet that displays data from the PeopleSoft Query and rowset data sources.

The following screenshot shows a table pagelet displaying data from a PeopleSoft Query data source:

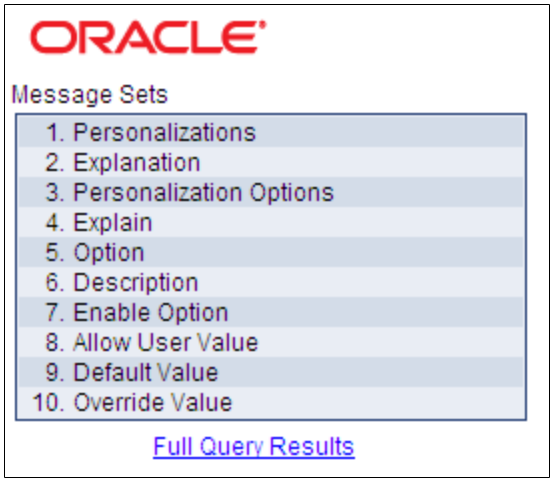


List Display Format

Image: List pagelet example

The list display format is appropriate for creating a pagelet that displays data from the PeopleSoft Query and rowset data sources.

The following screenshot illustrates about list pagelet that displays data from a PeopleSoft Query data source:



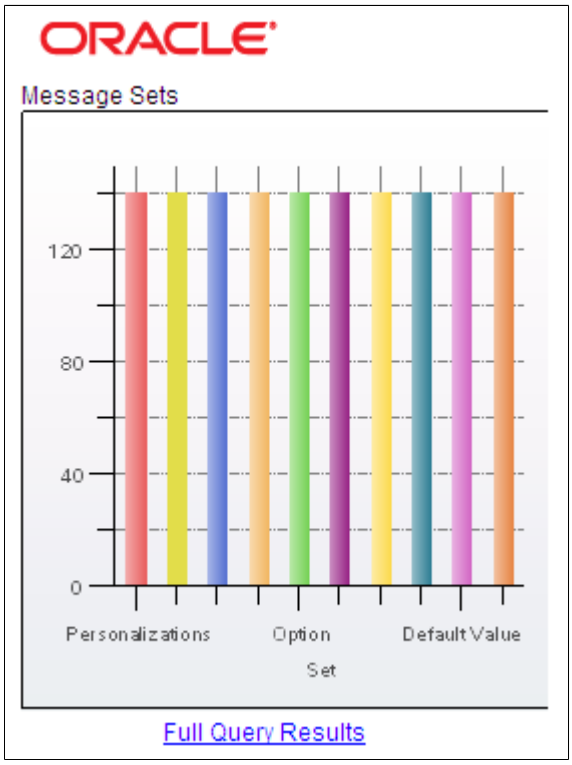
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chart Display Format

Image: Chart pagelet example

The chart display format is appropriate for creating a pagelet that displays data from the PeopleSoft Query and rowset data sources.

The following screenshot shows a chart pagelet displaying data from a PeopleSoft Query data source.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Menu Display Format

Image: Menu pagelet example

The menu display format is appropriate for creating a pagelet that displays data from a navigation collection data source.

The following screenshot illustrates a menu pagelet displaying data from a Navigation Collection data source.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

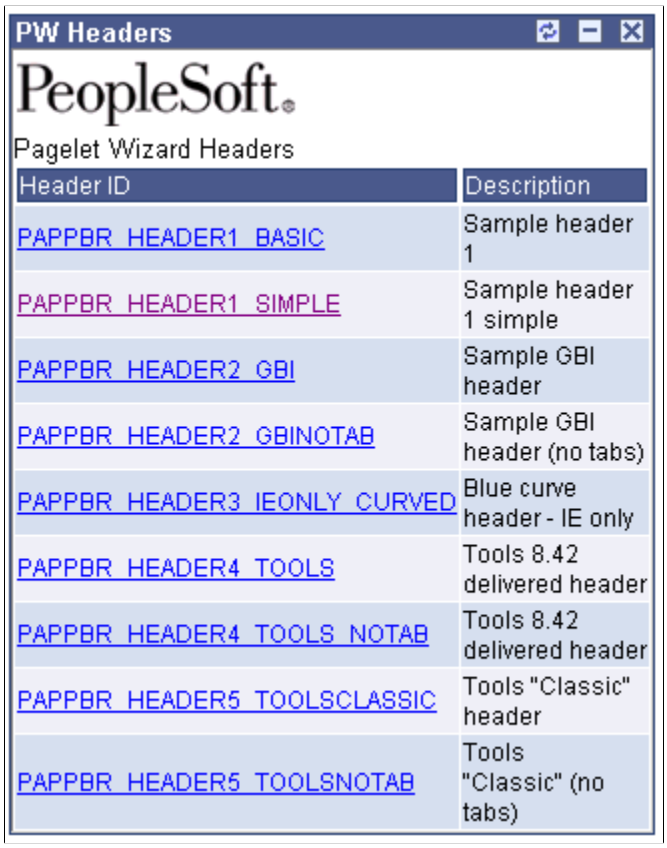
Search List Display Format

Image: Search list pagelet example (Show Results options set to N)

The search list display format is appropriate for creating pagelets that display data from the search record data source. The following examples show search list pagelets displaying data from a search record data source:

Image: Search list pagelet example (Show Results option set to Y)

This example shows the same search list pagelet when the Show Results option has been set to yes.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Passthru Display Format

Image: Passthru pagelet example

The passthru display format is appropriate for creating a pagelet that displays data from the HTML, IB Connector, and URL data sources. Because the passthru display format performs no further transformations on data from the data source, it should be used only with data sources that return HTML that can be directly displayed as a pagelet. The following example shows a passthru pagelet displaying data from an HTML data source:

Custom Display Format

The custom display format is appropriate for creating a pagelet using all data sources that return XML. Generally, most data types return XML, but circumstances potentially occur in which IBConnector, URL, and HTML datatypes do not return XML. In those instances, the custom display format would not be appropriate.

Use the custom display format when you want to use a display format other than the ones expressly listed in Pagelet Wizard. When you select the custom display format, Pagelet Wizard enables you to select a

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

format provided by one of the delivered XSL prototypes, or you can select an XSL template that you created on the Define XSL page.

Image: Custom pagelet example

The following example shows a custom pagelet displaying data from a PeopleSoft Query data source:

This custom pagelet uses the same PeopleSoft Query data source as the table pagelet example shown previously; however, this example has a custom XSL template applied.

Related Links

"Understanding the Charting Classes" (PeopleTools 8.54: PeopleCode API Reference)

Defining Pagelet Wizard XSL Prototypes

Step 4: Selecting a Pagelet Display Format

Understanding 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 so that pagelets are sized appropriately for PeopleSoft portal homepages. 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 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.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Understanding Pagelet Wizard Pagelet Types

Pagelet Wizard supports the publication of the following pagelet types:

- Homepage pagelets.
- Template pagelets.
- Embeddable pagelets.
- Web Services for Remote Portlets (WSRP) portlets.

Homepage Pagelets

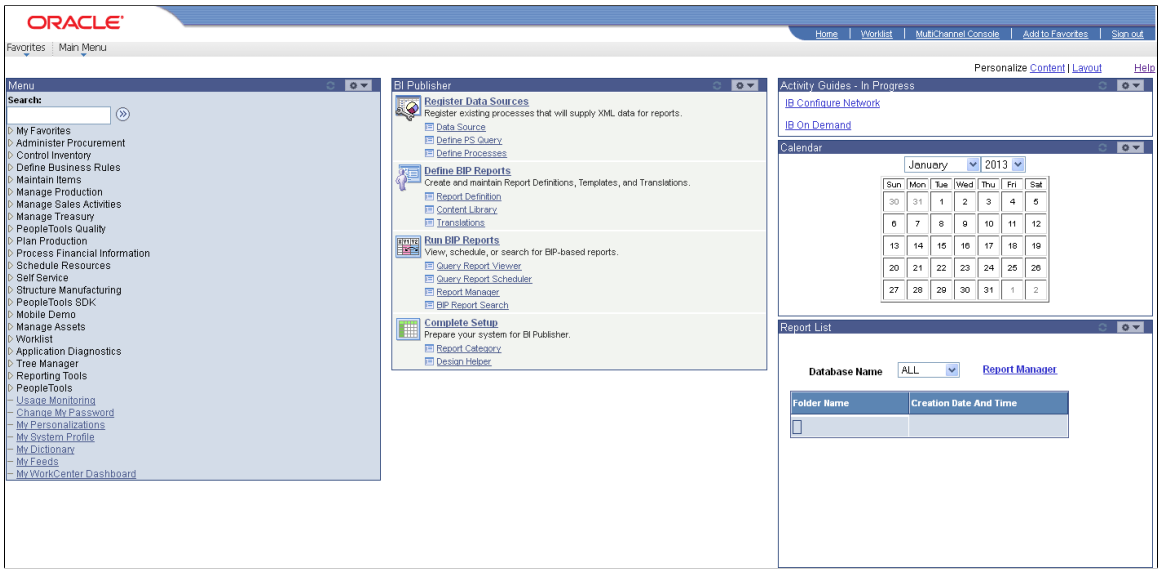
Homepage pagelets are pagelets that are specifically for use on portal homepages. The layout and content of the homepage is usually configurable by the end user, and homepage pagelets can be designated as optional or required.

Note: On any PeopleSoft homepage, only one pagelet or portlet should use the calendar prompt. If more than one pagelet or portlet on a homepage uses the calendar prompt, unexpected results may occur.

See Selecting the Pagelets That Appear on Homepages.

Image: A portal homepage displaying multiple pagelets

The layout and content of the homepage is usually configurable by the end user. As shown in this example, in addition to the Menu pagelet, the user has added four additional pagelets to his or her portal homepage.



Template Pagelets

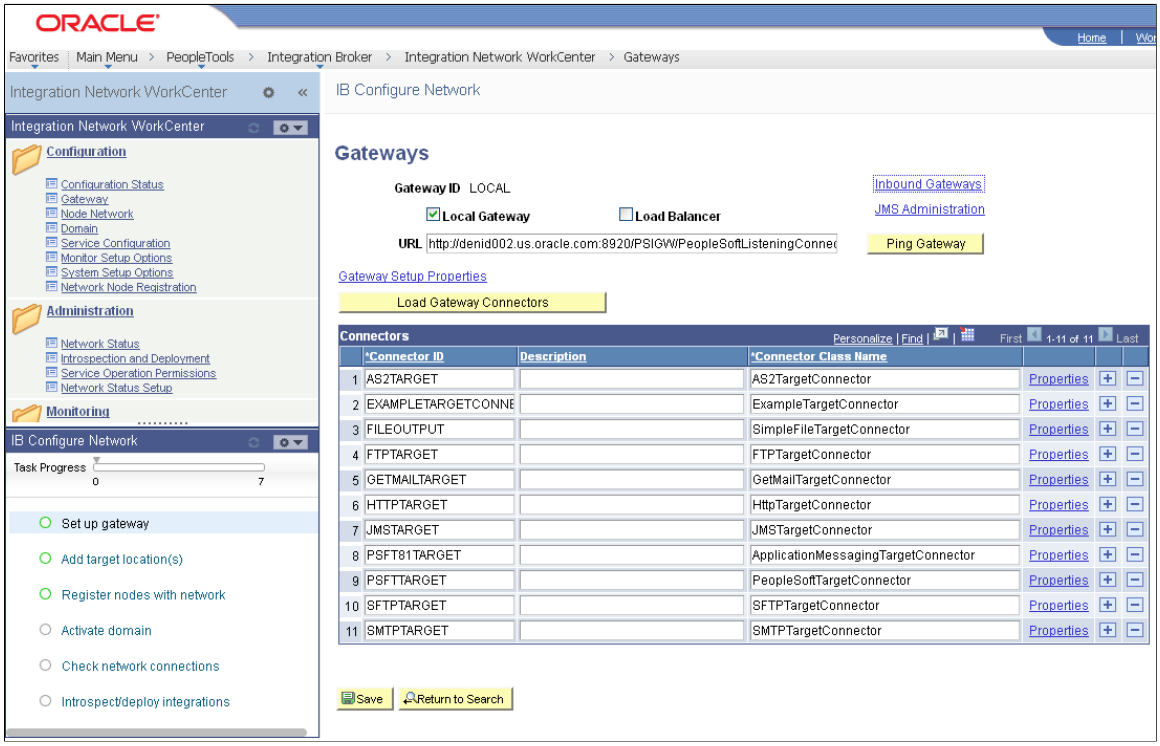
Template pagelets are pagelets that can be added to a PeopleSoft portal template. These pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful when you use Context Manager to display context-sensitive data in the template

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

pagelet. Pagelets that you intend to use in the task panel of WorkCenter pages can also be template pagelets.

Image: Template pagelet example

Each pagelet in the Integration Network WorkCenter is a template pagelet:



Embeddable Pagelets

Embeddable pagelets are pagelets that you can embed in PeopleSoft transaction pages or iScripts within an HTML area, or a grouplet tile for a Fluid Homepage. For example, you might create an OBIEE-based pagelet that displays a report of sales by region to embed on a Regional Sales page so that you could compare the sales of individual regions or each region to the whole.

WSRP Portlets

A WSRP portlet is a pagelet that can be used with external portals that support the WSRP standard.

Related Links

Step 6: Specifying Pagelet Publishing Options
Configuring WorkCenter Pages
Understanding Web Libraries
Understanding WSRP
Fluid User Interface Developer's Guide

Defining Pagelet Wizard Setup Values

This section discusses how to:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Set up the PeopleSoft Integration Broker (IB) gateway for Pagelet Wizard.
- Set up the default gateway for use with Pagelet Wizard data sources.
- Set up the Pagelet Wizard IB Connector data source security.
- Set up the Pagelet Wizard Integration Broker data source.

Setting Up the PeopleSoft Integration Broker Gateway for Pagelet Wizard

This section provides an overview of setting up the PeopleSoft Integration Broker Gateway for Pagelet Wizard and discusses how to set up the default gateway for use with Pagelet Wizard data sources.

When you use Pagelet Wizard with PeopleSoft Interaction Hub installed, the IB Connector and Integration Broker data sources become available for use. These data sources rely on PeopleSoft Integration Broker for web interactivity. You must set up the default gateway before these data sources are usable with your system.

Setting Up the Default Gateway for Use with Pagelet Wizard Data Sources

Access the Gateways page. (Select PeopleTools, Integration Broker, Configuration, Gateways.)

Set up the default gateway according to the instructions in the Integration Broker PeopleBook.

See "Understanding Managing Integration Gateways" (PeopleTools 8.54: Integration Broker Administration).

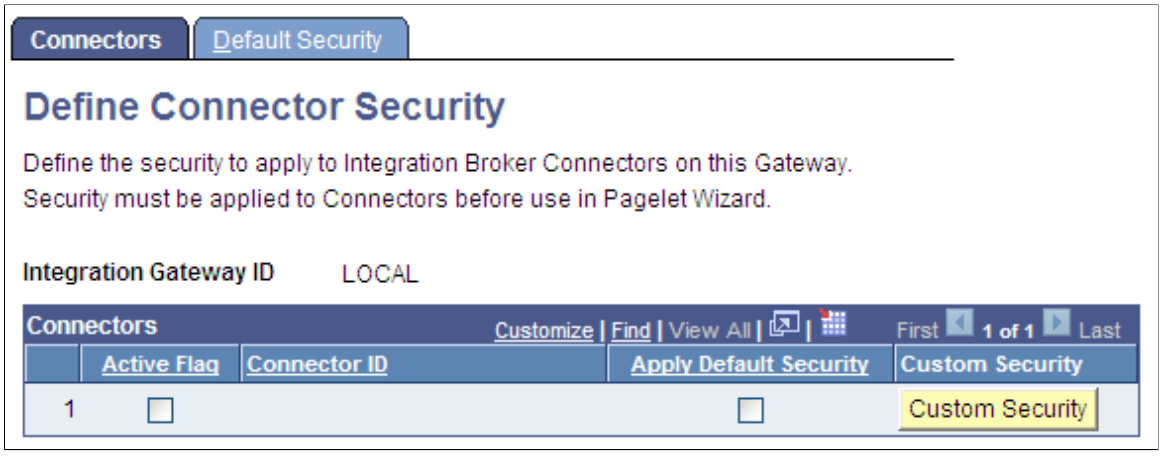
1. Click Save.
2. Click Load Gateway Connectors.
3. Click Save.

Setting Up the Pagelet Wizard IB Connector Data Source Security

Access the Define Connector Security - Connectors page. (Select PeopleTools, Portal, Pagelet Wizard, Define IB Connector Security, Connectors.)

Image: Define Connector Security - Connectors page

The following screenshot illustrates how to define connector security on the Define Connector Security - Connectors page.



Integration Gateway ID Select the appropriate Integration Gateway.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Active Flag	Select to activate the connector and make it available for use with the Pagelet Wizard IB Connector data source.
Apply Default Security	Select to apply the default security settings that are defined on the Define IB Connector Security - Default Security page to the connector.
Custom Security	Click to access the Define Connector Security page, where you can define custom security settings for the connector.

Related Links

"Configuring the Integration Network" (PeopleTools 8.54: Integration Broker Administration)

Setting Up the Pagelet Wizard Integration Broker Data Source

When you use Pagelet Wizard with PeopleSoft Interaction Hub installed, the Integration Broker data source becomes available for use. You must perform the following tasks before the Integration Broker data source is usable with your system.

1. Set up the Integration Broker listening connector.

   See "Configuring the Integration Network" (PeopleTools 8.54: Integration Broker Administration).

2. Set up the appropriate service, service operation, handlers, and routings for the operation that you want to use.

   See "Configuring the Integration Network" (PeopleTools 8.54: Integration Broker Administration).

Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types

This section discusses how to:

- Define transformation types.
- Define display formats.
- Define data types.

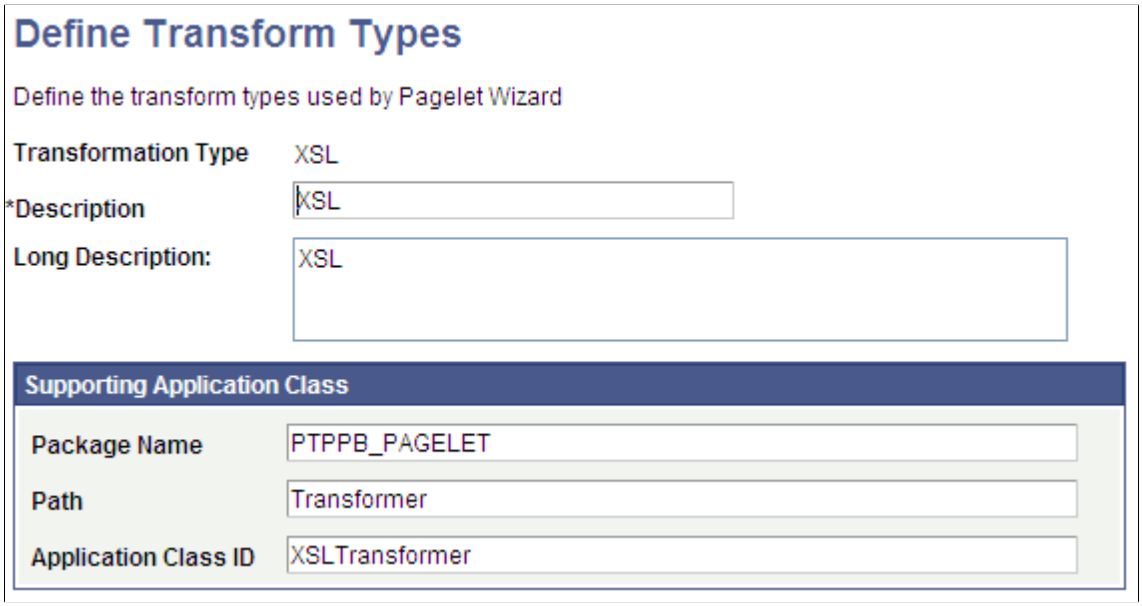
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Transformation Types

Access the Define Transform Types page. (Select PeopleTools, Portal, Pagelet Wizard, Define Transform Types.)

Image: Define Transform Types page

The following screenshot illustrates about the transformation types defined on the 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.

The 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 Understanding Pagelet Wizard Display Formats.

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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Supporting Application Class

The application class specified here refers to the PeopleCode that implements the transformation type.

Package Name Enter the name of the application class package that you want to use for the transformation type.

Path Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method to use for the transformation type.

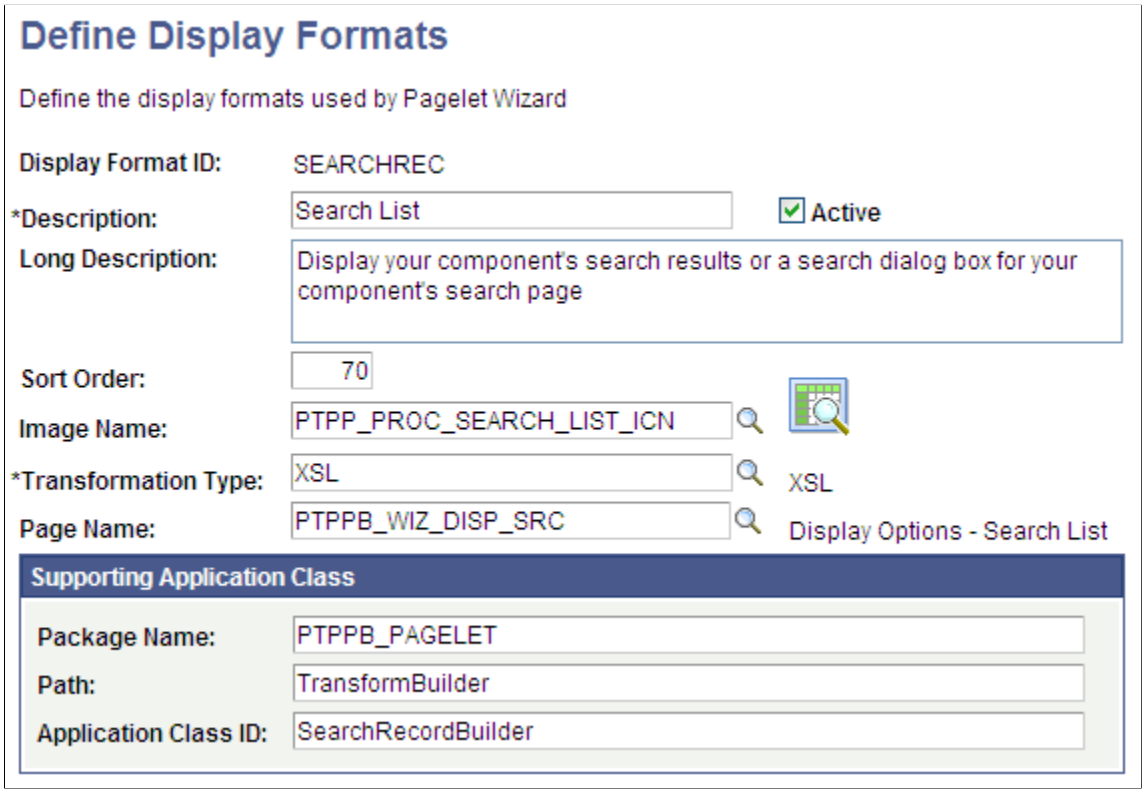
Application Class ID Enter the name of the application class that you want to use for the transformation type. The class must exist in the application package name that you have specified.

Defining Display Formats

Access the Define Display Formats page. (Select PeopleTools, Portal, Pagelet Wizard, Define Display Formats.)

Image: Define Display Formats page

The following screenshot explains about defining display formats on the 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.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

For example, the chart display format defines the code that builds charts, and it specifies that Pagelet Wizard must use the XSL transformation 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 Understanding Pagelet Wizard Display Formats.

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 by means of 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 Step 4: Selecting a Pagelet Display Format.
Image Name	Select the image that you want to appear with the display format name and description on the Select Display Format page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 Defining Transformation Types.

Page Name

Enter the object name of the page to be used to define display options for this display format. This page must exist in the PTPPB_WIZARD component in PeopleSoft Application Designer.

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 transformation) to display the pagelet.

Package Name

Enter the application class package name to use for the display format.

Path

Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method used to create the display format.

Application Class ID

Enter the name of the application class to use for the display format. The class must exist in the application package name that you specify.

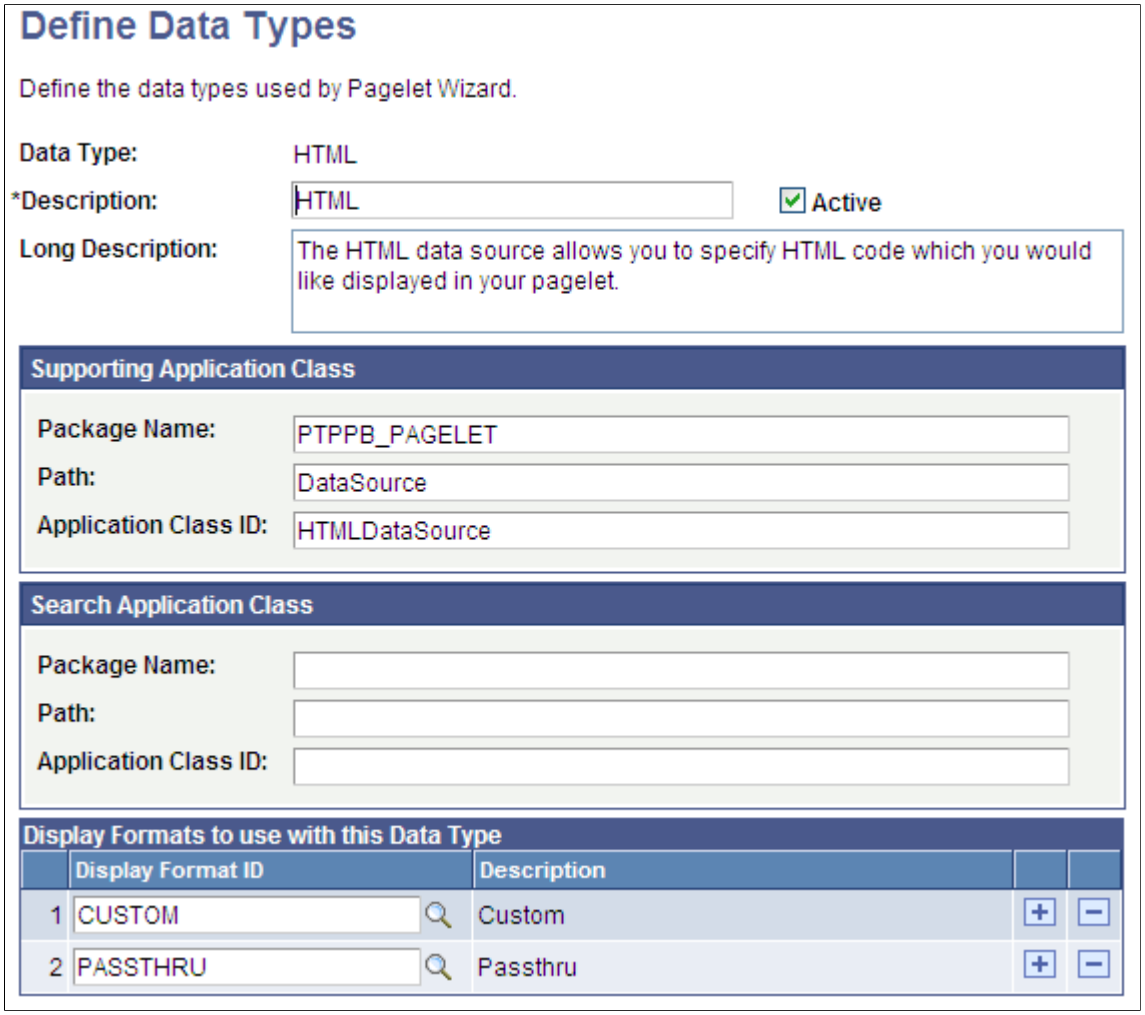
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Data Types

Access the Define Data Types page. (Select PeopleTools, Portal, Pagelet Wizard, Define Data Types.)

Image: Define Data Types page

The following screenshot illustrates data types available on the Define Data Types page.



Pagelet Wizard is delivered with the following data type definitions to support their associated data sources:

- ACTIVITY_GUIDE data type enables you to generate activity guide pagelets.
- FREE_TEXT data type allows you to generate pagelets from any complex, free-form HTML or text.
- HTML data type allows you to generate pagelets from any simple, formatted HTML code.
- IBCONNECTOR data type interacts directly with connectors registered to the Integration Broker and secured using Define IB Connector Security within Pagelet Wizard. This data type should be used only when it is not desirable to use the Integration Broker data type.
- IBDATASOURCE data type retrieves messages using Integration Broker. Specifically, it uses outbound synchronous messages to retrieve data from external sources.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- NAVIGATION_COLLECTION data type enables you to display navigation collections within the context of a pagelet.
- OBIEE data type interacts directly with the analytics server.
- PIVOTGRID data type enables you to display pivot tables and charts in pagelets.
- PSQUERY data type enables you to use PeopleSoft Query to create database queries on records and fields within the PeopleSoft system.
- ROWSET data type is used with embedded pagelets. Rowsets can exist within components.
- SEARCHREC data type displays the search box or the search results from a given component.
- URL data type retrieves data from external sources such as the internet. This includes sources such as news feeds.

Warning! Do not alter delivered field values for these data types.

See Understanding Pagelet Wizard Data Sources.

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 to 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.

Long 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.

This table describes the Supporting Application Class fields:

Package Name Enter the name of the application class package that you want to use for the data type.

Path Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method that defines the data type.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Application Class ID Enter the name of the application class that you want to use for the data type. The class must exist in the application package name that you specify.

Note: The search option of each pagelet has to be explicitly declared through this page or Pagelet Wizard to activate the search box, even for *Inline XSL* type of search box.

Search Application Class

The application class specified here refers to the PeopleCode that implements a default pagelet search for this data type. This is the code that Pagelet Wizard calls to build and place the search box in the pagelet, perform the search, and render and format the search results.

Changes to the default search application class at the data type level appear in any pagelet of this data type automatically the next time the pagelet is rendered, with the exception of the pagelets for which you define a custom search application class. Examples of these changes are included in the subsequent list.

- Changing the code in the search class.
- Adding a new search class.
- Removing a search class.
- Replacing a search class.

The *PTPPB_PAGELET:PageletSearch:PortalSearch* sample search application class is part of a standard PeopleTools installation. This class uses the portal registry index to perform the search. You can use this sample search class to test the pagelet search functionality for any pagelet or you can associate this search class with either a pagelet data type or with a pagelet.

Note: Pagelet search is supported for homepage and embedded pagelets only.

This table describes the Search Application Class fields:

Package Name Enter the Search application class package name that you want to use for the data type.

Path Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method to define the search.

Application Class ID Enter the name of the Search application class 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 Defining Display Formats.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

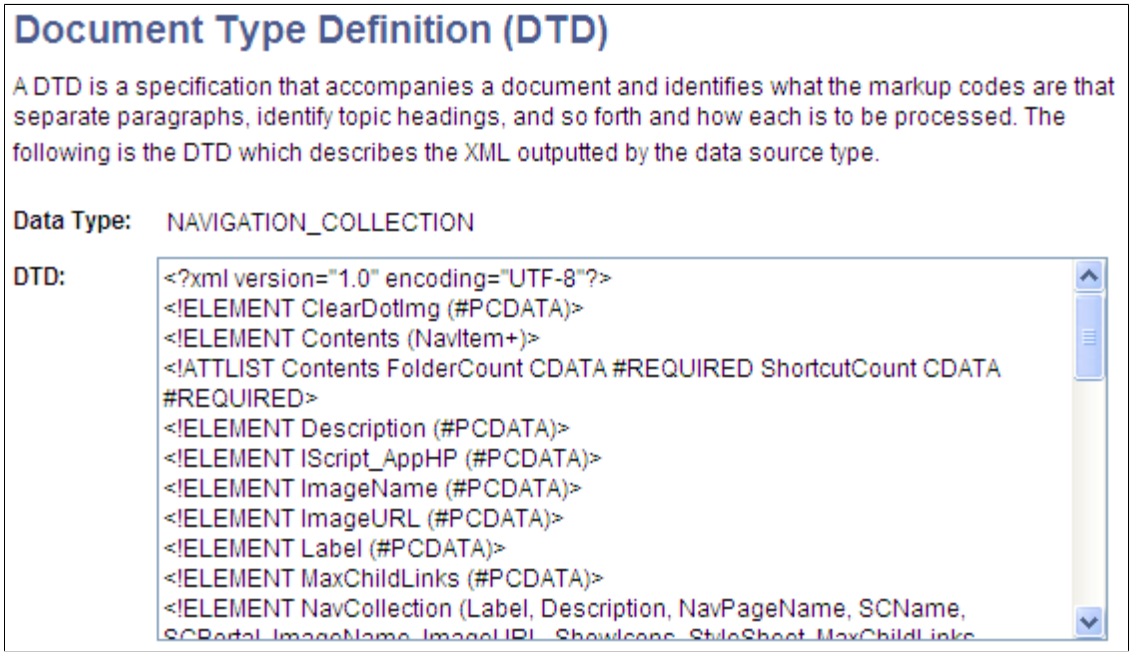
View Document Type Definition (DTD) Click to access the Document Type Definition (DTD) page, where you can access the DTD that describes the XML generated by the selected data type.

Viewing the Document Type Definition (DTD)

The Document Type Definition (DTD) page enables you to view the tags and attributes that are used to describe content in the XML from the data source type. This information is useful when you use third-party tools to generate XSL templates for the datatype.

Image: Document Type Definition (DTD) page

The following screenshot illustrates a DTD specification on the Document Type Definition (DTD) page.



Defining Pagelet Wizard XSL Prototypes

This section discusses how to define XSL prototypes for use in building Pagelet Wizard pagelets.

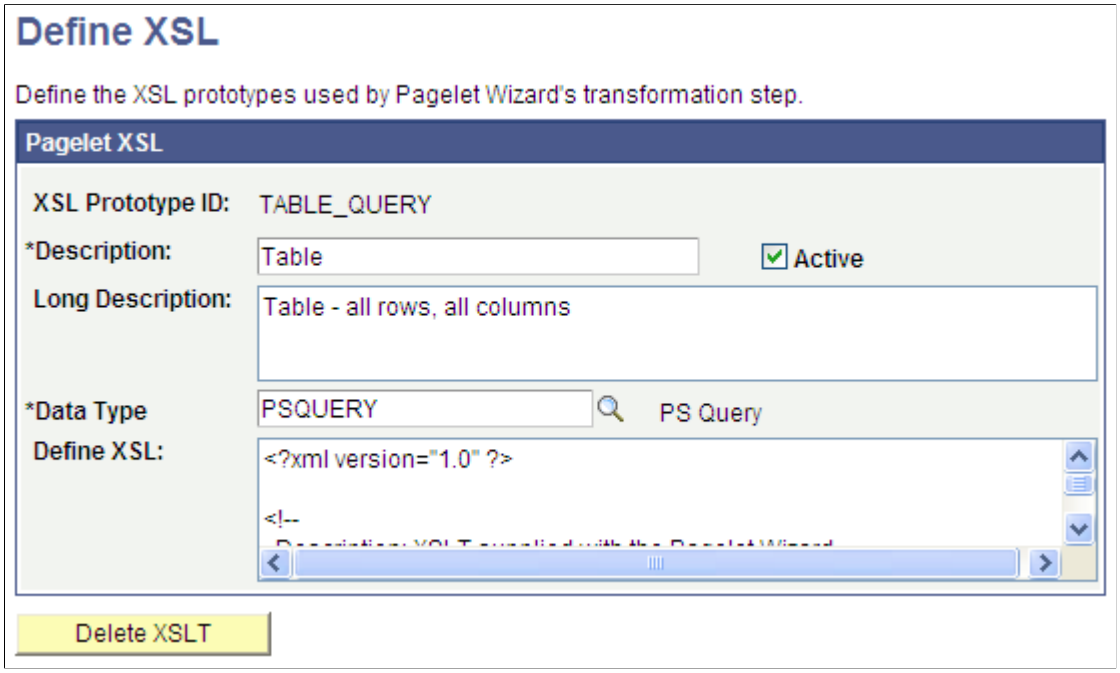
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining XSL Prototypes

Access the Define XSL page. (PeopleTools, Portal, Pagelet Wizard, Define XSL).

Image: Define XSL page

The following screenshot illustrates XSL prototypes on the Define XSL page.



Pagelet Wizard is delivered with the following XSL prototypes to support their associated data sources and display formats:

XSL Prototype Name	*Data Type*	*Description*
NAV_SMALLIMAGES	NAVIGATION_COLLECTION	Small Images
LIST_QUERY	PSQUERY	Bullet List
NUMBERED_LIST_QUERY	PSQUERY	Numbered List
TABLE_QUERY	PSQUERY	Table
PAPP_IB_BULLET_LIST	IBDATASOURCE	Bullet List
PAPP_IB_CUSTOM_LIST	IBDATASOURCE	Customizable List
PAPP_IB_CUSTOM_TABLE	IBDATASOURCE	Customizable Table
PAPP_IB_NUMBER_LIST	IBDATASOURCE	Numbered List
PAPP_IB_NUMBER_LIST_H	IBDATASOURCE	Numbered List - Header
PAPP_IB_SKELETON	IBDATASOURCE	Skeleton XSL
PAPP_IB_TABLE	IBDATASOURCE	Table

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

XSL Prototype Name	*Data Type*	*Description*
PAPP_URL_ATOM03	URL	ATOM 0.3
PAPP_URL_ATOM10	URL	ATOM 1.0
PAPP_URL_RSS	URL	RSS
PAPP_URL_RSS091	URL	RSS 0.91-0.93
PAPP_URL_RSS10	URL	RSS 1.0
PAPP_URL_RSS20	URL	RSS 2.0
PAPP_URL_SKELETON	URL	Skeleton XSL

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 Specifying Custom Display Options.

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 by means of 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 the XSL code for formatting pagelet XML data in the desired pagelet format.

Delete XSL Click to delete the XSL prototype definition. You will be prompted to confirm your deletion.

Defining Pagelet Wizard Headers and Footers

This section discusses how to:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

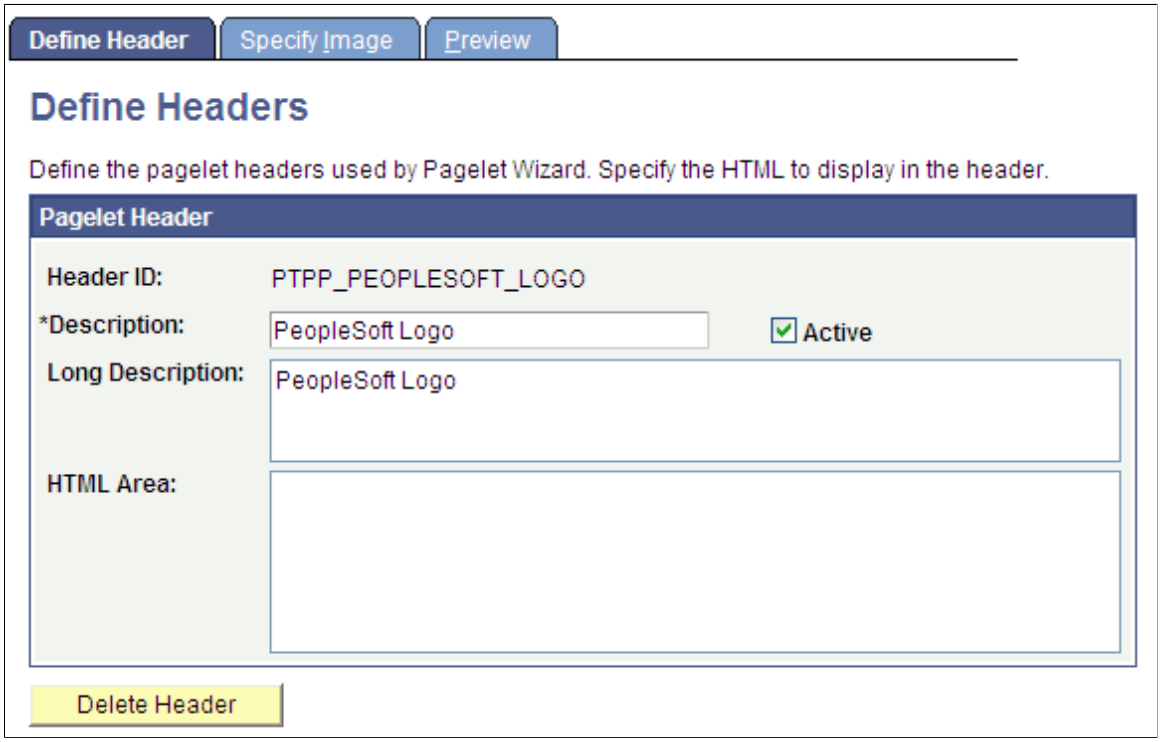
- Define headers.
- Define header images.
- Define footers.
- Define footer images.

Defining Headers

Access the Define Header page. (Select PeopleTools, Portal, Pagelet Wizard, Define Headers, Define Header.)

Image: Define Header page

The following screenshot illustrates how to define headers on the 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

If you deactivate a header that is in use, the header no longer appears on associated pagelets. In addition, the header is not 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.

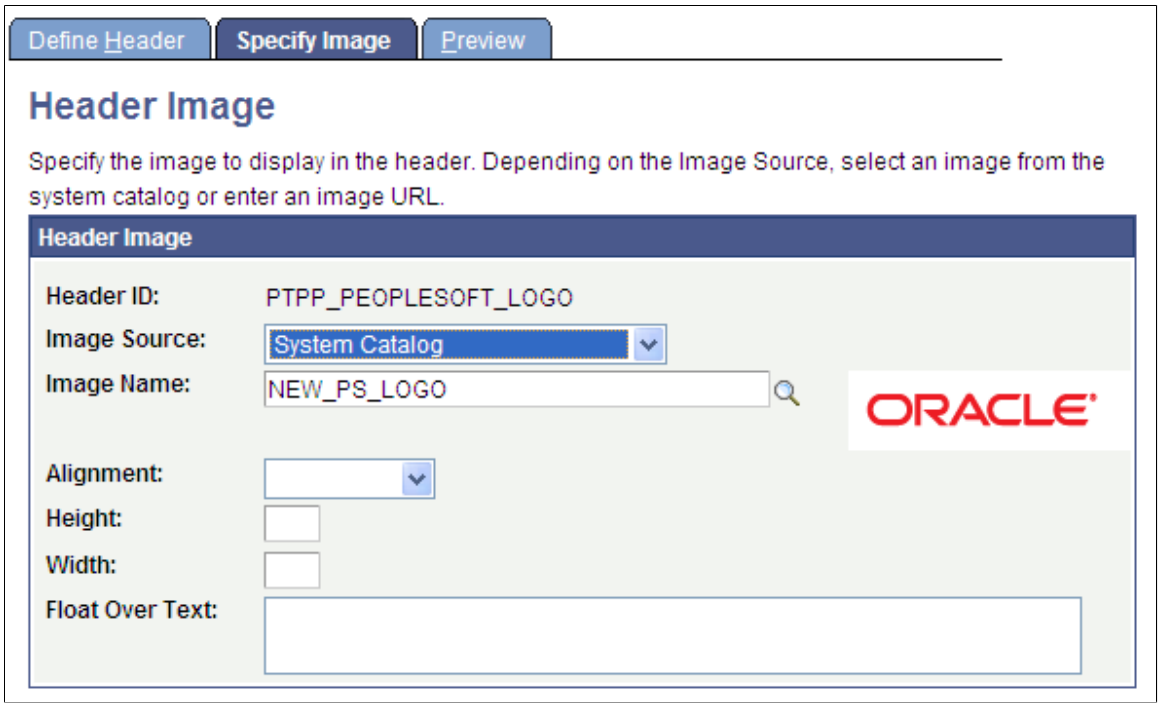
Delete Header Click to delete the pagelet header design. You will be prompted to confirm your deletion.

Defining Header Images

Access the Header Image page. (Select PeopleTools, Portal, Pagelet Wizard, Define Headers, Define Header, Specify Image.)

Image: Define Headers - Specify Image page

The following screenshot illustrates header images on the 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	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.
Image URL	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.

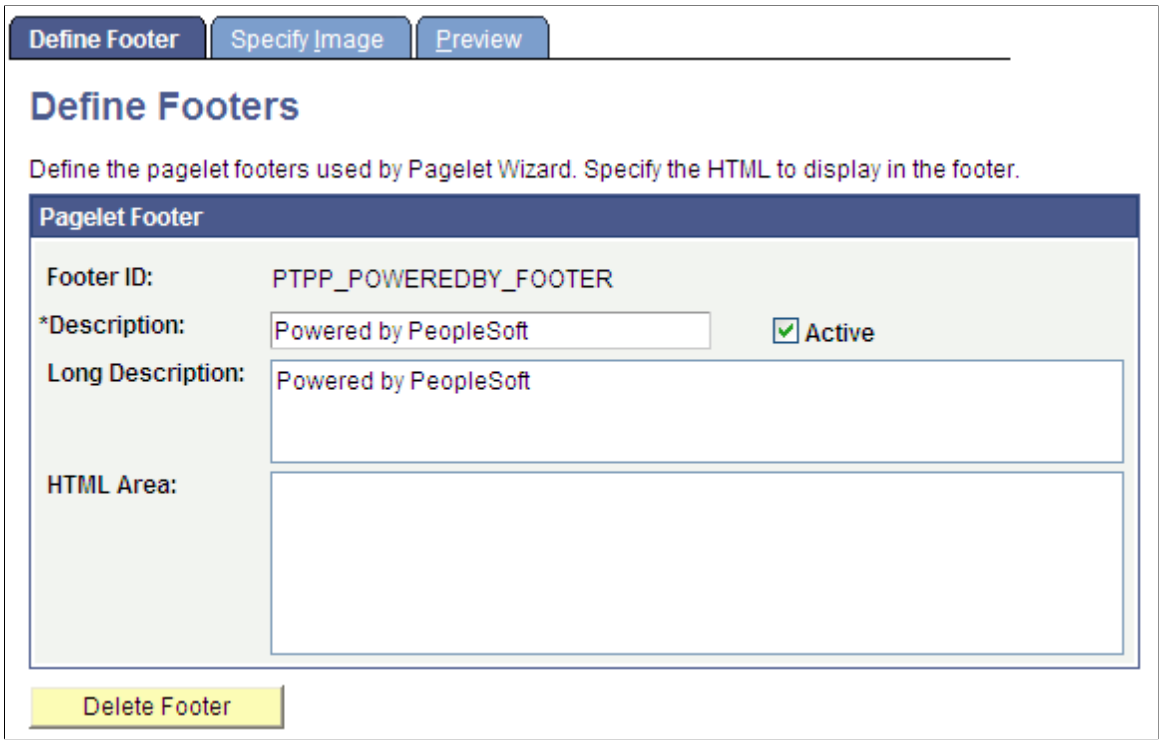
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining Footers

Access the Define Footer page. (Select PeopleTools, Portal, Pagelet Wizard, Define Footers, Define Footer.)

Image: Define Footers - Define Footer page

The following screenshot illustrates pagelet footers on the Define Footers - 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 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 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 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 the 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

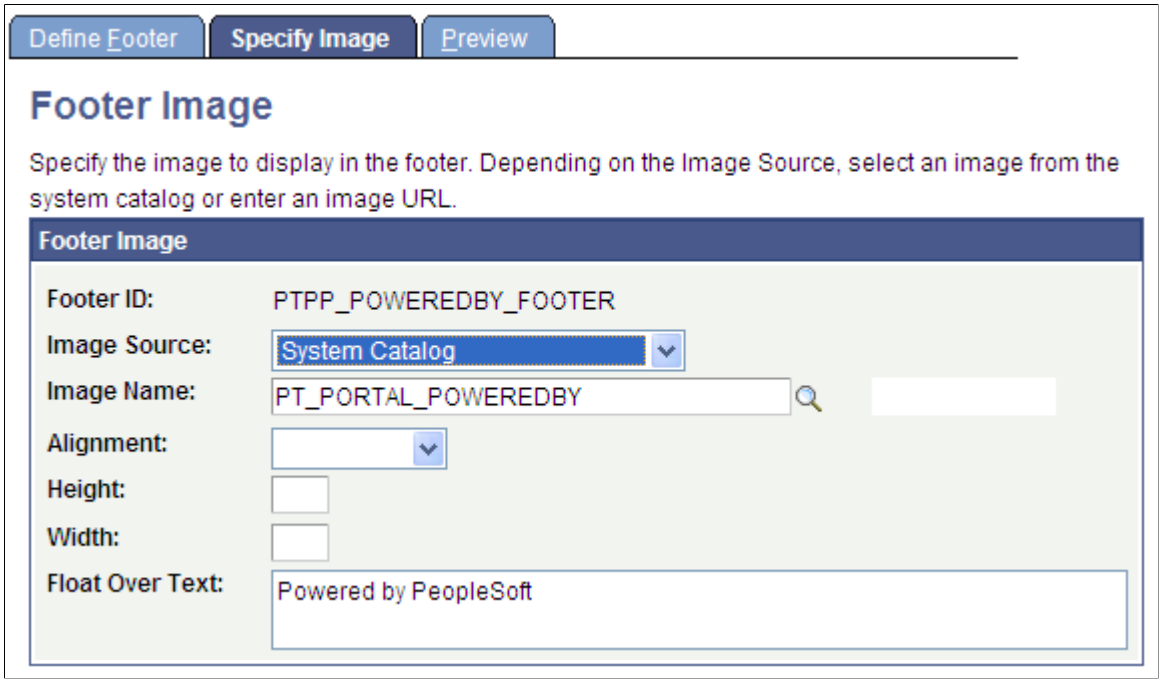
Delete Footer Click to delete the pagelet footer design. You will be prompted to confirm your deletion.

Defining Footer Images

Access the Specify Image page. (Select PeopleTools, Portal, Pagelet Wizard, Define Footers, Specify Image.)

Image: Define Footers - Specify Image page

The following screenshot illustrates footer images on the 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.

Image URL 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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

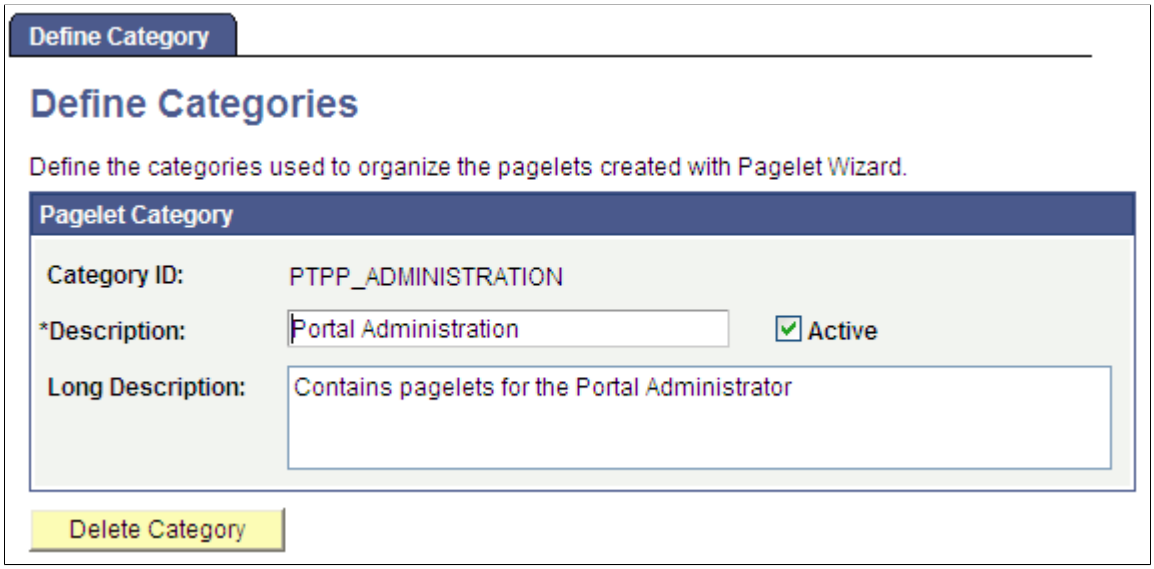
This section discusses how to define Pagelet Wizard pagelet categories.

Defining Pagelet Categories

Access the Define Categories page. (Select PeopleTools, Portal, Pagelet Wizard, Define Categories.)

Image: Define Categories page

The following screenshot illustrates pagelet categories on the 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.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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.

Step 1: Entering Pagelet Identifying Information

This section discusses how to specify pagelet information. This is step 1 in the Pagelet Wizard pagelet creation process.

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.

Related Links

Understanding Pagelet Wizard Data Sources

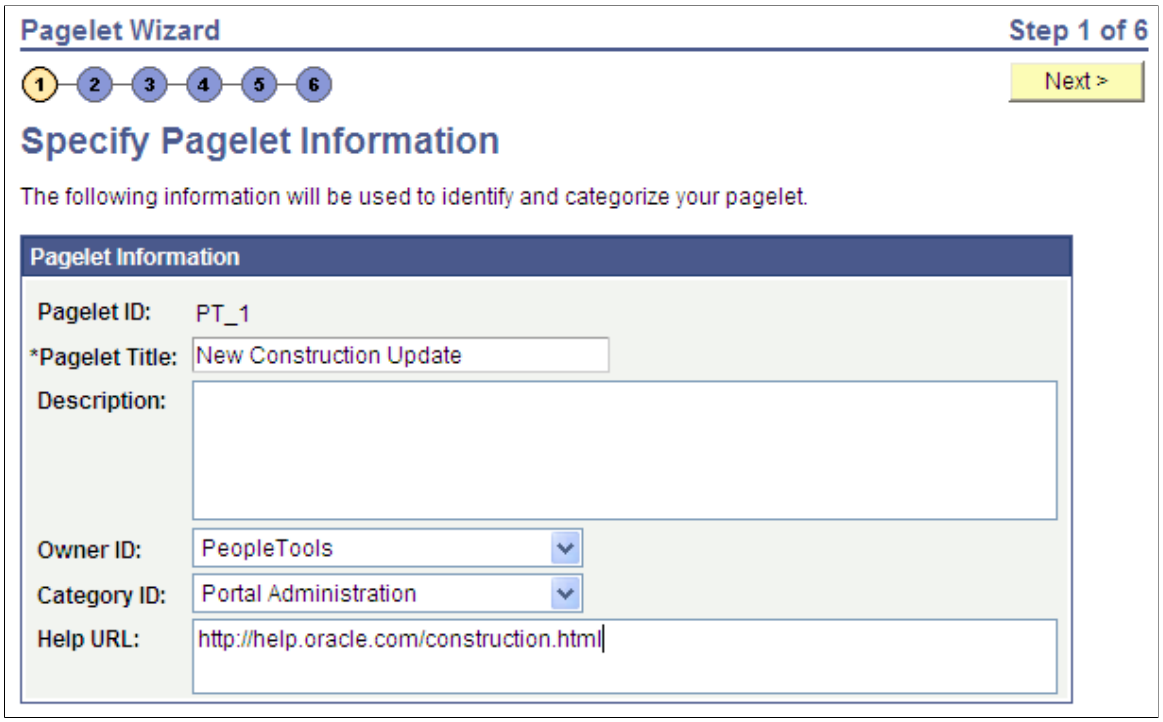
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Pagelet Information

Access the Specify Pagelet Information page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard.)

Image: Specify Pagelet Information page

The following screenshot illustrates pagelet information on the 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.

Note: Pagelet IDs can contain only the characters A–Z, a–z, and _. All other characters put into a pagelet ID will be removed by the system. If all characters are removed from the user-entered pagelet ID, then the Registry Object Prefix will be used instead. If after you remove illegal characters from the pagelet ID the name is found to match an existing pagelet ID, then Pagelet Wizard appends numbers to the end of the pagelet ID to make the new pagelet ID unique.

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

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 Defining Pagelet Wizard Pagelet Categories.

Help URL

Enter the URL of the help document to associate with the pagelet. Entering a URL causes the Help button to appear in the pagelet header bar.

When the pagelet is published as a homepage pagelet, a PTPPB_PAGELET_HELP_LINK content reference attribute is added to the pagelet content reference definition corresponding to the pagelet. The value of this content reference attribute will be the help URL entered by the user.

Note: The Help button appears only when the pagelet is on the homepage inside the PeopleSoft Portal; it does not appear when the pagelet is displayed as a WSRP portlet.

Note: The pagelet help URL takes precedence over the help ID, when both attributes are set for a pagelet.

Step 2: Selecting a Pagelet Data Source

This section discusses selecting a pagelet data source. This is step 2 in the Pagelet Wizard pagelet creation process. Specifically, this section discusses how to :

- Select an activity guide data source.
- Select a free text data source.
- Select an HTML data source.
- Select an IB connector data source.
- Select an Integration Broker data source.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Select a navigation collection data source.
- Select an OBIEE report data source.
- Select a PeopleSoft Query data source.
- Select a pivot grid data source.
- Select a rowset data source.
- Select a search record data source.
- Select a URL data source.

You can define other data sources by using the Define Data Types page.

Related Links

Understanding Pagelet Wizard Data Sources
Defining Data Types

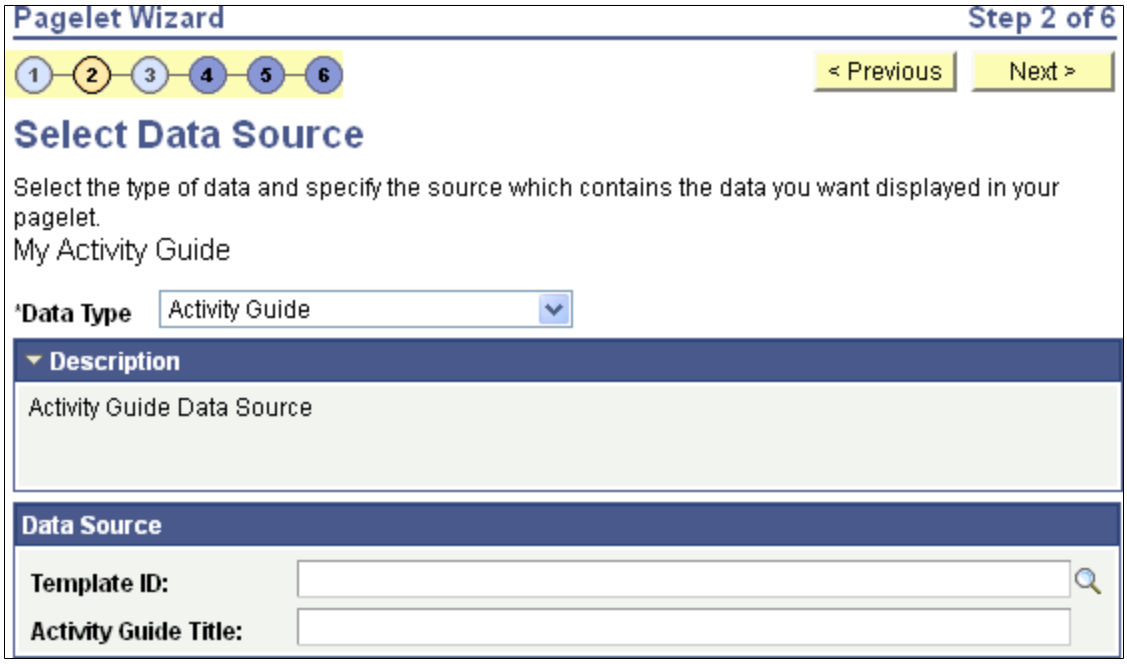
Selecting an Activity Guide Data Source

Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Select *Activity Guide* in the Data Type field.

Image: Select Data Source page (Activity Guide data source)

This example illustrates the fields and controls on the Select Data Source page for an Activity Guide data source. You can find definitions for the fields and controls later on this page.



Depending on whether you want to tie the activity guide pagelet to a specific activity guide instance or to dynamically create the activity guide instance, there are three approaches for generating the activity guide pagelet:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Provide the template ID, but not the instance ID, which allows the instance to be obtained or generated at runtime. At runtime, the required context data is passed as *query string parameters* in the WorkCenter URL as name-value pairs. If an activity guide instance exists for this context data, then it is retrieved. Otherwise, a new activity guide instance will be created dynamically.

  You can use the Publish as Pagelet task in the Setup Activity Guide WorkCenter to generate an activity guide pagelet of this type, which is typically all that is required in most circumstances. Alternatively, you can use Pagelet Wizard to generate this type of activity guide pagelet.

- Provide both the template ID and the activity guide instance ID, which creates a pagelet tied to an existing activity guide instance.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. The template ID is specified as a data source identifier in step 2 of Pagelet Wizard; the instance ID is specified as a data source parameter in step 3.

- Do not provide either the template ID or the activity guide instance ID, which generates an empty, generic activity guide pagelet that can be used to generate any instance from any activity guide template at run time.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. At run time, the template ID is passed as a data source parameter to the pagelet and the context data is provided as query string parameters in the WorkCenter URL as name-value pairs. This will create the activity guide instance dynamically.

Data Source

Template ID	Select a specific activity guide template or leave this field blank to generate a generic activity guide pagelet.
Activity Guide Title	Specify the title for the activity guide pagelet.

Related Links

Generating an Activity Guide Pagelet

Selecting a Free Text Data Source

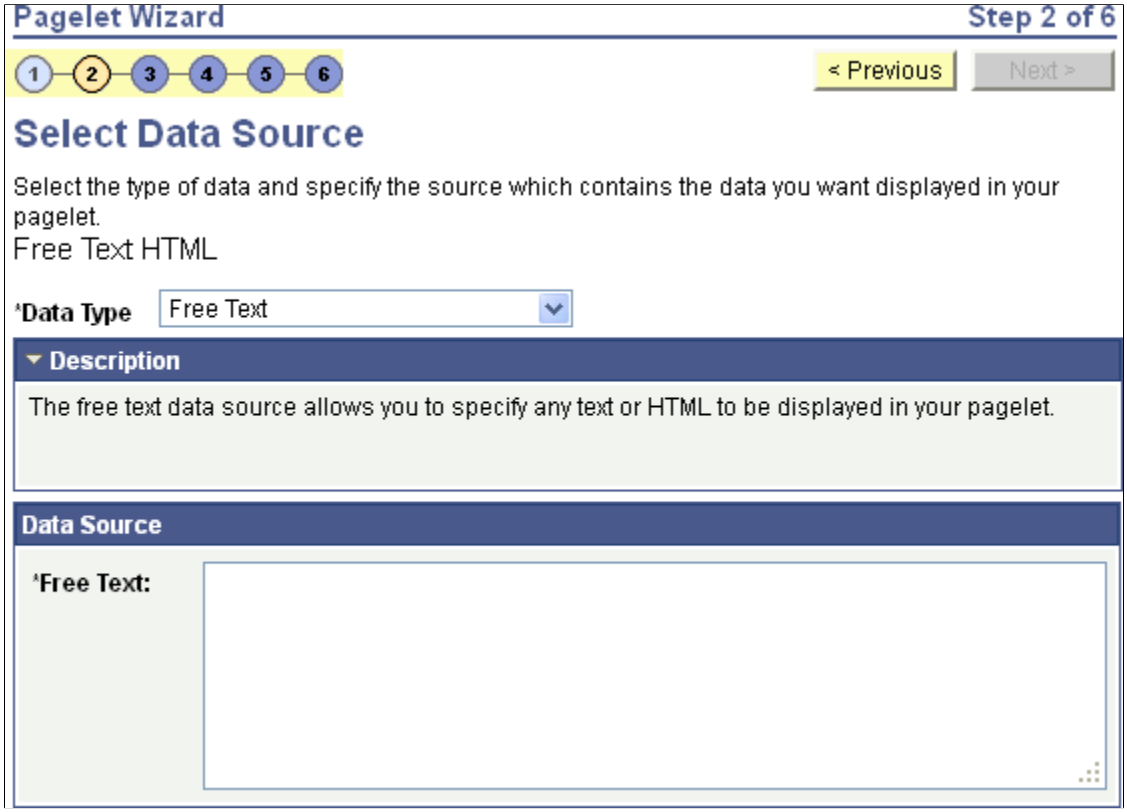
Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select *Free Text* in the Data Type field.

Image: Select Data Source page (Free Text data source)

This example illustrates the fields and controls on the Select Data Source page for a Free Text data source. You can find definitions for the fields and controls later on this page.



Use the free text data source to present complex formatted HTML in a pagelet. Select this data type instead of the HTML data type when you want to control all aspects of the HTML code, which will allow you to control use of cascading style sheets (CSS), JavaScript, and embedded media such as Adobe Flash.

Important! Free-form HTML can expose your system to potential security vulnerabilities. It is imperative that you understand the security risks before you allow the use of this data type on your system. You can use the Define Data Types page to inactivate or activate a Pagelet Wizard data type. See Defining Data Types.

Data Source

Free Text

Enter the HTML, XML, JavaScript, or text that you want to use as the data source for the pagelet. You can enter an HTML document or fragment. This code can include embedded JavaScript, custom CSS, or embedded media such as Adobe Flash.

Ensure that the code that you enter generates content that is appropriately sized for pagelet display.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: To activate the Next button, tab out of the Free Text field or click the Save button.

When you use free text as the data source for a pagelet, there are no configurable data source parameters. Consequently, no end-user personalization exists for free text data source pagelets. When creating an free text data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

Selecting an HTML Data Source

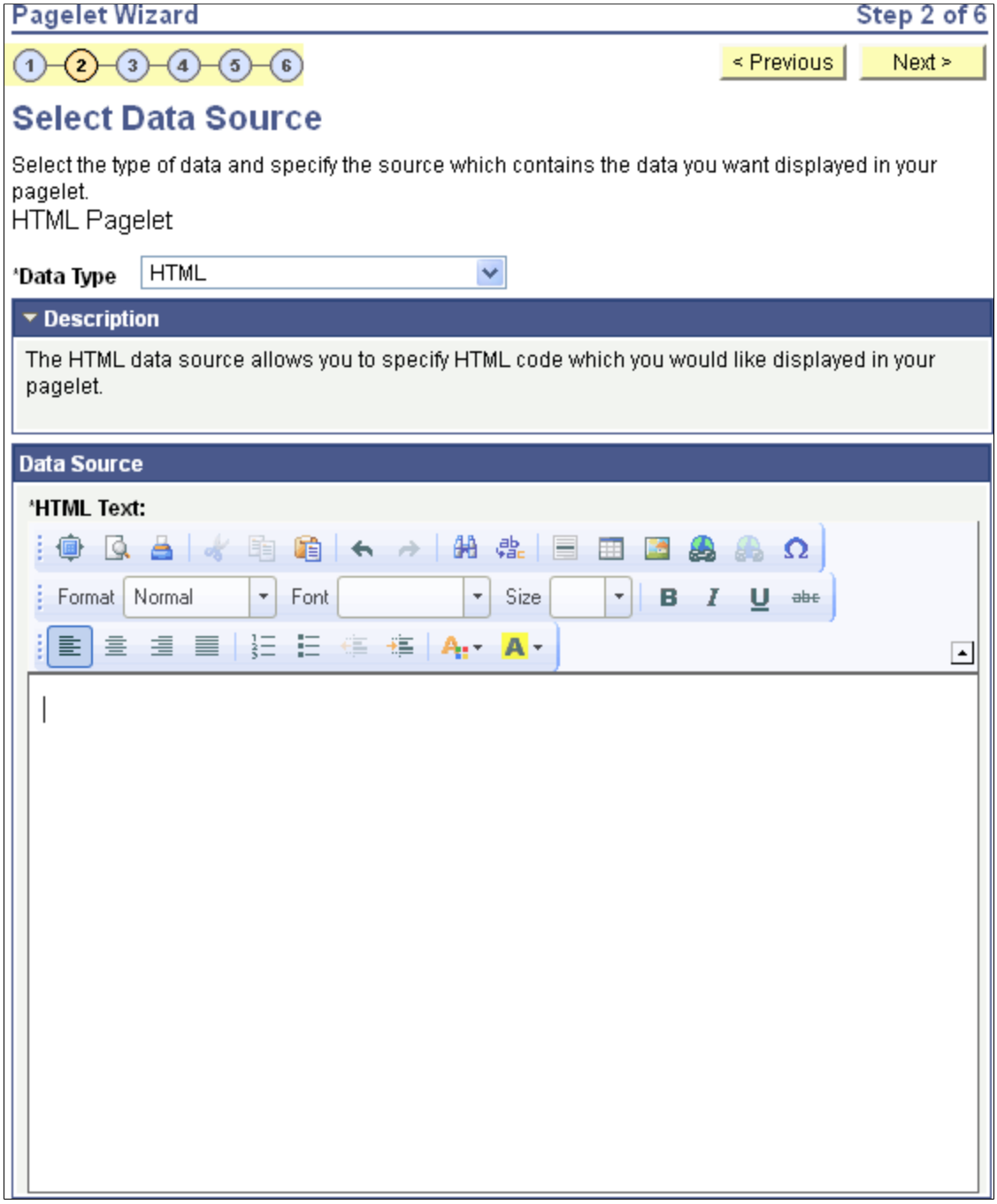
Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select *HTML* in the Data Type field.

Image: Select Data Source page (HTML data source)

This example illustrates the fields and controls on the Select Data Source page for an HTML data source. You can find definitions for the fields and controls later on this page.



Use the HTML data source to present simple, formatted HTML in a pagelet.

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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

However, complex HTML code that specifies custom CSS, JavaScript, or embedded media might not appear correctly in a pagelet after the rich text editor post-processes the entered data. In these circumstances, use the Free Text data source instead.

Ensure that the HTML that you enter generates content that is appropriately sized for pagelet display.

Note: Source code mode is disabled in the rich text editor on this field. This is for security reasons because malicious code could be inserted in source code mode. If you want to enter more complex HTML code, use the Free Text data source instead.

Note: To activate the Next button, click the Save button.

When you use HTML as the data source for a pagelet, there are no configurable data source parameters. Consequently, no end-user personalization exists 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.

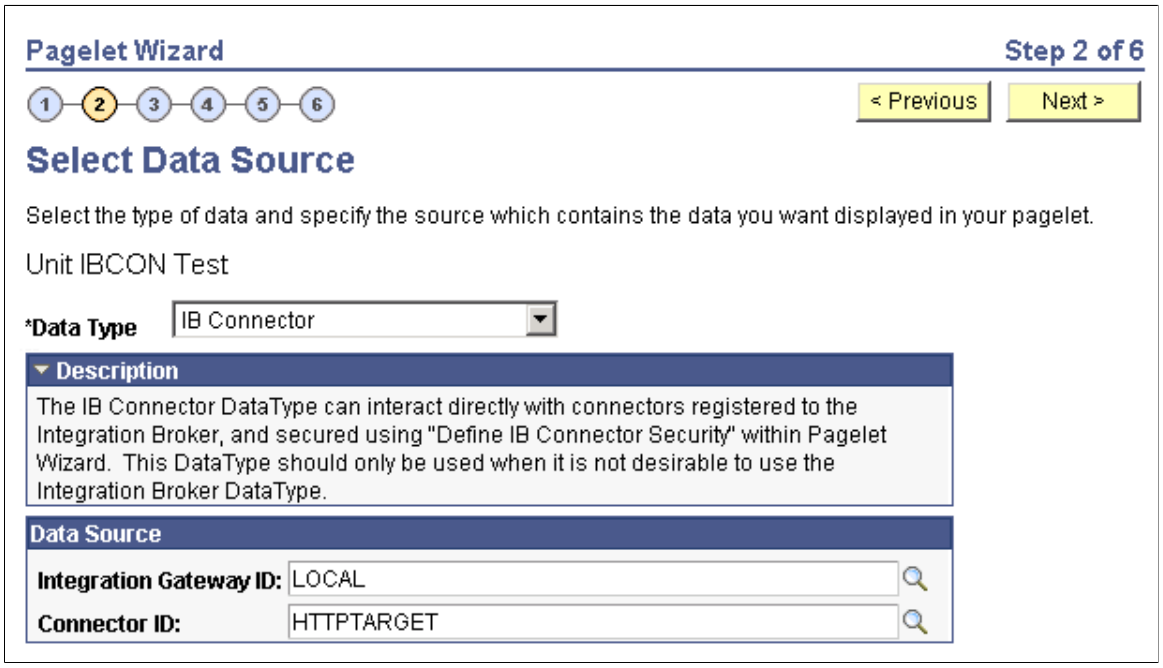
Selecting an IB Connector Data Source

Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard.) On the Specify Pagelet Information page, click the Next button

Select *IB Connector* in the Data Type field.

Image: Select Data Source page (IB Connector data source)

The following screenshot illustrates an example to display IB connector data source on the Select Data Source page.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Data Source

Integration Gateway ID Select the Integration Broker gateway that contains the connector that you want to use to supply content to the pagelet.

Connector ID Select the connector that you want to use to supply content to the pagelet.

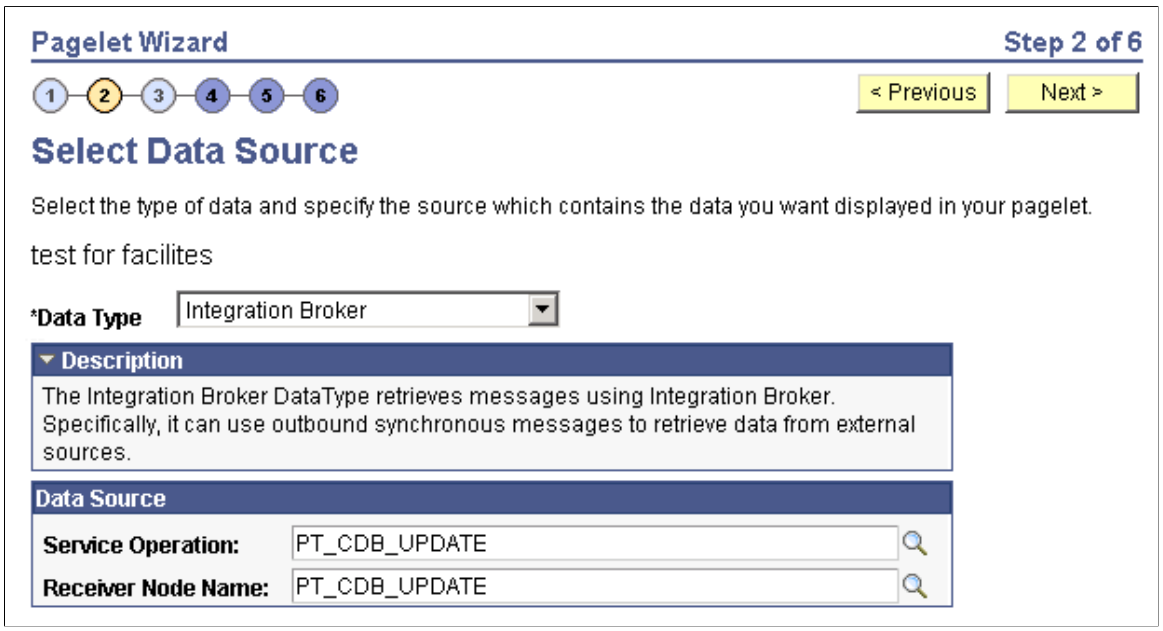
Selecting an Integration Broker Data Source

Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Select *Integration Broker* in the Data Type field.

Image: Select Data Source page (Integration Broker data source)

The following screenshot illustrates an example to display an integration broker data source on the Select Data Source page.



You can use an outbound synchronous message defined in the local database as a data source for a Pagelet Wizard pagelet. Outbound, synchronous messages are messages for which a request is made to a target system and a response is expected in real time.

The message can be either structured or unstructured. If the message is structured, then a record structure has been associated with the outbound message and that record structure is translated into data source parameters during step 3. If the message is unstructured, then no record structure has been associated with the outbound message, so no data source parameters can be supplied and Pagelet Wizard bypasses step 3.

Typically, because outbound synchronous messages are designed to run in real time, they are associated with transactions with good performance records. As with all pagelet data sources, you should understand that pagelet performance and homepage performance are tied to the performance of the underlying Integration Broker message and target transaction. You should review the performance of the target transaction before creating an Integration Broker message-based pagelet.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Data Source

Service Operation Select the requesting service operation that you want to be the data source for the pagelet.

Receiver Node Name Select the node that is associated with the selected requesting service operation.

Data Source Details

If the selected message is structured, the Data Source Details group box displays input and output records and fields that are defined in the message structure.

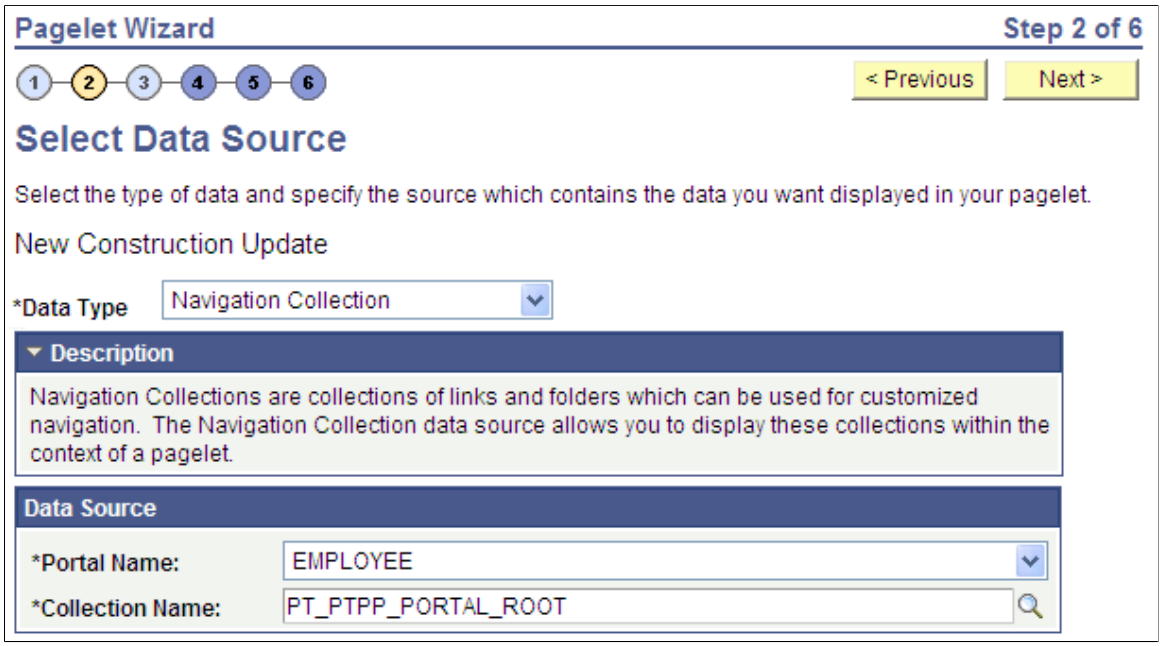
Selecting a Navigation Collection Data Source

Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Select *Navigation Collection* in the Data Type field.

Image: Select Data Source page (Navigation Collection data source)

The following screenshot illustrates an example to display an navigation Collection data source on the Select Data Source page.



The navigation collection that you use as the data source must already have been created using the navigation collections pages. 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.

Note: Although you can create a navigation collection pagelet directly from the navigation collections component, pagelets created using this component cannot be maintained in Pagelet Wizard. Only navigation collection pagelets created specifically with Pagelet Wizard can be maintained in Pagelet Wizard.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Understanding Navigation Collections.

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.

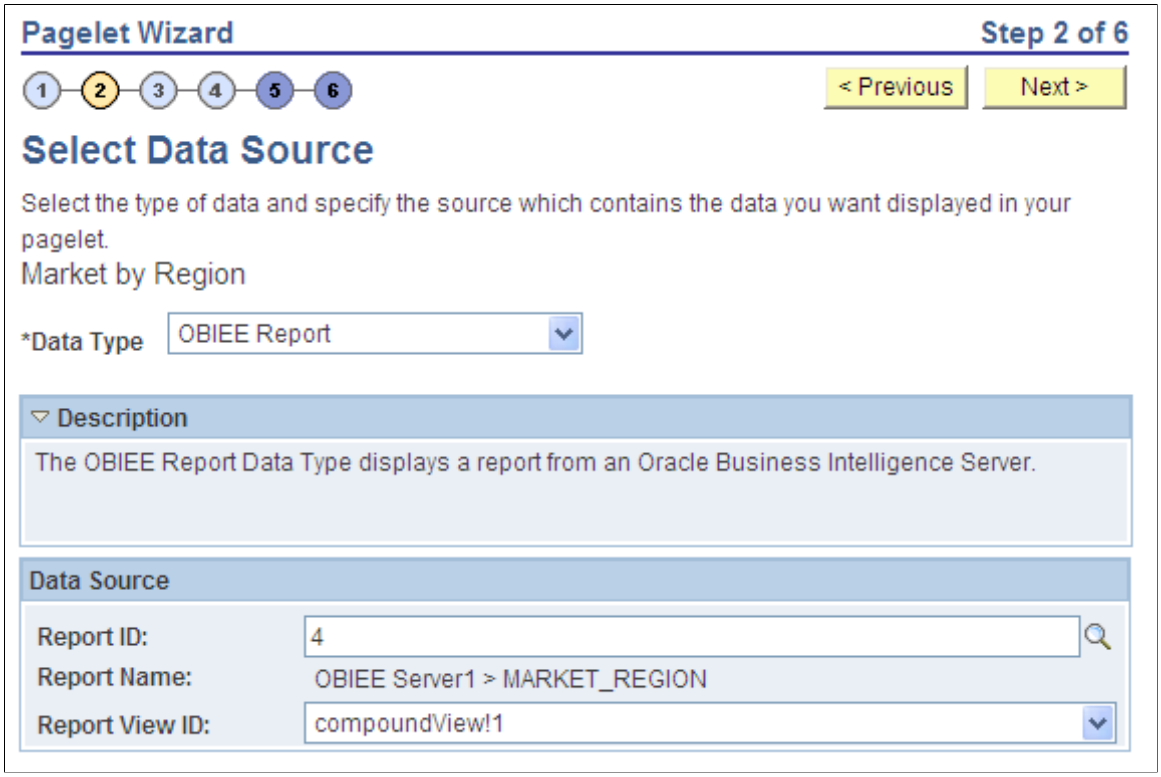
Selecting an OBIEE Report Data Source

Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Select *OBIEE Report* in the Data Type field.

Image: Select Data Source page (OBIEE Report data source)

The following screenshot illustrates an example to display OBIEE report data source on the Select Data Source page.



Data Source

Report ID Select the report that you want to be the data source. Only reports that have been loaded into the PeopleSoft application appear in this list.

See Loading OBIEE Report Definitions.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Report Name This field displays the OBIEE server name and the report name. This field is display-only.

Report View ID Select the view or report type as created in the OBIEE application.

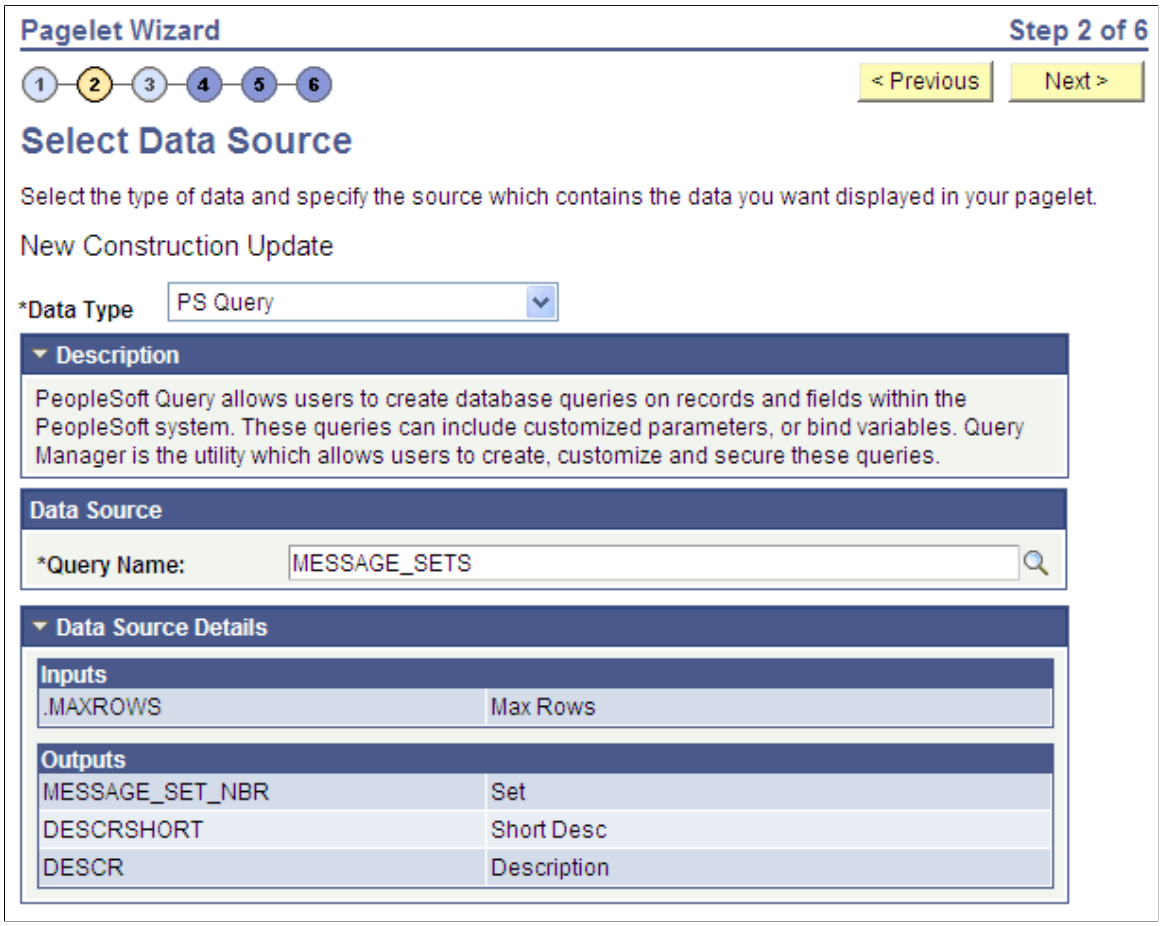
Selecting a PeopleSoft Query Data Source

Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Select *PS Query* in the Data Type field.

Image: Select Data Source page (PS Query data source)

The following screenshot an example to display a PeopleSoft query data source on the Select Data Source page.



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.

Note 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 "Creating New Queries" (PeopleTools 8.54: Query).

Note: Even though a query is marked as public, PeopleSoft Query still applies security to the underlying records. Ensure that the users of this pagelet have query security access to the underlying records to avoid runtime errors.

Data Source Details

Inputs

Displays the data source parameter fields that correspond to the query's input prompts. This field 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 a Pivot Grid Data Source

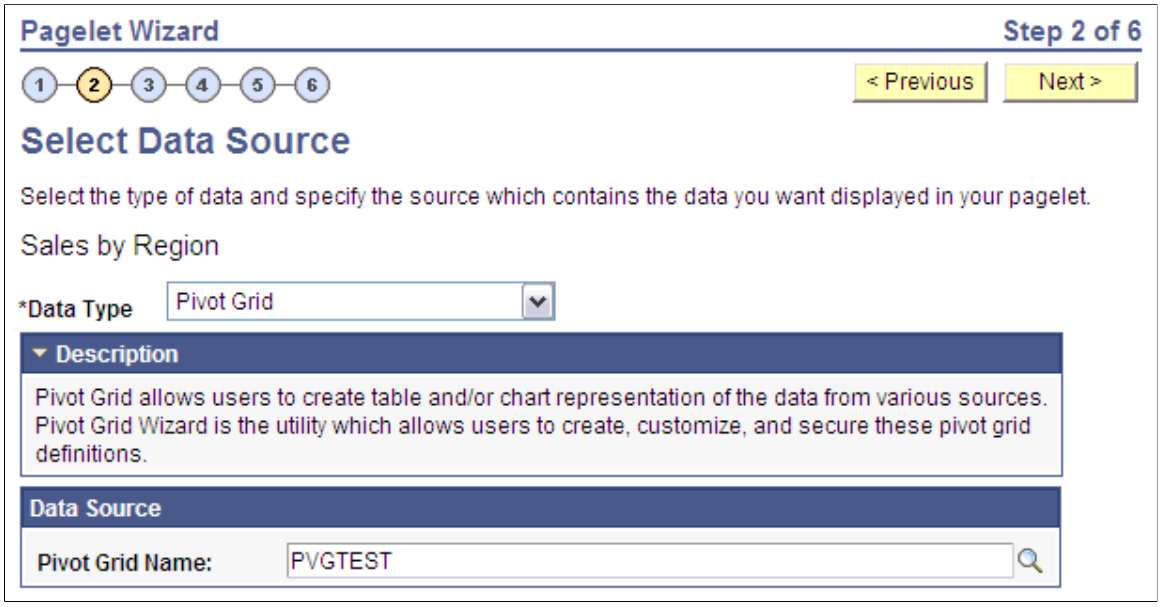
Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select *Pivot Grid* in the Data Type field.

Image: Select Data Source page (Pivot Grid data source)

The following screenshot illustrates an example to display a pivot grid data source on the Select Data Source page.



Pivot Grid is a PeopleSoft component that enables you to display application data in interactive pivot tables and charts so that you can pivot, slice, and dice your application data across multiple dimensions. Pivot grids are ideal data sources for pagelets that you use on dashboards and homepages as a form of operational reporting.

Data Source

Pivot Grid Name Select the pivot grid that you want the pagelet to display.

Related Links

"PeopleSoft Pivot Grid Implementation" (PeopleTools 8.54: Pivot Grid)

Selecting a Rowset Data Source

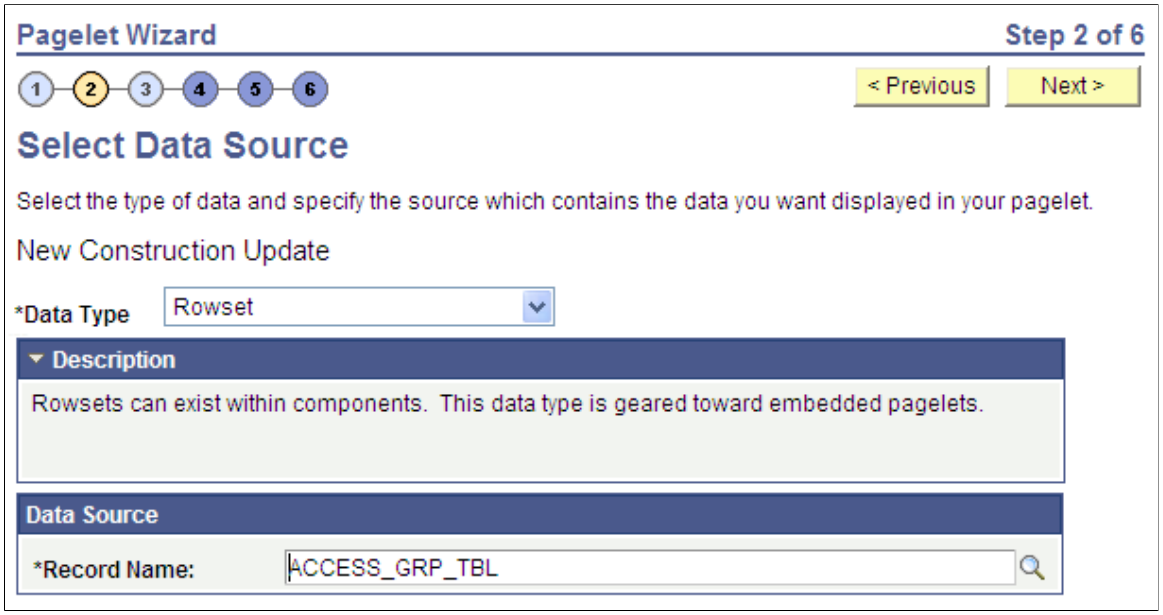
Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select *Rowset* in the Data Type field.

Image: Select Data Source page (Rowset data source)

The followings example illustrates the fields and controls on the Select Data Source page for a Rowset data source.



A rowset is a record that collects rows of data and can contain data from one or more records. The rowset data source is geared towards supplying data to embedded pagelets. This is because embedded pagelets require that a populated rowset be passed into the data source for it to be able to display the data as a chart or table, for example.

Embedded pagelets do not allow for customizable parameters that enable criteria selection. Consequently also, no end-user personalization for rowset data source pagelets.

When creating a rowset data source pagelet, you skip step 3 and proceed to step 4, in which you select your pagelet display format.

Data Source

Record Name Select the record that contains the layout of the rowset that you want to use to supply your pagelet with data. Records available for selection are records that are tables or views.

Selecting a Search Record Data Source

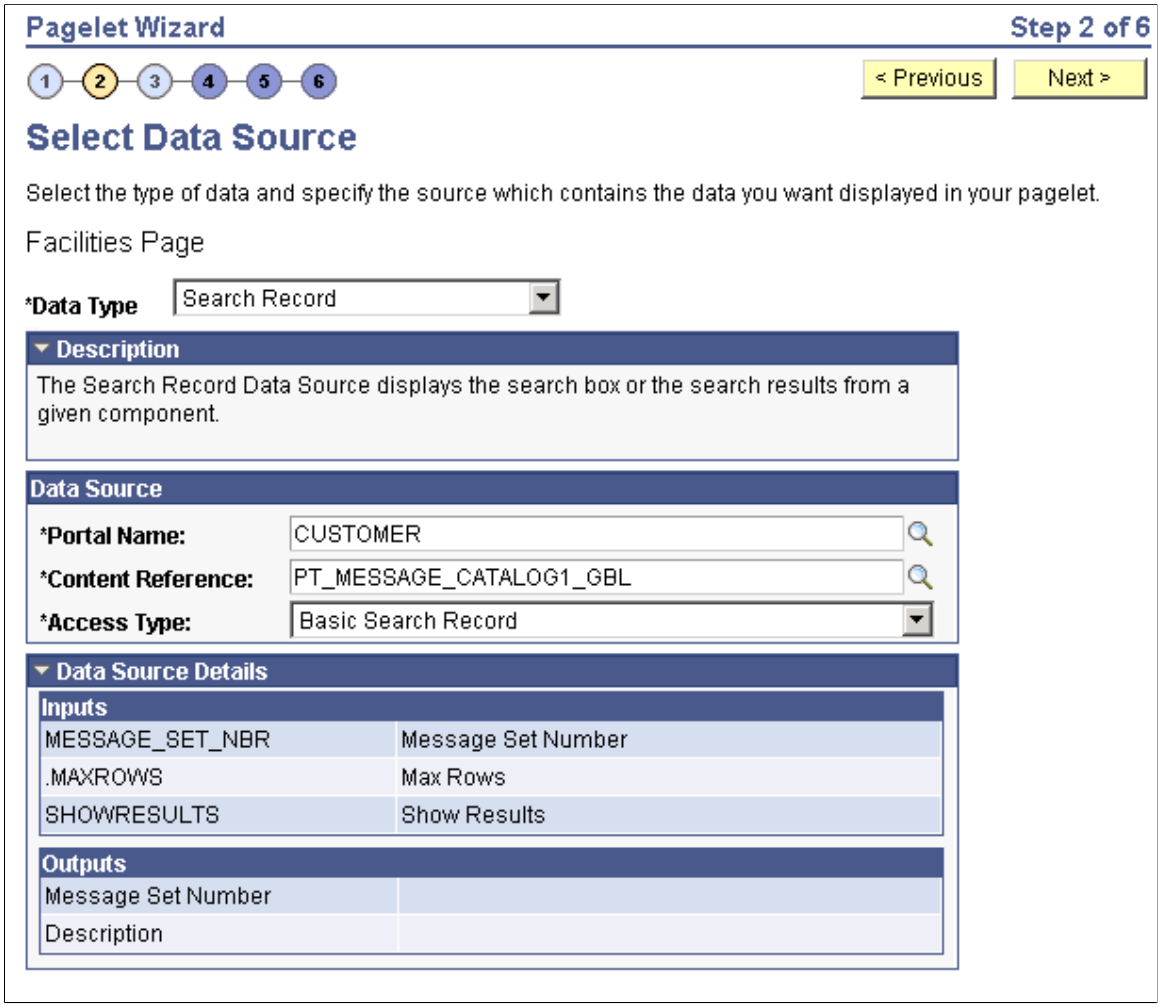
Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select *Search Record* in the Data Type field.

Image: Select Data Source page (Search Record data source)

The following screenshot illustrates an example to display search record data source on the Select Data Source page.



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 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,

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 "Creating Component Interface Definitions" (PeopleTools 8.54: Component Interfaces)"Implementing PeopleSoft Component Interfaces" (PeopleTools 8.54: Component Interfaces).

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 URL Data Source

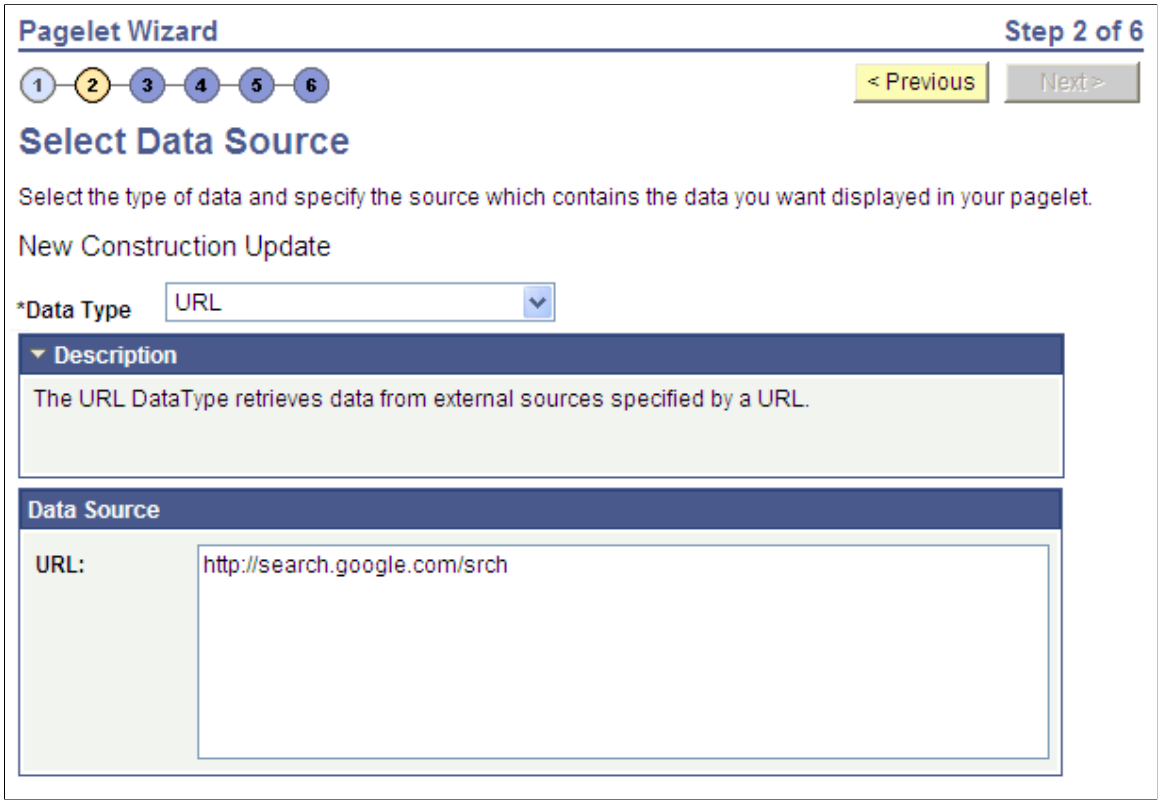
Access the Select Data Source page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Pagelet Information page, click the Next button.)

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select *URL* in the Data Type field.

Image: Select Data Source page (URL data source)

The following screenshot illustrates an example to display URL data source on the Select Data Source page.



URL

Enter any HTTP-accessible URL as the data source for the pagelet. The URL may or may not require query string parameters that can be specified and personalized as end-user prompts within Pagelet Wizard.

For example, the Yahoo! search page URL takes the form http://search.yahoo.com/search?p=searchterm. If you want to make searchterm a user-specified data parameter, enter a URL field value of *http://search.yahoo.com/search.* In step 3 on the Specify Data Source Parameters page, add a data source parameter Field Name value of *p* and set its Usage Type field value to *User Specified* with a default value. Pagelet Wizard then adds the *?p=searchterm* value, with the *searchterm* value able to be personalized by the user, to the URL before it retrieves the content to the pagelet.

Step 3: Specifying Pagelet Data Source Parameters

This section provides an overview of system variables supported as data source parameters, lists common elements, and discusses specifying pagelet data source parameters. This is step 3 in the Pagelet Wizard pagelet creation process. Specifically, this section discusses how to:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Specify activity guide data source parameter values.
- Specify free text data source parameter values.
- Specify HTML data source parameter values.
- Specify IB connector data source parameters.
- Specify Integration Broker data source parameters.
- Specify navigation collection data source parameters.
- Specify OBIEE data source parameters.
- Specify pivot grid data source parameters.
- Specify query data source parameters.
- Specify rowset data source parameters.
- Specify search record data source parameters.
- Specify URL data source parameters.
- Specify search record data source parameters.

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

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- %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:

- *Not Used:* Select if it will not be used on the pagelet.
- *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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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.

- *Admin Specified:* Enables users with administrative privileges to specify variables for this field as well as access the Configure link on the published pagelet and select from those parameters for users.

- *Context Sensitive:* Enables users to specify context sensitivity for activity guide, OBIEE, and embedded pagelets.

Default Value

If you select *User Specified* or *Admin 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, this field is populated with that default value.

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 Understanding System Variables Supported as Data Source Parameters.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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

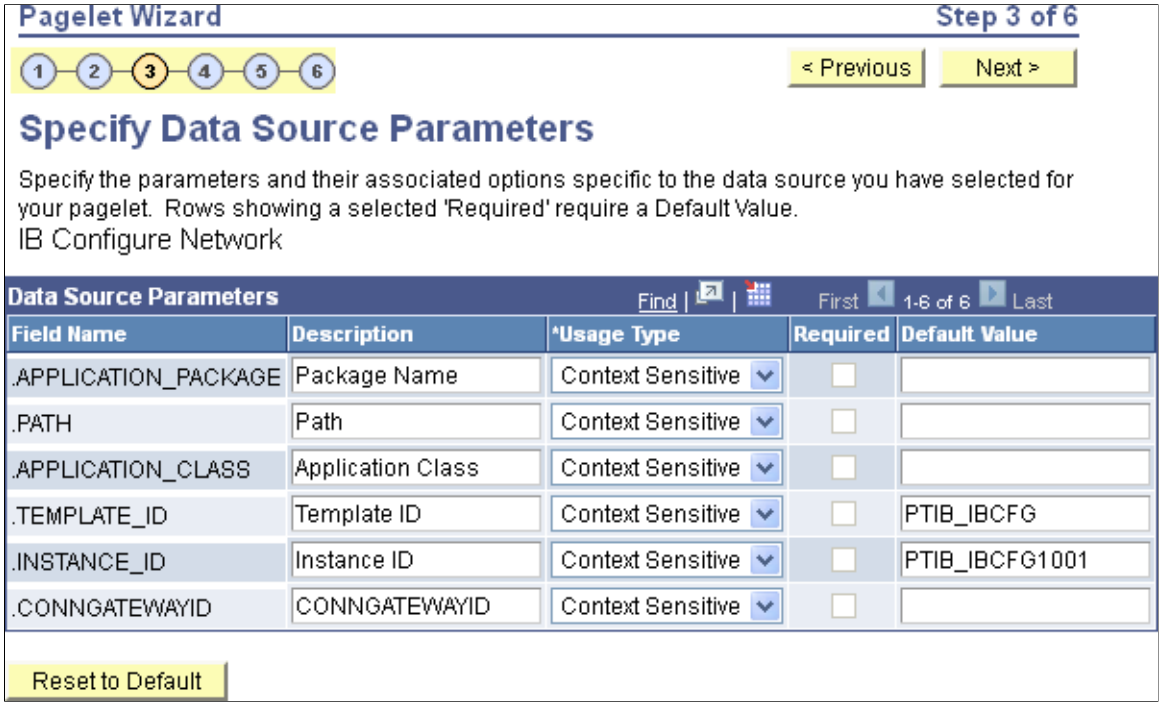
If you select *User Specified* as the usage type, click to access the Specify Data Source Parameter Values page.

See Step 3: Specifying Pagelet Data Source Parameters.

Specifying Activity Guide Data Source Parameters

Image: Specify Data Source Parameters (Activity Guide data source)

This example illustrates the fields and controls on the Specify Data Source Parameters for an activity guide data source *when generating a pagelet for a specific activity guide instance.* You can find definitions for the fields and controls later on this page.



When creating an activity guide pagelet, do one of the following:

- If you are creating an activity guide pagelet for a specific activity guide instance, specify the template ID at step 2 and specify the instance ID on this page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- If you are creating an activity guide pagelet for an activity guide template with the instance ID to be supplied at run time, specify the template ID at step 2 and skip this page.
- If you are creating a generic activity guide pagelet with both the template ID and instance ID to be supplied at run time, specify no template ID at step 2 and skip this page.

Data Source Parameters

.TEMPLATE_ID	Specified in step 2 when specifying the data source.
.INSTANCE_ID	Specify an instance ID *only when you want to generate a pagelet for a specific activity guide instance.*

Related Links

Generating an Activity Guide Pagelet

Specifying Free Text Data Source Parameters

When creating an free text data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

Specifying HTML Data Source Parameters

When creating an HTML data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying IB Connector Data Source Parameters

After selecting *IB Connector* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.)

Image: Specify Data Source Parameters (IB Connector data source) (1 of 2)

The following screenshot illustrates an example to display IB connector data source parameters on the Specify Data Source Parameters page.

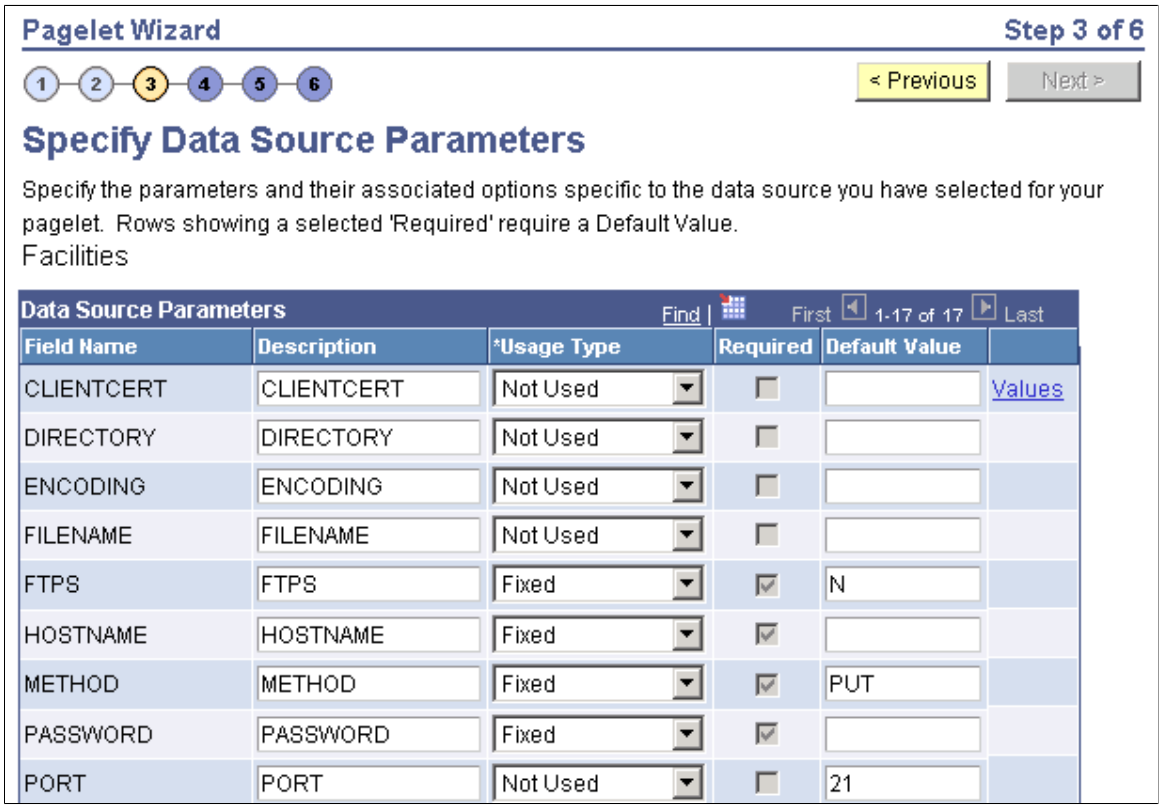


Image: Specify Data Source Parameters (IB Connector data source) (2 of 2)

The following screenshot continues to illustrate an example to display an IB Connector data source.

Define the data source parameters for the IB Connector data source. Click the User Values button to set the default values for users.

See "Understanding PeopleSoft Integration Broker" (PeopleTools 8.54: Integration Broker).

See Specifying User-Defined Data Source Paramters.

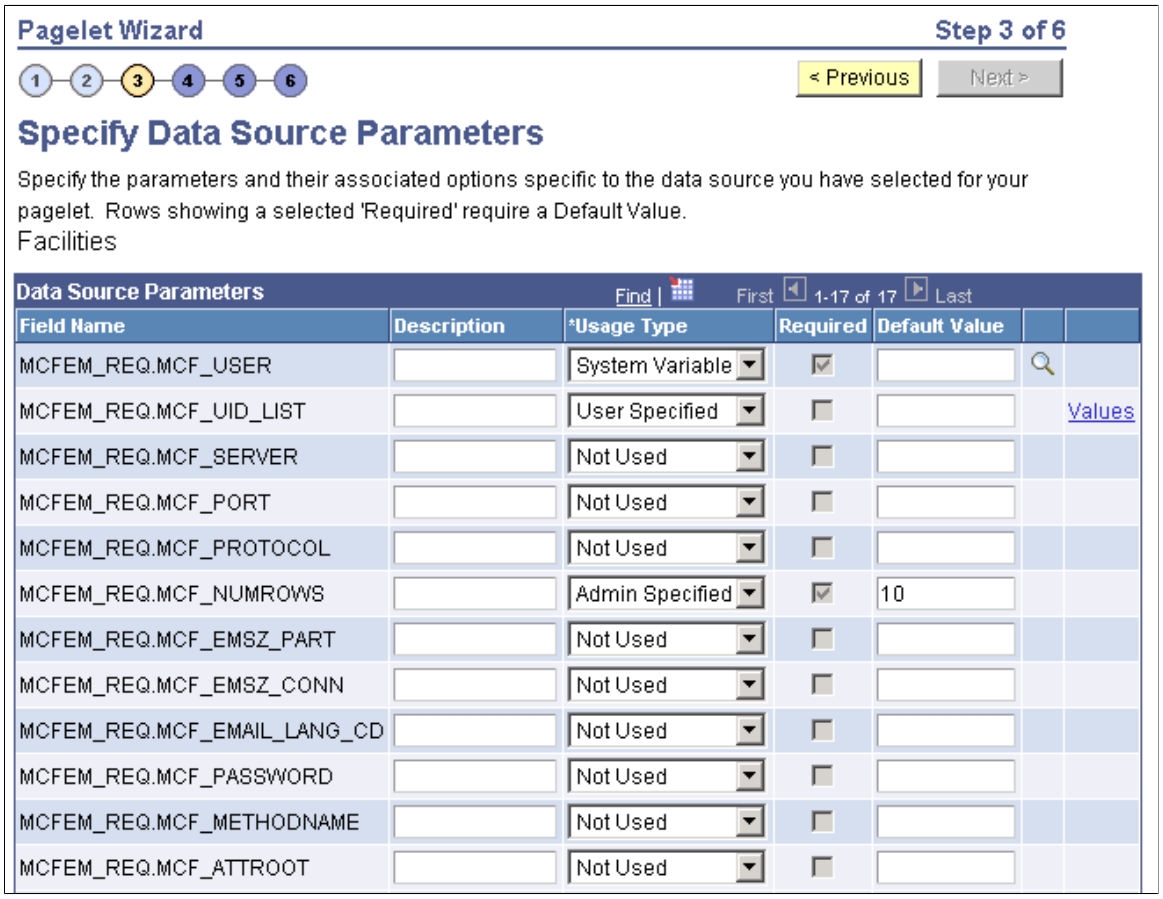
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Integration Broker Data Source Parameters

After selecting *Integration Broker* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.).

Image: Specify Data Source Parameters (Integration Broker data source)

The following screenshot illustrates an example to display IB data source parameters on the Specify Data Source Parameters page.



This step is applicable for the Integration Broker data source if the service operation is based on a structured message or if the message is structured but contains a schema. The structure that is associated with the message is translated into data source parameters on this page.

If the message is unstructured, then no schema structure has been associated with the message. Therefore, no data source parameters can be supplied and Pagelet Wizard bypasses this step.

You must supply values for all rows with the Required option selected.

Personalization Instructions

Use the Personalization Instructions text box . These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

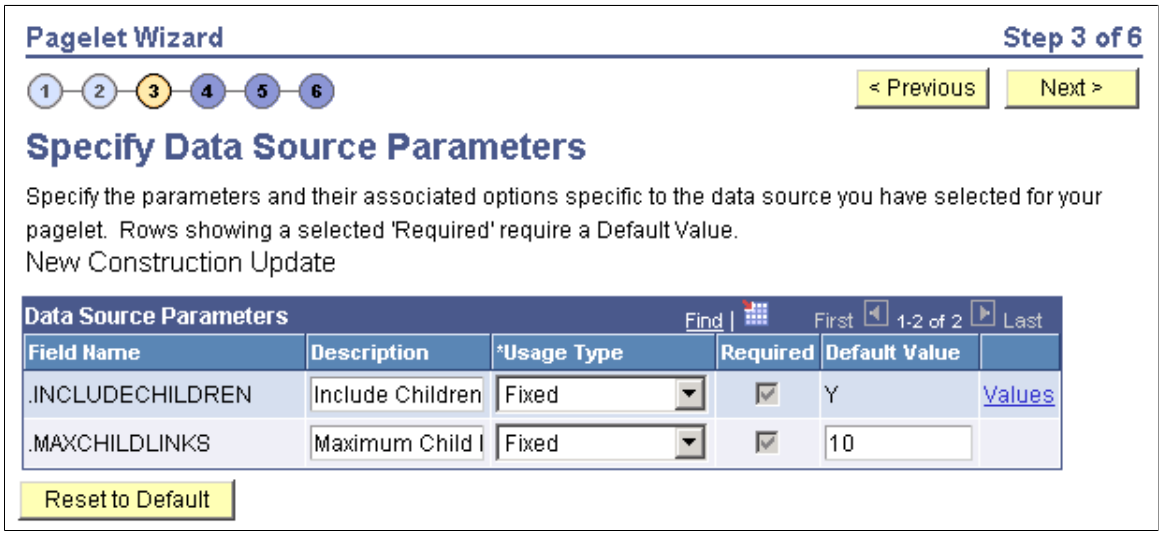
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Navigation Collection Data Source Parameters

After selecting *Navigation Collection* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.)

Image: Specify Data Source Parameters (Navigation Collection data source)

The following screenshot illustrates an example to display navigation collection data source parameters on the Specify data source parameters page.



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 is 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.

Values

Click the Values link to set the default values for users.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See "Understanding PeopleSoft Integration Broker" (PeopleTools 8.54: Integration Broker).

Personalization Instructions

Use the Personalization Instructions text box to enter instructional informs for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

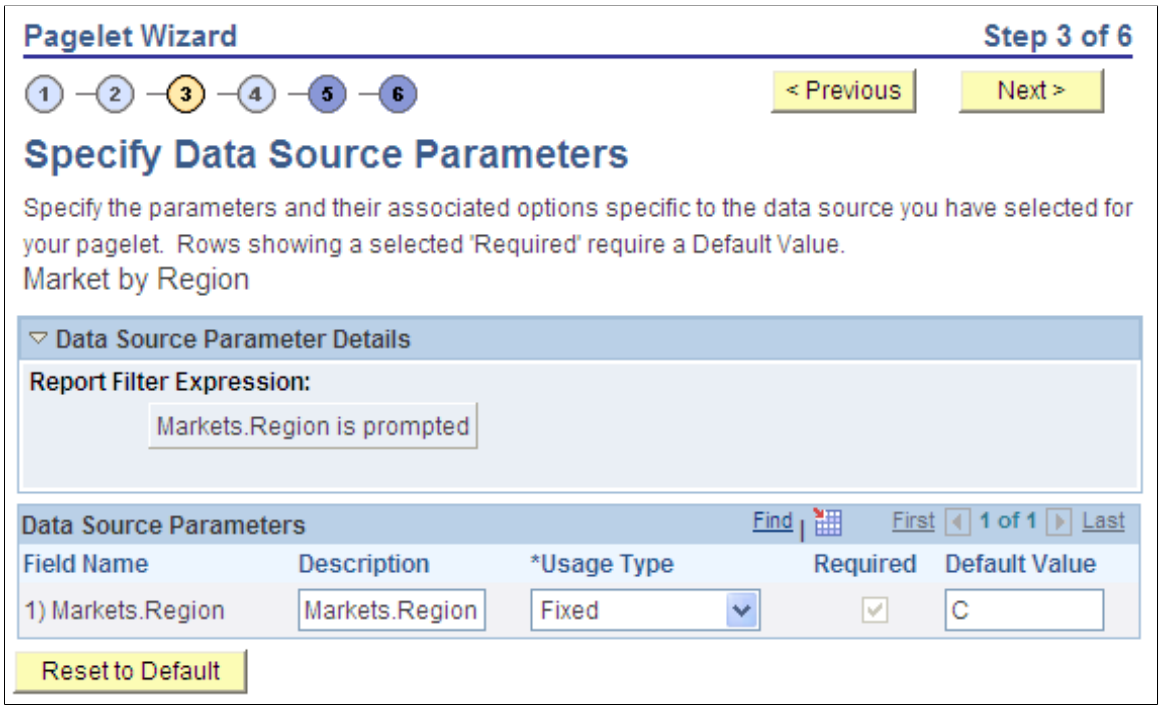
Note: Personalization instructions must be translatable.

Specifying OBIEE Data Source Parameters

After selecting *OBIEE Report* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.)

Image: Specify Data Source Parameters page (OBIEE Report data source)

The following screenshot illustrates an example to display OBIEE data source parameters on the Specify Data Source Parameters page.



Data Source Parameters

Pagelet Wizard analyzes the selected OBIEE report 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 OBIEE prompts associated with the selected OBIEE report. This field 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Description This field contains a description that identifies the purpose or location of the OBIEE 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 field is required.

If this option is selected and is display-only, the value is set in the application class at the application programming interface (API) level.

If you leave this option deselected, the data source parameter value can be cleared at runtime, or it can be entered to refine the data selection criteria.

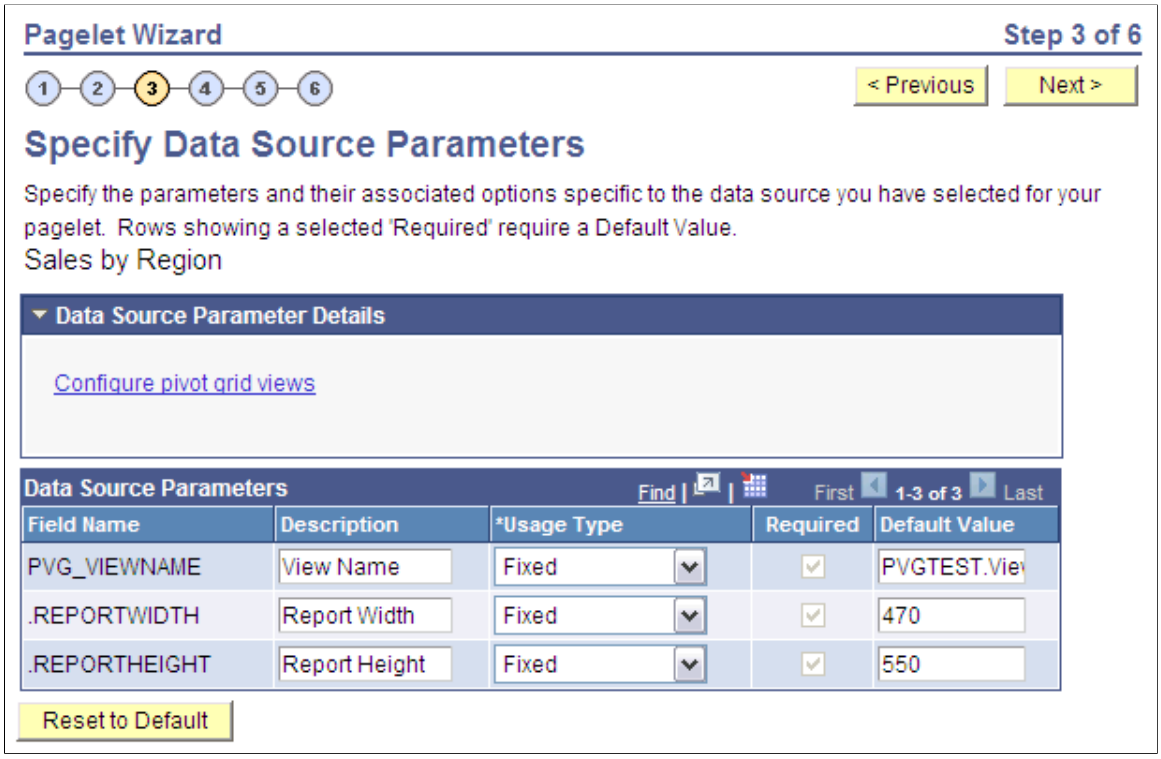
Reset to Default Click to reset the values on this page to the default values specified by the OBIEE data source API.

Specifying Pivot Grid Data Source Parameters

After selecting *Pivot Grid* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.)

Image: Specify Data Source Parameters page (Pivot Grid data source)

The following screenshot illustrates an example to display pivot grid data source parameters on the Specify Data Source Parameters page.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Data Source Parameter Details

Displays the Configure pivot grid views link, which you click to open the Pivot Grid Views Component page where you can define the pivot grid options.

See "Specifying Display Options" (PeopleTools 8.54: Pivot Grid).

Data Source Parameters

Pagelet Wizard analyzes the selected pivot grid and generates the list of pagelet parameters shown in this step. You cannot add to the list of parameters.

Description Enter text that describes the field.

Usage Type Select the way the system uses the parameter. Select one of these options:

- *Admin Specified*
- *Context Sensitive*
- *Fixed*
- *Not Used*
- *System Variable*
- *User Specified*

Required Select to require a value for the parameter.

Default Value Enter the parameter value that the system uses by default.

Reset to Default Click to set all values on this page to the default values specified by the pivot grid data source API.

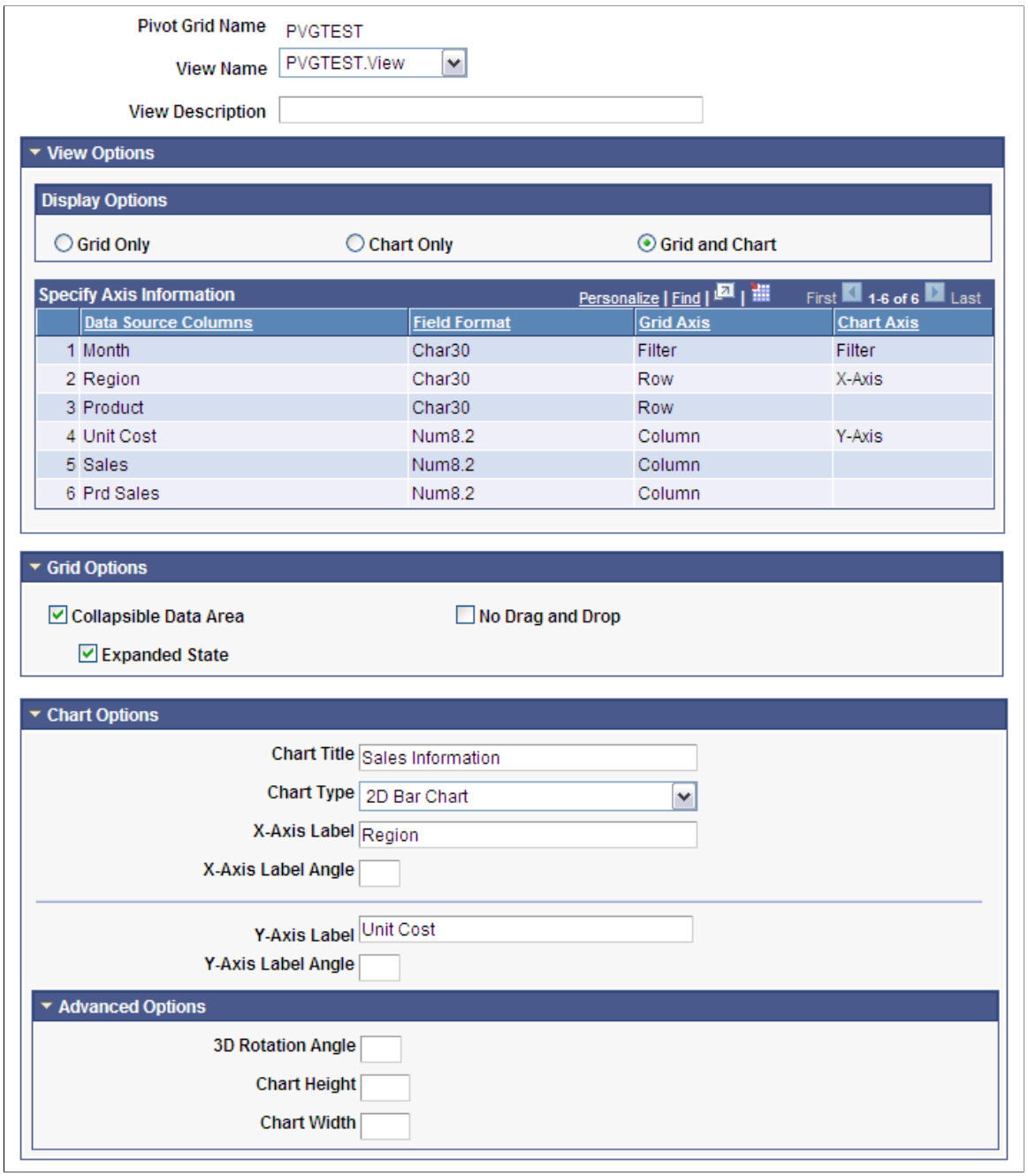
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pivot Grid Views Component

Access the Pivot Grid Views page (Click the Configure pivot grid views link on the Specify Data Source Parameters page).

Image: Pivot Grid Views page

The following screenshot illustrates an example to display pivot grid views component on the Pivot Grid Views page.



View Name Select the pivot grid view to use for this pagelet.

View Description Enter text that describes the pivot grid view.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Display Options	Select which elements appear on the pagelet. Choose one of these values: • *Grid Only* • *Chart Only* • *Grid and Chart*
Grid Options	• Collapsible Data Area- select to enable the user to collapse the pivot grid into the grid header. If you select this option, the system automatically selects the Expanded State check box. • Expanded State - select to display the grid in the expanded state when the pagelet initially appears on the homepage or dashboard. • No Drag and Drop - select to disable the drag and drop functionality of elements within the pivot grid.
Chart Options	Enter or select values for the title, type, x and y-axis labels and x and y-axis angles (in degrees) of the chart.
Advanced Options	Enter values for the three-dimensional rotation angle (in degrees), height (in pixels), and width (in pixels) of the chart.
Save As	Click to open the Save View As dialog box, where you can either save the current pivot grid model as a new view or update an existing view.

Related Links

"Pivot Grid Viewer Overview" (PeopleTools 8.54: Pivot Grid)

"Specifying Data Source Parameters" (PeopleTools 8.54: Pivot Grid)

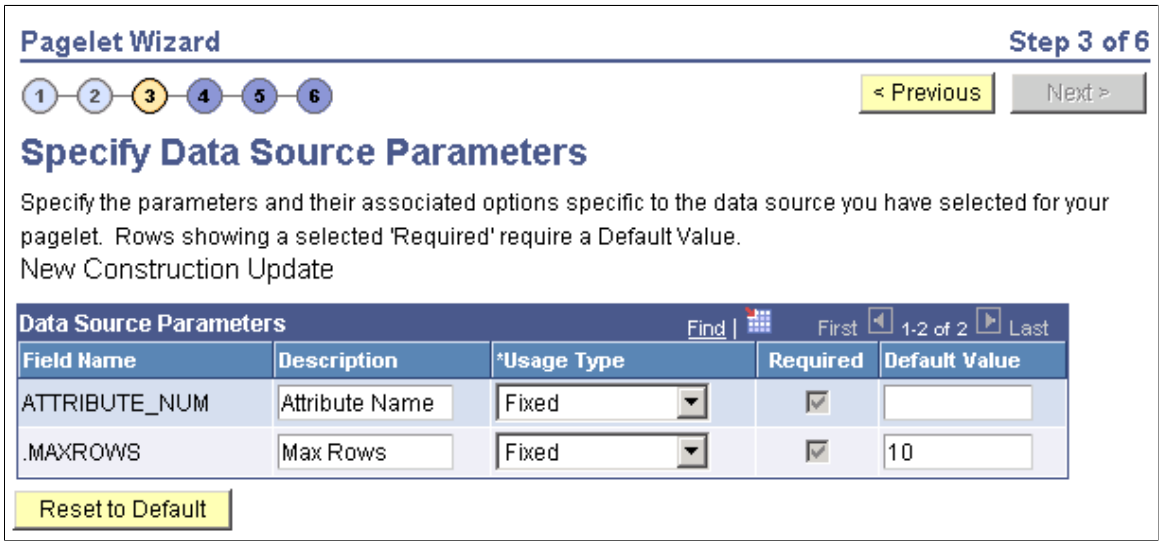
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Query Data Source Parameters

After selecting *PS Query* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.)

Image: Specify Data Source Parameters page (PeopleSoft Query data source)

The following screenshot illustrates an example to display query data source parameters on the Specify Data Source Parameters page.



Data Source Parameters

Pagelet Wizard analyzes the selected query and translates the query prompts into the list of pagelet parameters shown in this step. You cannot add to the list of parameters. Most prompts will be marked as required; however, any prompts that are used exclusively in the IN clause of the query will not be marked as required.

Important! Query Prompts that are based on dynamic views should not be used as User Specified data source parameters. Unexpected results will occur.

You can attain advanced functionality with proper design of the query and the Pagelet Wizard pagelet. For example, you might want to create a pagelet that will show open purchase orders for a list of vendors, and allow each user to specify up to five vendors. To achieve this functionality, you would first create a query to retrieve the open purchase orders for up to five vendors. When you create this query, ensure that the five vendor ID prompts are used only in the IN clause of the query. You can then create a pagelet using the PS Query data type and choose the query that you have created. The five vendor ID prompts will be shown as data source parameters. However, they will be marked as not required. End users will then be able to specify any number of vendors (up to five) when they personalize the pagelet. Finally, update the personalization instructions on the pagelet to explain to the end users that they can supply up to five vendor IDs.

Field Name Displays the data source parameters that correspond to the query prompts associated with the selected query. This field also displays the required system-supplied .MAXROWS query data

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	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 deselected, the data source parameter value can be cleared at runtime, or it can be entered to refine the data selection criteria. Prompts tied to the IN clause of the query will not be marked as required.
Reset to Default	Click to reset the values on this page to the default values specified by the query data source API.
Values	Click the Values link to set the default values for users.

Personalization Instructions

Use the Personalization Instructions text box to enter custom personalization instructions for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

Specifying Rowset Data Source Parameters

When creating a rowset data source pagelet, you skip step 3 and proceed to step 4, in which you select your pagelet display format.

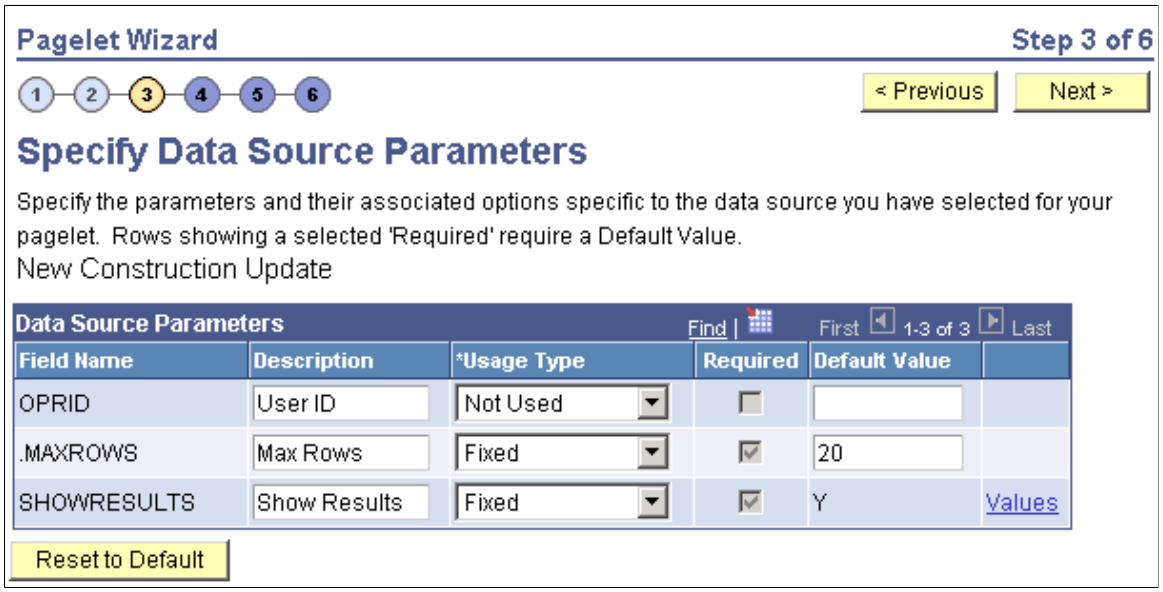
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Search Record Data Source Parameters

After selecting *Search Record* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.)

Image: Specify Data Source Parameters (Search Record data source)

The following screenshot illustrates an example to display search record data source parameters on the Specify Data Source Parameters page.



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. This field 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 box values that are available on the pagelet are the search keys for the search record.

If data source parameter default key field values have been defined, then 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 deselected, the data source parameter value can be omitted 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.

Values

Click the Values link to set the default values for users.

Personalization Instructions

Use the Personalization Instructions text box to enter custom personalization instructions for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

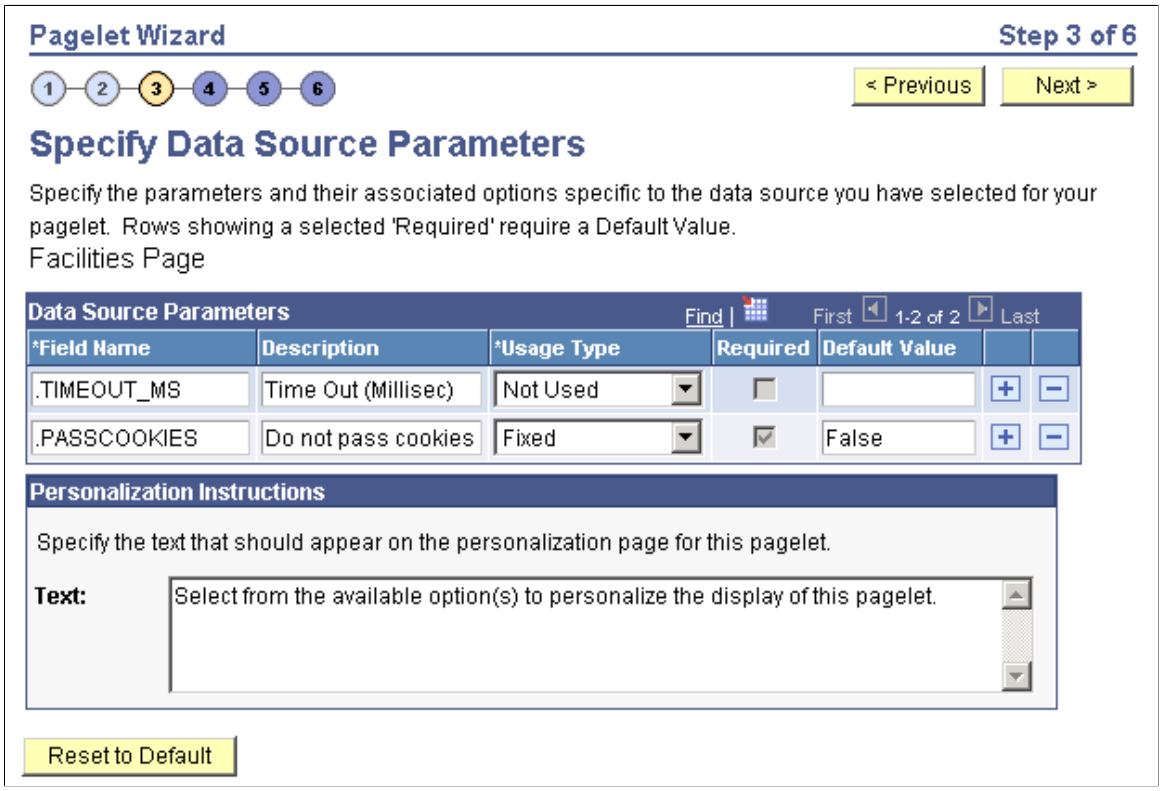
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying URL Data Source Parameters

After selecting *URL* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Data Source page, click the Next button.)

Image: Specify Data Source Parameters page (URL data source)

The following screenshot illustrates an example to display URL data source parameters on the Specify Data Source Parameters page.



The data source parameters that you can define on this page correspond to URL query parameters, such as an HTTP GET request.

An HTTP GET request is a URL type for which parameters are placed on the query string separated by ? and &, such as http://search.yahoo.com/search?p=searchterm.

For example, the Yahoo! search page URL takes the form http://search.yahoo.com/search?p=searchterm. If you want to make searchterm a user-specified data parameter, enter a URL field value of *http://search.yahoo.com/search* in step 2 on the Select Data Source page. In step 3 on this page, add a data source parameter Field Name value of *p* and set its Usage Type field value to *User Specified* with a default value. Pagelet Wizard then adds the *?p=searchterm* value, with the *searchterm* value able to be personalized by the user, to the URL before it retrieves the content to the pagelet.

You can specify a parameter for the number of milliseconds that a data source will wait for a response before timing out on the homepage. When you set *.timeout_ms,* you can then enter the number of milliseconds that the system will wait.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Passing Cookies

When the URL request is in the AuthenticationToken domain, then the URL data source will pass all cookies on the request to retrieve the content. (The only exception to this rule is if the request is being made back to the local webserver. In that instance, only the PS_TOKEN cookie will be passed.) Any requests to servers outside of the AuthTokenDomain will not have any cookies passed.

To turn off the passing of cookies other than PS_TOKEN, add the parameter .passcookies as a Fixed usage type with a default value of False. Adding this parameter with a value of False will instruct pagelet wizard to pass only the PS_TOKEN cookie (and no other cookies) for requests within the AuthenticationToken domain. Adding the .passcookies parameter with a value of True is not recommended because this is the same as the default behavior.

Note: Make sure to examine any security impact that passing cookies might present in your configuration before setting this parameter.

Personalization Instructions

Use the Personalization Instructions text box to enter custom personalization instructions for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

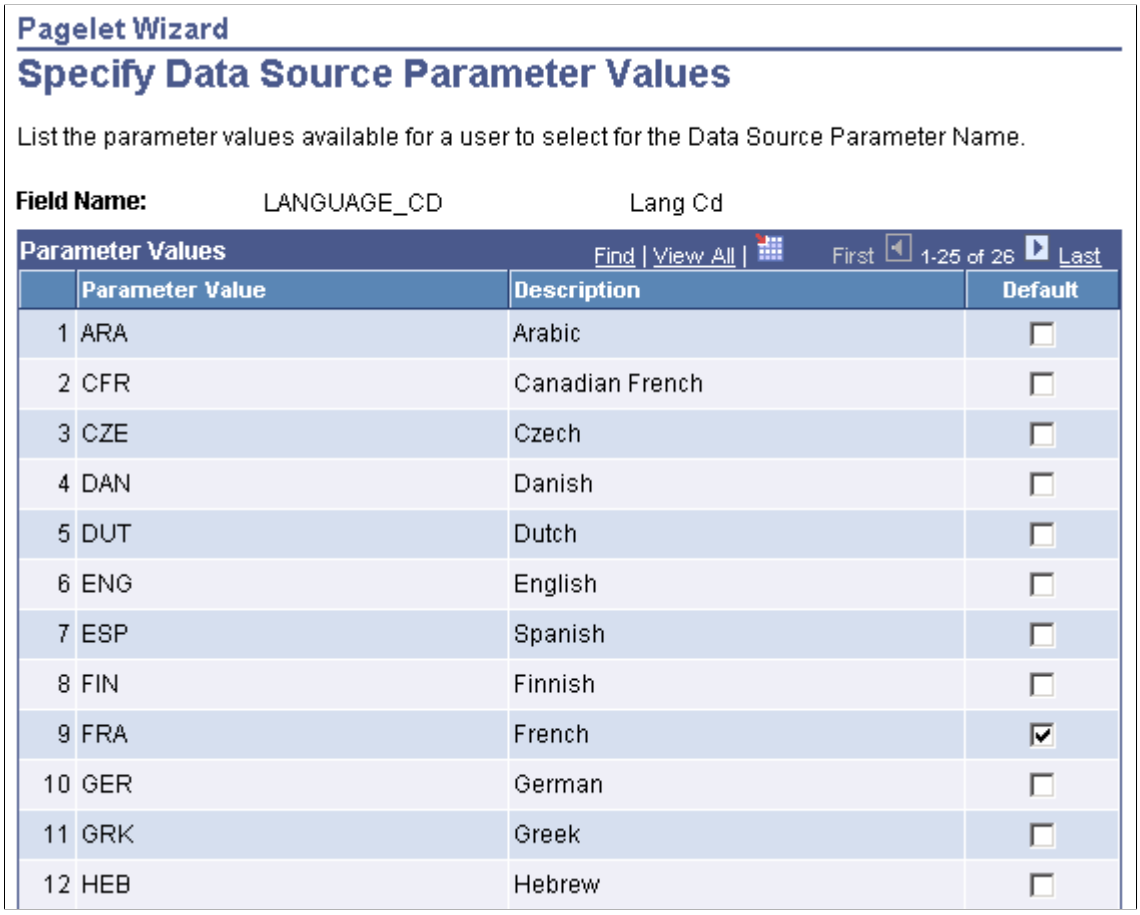
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying User-Defined Data Source Paramters

Click the User Values button on the Specify Data Source Parameters page to display the Specify Data Source Parameters Values page.

Image: Specify Data Source Parameter Values page

The following screenshot illustrates an example to display user defined data source parameters on the Specify Data Source Parameter Values page.



	Parameter Value	Description	Default
1	ARA	Arabic	☐
2	CFR	Canadian French	☐
3	CZE	Czech	☐
4	DAN	Danish	☐
5	DUT	Dutch	☐
6	ENG	English	☐
7	ESP	Spanish	☐
8	FIN	Finnish	☐
9	FRA	French	☑
10	GER	German	☐
11	GRK	Greek	☐
12	HEB	Hebrew	☐

Select the parameter value to be displayed to the user as the default value for the field. Users can select from this list of valid values when they personalize the pagelet.

Step 4: Selecting a Pagelet Display Format

This section discusses how to select a display format. This is step 4 in the Pagelet Wizard pagelet creation process.

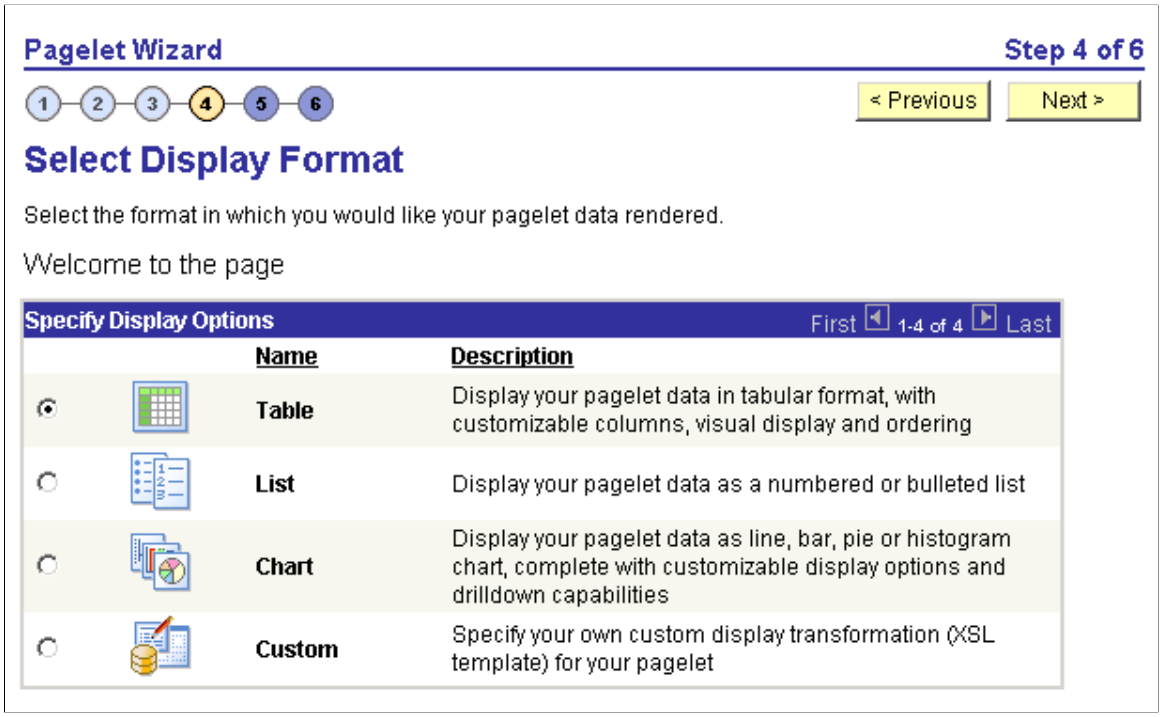
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting a Display Format

Access the Select Display Format page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Specify Data Source Parameters page, click the Next button.)

Image: Select Display Format page

The following screenshot illustrates an example to show a display format on the Select Display Format page.



Select the format to use for the pagelet. The display formats that are available for selection on this page depend on the data source that you have selected on the Select Data Source page. Options are:

Table	Available for query and rowset data source pagelets.
List	Available for query and rowset data source pagelets.
Chart	Available for query and rowset data source pagelets.
Menu	Available for Navigation Collection data source pagelets.
Search List	Available for search record data source pagelets.
Passthru	Available for activity guide, free text, HTML, OBIEE, and pivot grid data source pagelets.
Custom	Available for activity guide, free text, HTML, navigation collection, query, rowset, and search record data source pagelets.

Related Links

Step 5: Specifying Pagelet Display Options

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Step 5: Specifying Pagelet Display Options

This section lists common elements and discusses how to specify pagelet display options. This is step 5 in the Pagelet Wizard pagelet creation process. Specifically, this section 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.

Common Links

This table describes the common links used in this section.

	Click the Link Options icon to display the Link Details page, where you can associate URLs with field values, which at runtime appear on the pagelet as active links that you click to access additional relevant information. See Specifying Data Link Details.

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 Defining Headers.
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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See "Message Catalog" (PeopleTools 8.54: System and Server Administration).

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 to appear 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 "Using Administration Utilities" (PeopleTools 8.54: System and Server Administration).

Footer

Select a pagelet footer. You can select from footers that are defined on the Define Footer page.

See Defining Footers.

Search Options

The Search Options group box contains the options to override the default search functionality for the current pagelet only.

See Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types.

Search Box

Select a value from these options to determine the location of the search box field:

- *None:* Do not show the search box. This is the default option.
- *Top Left:* The search box appears at the top left corner of the pagelet, above the pagelet content but below any header and opening text.
- *Top Right:* The search box appears at the top right corner of the pagelet, above the pagelet content but below the header and opening text.
- *Bottom Left:* The search box appears at the bottom left corner of the pagelet, below the pagelet content but above any footer and closing text.
- *Bottom Right:* The search box appears at the bottom right corner of the pagelet, below the pagelet content but above any footer and closing text.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Inline XSL:* A PSSEARCHBOX node must be written into the transformation XSL to indicate where the search box appears. If multiple PSSEARCHBOX nodes are in the pagelet content after the transformation, only the first node is replaced by the search box. The PASSTHRU display format does not support this type of search box. If the display format is PASSTHRU, no search box appears in the pagelet.

Package Enter the application class package name that you want to use to define and build the search for the pagelet.

Path Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method used to define and build the search for the pagelet.

Class ID Enter the application class name that you want to use to define and build the search for the pagelet. The class must exist in the application package and subpackage that you specify.

Note: If you have not defined default Search Option values for the data type on the Define Data Types page (PeopleTools, Portal, Pagelet Wizard, Define Data Types), the Package, Path, and Class ID fields have no default values. You must enter values if you want to activate the pagelet search feature.

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 that you have defined so far. When you click Customize, you switch from a noncustom display format selected on the Select Display Format page to a custom format of your definition.

The XSL field is populated based on preexisting display options that are set for the original noncustom display format. You can edit the XSL directly in the XSL field and retain as much of the original noncustom display options as you want. Remember that if you apply an XSL template using the XSL Template field, the XSL is regenerated based on the selected template, and the original noncustom display options that 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 noncustom display format.

When you click the Customize button, the current pagelet definition is saved for restoration purposes. You can revert to this precustomization 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

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

return to the Specify Publishing Options page - Step 5, you access the version of the page for the Custom display format.

See Specifying Custom Display Options.

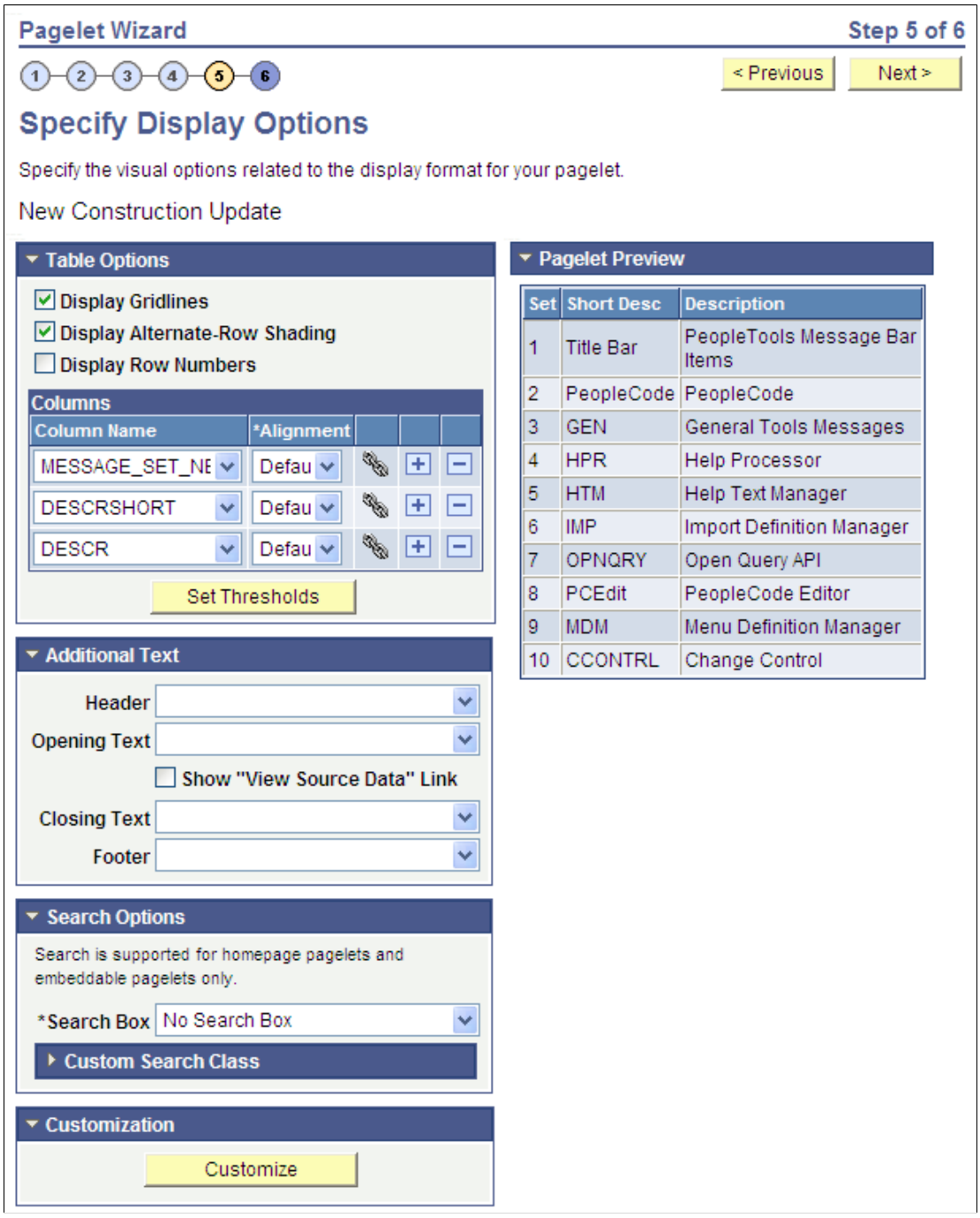
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Table Display Options

After selecting the *Table* option on the Select Display Format page, select the Specify Display Options page (PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Format page, click the Next button.)

Image: Specify Display Options page (table display format)

The following screenshot illustrates an example to show table display options on the Specify Display Options page.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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.*

Click the Link Options icon to display the Link Details page, where you can associate URLs with field values, which at runtime appear on the pagelet as active links that you click to access additional relevant information.

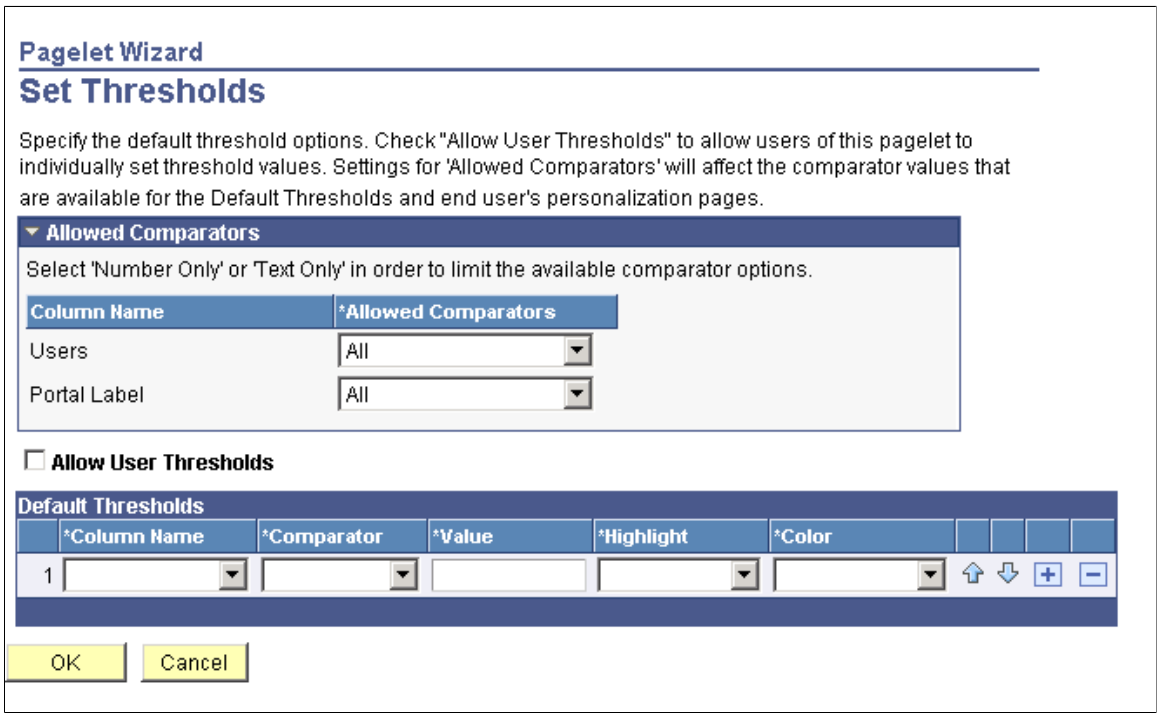
See Specifying Data Link Details.

Setting Thresholds

Access the Set Thresholds page by clicking the Set Thresholds button in the Table Options of step 5 in Pagelet Wizard.

Image: Set Thresholds page

The following screenshot illustrates an example to show how thresholds are set on the Set Thresholds page.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Use the Set Thresholds page to set the column alignment and threshold options.

Allowed Comparators	The system displays the column names and enables you to set the comparators for that column based on numbers, text, or both.
Allow User Thresholds	Select to enable users to set their own threshold values for the table from the Personalization page.

Set the display order and attributes of each column. Use the up and down arrows to change the display order of the columns, if necessary. Each column threshold can have a unique color associated with it—duplicates are allowed. If more than one threshold rule applies for a given value, the rule that is positioned highest in the list is the one that is applied.

Column Name	Select the column name from the drop-down list box.
Comparator	Select a comparator value for each column.
Value	Enter the threshold value for the column. When this value is reached, the highlight and color attributes become active.
Highlight	If you choose to have highlighting for the threshold, select either the entire row or just a single cell.
Color	Select the color for the column.
	Use the arrows to move each individual column up or down in the table.

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Passthru Display Options

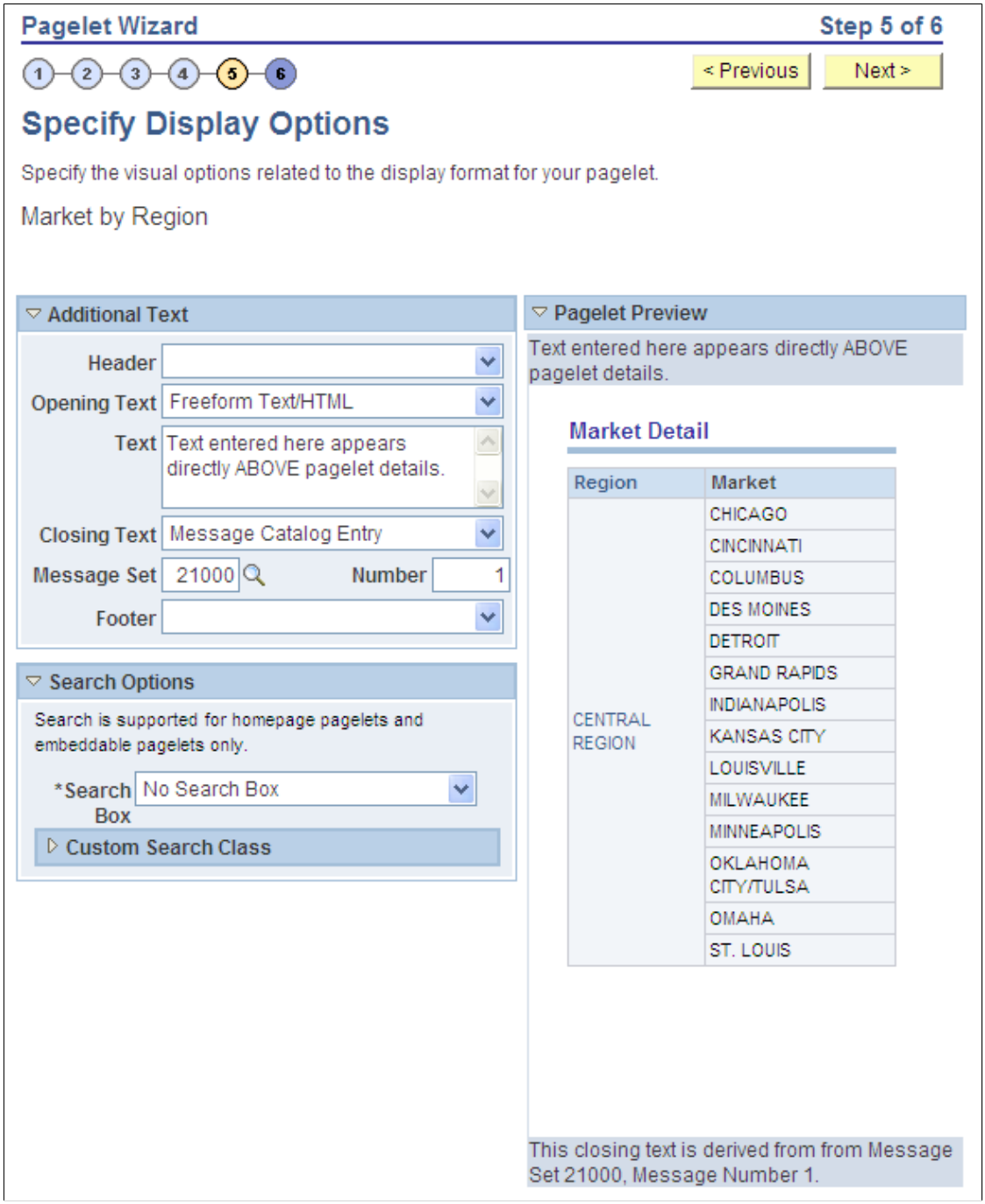
After selecting the *Passthru* option on the Select Display Format page, select the Specify Display Options page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Format page, click the Next button.)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: For OBIEE Reports and pivot grids, *Passthru* is the only display format option.

Image: Specify Display Options page (passthru display format)

The following screenshot illustrates an example to show Passthru display options on the Specify Display Options page.



Additional Text

Header

Select a value from this field to display a header below the pagelet header, but above any opening text.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Opening Text

Select a value to include text *above* the pagelet details.

Select from these values:

- *Freeform Text/HTML*

  Select to enable the Text field and manually enter static text or HTML.

- *Message Catalog*

  Select to enter the message set and message number from which to derive the opening text from the message catalog.

Text

Enter text or HTML. This field appears when you select *Freeform Text/HTML* as the opening or closing text.

Closing Text

Select a value to include text *below* the pagelet details.

Select from these values:

- *Freeform Text/HTML*

  Select to enable the Text field and manually enter static text or HTML.

- *Message Catalog*

  Select to enter the message set and message number from which to derive the opening text from the message catalog.

Footer

Select a value from this field to display a footer above the Footer bar, but below any closing text.

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

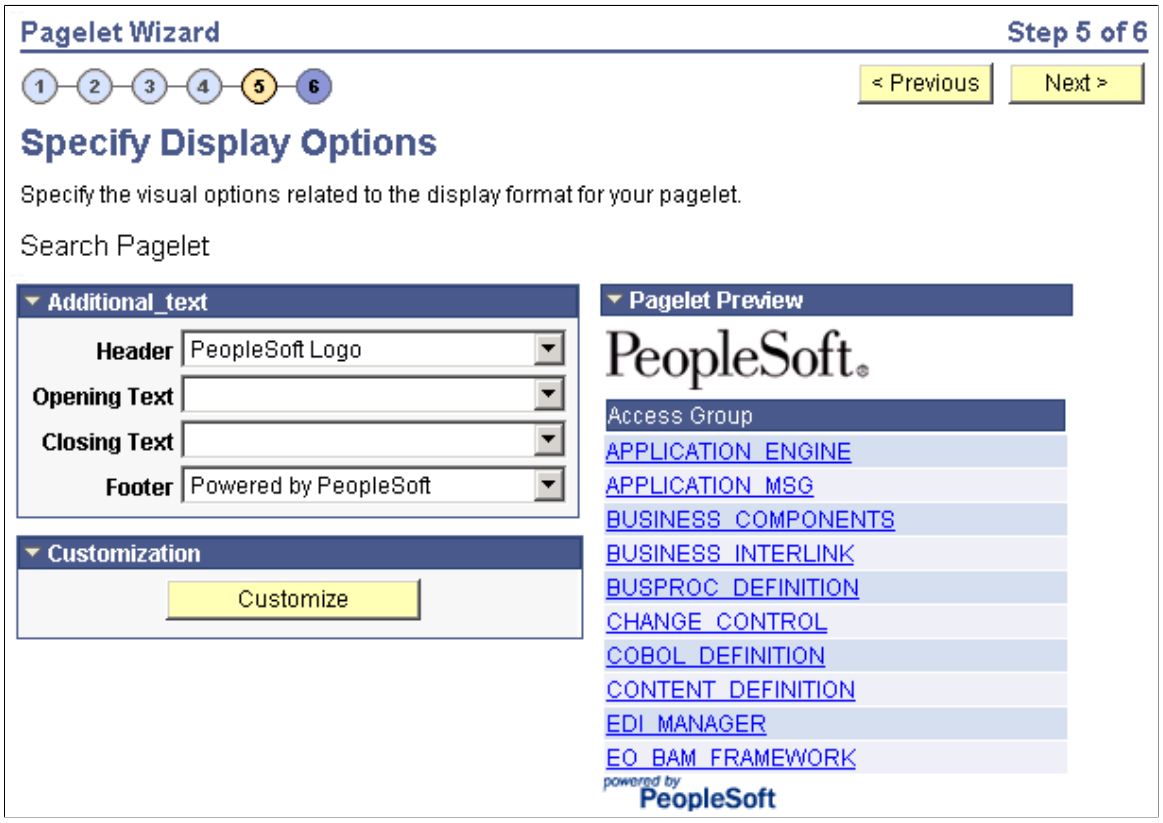
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Search List Display Options

After selecting the *Search List* option on the Select Display Format page, select the Specify Display Options page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Format page, click the Next button.)

Image: Specify Display Options page (search list display format)

The following screenshot illustrates an example to show search list display options on the Specify Display Options page.



Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

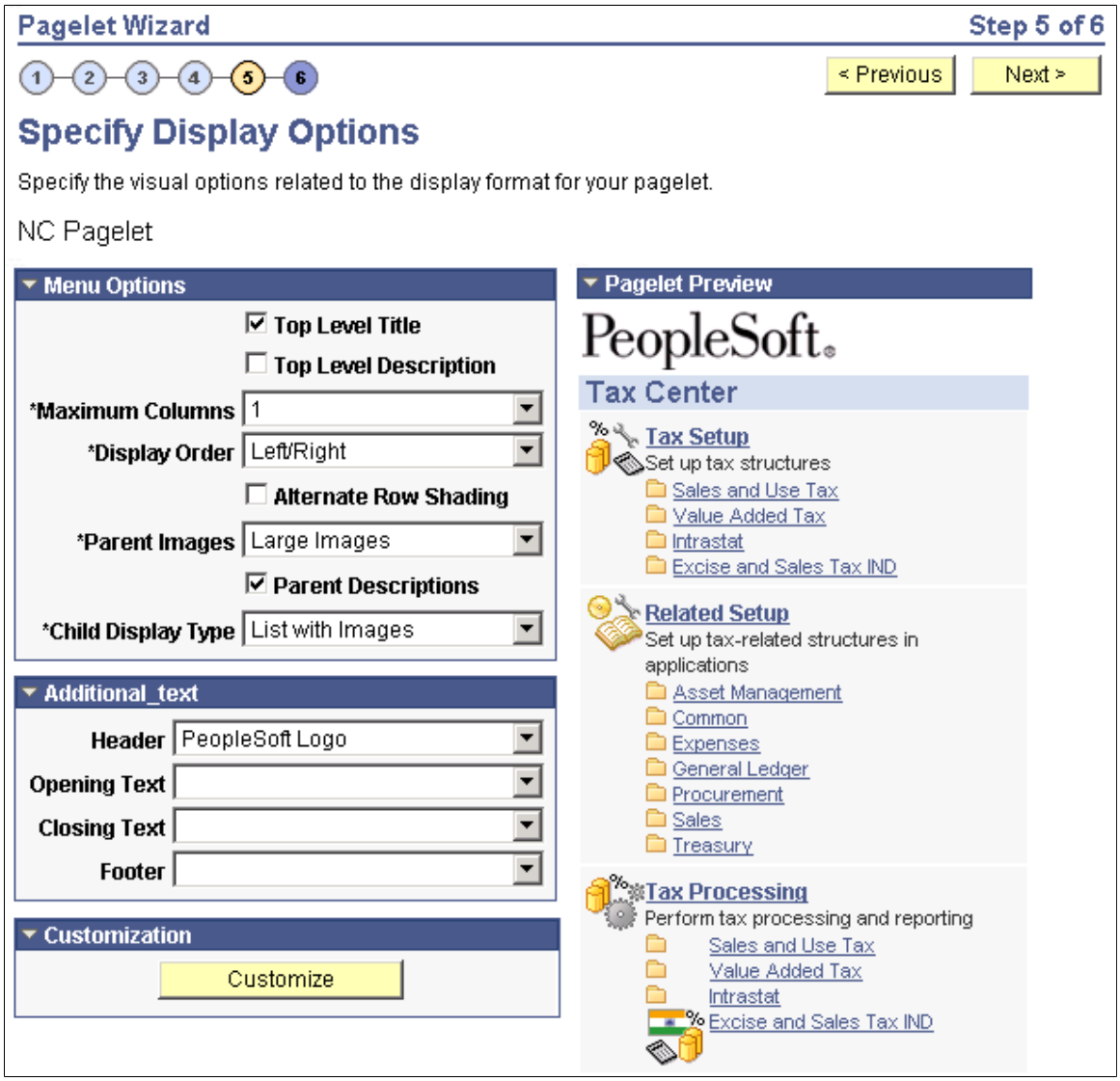
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Menu Display Options

After selecting the *Menu* option on the Select Display Format page, select the Specify Display Options page (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Format page, click the Next button.)

Image: Specify Display Options page (menu display format)

The following screenshot illustrates an example to show menu display options on the Specify Display Options page.



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

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

display contents in one column. Wide pagelets can display contents in up to four columns.

See Understanding Pagelet Wizard Display Sizes.

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 that 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Related Links

Navigation Collections

Specifying List Display Options

After selecting the *List* option on the Select Display Format page, select the Specify Display Options page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Format page, click the Next button.)

Image: Specify Display Options page (list display format)

The following screenshot illustrates an example to show list display options on the Specify Display Options page.

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 bullet list.

Numbered List: Displays query results in a numbered list.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	Simple List: Displays query results in a list with no bullets or numbers.
List Field Name	Select the field for which 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.

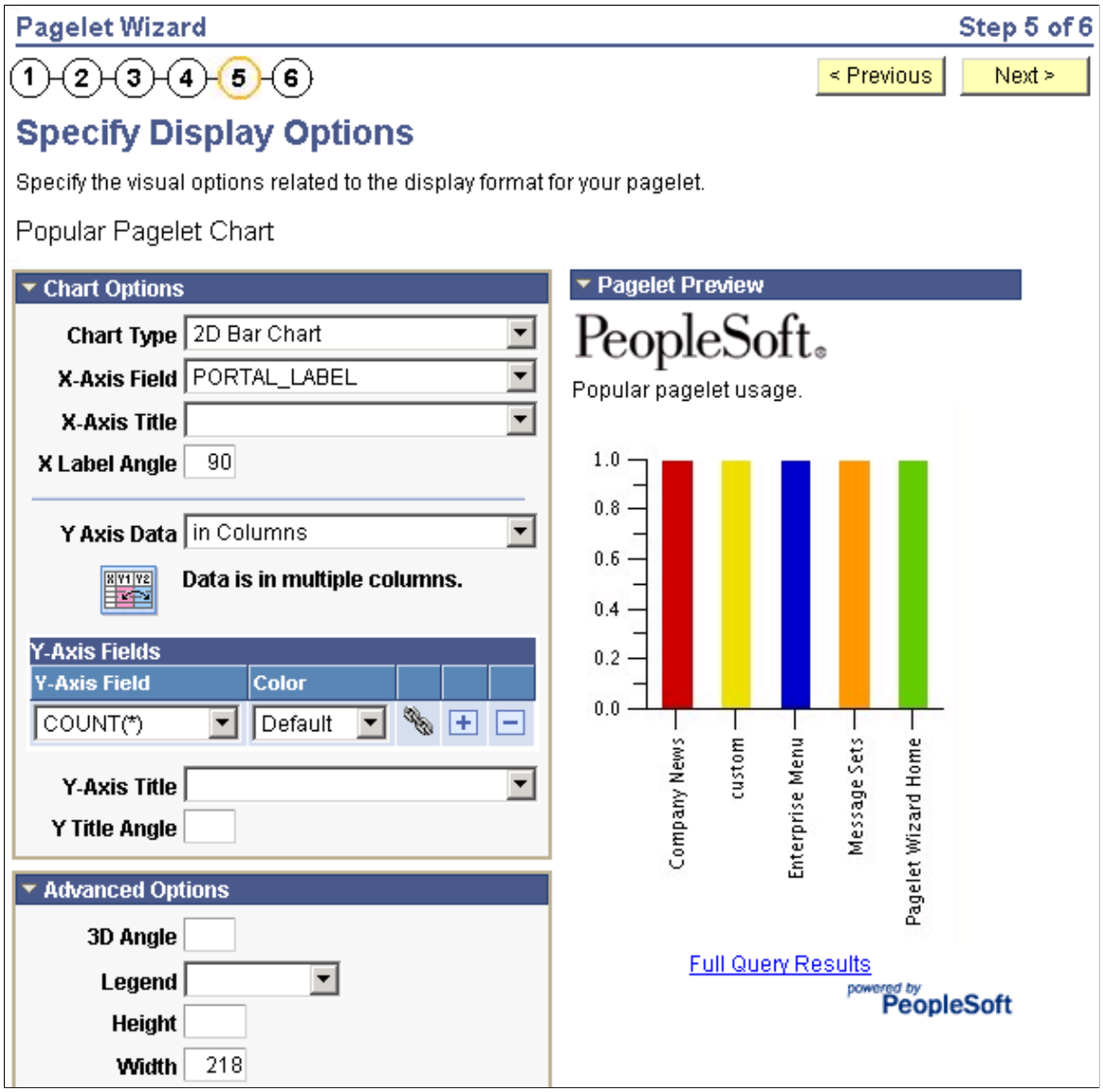
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Chart Display Options

After selecting the *Chart* option on the Select Display Format page, select the Specify Display Options page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Format page, click the Next button.)

Image: Specify Display Options page (chart display format) (1 of 2)

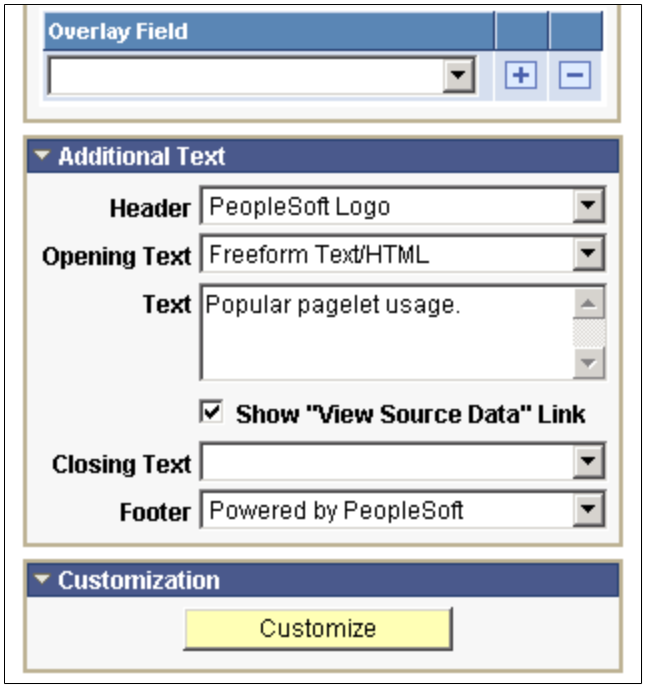
The following screenshot illustrates an example to show chart display options on the Specify Display Options page.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Image: Specify Display Options page (chart display format) (2 of 2)

The following screenshot illustrates chart display format on the Specify Display Options.



To generate a chart pagelet using Pagelet Wizard, you include information related to the chart's *x* and *y* axes. 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 might have a *y* axis that displays the number of times that a pagelet has been accessed. This access count could be displayed against an *x* axis that shows the name of the pagelet accessed.

Chart Options

Chart Type

Select the type of chart to use to display the query results on the pagelet. Available chart formats are those supported by the PeopleCode Chart class.

See "Understanding the Charting Classes" (PeopleTools 8.54: PeopleCode API Reference).

X-Axis Field

Select the field to use to derive *x*-axis values for the chart. Available values are determined by the output fields of the data source.

X-Axis Label

Select the field that you want to use to derive the labels for the x-axis values for the chart. Available values are determined by the output fields of the data source.

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.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Y Axis Data

Select to indicate whether the *y*-axis data is stored in rows or columns in the selected query. This enables 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 here:

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 output fields of the data source.

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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 *Rows.* Select the field to use 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 for the chart legend to appear. 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 Understanding Pagelet Wizard Display Sizes.

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 chart types except 2D and 3D pie charts.

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Custom Display Options

After selecting the *Custom* option on the Select Display Format page, select the Specify Display Options page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Format page, click the Next button.)

Image: Specify Display Options page (custom display format)

The following screenshot illustrates custom display format on the Specify Display Options page.

Custom Options

XSL Template

Select the XSL template to use 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 Defining Pagelet Wizard XSL Prototypes.

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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pagelet Preview

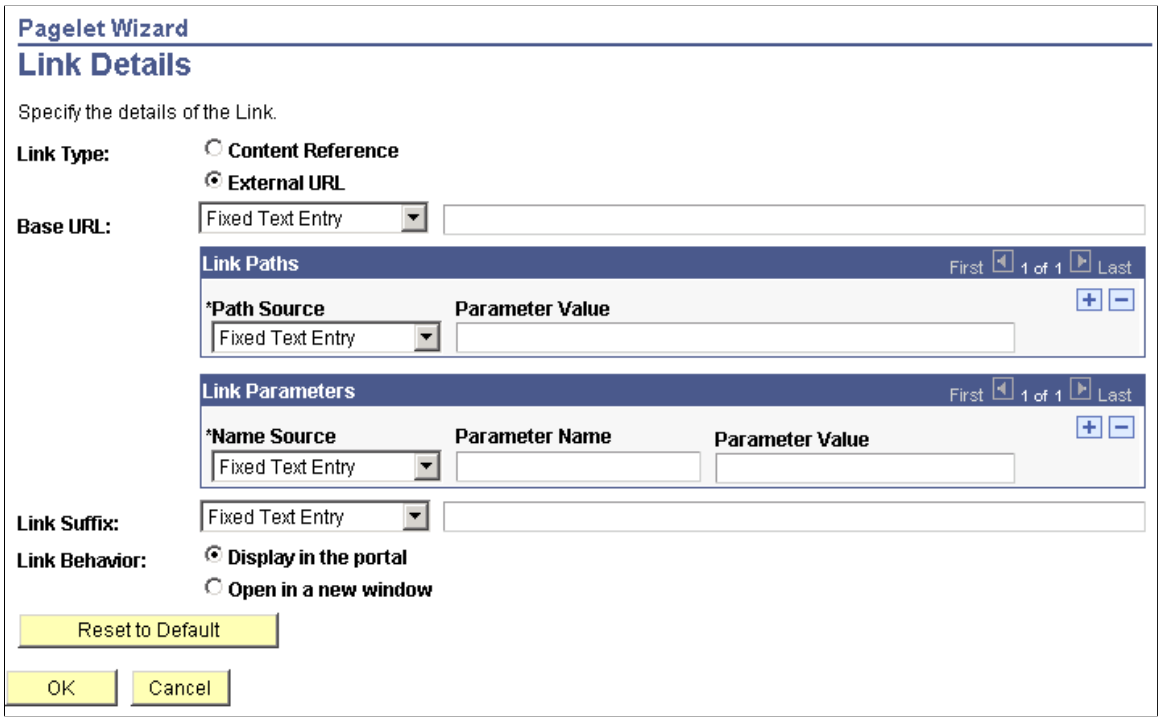
Displays a preview of the pagelet based on most display option settings on this page.

Specifying Data Link Details

Access the Link Details page. (Click the Link Options icon on the Specify Display Options page.)

Image: Link Details page

The following screenshot illustrates data link details on the Link Details page.



You can access this page for field or column names that appear on the Specify Display Options page if you select the table, rowset, list, and chart formats. Entering information on the Link Details page enables you to associate URLs with field values, which at runtime appear on the pagelet as active links that you click to access additional relevant information.

Pagelet Wizard builds the column data link in the follow way 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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 hard-code the link information by entering a value.

Link Paths

Path Source

Select the method to use 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 hard-code 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 to use 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 to use 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 hard-code 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 selected *Fixed Text Entry* as the name source, enter a parameter value.

Link Suffix

Select the method to use to determine the link destination. Options are:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	Specified by Data: Select to have the system dynamically generate the link information based on the selected data field. *Fixed Text Entry:* Select to hard-code the link information by entering a value. If you selected *Specified by Data* as the link suffix, select a data field. If you 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.

PeopleTools Portal Technologies XSL Templates

PeopleTools Portal Technologies delivers seven XSL templates that enable you to transform content from internet news sources that use industry-standard formats such as RSS and ATOM. When you select the URL data source and the custom display format, the following additional templates appear in the drop-down list box in Step 5 of Pagelet Wizard.

These include:

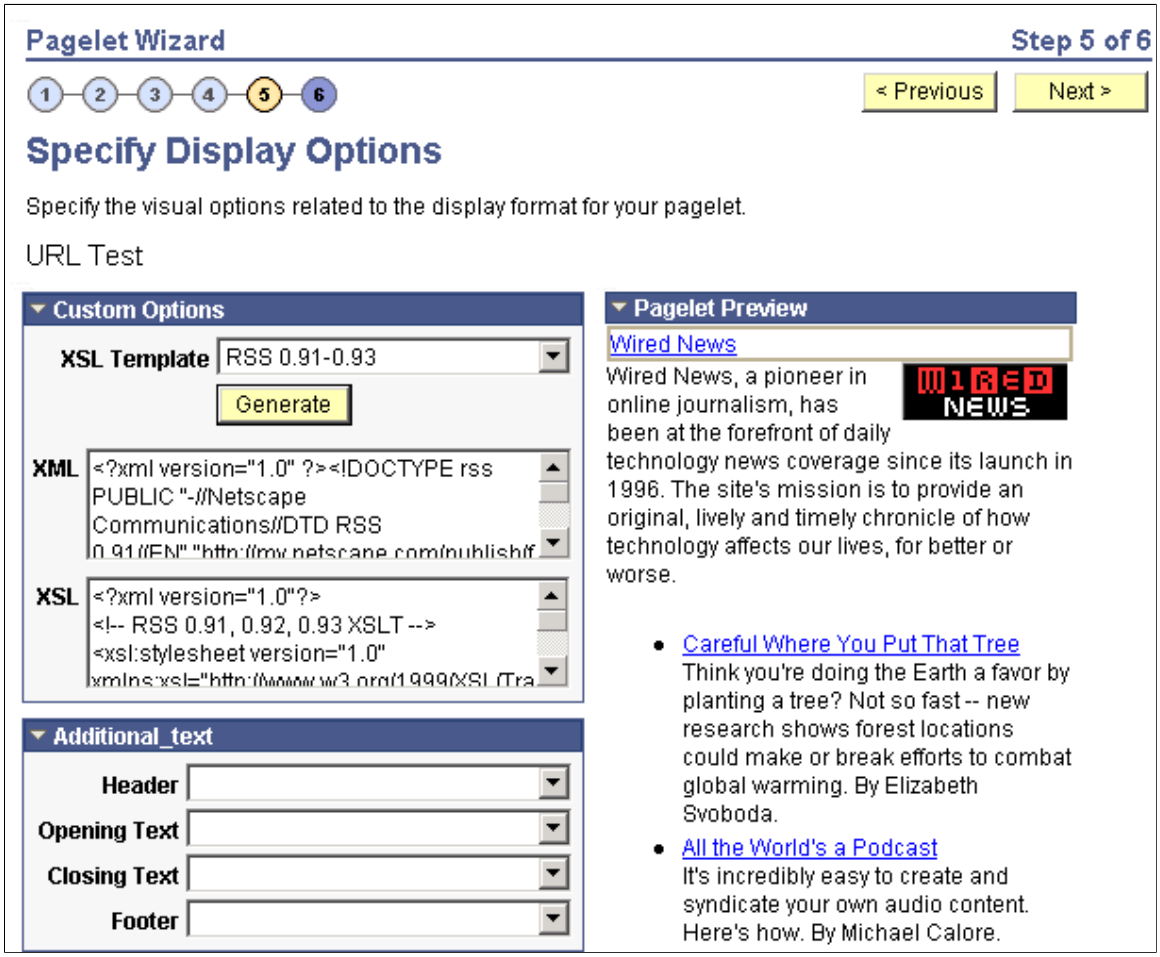
- ATOM 0.3
- ATOM 1.0
- RSS
- RSS 0.91 - 0.93
- RSS 1.0
- RSS 2.0
- Skeletal XSL

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select the appropriate XSL template from the list and click the Generate button. The formatted news articles appear in the pagelet preview on the right side.

Image: Step 5 of Pagelet Wizard with formatted newsfeed

The following screenshot illustrates XSL templates on the Specify Display Options page.



Step 6: Specifying Pagelet Publishing Options

This section provides an overview of caching options and discusses how to specify pagelet publication options. This is step 6 in the Pagelet Wizard pagelet creation process.

Understanding Caching Options

Enabling caching improves performance because after the pagelet is stored in the cache memory, the next time the pagelet is accessed, the information is retrieved and displayed from cache memory instead of requiring a trip to the server. Three caching options exist:

- Private.

  Private caching is user-based; pagelet caching is stored per user. For example, if user A accesses the pagelet, only user A will access the pagelet from the cache.

- Public.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Public caching is application-based; pagelet caching is stored per web server instance. For example, 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.

  Role caching is role-based; pagelet caching is stored per *user role combination*. For example, if user A has membership in roles 1, 2, and 3, only another user with all roles 1, 2, and 3 will access the pagelet from the cache.

Content Reference Attributes

If you enable caching, Pagelet Wizard adds the following attribute information to the pagelet content reference:

- Name: *PSCACHECONTROL*
- Label: *PSCache-Control*
- Attribute value: *Caching Type field value;max-age=Minutes to Cache field value*

  For example, *Public;max-age=30*.

After the pagelet is saved and registered, you can use Pagelet Wizard to alter caching options in the same way that you can alter other registration information. However, altering caching option values requires that you reregister the pagelet.

See Administering Content References.

Common Elements Used in This Section

Enable Caching	Select to enable caching for the homepage or template pagelet. Selecting this option displays the Caching Options group box.
Caching Type	*Private:* Select to enable user-based caching. *Public:* Select to enable application-based caching. *Role:* Select to enable role-based caching.
Minutes to Cache	Enter the number of minutes that the pagelet remains available in the cache. If you do not enter a value, a default value of 30 minutes is used.
WSRP Portlet	Select this option if you intend this pagelet to be produced by PeopleTools as a WSRP portlet. Web Services for Remote Portlets (WSRP) is a standard that defines the way in which remote web services should plug into the pages of a portal. WSRP standards enable you to easily embed a remote web service into a portal. The portal then displays interactive content and services that are dynamically updated from the remote servers. When creating WSRP portlets in Pagelet Wizard, consider the following aspects of WSRP portlet publishing:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Pagelet rendering.

  If the pagelet contains links, the links may transfer to the PeopleSoft portal or stay in-line within the portlet, or consuming portal. If the links point to the PeopleSoft portal, then access to the PeopleSoft system by the browser is necessary. Therefore, in many cases, remote portals consuming WSRP portlets must employ single sign-in between the consuming portal and the PeopleSoft portal for them to be fully functional.

- Pagelet personalization.

  Edit mode as defined by WSRP standards is equivalent to personalization mode in PeopleSoft portals. If the Pagelet being WSRP produced can be personalized (as in, it has user-specified parameters), then the produced WSRP portlet will support edit mode. When the portlet is in edit mode, invalid entries on the personalization page invoke error messages by way of WSRP.

- WSRP production.

  Selecting the WSRP Portlet option on the Specify Publishing Options page causes the pagelet to be registered for WSRP production.

  See Understanding the PeopleSoft WS-Security for WSRPUsing the WSRP Consumption Wizard (Consumer).

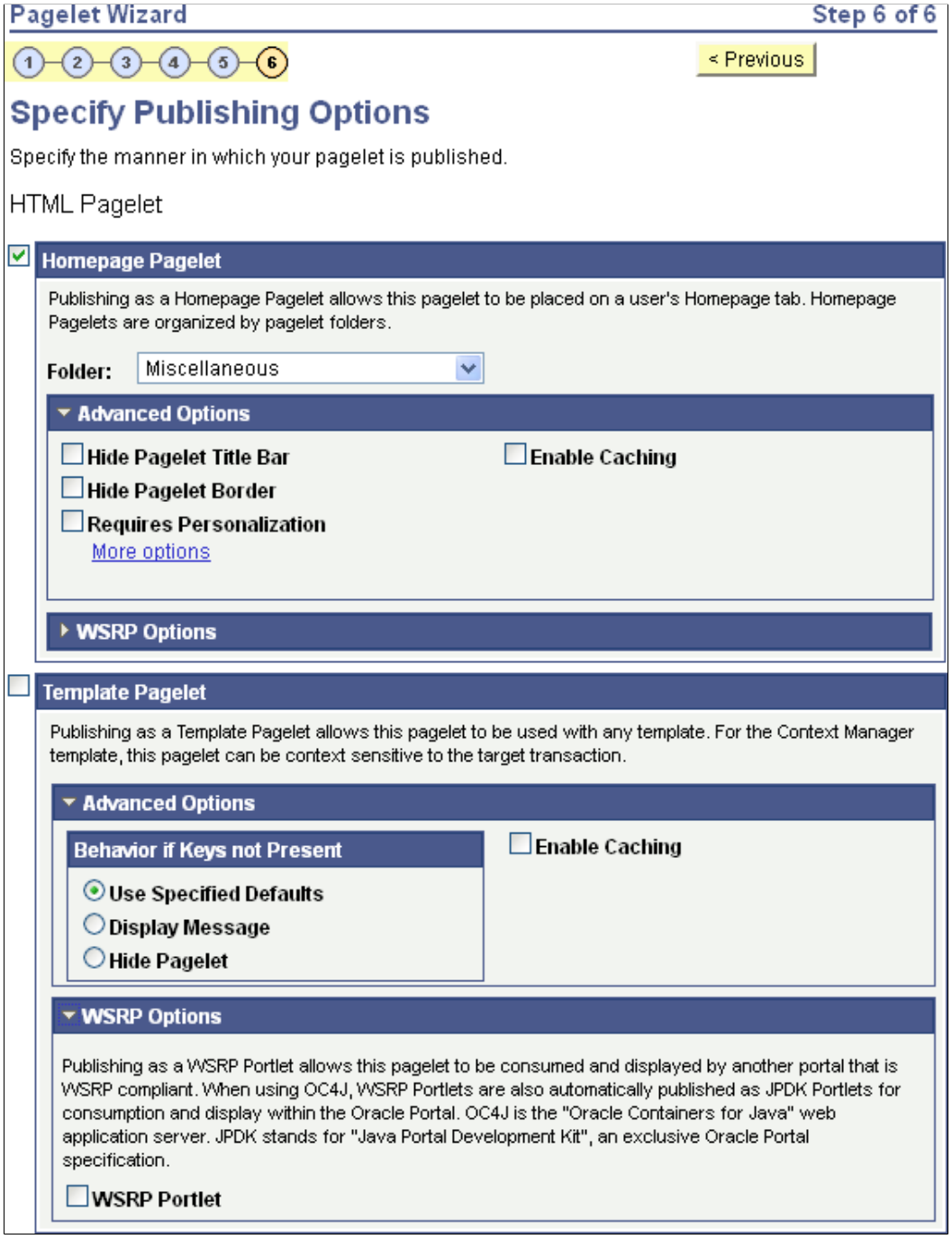
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Specifying Pagelet Publication Options

Access the Specify Publishing Options page. (Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard. On the Select Display Options page, click the Next button.)

Image: Specify Publishing Options page (1 of 2)

This example illustrates the fields and controls on the Specify Publishing Options page. You can find definitions for the fields and controls later on this page.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Image: Specify Publishing Options page (2 of 2)

This example illustrates additional fields and controls on the Specify Publishing Options page. You can find definitions for the fields and controls later on this page.

When you access a pagelet definition on this page, the pagelet registration options (Homepage Pagelet, Template Pagelet, Embeddable Pagelet, or WSRP Portlet) are reflected only as they exist in the portal that you are currently signed in to. Select the registration option to publish the pagelet to the portal where you are currently signed in.

For example, if you access ABC123 pagelet, the Homepage Pagelet option is selected only if the portal where you are currently signed in contains a content reference for the ABC123 pagelet. If no content reference for this pagelet is in the current portal, even if a content reference for this pagelet is in another portal on the same database, the Homepage Pagelet option will be deselected. Select the Homepage Pagelet option to publish the pagelet to the current portal.

Homepage Pagelet

Select to publish the pagelet as a homepage pagelet that can be added to a WorkCenter page, homepage or dashboard page.

Folder Select the portal registry folder in which you want to create the pagelet content reference. The pagelet also appears in this pagelet category on the Personalize Content page, which administrators and users access to manage pagelets on homepages and dashboards.

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. With this option selected, you are requiring 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 deselected, the pagelet displays data using the default data source parameter values provided.

More options After a pagelet has been published as a homepage pagelet, the More options link is displayed. Click the link to access the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pagelet Branding page, on which you can set additional pagelet branding attributes.

See Maintaining Pagelet Branding Attributes.

Template Pagelet

Select to publish the pagelet as a pagelet that can be added to any template. Template pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful when you use the PeopleTools Context Manager or WorkCenter pages to display the template pagelet.

When a pagelet is published as a template pagelet, any user-specified parameters are automatically available as parameters that can be passed in Context Manager. For the keys to pass correctly, the key name in Context Manager (the key name on the page) must be identical to the name of the parameter in the pagelet. If the names match, then any key values passed are used by the template pagelet in place of the default values. If the names are not an exact match, the default values appear. Because users cannot customize template pagelets, user-specified parameters can show only the default values or the values that come through Context Manager key passing.

See Configuring WorkCenter Pages.

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.

WSRP Portlet

Select to make your pagelet available for use with external portals that support the WSRP standard.

See Specifying Pagelet Publication Options.

Embeddable Pagelet

Select this option to make your pagelet available as an embeddable pagelet that can be generated by the Pagelet Wizard API for rendering within an HTML area of a PeopleSoft Pure Internet Architecture target page or using an iScript. You can also publish an embeddable pagelet as a grouplet, by specifying additional parameters.

Publish as Grouplet

This link appears only when the Embeddable Pagelet option is selected, and is available only for the following pagelet data

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

types: HTML, Free Text. This option enables you to make your pagelet available as a grouplet tile on a fluid homepage. Click the link to access the Publish Grouplet Definition page, where you can specify parameters for publishing the pagelet as a grouplet.

See Publishing a Pagelet as a Grouplet for a Fluid Homepage.

Pagelet Security

If you change any pagelet security options for a registered pagelet, you must reregister 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-based 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.

Update Security

This button appears when Pagelet Wizard determines that an inconsistency exists between the security defined for the pagelet on this page and the security on one or more of the content references defined for the pagelet. When you click this button, the content reference security is updated for all of the content references for Homepage Pagelet, Template Pagelet, and WSRP Portlet options.

Selected Security

Type

Select the type of security to assign 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.

Name

Enter the name of the permission list or role to which to authorize access to the pagelet.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Buttons

Previous	Click to move backwards one step in the pagelet creation process in Pagelet Wizard.
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.

Publishing a Pagelet as a Grouplet for a Fluid Homepage

When publishing an HTML or free text embeddable pagelet, you have the option to create a grouplet definition for a fluid homepage. During publishing, the system updates the content reference of the target page, adding the tile content as its grouplet attribute. A user can add the grouplet tile to the fluid home page; when the user presses or clicks the tile, the target content appears.

To publish a pagelet as a grouplet:

1. Create a new pagelet definition or open an existing one using Pagelet Wizard (select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard.)

   Note: The pagelet data type must be free text or HTML.

2. Access the Specify Publishing Options page, step 6 of the Pagelet Wizard.
3. Select the Embeddable Pagelet check box and save.

   The Publish as Grouplet link appears.

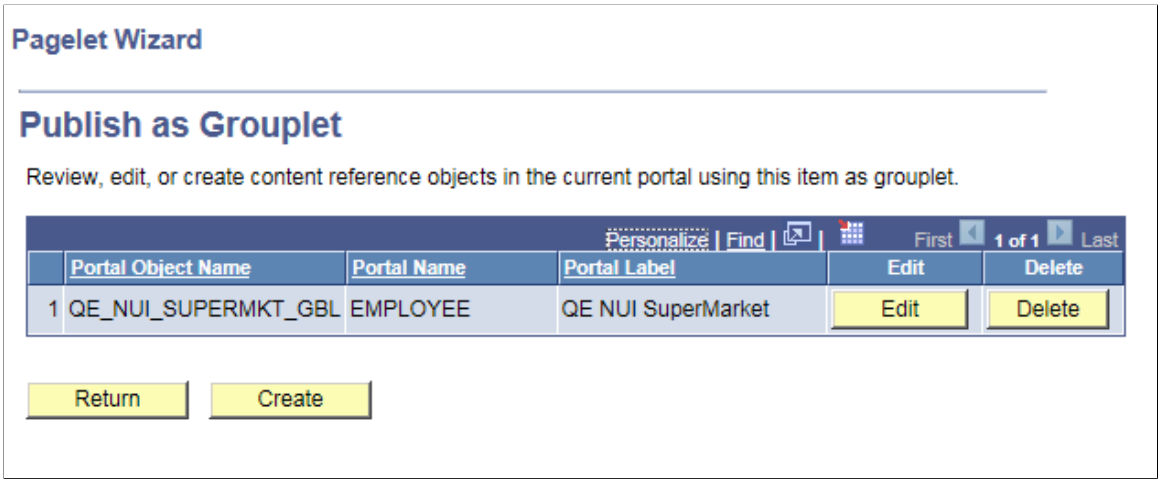
4. Click the Publish as Grouplet link.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

If this is the first time you are publishing the current pagelet as a grouplet, the Publish Grouplet Definition page appears. Otherwise, the Publish as Grouplet page appears, listing the content reference objects that use this pagelet as a grouplet.

Image: Publish as Grouplet Page

This example illustrates the fields and controls on the Publish as Grouplet page. You can use this page to create new grouplet definitions, or edit and delete the existing ones.



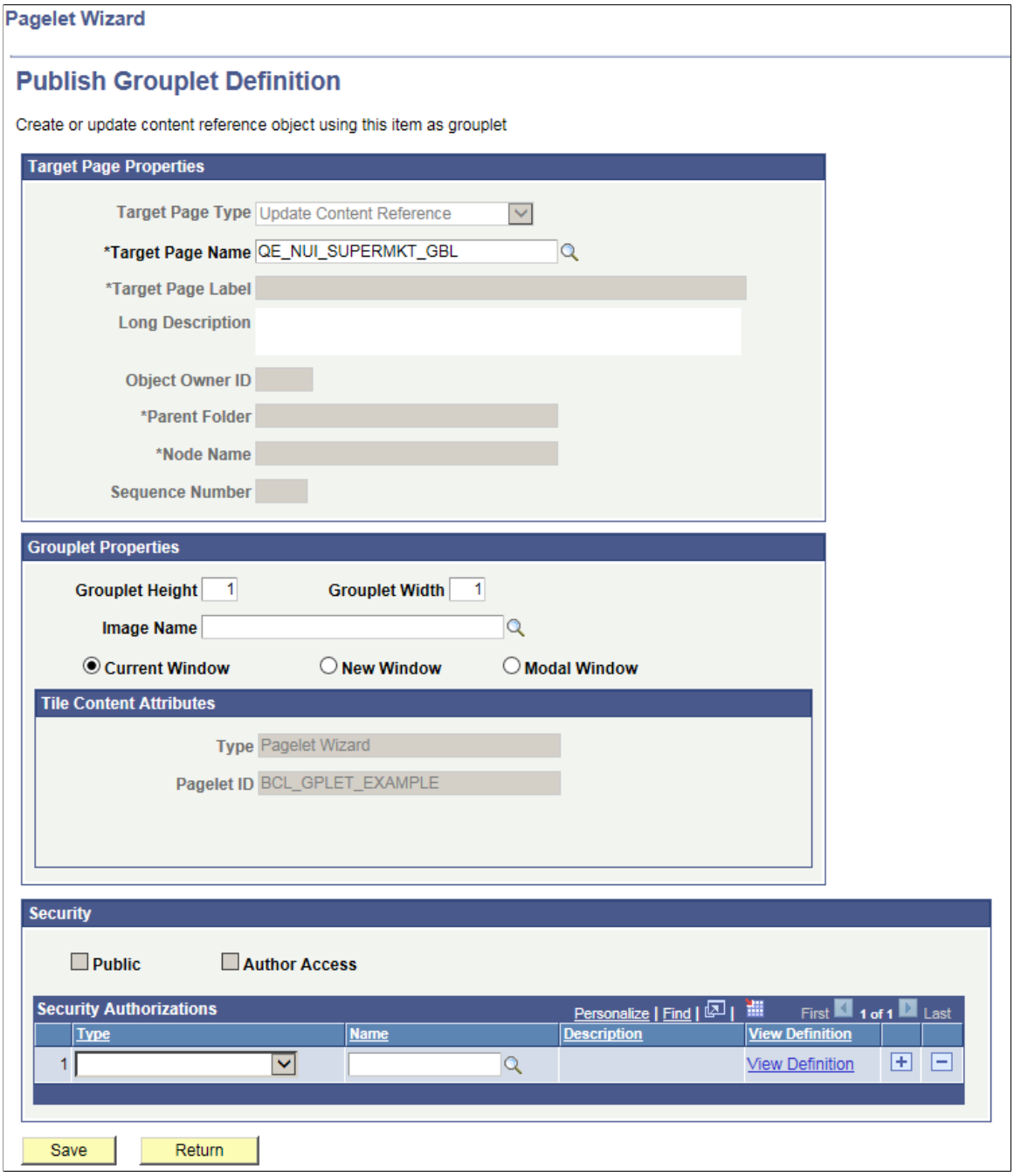
If applicable, click the Create button on the Publish as Grouplet page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

5. Complete the fields on the Publish Grouplet Definition page.

Image: Publish Grouplet Definition Page

This example illustrates the fields and controls on the Publish Grouplet Definition page.



Target Page Properties

In the Target Page Name field, enter the existing content reference that the user will access when they press the grouplet tile.

The remaining target page properties are unavailable for entry; they are automatically populated when you specify the target page name.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Grouplet Properties	You can modify the grouplet properties to change the size and behavior of the grouplet. The tile content attributes are unavailable for entry, as they are derived from the pagelet definition. For more information about grouplet properties, see "Configuring Tile Options" (PeopleTools 8.54: Fluid User Interface Developer's Guide).
Security	These fields are unavailable for entry, they are derived from the security defined for the content reference of the target page.

6. Save the page.
7. When you access the fluid homepage, the grouplet tile is available to add to the fluid homepage.

Using Pagelet Wizard Post-Transformation Processing

This section 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.
- Disable WSRP style substitution.
- Insert a content-reference-based link.
- Insert a content reference URL 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 run in post-transformation processing, the following transformation tasks are not automatically generated for custom and passthru transformations, but they can be added by means of 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.

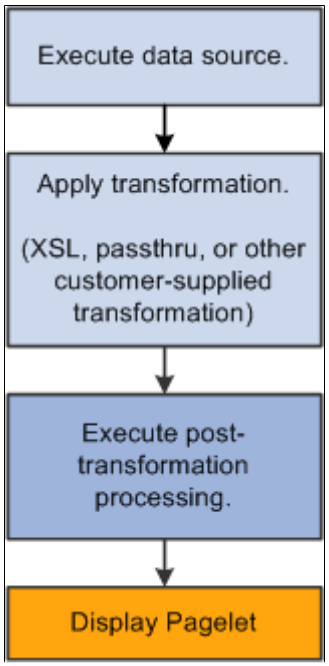
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

6. Insert pagelet headers and footers in specific locations.

These tasks are handled by the Post Transformation step, which is built into Pagelet Wizard.

Image: Pagelet Wizard pagelet-display processing flow

The following diagram illustrates the point at which the Post Transformation step runs in the process flow for displaying a Pagelet Wizard pagelet.



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 XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	*XHTML Tag*	*HTML After Processing*
Message Catalog Text	<PSMSGTEXT MSGSET="*message set*" MSGNBR="*message number*" DEFAULT="*defaulttext*"/>	Message text.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Item to Insert	*XHTML Tag*	*HTML After Processing*
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 Date, Time, DateTime, Currency, or Number elements. Variables that are user-supplied are set in italics.

Note: This processing step occurs only if the output of the Transformation step is XHTML-compliant.

See XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	*XHTML Tag*	*HTML After Processing*
Date	<PSFORMAT TYPE="DATE" [FORMAT="*format string*"][FORMAT_*3 character language code*="*format string*"]>DateToFormat</PSFORMAT> **Note:** FORMAT has the same syntax as the DateTimeToLocalizedString() PeopleCode Function.	Formatted Date. Language-specific 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.
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.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Item to Insert	*XHTML Tag*	*HTML After Processing*
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, the default is 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.

Disabling WSRP Style Substitution

Use the following post-processing tag to disable WSRP style substitution. <PSNOWSRPSTYLESSUBST/> takes no parameters and signals that the style classes referenced in the pagelet are already WSRP compliant. Including this tag can have a slight positive improvement in performance.

Inserting Content Reference-Based Links

Use the following syntax to insert content-reference-based links into a Pagelet Wizard pagelet. Variables that are user-supplied are set in italics.

Note: This processing step occurs only if the output of the Transformation step is XHTML compliant.

See XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	*XHTML Tag*	*HTML After Processing*
Content Reference Link	<PSREGISTEREDLINK 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 occurs only if the output of the Transformation step is XHTML-compliant.

See XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	*XHTML Tag*	*HTML After Processing*
Content Reference URL	<PSREGISTEREDURL PORTAL="*portal name*" CREF="*content reference name*" APPEND="*append*"/>	http…URL of content reference…[?/&]append

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 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>

Administering Pagelet Wizard Pagelets

This section lists common elements used in the section and discusses how to:

- Delete and copy Pagelet Wizard pagelets.
- Generate export and import scripts for Pagelet Wizard pagelets.
- Review Pagelet Wizard pagelet details.

Common Elements Used in This Section

This section discusses common elements used in this section.

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:

- *Activity Guide.*
- *All Data Types.*
- *Free Text.*
- *HTML.*
- *IB Connector.*
- *Integration Broker.*
- *Navigation Collection.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *OBIEE Report.*
- *PS Query.*
- *Pivot Grid.*
- *Rowset.*
- *Search Record.*
- *URL.*

Published Type

Select the pagelet type assigned to the pagelet that you want to appear in the Pagelets scroll area. Options are:

- *All Pagelets.*
- *Embeddable Pagelet.*
- *Homepage Pagelet.*
- *Non-published Pagelet.*
- *Template Pagelet.*
- *WSRP.*

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 signed 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 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 Pagelet Wizard.

Pagelet Title

Displays the pagelet title.

Homepage

If selected, indicates that the pagelet has been published as a homepage page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Template	If selected, indicates that the pagelet has been published as a template pagelet.
WSRP	If selected, indicates that the pagelet has been published as a WSRP portlet.
Embeddable	If selected, indicates that the pagelet has been published as an embeddable pagelet.

Deleting and Copying Pagelet Wizard Pagelets

This section discusses how to:

- Delete pagelets.
- Copy pagelets.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Deleting Pagelets

Access the Delete Pagelets page. (Select PeopleTools, Portal, Pagelet Wizard, Delete Pagelets.)

Image: Delete Pagelets page

The following screenshot illustrates about deleting pagelets on the 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.

The Published tab shows information about how the pagelet has been published.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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.
WSRP	If selected, indicates that the pagelet has been published as a WSRP portlet.
Embeddable	If selected, indicates that the pagelet has been published as an embeddable pagelet.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Copying Pagelets

Access the Copy Pagelets page. (Select PeopleTools, Portal, Pagelet Wizard, Copy Pagelets.)

Image: Copy Pagelets page

The following screenshot illustrates about copying Pagelets on the Copy Pagelets 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.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: Pagelet IDs can contain only the characters A–Z, a–z, and _. All other characters put into a pagelet ID will be removed by the system. If all characters are removed from the user-entered pagelet ID, then the Registry Object Prefix will be used instead. If, after removing illegal characters from the pagelet ID, the name is found to match an existing pagelet ID, then Pagelet Wizard will append numbers to the end of the pagelet ID to make the new pagelet ID unique.

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 Pagelet Wizard and publish the pagelet during Step 6 on the Specify Publishing Options page.

The Published tab shows information about how the pagelet has been published.

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.

WSRP If selected, indicates that the pagelet has been published as a WSRP portlet.

Embeddable If selected, indicates that the pagelet has been published as an embeddable pagelet.

Generating Export and Import Scripts for Pagelet Wizard Pagelets

This section provides an overview of export and import scripts for Pagelet Wizard pagelets and discusses how to:

- Generate export and import pagelet scripts.
- Generate export and import scripts for Pagelet Wizard setup data.

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 content reference by placing it into a PeopleSoft Application Designer project.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 or imported.

Generating Export and Import Pagelet Scripts

Access the Export/Import Pagelets page. (Select PeopleTools, Portal, Pagelet Wizard, Export/Import Pagelets.)

Image: Export/Import Pagelets page

The following screenshot illustrates about how to Export/ Import pagelets on the Export/Import Pagelets page.

Select Select the pagelets for which you want to generate export and import scripts.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

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 "Understanding PeopleSoft Data Mover" (PeopleTools 8.54: Data Management).

Note: Other Pagelet Wizard-related Data Mover scripts are 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 and Import Scripts for Pagelet Wizard Setup Data

PeopleTools Portal Technologies delivers the following Data Mover scripts to export and 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, for example, from a development database to a production database:

- Footers.
- Headers.
- Categories.
- XSL prototype templates.

You can find these scripts delivered in the PS_HOME scripts directory.

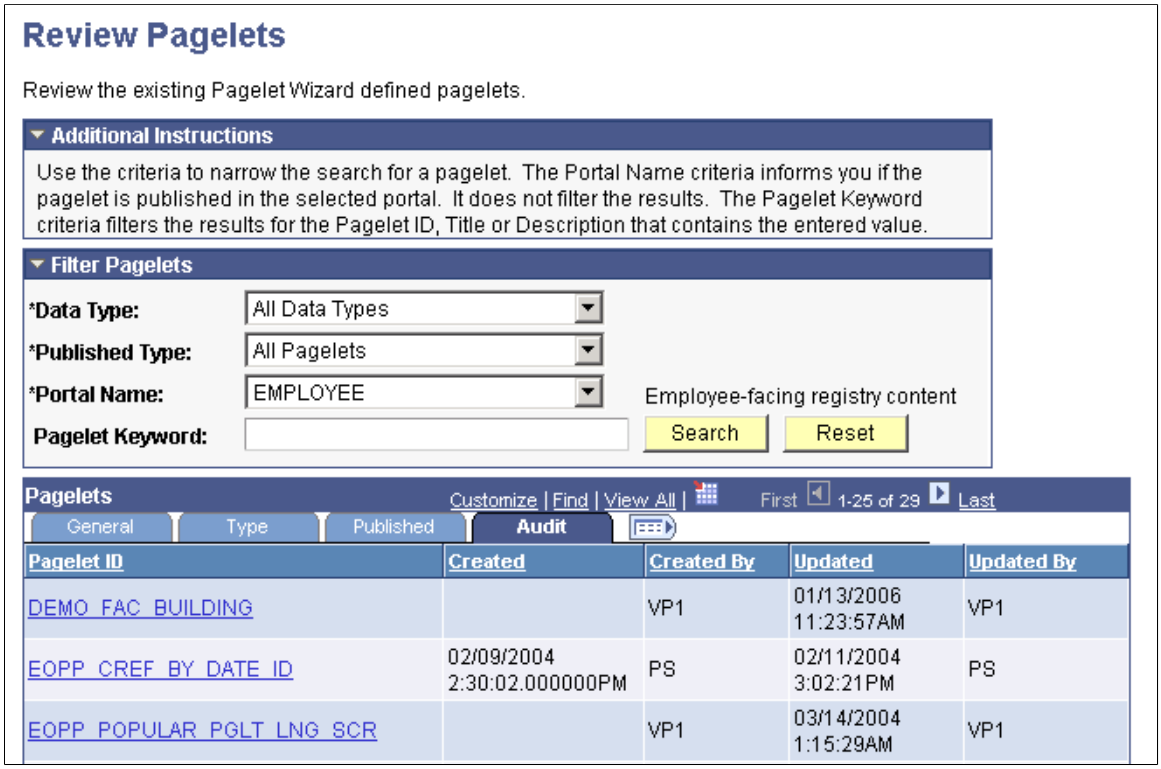
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Reviewing Pagelet Wizard Pagelet Details

Access the Review Pagelets page. (Select PeopleTools, Portal, Pagelet Wizard, Review Pagelets.)

Image: Review Pagelets page

The following screenshot illustrates how to review pagelets on the Review Pagelets page.



On any of the tabs for this component, you can click the Pagelet ID link to open Pagelet Wizard and the definition of the pagelet.

General Pagelet Information

Click the Pagelet ID link to open Pagelet Wizard and the definition of the pagelet.

Pagelet Type Information

Category Displays the category ID from step 1 of Pagelet Wizard.

Data Type Displays the pagelet data type.

Display Format ID Displays the format ID of the pagelet.

Pagelet Publishing Information

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.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

WSRP	If selected, indicates that the pagelet has been published as a WSRP portlet.
Embeddable	If selected, indicates that the pagelet has been published as an embeddable pagelet.
Cache	Displays a check box if caching is enabled for this pagelet.
Time	Displays the time the pagelet is held in the cache.

Pagelet Audit Information

This tab includes the timestamp and user ID of the person who created the pagelet as well as information about any updates to the pagelet.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 14

Developing and Registering Pagelets

Understanding Pagelets

This section discusses:

- Pagelets.
- Pagelet development.
- Pagelet extensions.
- Pagelet personalizations.

Prerequisites

This document provides a practical guide for technical users and programmers who develop pagelets (or portlets) for operation within a PeopleSoft portal. To take full advantage of the information covered in this document, Oracle recommends that you have a basic understanding of the PeopleSoft Pure Internet Architecture and portal technology. Extensive information can be found in the PeopleBooks documentation. Many of the fundamental concepts related to the portal are also discussed in the following PeopleTools document: *PeopleTools: System and Server Administration*.

Pagelets

Pagelets are small pages that provide display-only snapshots of useful and relevant content from PeopleSoft or non-PeopleSoft applications or websites. In other portal products, pagelets are sometimes referred to as portlets, gadgets, widgets, or modules.

Homepage pagelets are designed to appear on tabbed homepages or dashboard pages. Homepage pagelets typically present general, high-level information; however, many are designed to provide the user with easy access to more detailed information either in another pagelet on the homepage or a target transaction page. Users can select the homepage pagelets that appear on his or her homepages by way of the Personalize Content link that is in the upper left corner of the homepage.

Template pagelets are small, narrow-format pagelets that are designed to appear in a template iframe, much like the Menu pagelet. Usually, you incorporate template pagelets into a portal template when you create and configure the template. You also use template pagelets when you configure the task panel of WorkCenter pages or context-sensitive pagelets that use Context Manager. You should design template pagelets so that they display data that is related in some way to the target transaction.

Pagelets follow a basic set of rules so that they can be displayed properly on PeopleTools portal homepages, dashboard pages, related content frames, or in the pagelet area of WorkCenter pages. Homepage pagelet size should correspond to the homepage layout specified by the user. A user can specify either a two-column layout of one narrow pagelet and one wide pagelet, or a three-column layout of three narrow pagelets.

Template pagelets should use the narrow column format of a pagelet. Banner pagelets span the width of the homepage or dashboard page. Any pagelet that you design must conform to the dimensions of the narrow column and optionally, of the wide column. Column width is the only restriction on pagelet dimensions, although design principles suggest keeping pagelets as compact as possible.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The homepage pagelets that appear on the portal homepage depend on several factors, including the type of portal that you implement, the permissions that you grant to the user, and the manner in which the user personalizes his or her homepage.

PeopleSoft pagelets are URL-accessible HTML documents that are designed to be size-appropriate for display on a homepage. Pagelets can also originate from remote, non-PeopleSoft systems such as static HTML files or HTML that is generated dynamically from a variety of technologies.

Pagelet Development

You can use PeopleTools to develop pagelets as smaller-than-normal pages, just as you would any other PeopleSoft transaction page. PeopleSoft portals are delivered with a suite of pagelets, primarily built using PeopleTools technology. However, you can also base the design of a pagelet on PeopleCode internet scripts (iScripts). Use iScripts only when you can't accomplish the same task with a PeopleSoft transaction page, for example, if you are developing a pagelet that uses external content (content that does not originate in a PeopleSoft application). You can also create pagelets using other leading web-enabling technology such as JavaServer pages (JSP), Java servlets, Microsoft Active Server Pages (ASP), Perl scripts, and common gateway interface (CGI) programs.

A well-designed and well-developed pagelet serves these primary functions:

- Summarizes data that is relevant, personalized, and actionable. Page space is valuable; pagelets must be more than attractive.
- Provides links to more detailed information, such as content in another application.

In addition, pagelets should also:

- Aggregate data from multiple sources.
- Be simple and intuitive.
- Be sized appropriately.

  The size of pagelets corresponds to the homepage layout that a user chooses. Any pagelet that you design must first conform to the dimensions of the narrow column and optionally, of the wide column. Pagelet width is the only dimension to which you must adhere. The height of pagelets can vary; however, good design principles suggest keeping them as short as possible.
- Avoid the use of *scrollable grids.*

  Grids that you insert on pagelets are nonscrollable by default. If the grid appears on a pagelet and on a standard transaction page, Oracle recommends that you use the nonscrollable grid to ensure consistent user experiences.

Pagelet Header Icons

The icons that you see in the pagelet header function similarly across most pagelet types. You do not have to use all header icons on all pagelets. You can vary the icons that appear in the header based on business rules or technical constraints.

This table lists and describes the icons that you can use in pagelet headers.

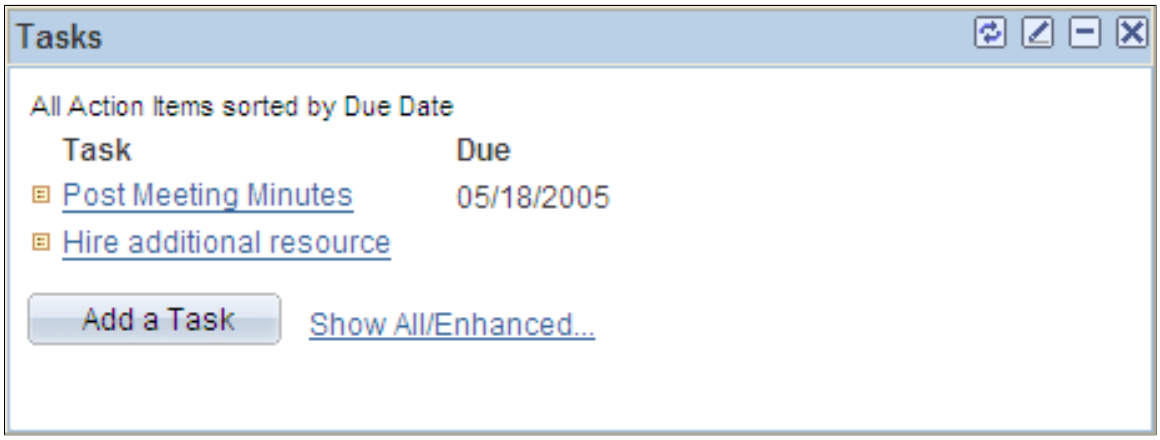
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Expand	Click to show the content of the pagelet so that more than just the pagelet header is visible. When you click this icon, in its place you will see the Hide icon.
Help	Click to view pagelet help. When you click this icon, any documentation that you configure appears. Not all pagelets provide documentation.
Personalize	Click to personalize the pagelet. When you click this icon, a page appears so that you can change any pagelet settings that you are allowed to configure. The options that you can personalize vary among users and pagelets, and not all pagelets have options that you can personalize.
Minimize	Click to hide the body of the pagelet so that only the header is visible. When you click this icon, in its place you will see the Show icon.
Refresh	Click to force an application server trip so that the pagelet data and settings are current.
Remove	Click to remove the pagelet from the page. You will be prompted to confirm the action and must accept or dismiss the confirmation.

Pagelet Example

Image: Tasks pagelet

The following screenshot demonstrates many characteristics of a well-designed pagelet and the pagelet header icons in the Tasks pagelet.



Pagelet Extensions

Some pagelets use pagelet extensions, which are the supporting or supplementary components for a pagelet. They are not displayed on your homepage; you access pagelet extensions by associating them with links and buttons on pagelets. Pagelet extensions are optional. They are a means of providing additional information or editing capability that is too cumbersome to display on the pagelet.

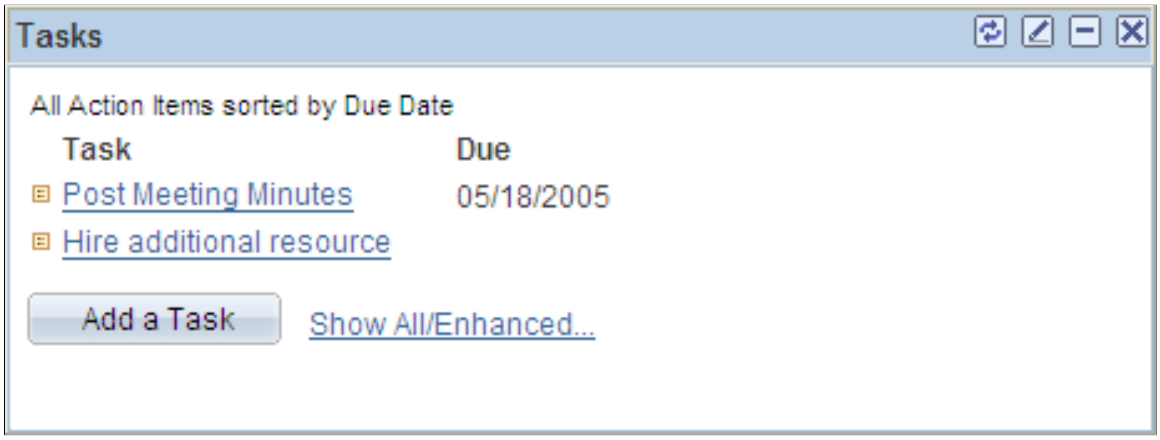
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

You could configure a homepage pagelet so that each item is a link to the worklist item in another database, or to the personal task defined in the portal, or to a variety of resources.

Image: Tasks pagelet

The Tasks pagelet is a homepage pagelet that is designed to display a short list of relevant tasks. If dozens of tasks were open, which is likely over time, the pagelet would grow to be too long and possibly crowd out relevant, actionable data on other pagelets.

The following screenshot illustrates that the tasks pagelet is configured with active links and buttons that access pagelet extensions, which enhance the functionality of the pagelet.



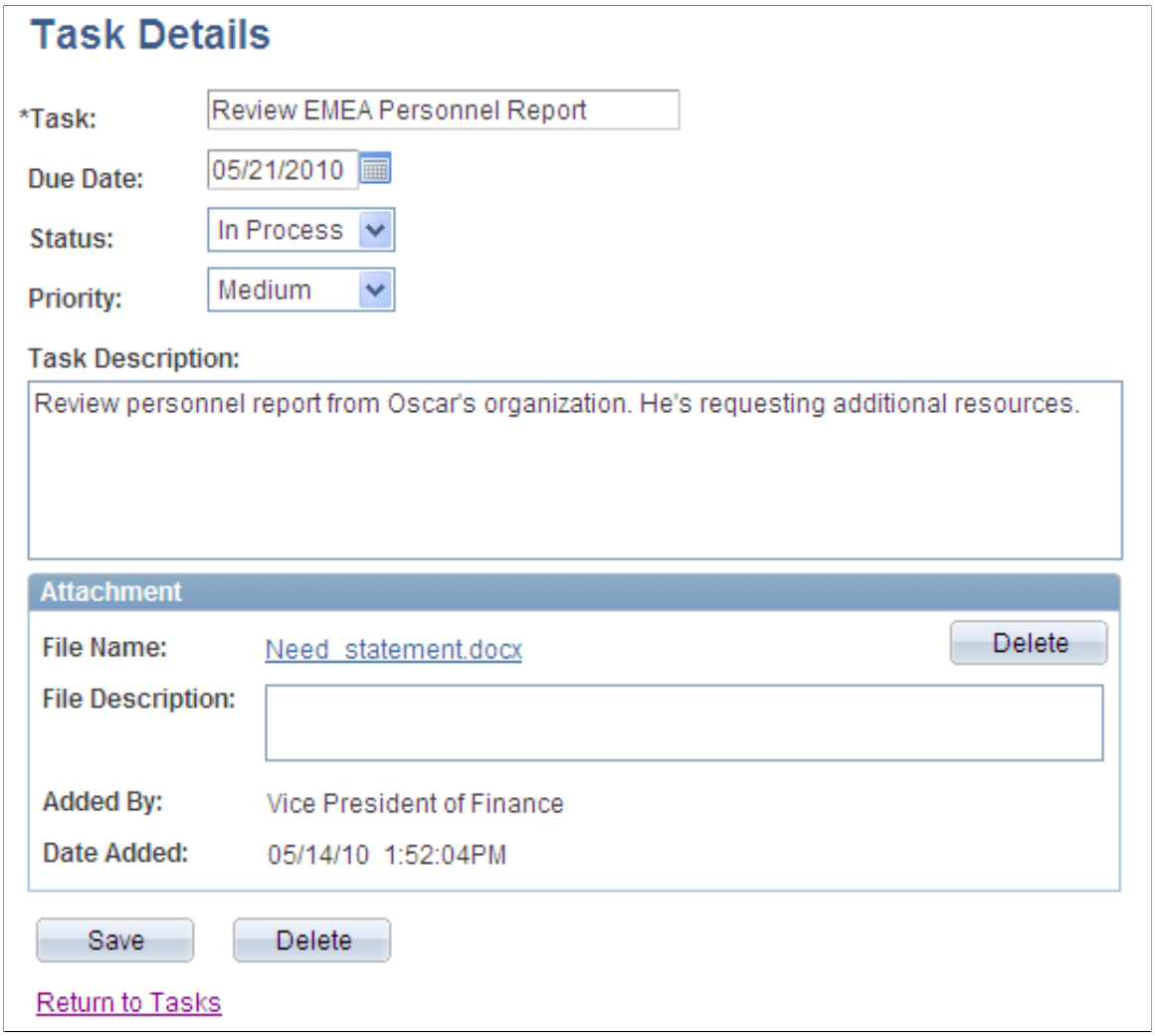
Here you see two extension pages that you access from the Tasks pagelet:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The Task Details page.

  Image: Task Details extension page

  Click a link for a task name to access the Task Details page, which you use to view or modify personal task information. The Task Details page enhances the pagelet by showing more detail and enabling you to perform further actions, such as attaching a file or deleting the task.



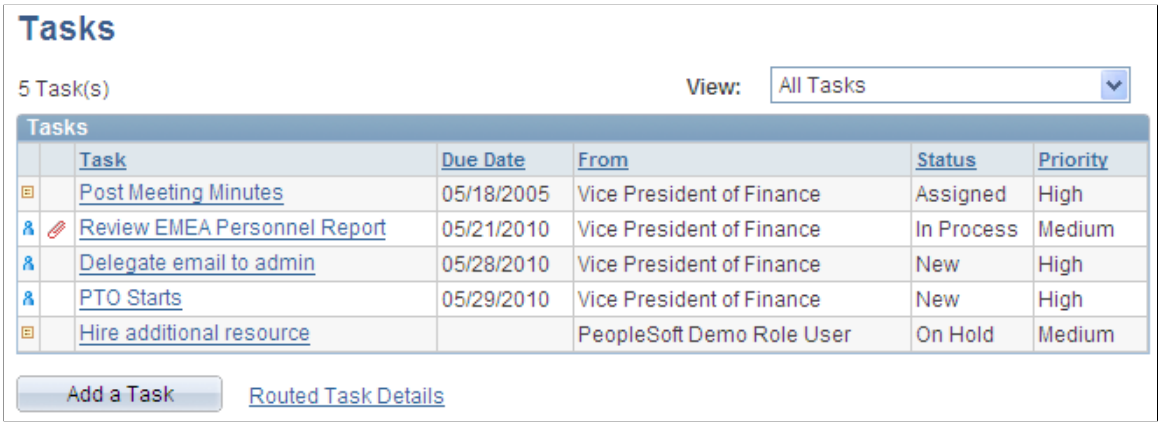
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The Tasks page.

  Image: Tasks page

  Click the Show All/Enhanced link to access the Tasks page, which provides the user with more options such as the ability to display all or certain types of tasks. Because it is a standard page, the extension page can use a grid to retrieve and display a longer list of tasks. In addition, the extension page is also better suited to display more details because it is wider than the pagelet and can show more columns

  The following screenshot displays more details with more columns.



Pagelet Personalizations

Like homepages, pagelets can be personalized in different ways. The data displayed can be filtered automatically to show only data that is relevant to a particular user. The user can also explicitly personalize the data shown by setting options on a personalization page.

Personalization pages are another type of pagelet extension. However, personalization page pagelet extensions are different from previously discussed pagelet extensions in the following ways:

- You do not click a button or link on the pagelet to access personalization pages. Instead, clicking the Personalize icon on the pagelet header accesses a personalization page where you can define user options that are specific to that pagelet.

- Setting values on the personalization page will change the way data appears on the pagelet for the user. Detail pages usually display data in read-only mode or allow you to change application data.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: Any text that the pagelet developer enters during Step 3 in Pagelet Wizard appears as instructions on the Personalize page.

Image: Pagelet personalization example - Personalize Tasks page

The following screenshot illustrates to specify default values to be able to personalize a Pagelet..

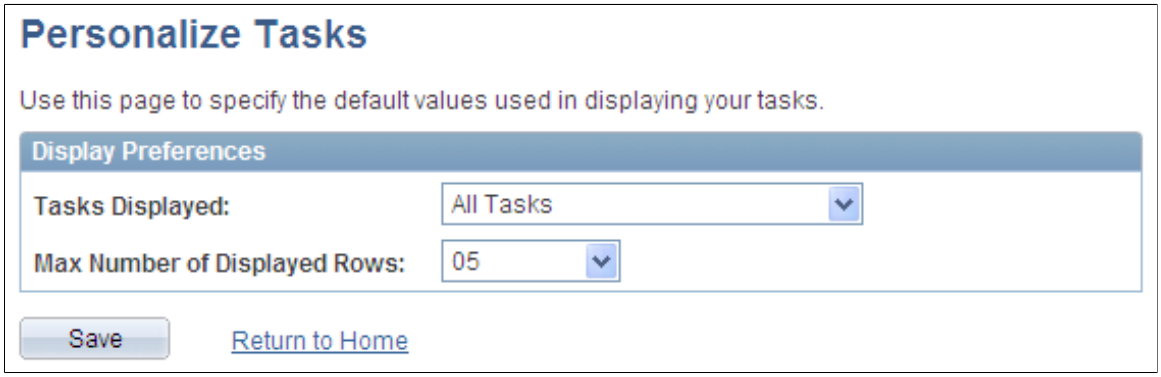
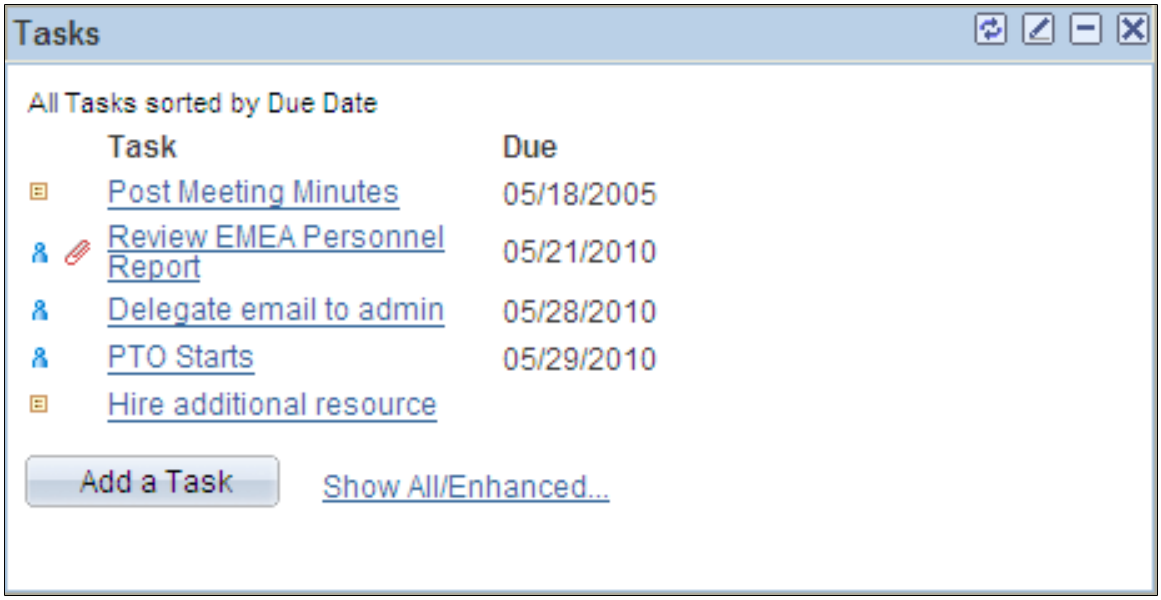


Image: Tasks pagelet after making personalizations

You can use the Personalize Tasks personalization page for the Tasks pagelet to change the type and number of tasks shown on the pagelet. Clicking Save stores these values and returns you to the homepage. The Tasks pagelet now reflects these changes.

The following screenshot illustrates about changing type and number of tasks on the Task Pagelet.



Understanding PeopleSoft Portal Architecture

This section discusses:

- PeopleSoft portal architecture.
- Client access.
- The web server.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The application server.
- The database server.
- The portal registry.
- Portal components.
- Page assembly.

PeopleSoft Portal Architecture

When you develop pagelets, you need to understand the overall architecture of the PeopleSoft portal, as well as the PeopleSoft Pure Internet Architecture. This will enable you to integrate your pagelets in the most efficient manner.

The PeopleSoft Pure Internet Architecture is the server-centric architecture used to deploy PeopleSoft internet applications to end users who access the applications through a web browser. This next generation architecture leverages a number of internet technologies and concepts to deliver simple, ubiquitous access to PeopleSoft applications and to enable the open flow of information between systems.

Using PeopleSoft Pure Internet Architecture as the foundation, you can provide a wide range of end users with access to PeopleSoft applications over the web, as well as more easily integrate your PeopleSoft applications with existing internal systems and external trading partner systems.

The following diagram highlights the systems and the primary portal components of the PeopleSoft Pure Internet Architecture:

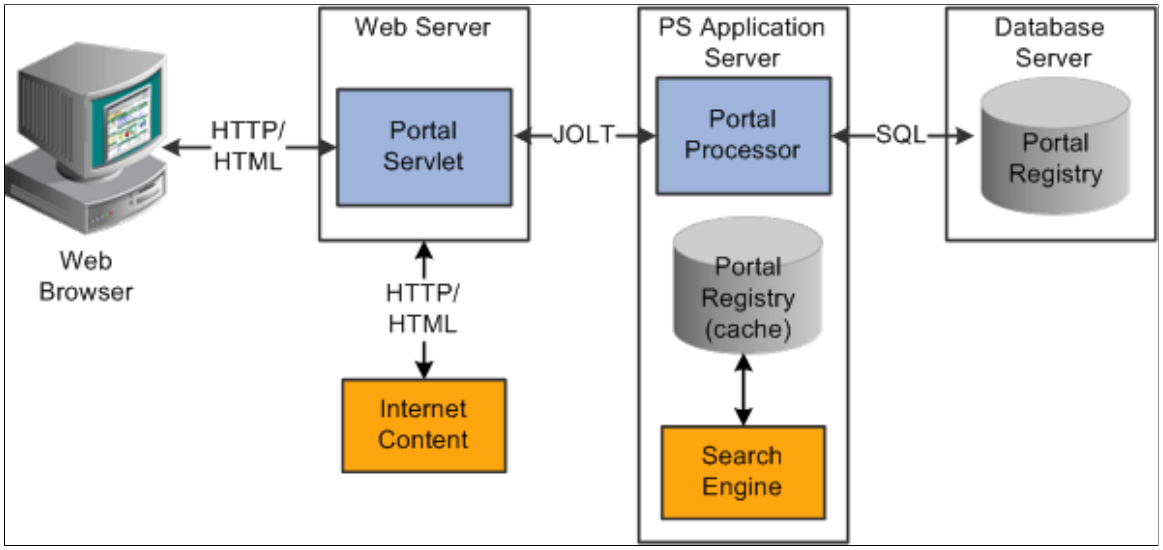
- The web browser communicates with the web server over HTTP/HTTPS.
- The portal servlet runs on the web server. The web server communicates with the application server over Jolt. The web server also communicates with other internet content providers over HTTP/HTTPS.
- The portal processor runs on the application server. The search engine retrieves index entries from the portal registry cache on the application server. The application server communicates with the database server through SQL.
- The portal registry data resides on the database server.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Warning! Always refer to the certifications section of My Oracle Support for supported configurations. Versions and products supported can change frequently.

Image: Primary PeopleSoft Pure Internet Architecture portal components

The following diagram highlights these primary portal components of the PeopleSoft Pure Internet Architecture: the portal servlet (web server); the portal processor, portal registry cache, and search engine (application server); and the portal registry (database server).



Client Access

PeopleSoft Pure Internet Architecture is a completely server-based architecture. Clients to this architecture can be nearly any kind of internet access device, such as:

- A web browser running on supported operating system.
- A wireless device or cell phone.
- An external or third-party system with extensible markup language (XML)/hypertext transfer protocol (HTTP) protocols.

No PeopleSoft executables are on the client; thus the client can be any internet device that uses standard internet technologies such as HTTP, hypertext markup language (HTML), and XML to communicate with the PeopleSoft internet application server.

A web browser running on a PC is the most common internet client. The PeopleSoft internet application server simply serves HTML and JavaScript to the web browser and the end user navigates through the PeopleSoft application as if he or she were navigating any other website.

A key concept of PeopleSoft Pure Internet Architecture is that no complex client software installation is required. The internet client device accessing the internet architecture already has all of the software and configuration that it needs. No additional software must be installed on the client for interaction with PeopleSoft applications. For example, no Java applets, Windows .DLLs, or browser plug-ins are needed.

The Web Server

The web server acts as the front end of PeopleSoft Pure Internet Architecture. When a client connects to the server by way of a URL, the system displays a sign-in screen sent to the browser in HTML. The web server manages communication with the browser.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The following web server products can be configured to deploy your PeopleSoft applications:

- Oracle WebLogic server.
- IBM WebSphere server.

Two key PeopleSoft servlets run on the web server—the presentation relay servlet and portal servlet.

Presentation Relay Servlet

The presentation relay servlet is used to process all inbound and outbound HTTP requests for PeopleSoft transactions and queries. This very thin servlet acts as a relay between the client device and the core back-end services on the application server. It receives and serves HTML, XML, and wireless markup language (WML) requests over HTTP.

Portal Servlet

The portal servlet is a Java servlet that runs on the portal web server. It intercepts user requests for HTML pages, retrieves the requested page, wraps additional content around it, and then sends the resulting page to the user's browser. The servlet acts as an invisible browser that sits between the user's browser and requested content.

The portal servlet:

- Provides a consistent user interface.

  The portal servlet checks properties associated with each content reference, including the name of a portal template. When a user accesses content through the portal, the portal servlet wraps the target page with the portal template specified in the content reference. This template provides a consistent user interface.

- Ensures that PeopleSoft-specific tags are processed correctly.

  Developers create portal pages using a template-based layout system. In addition to traditional HTML tags, templates can contain PeopleSoft-specific tags that a normal browser cannot interpret. At runtime, the portal servlet can interpret these PeopleSoft-specific tags when constructing templates, as well as any other HTML content. The portal servlet then sends the resulting page to a browser as a single HTML document.

One of the most important aspects of portal technology is its role in integrating content from a wide variety of data sources and displaying that content on a single page in a coherent, understandable, and presentable way. We refer to this complex process as “page assembly and proxying.” Portal processing assembles the page to be displayed based on the content retrieved from various data sources. It uses portal templates to wrap the contents of the assembled document into a single page that fits into the context of the site.

- For page-based templates, the portal servlet assembles pages for the browser.

  It ensures that all URL references in the HTML on the assembled page are references back to the portal servlet itself. In some cases, each URL in the HTML document assembled by the portal servlet must be rewritten to reference the portal servlet, not the originally requested URL. This process of redirecting URLs so that they point to the portal servlet is called proxying.

- For frame-based templates, the portal server updates the src tags in the frameset with the target content and sends it to the browser.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

When working with a frame-based template, the portal servlet inserts a URL into each frame in the src tag and sends it to the browser rather than retrieving documents for the browser as it does with page-based templates.

The Application Server

The application server is the core of PeopleSoft Pure Internet Architecture; it carries out business logic and issues SQL to the database server. The application processing logic that ran on the client in previous releases now runs on the application server. The application server consists of numerous PeopleSoft services and server processes that handle transaction requests. These include requests to:

- Authenticate users.
- Build application pages.
- Save application pages.
- Run some PeopleCode.
- Run SQL (prompts, page-specific SQL, validations, and so forth).

The application server is responsible for maintaining the SQL connection to the database for the browser requests and the Windows development environment. PeopleSoft uses Tuxedo to manage database transactions and Jolt, Tuxedo's counterpart, to facilitate transaction requests issued from the internet. Both Tuxedo and Jolt are products of Oracle Systems.

The Portal Processor runs as an application service of the PeopleSoft application server. It runs with the other application services, such as the Component Processor, Security Manager, and SQL Access Manager. Portal Processor tasks include:

- Fetching content templates from the database.
- Fetching content references from the database portal registry and caching them in the application server portal registry.
- Processing personalizations.
- Interacting with other application services (lightweight directory access protocol (LDAP), role-based security, and so forth).

The Database Server

The PeopleSoft database is the repository for all information managed by PeopleSoft applications. Not only is application data stored in the database, but PeopleSoft metadata is also maintained in the database. PeopleSoft Application Designer enables you to define and maintain this metadata that the system uses to drive the runtime architecture. The application server carries out business logic based on the PeopleSoft metadata.

At runtime, the application server fetches the most recent application object definitions from the metadata repository, compiles and caches the application object into memory, and carries out the business rules based on the definition.

Note: In general, PeopleSoft Interaction Hub 9.x can retrieve content from any RDBMS-based application as long as the content that is being retrieved is URL-accessible and is HTML-based. Always refer to the certifications section of My Oracle Support for supported RDBMS products.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The Portal Registry

The portal registry is a key administrative component within the metadata of a PeopleSoft database. A portal registry is a hierarchical structure in which URLs accessed by way of the portal are organized, classified, and registered. Each portal registry consists of the following objects:

- Folders.

  Folders group and organize content references into a hierarchy. Each folder can contain content references as well as other folders.

  Every portal registry contains a root folder and a Portal Objects folder. The Portal Objects folder contains administrative objects specific to the portal and includes the following folders: Homepage, Navigation Collections, Pagelets, Template Pagelets, and Templates.

  In addition to these standard folders, several folders typically are located directly under the root folder: one folder for PeopleTools (administrative references) and other main folders for each PeopleSoft application. These main application folders contain the folders and content references associated with each PeopleSoft application that you've licensed.

- Content references.

  Content references are URLs that have been registered in a portal registry. They can be PeopleSoft application pages or external web pages. Content references fall into four main categories: pagelets, target content, templates, and homepage tabs.

  In addition to specifying a URL, each content reference includes additional information such as its creator, effective dates, associated template, search keywords, and so forth. Registry URLs can point to any website that responds to HTTP requests with an HTML response—that is, a static or dynamic web page.

  For example, a content reference could be a URL pointing to a page in a PeopleSoft application. Or a content reference could point to static HTML pages on an intranet site, such as a procurement or expense policy document, or dynamic pages created by an analytic or reporting system. Access to content references is controlled by security permission lists assigned to each content reference. Any portal content can be limited to a specified group of users, or made public and available to any authorized portal users.

- Nodes.

  Nodes refer to a source of HTML content. They primarily identify the universal resource indicator (URI). A URI is a subset of the URL that points to the location of the resource. It does not include the content information, such as the target file or application and any parameters passed to that resource.

The portal registry's hierarchical structure enables a portal administrator to more easily create a classification and navigation system in which content references can be registered and managed for all portal users. PeopleTools provides portal administration pages for this purpose. Additionally, the portal includes a registry application programming interface (API) for maintaining the portal registry from PeopleCode.

To improve performance, a copy of the registry is also stored on the application server in cache files. This is how the portal servlet accesses the registry at runtime.

Portal Components

The portal technology framework includes the following services:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Personalization.

  Pagelets optionally can be assigned a user personalization settings page. This allows the user of the pagelet to specify selections, settings, or attributes specific to the pagelet's content. These personalizations commonly alter the content of the pagelet. An example of this would be setting the city for which a weather pagelet displays forecast information. A personalization page is accessible to the user by means of clicking the Personalize button on the pagelet's frame header.

- Search.

  The portal uses a search engine to quickly search for registered content in the portal registry. This is a popular means of portal navigation. Verity, the industry leading search engine, is packaged with the portal technology. PeopleTools search capabilities were built to assume multilanguage support, including double-byte languages.

- Navigation.

  PeopleTools portal technology provides a set of navigation components based on the portal registry. These components are the drop-down menu and Favorites. Navigation has been engineered to provide rapid access to complex information based on the role of the user.

  - Menu navigation.

    PeopleTools portal menu navigation provides a consistent method of content access, categorization, and organization. The menu navigation presents a dynamic hierarchy-based view of the folders and references within the portal registry. Menu navigation is available through the Menu pagelet or from the Main Menu drop-down navigation depending on your portal settings.

  - Favorites.

    The PeopleTools portal includes a My Favorites folder that you use to store frequently accessed pages. In the Menu pagelet, this folder is at the root level. In drop-down navigation, My Favorites appears under the Favorites drop-down menu.

  - Recently Used.

    The PeopleTools portal include a Recently Used listthat includes up to five of your most recently accessed components. This list appears under the Favorites drop-down menu.

Related Links

Managing General Settings for Portals

Page Assembly

Previous sections in this PeopleBook have described the individual components of PeopleSoft Pure Internet Architecture. This section provides a high-level overview.

1. An HTTP request is sent from the user's browser to the portal's web server, which invokes the portal servlet. Assume that the browser is requesting a PeopleSoft application page.

2. The portal servlet makes a request to the Portal Processor to fetch the content reference for the link selected by the user. The Portal Processor fetches the content reference from the portal registry and passes a partially completed HTML document back to the portal servlet.

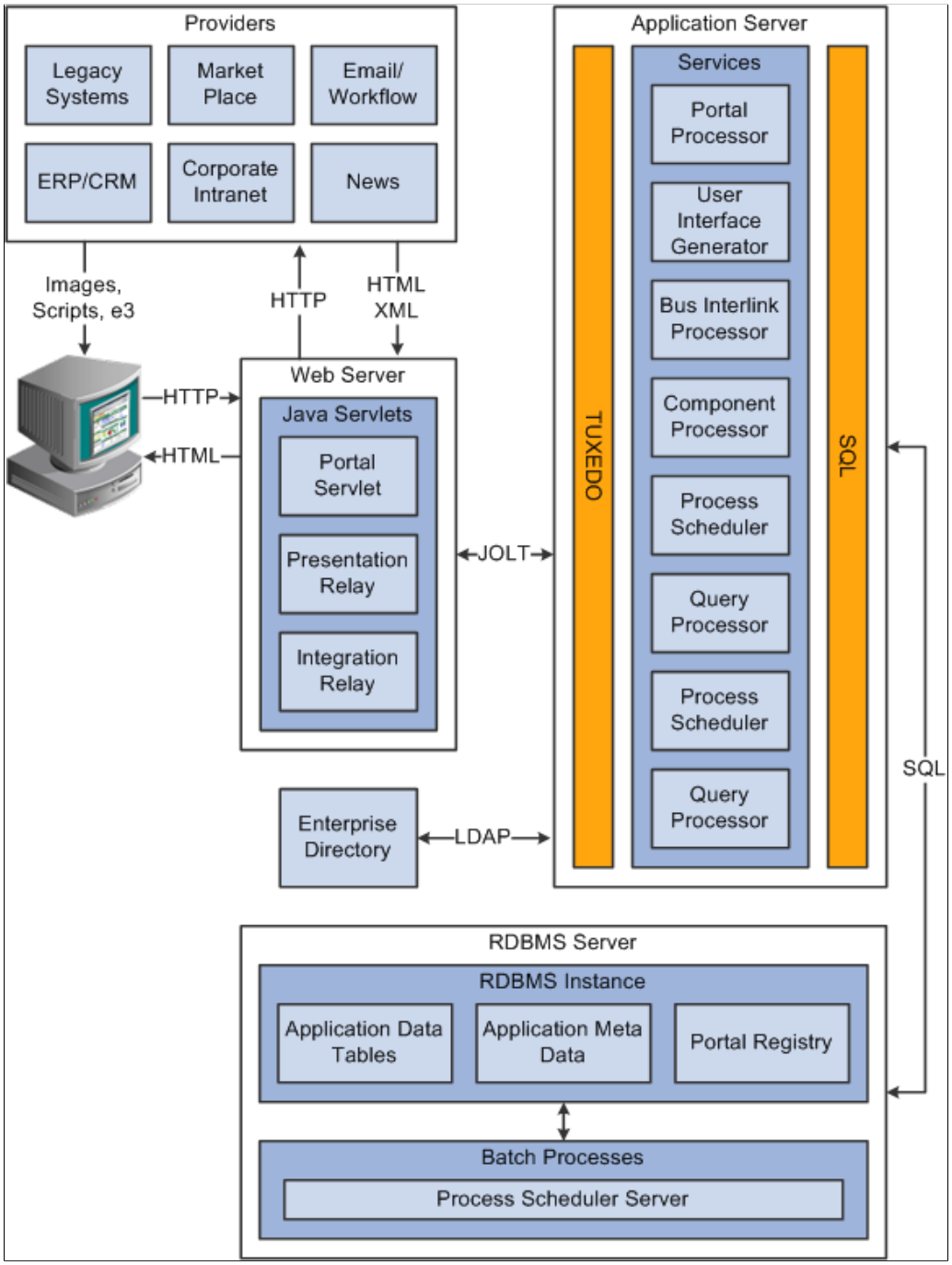
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

3. The content reference could be pointing to any or several of the content providers (specified by a node). Each content reference is referenced in a partially completed HTML document. The portal servlet issues the HTTP request to the appropriate web server and retrieves the HTML document.

4. The portal servlet then completes the page assembly process with the template, merging the content from the HTML document, and then serves the completed page back to the user's web browser.

Image: PeopleSoft Pure Internet Architecture processing flow

The following diagram illustrates the numerous sources of data for page assembly in the PeopleSoft Pure Internet Architecture:



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Understanding Pagelet Development

This section discusses:

- Pagelet development guidelines.
- Pagelet development considerations.
- Pagelet development options.
- Size considerations.
- HTML design considerations.
- Branding considerations.

Pagelet Development Guidelines

Follow these guidelines when developing new pagelets:

- Adhere to object naming standards.
- Do not provide any filter, sort, or refresh buttons because they also require trips to the application server.
- If you enable the cache homepage option in the web profile, the pagelet must be display-only and cannot be interactive.

Because the portal servlet performs page assembly and proxying only on blocks of HTML, a pagelet must:

- Be URL-accessible.

  The portal servlet will issue an HTTP request to the registered URL.

- Be HTML 4.0 compliant.

  The HTML returned will be combined with HTML from other pagelets to form the entire web page.

- Be sized appropriately.

  The pagelet's width should conform to the narrow or wide guidelines discussed in the Sizing section in this document.

  See Understanding Pagelet Wizard Display Sizes.

- Use JavaScript namespacing.

  If you write custom JavaScript code, ensure that JavaScript from multiple portlets can coexist on the same page. Only one namespace is shared by all portlets on a portal page. For example, if portletA and portletB both use the GETURL function, only one definition will exist for GETURL defined by the most recently processed portlet. If you need a different GETURL function for each portlet, then you must uniquely name the GETURL function for each portlet, for example, portletA calls GETURL1 and portletB calls GETURL2. If the same GETURL function is called by both portlets, then you don't need to rename GETURL because the most recent definition of GETURL will be used by both functions.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pages created by way of PeopleSoft Pure Internet Architecture conform to these requirements. However, these requirements are not exclusive to PeopleSoft. As mentioned, pagelets can also be from remote, non-PeopleSoft systems, such as static HTML files or dynamically generated HTML from a variety of technologies such as JavaServer Pages (JSPs), Java Servlets, Microsoft Active Server Pages (ASPs), Perl scripts, and CGI programs.

Pagelet Development Considerations

When developing pagelets, you should also consider the following topics.

Single Signon

Oracle delivers a component interface named PRTL_SS_CI that enables external applications to support a single signon solution with PeopleSoft portal applications. This enables users to sign in once when accessing the portal and not have additional sign-ins for each system that is referenced in your portal.

See Understanding Single Signon and Pagelets.

Layout

Observe the following rules for layouts:

- Avoid horizontal scrolling.
- Do not use page titles.

  The name of the content reference in the portal registry is used as the pagelet title.
- Pagelet instructions should not be necessary.
- Avoid group boxes whenever possible.

  Note: They may be necessary to separate sections.
- Left align text wherever possible.
- Deselect the Show Borders option for group boxes and scroll areas.
- Select View, Internet Options in PeopleSoft Application Designer to ensure that you can access all the styles available.
- Leave more than 20 percent spacing between field labels and field edit boxes because the rendered styles are larger in the browser than they appear in PeopleSoft Application Designer.
- Before bringing your pagelet into the portal, view it in PeopleSoft Application Designer. Select Layout, View in Browser.

Appearance and Character

The appearance of objects on pagelets, such as text, links, tables, and so forth, should be modified by means of the PeopleSoft style classes to retain a consistent appearance and character between your pagelet and the rest of the portal. PeopleSoft uses cascading style sheets (CSS). Style sheets are useful because they enable developers to change page attributes across multiple pages quickly and easily.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Each style sheet is composed of individual classes, which affect the formatting of each page. Classes control a broad range of characteristics, including font size, spacing, alignment, border width, weight, and color.

When creating the page, select the *Use Default Style Sheet* option. For any controls on the page, select the *Use Default Style* option.

All style sheets used in each pagelet on the page are referenced on the final page that is assembled by the portal. Therefore, a pagelet should not redefine any class that might be shared by other pagelets on the assembled page. If a class needs to be changed, define a new one.

The order of precedence for style sheets is:

1. Template.
2. Target component.
3. Pagelets – in the order that they are assigned.

Note: In the same way that PeopleTools supports subrecords to enable shared definitions between records, you can define substyle sheets to share a common set of classes. A substyle sheet has all of the properties of a style sheet.

Style sheet definitions are stored in the database. They are accessed and modified by means of PeopleSoft Application Designer. The default style sheet used by PeopleTools depends on your application release, but is either PSSTYLEDEF or PSSTYLEDEF_SWAN. The PeopleSoft Interaction Hub uses a PSSTYLEDEF or PSSTYLEDEF_SWAN derivative.

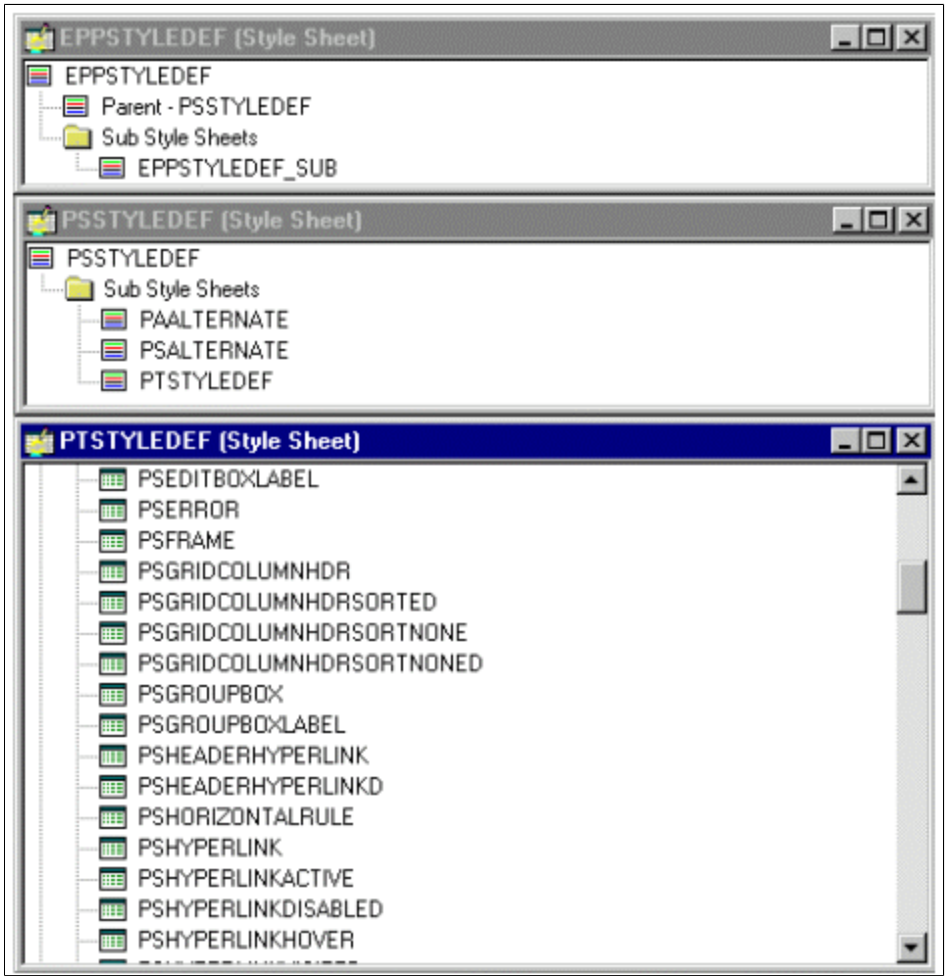
You can determine which style sheet is being used by selecting PeopleTools, Utilities, Administration, PeopleTools Options. The value in the Style Sheet Name field designates the style sheet used by the database.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Two ways are available by which you can see the style classes defined in a style sheet.

Image: Style sheet definitions in PeopleSoft Application Designer

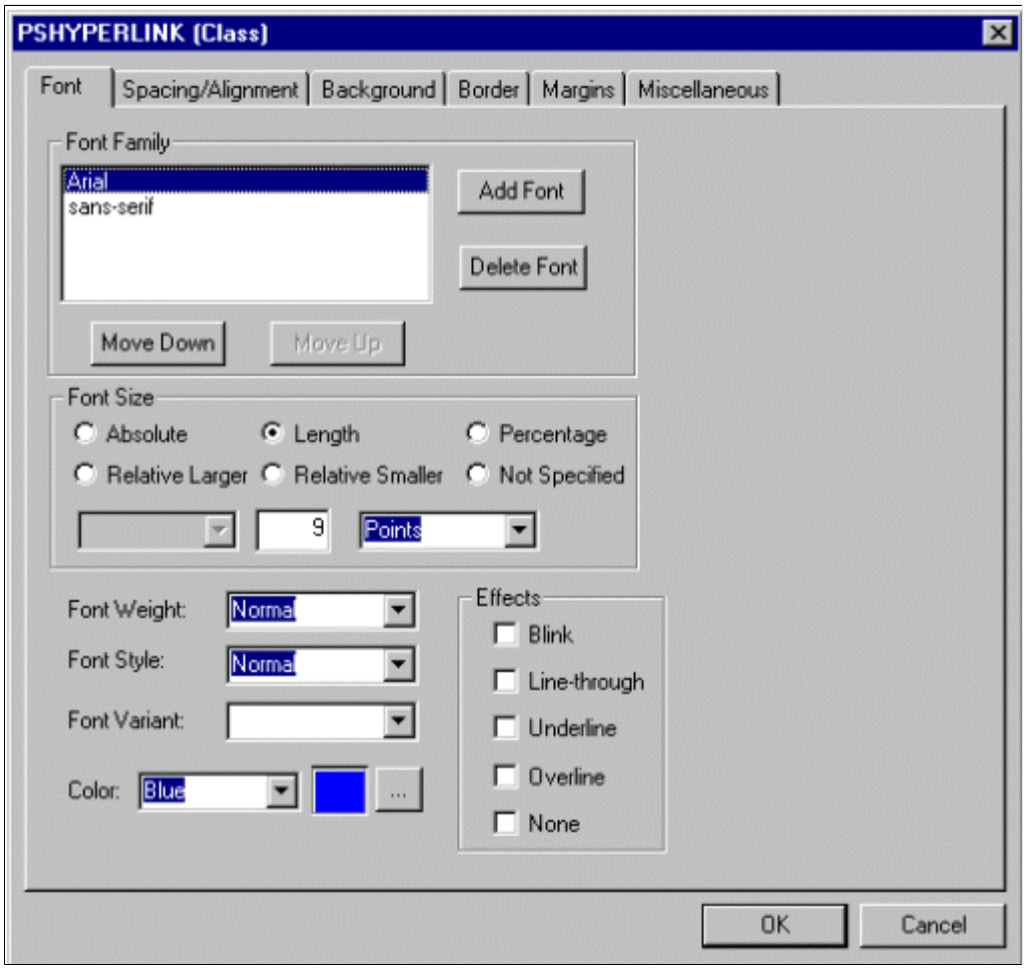
If you access PeopleSoft Application Designer, you can open the designated style sheet and view its definitions in a graphical interface.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Image: Style class definition in PeopleSoft Application Designer

Double-click any style class to view and modify its definition.



If you don't have access to PeopleSoft Application Designer and want to see the CSS file, you can access the file cached for the CSS on the web server. For example, on WebLogic you'll find the file in this directory:

`....\peoplesoft\applications\PORTAL\`*`site_name`*`\cache`

The file is generally named PSSTYLEDEF_1.css or PSSTYLEDEF_SWAN_1; however, it may have a slightly different name. For example, it might have a different numbered suffix.

Some of the most common styles used by pagelets include:

- PSTEXT

  General text label.

- PAADDITIONALINSTRUCTIONS

  Instructional text.

- PABOLDTEXT or PABOLDBLUETEXT

  Commonly used for group and category titles.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- PSPSMALLTEXT

  Commonly used for footers and branding (credits).

- PSPAGE

  Commonly used for background colors.

- PSLEVEL1GRIDODDROW

  Shaded row, which is typically light gray or light blue.

- PSLEVEL1GRIDEVENROW

  Lighter-colored row, which is typically white.

- PSCHECKBOX
- PSDROPDOWNLIST
- PSEDITBOX
- PSGROUPBOX

  Commonly used to create groupings or dividers.

- PSHYPERLINK
- PSPUSHBUTTON
- PSRADIOBUTTON

The following examples show ways in which these styles are used in HTML and translated into elements in the PeopleSoft Pure Internet Architecture interface:

This HTML generates the following interface:

```
<table border='0' cellpadding='0' cellspacing='0' cols='1' CLASS='PSGROUPBOX' style⇒
="border-style:none;" width='639'>  <tr>
   <td class='PSGROUPBOXLABEL' align='LEFT'>Language Settings</td>
  </tr>
   <td colspan='2'  valign='top' align='LEFT'>
    <label for='PSOPTIONS_LANGUAGE_CD' class='PSDROPDOWNLABEL'>Language Code:
    </label>
   </td>
   <td  valign='top' align='LEFT'>
    <span class='PSDROPDOWNLIST' >English</span>
   </td>
...
<input type='check box' name='PSOPTIONS_TRANS_CHG_LASTUPD' id='PSOPTIONS_TRANS_CHG_⇒
LASTUPD' tabindex='14' value="Y" />
 <label for='PSOPTIONS_TRANS_CHG_LASTUPD' class='PSCHECKBOX'>
   Translations Change Last Update
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
 </label>
```

Image: Language Settings on the PeopleTools Options page

This example illustrates the fields and controls in the Language Settings group box of the PeopleTools Options page.

This HTML generates the following interface:

```
<input type='text' name='STYLESHEET' id='STYLESHEET' tabindex='28' value="" class='⇒
PSEDITBOX' style="width:23px; text-align:RIGHT; " maxlength='2'/>
  <label for='PSOPTIONS_TEMPTBLINSTANCES' class='PSEDITBOXLABEL'>    Style Sheet Na⇒
me:
  </label>
```

Image: Style Sheet Name field on the PeopleTools Options page

This example illustrates the Style Sheet Name field of the PeopleTools Options page.

Links and URLs

When processing page-based templates, the portal servlet uses a process called proxying to help ensure that users always stay within the context of the portal and that familiar portal features such as the universal navigation header do not disappear when a user clicks a link.

When a user signs in to a PeopleSoft portal, he or she signs in to a web server on which the portal servlet is running. The portal servlet processes all HTML that isn't in the simple URL format, converting all URL references to point to the portal web server rather than the original URL. The original URL is still necessary to retrieve the requested content; it is stored in the new URL in the URL query string parameter. The portal servlet proxies all links and all form actions in this manner.

For example, assume that a user requests a page from an external website through a proxied link in the portal. The request arrives at the portal web server, invoking the portal servlet. The portal servlet then programmatically retrieves the page from the web server associated with the requested page. It proxies all the links on the retrieved response and sends the page (the contents of the HTTP response) back to the browser, formatted as the user would expect within the portal.

Note: If URLs are included in your HTML, you must use absolute URLs as opposed to relative URLS.

When a URL is invoked on a target page, as opposed to the homepage, the content associated with the URL is rendered within the target frame. The PeopleSoft Interaction Hub and header and left-hand area will remain. Therefore, proxying isn't required to have the new content rendered in the context of the PeopleSoft Interaction Hub.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Performance

Homepages are cached on each user's browser. The browser does not access the web server after the homepage is initially retrieved. You can enable or disable this feature, and also adjust the time interval before the web server is accessed again to get a fresh homepage. In any case, if a user clicks the browser Refresh button, the homepage is accessed from the web server again, overwriting the homepage cached on the browser.

Important! If any homepage pagelet in your implementation uses interactive mode, you must disable browser homepage caching in the web profile.

The following PeopleCode function is used to trigger a refresh of the homepage:

```
FUNCLIB_PORTAL.TEMPLATE_FUNC.FieldFormula.ForceRefreshHomePage().
```

Additionally, the following configuration properties are associated with homepage caching. Any changes to these settings are applied to all users signing in to the web server.

- PortalCacheHomepageOnBrowser=<*True* or *False*>

  If set to *True,* the homepage is cached on the browser. If set to *False,* the homepage is not cached on the browser.

- PortalHomepageStaleInterval=<*seconds until stale*>

  A homepage cached on the browser is considered stale after the specified number of seconds. The next time a user accesses the homepage by clicking a link, the web server is accessed and the homepage is refreshed on the browser.

Because different browser versions do not process HTML in exactly the same way, the browserprops.xml file on the web server on which the portal servlet is installed enables you to disable homepage caching for selected browser versions.

This can be useful if you have one or two supported browsers and want to disable cache for nonstandard browsers that could pose an administration problem. Follow the instructions in the file to disable caching for certain browser types.

As with homepages, navigation pages are cached on each user's browser. You can set options for navigation caching by using the Time page held in cache (METAXP) option.

Set the Time page held in cache option by navigating to My Personalizations and clicking the Personalize Option button for the General Options personalization category. Note that this option is set in minutes, not seconds. A change to this option is picked up by the application server immediately. However, because the users' browsers already have cache control set by the previous value of the option, you must delete the browser cache for the new Time page held in cache value to take effect.

Oracle enables you to prevent the system from caching PeopleSoft pages to the browser. You control browser caching using the EnableBrowserCache property in the configuration.properties file.

Being able to control browser caching is useful for situations in which PeopleSoft applications are deployed to kiosk workstations where multiple users access the applications. Enabling you to prevent caching means that users can't click the Back button to view another individual's transaction or view any other sensitive data.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The side effect of completely disabling caching is degraded performance. For each new page, the system must access the database. However, PeopleTools offers a compromise related to browser caching in the form of the Time page held in cache (METAXP) option discussed earlier.

You can enable browser caching for the navigation pages that remain relatively static. This option applies to PeopleSoft Pure Internet Architecture navigation pages, portal homepages, and navigation pages. Use it to take advantage of the performance gains of caching while limiting the amount of time that navigation pages—the menu pages—remain in cache.

The Time page held in cache option is set to 20 by default. To disable this option, enter *0* in the Override Value edit box. The minimum value in minutes for this option is *0* (disabled) and the maximum value is *525600,* which is one year.

If the Time page held in cache option is set to *20,* and if you assume that the time is 7 p.m. on August 24, then the header information in the HTML is dated like this:

```
Wed 24 Aug 2011 07:20:00 PM
```

This header information indicates that in 20 minutes the system needs to check for a new page. This reduces the performance degradation when no caching occurs at all.

By default, the EnableBrowserCache property is set to *True.*

If the EnableBrowserCache property is set to *False*:

- The system never caches pages. When a user clicks the browser Back button, she receives a Page Expired message in .
- The setting overrides any date and time header settings.

The following table helps illustrate the way in which the EnableBrowserCache option works with the METAXP option.

EnableBrowserCache	***METAXP***	***Caching Behavior***
True	*0*	No caching due to the *0* value in METAXP.
True	*> 1*	Pages are cached with expiration values set in Greenwich Mean Time (GMT) based on the Time page held in cache value (METAXP).
False	*0*	No caching.
False	*>1*	No caching. The EnableBrowserCache option setting overrides the Time page held in cache value (METAXP).

Multilanguage Support

PeopleSoft applications currently support the following 33 languages:

- Arabic.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Bahasa Malay.
- Bulgarian.
- Canadian French.
- Chinese (Simplified).
- Chinese (Traditional).
- Croatian.
- Czech.
- Danish.
- Dutch.
- English.
- Finnish.
- French.
- German.
- Greek.
- Hebrew.
- Hungarian.
- Italian.
- Japanese.
- Korean.
- Norwegian.
- Polish.
- Portuguese.
- Romanian.
- Russian.
- Serbian.
- Slovak.
- Slovenian.
- Spanish.
- Swedish.
- Thai.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Turkish.
- UK English.

Pagelet Wizard can access the PeopleSoft runtime environment to determine details of a user's profile, such as his or her language. For more information about how you can use this Pagelet Wizard feature to facilitate passing runtime parameters, such as language, see Pagelet Wizard documentation in this PeopleBook.

See Understanding Pagelet Wizard.

In addition, Java classes delivered with PeopleTools enable you to call PeopleCode from your Java program and access contextual information from the runtime system. If needed, language can be retrieved through a PeopleCode function that is accessible from Java. Further information about this feature can be found in this document.

See Developing Java-Based Pagelets.

Pagelet Development Tools

You can create a pagelet using several ways. Ultimately, the portal servlet assembles HTML, so the key is determining how to generate the HTML. Some methods leverage PeopleTools, while other options allow pagelet creation without PeopleTools.

One set of options is to develop pagelets with PeopleTools. This is the most straightforward approach if you are dealing solely with data from PeopleSoft applications. The two types of PeopleTools-based pagelets are:

- PeopleSoft pages.

  PeopleTools dynamically generates the appropriate HTML to render the page data based on the definitions created within PeopleSoft Application Designer. This is the most straightforward approach when the data being rendered is in a PeopleSoft application database.

- iScripts.

  You can write a PeopleCode function that renders HTML using %Request and %Response objects, which are similar to ASP or JSP programs. This allows more control over data retrieved and enables you to conditionally render HTML. This approach gives you maximum flexibility, but unlike the PeopleSoft Pure Internet Architecture page approach, you must code for things such as multilanguage and browser support.

The focus of this document is a set of options that can be used to create a pagelet application with one of the many technologies that generate HTML. This may be the approach you take if you're integrating a non-PeopleSoft system with the PeopleSoft Interaction Hub. For example, you're applying business logic to data from a non-PeopleSoft system, or you might be combining data from a PeopleSoft application with other systems.

The pagelet can be written in a variety of technologies, including:

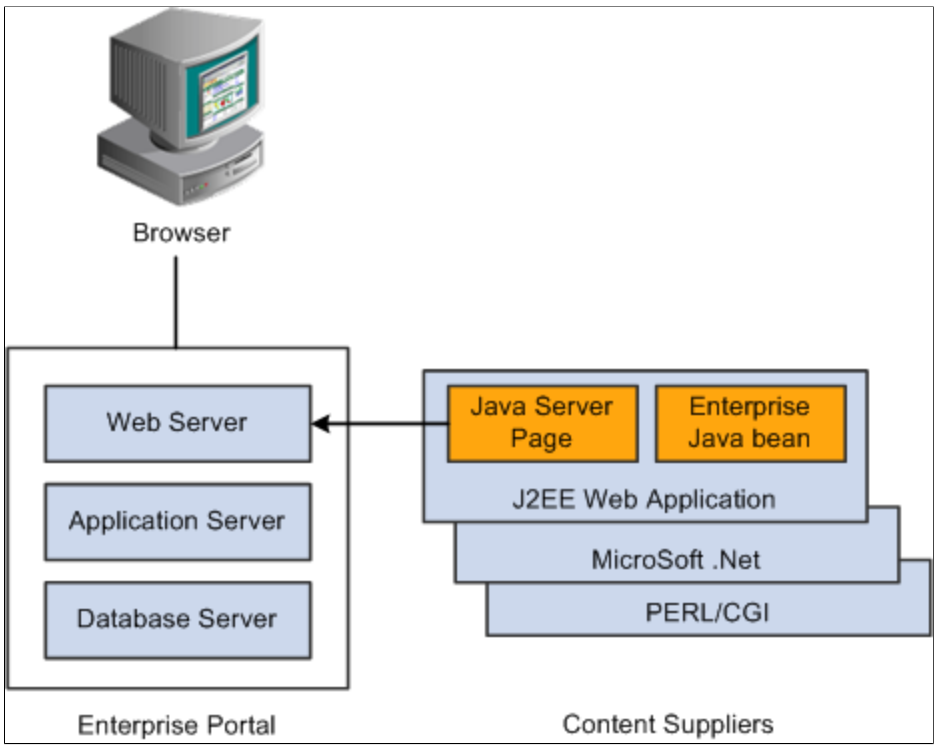
- Adobe ColdFusion.
- AJAX.
- ASP.
- CGI.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- HTML.
- Internet server API (ISAPI).
- JavaScript.
- JSP.
- Perl.
- Servlets.
- Tool Command Language (Tcl).

Image: Developing in a PeopleSoft environment with non-PeopleTools technologies

The following diagram illustrates some of the ways that you develop in a PeopleSoft environment using technology other than PeopleTools:



You can convert your own HTML into a pagelet in several ways:

- Pagelet Wizard.

  Pagelet Wizard walks you through a series of steps involved in creating and publishing a pagelet. Portal administrators and nontechnical users can integrate and transform data from a wide variety of data sources, both internal and external to PeopleSoft applications.

  External sources include web applications that can be referenced with a URL, HTML block (such as a form or table), and Java classes. You do not need to have PeopleSoft-specific application development tools or skills to be able to create pagelets.

- Registered URL.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

You can also choose to register a URL, such as a JSP or ASP, directly in the portal registry.

Pagelet Sizes

The homepage layout can vary based on end-user personalization. These two layout choices are available for homepages and dashboard pages:

- The three-column layout divides the homepage into three equally-sized columns. In PeopleSoft applications, these are called narrow columns.
- The two-column layout divides the homepage into two differently-sized columns: one narrow column and one wide column, which is twice the width of the narrow column.

Because pagelets must function in two possible homepage layouts, it is column size that drives pagelet size. This fact is evident in the three categories of pagelet widths that are listed here:

- Narrow pagelets.
- Wide pagelets.
- Banner pagelets.

Narrow Pagelets

With three-column layout, all pagelets are rendered as narrow pagelets. Narrow pagelets are designed to be approximately 300 pixels wide. Subtracting the border and the internal margin leaves 285 pixels for the content. A narrow pagelet appears in a column that is one-third the width of the portal homepage. Because the pagelet is narrow, you should provide a succinct list of values that users can quickly traverse. Select the minimum set of data that best encapsulate the pagelet content. A narrow pagelet typically accommodates a single 30-character field.

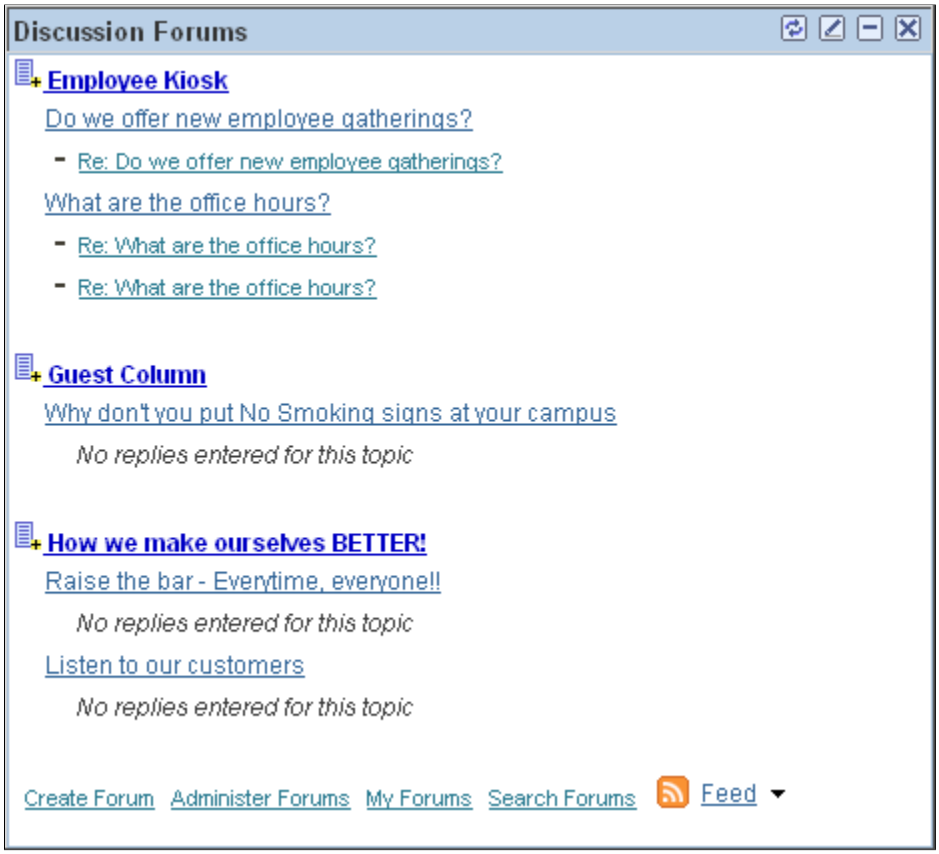
A pagelet that has been designed to fit in a narrow column can be rendered in a wide column as well. However, user interface issues can occur when a pagelet that has been designed to render in a wide

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

column only is rendered in a narrow column. Therefore, a pagelet should always be able to operate in a narrow format.

Image: Example of the Discussion Forums pagelet as a narrow pagelet

This example shows the Discussion Forums pagelet as a narrow pagelet. The essential links and information are displayed in the narrow format.



Generally, all pagelets should be developed with the narrow column size as the default. However, a pagelet can be designed to take advantage of the additional space when it is rendered in a wide column.

Wide Pagelets

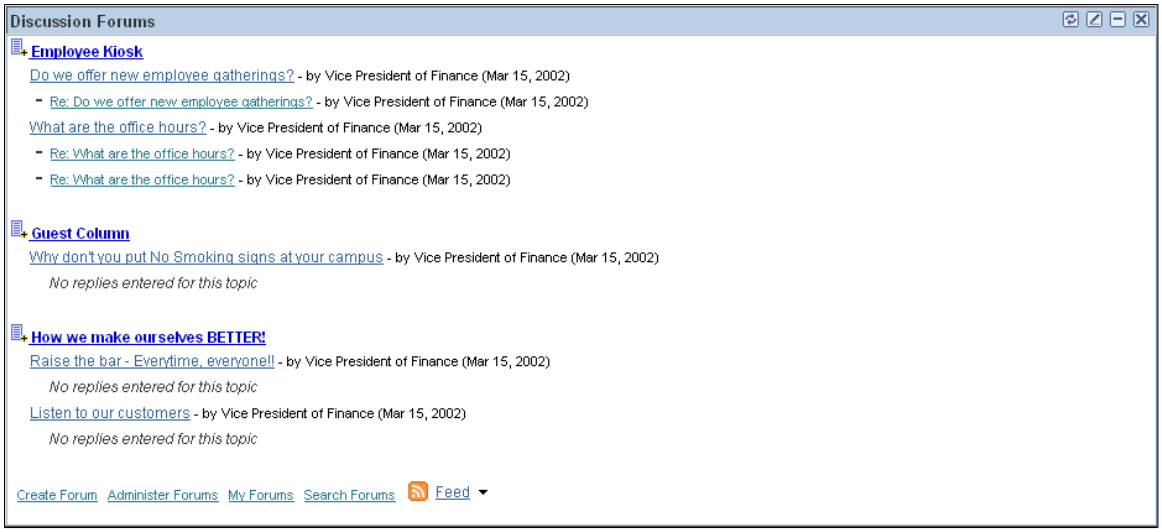
With two-column layout, the first column allows spacing for a narrow pagelet and the second column allows spacing for a wide pagelet. Wide pagelets are designed to span two-thirds of the width of the

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

homepage. Subtracting the border and the internal margin leaves 650 pixels for the content. Although you can fit more data on the pagelet, the data must remain meaningful.

Image: Example of the Discussion Forums pagelet as a wide pagelet

This example shows the Discussion Forums pagelet as a wid pagelet. In addition to the essential links and information that are also displayed in the narrow format, the wide format includes who posted a discussion topic or reply and when.



When the layout contains the wide column, the **PORTALPARAM_COMPWIDTH** query string parameter is set to wide and passed to the HTML denote the wide version. Developing a wide version of a pagelet is optional.

The PORTALPARAM_COMPWIDTH Parameter

The parameter `PORTALPARAM_COMPWIDTH` can contain the value *Wide* or *Narrow* and can be used to specify whether a pagelet is being rendered in a narrow or wide column. The function **pgltAtion**does not check or use this parameter, the narrow pagelet will just be rendered in a wider area.

This sample PeopleCode references the query string parameter `PORTALPARAM_COMPWIDTH:`

```
Component string &CompWidth;

&CompWidth = %Request.GetParameter("PORTALPARAM_COMPWIDTH");

If &CompWidth = "Wide" Then;
  TransferPage(Page.PT_CONTRK_PGLT_W);
End-If;
```

Banner Pagelets

Banner pagelets are the widest of all pagelets. Banner pagelets extend across the entire width of the homepage or dashboard page and span all columns regardless of layout. Both the two-column and three-column layouts enable you to add one banner pagelet that is positioned at the top of the page. Banner pagelets are designed to be 1240 pixels wide.

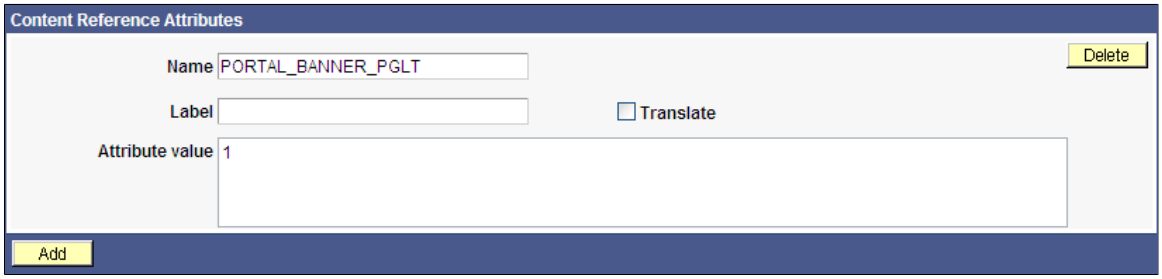
Note: The actual number of pixels that are available for content varies based on the resolution of your display and the rendering engine of your browser.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The portal uses the content reference attribute, *PORTAL_BANNER_PGLT* to appropriately position banner pagelets. You must manually enter this attribute name and its accompanying value.

Image: PORTAL_BANNER_PGLT content reference attribute

This example shows the entry for the PORTAL_BANNER_PGLT content reference attribute:



Important! Homepages and dashboard pages can display one banner pagelet only. If more than one pagelet on a homepage includes the PORTAL_BANNER_PGLT attribute, the system recognizes the attribute for the first pagelet and ignores the attribute for any other pagelets.

See Creating Banner Pagelets.

Pagelet Extensions

Pagelet extensions are standard pages. They can be registered with a template that allows it to use the entire width of the browser with no side frame for navigation, or with a template that includes a side frame.

In the former case, no inherent sizing requirements are imposed. PeopleSoft pages are designed for browsers running with a resolution of at least 1280 x 1024. Therefore, when a side frame is not included, a pagelet extension should not be wider than 1190 pixels, accounting for borders and so forth.

In the latter case, a pagelet extension should not be wider than 375 pixels.

HTML-Related Design Considerations

Consider these HTML-related guidelines when you are designing pagelets:

- Ensure that the pagelet encapsulates data to provide at-a-glance summary information.
- Ensure that the pagelet provides links to detailed application pages.
- Avoid large borders or a design that creates extraneous white space. Screen area on a homepage is valuable and data should be maximized.
- Avoid designs that make the pagelet wider than the prescribed size. This undesirable design forces the user to scroll horizontally, so you should design your HTML to be vertically oriented. For instance, radio buttons should be arranged vertically, not side-by-side. When you create a form, buttons should appear below a text box, not next to it.
- Because this is a business-oriented portal, avoid extravagant or extraneous graphics. If you use graphics, they should be purposeful and unobtrusive. No graphic should be wider than 218 pixels, which would force the pagelet to be wider than a narrow pagelet.
- Avoid using any text or HTML tags that force the pagelet to a width that is greater than 218 pixels.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- In general, avoid explicit sizing. Let the browser render content in the space that is available, which should allow your text and graphics to fit appropriately. For example, place text in an HTML table to enable the table to wrap any long strings.
- When appropriate, use the PeopleSoft style sheet to ensure that the appearance and character of your pagelet is consistent with the rest of the PeopleTools portal content.
- When your pagelet does not use the entire width of the column, center the pagelet content.
- If a link on the pagelet takes the user to another website and you want to display the content in a new window, rather than using *target=_blank* in the link tag, use the following code:

  ```
  <a href="javascript:void window.open
  ('http://www.company.com/cgi-bin?article3');" class="PSHYPERLINK">
  ```

- If a pagelet has personalization options, the pagelet should have a default mode. In default mode, ensure that the pagelet includes a standard set of data and a message conveying that default data can be personalized.

PeopleSoft Interaction Hub Guidelines

This section lists some HTML-related development guidelines for pagelets that you use in the PeopleSoft Interaction Hub.

- Within PeopleTools, attributes from all content are merged into the single <BODY> tag; be aware of the order of attribute precedence.

  The order of attribute precedence is:

  1. Template.
  2. Target content.
  3. Template components.

  Events beginning with *on* are naively combined, so if one of them issues a *return,* the following events will not run. All other attributes are not examined; rather they are used in order of precedence. For example, if both the template and a content component contain *topmargin,* the template attribute will be applied.

- Avoid large borders or anything that creates extraneous white space. Screen area on a homepage is valuable and data should be maximized.
- Avoid implementing anything that will make the pagelet width wider than the prescribed size. Making the user scroll horizontally is undesirable.

  Make your HTML vertically oriented. For instance, radio buttons should be placed vertically, not side-by-side. When you create a form, any buttons should appear under a text box, not next to it.

- Avoid using graphics that do not have thoughtful purpose. This is a business-oriented portal, so extravagant or extraneous images should not be used. If any are used, they should be purposeful and unobtrusive. No graphic should be wider than 218 pixels. A graphic larger than this size will force the width of the pagelet to exceed the size of a narrow pagelet.
- Avoid using any text or directives that force the pagelet to a width greater than 218 pixels.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- In general, avoid explicit sizing and allow the portal to set the width. Your text and graphics should then fit appropriately. For instance, placing text within an HTML table will allow it to wrap any long strings.
- When appropriate, use the PeopleSoft style sheet to ensure that the appearance and character of your pagelet is consistent with the rest of the PeopleSoft Interaction Hub content.
- When your pagelet does not use the entire width of the column, center the pagelet content.
- If a link takes the user to another website and you want it to display the content in a new window, rather than using *target=_blank* within the link tag, use:`<a href="javascript:void window.open ('http://www.company.com/cgi-bin?article3');" class="PSHYPERLINK">`

Branding Considerations

As mentioned earlier, this is a business-oriented portal. Although content and name recognition are important, they are secondary to the primary goal of enabling end users to perform their work in a more efficient manner. Thus, any branding should be subtle and never detract from the pagelet content or the rest of the portal.

Thus, for a user to operate within the PeopleSoft Interaction Hub product, a pagelet developed outside of PeopleSoft can credit the source developer, but must follow these standards:

- Branding on a homepage pagelet should be done by means of text only and should be placed at the bottom of the pagelet. No organization logos can be placed on the pagelet without permission from Oracle. You can use “Provided by *XXX*,” where *XXX* is the name of your company. Also, *XXX* can be a link to an appropriate web page.
- Organization logos and further information about your organization, products, and services should be located on the pagelet personalization page or other pagelet extensions.
- Generally, graphics for organization logos should not be larger than 100 pixels wide x 40 pixels high.

Image: Sample pagelet

If you wanted to create a pagelet that an associate could use to search for keywords on another website, it might look like the following example.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The HTML used is shown here:

```
<table border=0 cellspacing=1 cellpadding=0>
  <tr>
   <td><!-- start of search form -->
    <form method="POST" action="http://www.contentSource.com/cgi-bin/dofind.cgi">
     <center>
     <font class="PSEDITBOXLABEL">Enter keyword:</font>
     <input name="word" value="">
     <br>
     <input type="submit" value="Search"></center>
    </form>
<!-- end of search form --></td>
</tr>

<tr>
  <td><!-- start of optional directions -->
   <font class="PAADDITIONALINSTRUCTIONS">General directions can be provided here.
    Remember that pagelets should be as easy and intuitive to use
    as possible.<br>
    <br>
   </font>
<!-- end of optional directions --></td>
</tr>

<tr>
  <td><!-- start of footer for branding -->
   <center>
   <font class="PSPSMALLTEXT">Provided by
    <a href="http://www.contentSource.com/">Content Source Company</a>
   </font>
   </center>
<!-- end of footer for branding --></td>
</tr>
</table>
```

Note how this sample pagelet adheres to the standards mentioned previously:

- It uses a table to help wrap words, rather than allowing long strings to dictate the width of the pagelet.
- All text uses the PeopleSoft styles, such as PSEDITBOXLABEL, PAADDITIONALINSTRUCTIONS, and PSPSMALLTEXT.
- The branding is small and no graphics are used, thus adhering to the branding standards mentioned previously.

Understanding Single Signon and Pagelets

This section discusses:

- Single signon.
- The PeopleSoft authentication process.
- The PeopleSoft authentication API.

Single Signon

The examples shown thus far have used publicly-available URLs. Even if the examples represent third-party applications, the discussion has concentrated on retrieving data and rendering a pagelet. The topic has not yet discussed the possible need to sign in to a non-PeopleSoft system.

When two or more systems need separate authentication, automating the sign in process is preferable. Needing to manually sign in to several different systems each day is inconvenient and annoying. Users

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

often expect a business portal to be similar to accessing a variety of internet websites. Once signed in to the portal, a user should rarely (if ever) need to sign in to another system.

Accomplishing single signon between PeopleSoft and other systems can be done in several ways. First, you need to determine the primary (or master) and secondary (or slave) authentication systems.

- PeopleSoft system as primary.

  Once a PeopleSoft user has signed in, an authentication cookie is sent to the browser's memory. Other applications can choose to authenticate using this cookie. Oracle provides an API that other applications can leverage. This is the option that is discussed in detail in this section.

- PeopleSoft system as secondary.

  The PeopleSoft authentication process is flexible enough to enable accessing another system.

  See "Enabling Signon PeopleCode for LDAP Authentication" (PeopleTools 8.54: Security Administration).

- PeopleSoft system and other applications as secondary to third-party authentication system.

  A variant of the previous option would be for all applications (including PeopleSoft) to leverage, or trust, a third-party authentication system such as Netegrity, Oblix, or Securant.

  If you are writing a pagelet for the PeopleSoft portal, you have no guarantee that all possible customers for the pagelet will have access to a third-party authentication system. Thus, this option is not discussed in this PeopleBook.

PeopleSoft Authentication Process

Before discussing how your pagelet could leverage PeopleSoft authentication, you need to understand the authentication process.

After the first application server or node authenticates a user, the PeopleSoft application delivers a web browser cookie containing an authentication token. PeopleSoft Pure Internet Architecture uses web browser cookies to store a unique access token for all users after they are initially authenticated. When the user connects to another PeopleSoft application server/node, the second application server uses the token in the browser cookie to reauthenticate the user in the background so that they do not have to complete the sign-in process again. Your non-PeopleSoft application could do something similar.

Single signon is critical for PeopleSoft portal implementations because the portal integrates content from various data sources and application servers and presents them in a unified interface. When users sign in through the portal, they always take advantage of single signon. Users need to sign in once and be able to navigate freely without encountering numerous sign-in screens.

The following table presents the fields that appear in the PeopleSoft authentication token.

Field	***Description***
UserID	Contains the user ID of the user to which the server issued the token. When the browser submits this token for single sign-on, this is the user that the application server signs in to the system.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Field	*Description*
Language Code	Specifies the language code of the user. When the system uses this token for single sign-on, it sets the language code for the session based on this value.
Date and Time Issued	Specifies the date and time that the token was first issued. The system uses this field to enforce a time-out interval for the single sign-on token. Any application server that accepts tokens for sign-in has a *time-out minutes* parameter configured at the system level. A system administrator sets this parameter using the Single Signon page. The value is in Greenwich Mean Time (GMT), so the application server's time zone is irrelevant.
Issuing System	Specifies the name of the system that issued the token. When it creates the token, the application server retrieves this value from the database. Specifically, it retrieves the defined local node. Single sign-on is not related to PeopleSoft Integration Broker messaging, except that single sign-on functionality leverages the messaging concept of nodes and local nodes. You configure only a node to trust single sign-on tokens from specific nodes. Consequently, an application server needs an Issuing System value so that it can check against its list of trusted nodes to determine whether it trusts the issued token.
Signature	Contains a digital signature that enables the application server using a token for single sign-on to ensure that the token hasn't been tampered with after it was originally issued. The system issuing the token generates the signature by concatenating the contents of the token (all of the fields that appear in this table) with the message node password for the local node. The system then hashes the resulting string using the SHA1 hash algorithm for example, `signature = SHA1_Hash ( UserID + Lang + Date Time issued + Issuing System + Local Node Pswd )` **Note:** The "+" indicates concatenation. Only one way is available to derive the 160 bits of data that make up the signature, and that is by hashing exactly the same user ID, language, date time, issuing system, and node password. **Note:** If you are using digital certificate authentication, the signature of the digital certificate occupies this space. The preceding description applies only to using password authentication.

Note: Single signon does not depend on the use of an LDAP (Lightweight Directory Access Protocol) directory. You can store user credentials in an LDAP directory if desired, but it is not required.

These are the key security features of the cookie authentication token:

- The cookie exists in memory; it is not written to disk.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- No password is stored in the cookie.
- You can set the expiration of the cookie to be a matter of minutes or hours. This expiration option is a useful security feature.
- The cookie is encrypted and digitally signed using a checksum to prevent tampering.

The PeopleSoft Authentication API

Oracle delivers a component interface named PRTL_SS_CI, which enables external applications to seamlessly integrate a single sign-on solution with the PeopleSoft portal applications. This component interface helps ensure that users who have already signed in to the portal don't have to sign in again for every system that you reference in your portal.

Component interfaces are the focal points for externalizing access to existing PeopleSoft components. They provide real-time synchronous access to the PeopleSoft business rules and data associated with a component outside the PeopleSoft online system. Component interfaces can be viewed as *black boxes* that encapsulate PeopleSoft data and business processes, and hide the details of the structure and implementation of the underlying page and data.

To take advantage of the Single Signon API, you need to create a custom API, which includes building the dynamic link libraries, classes, and registry settings necessary to enable an external application to communicate with the portal. This can be done automatically through PeopleTools. More information about building dynamic link libraries, classes, and registry settings, as well as other details about PeopleSoft component interfaces, can be found in *PeopleTools: Component Interfaces*

See Understanding the Authentication DomainSecurity Properties.

Only external applications, such as COM, C/C++, or Java programs require a component interface API. PeopleCode programs do not require a component interface API and, in fact, Oracle does not recommend building a component interface API if the component interface is to be accessed from PeopleCode only.

The files of your custom API need to reside on the client machine, that is, the web server for ASP and the machine running the Java program for Java. The registry file may also need to be run to update the registry with the new libraries.

The Signon Process with the API

The PRTL_SS_CI component interface contains two user-defined methods:

- **Authenticate( ).**

  Your external authentication program distributes an authentication token that can be retrieved from a cookie in the browser. The Authenticate function determines whether an authentication token is valid.

- **GetUserID( )**.

  If the token is valid, you use the GetUserID function to retrieve the user ID associated with the authentication token.

Before reading about the development requirements of your API, examine the steps that occur internally when you use the API in conjunction with the delivered PRTL_SS_CI:

1. The user enters the user ID and password into the PeopleSoft portal sign-in page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. If the sign-in to the portal application server is successful, the server generates a single signon token. The web server receives the single sign-on token from the application server and issues a cookie to the browser.

3. The user navigates in the portal and encounters a link to the external system. The user clicks the link.

4. The browser passes the PS_TOKEN cookie to the external web server.

5. The external web server checks for the PS_TOKEN cookie before displaying a sign-in page.

6. After the system determines that the user is accessing the application through the PeopleSoft portal, you retrieve the authentication token and send it to the PRTL_SS_CI component interface to verify authentication. For instance, it calls PRTL_SS_CI.Authenticate(Auth. token string).

7. After the system authenticates the token, it can then make calls to the PRTL_SS_CI.Get_UserID() function to return the appropriate user ID.

In general, cookies are not transferable across domains. The only domain that can access the cookie is the domain that created it. Therefore, the web server for the non-PeopleSoft system must be on the same domain as the PeopleSoft system so that the cookies are passed appropriately.

External Application Support for Single Signn

Developers of external applications need to alter the sign-on process to conform to the following requirements:

1. Check for the PS_TOKEN cookie. If the cookie doesn't exist, continue with the normal sign-in process. Otherwise, bypass the sign-in page.

2. Retrieve the authentication token from the PS_TOKEN cookie.

3. Connect to the PeopleSoft portal through the PRTL_SS_CI API.

4. Pass the authentication token to the **Authenticate()** function of the API.

5. If **Authenticate()** returns *True,* retrieve the user ID associated with the authentication token using the Get_UserID() function.

Authentication API PeopleCode Example

The following PeopleCode example shows the process of validating your authentication token and retrieving the user ID. This example is designed to provide a general idea of the process involved and to help you incorporate the PRTL_SS_CI API into your sign-in process.

```
Local ApiObject &THISSESSION;
Local ApiObject &THISCI;
Local string &AUTHTKN;
/* Assigns the Authentication Token to a variable */
  &AUTHTKN = %AuthenticationToken;
/* Open a session and make a connection */
  &THISSESSION = GetSession();
   If &THISSESSION.connect(1, "EXISTING", "", "", 0) <> True Then
    WinMessage(MsgGet(30000, 1, "Session Connect Failed."));
    Exit (1);
   End-If;
/* Retrieves the component interface PRTL_SS_CI */
  &THISCI = &THISSESSION.GetCompIntfc(CompIntfc.PRTL_SS_CI);
/* Checks to see if the component interface is NULL */
   If &THISCI = Null Then
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
    WinMessage("Component Interface PRTL_SS_CI not found.
    Please ensure Component Interface Security access is granted
    to this user.");
    Exit (1);
   End-If;
/* Key fields would usually be set before the Get() function is
called in order to map the component interface to a particular
set of data. This component interface is not mapped to data.
* Therefore, the component interface is retrieved and then the
user defined methods are retrieved */
  &THISCI.get();
  PRTL_AUTH = &THISCI.Authenticate(&AUTHTKN);
  PRTL_USER_ID = &THISCI.Get_UserID();
```

Note: The component interface is not mapped to data because the key field for the data would be the authentication token. This token is dynamically assigned when the user signs in to the portal, and it is not stored as data anywhere in the system. Therefore, no key fields are needed and the token is passed directly to the user-defined functions.

Authentication API Java Example

Here is an example of a similar operation written in Java. This is a file named SingleSignon.java.

```
package examples.migration.sso;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import psft.pt8.joa.*;
import PeopleSoft.Generated.CompIntfc.*;

public class SingleSignon extends HttpServlet {
  public static ISession oSession;
  String psfttoken;
  public static void ErrorHandler() {
    //***** Display PeopleSoft Error Messages *****
if (oSession.getErrorPending() || oSession.getWarningPending()) {
     IPSMessageCollection oPSMessageCollection;
     IPSMessage oPSMessage;

     oPSMessageCollection = oSession.getPSMessages();
for (int i = 0; i < oPSMessageCollection.getCount(); i++) {
      oPSMessage = oPSMessageCollection.item(i);
      if (oPSMessage != null)
       System.out.println("(" + oPSMessage.getMessageSetNumber() + ","
 + oPSMessage.getMessageSetNumber() + ") : " + oPSMessage.getText());
     }
     //***** Done processing messages in the collection;
OK to delete *****
     oPSMessageCollection.deleteAll();
    }
 }

  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    try {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    Cookie[] cookies = request.getCookies();
     if (cookies == null) {
       out.println("<TR><TH COLSPAN=2>No cookies");
     } else {
       Cookie cookie;
       for(int i=0; i<cookies.length; i++) {
         cookie = cookies[i];
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
            String pstoken = cookie.getName();
               psfttoken = cookie.getValue();
               if (pstoken.equals ("PS_TOKEN"))

                  out.println("<TR>\n" +
                      "  <TD>" + pstoken + "\n" +
                      "  <TD>" + psfttoken);
         }

       }

       String strServerName, strServerPort, strAppServerPath;
    String strUserID, strPassword;
    strServerName = "jfinnon09";
                  strServerPort = "9500";
                  strUserID = "VP1";
                  strPassword = "VP1";

    //Build Application Server Path
    strAppServerPath = strServerName + ":" + strServerPort;

        //***** Create PeopleSoft Session Object *****
        //       ISession oSession;

    oSession = API.createSession();

    //***** Connect to the App Server *****
    if (!oSession.connect(1, strAppServerPath, strUserID, strPassword,
   null)) {
     out.println("\nUnable for Jason to Connect to
  Application Server.");
     ErrorHandler();
     return;
    }

       //***** Get Component Interface *****
    IPrtlSsCi oPrtlSsCi;
    String ciName;
     ciName = "PRTL_SS_CI";
    oPrtlSsCi = (IPrtlSsCi) oSession.getCompIntfc(ciName);
    if (oPrtlSsCi == null) {
     out.println("\nUnable to Get Component Interface " + ciName);
     ErrorHandler();
     return;
    }

    //***** Set the Component Interface Mode *****
      oPrtlSsCi.setInteractiveMode(false);
      oPrtlSsCi.setGetHistoryItems(true);
      oPrtlSsCi.setEditHistoryItems(false);

    //***** Set Component Interface Get/Create Keys *****

      //***** Execute Get *****
      if (!oPrtlSsCi.get()) {
       out.println("\nNo rows exist for the specified keys.
   \nFailed to get the Component Interface.");
       ErrorHandler();
       return;
      }

      //***** BEGIN:  Set/Get Component Interface Properties *****

      //***** Set Level 0 Properties *****
      //out.println("oPrtlSsCi.PrtlToken: " +
   oPrtlSsCi.getPrtlToken());
      //out.println("Checking token: " + psfttoken);

      //oPrtlSsCi.setPrtlToken(<*>);
                    //String psfttoken;
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
                  //psfttoken = oPrtlSsCi.getPrtlToken();
                  //System.out.println("oPrtlSsCi.PrtlToken: "
 + psfttoken);

    //***** END:  Set Component Interface Properties *****

    //***** Execute Standard and Custom Methods *****
    //***** Execute Authenticate *****
    boolean auth;

           auth = oPrtlSsCi.authenticate(psfttoken);
           //out.println("Auth: " + auth);
    //Execute Get_UserID
                  String psftuser;
    psftuser = oPrtlSsCi.getUserid();
                //out.println("Psftuserid: " + psftuser);

                 String title = "Welcome  " + psftuser;

                //****** Get HTML ***********
                 out.println(ServletUtilities.headWithTitle(title));
                 out.println("</BODY></HTML>");

                //***** Disconnect from the App Server *****
           oSession.disconnect();
           return;
  }


  catch (Exception e) {
    e.printStackTrace();
    System.out.println("An error occurred: ");
    ErrorHandler();
   }
 }


  /** Let the same servlet handle both GET and POST. */

  public void doPost(HttpServletRequest request,
                     HttpServletResponse response)
      throws ServletException, IOException {
    doGet(request, response);
  }

}
```

Developing PeopleSoft Component-Based Pagelets

To develop pagelets that are based on PeopleSoft components:

1. Open PeopleSoft Application Designer.
2. Select File, Open to display the Open Definition dialog box.
3. Use the drop-down list box to select *Page* as the definition.
4. Enter selection criteria and open the desired page.
5. Select File, Definition Properties and then select the Use tab.
6. Select the page size.

   For narrow pagelets, select *240xVar portal home page comp.*

   For wide pagelets, select *490xVar portal home page comp.*

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

For banner pagelets, select *Custom size* then enter these width and height dimensions: *1193* x *259*.

Note: You can adjust the pagelet dimensions at any time in the process.

---

7. Design the new pagelet using the guidelines described previously and similar design techniques that are used to design a page.

8. Save the page.

Note: When the page is added to a component, the search record is specified. As noted in the preceding guidelines, you should avoid a search interaction so that the pagelet can render its data on the homepage without any prompting for keys. If keys are needed, then the pagelet should be coded to use some default key values and the personalization options should initially reflect these defaults.

---

9. Register the pagelet in the portal registry. This step also involves setting up security access to the component and the page.

   See Administering Pagelets in the Portal Registry.

Developing iScript-Based Pagelets

Use an iScript only for pagelets that reference non-PeopleSoft data or if a PeopleSoft page does not provide the necessary functionality. When you develop pagelets that are based on iScripts, you should be aware of the advantages and disadvantages.

Advantages

These are some advantages of iScripts:

- You may be familiar with the technique because it's similar to creating Active Server Pages (ASP) or Java Server Pages (JSP). Basically, you are using a script to produce HTML and JavaScript.
- Greater flexibility is available in pagelet appearance because you have direct control of the HTML or JavaScript.
- Greater flexibility is available in pagelet sizing. PeopleSoft Internet Architecture pages tend to be fairly static in their object placement and sizing.
- Many PeopleSoft Application Designer definitions and their attributes are available for you to take advantage of programmatically. For example, almost all PeopleSoft pages share a consistent appearance through the use of style sheets. Your pagelet, too, can use these styles and thus maintain uniformity of appearance with your PeopleSoft application because iScript-based pagelets can access the same style sheet definitions and their properties,

Disadvantages

These are some disadvantages of iScripts:

- You must manually program all aspects of the transaction. You must handle and account for support for multiple browsers and international considerations, such as language, date and currency formatting, and so on. PeopleTools handles these details automatically on transaction pages and component-based pagelets.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- You must be extremely meticulous and thoughtful. The poor design and implementation of many complex and interrelated programs can lead to loss of functionality and degradation of performance.
- When you develop a template pagelet based on an iScript, some of the functions provided by PeopleSoft Application Designer, such as currency codes, language support, and multiple browser support, are not automatically available in your iScript. Therefore, if you want any such functionality, you must develop it.

Template Pagelets

To create an iScript-based template pagelet:

Note: Use steps 1 and 2 when you are getting HTML from another website or if you want to create some relatively static HTML that is used by the iScript to render a pagelet. If you are creating a pagelet iScript that is going to render highly dynamic HTML, then begin with step 3 and create any necessary or complex PeopleCode to conditionally render the HTML or JavaScript.

---

1. Create the HTML code for the pagelet.

   In many cases you can navigate to a Uniform Resource Locator (URL) that you want to turn into a pagelet and copy the HTML associated with that web page. For form-based web pages, copy all of the HTML code located between the Form tags of an existing HTML document.

2. Store the new HTML code in PeopleSoft Application Designer as an HTML definition.

3. Create an iScript that calls the new HTML definition in a web library.

   A web library is a derived record that is created in PeopleTools for storing iScripts. The name of all web library records begin with WEBLIB_.

   Note: All iScripts should be coded in the FieldFormula PeopleCode event of a derived record called WEBLIB_*XX,* where *XX* is the product code. Only functions that start with *IScript_,* such as IScript_iTracker, can be invoked from a URL.

   ---

4. Find to the appropriate field name and edit the FieldFormula PeopleCode that is associated with the field.

   This is where you create an iScript that calls the HTML definition that you created.

5. Create a new iScript and give it a meaningful name.

   An easy way to create a new iScript is to copy and paste an existing iScript that is located in the web library and then make the desired changes. All iScript names must begin with *IScript_*.

6. Save the web library.

7. Register the iScript as a pagelet in the portal registry. This step also involves setting up security access to the iScript.

   See Administering Pagelets in the Portal Registry.

Modifying Security for the Web Library

Security for the web library can be set up initially by means of the Registration Wizard. You can use the following steps to modify the security settings created by the wizard.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To modify security:

1. Select PeopleTools, Security, Permissions & Roles, Permission Lists.
2. Open the desired permission list, such as *ALLPAGES* or *PAPP9000.*
3. Go to the Web Libraries tab.
4. Select the Edit link next to the appropriate web library.
5. Modify the web library permissions as desired by changing the Access Permissions drop-down option. The choices are *Full Access* and *No Access.* Click OK to return to the main Permission List page.
6. Click Save to save your changes to the permission list.

Related Links

"Setting Web Library Permissions" (PeopleTools 8.54: Security Administration)

"Understanding the Registration Wizard" (PeopleTools 8.54: Application Designer Developer's Guide)

"iScript Security" (PeopleTools 8.54: PeopleCode API Reference)

Developing Java-Based Pagelets

This section provides overviews of how to develop pagelets in Java and Java pagelet development considerations and discusses how to:

- Write business logic in Java.
- Use Java to write business logic for a PeopleSoft or non-PeopleSoft system.
- Invoke PeopleSoft components from Java.

Understanding Java Development

Java is an extremely popular programming language today, especially for internet applications. PeopleTools provides support in several key areas to enable developers to create Java-based applications. Though the PeopleSoft development environment allows you to write PeopleSoft business logic in Java, the focus of the following sections is to provide information that is relevant to the developer who is integrating applications with the PeopleSoft Interaction Hub. An example is development that is being done to integrate your non-PeopleSoft application with the PeopleSoft Interaction Hub. Another example is a scenario in which you want to pull together data from your non-PeopleSoft system along with data from your PeopleSoft system for integration with the PeopleSoft Interaction Hub.

Two primary areas are available in which you can program in Java within the PeopleTools environment:

1. Writing business logic in Java.

   You can use Java used to write business logic for PeopleSoft or non-PeopleSoft applications that are to be rendered in the PeopleSoft Interaction Hub. Java can be called from PeopleCode and the Java program can then reference the PeopleSoft runtime environment as needed.

   In addition, Pagelet Wizard facilitates Java development in that it enables the Java programmer to concentrate on business logic, while Pagelet Wizard handles all the development aspects of rendering the application in the portal framework without any coding necessary.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. Invoking PeopleSoft components from Java.

   All PeopleSoft components can be invoked from Java programs through the component interface technology. This is useful if you want to create Java servlets, applets, JSP programs, or Enterprise JavaBeans that invoke PeopleSoft business logic.

Another consideration when developing applications with Java is that any application developed by means of Java that is URL-addressable and produces HTML can be easily integrated into the PeopleSoft Interaction Hub through the PeopleSoft Interaction Hub registry functionality. This enables you to distribute your development and runtime environments making for better scalability for both.

For more information about leveraging the PeopleSoft Interaction Hub registry functionality to integrate your Java application with the PeopleSoft Interaction Hub, see the following documentation.

See Administering Pagelets in the Portal Registry.

These areas of Java programming within the PeopleTools development environment are discussed further in the following sections.

Describing Java Pagelet Development Considerations

This section discusses the following Java development elements:

- Business rules.
- User interface.
- User personalization.
- Navigation.
- Navigation on extended pages.
- Portal integration.

Business Rules

Java can be used to apply business logic to data to create a result set. Pagelet Wizard can then be used to invoke the Java program. Parameters that are required by the Java program can be defined, managed, and passed to the Java program through Pagelet Wizard. These parameters may be derived from user personalization parameters, as well as administrator-defined parameters and system variables.

For example, the Java program may need the user's ID, the user's language, and the user's personalization selection regarding whether to include historical transaction information or just current information. These parameters can be defined in Pagelet Wizard in relation to a Java class. Pagelet Wizard can access the PeopleSoft runtime environment to determine the user's ID as well as other profile information, such as the user's language.

See Understanding System Variables Supported as Data Source Parameters.

Pagelet Wizard will also manage the user interface and storage of user personalization data. This topic is discussed in detail in the subsequent User Personalization section. When the Java program is invoked, Pagelet Wizard will handle passing all required parameters to the Java program.

In addition, Java classes delivered with PeopleTools enable you to call PeopleCode from your Java program and access contextual information from the runtime system. If needed, business data can be retrieved, as necessary, through the numerous PeopleCode functions that are accessible from Java.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The result set created by the Java program may be in an HTML format or in an XML format. Pagelet Wizard will accept the output from the Java program and will manage the presentation of the information within the portal. This topic is discussed in the following User Interface section.

User Interface

Pagelet Wizard will manage all aspects of a pagelet's user interface. No user-interface-related programming for the pagelet is required in the Java program. Pagelet Wizard can accept an HTML- or XML-based result set from a Java program. The HTML from the Java program can be rendered directly. Alternatively, XML can be transformed with XSL to create an appropriate pagelet user interface.

In either case, Pagelet Wizard will manage the XSL and invoke the transformation at runtime. Extensible Stylesheet Language templates are provided with Pagelet Wizard for general use scenarios. Extensible Stylesheet Language development may be required, depending on the specific user interface required for a pagelet.

See Defining Pagelet Wizard XSL Prototypes.

User Personalization

Pagelet Wizard supports the persistence of user personalization selections for a pagelet application and handles all aspects of the user interface that enable users to enter their preferences and selections. Personalization parameters are defined within Pagelet Wizard administration pages. No programming is required, Java or otherwise.

User personalization selections can occur at two levels. From the homepage, users can select which pagelet applications they want to include on their homepage. After they've selected a pagelet application and have decided where they want the pagelets to appear on their homepage, they can personalize a particular pagelet application.

Examples of user personalizations at the second level can include scenarios in which a user selects transactions from a particular region or chooses to include transactions using a particular date as opposed to same-day transactions.

The user interface for all of these activities, as well as the storage of the personalization data for persistence, is managed by Pagelet Wizard without any additional programming, Java or otherwise.

Pagelet Wizard supports functionality that enables an end user to select a pagelet application to appear on his or her homepage. When a pagelet application is registered with Pagelet Wizard, a folder, which represents a portal application category, can be selected. When users personalize their homepage by selecting specific pagelet applications that they want to appear on their homepage, the selection is presented within the category (folder) that is defined when they register the pagelet application through Pagelet Wizard.

See Step 6: Specifying Pagelet Publishing Options.

Pagelet Wizard also supports functionality that enables an end user to select values that allow a specific pagelet application to be personalized. The functionality may dictate that some input parameters that are required by a Java program be entered by an end user, while other parameters can be set by an administrator. Pagelet Wizard manages parameter passing and the user interface. Pagelet Wizard inspects the Java source code for input parameters. Pagelet Wizard administrators can then select which parameters can be accessed by an end user for personalization. The user interface to enables an end user to personalize a pagelet application and update the appropriate Java input parameters is automatically generated and managed by Pagelet Wizard.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

When creating links in your pagelet application that enable a user to navigate to a related page for details and so forth, the link may need to be proxied by the portal to keep the user within the portal. For example, if a pagelet link is not proxied, then the pagelet can take a user to a site that is completely independent from the portal. In this scenario, the navigational links on the portal header, the homepage menu pagelet, and the left-hand menu pagelet will not be available to the user.

If the link is proxied by the portal, then the link can access and display content from a site that is independent of the portal. However, the content will be rendered within the portal. For example, the portal's header and left-hand menu will be available.

With the PeopleSoft Interaction Hub, links that are included in a pagelet will be altered automatically to support proxying by the portal. No programming is required, Java or otherwise.

Note: Content that is referenced by the link must be HTML-based to be rendered appropriately in the portal.

Navigation on Extended Pages

Clicking a link on an extended page displays content within the target frame. The PeopleSoft Interaction Hub, header, and left-hand menu area will remain. Therefore, the new content will be rendered in the context of the PeopleSoft Interaction Hub.

Note: Content that is referenced by the link must be HTML-based to be rendered appropriately in the portal.

The Return to Home link, which returns the user to the portal homepage, may be required from an extended page. The GenerateHomepagePortalURL function creates a URL string that represents an absolute reference to the portal homepage. Because you can access the PeopleSoft runtime environment from a Java program and you can access PeopleCode built-in functions, you can invoke the GenerateHomepagePortalURL function from your Java program.

See Developing Java-Based Pagelets.

See "GenerateHomepagePortalURL" (PeopleTools 8.54: PeopleCode Language Reference)

Portal Integration

The pagelet application needs to be integrated with the portal registry so that the pagelet application is managed, secured, and presented through the PeopleTools portal. Registration makes the portal aware of a pagelet. Information included in registration determines the name and security for your pagelet application. In addition, registration data determines the category in which the pagelet application resides. Users view pagelet applications within these categories when selecting items that they want to see on their homepage. No programming is required, Java or otherwise, to complete portal registration.

Writing Business Logic in Java

Many reasons exist why you would want to write business logic for your non-PeopleSoft applications (and even your PeopleSoft applications) in Java. Perhaps you have licensed a third-party set of Java classes to do some very specific processing (tax calculation, for example). Or perhaps you have developed some internal Enterprise JavaBeans for your specific business processes. Or perhaps you simply like to code in Java. Whatever the reason, you can easily integrate Java code with your applications through the Java PeopleCode functions.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Invoking Java from PeopleCode

Three primary Java PeopleCode functions are used to invoke Java from PeopleCode:

- GetJavaClass
- CreateJavaObject
- CreateJavaArray

Java PeopleCode Function	***Usage***	***Example***

GetJavaClass	Finds a Java class that you can manipulate in PeopleCode. This is used for those classes that have static members for which instantiating an object of the class isn't appropriate. You can call only static methods, that is, class methods, with the object created with this function.	In Java, you access such static members of a class by using the class name:
		`result = java.class.name. SomeStaticMethod();`
		In PeopleCode, you use the following code:
		`&Result = GetJavaClass( "java.class.name"). SomeStaticMethod();`
		The following example is simple PeopleCode that uses GetJavaClass to get a system class:
		`&Sys = GetJavaClass("java. lang.System"); &Sys.setProperty("java. security.policy", "C:\java\policy"); WinMessage("The security property is: " | &Sys.getProperty("java. security.policy")); &Props = &Sys. getProperties(); &Props.put("java.security. policy", "C:\java\policy"); &Sys.setProperties(&Props); WinMessage("The security property is: " | &Sys.getProperty("java. security.policy"));`

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Java PeopleCode Function	*Usage*	*Example*

CreateJavaObject	Creates a Java object that can be manipulated in your PeopleCode. You can use the CreateJavaObject function to create a Java array. If ClassName is the name of an array class (ending with [ ]), ConstructorParams are used to initialize the array.	In Java, use the following code to initialize an array:
		`intArray = new int[]{1, 2, 3, 5, 8, 13};`
		In PeopleCode, you use the following code to initialize a Java array:
		`&IntArray = CreateJavaObject("int[]", 1, 2, 3, 5, 8, 13);`
		If you want to initialize a Java array without knowing the number of parameters until runtime, use the CreateJavaArray function.
		The following example is a simple PeopleCode program that creates a Java object from a sample program that generates a random password:
		`/* Example to return Random Passwords from a Java class */`
		`Local JavaObject &oGpw;`
		`/* Create an instance of the object */`
		`&oGpw = CreateJavaObject ("com.PeopleSoft.Random.Gpw_Demo");`
		`&Q = "1";`
		`/* Call the method within the class */`
		`&NEW_VALUE = &oGpw. getNewPassword(&Q, PSRNDMPSWD.LENGTH);`
		`/* This is just returning one value for now */`
		`PSRNDMPSWD.PSWD = &NEW_VALUE;`

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Java PeopleCode Function	*Usage*	*Example*

CreateJavaArray	Enables you to create a Java array without knowing the number or value of the elements.
	
	When you create an array in Java, you already know the number of elements in the array. If you don't know the number of elements in the array, but you want to use a Java array, use the CreateJavaArray function in PeopleCode. This will create a Java object that is a Java array, and you can pass in the number of elements that are to be in the array.
		The following PeopleCode example passes a PeopleCode array of strings (&Parms) to a Java method *xyz* of class *Abc.* This example assumes that when you are writing the code, you don't know how many parameters you will have.
		
		`Local JavaObject &Abc, &RefArray;`
		`Local array of String &Parms;`
		`&Parms = CreateArray();`
		`/* Populate array how ever you want to populate it */`
		`&Abc = GetJavaObject("com.peoplesoft.def.Abc");`
		`/* Create the java array object. */`
		`&JavaParms = CreateJavaArray("java.lang.String[]", &Parms.Len);`
		`/* Populate the java array from the PeopleCode array. */`
		`&RefArray = GetJavaClass("java.lang.reflect.Array");`
		`For &I = 1 to &Parms.Len`
		`&RefArray.set(&JavaParms, &I - 1, &Parms[&I]);`
		`End-For;`
		`/* Call the method. */`
		`&Abc.xyz(&JavaParms);`

Accessing the PeopleSoft Runtime System from Java

After a Java class has been invoked, the class can access the PeopleSoft runtime system. Java classes delivered with PeopleTools enable you to call PeopleCode from your Java program and access contextual information from the runtime system, such as the current user's role and language preference. By importing the PeopleTools-delivered Java classes in your Java program, you can have access to PeopleCode objects and methods. Hundreds of PeopleSoft system variables, constants, and built-in functions are available for use with this approach. Discussions of the various methods follow.

Accessing the runtime system works only from a Java program that was initially called from PeopleCode. You must call PeopleCode facilities only from the same thread that was used for the call into Java. You cannot call any PeopleCode facility that would cause the server to return to the browser for an end-user action because the state of the Java computation cannot be saved and restored when the action is complete.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Java Class	***Usage***
SysVar	Use to refer to PeopleSoft system variables, such as %Language or %Oprid. For example, %Session, becomes SysVar.Session(). See "System Variables" (PeopleTools 8.54: PeopleCode Developer's Guide).
SysCon	Use to refer to system constants, such as %SQLStatus_OK or %FilePath_Absolute. For example, %CharType_Matched becomes SysCon.CharType_Matched. See "Constants" (PeopleTools 8.54: PeopleCode Developer's Guide).

Name	Enables you to use the PeopleSoft-reserved item references. This enables you to reference pages, components, records, fields, and so forth.
	
	For example, you can refer to a record field using the following PeopleCode:
	
	`recname.fieldname`
	
	With the Name class, you can use a similar construct:
	
	`new PeopleSoft.PeopleCode.Name("RECNAME", "FIELDNAME");`
	
	As another example, you can refer to a PeopleSoft page using the following PeopleCode:
	
	`PAGE.pagename`
	
	In Java, it would be:
	
	`new PeopleSoft.PeopleCode.Name("PAGE", "PAGENAME");`
	
	See "Definition Name References" (PeopleTools 8.54: PeopleCode Developer's Guide).

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Java Class	***Usage***
Func	Use to refer to built-in PeopleCode functions, such as CreateRowset or GetFile. For example, SetLanguage(LANG_CD) becomes Func. SetLanguage(LANG_CD). The existing PeopleCode classes (Array, Rowset, and so forth) have properties and methods that you can access from Java. PeopleCode classes have the same name, so Record becomes Record, SQL becomes SQL, and so forth. Methods are accessed by the method name. The name of a property is prepended with either *get* or *set,* depending on whether you're reading or writing to the property. For example, to get the IsChanged property would be getIsChanged. To set the value for a field would be &MyField.setValue.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Java Class	***Usage***

Func (continued)	Here is an example of a Java program that uses PeopleCode objects to access the database:
	
	`/* This code is used to test the Java/ PeopleCode interface. */`
	`import PeopleSoft.PeopleCode.*;`
	`public class Test {`
	`/*`
	`* Add up and return the length of all the`
	`* item labels on the UTILITIES menu,`
	`* found two different ways.`
	`*/`
	`public static int Test() {`
	`/* Get a Rowset to hold all the menu item records. */`
	`Rowset rs = Func.CreateRowset(new Name( "RECORD", "PSMENUITEM"), new Object[]{});`
	`String menuName = "UTILITIES";`
	`int nRecs = rs.Fill(new Object[]{"WHERE FILL.MENUNAME = :1", menuName});`
	`int i;`
	`int nFillChars = 0;`
	`for (i = 1; i <= rs.getActiveRowCount(); i ++) {`
	`String itemLabel = (String)rs.GetRow(i) .GetRecord(new Name("RECORD", "PSMENUITEM")) .GetField(new Name("FIELD", "ITEMLABEL")) .getValue();`
	`nFillChars += itemLabel.length();`
	`}`
	`/* Do this a different way - use the SQL object to read each menu item record. */`
	`int nSQLChars = 0;`
	`Record menuRec = Func.CreateRecord(new Name("RECORD", "PSMENUITEM"));`
	`SQL menuSQL = Func.CreateSQL("%SelectAll( :1) WHERE MENUNAME = :2", new Object[]{menuRec, menuName});`
	`while (menuSQL.Fetch(new Object[] {menuRec})) {`
	`String itemLabel = (String)menuRec .GetField(new Name("FIELD", "ITEMLABEL")) .getValue();`
	`nSQLChars += itemLabel.length();`
	`}`
	`return nFillChars + 100000 * nSQLChars;`
	`}`
	`}`

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Java Class	*Usage*

Func (continued)	This can be run from PeopleCode in the following way:
	
	`Local JavaObject &Test;`
	`Local number &chars;`
	`&Test = GetJavaClass("Test");`
	`&chars = &Test.Test();`
	`&Test = Null;`
	`WinMessage("The character counts found are: " | &chars, 0);`
	
	See "PeopleCode Classes Alphabetical Reference" (PeopleTools 8.54: PeopleCode API Reference)"PeopleCode Syntax Quick Reference" (PeopleTools 8.54: PeopleCode API Reference).

Using Java to Write Business Logic for a PeopleSoft or Non-PeopleSoft System

This section presents an example of pagelet application development for which the intention is that all programming be done in Java. Business logic might be for a PeopleSoft system or a non-PeopleSoft system. In this development example, PeopleTools programming is not required. All programming is done with Java.

The PeopleTools infrastructure can be leveraged without any PeopleTools programming through the use of PeopleSoft Interaction Hub features that support administration, integration, and implementation for the portal. The ability to leverage the PeopleTools infrastructure can significantly reduce the amount of Java programming that is required to develop a pagelet application for use in the PeopleSoft Interaction Hub.

Some of the information in this section is covered in other Pagelet Wizard documentation. The main difference in this section is that it presents the information as a development scenario as opposed to a feature description. Also, the ability to access the PeopleSoft runtime environment from Java is discussed in this section.

The following development scenario addresses the following elements of portal application development, integration, and implementation:

- Business rules.

  Apply business logic and rules to data to create a result set.

- User interface.

  Create the appropriate presentation format for a set of business data.

- User personalization.

  - User personalization data.

    Allow pagelet applications to be selected by the end user to be placed on his or her homepage. Also, allow a user to personalize a specific pagelet application. These selections should persist.

  - User interface.

    Create the appropriate presentation format for personalization data.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Navigation for a pagelet.

  Create links on your pagelet to enable a user to navigate to a related page for details and so forth.

- Navigation on extended pages.

  An extended page is a page that a user can get to from a homepage pagelet. Extended pages might show details as well as have links to enable a user to navigate to other related pages.

  - Links.

    Create links from your extended page to enable a user to navigate to a related page for further details and so forth.

  - Return links.

    Create links that enable the user to return to the portal's homepage.

- Portal integration and implementation.

  Integrate the pagelet application with the portal registry so that the pagelet application is managed, secured, and presented through the PeopleTools portal.

Invoking PeopleSoft Components from Java

All PeopleSoft components can be invoked from Java programs through component interface technology. This is useful for developers who want to create Java servlets, applets, JSP programs, or Enterprise JavaBeans that invoke PeopleSoft business logic. This section provides an example of how to invoke a PeopleSoft component from Java. A Business Expense component is used as the example.

Creating the Component Interface

To expose a component to a third party, you must first create a component interface definition. You do this through the PeopleSoft Application Designer. Using drag-and-drop functionality, you can specify

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

the properties and methods of the component that you want to expose. Numerous component interface definitions are delivered with the each PeopleSoft application.

Image: BUSINESS_EXPENSES component interface definition

The BUSINESS_EXPENSES component definition appears in the left frame in the following example. The properties and methods that are exposed through this interface are displayed in the right frame.

The following screenshot displays Bus_Exp component interface definition in the left frame and properties and methods in the right frame.

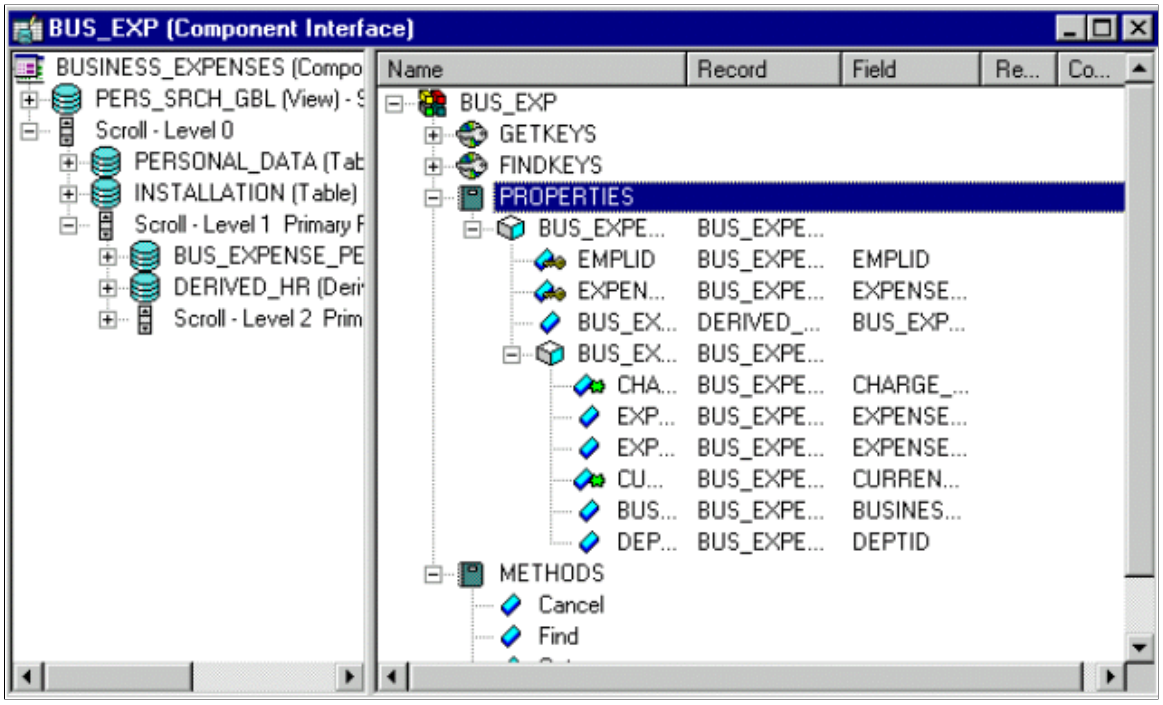
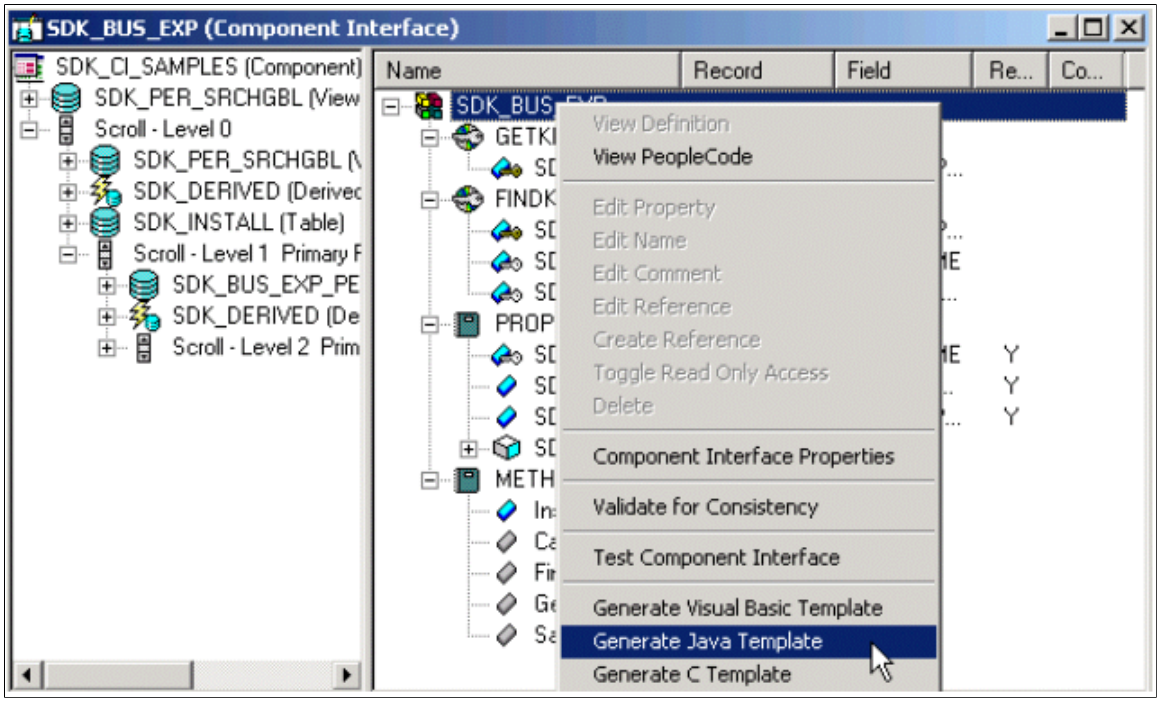


Image: SDK_BUS_EXP component interface

After the component interface definition is saved, you can then generate the Java classes for invoking this interface. You do this using the Generate Java Template command in PeopleSoft Application Designer:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The following screenshot illustrates an example of invoking the SDK_BUS_EXP component interface and generating Java classes.



See "Understanding Component Interface Class" (PeopleTools 8.54: PeopleCode API Reference).

Invoking the Component Interface from Java

To invoke the Business Expense component interface from Java.

1. Connect to the application server.

   To access a component interface, you need to establish a PeopleSoft session. To create a session object, use the Session.Connect () method. The Connect method, which takes five parameters, actually signs in to a PeopleSoft session. The **Connect()** method connects a session object to a PeopleSoft application server. Note that various options are available for using an existing connection and disconnecting and switching connections.

   ```
   import PeopleSoft.ObjectAdapter.*;
   import PeopleSoft.Generated.PeopleSoft.*;
   import PeopleSoft.Generated.CompIntfc.*;
   private ISession oSession;
   private CAdapter oAdapter;
   oAdapter = new CAdapter();
   oSession = new CSession(oAdapter.getSession());
   oSession.Connect(1,"//EHARRIS032000:9000","PTDMO","PTDMO",new byte[0]);
   ```

2. Get an instance of the component interface.

   Use the GetComponent() method with a session object to get an instance of a previously created component interface.

   ```
   busExpense = new CBusExp( oSession.GetComponent( "BUS_EXP" ));
   ```

3. Find an existing record.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

You can query a component interface to find relevant data instances based on primary and alternate search keys.

```
busExpense.setName( searchDialogStrings[ 0 ]);
busExpense.setLastNameSrch( searchDialogStrings[ 1 ]);
busExpense.setEmplid( searchDialogStrings[ 2 ] );
return( busExpense.Find() );
```

4. Get an instance of data.

   GetKeys are the key values required to return a unique instance of existing data. GetKeys can be set by means of simple assignment to the properties of the component interface and then the Get() method can be invoked. This populates the component interface with data based on the key values that you set; this has been referred to here as a data instance.

```
busExpense.setEmplid( getKey );
boolean result = busExpense.Get();
```

5. Migrate through collections of data.

   After getting a data instance, get access to the data in the component interface. PeopleSoft organizes component interface data within collections. Rows of data in a collection are called items.

   The following code creates a connection to the application server, gets the component interface, and fetches the first item in a collection.

```
oAdapter = new CAdapter();
oSession = new CSession(oAdapter.getSession());
oSession.Connect(1,"//EHARRIS032000:9000","PTDMO","PTDMO",new byte[0]);
busExpense = new CBusExp( oSession.GetComponent( "BUS_EXP" ));
busExpense.setEmplid( getKey );
boolean result = busExpense.Get();
busExpenseFirstScrollItemCollection = busExpense.getBusExpensePer();
busExpenseFirstScrollItem = firstScrollCollection.Item
( firstScrollIndex );
return( busExpenseFirstScrollItem.getBusExpenseDtl() );
```

6. Edit and access data in an item.

   Editing and accessing component interface data in Java is rather straightforward. The following Java code accesses the various public members of the class.

```
long j = busExpenseSecondScrollCollection.getCount();
Object [][] data = new Object[ ((int)j + 1) ][ 7 ];
for( int i = 1; i < j + 1 ; i++ )
{
busExpenseSecondScrollItem = busExpenseSecondScrollCollection.Item( i );
data[(i - 1)][0] = busExpenseSecondScrollItem.getBusinessPurpose();
data[(i - 1)][1] = busExpenseSecondScrollItem.getChargeDt();
data[(i - 1)][2] = busExpenseSecondScrollItem.getCurrencyCd();
data[(i - 1)][3] = busExpenseSecondScrollItem.getDeptid();
data[(i - 1)][4] = busExpenseSecondScrollItem.getExpenseAmt();
data[(i - 1)][5] = busExpenseSecondScrollItem.GetPropertyByName
("ExpenseCd");
data[(i - 1)][6] = busExpenseSecondScrollItem.GetPropertyByName
("CurrencyCd");
}return( data );
```

   In the following example, data is accessed by means of the **get***NAME_OF_PRPERTY***()** method of an item or by means of the generic **getPropertyByName()** method. This code illustrates the way in

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

which an entire collection of data can be captured and packaged into an object for transfer to a calling object.

```
busExpenseFirstScrollItem.setEmplid( emplid );
busExpenseFirstScrollItem.setExpensePeriodDt( expensePeriodDt );
return( busExpense.Save() );
```

Just as before, data is edited by means of item objects and the setNameOfPropery() method of those items. Also note that you needed to call the Save() method on the component interface to commit the changes.

7. Insert an item into a collection and delete an item from a collection.

   Collection objects in Java contain the InsertItem() method in which the return value is the item that has just been inserted. After a new item is created, simply edit data in it and then remember to call the Save() method to commit the changes.

```
busExpenseSecondScrollItem = busExpenseSecondScrollCollection.
InsertItem( secondScrollIndex );
```

   Similarly, a DeleteItem() method is available:

```
busExpenseSecondScrollCollection.DeleteItem( secondScrollIndex );
```

8. Disconnect from a session.

   After a session is no longer needed, it should be disconnected from the application server. This is done by calling the Disconnect() method on the session object.

```
oSession.Disconnect();
```

Developing Contextual Embeddable Pagelets

This section discusses how to develop contextually relevant pagelets that you embed in transaction pages.

1. Create an embeddable pagelet using Pagelet Wizard.

   See Understanding Pagelet Wizard Pagelet Types.

2. Create a transaction page definition using Application Designer.

   See "Creating New Page Definitions" (PeopleTools 8.54: Application Designer Developer's Guide).

3. Place an HTML area on the page definition.

   See "Inserting HTML Areas" (PeopleTools 8.54: Application Designer Developer's Guide).

4. Write a PeopleCode function and map the pagelet parameters with any available values from the component buffer.

PeopleCode Sample for Rendering the Context-Based Embeddable Pagelet

This PeopleCode example uses application packages to create an embeddable pagelet.

```
 /* Import the Pagelet Wizard application package to create the embeddable pagelet*⇒
/

      import PTPPB_PAGELET:*;
       import PTPPB_PAGELET:UTILITY:*;
           Component object &Pagelet, &myDataSource;
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
            /* Create the Pagelet Wizard application pakage and assign the pagelet I⇒
D*/
            &PWAPI = create PTPPB_PAGELET:PageletWizard();
            &PageletID = "EMBEDED_PAGELET";

            /* Get the pagelet's pointer by passing the pagelet id*/
            &Pagelet = &PWAPI.getPageletByID(&PageletID, False);
            &myDataSource = &Pagelet.DataSource;

            /* Set the pagelet parameters to default values*/
            &Pagelet.PopulateInputsWithDefaults();

            /* Read the pagelet parameters */
            &DSParamColl = &myDataSource.getParameterCollection();
            &CollectionParamArray = &DSParamColl.getCollectionAsArray();

            /* To override the pagelet parameter default values, */
      /*  read the CollectionParamArray and set the parameter values */
      /*  from the component buffer based on the business requirement */
            If &CollectionParamArray.Len > 0 Then
             For &i = 1 To &CollectionParamArray.Len
               If (&i = 1) Then
                 &DSParameter = &CollectionParamArray [&i];
                 &CollectionParamArray [&i].value = PSOPRDEFN.OPRDEFNDESC;
               End-If;
             End-For;
            End-If;

            /* Get the Embeddable Pagelet HTML */
            &PgltHTML = &Pagelet.Execute();

            /* Associate the Pagelet HTML with the HTML Area */
            PSUSRPRFL_WRK.HTMLAREA = "<div class='PSTEXT'>" | &PgltHTML | "</div>";
```

Administering Pagelets in the Portal Registry

This section discusses how to:

- Register homepage and template pagelets.
- Modify pagelet attributes and security.
- Register URL-based pagelets.
- Register pagelet extensions.

Note: When you use Pagelet Wizard to register pagelets in the portal registry, Pagelet Wizard automates much of the process by interacting with the portal registry. You supply some key information in the pagelet definition, such as pagelet name, title, folder, and so on; Pagelet Wizard passes this information on to the portal registry.
However, the portal registry encapsulates other metadata about URLs that are accessed by way of the portal. Even if you use Pagelet Wizard to define and initially register a pagelet, you may still need to access the portal registry to update an attribute or to register additional entries.

Related Links

Understanding Portal Administration

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Registering Homepage and Template Pagelets

Before you can access a pagelet through the portal or associate a template pagelet with a target or WorkCenter page, you must register the pagelet in the portal registry. You can register pagelets by using the Component Registration Wizard, or by using the following procedure:

To register a new homepage pagelet or template pagelet:

1. Select PeopleTools, Portal, Structure and Content. Then:
   - For template pagelets, click these links in this order: Portal Objects, Template Pagelets.

     Template pagelets must be registered in this folder.
   - For homepage pagelets, click these links in this order: Portal Objects, Pagelets. Then select any subfolder; for example, you can register a homepage pagelet in the Portal Objects, Pagelets, Organizers folder.
2. Click the Add Content Reference link.
   - For template pagelets, select *Target* as the usage type.
   - For homepage pagelets, select *Pagelet* as the usage type.
3. Select *Always use Local* as the node name.
4. Select the URL type based on the location of the pagelet.
   - Select *PeopleSoft Component* for component-based template pagelets.
   - Select *PeopleSoft Script* for iScript-based template pagelets.
   - Select *PeopleSoft Generic URL* for Pagelet Wizard-based pagelets.
5. Set pagelet attributes as necessary.
6. Set the content reference attributes as necessary.
7. Set additional parameters, if needed.
8. Save the content reference.

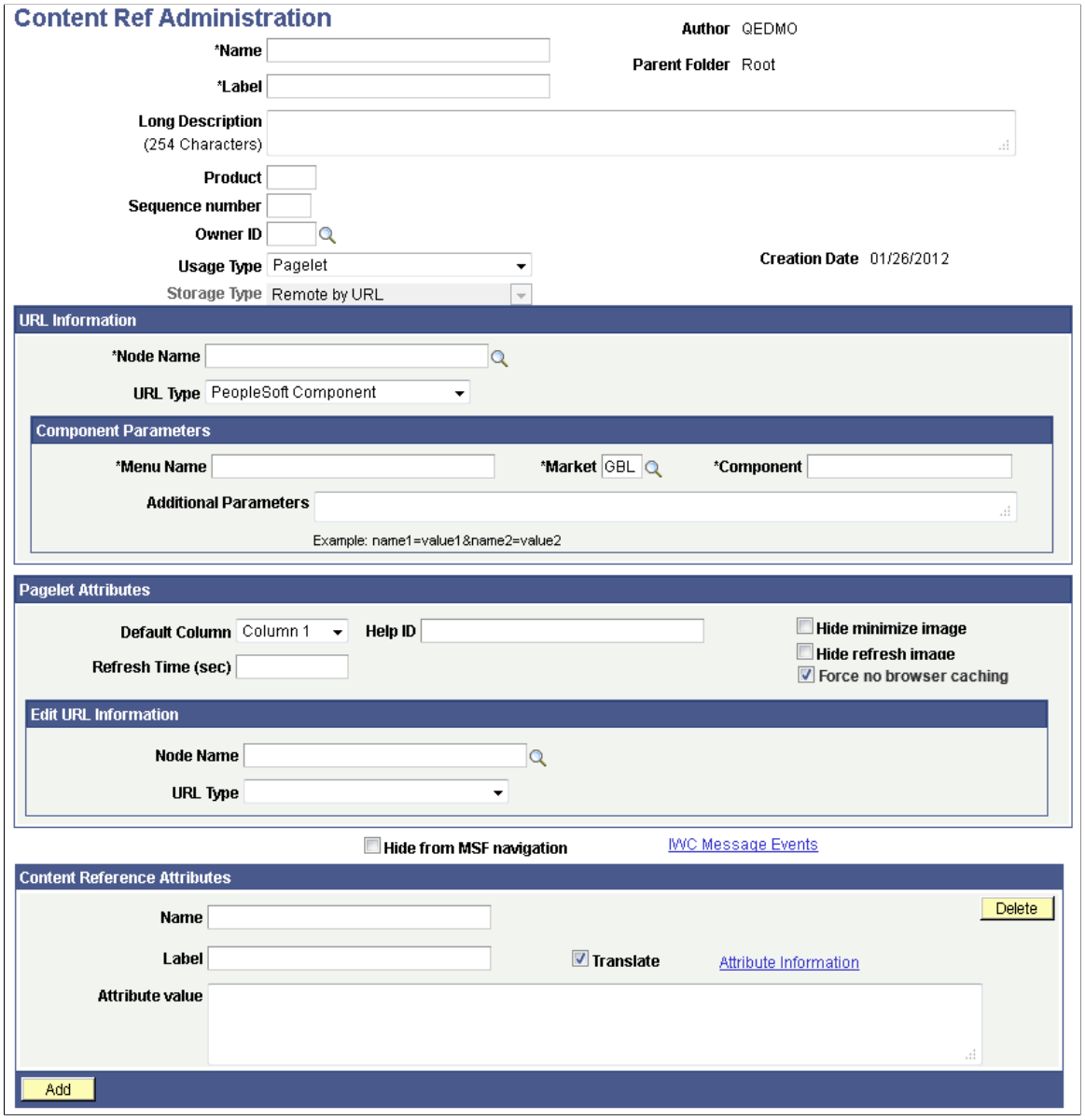
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Modifying Pagelet Attributes and Security

Access the Content Ref Administration page. (Select PeopleTools, Portal, Structure and Content. Click the Edit link for a content reference.)

Image: Content Ref Administration page for a pagelet

The following screenshot illustrates about modifying Pagelet attributes and security on the Content Ref Administration page.



The portal registry stores every content reference that is available through the portal. The label and description defined here affect the appearance of the label and description in the navigation menu. Other attributes on this page affect the URL that will be used to reference the content associated with a definition.

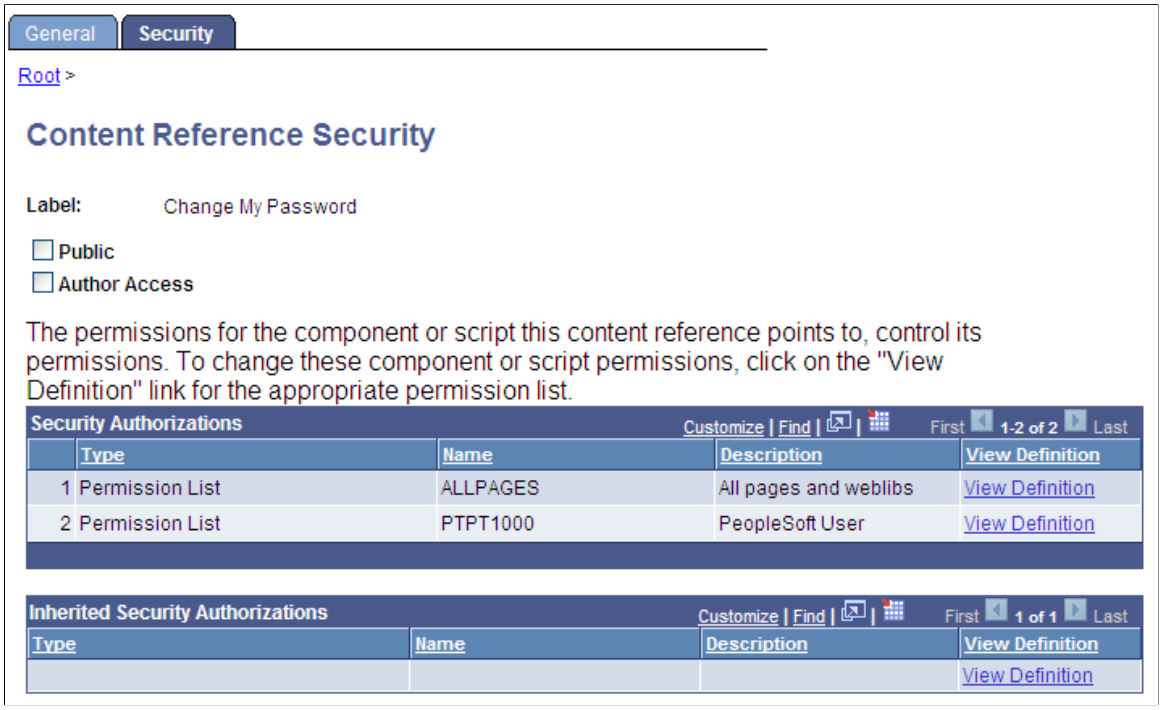
See Understanding Portal Administration.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Content Reference Security page. (Select PeopleTools, Portal, Structure and Content. Click the Edit link for a content reference. Select the Security tab.)

Image: Content Reference Security page for a pagelet

The following screenshot illustrates how a Content Reference Security page looks like.



You can grant and review the permission lists and roles that provide access to homepage tabs and template pagelets on the Content Reference Security page. The page allows for any combination of permission lists and roles. You can also make the object publicly available or available to the author only, which can be useful for testing purposes. Permission lists and roles can be used to grant access to the content item associated with this definition.

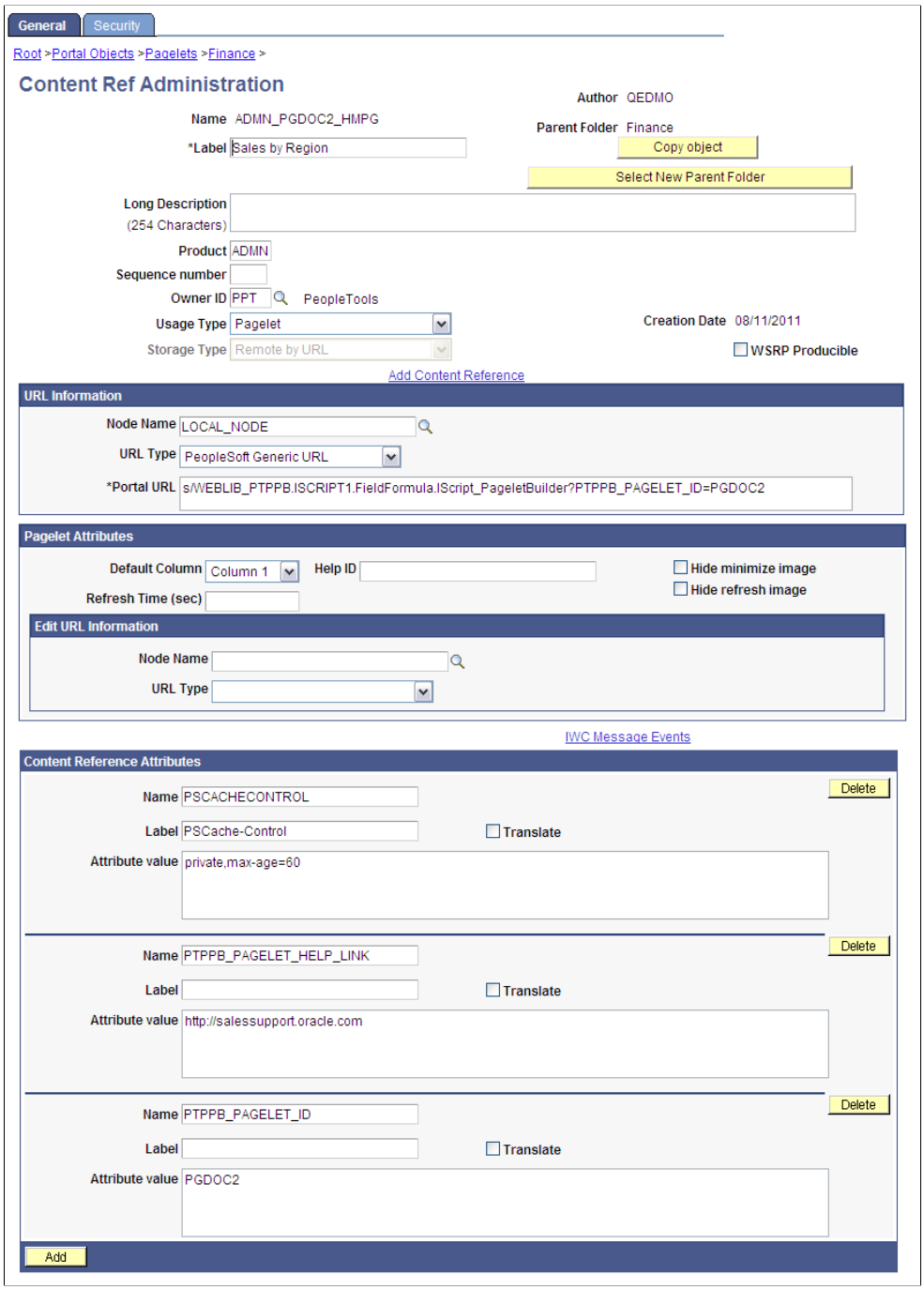
See Understanding Portal Administration.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Registering URL-Based Pagelets

Image: Content Ref Administration page for a URL-based pagelet

The following screenshot illustrates registering URL-Based Pagelets on the Content Reference Administration Page.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

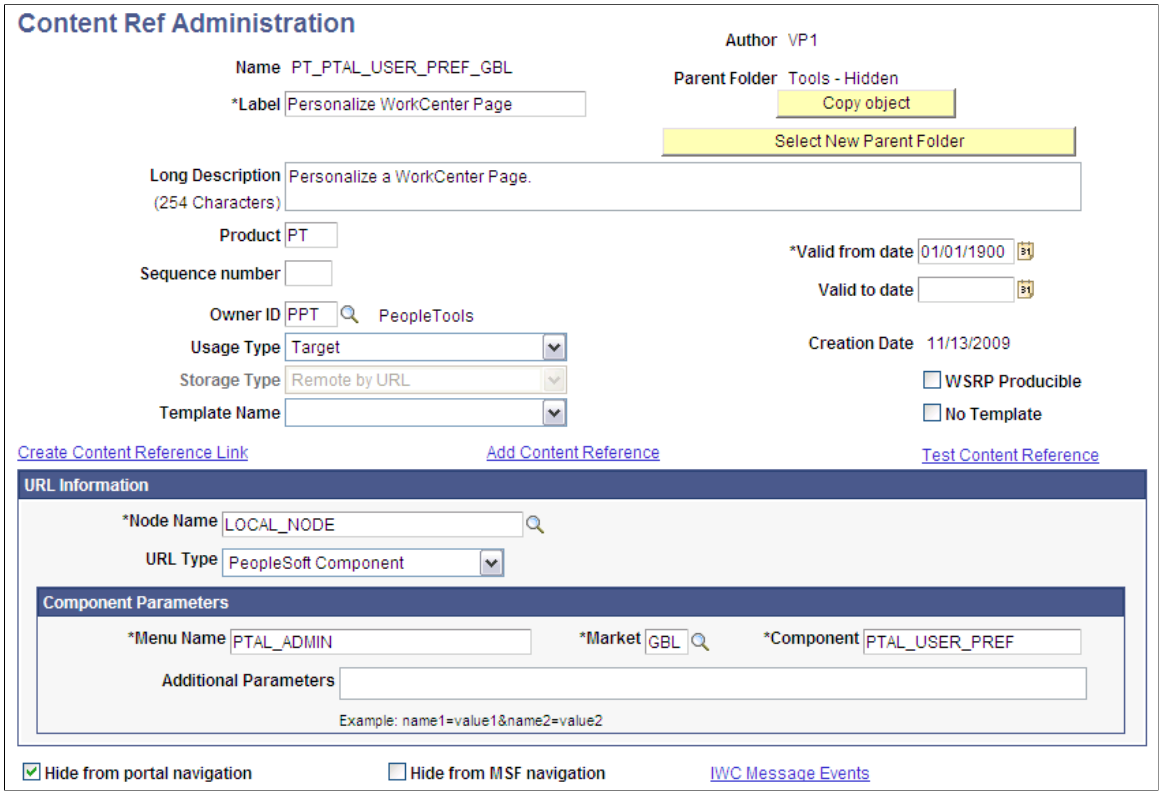
Registering Pagelet Extensions

This is an example of the registration a pagelet personalization page.

Image: Content Ref Administration page for a pagelet personalization page

In this example, the personalization component was built with PeopleTools. However, a non-PeopleSoft URL-based personalization page could have been referenced instead.

The following screenshot is an example of registering pagelet extensions for a Personalization page.



A personalization page must be identified in the registry as being related to a specific pagelet application.

Note that the personalization component (PTAL_USER_PREFS) is referenced in the Component Parameters group box. This is where the personalization page that is related to the pagelet application is identified.

Using Attributes to Enhance Pagelets

This section provides an overview of pagelet and content reference attributes and discusses how to:

- Manage pagelet attributes.
- Configure pagelet Help links.
- Configure pagelet time-outs.
- Configure automatic pagelet refresh attributes.
- Create banner pagelets.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Add images to pagelet headers.

Understanding Pagelet and Content Reference Attributes

Pagelet and content reference attributes provide a framework to specify and store free-form information about a content reference or pagelet. Attributes also enable you to manage pagelet behavior and appearance.

When you click a content reference, the portal retrieves the URL for the pagelet or content reference, it reads and processes the attribute and its value, and renders the page or pagelet incorporating the attribute.

You add attributes by entering Name-Value pairs. For example, an attribute name is PORTAL_BANNER_PGLT. You use this attribute to define a pagelet as a banner pagelet, which is the only pagelet that can span the enter width of the transaction page that appears in the browser in your PeopleSoft applications. The possible attribute values for this attribute are 1, 2, and 3. Each number represents different options regarding the appearance of the banner pagelet.

Managing Pagelet Attributes

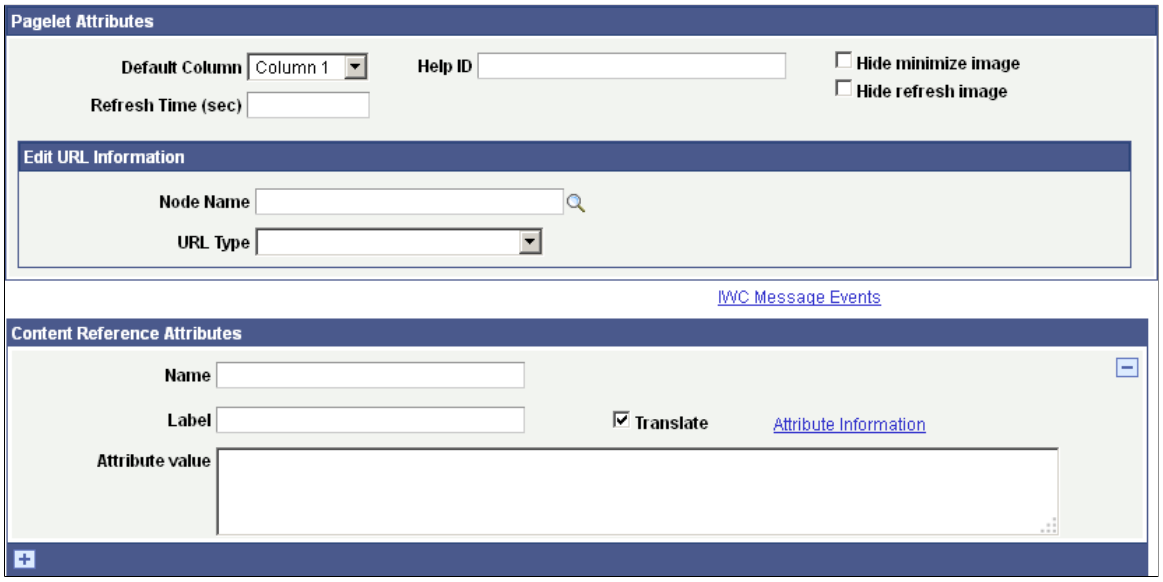
Access the Content Ref Administration page. (Select PeopleTools, Portal, Structure and Content. Click the Edit link for the appropriate content reference.)

1. Select *Pagelet* in the Usage Type drop-down list box.
2. Use the Pagelet Attributes group box to select the default column for a pagelet, associate a help topic with the pagelet, set refresh time, and hide certain buttons from users.
3. Use the Content Reference Attributes group box to create banner pagelets and add images to pagelet headers

Image: Pagelet Attributes and Content Reference Attributes group boxes

Use the Pagelet Attributes and Content Reference Attributes group boxes to enhance pagelets as shown in this example:

The following screenshot illustrates an example to enhance pagelets by using pagelet and content reference attributes.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pagelet Attributes

Default Column

Select the column in which the pagelet appears.

Help ID

Enter the help ID for the pagelet. When users click the Help link, the PeopleSoft system opens a new window displaying online help documentation for this item.

Note: You must also configure a help URL on the Web Profile Configuration - General page.

Refresh Time (sec)

Enter the number of seconds after which the system auto-refreshes the pagelet. To activate the refresh timer, enter a value of *1* or greater.

The timer refreshes the pagelet as soon as the user accesses the homepage and then automatically refreshes the pagelet every *n* number of seconds.

Note: Homepages and dashboards use automatic refresh.

Hide minimize image

Select to hide the minimize button that normally appears in the pagelet headerso that users are prevented from minimizing the pagelet.

Hide refresh image

Select to hide the refresh button to prevent users from refreshing the pagelet.

If you implemented pagelet caching for this pagelet, a refresh button automatically appears in the pagelet header.

Force Pagelet Refresh

Some components can render themselves differently based on the column width. When a homepage has a two-column layout and the user moves a component-based pagelet to a different sized column by using the drag-and-drop method, this pagelet needs to be refreshed so that the content displays correctly.

The drag-and-drop iScript that persists the pagelet changes forces a homepage refresh only when all of the following conditions are met:

- A pagelet changes columns.
- The homepage layout is two-column.
- The pagelet is component-based.
- You select the Force page refresh property for the pagelet.

Note: When `document.write` or `document.writeln` commands are detected inside a pagelet, the whole homepage will be refreshed instead of just that pagelet. This is required to properly assemble that pagelet's content.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

You can use a combination of a web profile setting and the PORTAL_NOBROWSERCACHE content reference attribute to determine whether an individual pagelet can be cached. See Configuring Caching and Implementing Homepage Caching.

Separately, you can also administer server-based caching. See Implementing Pagelet Caching.

Edit URL Information

Use the Node Name and URL Type fields to specify a page to be used for personalizing this pagelet and to make the personalization button appear in the pagelet header.

Node Name Select the node name for the edit URL of the pagelet. This node name can be different from the pagelet node name.

URL Type Select the type of edit URL. The specific URL entry fields appear based on the setting of this field. Options are:

Non-PeopleSoft URL: The Edit URL field appears. Enter the URL of the personalization page to use.

PeopleSoft Component: The Component Parameters group box appears, containing the same fields as the Component Parameters group box for the content reference. Use these fields to identify the personalization page to use.

PeopleSoft Script: The iScript Parameters group box appears, containing the same fields as the iScript Parameters group box for the content reference. Use these fields to identify the personalization page to use.

Content Reference Attributes

You can use the content reference attributes of a pagelet to change its character and appearance and to improve its usability.

Use content reference attributes to:

- Configure pagelet help links.
- Configure pagelet time-out settings.
- Configure automatic pagelet refresh.
- Create banner pagelets.
- Add icons to pagelet headers.

Related Links

Administering Content References
Configuring Pagelet Help Links
Configuring Pagelet Time-out Settings
Configuring Automatic Pagelet Refresh
Creating Banner Pagelets

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Adding Images to Pagelet Headers

Configuring Pagelet Help Links

To specify pagelet help:

1. Select PeopleTools, Portal, Structure and Content.
2. Click the Portal Objects link.
3. Click the Pagelets link.
4. Navigate to the Content Ref Administration page for the pagelet.

   You define the pagelet help in the Content Reference Attributes region of the page.
5. Enter *PTPPB_PAGELET_HELP_LINK* in the Name field.
6. Leave the Label field blank.
7. Deselect the Translate check box.

   Note: You *must* deselect the Translate check box or unexpected results might occur.
8. In the Attribute Value field, enter the URL of the location of your help document, for example *http://www.mycompany.com/help_doc1.html*
9. Save the content reference.

Note: The pagelet Help URL takes precedence over the Help ID when both attributes are set for a pagelet.

Note: The Help button appears only when the pagelet is on the homepage inside the PeopleSoft Portal; it does not appear when the pagelet is displayed as a WSRP portlet.

Related Links

Administering Content References

Configuring Pagelet Time-out Settings

To specify a pagelet time-out:

1. Select PeopleTools, Portal, Structure and Content.
2. Click the Portal Objects link.
3. Click the Pagelets link.
4. Navigate to the Content Ref Administration page for the pagelet.

   You define the pagelet help in the Content Reference Attributes region of the page.
5. Enter *PSTIMEOUT* in the Name field.
6. Enter *PSTimeout* in the Label field.
7. In the Attribute Value field, enter the number of seconds before the pagelet is considered unavailable. This can be any positive integer.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: The portal ignores this attribute if you enter *0* or a negative value, or if the content reference isn't a pagelet.

8. Save the content reference.

When your specified time-out expires, the portal stops attempting to load the pagelet and generates an error message.

Related Links

Administering Content References
Configuring Appearance and Character

Configuring Automatic Pagelet Refresh

To specify a pagelet refresh time in seconds:

1. Select PeopleTools, Portal, Structure and Content.
2. Click the Portal Objects link.
3. Click the Pagelets link.
4. Navigate to the Content Ref Administration page for the pagelet.

   You define the pagelet help in the Content Reference Attributes region of the page.
5. Enter the number of seconds for automatic refresh in the Refresh Time (sec) field.
6. (Optional) Select the Hide refresh image check box to prevent manual pagelet refresh by the user.
7. Save the content reference.

Note: Pagelets with document.write or document.writeln must not have the refresh timer set (to automatically refresh every *n* seconds) because the whole homepage will also be refreshed every *n* seconds.

Creating Banner Pagelets

To create banner pagelets:

1. Create and register your pagelet.

   See Administering Pagelets in the Portal Registry.

   See Understanding Pagelet Development.
2. Access the Structure and Content page (Select PeopleTools, Portal, Structure and Content).
3. Select Portal Objects, then Pagelets in the portal registry folder structure and click the Add Content Reference link.
4. If you have already registered the pagelet, navigate to the Pagelets folder and click the Edit link that is associated with the pagelet.
5. In the Content Reference Attributes group box, enter *PORTAL_BANNER_PGLT* in the Name field.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

6. In the Attribute value field, enter one of these values:

   1 - The banner pagelet will display a pagelet header and a border around the pagelet content like a standard pagelet.

   2 - The banner pagelet will display a border around the pagelet content, but *no* pagelet header.

   3 - The banner pagelet will display *neither* a pagelet header *nor* a border around the pagelet content.

7. Add the pagelet to the homepage.

   See Selecting the Pagelets That Appear on Homepages.

8. Click the Save button.

Adding Images to Pagelet Headers

To add images to pagelet headers:

1. In PeopleSoft Application Designer, create an image definition and store the image in the image catalog.

2. In your browser, access the Structure and Content page (Select PeopleTools, Portal, Structure and Content.)

3. Navigate through the Portal Objects folder structure to the Pagelets folder until you find the pagelet content reference. Then, click the Edit link.

4. In the Content Reference Attributes group box, enter *PT_PORTAL_PGLT_HDR_IMG* in the Name field.

5. In the Attribute value field, enter the name of the image definition that you want to appear in the pagelet header.

   Important! The image definition must exist in the image catalog of the PeopleTools portal database. Use PeopleSoft Application Designer to manage the image definitions that comprise the image catalog.

6. Click the Save button.

Related Links

"Understanding Image Definitions" (PeopleTools 8.54: Application Designer Developer's Guide)

Handling Special Situations

This section discusses how to:

- Determine pagelet running location.
- Use refresh tags in pagelets.

Determining Pagelet Running Location

You can determine whether your code is running within the portal environment (that is being invoked by means of an https request coming from a portal servlet) as opposed to running in PIA outside of the

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

portal environment. The PeopleCode **%RunningInPortal** system variable returns a Boolean value that identifies the environment. This variable works in both iframe templates and HTML templates.

Using Refresh Tags in Pagelets

The order of precedence of refresh tags in pagelets is:

1. Template
2. Target content
3. Pagelet

Among pagelets, the first pagelet in the HTML to include a metarefresh tag is included. Subsequently found refresh tags are not included in the HTML.

Related Links

Understanding Portal Caching

Applying Changes

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 15

Configuring OBIEE Report Pagelets

Understanding OBIEE Integration Within the PeopleSoft Framework

The OBIEE analytics servlet enables application developers to integrate with OBIEE analytics and display OBIEE reports as embedded pagelets on application pages, portal homepages, and as template pagelets, which are part of the Context Manager framework. This functionality defines the OBIEE reports as iScript-based pagelets in PeopleSoft applications. The iScript essentially acts as a client for the report request and sends Service Oriented Architecture Protocol (SOAP) requests to the OBIEE server.

PeopleSoft applications store the credentials of an OBIEE administrator—a user with adequate privileges—in the PeopleSoft database and use these credentials to sign in to the OBIEE server, impersonate a real OBIEE user, and establish a session. The OBIEE server returns a session ID for this user session, which is sent as a query parameter with each SOAP/HTTP request being made to the OBIEE server. The OBIEE server responds to all valid requests. If the session ID is expired or is invalid, a sign-inscreen appears to the user. The OBIEE server provides a session ID and then appends a session ID in each HTTP request that is part of the HTML markup provided by the OBIEE server. If the user ID value in the PeopleSoft application is not found on the OBIEE server, an authorization error message appears to the user in the PeopleSoft application. Similarly, if the user exists in OBIEE but does not have authorization to the requested report, an authorization error message appears to the user.

If authentication passes, then the OBIEE server generates the basic HTML markup for the PeopleSoft application to display. The application uses this basic HTML to send HTTP requests to the OBIEE server from the browser. Before reaching the OBIEE server, the HTTP requests pass through the PeopleSoft bridge servlet. The bridge servlet:

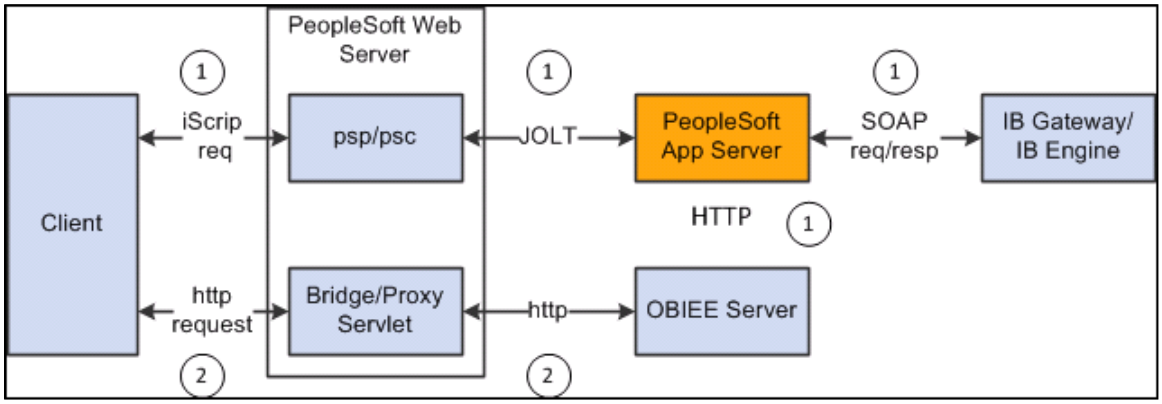
1. Acts as a reverse proxy server to avoid any cross domain issues.
2. Checks the validity of the OBIEE server for each request
3. Forwards each request to the OBIEE server by copying the PeopleSoft headers—including PeopleSoft cookies.
4. Receives and unpacks the OBIEE server response.
5. Loads the unpacked server response into the client.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

6. Renders the OBIEE report in the PeopleSoft system

Image: PeopleSoft Architecture with OBIEE Integration

This diagram shows the architecture and the flow of information among the PeopleSoft web server, the PeopleSoft application server, the OBIEE server, and the Integration Broker gateway.



Note: OBIEE developers design and create reports in the OBIEE system. The OBIEE server generates the complete HTML markup for the report. The PeopleSoft system does not format the report; rather the HTML passes through the PeopleSoft system directly to the pagelet to be displayed as formatted by the OBIEE developer.
You should include only one OBIEE report on any PeopleSoft page or homepage. PeopleSoft applications do not currently support the appearance of multiple OBIEE reports on the same page, whether from the same OBIEE server or from multiple OBIEE servers.

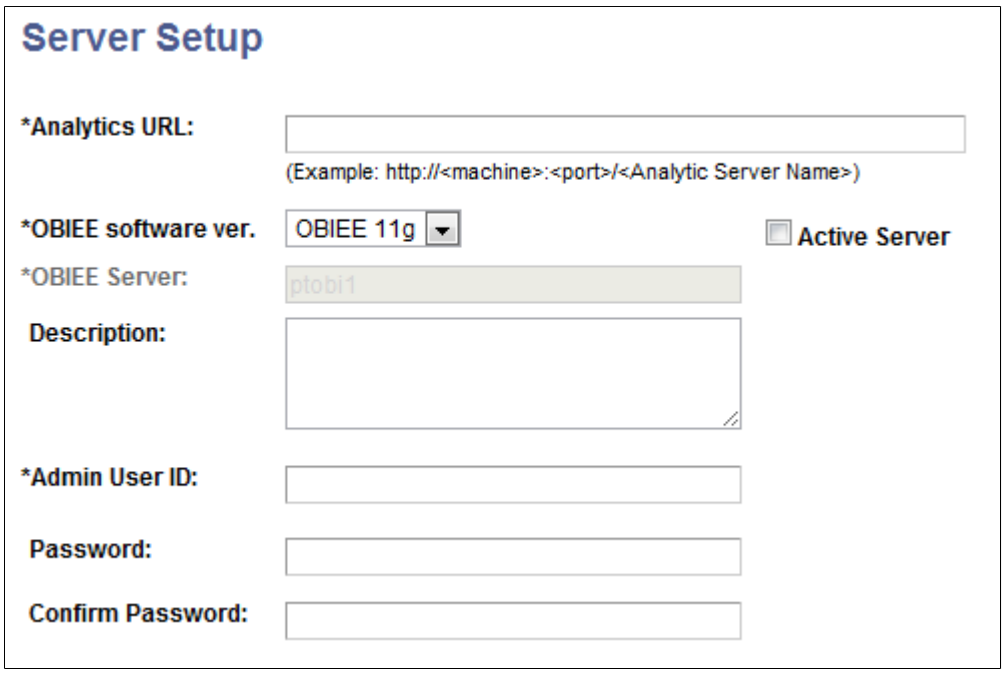
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring OBIEE Server Setup

Access the OBIEE Server Setup page. (Select PeopleTools, Portal, OBIEE Setup, Server Setup.)

Image: Server Setup page

The following screenshot illustrates configuring OBIEE server setup on the Server Setup page.



Use the OBIEE Server Setup page to define the location and administrative information that is necessary to access the OBIEE reports.

Analytics URL Enter the URL of the OBIEE server, for example, http://ptsft12.us.oracle.com:8739/AnalyticServer1

OBIEE software ver. Specify the OBIEE software version. PeopleSoft applications support OBIEE 10g and 11g. The default value is *OBIEE 11g*.

OBIEE Server This value is supplied by default from the OBIEE server name that you define when you add a server.

Active Server Toggle this check box to make this server available or to take it offline in the PeopleSoft application.

Description Enter a description of the server definition. This field works well for documentation purposes and when you are using multiple analytic servers.

Admin User ID Enter the user ID of an OBIEE server administrator. Enter a value to ensure that you do not have to sign in when the system displays the OBIEE pagelet.

Note: This user should already exist in the OBIEE server.

Password Enter the OBIEE administrator's password.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Confirm Password Reenter the OBIEE administrator's password for confirmation.

Configure the Web Server Files

You must include the OBIEE server URL in the configuration.properties file.

To configure the web server files:

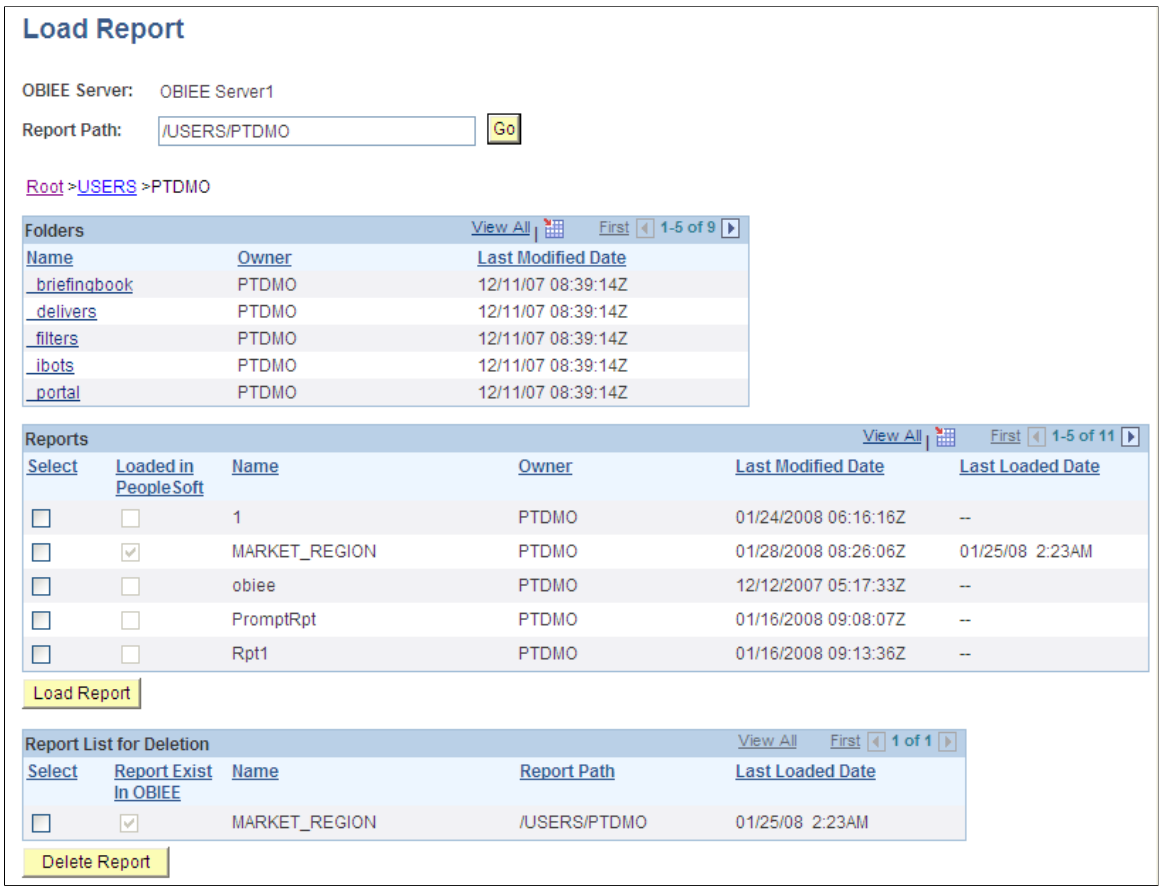
1. Using a text editor, open the configuration.properties file in the web server deployment folder located here: <*Web_server_deployment_home*>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\ps\configuration.properties.
2. Find OBIEEServers= and enter the server URL. If you have multiple OBIEE servers, enter the URLs as a comma separated list.
3. Save the file.
4. Stop and restart the web server.

Loading OBIEE Report Definitions

Access the Load Report page. (Select PeopleTools, Portal, OBIEE Setup, Load Report Definition.)

Image: Load Report page

The following screenshot illustrates loading OBIEE report definitions on the Load Report page.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Report Path Enter the directory from which to retrieve OBIEE reports. The format should be */USERS/<PS_UserID>/,* for example */USERS/PTDMO.*

Go Click to retrieve the OBIEE reports defined in the report path directory

Folders

The PeopleSoft application retrieves this information from the OBIEE system.

Name This display-only field displays the names of folders within the root directory.

Owner This display-only field displays the user ID of the owner as indicated in the OBIEE server.

Last Modified Date This display-only field displays the date that the OBIEE report folder was last modified.

Reports

Select Select to choose the OBIEE report to load into the PeopleSoft application.

Loaded in PeopleSoft This display-only field indicates whether the report has been loaded into the PeopleSoft application.

Name This display-only field indicates the name of the OBIEE report as entered in the OBIEE system.

Owner This display-only field displays the user ID of the owner as indicated in the OBIEE server.

Last Modified Date This display-only field displays the date that the OBIEE report was last modified.

This information is helpful in determining whether the version of the report loaded into the PeopleSoft application is the most current.

Last Loaded Date This display-only field displays the date that the OBIEE report was last imported into the PeopleSoft application.

Report List for Deletion

Select Select to choose the OBIEE report to delete from the PeopleSoft application.

Report Exist In OBIEE This display-only field indicates whether the report has been deleted from the OBIEE server.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	Pagelets based on reports that you remove from the OBIEE server will *not* produce results in the pagelet.
Name	This display-only field indicates the name of the OBIEE report as entered in the OBIEE system.

Loading OBIEE Reports into PeopleSoft Applications

To load OBIEE reports into PeopleSoft applications:

1. Select the Select check box for the reports that you want to load into the PeopleSoft application.

   Note: You can select multiple check boxes.

2. Click the Load Report button.

Note: Loading OBIEE reports into the PeopleSoft application does *not* remove the report from the OBIEE server.

Deleting OBIEE Reports from PeopleSoft Applications

To delete OBIEE reports from PeopleSoft applications:

1. Select the Select check box for the reports that you want to delete from the PeopleSoft application.

   Note: You can select multiple check boxes.

2. Click the Delete Report button.

   Note: Deleting OBIEE reports from the PeopleSoft application does *not* delete the report from the OBIEE system.

Viewing OBIEE Report Details

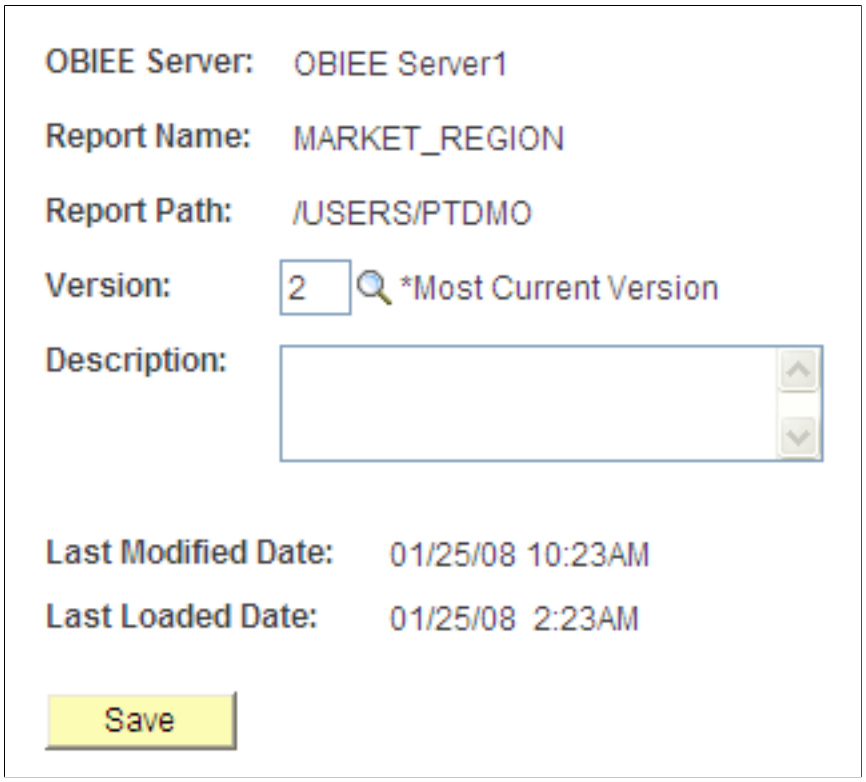
Use the Report Details pages to view information about OBIEE reports.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Report Detail page. (Select PeopleTools, Portal, OBIEE Setup, View Report Detail.)

Image: Report Detail page

The following screenshot illustrates viewing OBIEE report details on the Report Detail page.



Version

Click the prompt to select the version of the OBIEE report to view.

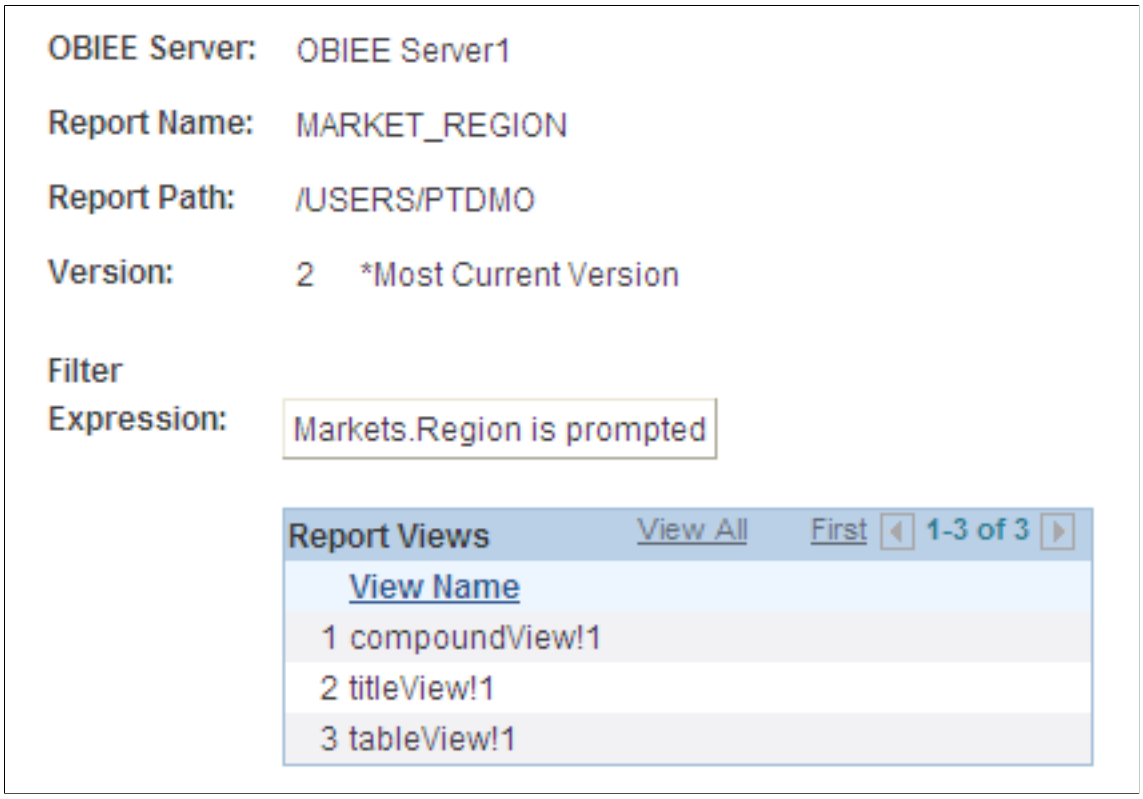
Note: Each time you load the same report from the OBIEE server into the PeopleSoft application—regardless of whether the report has changed since last being loaded— the report version number increases by 1.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Report Expression page. (Select the Report Expression tab.)

Image: Report Expression page

The following screenshot illustrates an example of Report Expression page.



Filter Expression

This display-only field indicates whether the OBIEE report has embedded filters. The *is prompted* filter indicates that the preceding field—in this case Markets.Region—is context-sensitive to the target page on the PeopleSoft application.

View Name

This display-only field indicates the types of views created by the OBIEE report builder as stored on the OBIEE server.

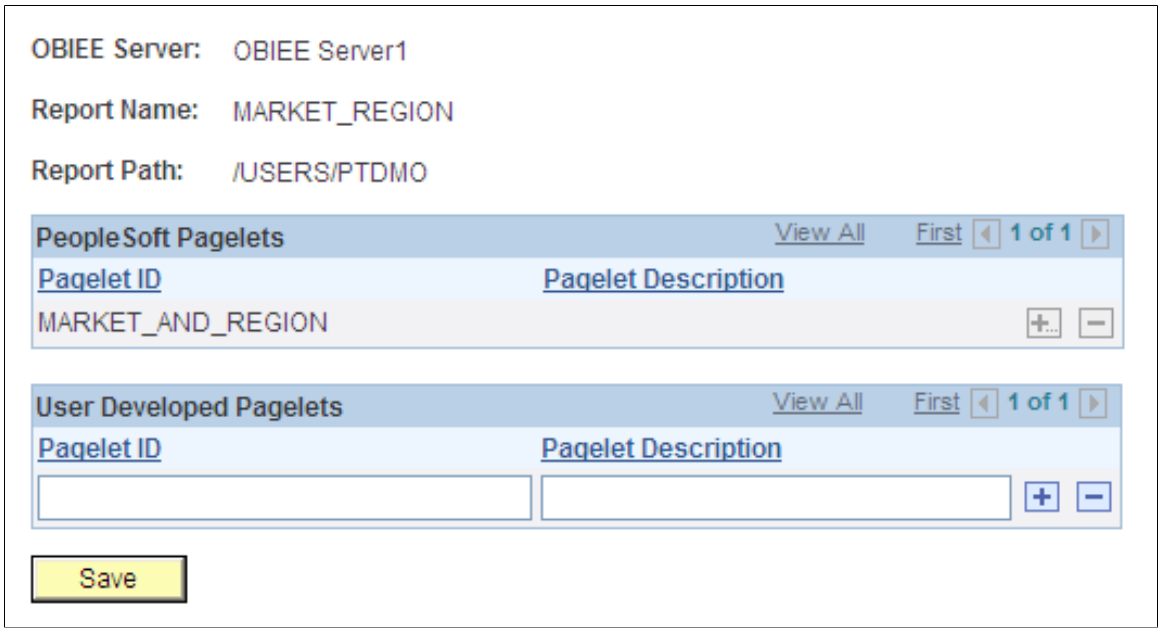
See Oracle Business Intelligence documentation for explanations of the available types of views.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Report References page. (Select the Report References tab.)

Image: Report References page

Thi example illustrates the fields and controls on the Report References page.



PeopleSoft Pagelet ID This value is supplied by default from the *PTPPB_PAGELET_ID* content reference attribute value that you enter on the pagelet Content Ref Administration page.

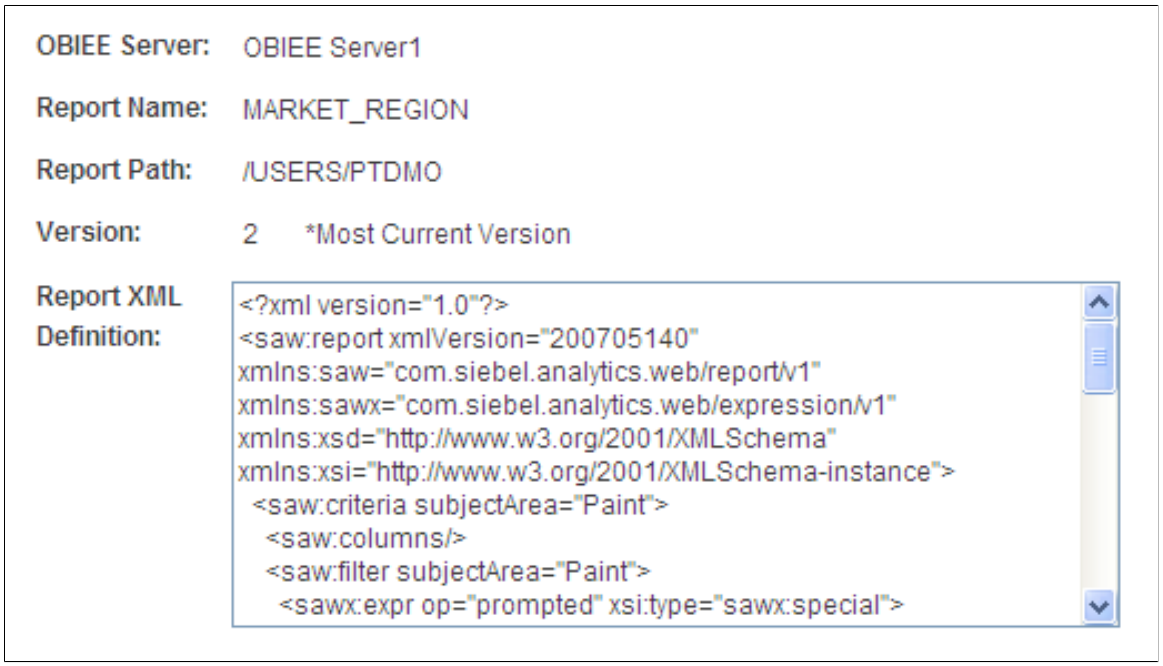
User Developed Pagelet ID Enter the pagelet ID of any user-developed pagelets that you want to track. This field is informational only and has no validation behind it.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Xml Definition page. (Select the Xml Definition tab.)

Image: Xml Definition page

The following screenshot illustrates an example of Xml definition page.



Report XML Definition This field displays the XML tags and embedded data for the version of the report as it is currently loaded. Move the cursor inside the field and press Ctrl + A to select the text in its entirety.

Creating OBIEE Report Pagelets

This section discusses how to use Pagelet Wizard to create OBIEE report pagelets:

Using Pagelet Wizard to Create OBIEE Report Pagelets

To create OBIEE report pagelets:

1. Load the OBIEE report into the PeopleSoft application.
2. Select PeopleTools, Portal, Pagelet Wizard, Pagelet Wizard.

   Pagelet Wizard guides you through the process, by displaying a sequence of pages. Each page corresponds to a step in the process. A guide appears in a header above the page to indicate your progress.
3. Specify the pagelet information in step 1.

   See Step 1: Entering Pagelet Identifying Information.
4. Specify *OBIEE* as the data type in step 2. In addition, select the report ID and report view ID from which the pagelet will read data.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: If you have not loaded the OBIEE report into the PeopleSoft application, this step fails.

See Selecting an OBIEE Report Data Source.

5. Specify the data source parameters.

   See Specifying OBIEE Data Source Parameters.

6. Select *Passthru* as the display format.

7. Specify display options, such as additional text and a search capability.

   See Step 5: Specifying Pagelet Display Options.

8. Specify the publishing options.

   See Specifying Pagelet Publication Options.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Part IV

Working with Related Information

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 16

Configuring Interwindow Communication

Understanding Interwindow Communication

This topic provides an overview of interwindow communication and discusses:

- Interwindow communication process flow.
- Interwindow communication events and messages.
- Interwindow communication considerations and limitations.

Related Links

Special Uses of Interwindow Communication

Interwindow Communication

Interwindow communication (IWC) is a frame-to-frame messaging mechanism that enables browser window objects—such as pages or pagelets—to be aware of and react to field-level data changes that occur in other browser window objects that share the same browser session. Interwindow communication enables you to publish field-level changes as they occur so that any subscribing page or pagelet becomes instantly aware of the change and then, can act on it in real time.

Interwindow communication does not depend on any specific portal template, frame, or iframe and works throughout PeopleSoft applications in many scenarios for which you require interaction between browser window objects. You can configure interwindow communication to operate between pagelets, between the page and a pagelet, and also within the same pagelet, if necessary. Although interwindow communication is ideal for use with related content and WorkCenter pages, interwindow communication is independent of both technologies. In addition, interwindow communication does not depend on Integration Broker.

In a simple example, a dashboard includes the Employees pagelet, which contains a list of employee names. The dashboard also includes an Employee Projects pagelet, which contains the project listing for a specific employee. Interwindow communication enables you to click a name in the Employees pagelet and *simultaneously* have the Employee Projects pagelet retrieve the project listing for the selected employee.

In a different example, interwindow communication can be used to refresh a pagelet with data that has been updated in the database. For example, the target content area of a WorkCenter displays transactions related to the benefits enrollment process. The pagelet area of the WorkCenter includes a Benefits Summary pagelet that itemizes the cost to the employee of each benefit selection. When the employee saves a benefit selection on a transaction page in the WorkCenter, standard component processing saves the data to the database. When the server processing returns, the interwindow communication publish

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

event fires indicating that a user action has occurred. The Benefits Summary pagelet subscribing to this event performs a full refresh to retrieve the revised data from the database.

Image: Interwindow communication example: A WorkCenter page before message publication

In the following example from a performance management WorkCenter, a publication event on the transaction page causes a full pagelet refresh in the pagelet area. The WorkCenter page displays a document for approval. After clicking the Approve button, the action of clicking the Confirm button triggers the publication of an interwindow communication message.

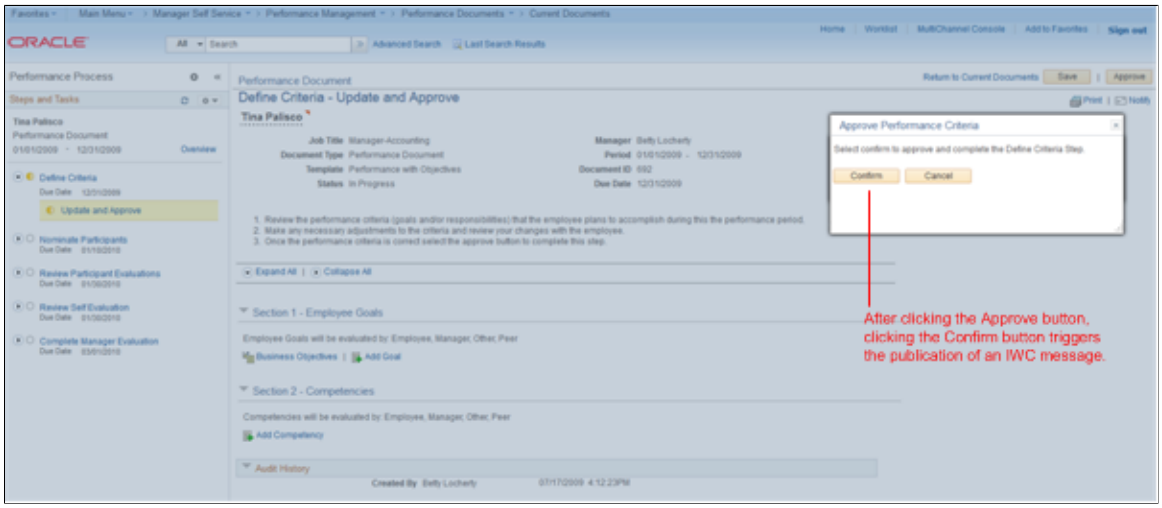
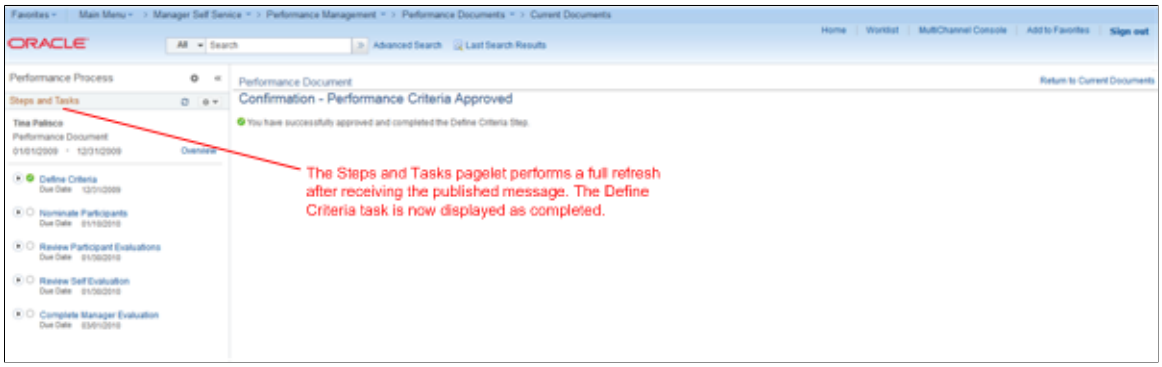


Image: Interwindow communication example: The WorkCenter page after message publication

The Steps and Tasks pagelet on the left has subscribed to this interwindow communication message and performs a full refresh after receiving the published message. The Define Criteria task in the pagelet now shows that it has been completed:



Interwindow Communication Process Flow

Interwindow communication interactions occur according to this simplified process flow:

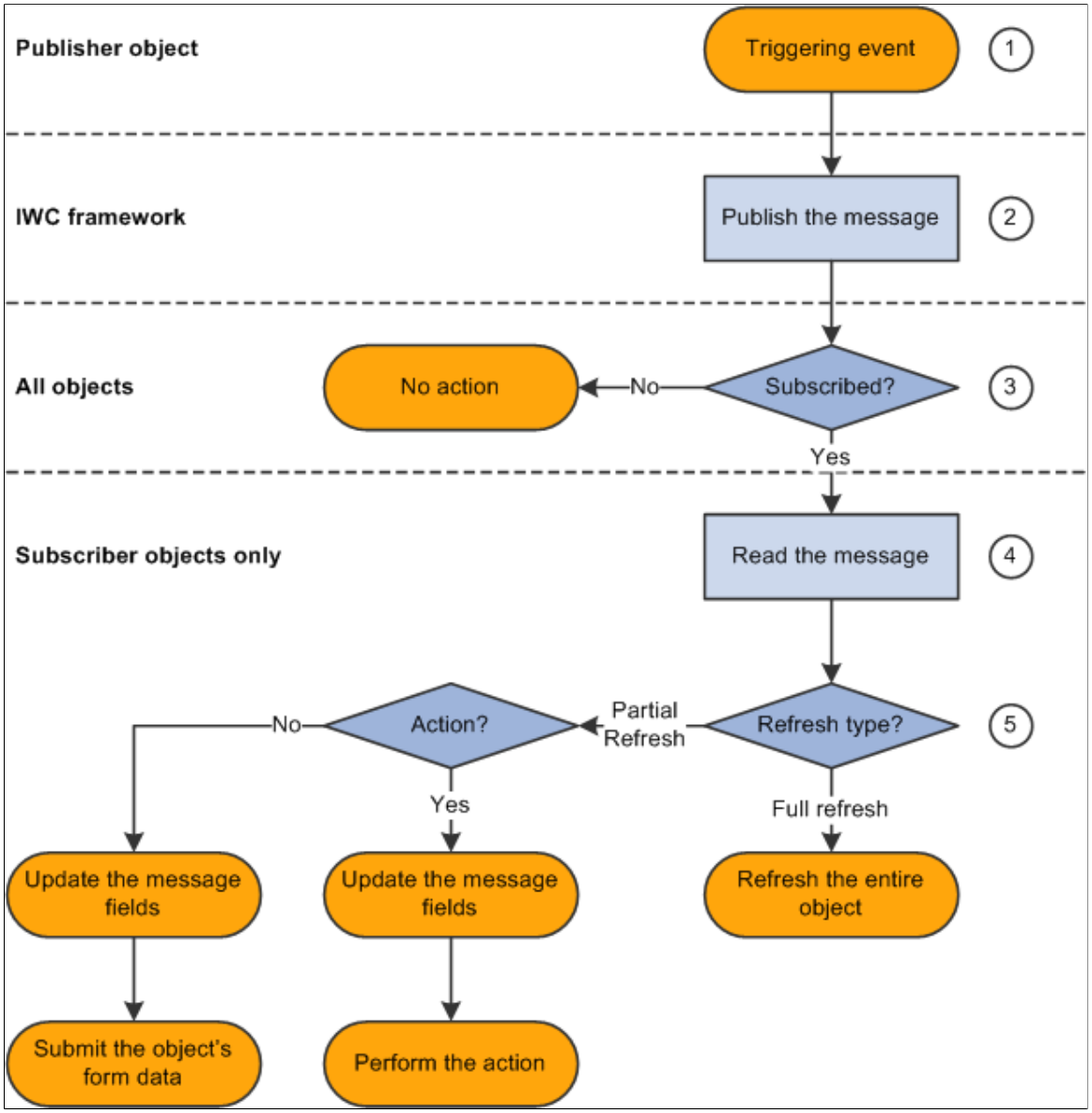
1. The triggering event occurs (click or change).
2. The interwindow communication message is published.
3. Each object (pagelet or page) determines whether it is subscribed to the message.
4. Subscriber objects only read the message.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

5. Subscriber objects only act according to the subscription definition.

Image: Interwindow communication process flow

The following diagram presents the flow of these actions and events, which comprise the publish and subscribe process of interwindow communication:



The following table presents the process flow in more detail:

Actor	*Step*	*Description*
Publisher object	**1.** Triggering event occurs.	User clicks a button or link or changes the value of a field that is configured for publication.
Interwindow communication framework	**2.** Publish the message.	A simple string formatted message is sent to the top window in the browser session—for example, that would be the homepage or dashboard window.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Actor	*Step*	*Description*
All objects	**3.** Determine whether subscribed.	All objects determine whether they are subscribed to the event.
Subscriber objects only	**4.** Read the message.	All objects in the window that are subscribed to the event read the message.

Subscriber objects only	**5.** Act according to the subscription definition.	Objects in the window that are subscribed to the event do one of the following:
		• If a full refresh is set, refresh the entire object (with URL parameters if specified). Any message data is essentially lost.
		• If a partial refresh is set:
		  • If an event action is defined, refresh only the field(s) specified in the subscribing message definition with the published message data *and then* perform that action.
		  • Otherwise, refresh only the field(s) specified in the subscribing message definition with the published message data *and then* submit the form data.
		  Important! Submitting the form data forces a trip to the application server regardless of whether the field or component is in deferred processing mode.

Note: Interwindow processes occur *after* any application server interactions invoked by the user's action. For example, a publication event could be defined for clicking a link, which merely retrieves data from the database, or it could be defined for clicking a Save button, which invokes component save processing to save updated data to the database. Interwindow communication fires after the server interaction returns to the browser.

Interwindow Communication Events and Messages

Interwindow communication involves publishing and subscribing to messages.

Triggering Events

The first half of interwindow communication is a JavaScript event that triggers the publication of a message:

- **OnChange**

  The user changes a value in a field on a transaction page or pagelet.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- **OnClick**

  The user clicks a button or link on a transaction page or pagelet.

Message Subscriptions

The second half of interwindow communication is the consumption of the published message by subscribing objects only:

- **Full refresh**

  If full refresh is set for the object, the entire object is refreshed from the database (with URL parameters if specified). Therefore, any message data is essentially lost.

- **Partial refresh**

  If partial refresh is set for the object, only the data as specified in the message definition is updated, not the entire object.

Important! In addition, by requiring that some HTML field be identified as the acted upon field, interwindow communication ensures that a server trip is invoked regardless of whether the component or field is in deferred processing mode.

Interwindow Communication Considerations and Limitations

Page and Pagelet Types

This table describes the types of pages and pagelets that interwindow communication supports and the type of refresh that is available to each page and pagelet type:

Page or Pagelet Type	*Description*	*Refresh Support*
Transaction pages	The standard PeopleSoft page that appears in the target content area.	• Partial refresh. • Full refresh.
Component-based pagelets	A pagelet built by using Application Designer.	• Partial refresh • Full refresh.
iScript or Pagelet Wizard pagelets	The PeopleCode-based pagelets built by using Pagelet Wizard or by registering an iScript.	Full refresh.

Considerations and Limitations

Note the following considerations when implementing interwindow communication:

- The domain for interwindow communication is the entire, *visible* HTML page.

  This domain includes all pagelets and active transaction pages visibly displayed, but excludes any tabbed pages of the component definition that are currently not displayed (not active). Therefore, the visible HTML page can encompass multiple component and page definitions.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- A subscribing page or pagelet must be completely loaded or it will fail to consume a published message.
- A field that will participate in interwindow communication must have a unique HTML field ID within this domain.

  Use developer tools in the browser to determine HTML field IDs. See Identifying HTML Field IDs.
- Component definitions for objects participating in interwindow communication must remain in deferred processing mode.

  Page definitions and field definitions can be set to either deferred or interactive mode as needed.
- A publishing field or a subscribing field can include FieldChange PeopleCode.
- However, a publishing field *should not* include FieldEdit PeopleCode, which typically includes errors and warnings.

  An interwindow communication event will not fire on a publishing field that includes FieldEdit PeopleCode.
- If a non-visible field is required to either publish or subscribe to an interwindow communication event, change the style of the page field properties to PTTRANSPARENT (on the Record tab in Application Designer).

  Do not set the field to invisible (on the Use tab); fields set to invisible in this manner do not get generated as part of the HTML.
- In a modal secondary window, you cannot configure a publication event on the close icon (the "X" with id="ptModClose_0") in the upper right corner of the window.

  Instead, you must configure the publication event on a button such as a Save, OK, or Cancel button within the window.

Publishing and Subscribing to IWC Messages

This topic discusses:

- Identifying HTML field IDs.
- Publishing interwindow communication messages.
- Subscribing to interwindow communication messages.
- Special uses of interwindow communication.

Identifying HTML Field IDs

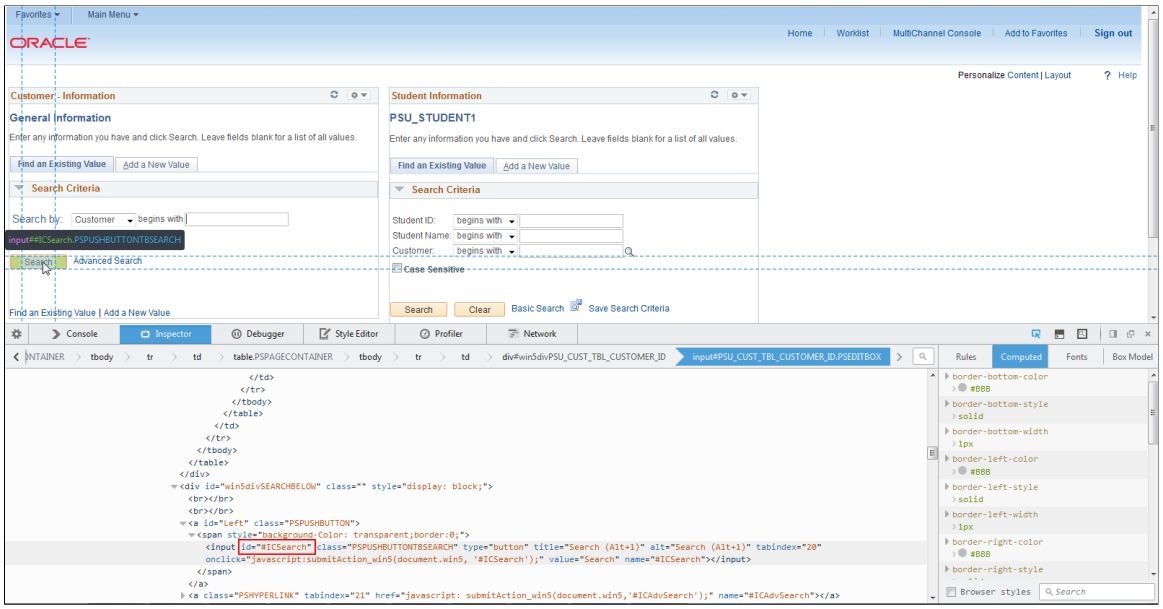
To participate in interwindow communication as a triggering field, as the acted upon field, or as a message data field, the HTML field must have a value for the id attribute, and this value must be unique within the HTML page domain. You can use your browser's built-in developer tools to examine the HTML code for a page and to extract the value of the id attribute for a field. Developer tools are available

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

as part of Chrome, Firefox, Internet Explorer, and Safari. In addition, there are third-party add-ons (for example, Firebug) that provide an alternative toolset for web developers.

Image: Identifying the HTML id attribute using web developer tools

In the following example showing the web developer tools integrated into Firefox, the Search button in the Customer Information pagelet has been selected.



In developer tools pane at the bottom of the browser window, the corresponding line of code has been highlighted by the web developer tool:

```
<input id="#ICSearch" class="PSPUSHBUTTONTBSEARCH" type="button" title="Search (Alt⇒
+1)" alt="Search (Alt+1)" tabindex="20"
onclick="javascript:submitAction_win5(document.win5, '#ICSearch');" value="Search" ⇒
name="#ICSearch"></input>
```

Within that line of HTML code, the id attribute and its value (`#ICSearch`) has been enclosed in a red rectangle:

```
id="#ICSearch"
```

Publishing Interwindow Communication Messages

Use the IWC Message Events page to define interwindow communication publication and subscription events.

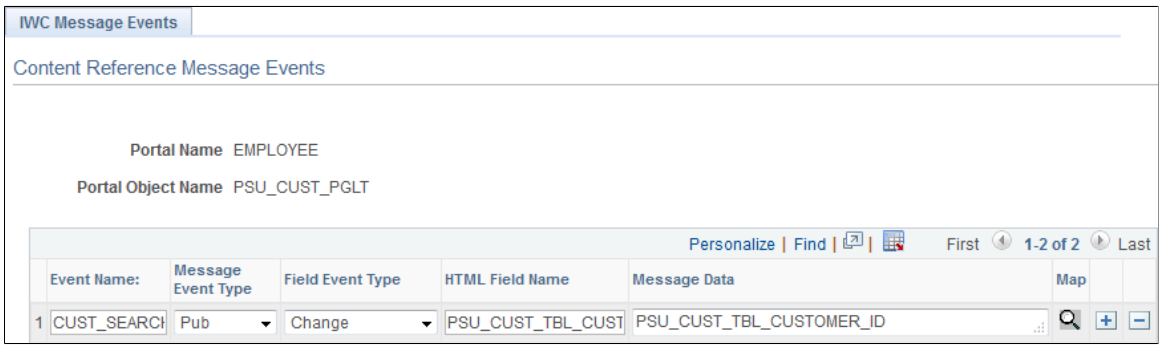
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

- PeopleTools, Portal, Inter Window Communication.
- Navigate to the content reference definition:
  1. PeopleTools, Portal, Structure and Content.
  2. Browse through the folder hierarchy in the portal registry.
  3. Click the Edit link for a content reference definition.
  4. Click the IWC Message Events link.

Image: IWC Message Events page (publication event)

This example illustrates the fields and controls on the IWC Message Events page for a publication event.



To create an interwindow communication publication message:

1. In the Event Name field, enter a name for this message. Because messages are subscribed to by event name only, this value should be unique for all messages defined within this portal.
2. Select *Pub* as the message event type.
3. Select either *Change* or *Click* as the field event type.
4. In the HTML Field Name field, enter the HTML field ID of the *triggering field* as it appears in the HTML source.

   Important! To participate in interwindow communication as the triggering field, as the acted upon field, or as a message data field, the HTML field must have a value for the id attribute, and this value must be unique within the HTML page domain. See Identifying HTML Field IDs for more information.

5. Enter the HTML field ID of the field (or fields) that you want to publish as the message data. Separate multiple field IDs with a comma.

   Note: The triggering field and the message data field are not necessarily the same field.

6. Save your changes.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

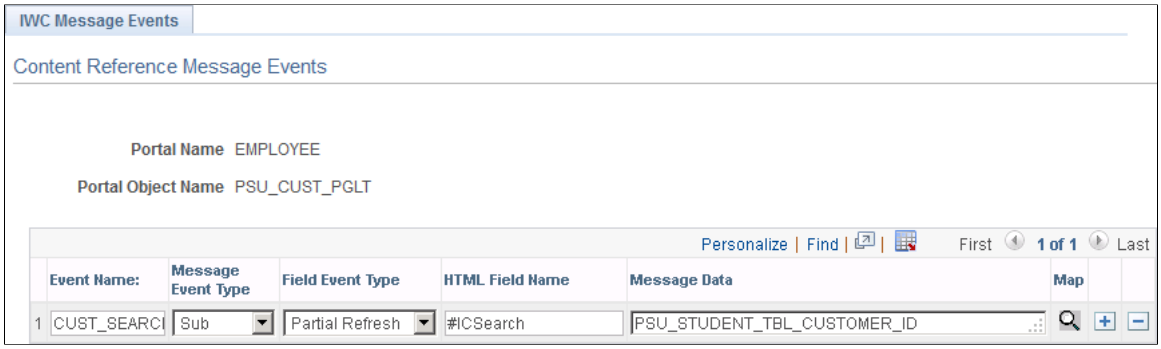
Note: While the Map button is displayed for all interwindow event definitions, the Map button is active only for push notification event subscriptions of type *Server Sub*. See Special Uses of Interwindow Communication.

Subscribing to Interwindow Communication Messages

Use the IWC Message Events page to define interwindow communication publication and subscription events.

Image: IWC Message Events page (subscription event)

This example illustrates the fields and controls on the IWC Message Events page for a subscription event.



To subscribe to an interwindow communication message:

1. In the Event Name field, enter the name of the message as it was defined in the publication event.
2. Select *Sub* as the message event type.
3. Select either *Full Refresh* or *Partial Refresh* as the field event type.

   Important! For iScript and Pagelet Wizard pagelets, a full refresh is the only valid event type.

4. In the HTML Field Name field for a partial refresh subscription only, enter the HTML field ID of the *acted upon field* as it appears in the HTML source.

   Important! For a full refresh subscription, the *field to be acted upon* is required to be blank.

   Important! To participate in interwindow communication as the triggering field, as the acted upon field, or as a message data field, the HTML field must have a value for the id attribute, and this value must be unique within the HTML page domain. See Identifying HTML Field IDs for more information.

5. In the Message Data field, enter the HTML field ID of the field (or fields) that you want to receive the message data. Specify receiving fields in the same order as the message data is defined in the publishing message; separate multiple fields IDs by a comma.

   Note: For a full refresh subscription, the *message data field* is optional.

6. Save your changes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: While the Map button is displayed for all interwindow event definitions, the Map button is active only for push notification event subscriptions of type *Server Sub*. See Special Uses of Interwindow Communication.

Special Uses of Interwindow Communication

Several PeopleTools features use interwindow communication in specific and unique manners:

- Interwindow communication can be used to synchronize changes between pivot grid pagelets. The publishing field name is always *PVGPUB*; the subscribing field name is always *PVGSUB*. In addition, there is a specific nomenclature for specifying a filter or a prompt as either the published message data or as the subscribing field.

  See "Pivot Grid Pagelet Overview" (PeopleTools 8.54: Pivot Grid) for more information.

- Interwindow communication can be used to subscribe to push notification events defined on the server. Always select *Server Sub* as the event type; then use the Map Event Data To Page Elements page to complete configuration of the remaining items by mapping event data to page elements.

  See "Subscribing to an Event" (PeopleTools 8.54: System and Server Administration) for more information.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 17

Developing and Configuring Related Content Services

Understanding the PeopleSoft Related Content Framework

This section lists prerequisites and limitations, and discusses:

- PeopleSoft Related Content Framework flexibility.
- Explicit definition versus anonymous services.
- The related content development process.
- Delivered related content services for PeopleSoft Interaction Hub.
- Delivered related content services for Oracle WebCenter.

The PeopleSoft Related Content Framework enables developers and subject matter experts to link application pages with contextually relevant collaborative content.

The primary goal of related content services is to create opportunities for the user to view or quickly access information that supports, extends, and augments any PeopleSoft transaction. The PeopleSoft Related Content Framework acts as an enterprise *mashup,* tying together all types of content such as queries, discussion threads, PeopleSoft component data, Oracle Business Intelligence Enterprise Edition (OBIEE) analytics, queries, discussion threads, and any relevant non-PeopleSoft data on PeopleSoft application pages.

Related content services can be any collaborative, analytical, or informational content that is useful for completing business tasks. Related content services include any content such as PeopleSoft content references; pagelets that are offered as services to be consumed by the applications; or related discussion forums, wiki pages, or links. Related content services can be configured within the same database, among multiple PeopleSoft databases, and between a PeopleSoft database and external sources.

Prerequisites

This table lists the necessary and optional prerequisites to defining and configuring related content services:

Task	***Reference***
Enter an authentication domain in the web profile.*	See Configuring General Portal Properties.
Define the content URI text and portal URI text for each active node in your system.*	See Setting Portal Node Characteristics
Configure single signon between your system and any systems providing remote content.	See "Understanding Single Signon" (PeopleTools 8.54: Security Administration).

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Task	*Reference*
Configure the security authorization service, if required, as required by each related content service.	See "Understanding Developing and Invoking the Security Authorization Service" (PeopleTools 8.54: Security Administration).

* Indicates the prerequisite is required.

Limitations

Note the following limitations on the display and use of related content:

- A component must be registered in order to select it to assign related content.
- The related content menu and the Related Content frame will not be displayed on pages that have the pagebar option set to disabled.
- The related content menu and the Related Content frame will not be displayed for modal or secondary pages.
- If the modal or secondary page has a related actions menu or glyph on a field that has a service targeted to open in *Target Content,* the frame will be opened in a new window instead.
- The related content menus, the Related Content frame, and related actions menus and links will not be displayed for content that has been produced for WSRP consumption. For example, the Related Actions menu will not be displayed on a pivot grid that has been produced as WSRP content.

Related Links

Limitations on PeopleSoft Content for WSRP Consumption

PeopleSoft Related Content Framework Flexibility

The PeopleSoft Related Content Framework provides flexibility in how related content is defined, assigned, and accessed. You can assign, configure, and maintain related content services without access to developer tools such as PeopleSoft Application Designer. The related content service pages provide a variety of configuration options, which enable you to define related content services, associate them with pages or pivot grids, and grant access to these services based on your business rules. This flexible framework allows you to:

- Determine whether information is displayed as related content or related actions.
- Place the Related Content frame at the bottom or on the right of the transaction page.
- Configure the layout of related content and actions menus.
- Assign explicitly defined services or anonymous (non-explicit) services.
- Define local or remote resources as related content services.

Related Content Versus Related Actions

The PeopleSoft Related Content Framework allows related content services to be deployed in two ways—as related content and as related actions—based on where the content appears when the user accesses the service.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Related content* appears in the Related Content frame only, which can be selected to be displayed at the bottom or to the right of the transaction content.
- *Related actions* appear outside of the Related Content frame in a modal window, in a new browser window, in the existing browser window replacing the current transaction content, or in the existing browser window replacing the entire page.

Configure the related content service as *related content* by selecting the Configure Related Content page:

- If you want the related content to appear in a separate frame beside or below the transaction content.
- If you want the related content to be dynamically updated when the user moves to a different transaction instance, or changes a field value, and so on.

Configure the related content service as a *related action* by selecting the Configure Related Actions page:

- If you want the related content to appear in a modal window, in a new browser window, or to replace the current page.
- When you are configuring related content for pivot grids, keyword search results, SmartNavigation menus, or SmartNavigation pages.

The Related Content Frame

The Related Content frame can be configured to display at the bottom or to the right of the transaction content.

Note these points about the Related Content frame when it appears at the bottom of the page:

- The frame extends across the width of the transaction page.
- The visible area increases and decreases when you drag the handle on the upper edge of the frame in a vertical direction.
- The frame appears and disappears when you click the Expand/Collapse button on the left side of the frame header.
- The frame reloads when you click the Refresh button in the frame header .
- The frame loads the current active service in a new window when you click the Zoom icon.
- The frame closes when you click the Close button.

Note these points about the *services* that appear in the bottom frame:

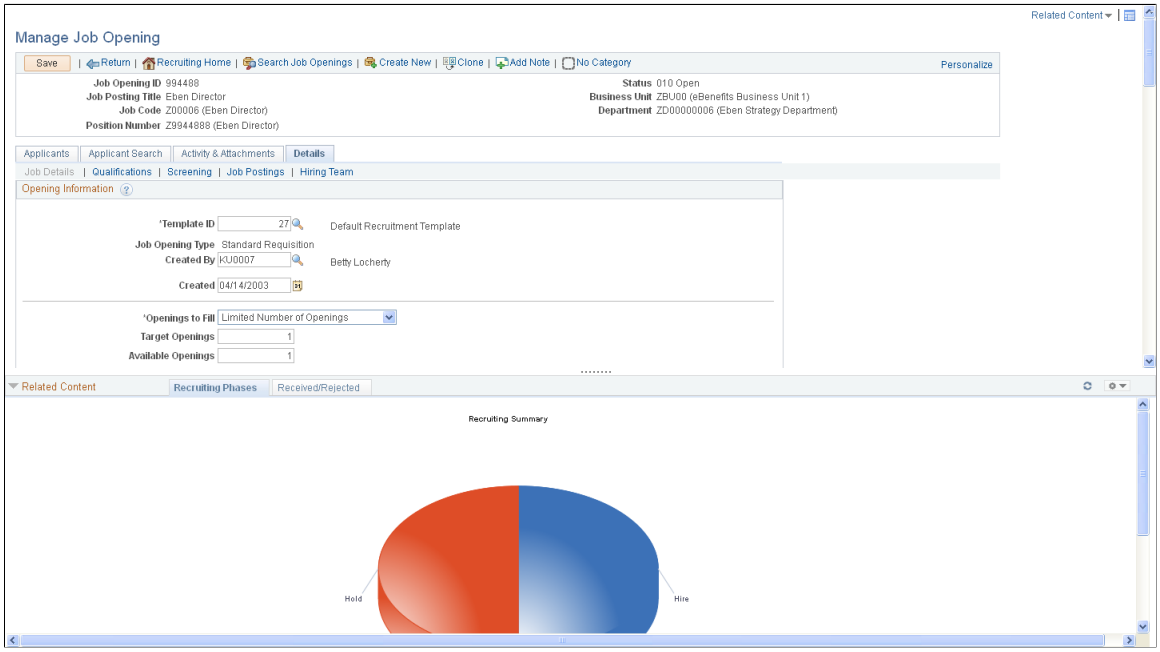
- Only one service can be active and have focus at a time.
- Multiple services appear as tabs across the top of the frame.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Users click the tab to bring the service into focus.

Image: Transaction page showing the Related Content frame at the bottom

The following example shows a transaction page with related content in the Related Content frame at the bottom of the page:



Note these points about the Related Content frame when it appears at the right of the page:

- The frame extends down the length of the transaction page.
- The visible area increases and decreases when you drag the left edge of the frame in a horizontal direction.
- The frame appears and disappears when you click the Maximize and Minimize buttons on the left border of the frame.
- The frame divides the visible area equally among all open pagelets.
- The bottom of the frame displays a More menu, which you use to access minimized pagelets.

Note these points about the *services* that appear in the right frame:

- Each service exists as a self-contained pagelet inside the Related Content frame.
- Any number of services can open be active and in the expanded state simultaneously.
- All maximized service pagelets divide the available space equally and always share the same vertical and horizontal dimensions. As users maximize and minimize other service pagelets in the frame, any service pagelets that remain open resize automatically so that the space appears equally divided.

  Note: Users cannot resize individual service pagelets.

- Each pagelet header can contain:
  - A Zoom button that users click to display the service in a new window.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

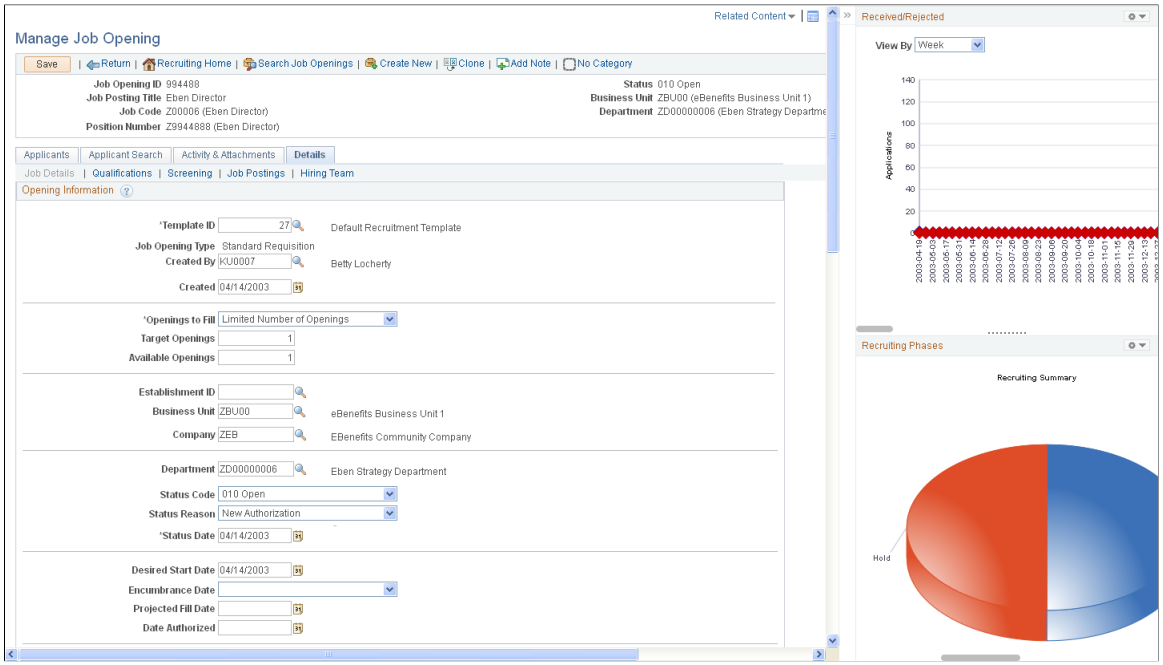
- A Refresh button that users click to reload the service.
- A Maximize/Minimize button that users click to toggle the service between the expanded and collapsed states.

Note: The expanded or collapsed state of each pagelet persists between sessions.

- Scroll bars appear on the right and bottom of a service pagelet if it is too small to fully display the content.

Image: Transaction page showing the Related Content frame at the right

The following example shows a transaction page with related content in the Related Content frame at the right of the page:



Related Content Menus

Three types of menus provide access to related content services:

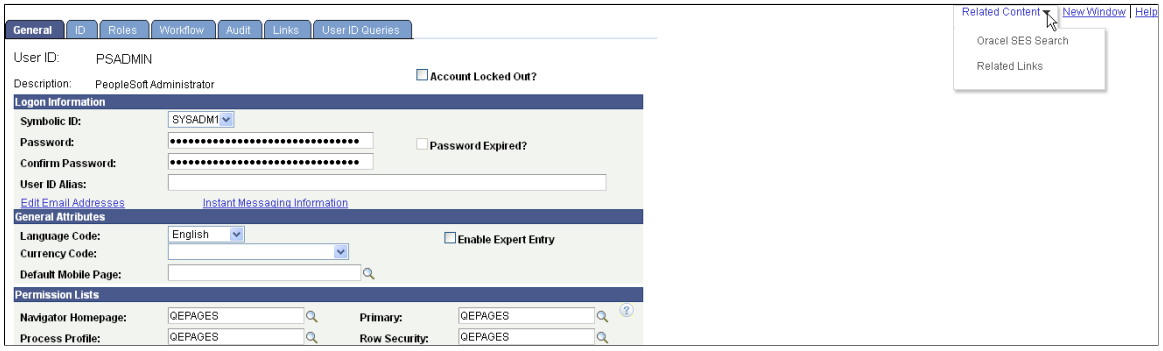
- The Related Content menu, which appears in the page bar at the top of the transaction page. The Related Content menu displays services configured as *related content* only.
- The related actions menus, which can appear on pivot grids, on any page that includes the free-standing Actions button, in SmartNavigation folders, on chart nodes in a SmartNavigation chart, and in keyword search results. The related actions menus displays services configured as *related actions* only.
- Page field contextual menus, which appear when you click the red page-field glyph. A page field contextual menu can combine both *related content* and *related actions.*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The Related Content menu, which appears in the page bar at the top of the transaction page, displays services configured as *related content* only.

Image: Related Content menu showing two services

The following example shows two services in the Related Content menu:

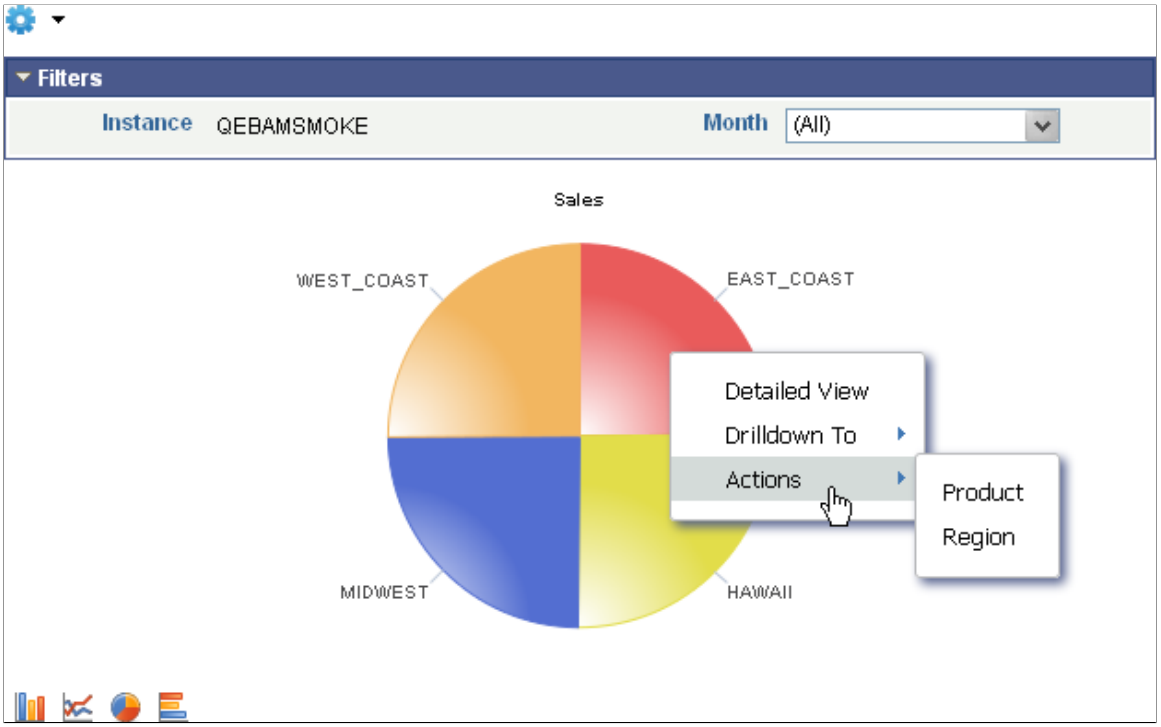


The related actions menus can appear on pivot grids, on any page that includes the free-standing Actions button, in SmartNavigation folders, on chart nodes in a SmartNavigation chart, and in keyword search results. The related actions menus displays services configured as *related actions* only. A related action menu can appear as:

- An Actions menu on a pivot grid chart.

  Image: Related actions on a pivot grid chart

  This example shows an Actions menu on a pivot grid chart:



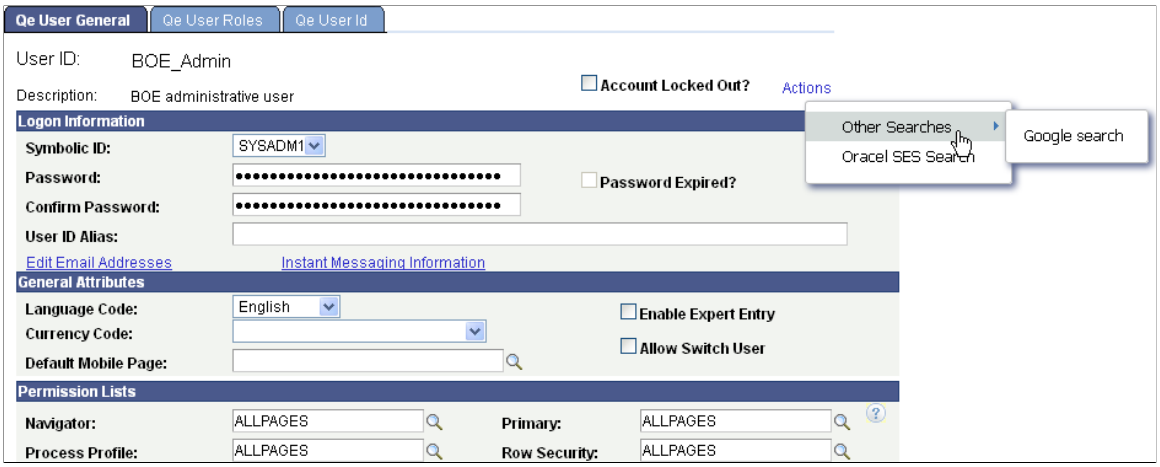
- A free-standing Actions button on any page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: The label on the Actions button is configurable within PeopleSoft Application Designer.

Image: Related actions from a free-standing Actions button

This example shows a free-standing Actions button that has been added to the page to display a list of related actions:

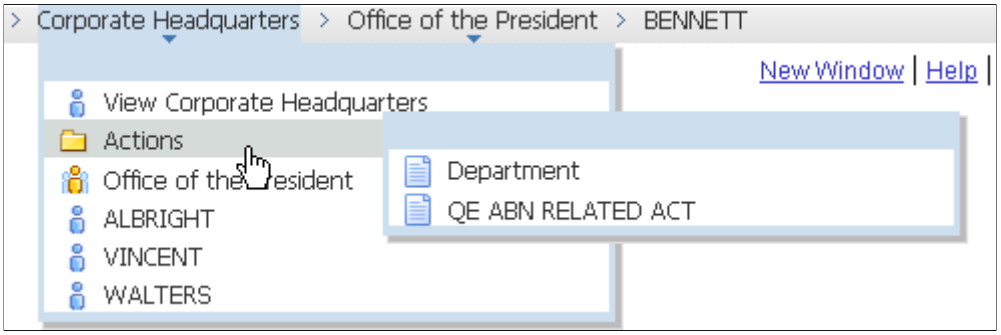


- An Actions folder in drop-down SmartNavigation menus.

Important! The Actions folder displays depending on the configuration of related actions for SmartNavigation. If the related actions are attached at the chart component level, the Actions folder will not appear in the drown down menu. If the related actions are attached to the data source's detail component, the Actions folder will appear in the drop down menu.

Image: Related actions in a drop-down SmartNavigation menu

This example shows the Actions folder in a drop-down SmartNavigation menu:



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- A drop-down Actions menu on chart nodes of SmartNavigation charts.

  Image: Related actions on a SmartNavigation chart node

  This example shows a drop-down Actions menu on a chart node of a SmartNavigation chart:

- As pop-up related actions in the search results of a keyword search.

  Image: Related actions on a keyword Search Results page

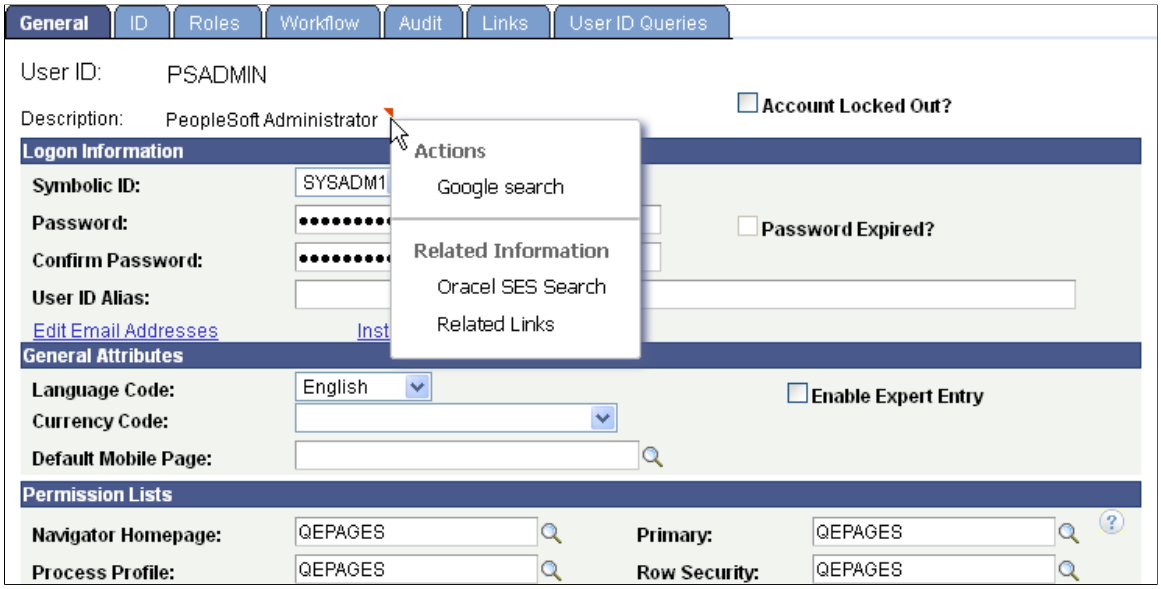
  The following example shows pop-up related actions links in the search results of a keyword search:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Page field contextual menus appear when you click the red page-field glyph. A page field contextual menu can combine both *related content* and *related actions.*

Image: A page field contextual menu showing both related actions and related content

The following example shows a page field contextual menu with a related action (under the Actions folder) plus related content (under the Related Information folder):



Related Content from Remote Systems

When you configure single signon between PeopleSoft systems, you can use these remote resources as related content services:

- Embeddable pagelets
- OBIEE pagelets
- PeopleSoft components
- PeopleSoft queries
- PeopleSoft iScripts

Explicit Definition Versus Anonymous Services

Some PeopleSoft content such as components and locally registered pagelets can be used directly as *anonymous services* and do not need to be manually defined as a related content service. Content references, homepage pagelets, and template pagelets that are currently defined in the portal registry can be assigned as anonymous services and as such require no explicit related content service definition.

While the framework enables you to assign most PeopleSoft components as anonymous services without first creating an explicit service definition, you must create a PeopleSoft component service definition when:

- You want to specify the page within the component that opens when the service is accessed.
- You want to pass parameters to the component other than the key structure.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- You want to assign a component as a related action on SmartNavigation pages; only components that you specifically define as services appear in the Actions drop-down menu for SmartNavigation.

  Important! The Actions folder displays depending on the configuration of related actions for SmartNavigation. If the related actions are attached at the chart component level, the Actions folder will not appear in the drown down menu. If the related actions are attached to the data source's detail component, the Actions folder will appear in the drop down menu.

- You want to use the component as an action item in an activity guide.

The Related Content Development Process

This table lists the steps in the process flow, the roles involved during each step, and provides cross-references to supporting PeopleBooks documentation:

Step	*Role*	*Reference*
Define related content services (including entering help text).	Developer	See Defining Related Content Services. See Writing Help Text for Related Content Service Definitions.
Test related content services.	Developer	See Testing Related Content Service Definitions.
Assign related content services to content references or pivot grids.	Subject matter expert	See Selecting an Application Page (Content Reference) or Selecting a Pivot Grid.
Map service parameters and configure service layout.	Subject matter expert	See Configuring Service Parameters, Filters, and Security. See Configuring Menus. See Configuring the Related Content Frame.
(Optional) Develop integration of related content—for example, inter-service communication.	Subject matter expert	See Understanding Interwindow Communication.
Access the transaction and test the related content display.	Subject matter expert Application user	See "Accessing Related Content and Actions" (PeopleTools 8.54: Applications User's Guide).

Delivered Related Content Services for PeopleSoft Interaction Hub

PeopleTools provides the service definitions for three related content services for PeopleSoft Interaction Hub:

- Related Discussion service
- Related Links service
- Related Tags service

These related content services provide features of Oracle's PeopleSoft Interaction Hub that can be added to PeopleSoft application pages through the PeopleSoft Related Content Framework. While the service

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

definitions are provided in PeopleTools, an active PeopleSoft Interaction Hub database is required to provide the services and to store the related data.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, "Setting Up Related Content Services."

Related Discussion Service

Using the Related Discussion service, you can collaborate to share relevant information about a transaction or a transaction instance. Discussion forums in the Related Discussion service provide a platform that groups can use to discuss topics of interest in a manner similar to standalone forums in PeopleSoft Interaction Hub. Discussion forums enable multiple relevant individuals to contribute to the review and resolution of a question. Your can post discussion topics, such as issues, suggestions, or questions, and receive replies and feedback.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, "Working With the Related Discussion Service."

Related Links Service

The Related Links service enables you to create links to relevant content for the current transaction instance or for all instances in a transaction. You can link to content residing in PeopleSoft Interaction Hub—specifically, blogs, discussions, or collaborative workspaces—add file attachments, or create links to external URLs. In addition, the Related Links service enables you to create a new workspace and to link that workspace to the transaction.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, "Working With the Related Links Service."

Related Tags Service

The Related Tags service enables you to create user-specified tags for the current transaction instance. You can characterize and bookmark these transactional or business objects with terminology of your choosing, providing ease of access as well as collaborative classification of the transaction.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, "Working With the Related Tags Service."

Delivered Related Content Services for Oracle WebCenter

PeopleTools provides these service definitions for use with Oracle WebCenter:

- WebCenter Blogs service
- WebCenter Discussions service
- WebCenter Documents service

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- WebCenter Links service
- WebCenter Wikis service

Note: As of PeopleTools 8.54, integration with Oracle WebCenter through the delivered related content services has been deprecated. These service definitions may be removed from a future release of PeopleTools.

These related content services provide features of Oracle WebCenter that can be added to PeopleSoft application pages through the PeopleSoft Related Content Framework. While the service definitions are provided in PeopleTools, an active Oracle WebCenter database is required to provide the services and to store the related data.

See Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter.

Note: Oracle recommends that you have a WebCenter that you dedicate solely to your PeopleSoft applications integration.

WebCenter Blogs Service

Blogs provide a useful tool for discussing or evangelizing any type of idea, strategy, or point of view. Blogs may be projected out to a select group of people or to a wider audience. Typically, blogs invite readers to comment on the overall concepts.

The Blogs service enables users to create instance specific or shared WebCenter blogs, which appear for all transaction instances of the transaction page.

WebCenter Discussion Service

The Discussions service lets you expose discussion forums on your application pages, so users can create forums, post questions, and search for answers. For example, customers can share product reviews, or a customer service department can answer questions online. Discussion forums additionally provide the means to preserve and revisit discussions.

The Discussion service enables users to create instance specific or shared WebCenter discussions, which appears for all transaction instances of the transaction page.

WebCenter Documents Service

The Documents service provides features for accessing, adding, and managing folders and files; configuring file and folder properties; and searching file and folder content in the connected content repositories. When you set up a content repository connection to integrate the Documents service, you can add content to the transaction pages of the PeopleSoft application.

The Documents service enables users to create instance specific or shared WebCenter documents, which appears for all transaction instances of the transaction page.

WebCenter Links Service

The Links service provides a way to view, access, and associate related information. For example, in a list of project assignments, you can link to the specifications relevant to each assignment. In a discussion thread about a problem with a particular task, you can link to a document that provides a detailed description of how to perform that task.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The Links service enables users to create instance specific links or shared links, which appears for all transaction instances of the transaction page. When the transaction page or instance also uses other WebCenter related content services, related links to the other WebCenter group spaces appear by default and cannot be removed.

Unlike other WebCenter related content services, the Links service is a PeopleSoft component-based page. When you click the links in the service, the target opens in a new browser window.

WebCenter Wiki Service

A wiki is a collection of useful content or information that users can browse, update, and remove, sometimes without the need for registration. This ease of interaction and the variety of operations make wiki an effective tool for collaborative authoring, where multiple people create written content together.

The wiki service enables users to create instance specific wiki pages or shared WebCenter wiki pages, which appears for all transaction instances of the transaction page.

Related Links

Understanding WebCenter Services

Defining Related Content Services

This section provides overviews of service types, service URL parameters, and service security, and discusses how to:

- Create application class service definitions.
- Create embeddable pagelet service definitions.
- Create non-PeopleSoft URL service definitions.
- Create OBIEE pagelet service definitions.
- Create PeopleSoft component service definitions.
- Create PeopleSoft query service definitions.
- Create PeopleSoft script service definitions.
- Test related content service definitions.
- Select bulk action for related content service definitions.
- Write help text for related content service definitions.
- Upgrade or copy related content data.

Understanding Service Types

The PeopleSoft Related Content Framework supports seven explicitly definable service types. Each type is defined by the characteristics of its data source, which determines the information to form the complete URL that locates and requests the actual data.

When you select a service URL type, various service parameters appear for you to configure. The service parameters differ based on the service type that you select. For example, when you select *Embeddable*

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pagelet, the Node and Pagelet Name fields appear as service parameters because the framework must know the name of the node where the pagelet resides and the name of the pagelet to construct the URL that retrieves the pagelet and its data.

In addition to system-defined service parameters, you can enter user-defined parameters, which offers maximum flexibility in configuring the service to meet business requirements.

Definable Service Types

You can explicitly define the following related content service types:

Application Class

This service type accesses any service that is defined by means of PeopleCode application classes. Application class services can:

- Access data in the component buffer.
- Run any PeopleCode program that is valid in the FieldChange event, for example, transferring the user to another page, opening a modal window, or modifying data in the component buffer.
- Be assigned as related actions only. Application class services cannot be targeted to the Related Content frame because the target frame and the Related Content frame maintain separate sessions and separate buffer structures.

Embeddable Pagelet

This service type accesses Pagelet Wizard pagelets. The pagelet can be of any pagelet data source but it must be published as an *embeddable* pagelet.

Non-PeopleSoft URL

This service type accesses any non-PeopleSoft pages, external content, or services.

Note: Use this service type to deploy Oracle User Productivity Kit (UPK) modules and topics as related content.

See Creating UPK-Based Related Content Service Definitions.

OBIEE Pagelet

This service type accesses analytics-based OBIEE pagelets. You use Pagelet Wizard to create and publish OBIEE pagelets that use OBIEE reports as the data source. These OBIEE analytics reports are then consumable in the PeopleSoft system.

PeopleSoft Component

This service type accesses any PeopleSoft components. The component can be from the local PeopleSoft system or from a remote PeopleSoft system. The node name specifies whether the service resides in the local or remote PeopleSoft system.

The framework also enables you to assign most PeopleSoft components as *anonymous services* without first creating a service definition.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	However, in certain circumstances, you must create a PeopleSoft component service definition. See Explicit Definition Versus Anonymous Services.
PeopleSoft Query	This service type accesses PeopleSoft query definitions. The query results can also be filtered based on the query definition and the parameters that it requires.
PeopleSoft Script	This service type accesses iScript-based services. Use this service type in situations in which an iScript has been used to develop the application functionality based on PeopleSoft data or non-PeopleSoft data, such as stock quotes or weather information.

The information that the system uses to construct the seven service types differs for each type and is described in the following sections.

Related Content Services as Action Items

Action items in activity guides must be defined as related content services. When creating the service definition for an action item, you can ignore the display options and the security options. The basic PeopleTools transaction security model (user-role-permission list) is used to provide user access to the service. The following service types can be used as action items:

- Application class
- Non-PeopleSoft URL
- PeopleSoft component
- PeopleSoft query
- PeopleSoft script

See Creating and Maintaining Action Items.

Understanding Service URL Parameters

A service might need additional parameters to obtain the context for the data that the service provides. You define these parameters in the Service URL parameter grid. When the subject matter expert assigns the service to a content reference, she or he maps values to the parameters. Then, when the service is accessed from the transaction page, contextual values from fields in the buffer, system variables, or even fixed values that were mapped to the parameters are passed as arguments to the service.

Content Reference Attributes as Service URL Parameters

When you assign a content reference as an anonymous service, the system automatically defines service parameters for the components that provide either a valid search record or contextual record. You can manually define the RCF_SERVICE_PARAMETER attribute on the Content Ref Administration page for any content reference, such as for pagelets, iScripts, or for other content references that do not have a valid search record. If this RCF_SERVICE_PARAMETER attribute exists, the system automatically uses the attribute values as the service parameters for the content reference.

Note these points about the RCF_SERVICE_PARAMETER attribute.

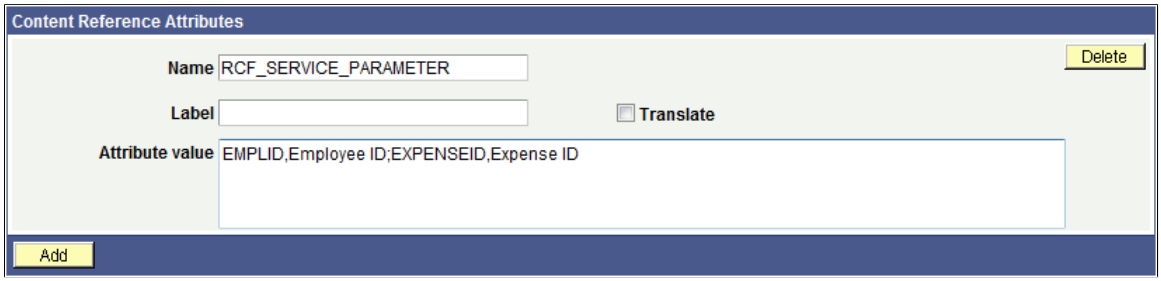
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The RCF_SERVICE_PARAMETER attribute overrides the default behavior of component-based URLs. The system will not retrieve the service parameters from a valid search record or contextual record, rather the system supplants any search record parameters with the parameters included in the Attribute Value field.
- The attribute value format should follow this pattern:

  NAME1,Label1;NAME2,Label2 with NAME identifying the service parameter and Label identifying the label for the service parameter. For example, *EMPLID,Employee ID;EXPENSEID,Expense ID*.
- The NAME attribute value should *not* include spaces.

Image: Content Reference Administration page showing the RCF_SERVICE_PARAMETER attribute

The following example shows content reference definition that uses the RCF_SERVICE_PARAMETER attribute:



Related Links

Defining Content Reference Links

Understanding Service Security

PeopleSoft Related Content Framework uses the basic PeopleTools security model, which secures local and remote PeopleSoft transaction pages and the application data that underlies these pages based on the user's assigned permission lists and roles.

For services that provide access to non-PeopleSoft content, the framework enables you to use the security model of the content provider. You should consult the content provider and your security administrator to determine the best course of action.

PeopleSoft Security Authorization Web Service

The PeopleSoft security authorization web service runs on PeopleSoft systems and certain service consumer systems such as Oracle WebCenter. The service examines the basic PeopleTools security on the providing system and can respond to a web service call from another PeopleSoft system.

The PeopleSoft Related Content Framework uses the authorization service in these two ways:

- To show and hide related content links that access PeopleSoft content.
- To grant access to third-party systems, such as Oracle WebCenter that contain PeopleSoft content.

The authorization service can check for basic PeopleTools security at the definition level and at the data, or row, level.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

At the definition level, the authorization service response determines whether related content or related actions links to PeopleSoft content appear for the user. This level of authorization is automatically engaged for any PeopleSoft content, and no other configuration is required beyond basic PeopleTools security setup.

However, row-level, or data permission, security is *not* automatic. Use the Authorization Configuration component to configure the authorization service to implement row-level security on remote systems. When the user has the appropriate permissions to view the content, then the authorization service determines whether data, or row-level security has been configured. If any such configurations exist, then the appropriate data is returned when the user clicks the related content or related actions link to access the data. Procedures for configuration of the security authorization service are covered elsewhere.

See "Understanding Developing and Invoking the Security Authorization Service" (PeopleTools 8.54: Security Administration).

Service Definition Security Options

At design time, you select which security options will be available when the service is assigned and configured for an application page. You can select one or more options to make available to the subject matter expert who assigns the service to a content reference. During the service configuration process, he or she selects the security option to be in effect when the service is accessed from a specific application page; therefore, a different security could be selected for use with a different application page. Before you allow a security option for a service, you should understand the service's security model.

Important! If you allow a security option for a service definition that isn't applicable or hasn't actually been implemented by the service itself, unexpected results can occur.

The framework offers four security options.

- Public access.
- Related content provider security.
- Related content consumer security.
- App class required.

Important! The security authorization web service is called for all PeopleSoft content regardless of the option that you select; however, the service is not called for non-PeopleSoft content.

Public Access

The Public Access security option provides access to the service based on the permissions of the public user that you specify in the web profile. The Public Access option is intended for use with services that require few parameters, such as a generic help policy or an internet search engine.

Components that use basic PeopleTools security, which provides authorization and access through permission lists and roles, are not public. If you select the Public Access option for a PeopleSoft component service definition, users must have access to the PeopleSoft component associated with the service.

See "Public Users" in Configuring Portal Security.

See "Setting Page Permissions" (PeopleTools 8.54: Security Administration).

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Content Provider Security

Some content providers offer complex and industry-specific security models and mechanisms that are native to the service. The Related Content Provider Security option is intended for use with services that implement security that is specific to their data model or the content that they provide.

As an example, consider the Compensation Details component in Payroll as the producer and the Employee component in an HCM environment as the consumer. If you select Related Content Provider Security, then the provider system (Compensation Details/Payroll) must provide the security.

As another example, the Related Discussion service of PeopleSoft Interaction Hub implements both provider security and consumer security. When provider security is selected, then the security model of the PeopleSoft Interaction Hub discussion forums is used, which means that the forum moderator is responsible for adding members and privileges for access to the forum.

Related Content Consumer Security

PeopleSoft applications use basic PeopleTools security, which provides authorization and access to the many elements of your PeopleSoft application through the assignment of permission lists and roles to users. The Related Content Consumer Security option is intended for use with services that rely on the basic PeopleTools for authorizing access to the service content, for example:

- PeopleSoft component, embeddable pagelet, OBIEE pagelet, PeopleSoft query, and PeopleSoft iScript services that access remote PeopleSoft nodes.
- Oracle WebCenter services that create group spaces linked to PeopleSoft transaction pages. The WebCenter maintains the group space security, but authorization and access are based on the permissions of the PeopleSoft transaction page.

For example, consider the Compensation Details component in Payroll as the producer and the Employee component in HCM as the consumer. If you select Related Content Consumer Security, then the consumer system (Employee/HCM) must provide the security. Therefore, the provider system (Compensation/Payroll) creates an authorization request to the consumer (Employee/HCM), which responds with the authorization details for that user.

App Class Required

When selected, this option indicates that a service filter is required when this service is assigned to an application page. This option can be selected as the only security option, and it can be selected in conjunction with other security options, which forces the selection of a service filter when the service is configured for an application page.

Related Links

"Understanding Permission Lists" (PeopleTools 8.54: Security Administration)

"Understanding Roles" (PeopleTools 8.54: Security Administration)

"Understanding User Profiles" (PeopleTools 8.54: Security Administration)

"Understanding Developing and Invoking the Security Authorization Service" (PeopleTools 8.54: Security Administration)

Common Elements Used in This Section

This section describes the common elements used in defining related content services. The elements appear in the lists in the order that they appear on the respective pages

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Content Service

This table describes the fields on the Define Related Content Service page that are common to all service URL types.

Service ID	Specifies the unique identifier of the service definition.
Service Name	Enter a short name for the service, which will be used as the default label for the service when it is assigned as related content.
Object Owner ID	Select which product, feature, or application owns this service definition.
Description	Enter a description for the related content service.
Bulk Action	Select this option to enable service definition to execute a related action on multiple rows of data. If the Bulk Action option is not selected, you can execute a related action on a single row of data at a time. The Bulk Action option is available for the following URL types: • Application Class • PeopleSoft Script • PeopleSoft Component See Selecting Bulk Action for Related Content Service Definitions.
URL Type	Select from seven service types that build the URL that defines the location and parameters of the service: *Application Class* *Embeddable Pagelet* *Non-PeopleSoft URL* *OBIEE Pagelet* *PeopleSoft Component* *PeopleSoft Query* *PeopleSoft Script* See Understanding Service Types.
Write help text	Click to open a help text page where you can create instructions that are specific to assigning this related content service and mapping its service parameters. A subject matter expert can access the help text for assistance when configuring a related content assignment.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Writing Help Text for Related Content Service Definitions.

Copy Service Definition

Click to clone the service definition and its parameters.

See Copying Related Content Service Definitions.

URL Information

This table describes URL information fields. These fields are common to several or all service URL types.

Node Name

Enter the node name of the PeopleSoft system where the service resides.

If the service resides in the same PeopleSoft system, the node name will be either *LOCAL_NODE* or any default local node of the PeopleSoft system.

Populate parameters

Supplies the service parameters in the Service URL parameter grid based on the URL type. This button appears for embeddable pagelet, OBIEE pagelet, and PeopleSoft Query service types only.

Note: For embeddable and OBIEE pagelets, only context-sensitive parameters are supplied. For query services, only prompt fields are populated.

See Step 3: Specifying Pagelet Data Source Parameters.

Post mapping definition data

Select to have the mapping definition data, such as the parameter name and its mapping page field or key field name, and the parameter's actual mapped value posted with the service request. This option is deselected by default.

Escape URL Parameters

Select to escape, or encode, the service URL parameters and values when the application forms the complete service URL. The default value is selected.

Use when either service parameters or parameter values include special characters, such as the single quote (') and the percent sign (%).

Note: The application encodes only the portion of the URL that follows the equal (=) sign and excludes the sign itself.

Show Formed URL

Click to display the entire formed URL with test parameter values in a display-only edit box beneath the Service URL Parameter grid.

Note: Application class URLs cannot be formed by means of this link because the URL is dynamically generated by PeopleCode using data in the buffer.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Formed URL

This field appears only after you click the Show Formed URL link. The formed URL appears within a display-only edit box.

Test Related Content Service

Click to test the related content service in a new window. The Test Related Content Service page appears enabling you to enter a parameter value and test the functionality of the service.

Service URL Parameters

This table describes service URL parameter fields. These fields are common to all service URL types.

Parameter Name

Enter any additional service parameters for the service. For example, a service might require a search parameter.

You can map the parameter to a contextual field on an application page when you assign the service to the page.

See "Service Parameter Mapping" in Understanding Service Parameter, Filter, and Security Configuration.

Required

Select this check box when the additional parameter is necessary for a service to return a valid output. This check box is deselected by default.

For instance, when a service is launched, it might display default output if a null or blank value is passed to a parameter. If the service is designed with no default value and a blank value is passed to the parameter, an error or blank output could occur, depending on how the service is designed to handle this scenario. To avoid this situation, you can mark one or more additional service parameters as required so that the service will be invoked with a valid value for its additional service parameters.

Important! When this required check box is selected, the subject matter expert or administrator must map a field or value to the parameter on the Related Content - Map Fields page or the system does not allow the service assignment to be saved.

See "Service Parameter Mapping" in Understanding Service Parameter, Filter, and Security Configuration.

Description

Enter a parameter description that is meaningful for the subject matter expert or administrator who configures the related content assignment and mapping.

Display Options

This table describes display option fields. Display options apply only when the subject matter expert configures the service to appear in the Related Content frame. These fields are common to all service URL types.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Refresh	Select to enable the administrator to include the refresh icon in the related content frame; this icon enables the user to force a server trip to retrieve and refresh the data.
New Window	Select to enable the administrator to include the new window icon in the related content frame; this icon enables the user to send the service to a new browser window. When the service opens a new browser window, the service data in the new window is no longer actively linked to the parent transaction page. If the user searches for a different key value on the parent page, the content of the new window is *not* updated; however, you can continue to view and use it. This is important in situations in which you are viewing content and might want to continue using it after completing work in the transaction page.

Note: Any browser setting that enables the browser to determine whether to open pages in a new window or a tab can affect the expected functionality of the New Window property. Configure the *BNEWWIN* navigation personalization option to override any user-specific browser settings so that the browser never uses tabbed browsing for PeopleSoft applications.

See "Working with Personalization Options" (PeopleTools 8.54: Security Administration).

Select Security Options

This table describes security options.

Public Security	Select this option to provide access to the service based on the permissions of the public user.
Related Content Provider Security	Select this option to use the security implemented by the content provider system.
Related Content Consumer Security	Select this option to use the security implemented by the consuming system, which is the basic PeopleTools transaction security model (user-role-permission list).
App Class Required	Select this option to indicate that a service filter is required when this service is assigned to an application page in addition to any other security option that is selected for a specific service assignment. **Note:** The option can be selected as the only security option, or in addition to other security options.

Before you allow a security option for a service, you should understand the service's security model.

Important! If you allow a security option for a service definition that isn't applicable or hasn't actually been implemented by the service itself, unexpected results can occur.

See Understanding Service Security

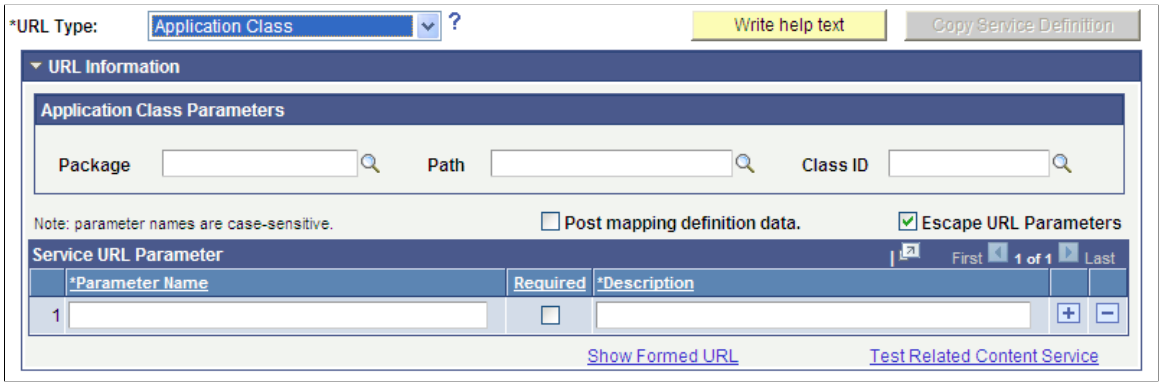
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating Application Class Service Definitions

Access the Define Related Content Service page. (Select PeopleTools, Portal, Related Content Service, Define Related Content Service) Select the *Application Class* URL type.

Image: Define Related Content Service page for an application class

This example illustrates some of the fields and controls on the Define Related Content Service page for an application class. You can find definitions for some of the fields and controls later on this page.



Application Class Parameters

Package Enter the name of the application class package that you want to use to define the related content service.

Path Enter the names of each subpackage in the application class hierarchy that define the location of the application class that defines the service. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Class ID Enter the name of the application class that defines the related content service. The class must exist in the application package and subpackage that you specify.

These fields are validated when you save the definition; the application does not allow you to save the service if you enter an invalid application class.

Service URL Parameters

Application class service definitions do not require service URL parameters except when you want to map field-level data to the service and display the service in a field-level contextual menu.

Application class URLs cannot be formed by clicking the Show Formed URL link because the URL is dynamically generated by PeopleCode using data in the buffer. For the same reason, application class URLs cannot be tested by clicking the Test Related Content Service link.

Application Class PeopleCode Example

An application class that will be used to define a related content service must import and extend the PT_RCF:ServiceInterface class. In addition, it must define a single method named execute.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

This is an example of application class PeopleCode that could be used to create an application class service for use as a related action:

```
import PT_RCF:ServiceInterface;


class res_det implements PT_RCF:ServiceInterface
   method execute();
end-class;

Global number &PRL_Schedule_Current_Row;

method execute
   /* Extends/implements PT_RCF:ServiceInterface.execute */

   Evaluate %Component
      When = Component.PGM_RESOURCE_LIST

/*****************************************************************
  Transfer to the Resource detail page.
*****************************************************************/
         If (%Page = Page.PC_PRL) Then
            GetPage(Page.PC_PRL_SCHED).Visible = True;
            &PRL_Schedule_Current_Row = CurrentRowNumber();

          /*AN: Hide the grids if the resource is not a resource class */
            If (GetRow().PROJECT_TEAM.TEAM_MEMBER_CLASS.Value <> "L") Then
               GetRow().GetRowset(Scroll.PROJ_TEAM_SCHED).HideAllRows();
               GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
            Else
               Local Rowset &ipRowSet = GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD);
               If (&ipRowSet.ActiveRowCount > 1 Or
                  (&ipRowSet.ActiveRowCount = 1 And
                     All(&ipRowSet(1).PC_TEAM_IP_SCHD.START_DT.Value))) Then
                  GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD).ShowAllRows();
               Else
                  GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
               End-If;
            End-If;

            Local Rowset &arlResDtlVw = GetRow().GetRowset(Scroll.PC_ARL_RESDTLVW);
            &arlResDtlVw.Flush();
            &arlResDtlVw.Select(Record.PC_ARL_RESDTLVW, "where BUSINESS_UNIT = :1 A⇒
ND PROJECT_ID = :2 AND TEAM_MEMBER = :3 and TEAM_MEMBER_CLASS = :4 and TEAM_MEMBER_⇒
IND = :5 ", PROJECT.BUSINESS_UNIT, PROJECT.PROJECT_ID, GetRow().PROJECT_TEAM.TEAM_M⇒
EMBER.Value, GetRow().PROJECT_TEAM.TEAM_MEMBER_CLASS.Value, GetRow().PROJECT_TEAM.T⇒
EAM_MEMBER_IND.Value);

            TransferPage(Page.PC_PRL_SCHED);

         End-If;

            /*AN: Hide the grids if the resource is not a resource class */

         If (&teamRow.PROJECT_TEAM.TEAM_MEMBER_CLASS.Value <> "L") Then
            &teamRow.GetRowset(Scroll.PROJ_TEAM_SCHED).HideAllRows();
            &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
         Else
            Local Rowset &ipSet = &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD);
            If (&ipSet.ActiveRowCount > 1 Or
               (&ipSet.ActiveRowCount = 1 And
                  All(&ipSet(1).PC_TEAM_IP_SCHD.START_DT.Value))) Then
                  &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD).ShowAllRows();
            Else
               &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
            End-If;
         End-If;

            TransferPage(Page.PC_PRL_SCHED);
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

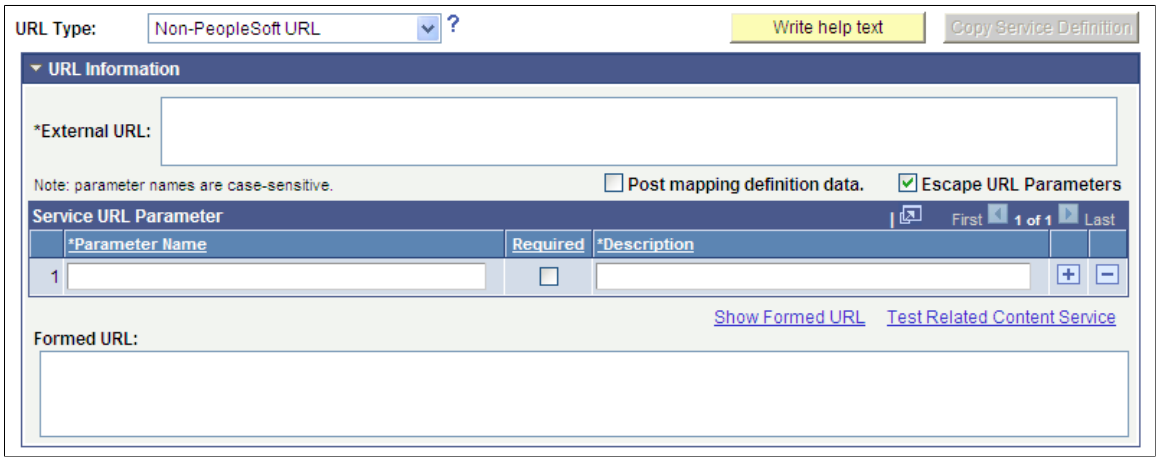
```
        End-If;

    End-Evaluate;

end-method;
```

Creating Non-PeopleSoft URL Service Definitions

Access the Define Related Content Service page. (Select PeopleTools, Portal, Related Content Service, Define Related Content Service.) Select the *Non-PeopleSoft URL* URL type.

Image: Define Related Content Service page for a non-PeopleSoft URL

This example illustrates some of the fields and controls on the Define Related Content Service page for a non-PeopleSoft URL. You can find definitions for some of the fields and controls later on this page.



URL Information

External URL Enter a URL external to the PeopleSoft application. For example: `http://search.oracle.com/search/search`

Service URL Parameters

Enter any parameters that the external service requires. For example: q

Non-PeopleSoft URL Example

This is an example of a fully formed non-PeopleSoft URL:

```
http://search.oracle.com/search/search?q=PeopleTools
```

Creating PeopleSoft Component Service Definitions

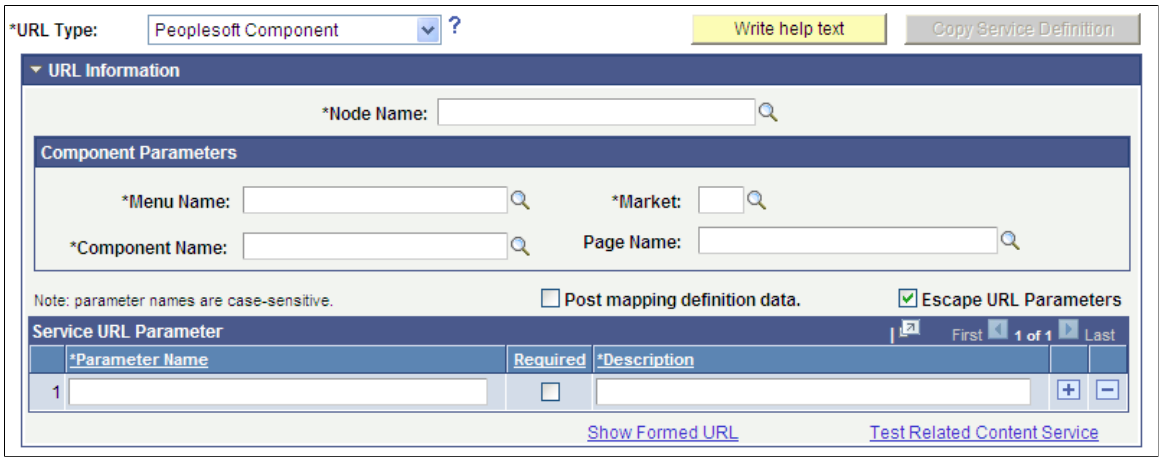
Access the Define Related Content Service page. (Select PeopleTools, Portal, Related Content Service, Define Related Content) Select the *PeopleSoft Component* URL type.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: PeopleSoft components can implicitly act as anonymous services. You must define this type of service explicitly in certain circumstances. See Explicit Definition Versus Anonymous Services.

Image: Define Related Content Service page for a PeopleSoft component

This example illustrates some of the fields and controls on the Define Related Content Service page for a PeopleSoft component. You can find definitions for some of the fields and controls later on this page.



Component Parameters

Menu Name Enter the definition name of the PeopleSoft menu on which the component is located as configured in PeopleSoft Application Designer.

Market Enter the market, such as *GBL,* of the component as defined in PeopleSoft Application Designer.

Component Name Enter the component definition as defined in PeopleSoft Application Designer.

Page Name Enter the page definition as defined in PeopleSoft Application Designer. Use this field to open a specific page in the component when it appears in the related content target area.

PeopleSoft Component URL Example

This is an example of a fully formed PeopleSoft component URL:

```
http://myserver.example.com/psp/ps/EMPLOYEE/PT_LOCAL/c/MAINTAIN_SECURITY.USERMAINT.⇒
GBL?OPRID=PTPORTAL
```

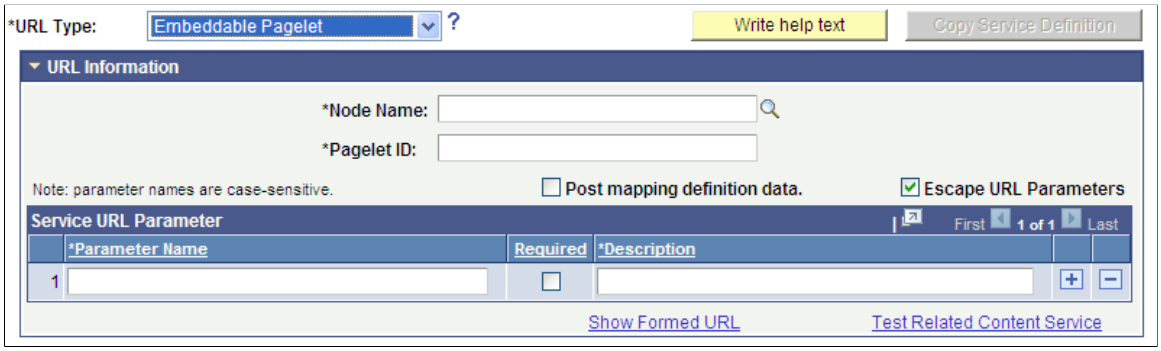
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating Embeddable Pagelet Service Definitions

Access the Define Related Content Service page. (Select PeopleTools, Portal, Related Content Service, Define Related Content) Select the *Embeddable Pagelet* URL type.

Image: Define Related Content Service page for an embeddable pagelet

This example illustrates some of the fields and controls on the Define Related Content Service page for an embeddable pagelet. You can find definitions for some of the fields and controls later on this page.



URL Information

Pagelet ID

Enter the unique identifier of the embeddable pagelet created in Pagelet Wizard.

See Step 6: Specifying Pagelet Publishing Options.

Service URL Parameters

Click the Populate parameters button to populate the Service URL Parameter grid with parameters that you defined as *context sensitive* in step three of the pagelet construction in Pagelet Wizard.

If the pagelet resides in the same system, as soon as you select the node, a lookup appears next to the pagelet ID field. When clicked, the lookup lists all of the available embeddable pagelets in the system.

See Step 3: Specifying Pagelet Data Source Parameters.

Embeddable Pagelet URL Example

This is an example of a fully formed embeddable pagelet URL:

```
http://myserver.example.com/psc/ps/EMPLOYEE/PT_LOCAL/s/WEBLIB_PTEMBED.ISCRIPT1.Fiel⇒
dFormula.IScript_EmbeddedPageletBuilder?PAGELETID=USER_ROLES&OPRID=PTDMO&.MAXROWS=1⇒
0
```

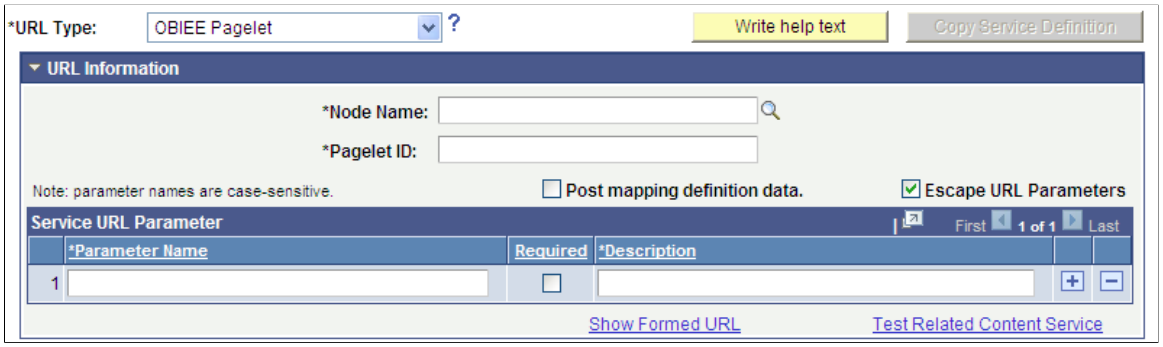
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating OBIEE Pagelet Service Definitions

Access the Define Related Content Service page. (Select PeopleTools, Portal, Related Content Service, Define Related Content) Select the *OBIEE Pagelet* URL type.

Image: Define Related Content Service page for an OBIEE pagelet

This example illustrates some of the fields and controls on the Define Related Content Service page for an OBIEE pagelet. You can find definitions for some of the fields and controls later on this page.



URL Information

Pagelet ID

Enter the unique identifier of the OBIEE pagelet created in Pagelet Wizard.

Service URL Parameters

Click the Populate parameters button to populate the Service URL Parameter grid with parameters that you defined as *context sensitive* in step three of the pagelet construction in Pagelet Wizard.

If the pagelet resides in the same system, then as soon as you select the node, a lookup appears next to the pagelet ID field. When clicked, the lookup lists all of the available OBIEE pagelets in the system.

The Populate parameters button supplies the parameters only for the locally available pagelets. If a pagelet resides in a remote PeopleSoft system, you must manually enter the pagelet ID and pagelet parameters.

See Step 3: Specifying Pagelet Data Source Parameters.

OBIEE Pagelet URL Example

This is an example of a fully formed OBIEE pagelet URL:

```
http://myserver.example.com/psc/ps/EMPLOYEE/PT_LOCAL/s/WEBLIB_PTEMBED.ISCRIPT1.Fiel⇒
dFormula.IScript_EmbeddedPageletBuilder?PAGELETID=OBIEE_REPORT&1) Markets.Region=EA⇒
STERN REGION
```

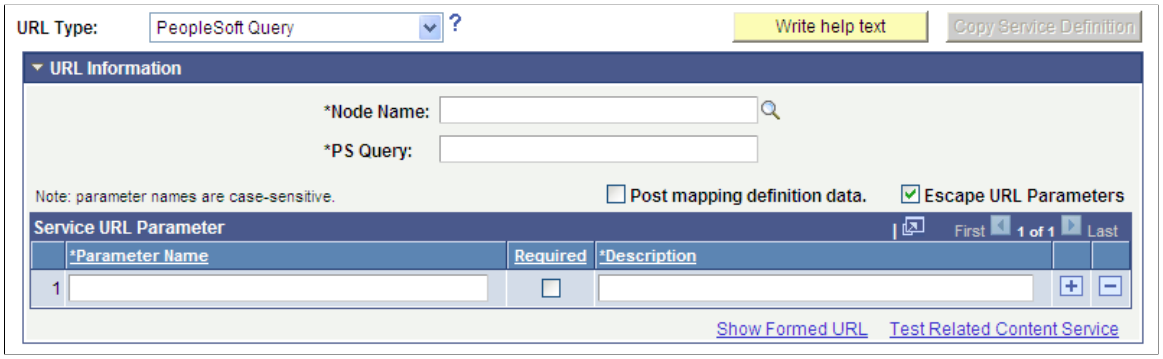

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating PeopleSoft Query Service Definitions

Access the Define Related Content Service page. (Select PeopleTools, Portal, Related Content Service, Define Related Content.) Select the *PeopleSoft Query* URL type.

Image: Define Related Content Service page for a PeopleSoft query

This example illustrates some of the fields and controls on the Define Related Content Service page for a PeopleSoft Query. You can find definitions for some of the fields and controls later on this page.



URL Information

PS Query Enter the query definition name as defined in Query Manager.

Service URL Parameters

Click the Populate parameters button to populate the Service URL Parameter grid with fields that you defined as *prompts* when creating the query definition.

If the query resides in the same system, as soon as you select the node, a lookup appears next to the PS Query field. When clicked, the lookup lists all of the available queries in the system.

The Populate parameters button supplies the parameters only for the locally available queries. If a query resides in a remote PeopleSoft system, you must manually enter the query name and query parameters.

See "Defining Prompts" (PeopleTools 8.54: Query).

PeopleSoft Query URL Example

This is an example of a fully formed PeopleSoft query URL:

```
http://myserver.example.com/psc/ps_newwin/EMPLOYEE/PT_LOCAL/q/?ICAction=ICQryNameUR⇒
L=PUBLIC.MESSAGES_FOR_MSGSET&MESSAGE_SET_NBR=240
```

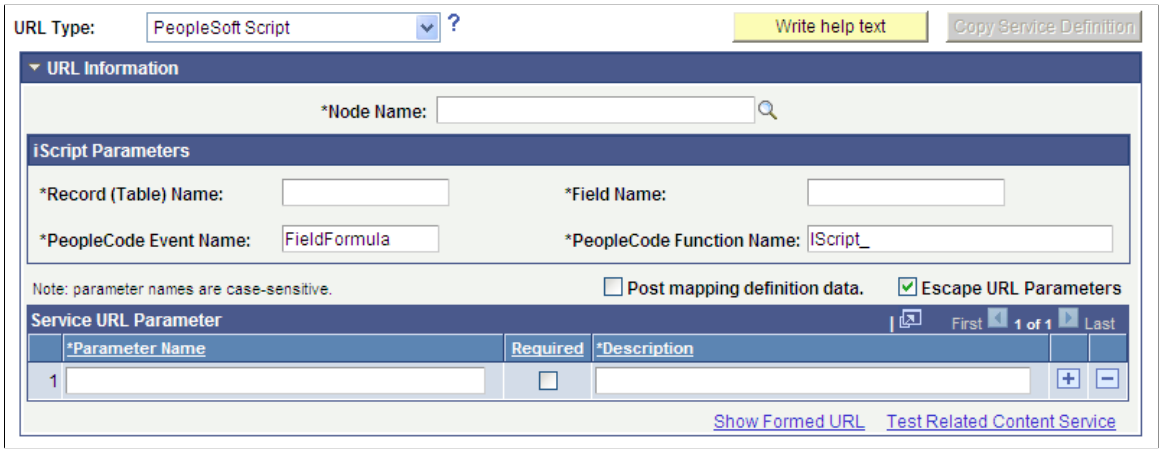
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating PeopleSoft Script Service Definitions

Access the Define Related Content Service page. (Select PeopleTools, Portal, Related Content Service, Define Related Content.) Select the *PeopleSoft Script* URL type.

Image: Define Related Content Service page for a PeopleSoft script (iScript)

This example illustrates some of the fields and controls on the Define Related Content Service page for a PeopleSoft Script (iScript). You can find definitions for some of the fields and controls later on this page.



iScript Parameters

Record (Table) Name

Enter the name of the WEBLIB_ record where the iScript is stored.

See "Web Libraries" (PeopleTools 8.54: PeopleCode API Reference).

Field Name

Enter the name of the field where the iScript is stored.

PeopleCode Event Name

Enter the name of the PeopleCode event under which the iScript is stored.

The default event is FieldFormula.

PeopleCode Function Name

Enter the name of the function that performs the logic of the service.

Service URL Parameter

Enter the names of all parameters that the iScript must have to render the service in the related content service frame.

PeopleSoft Script URL Example

This is an example of a fully formed PeopleSoft script URL:

```
http://myserver.example.com/psc/ps/EMPLOYEE/PT_LOCAL/s/WEBLIB_STOCK.ISCRIPT1.FieldF⇒
ormula.IScript_StockQuote?stockquote=ORCL
```


Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Testing Related Content Service Definitions

Access the Test Related Content Service page. (On the Define Related Content Service page, click the Test Related Content Service link.)

Image: Test Related Content Service page

This example illustrates the fields and controls on the Test Related Content Service page. You can find definitions for the fields and controls later on this page.

Parameter Name	Displays the parameters that you enter on the Define Related Content Service page. This field is display-only.
Parameter Value	Enter a test value for the parameter.
Test Related Content Service	Click to test the related content service with the parameter values that you enter.
Return	Click to return to the Define Related Content Service page after testing the service.

Testing Related Content Service Parameter Values

To test parameter values:

1. Enter the value in the Parameter Value field.
2. Click the Test Related Content Service link.

   The service appears in a separate browser window using the specified parameter values.
3. Click the Return button to return to the Define Related Content Service page.

Selecting Bulk Action for Related Content Service Definitions

When you define related content services, you can choose whether a related action should be performed on multiple rows of data or on a single row of data.

Selecting the Bulk Action check box enables you to perform a related action on multiple rows of data.

The Bulk Action option is available for the following service types:

- Application Class

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- PeopleSoft Script
- PeopleSoft Component

When you select the Bulk Action check box, only the supported service types are displayed in the URL Type drop down. If you select a service type that is not supported by the Bulk Action option, and then select the Bulk Action check box, a message displays that the selected service type is not supported. When you click OK, the Bulk Action check box is no longer selected.

Component Level Related Action

At the component level, you can associate bulk action with a service. The Look Up Service ID page and the Configure Service page reflect the selection of bulk action.

Page Level Related Action

At the page level, bulk action is available for a service that is linked to related action widget. Bulk action is not available for a glyph. The Related Action Look Up page and the Configure Service page reflect the selection of bulk action.

Note: If bulk action is mapped to a glyph field, the related action is applicable for a single row of data; not multiple rows.

Limitations

Bulk action functionality is not supported:

- In the Related Content frame.
- For the following service types:
  - Pagelets
  - PS Query
  - Non-PeopleSoft URL

Bulk Action Availability for Authorized/Unauthorized Rows

The bulk action functionality is available or not available in the following scenarios:

Scenario	***Bulk Action Availability***
User is authorized for all rows of a transaction.	Yes
User is not authorized for any of the rows of a transaction.	No
User is authorized for partial number of rows of a transaction.	Yes Only the authorized rows are processed and a message displays the number of processed rows.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Writing Help Text for Related Content Service Definitions

When you define related content services, you should document the service and how the service parameters should be mapped to transaction content, especially if your service uses multiple, complex service parameters. The components and pages can include numerous fields, some of which are named similarly, which can make mapping the service parameters difficult.

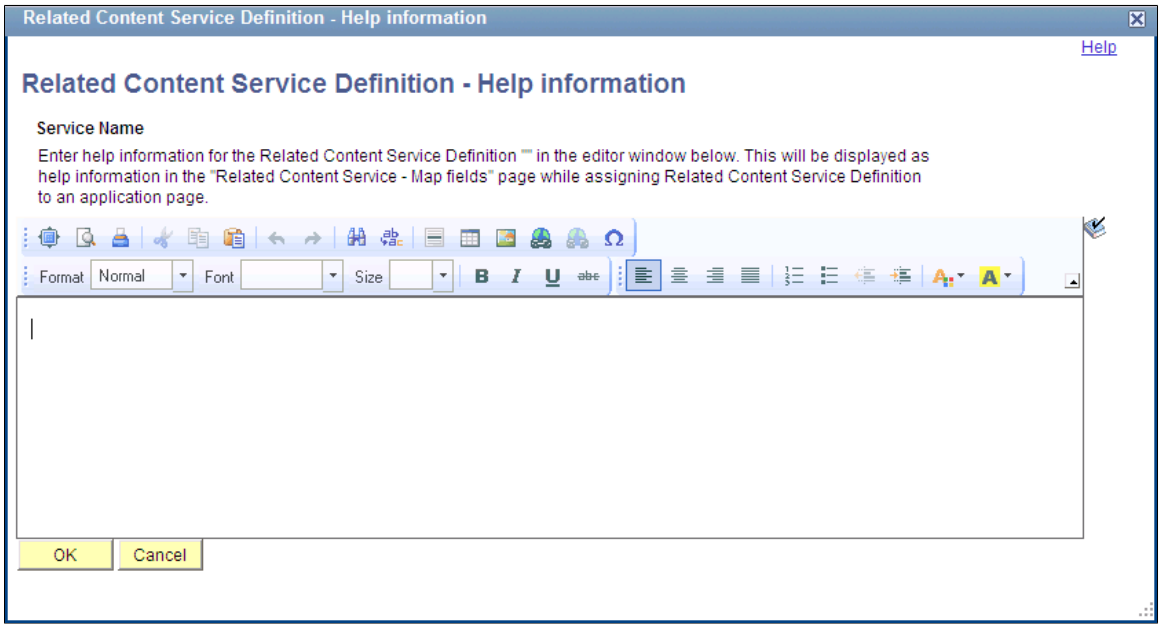
Use the help text to clearly communicate the values that must be assigned to the service parameters. The text that you enter is accessible to the subject matter expert when she clicks the Read help text link on the Related Content Service - Configure Service page.

Important! You must use explicit, concise, clear, understandable language in your instructions.

Access the Related Content Service Definition - Help information page. (On the Define Related Content Service page, click the Write help text button.)

Image: Related Content Service Definition - Help information page

This example illustrates the fields and controls on the Related Content Service Definition - Help information page. You can find definitions for the fields and controls later on this page.



Service Name	Displays the name of the related content service. This field is display-only.
OK	Click to save the text and return to the Related Content Service Definition page. **Note:** You must save the service definition to save the help text.
Cancel	Click to discard any changes and return to the Related Content Service Definition page.

To write help text for a related content service definition:

1. Enter help information for the related content service definition in the editor window.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. Click the OK button.

   This returns you to the Related Content Service Definition page.

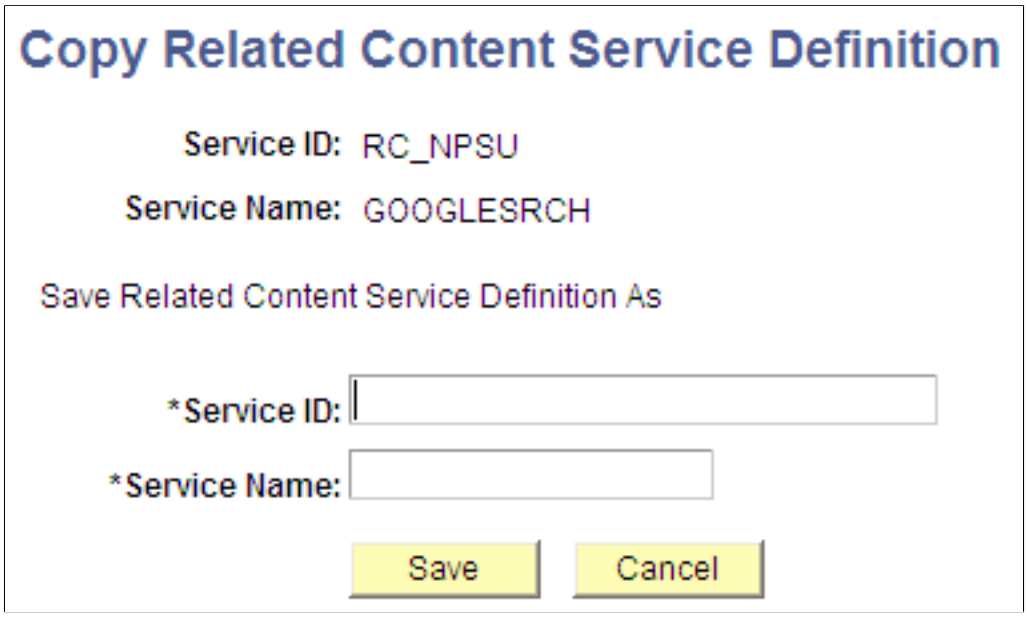
3. Save the definition.

Copying Related Content Service Definitions

Access the Copy Related Content Service Definition page. (On the Define Related Content Service page, click the Copy Service Definition button.)

Image: Copy Related Content Service Definition page

This example illustrates the fields and controls on the Copy Related Content Service Definition page. You can find definitions for the fields and controls later on this page.



To copy a related content service definition:

1. Enter a new value in the Service ID field.
2. Enter a new value in the Service Name field.
3. Click the Save button.

   A message appears indicating a successful save.

4. Click the OK button to dismiss the message box.
5. Click the Go to Related Content Service Definition link that appears to access the new related content service definition or click the Cancel button to close the dialog box.

Upgrading or Copying Related Content Data

Upgrading Related Content Data from PeopleTools 8.51 or PeopleTools 8.50

Use the Upgrade Related Content page to complete the import and upgrade of related content data from PeopleTools 8.51 or PeopleTools 8.50.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To upgrade related content data from PeopleTools 8.51 or PeopleTools 8.50:

1. On the system on the prior release, use PeopleSoft Data Mover to export the related content data.
2. On your system on the current release, use PeopleSoft Data Mover to import the related content data.
3. Access the Upgrade Related Content page (PeopleTools, Portal, Related Content Service, Upgrade Related Content).
4. Click the Run button to upgrade the imported related content data.

Copying Related Content Data

You can use PeopleSoft Application Designer to create upgrade projects that contain these types of related content data:

- Related Content Definitions (that is, services defined on the Define Related Content Service page)

  Also select the Application Package related definition when the service type of the selected related content service is *Application Class.*

- Related Content Services (that is, service assignments made through the Manage Related Content Service menu item).
  - *Always* select the Related Content Configurations related definitions.
  - Also select the Related Content Layouts related definitions when the service was added to or deleted from a menu, or when the menu order was changed.

You can save these projects to files or copy them directly to other PeopleSoft databases.

See "Inserting Definitions Into Projects" (PeopleTools 8.54: Application Designer Developer's Guide).

Upgrading Related Content Data Using ADS Definitions

Using ADS definitions for upgrading or migrating Related Content Framework data, PeopleSoft enables you to preserve configurations and personalizations that are effected on a target database. Related Content Framework leverages the ADS relationship framework and ADS merge framework to preserve configurations and personalizations on a target database.

Related Content Framework identifies configurable changes that are preserved during upgrade and non-reconcilable changes that are not preserved during upgrade. The configurable and non-reconcilable changes are associated with database tables and its fields that contain related content service definitions and related content services data.

You can use Data Migration Workbench to migrate related content data. See Data Migration Workbench.

PeopleSoft recommends that you:

- Do not delete any PeopleSoft delivered objects.
- Restrict your configurations to configurable fields specified for each table because these configurations are preserved.
- Avoid modification to any field that is in the non-reconcilable section because any modification to these fields is considered to be a customization and will not be restored.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The following table lists the configurable and non-reconcilable changes associated with related content service definitions. The database tables and its configurable and non-reconcilable fields associated with related content services are described later.

Configurable Changes	***Non-reconcilable Changes***
Modification to service name of a definition.	Changing the URL Type of a Service Definition
Customer modifies the Node name to match the environment	Deleting a required Parameter of a Service Definition
Modification to Service Definition's Description and Help text	Adding a required Parameter of a Service Definition
Modification to the Display Options: Refresh and/or New Window	Modifying the following options: • Escape URL parameters • Post Mapping Definition • Security Options
Modifying the Parameter's Description	Modifying a Parameter Name

The following table lists the configurable and non-reconcilable changes associated with related content services.

Configurable Changes	***Non-reconcilable Changes***
Modification of 'Layout Options' (Open in Expanded view)	Deleting a CREF association.
Modification of the 'Related Content Location' of Related Content Frame	Deleting an existing configuration from a CREF.
Modification of the Service Label of related Content/Action	Mapping type of parameter of a configuration (Key field, fixed etc).
Making/ removing the default service of an Related Content	Changing the Page Field Menu option.
Modifying the Service Target of an Related action	Deleting a PeopleSoft delivered folder.
Enabling or Disabling a Related Content / Action	
Modifying the values of Service Filter (Package, Path, Class ID) App Package	
Modifying the options of the Related Content Frame to be minimized when it is displayed.	
Modifying the 'Display in Field Menu' option for Related Action	
Modifying 'Refresh Service On Change' Option	
Modifying 'Is Value Required?' option	
Customer reorders the sequences of the folder.	

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The following sections list the database tables and its configurable and non-reconcilable fields.

PSPTCSSRVDEFN (Service Definition)

This table holds the service definition data.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	*Comment*
PTCS_SERVICENAME	Service Name
DESCR254	Description
MSGNODENAME	Node Name
PTCS_SRVHELPTEXT	Help Text
USEEDIT	Display Options
PORTAL_URL_ADM	URL (for URL Type == Non PeopleSoft URL)

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

Field	*Comment*
PTCS_SERVICEURLTYP	URL Type
PTCS_ESCAPEPARAM	Escape URL parameters option
PTCS_POST_MAPDATA	Post Mapping Definition Data option
PTCS_SECUSEEDIT	Security options

PSPTCS_PARAMS (Parameters table)

This is the child table of the service definition table, which holds the parameters details defined for the service definition. This is a metadata table and hence the scope for secondary merge is very limited. PeopleSoft recommends that users do not delete any of the delivered parameters, update the parameter name and modify the required flag options.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	*Comment*
PTCS_DESCR128	Description
SEQNUM	Parameter’s order of definition

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

Field	*Comment*
PTCS_PARAMETERNAME	Parameter Name
REQUIRED_FLG	Required Flag

PSPTCS_SRVCFG

This table stores the top-level related content metadata for all the content references that have related content associated with them. It also stores the metadata of how the related content frame is displayed (Right Frame or Bottom Frame).

Any changes, other than the ones specified in the following table are non-reconcilable.

Field	*Comment*
PTCS_EXPANDFRAME	Open in Expanded View Options
PTCS_RCFRAMELOC	Related Content Frame Location

PSPTCSSRVCONF (Service Configurations)

This is the actual table that stores the relationship between instances of service definitions and content references.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	*Comment*
IS_DEFAULT	Default service
PTCS_INSTANCELBL	Service Label
PTCS_URLTARGET	Service Target
PACKAGEROOT	Service Filter – Package
QUALIFYPATH	Service Filter – Path
APPCLASSID	Service Filter – Class ID
PTCS_INITMIN	RC Frame to be open minimized
PTCS_ENABLE	Enable/Disable Related Content/Actions

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Field	Comment
SEQNUM	Customer cannot recreate this configuration as the instance ID will be different.
PTCS_MAPFIELDTYPE	Mapping Type of the parameter
PTCS_FIELDMENUOPT	Applicable for Related Actions as this is the field to which the glyph is associated.

PSPTCS_MAPFLDS

This is a child table of PSPTCSSRVCONF and stores the field mappings.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	Comment
PTCS_FIELDMENUOPT	Display In Field Menu option. Applicable for related Actions
PTCSREFRESHSERVICE	Refresh Service on Change option. Applicable for Related Content.
PTCS_MAPNULLVALUE	Is Value Required Option

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

Field	Comment
PTCS_MAPFIELDTYPE	Mapping Type
PTCS_PNLFLDNAME	Page Field Menu Option

PSPTCS_MNUFLDRS (Folders)

This table stores the data of the folders and their sequencing. A maximum of one child folder for a non root folder is allowed. From the user interface, a user can add or delete folders, but cannot rename a folder. Also, a user can reorder the folder sequence by using the Move Up and Move Down options.

Note: Deleting a delivered folder is a non-reconcilable change.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	Comment
PTCS_FOLDERSEQNUM	Sequence/ ordering of folder

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

ADS Definitions for Related Content Data

PeopleSoft delivers the following ADS definitions for related content data:

ADS Definition	*Description*
RCF_SERVICE_DEFINITIONS (RC Service Definition)	Defines the related content services that can be any collaborative, analytical, or informational content, which is useful for completing business tasks.
RCF_SERVICES (RC Services)	Encapsulates the content reference for which the services are assigned along with the configuration data (Related Content Configuration) and how they are displayed in the related content frame (Related Content Layout). This ADS dataset should be used for shipping Related Content Framework service changes from PeopleTools 8.54 onwards. In PeopleTools 8.53, this ADS dataset was recommended for delivering new services only. In PeopleTools 8.54, this restriction is not valid, and this dataset can be used for any update or new service.

Related Links

"Defining Data Migration Project" (PeopleTools 8.54: Data Migration Workbench)
"Defining Data Set Content" (PeopleTools 8.54: Data Migration Workbench)
"Inserting Data Content" (PeopleTools 8.54: Data Migration Workbench)
"Copy Compare Overview" (PeopleTools 8.54: Data Migration Workbench)

Managing Related Content

This section provides an overview of related content assignment, and discusses how to:

- Manage related content for content references.
- Select an application page (content reference).
- Manage related actions for pivot grids.
- Select a pivot grid.

Understanding Assignment of Related Content and Related Actions

The Related Content Framework is designed so that subject matter experts or business analysts with an understanding of PeopleSoft applications and your business processes can assign related content services to components with little or no assistance from the developer after the service is defined. Almost every aspect of assigning and configuring related content services is done in the browser; no access to developer tools, such as PeopleSoft Application Designer, is necessary.

The Manage Related Content Configuration page is the starting point for creating and assigning related content services. The first time you access this page, you will see that no application pages have been assigned related content. After you assign related content services to application pages, a grid appears listing the related content assignments.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

When you design and configure related information correctly, the application user is presented with seamless access to related information, which can increase productivity and improve job performance.

Process Flow for Assigning Related Content and Related Actions

Related content assignment follows this basic process flow:

To assign related content or related actions to an application page or to assign related actions to a pivot grid, you will:

1. Select the application page (content reference) or pivot grid.

   See Selecting an Application Page (Content Reference) and Selecting a Pivot Grid.

2. Determine whether the related content service is to be assigned as related content or as a related action. For pivot grids, related content can be assigned as related actions only.

   See Related Content Versus Related Actions.

3. Determine whether the related content service is to be assigned at the component or page field level.

   See Component Level Versus Page Level Assignment.

4. Select the specific service, content reference, or pagelet to use as the related content service.

   See Assigning Related Content and Related Actions

5. Configure the related content service.

   Configuring Service Parameters, Filters, and Security

6. Configure the menu and layout of the related content service.

   See Configuring Menus and Configuring the Related Content Frame.

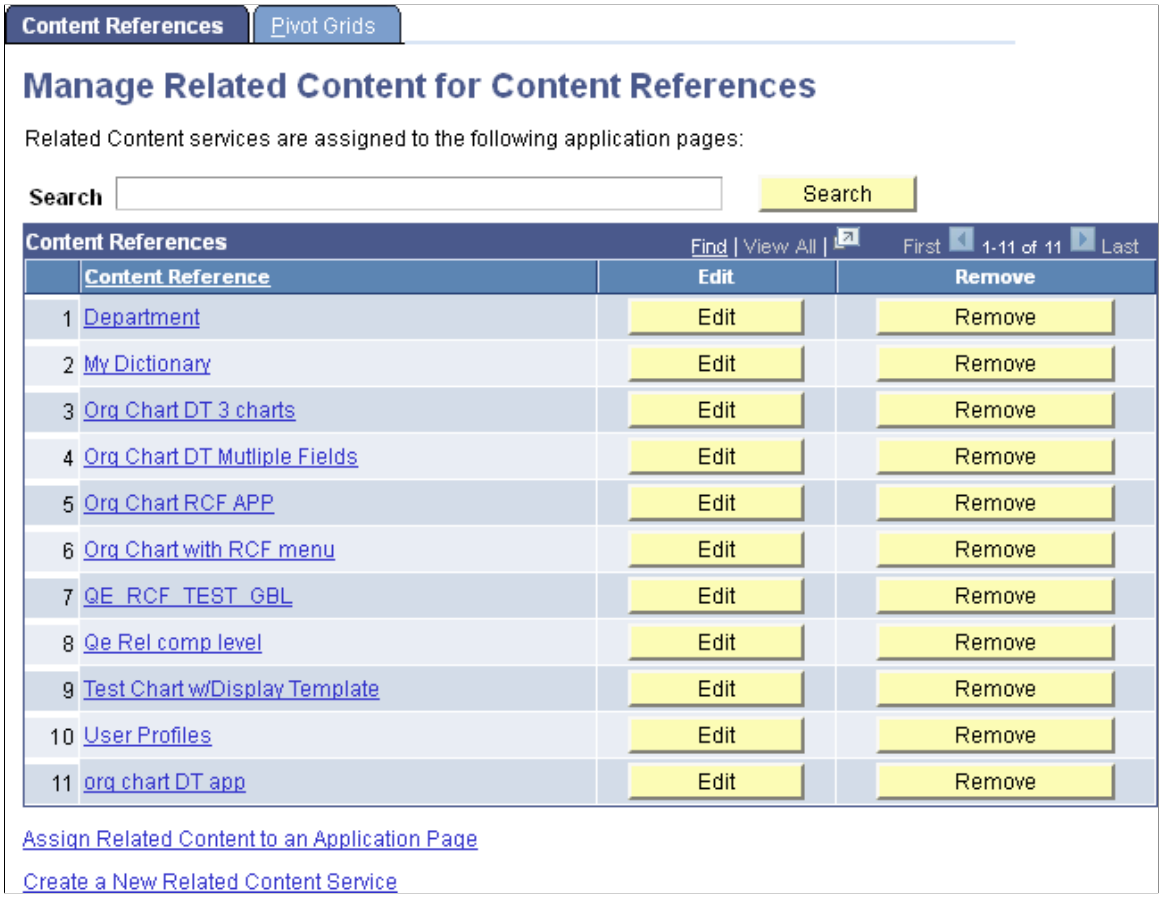
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Managing Related Content for Content References

Access the Manage Related Content for Content References page. (Select PeopleTools, Portal, Related Content Service, Manage Related Content Service.)

Image: Manage Related Content for Content References page

This example illustrates the fields and controls on the Manage Related Content for Content References page. You can find definitions for the fields and controls later on this page.



Search

Enter a search term or phrase and click the Search button to filter the list of content references using the search terms.

Search terms are case insensitive and results are returned for partial word matches. For example, both *dep* and *dept* will match the Department (DEPT_TBL.GBL) content reference. When you click the Search button, a database search is performed on the following values:

- Content reference name
- Content reference label
- Values stored in the KEYWORD content reference attribute.

See “Content Reference Attributes” in Defining Content References.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Content Reference	Click a content reference link to access the actual transaction page, which allows you to test any configured related content.
Edit	Click an Edit button to edit an existing related content configuration. See Assigning Related Content and Related Actions.
Remove	Click a Remove button to remove an existing related content configuration.
Assign Related Content to an Application Page	Click this link to access the Select a Content Reference page to create a new related content assignment. See Selecting an Application Page (Content Reference).
Create a New Related Content Service	Click the link to access the Define Related Content Service page. See Defining Related Content Services.

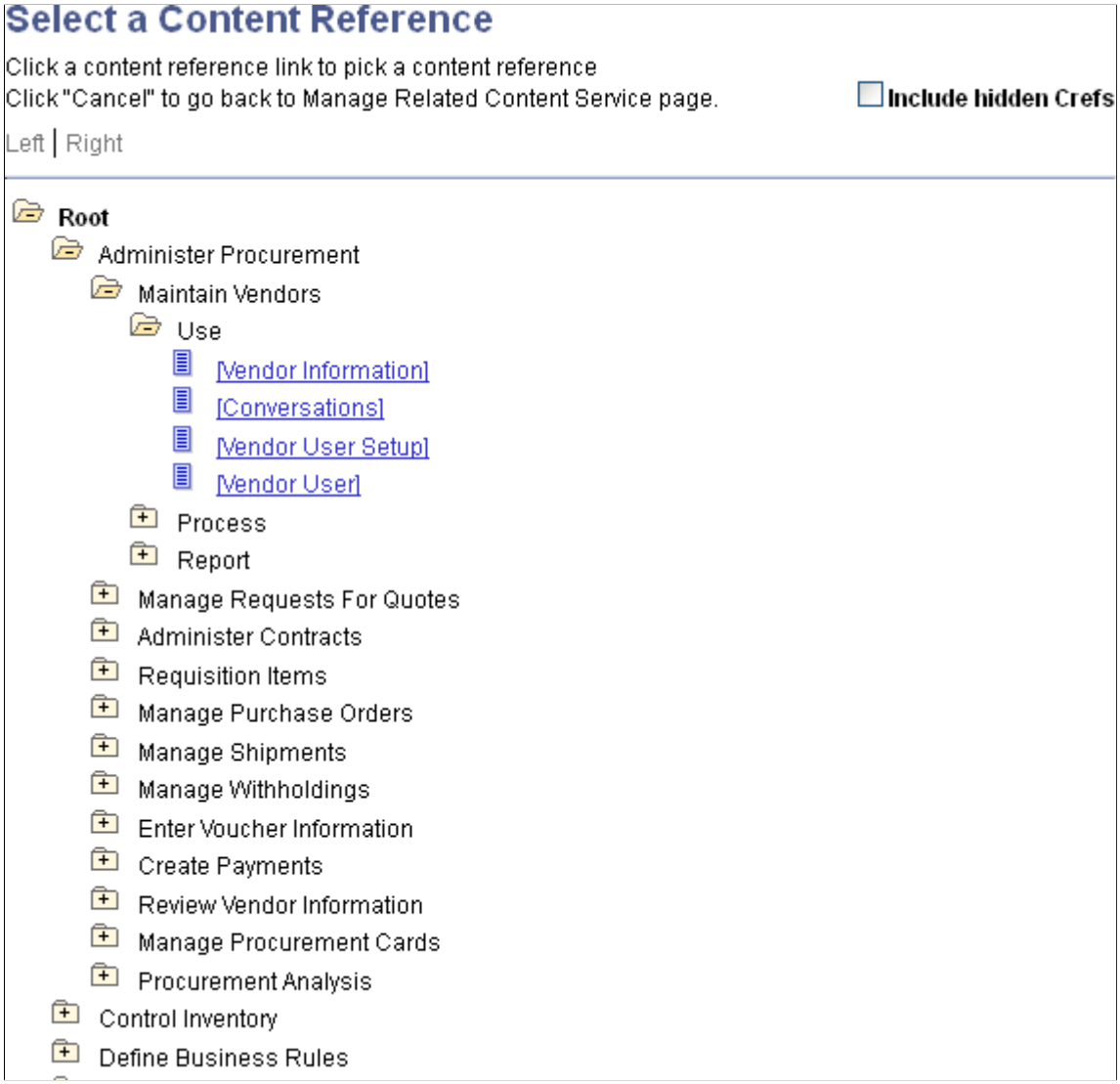
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting an Application Page (Content Reference)

Access the Select a Content Reference page. (On the Manage Related Content for Content References page, click the Assign Related Content to an Application Page link.)

Image: Select a Content Reference page

This example illustrates the fields and controls on the Select a Content Reference page. You can find definitions for the fields and controls later on this page.



Include hidden Crefs Select to make hidden content references available for assignment.

Note: The tree shows both PeopleSoft component URLs and PeopleSoft component-based generic URLs.

Click to expand the folder and display child folders and links.

Click to collapse the folder and hide all children.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

[link] Click a content reference link to select that content reference to which related content will be assigned.

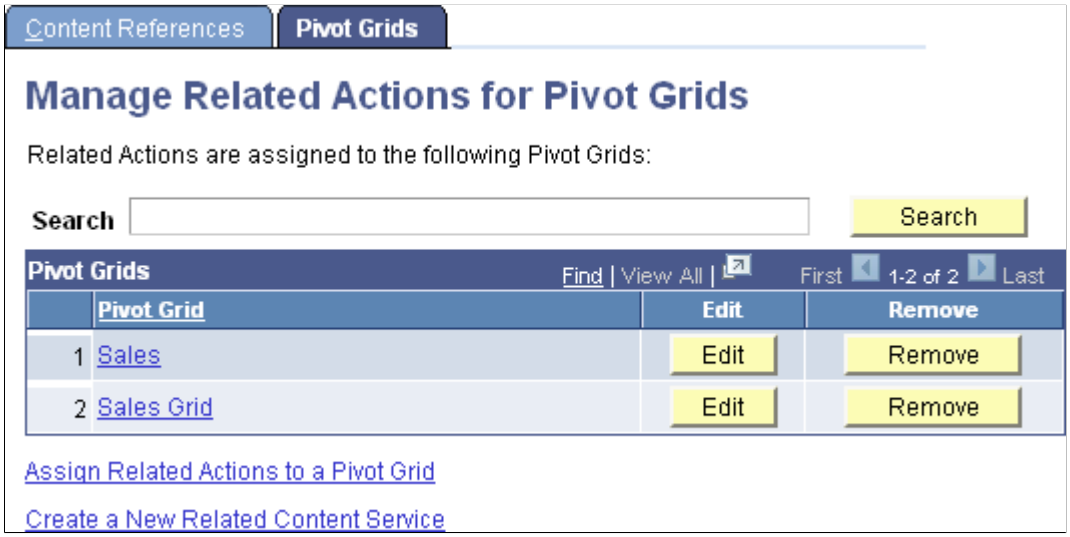
After selecting a content reference, continue by assigning and configuring related content or related actions. See Assigning Related Content and Related Actions.

Managing Related Actions for Pivot Grids

Access the Manage Related Actions for Pivot Grids page. (On the Manage Related Content for Content References page, select the Pivot Grids tab.)

Image: Manage Related Actions for Pivot Grids page

This example illustrates the fields and controls on the Manage Related Actions for Pivot Grids page. You can find definitions for the fields and controls later on this page.



Search Enter a search term or phrase and click the Search button to filter the list of pivot grids using the search terms.

Search terms are case insensitive and results are returned for partial word matches. For example, both *gri* and *pv_gr* will match the Sales Grid (PV_GRID) pivot grid. When you click the Search button, a database search is performed on the following values:

- Pivot grid name
- Pivot grid title

Pivot Grid Click a pivot grid link to open this pivot grid in the Pivot Grid Viewer, which allows you to test any configured related actions.

Edit Click an Edit button to edit an existing related actions configuration.

See Assigning Related Content and Related Actions.

Remove Click a Remove button to remove an existing related actions configuration.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Assign Related Actions to a Pivot Grid	Click this link to select a pivot grid for a new related actions assignment. See Selecting a Pivot Grid.
Create a New Related Content Service	Click the link to access the Define Related Content Service page. See Defining Related Content Services.

Selecting a Pivot Grid

To select a pivot grid for a new related actions assignment:

1. Click the Assign Related Actions to a Pivot Grid link on the Manage Related Actions for Pivot Grids page.
2. On the Add a New Value tab, click the Look up Pivot Grid button.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

3. Select a pivot grid on the Look Up Pivot Grid page:

   Image: Look Up Pivot Grid page

   Search and select a pivot grid:

4. Click the Add button to begin assigning and configuring related actions to the selected pivot grid.

5. See Assigning Related Content and Related Actions.

Configuring Related Content, Related Actions, and Layout

This section provides overviews of:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Related content versus related actions.
- Related actions and SmartNavigation
- Component level versus page level assignment
- Service parameter, filter, and security configuration.
- Menu configuration.
- Related Content frame configuration.
- PeopleCode FieldChange event based related actions configuration.

Then, this section discusses how to:

- Assign related content and related actions.
- Configure service parameters, filters, and security.
- Configure menus.
- Copying menu layouts.
- Configure the Related Content frame.
- Configure PeopleCode FieldChange event based related actions.

Prerequisites

This table discusses the prerequisites for assigning related content.

Prerequisite	*Cross Reference*
For the Service service type, explicit service definitions must exist before you can assign these related content services to a component, page field, or pivot grid as related content or as related actions.	See Defining Related Content Services.
For the Content Reference and Pagelet service types, which are accessed as anonymous services, an explicit service definition is not required. However, in certain circumstances, you might require an explicit service definition.	See Explicit Definition Versus Anonymous Services.
For pagelets that require parameters, you must enter the RCF_SERVICE_PARAMETER attribute.	See Understanding Service URL Parameters.

Related Content Versus Related Actions

The PeopleSoft Related Content Framework allows related content services to be deployed in two ways—as related content and as related actions—based on where the content appears when the user accesses the service.

- *Related content* appears in the Related Content frame only, which can be selected to be displayed at the bottom or to the right of the transaction content.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Related actions* appear outside of the Related Content frame in a modal window, in a new browser window, in the existing browser window replacing the current transaction content, or in the existing browser window replacing the entire page.

Configure the related content service as *related content* by selecting the Configure Related Content page:

- If you want the related content to appear in a separate frame beside or below the transaction content.
- If you want the related content to be dynamically updated when the user moves to a different transaction instance, or changes a field value, and so on.

Configure the related content service as a *related action* by selecting the Configure Related Actions page:

- If you want the related content to appear in a modal window, in a new browser window, or to replace the current page.
- When you are configuring related content for pivot grids, keyword search results, SmartNavigation menus, or SmartNavigation pages.

Understanding Related Actions and SmartNavigation

Finding the Details Component for Tree-Based SmartNavigation Pages

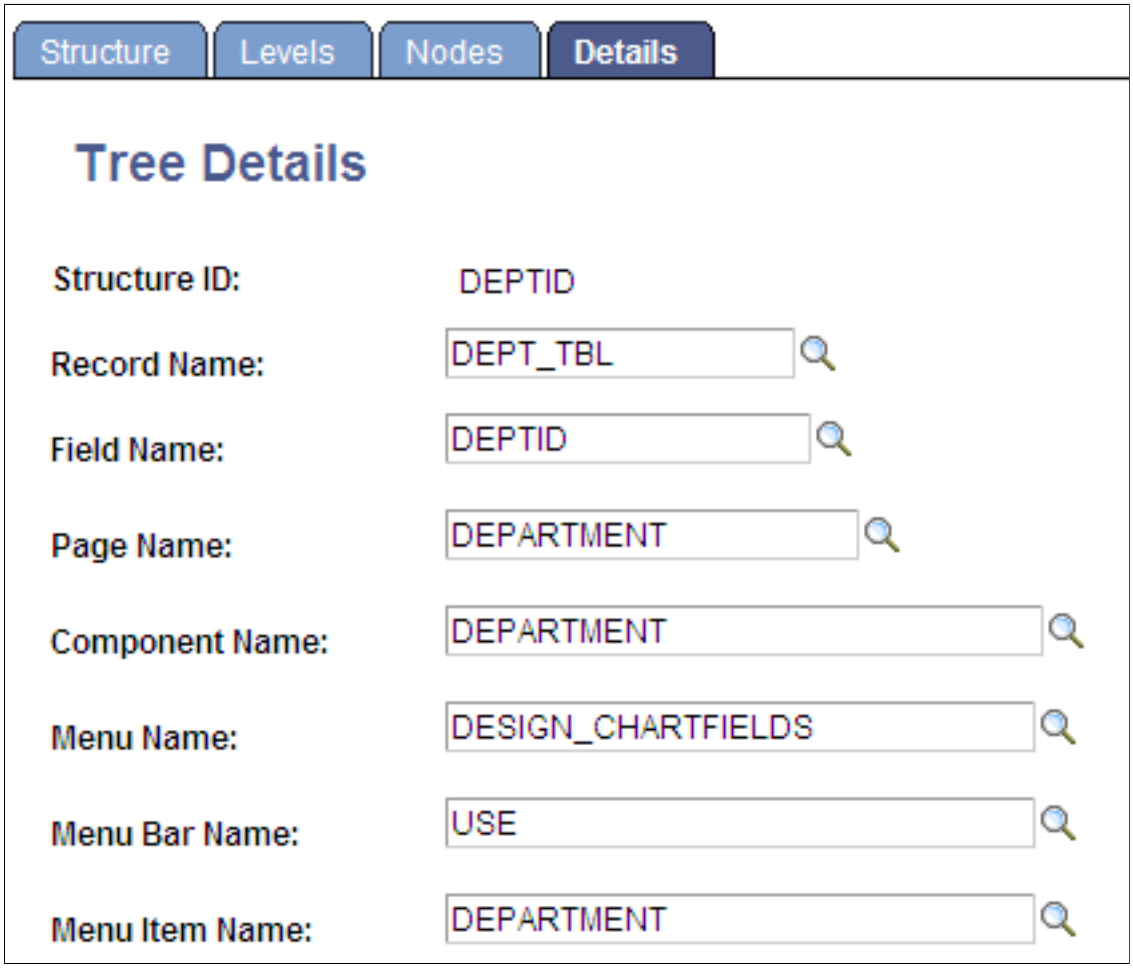
Trees are hierarchical collections of data that you create and maintain by using PeopleSoft Tree Manager. Tree-based SmartNavigation folders use a specific type of tree called a detail tree as the data source. When you use PeopleSoft Tree Manager to configure a detail tree structure, you define the location of the component that you use to maintain the detail values in the tree. When you assign a related content service that you want to appear as a related action, you assign the service to the detail component, which underlies the nodes and leaves of the tree structure. To find this information, access the Details page on the Tree Structure component. (Select Main Menu, Tree Manger, Tree Structure and then select the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Details tab.) Use the value in the Component Name field, which is the *DEPARTMENT* component in this example:

Image: Example of the Tree Details page

The following example shows the Tree Details page.



Note: Nodes and leaves can have different details components. The Nodes tab displays the tree node details component. When setting up related actions for a tree, you can configure services for either one or both.

Finding the Details Component for Rowset-Based SmartNavigation Pages

Rowsets are collections of data that are associated with buffer data. Rowset-based SmartNavigation pages use PeopleCode to populate the component buffer, generate the chart, and retrieve related actions. When the developer writes the PeopleCode programs that populate the chart, she must define a detail component from which the data is retrieved. When you assign a related content service that you want to appear as a related action, you assign the service to the detail component indicated in the PeopleCode program.

To find this information you can consult the developer or you can follow these steps:

1. Select PeopleTools, Portal, Structure & Content.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. Navigate to the SmartNavigation folder by clicking the folder labels until you reach the appropriate folder.

   When you find the folder, the label will be text, not a link.

3. Click the Edit link for that folder.

4. Use the PeopleCode Type and the iScript Parameters or Application Class Parameters fields to determine the name of the PeopleCode function or class that contains the information.

5. Open PeopleSoft Application Designer and access the record or class that you ascertained in the previous step.

6. Search for a line that sets the detail component. The exact code can vary depending on the naming conventions of the programmer, but it should set these elements:

   - MenuName
   - BarName
   - ItemName
   - Page
   - Action
   - Record
   - KeyFieldNumber
   - FieldName

   This is an example: `"MenuName.QE_SAMPLE_APPS, BarName.USE, ItemName.QE_EMPLOYEE, Page.QE_EMPLOYEE, Action.U, Record.QE_EMPLOYEE, KeyFieldNumber.001, FieldName.EMPLID"`

Related Links

"Defining Tree Details" (PeopleTools 8.54: Tree Manager)

"Accessing Definitions and Associated PeopleCode" (PeopleTools 8.54: PeopleCode Developer's Guide)

Component Level Versus Page Level Assignment

Assign the related content service or related action at the component level:

- If you want the related content to appear on each page of the component.
- If you want data to appear or change in the related content based on key values from the component.
- If you are configuring related actions for component search results
- If you are configuring related actions for SmartNavigation drop-down menus.

Assign the related content service or related action at the page field level:

- If you want the related content to appear on one, but not all, pages in a component.
- If you want data to appear or change in the related content based on non-key, page field values.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- If you are assigning the service as a *related action* and you want to add this action to an application page.

Understanding Service Parameter, Filter, and Security Configuration

When you assign a related content service to a page or component, you must configure the service parameters. Configuration can include mapping the field data to the service parameters, specifying which type of menu the service link is listed in, adding filters to the service, and specifying security options. The service parameter list and other configuration options vary based the service type, the service definition, and the service target that you choose.

Related content service configuration establishes the relationships between the service and the transaction data so that the service appears in context with the current transaction. To establish contextually relevant relationships, service configuration includes working with:

- Service parameter mapping.
- Service filters.
- Security options.

Service Parameter Mapping

Many services include parameters that you must map to key fields or page fields, system variables, and fixed values. You map the parameters to these elements so that they can pass the data to the service and create the relevant context for the related content.

This table discusses the source of the parameters for each of the service types.

Service Type	*Parameter Source*
Content reference	Search key fields from the component search record **Note:** If the search record has no key fields, the system retrieves search key fields from the component context search record.
Pagelet	Values in the Name field if the RCF_SERVICE_PARAMETER content reference attribute is defined on the Content Reference Administration page for this pagelet.
Service	Values in the URL Parameters grid for the service definition.

When you map parameters, usually you must first select the mapping type and then select the mapping data. Some services might already have the mapping defined.

This table discusses the mapping types.

Mapping Type	*Description*	*Example*
Fixed value	A hard-coded value that does not change respective to context	Service that includes the service ID as a parameter for example, PTWC_WIKIS

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Mapping Type	*Description*	*Example*
Key field	Primary or composite keys from the component	In the department table, DEPT_ID
Page field	Any level 0, 1, 2, or 3 field in the component buffer	On the education page, SCHOOL_ NAME
System variable	Any variable that the PeopleSoft system resolves at run time	%USERID, the current user

After you select a mapping type, click the Select button to see the mapping type prompt page.

Note these points about the prompt pages:

- If the mapping type is *Key Field,* then the prompt page displays a list of search key fields from the component search record.
- If the mapping type is *Page Field,* then the prompt page constructs a grid and dynamically populates it with a row for each page field that is available to the service. For each page field row, the grid also includes a hidden field indicator, the occurs or scroll level, the record (table) name, the field definition name, and the page field ID number.

  Use the page field ID number to identify a specific occurrence of a page field when the field appears multiple times in the same occurs level on the page. Each instance of a page field has a unique ID number that you find on the Order tab of the page definition in PeopleSoft Application Designer.
- If the mapping type is *System Variable,* then the prompt page displays valid system variables.

  See "Understanding System Variables" (PeopleTools 8.54: PeopleCode Language Reference).

  Important! Only %UserID, %EmployeeId, and %EmailAddress are supported for services to be displayed in the Related Actions menu of SmartNavigation pages.

If the mapping type is *Fixed Value,* then the Select button is hidden.

Note: To reduce mapping effort, the PeopleSoft Related Content Framework defines a set of reserved words and service parameters.

See Reserved Words Default Mapping.

Service Filters

A service filter hides or displays a service link in a field-level menu or the Related Content menu based on the value in the field at runtime. The Application Package PeopleCode that you specify can access the component buffer, using field values to trigger data-specific logic that causes the related content service link to be visible or hidden based on the value of the field. The filter enables the context of the data to determine whether the service link appears.

Security Options

The service definition determines the choices of security options that appear.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Understanding Service Security.

Understanding Menu Configuration

Menus provide access to related content services. Three menu layout options exist:

- Related Actions menus can contain up to three levels of folders. Folders appear first in the menu followed by links.
- The Related Content menus are a flat hierarchy and can contain no folders.
- Page field contextual menus can contain two sections: related actions and related content.

  Related actions appear first; this part of the menu can be hierarchical displaying folders and links.

  Related content appears last; this part of the menu is flat displaying links only.

Understanding the Configuration of the Related Content Frame

PeopleSoft Related Content Framework enables you to configure related content to appear in two related content frames. The Related Content frames appear beneath and to the right of the transaction page so that users can view the transaction page and the related content simultaneously. When you assign related content to a content reference, you configure the services to appear either in the bottom frame or the right frame; the two frames cannot appear simultaneously.

Understanding PeopleCode FieldChange Event Based Related Actions Configuration

PeopleSoft Related Content Framework provides a mechanism to associate a field change event of a particular field to an existing related action widget. This mechanism simplifies configuration. Also, developers no longer have to define an application class based related content service for each action and copy their FieldChange PeopleCode to the application class. The association of field change event to an existing related action widget is valid only when the following conditions are met.

- The field type must be Push Button or Hyperlink.
- The destination must be PeopleCode Command.

In addition, when a related action widget includes only a single action for a user, PeopleSoft Related Content Framework provides three options to display the single action:

- Action widget
- Link
- Button

Common Elements Used in This Section

Configure	Click to access the Configure Service page, where you map field values to the service parameters.
Content Reference	This is the content reference to which you are assigning the related content. You select the content reference on the Select a Content Reference page. This field is display-only.
Display in field menu	Select to enable access to a related content field-level contextual pop-up menu in the field to which you are mapping the service parameter. This field is enabled only when the mapping type is *Page Field.*

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting this check box causes the red related content glyph to appear in the field at runtime, even if the field is display-only.

Important! When mapping a service parameter to a page field, always examine the page to determine whether the selected page field lies within a collapsible page control such as a group box, grid, and scroll area. Services that access page fields that lie within a collapsible control that is set to the collapsed state by default must be configured to use the Display in field menu contextual menu option only.

Note: If you select a hidden field, the menu does not appear at runtime.

Is Value Required?

Some service might accept valid null values being passed to its parameters. If you select the Refresh Service On Change and Is Value Required? check boxes for this parameter, then when the mapped field value of that parameter on the transaction page changes to a null value, the system does not refresh the related content frame.

Mapping Data

Enter the constant if the mapping type is *Fixed Value.*

For all other mapping types, this field is disabled and displays the name of the field or variable that you map to the service parameter.

Mapping Type

Select from four possible mapping types to map the service parameter to:

Fixed Value: Select to map the service parameter to a constant. Selecting this option enables the Mapping Data field.

Key Field: Select to map the service parameter to one of the level-0 search keys of the component assigned to the content service.

Note: For related actions, the key name used for the related actions component is the field name that you enter here. The key value comes from the corresponding key field from the detail component key list.

Page Field: Select to map the service parameter to any level-0, level-1, or level-2 page field, including hidden fields. This option is available for page-level services only.

System Variable: select to map the service parameter to one of the available system variables.

Package

Enter the application class package name to use for the service filter.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Path

Enter the names of each subpackage in the application class hierarchy that define the location of the application class that defines the service filter. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Note: The application class parameters are validated when you save the definition; the service does not allow you to save an invalid application class.

ClassID

Enter the class name that defines the data-specific logic.

Page

Select the specific page in the component to which you are assigning related content.

This field appears in the grid when you select the Services at Page level option.

Page Name

Displays the page label. This field is display-only.

This field appears in the grid when you select the Services at Page level option.

Portal Name

This is the name of the current portal where the related content will be assigned. This field is display-only.

Read help text

Click to display the help text as defined on the Related Content Service Definition page.

See Writing Help Text for Related Content Service Definitions.

Refresh Service On Change

Select to have the service data refresh automatically when the key values on the transaction page changes. The default value is deselected.

Important! Use this option judiciously. If the service is linked to a level 1 or level 2 key field and the application user scrolls through data rapidly, performance can be affected.

Note: If the page or the page field to which a service is mapped is in *deferred processing mode,* a trip to the server is required to refresh related content.

Required

Specifies whether the service parameter is defined as *Required* in the service definition on the Related Content Service Definition page in the Service URL Parameter grid.

See Understanding Service URL Parameters.

Select

Click to access the lookup page associated with the corresponding service type. The lookup pages enable you to choose the content reference, pagelet, or service that you want to

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

assign as related content. This button is enabled only after you select the service type.

Service Parameter

Displays the list of parameters that the service requires to form the contextual relationship to the primary transaction page.

See "Using Search Records" (PeopleTools 8.54: Application Designer Developer's Guide).

See Understanding Service URL Parameters.

Service ID

Specifies the unique identifier of the service definition. For content references and pagelets, the service ID corresponds to the Name field on the Content Reference Administration page.

Service Label

Enter the label that appears in the contextual menu, on the tab when the service appears in the bottom Related Content frame, and in the pagelet header when the service appears in the right Related Content frame. The content reference name or service definition name appears as the service label by default when you select the service; you can change the default value.

Service Target

Select the target location where the related action appears. Select from these options:

Modal Window - select to have the related content appear in a modal window that hovers in front of the transaction page.

New Window - select to have the related content appear in a new browser window.

Replace Window - select to have the related content refresh the entire browser window, including navigation.

Target Content - select to have the related content appear in the target area, replacing the current page.

Service Type

Select from these three options.

Content Reference – Select this option to assign an existing content reference as related content. Because content references can be used as anonymous services, you do not have to create an explicit service definition. However, in certain circumstances, you might require an explicit service definition. See Explicit Definition Versus Anonymous Services.

Pagelet - – Select this option to assign an existing locally registered pagelet as related content. Because locally registered pagelets can be used as anonymous services, you do not have to create an explicit service definition.

Service - – Select this option to assign an existing service that you configured using the Define Related Content Service page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: After you select a service type, the system enables the Select button.

Assigning Related Content and Related Actions

To assign related content or related actions:

1. Determine whether you are making the assignment as related content or as related actions. See Related Content Versus Related Actions.

2. Access one of the following pages:

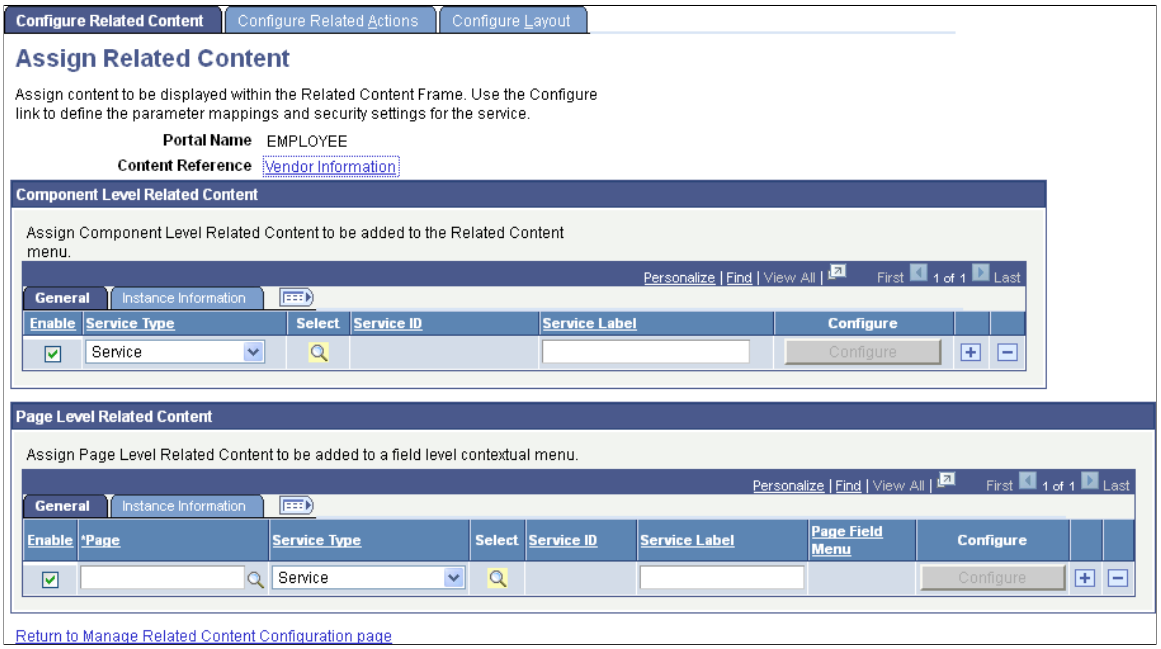
   - Configure Related Content page (click an Edit button on the Manage Related Content for Content References page or select a content reference on the Select a Content Reference page).

   - Configure Related Actions page (for content references, select the Configure Related Actions tab on the Configure Related Content page).

   - Configure Related Actions page (for pivot grids, click an Edit button on the Manage Related Actions for Pivot Grids page or select a pivot grid on the Look Up Pivot Grid page).

   Image: Configure Related Content page

   This example illustrates the fields and controls on the Configure Related Content page. The Configure Related Actions page has the same fields and controls.



3. Determine whether the assignment is to be made at the component level or at the page level. See Component Level Versus Page Level Assignment

   If the assignment is to be made at the page level, in the Page Level Related Content group box, click the Page lookup button and then, select the page to which you want to assign the service.

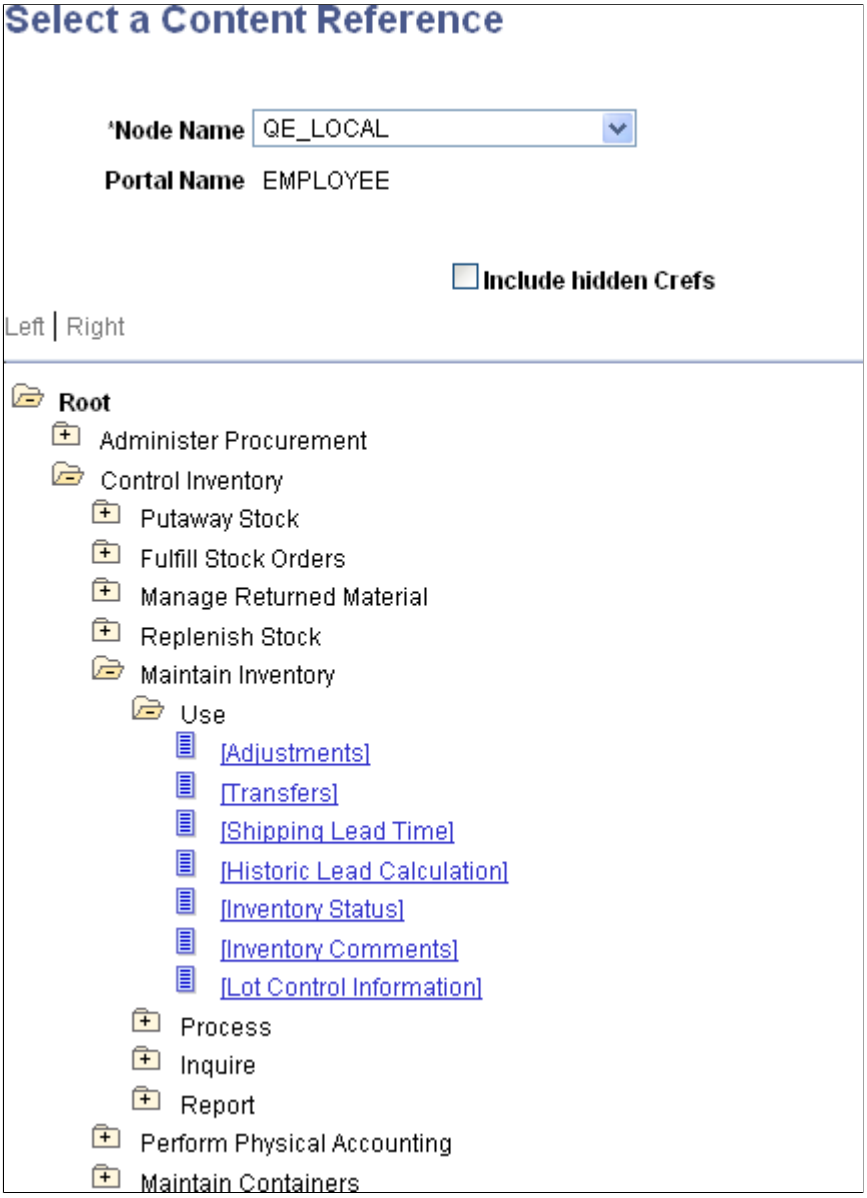
4. Select the service type:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- If you select *Content Reference,* click the Select button and select the content reference to use as an anonymous service on the Select a Content Reference page:

  Image: Select a Content Reference page

  This example illustrates the fields and controls on the Select a Content Reference page:



  Select the local node or a remote node to select a local or remote content reference.

- If you select *Pagelet,* click the Select button and select the pagelet from the Pagelet Search page to use as an anonymous service.

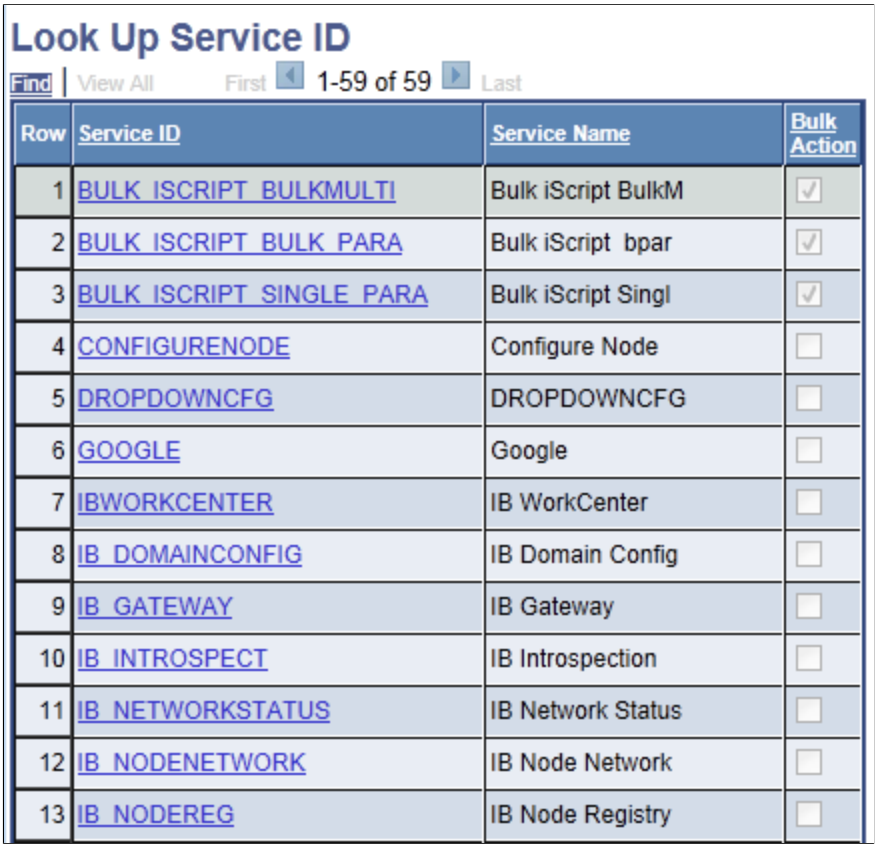
  See Searching for and Importing Pagelets for more information on searching for pagelets on local and remote nodes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- If you select *Service,* click the Select button and select the explicitly defined service on the Look Up Service ID page:

  Image: Look Up Service ID page

  This example illustrates the fields and controls on the Look Up Service ID page.



Look Up Service ID

Find | View All First 1-59 of 59 Last

Row	Service ID	Service Name	Bulk Action
1	BULK_ISCRIPT_BULKMULTI	Bulk iScript BulkM	✓
2	BULK_ISCRIPT_BULK_PARA	Bulk iScript bpar	✓
3	BULK_ISCRIPT_SINGLE_PARA	Bulk iScript Singl	✓
4	CONFIGURENODE	Configure Node	
5	DROPDOWNCFG	DROPDOWNCFG	
6	GOOGLE	Google	
7	IBWORKCENTER	IB WorkCenter	
8	IB_DOMAINCONFIG	IB Domain Config	
9	IB_GATEWAY	IB Gateway	
10	IB_INTROSPECT	IB Introspection	
11	IB_NETWORKSTATUS	IB Network Status	
12	IB_NODENETWORK	IB Node Network	
13	IB_NODEREG	IB Node Registry	

5. If necessary, modify the label in the Service Labelfield.

6. Click the Configure button to configure the service parameters, filters, and security. See Configuring Service Parameters, Filters, and Security.

Viewing Instance Information

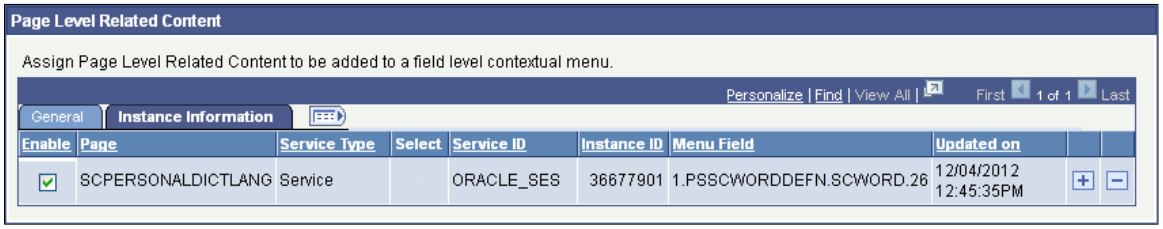
After a service configuration is saved, instance information is generated by the system. In either the Component Level Related Content, Page Level Related Content, Component Level Related Actions, or

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Page Level Related Actions group box, select the Instance Information tab to view the generated system data.

Image: Instance Information tab

This example illustrates the fields and controls on the Instance Information tab:

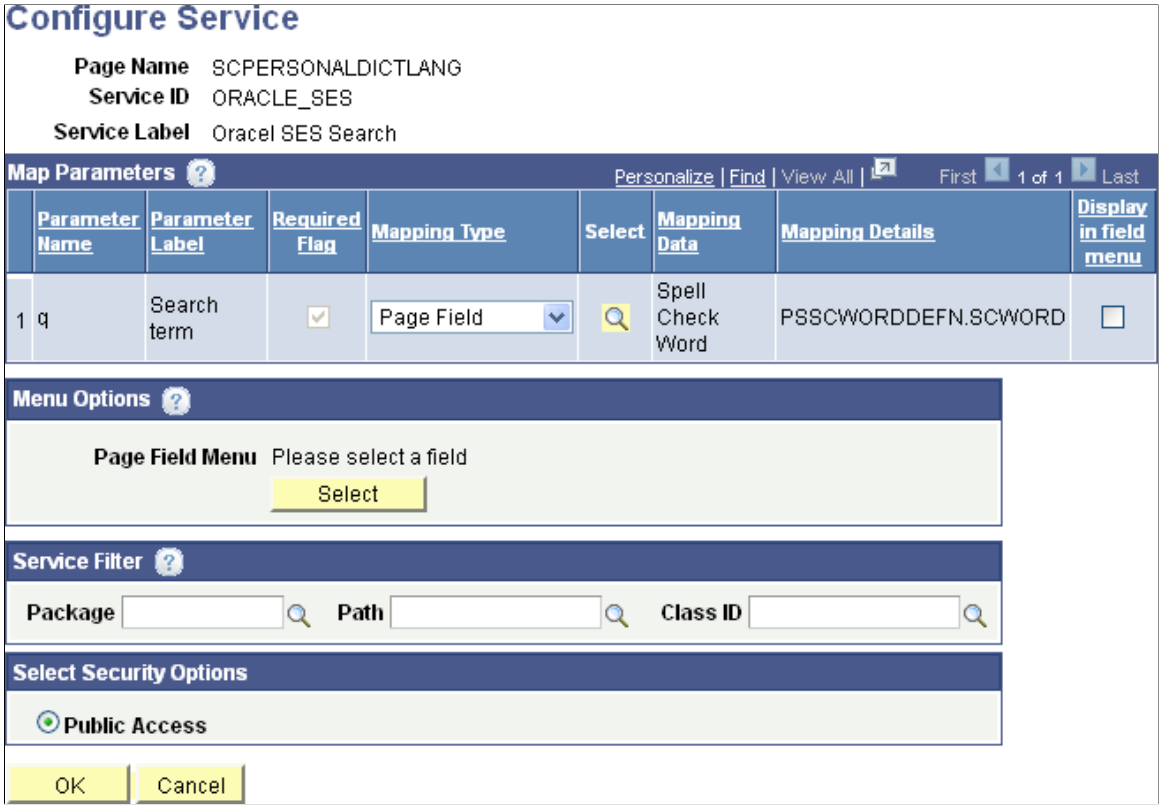


Configuring Service Parameters, Filters, and Security

Access the Configure Service page. (On the Configure Related Content page or the Configure Related Actions page, click the Configure button for a selected related content service.)

Image: Configure Service page

This example illustrates the fields and controls on the Configure Service page. You can find definitions for the fields and controls later on this page.



To configure the service parameters, filters, and security:

1. For each service parameter, select the mapping type.
2. For each service parameter, do one of the following:
   - For a fixed value mapping type, enter the value in the Mapping Data field.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- For a key field, page field, or system variable mapping type, click the Select button.

  Click a link to select the field or variable on the prompt page.

3. For page-level services, do one of the following:

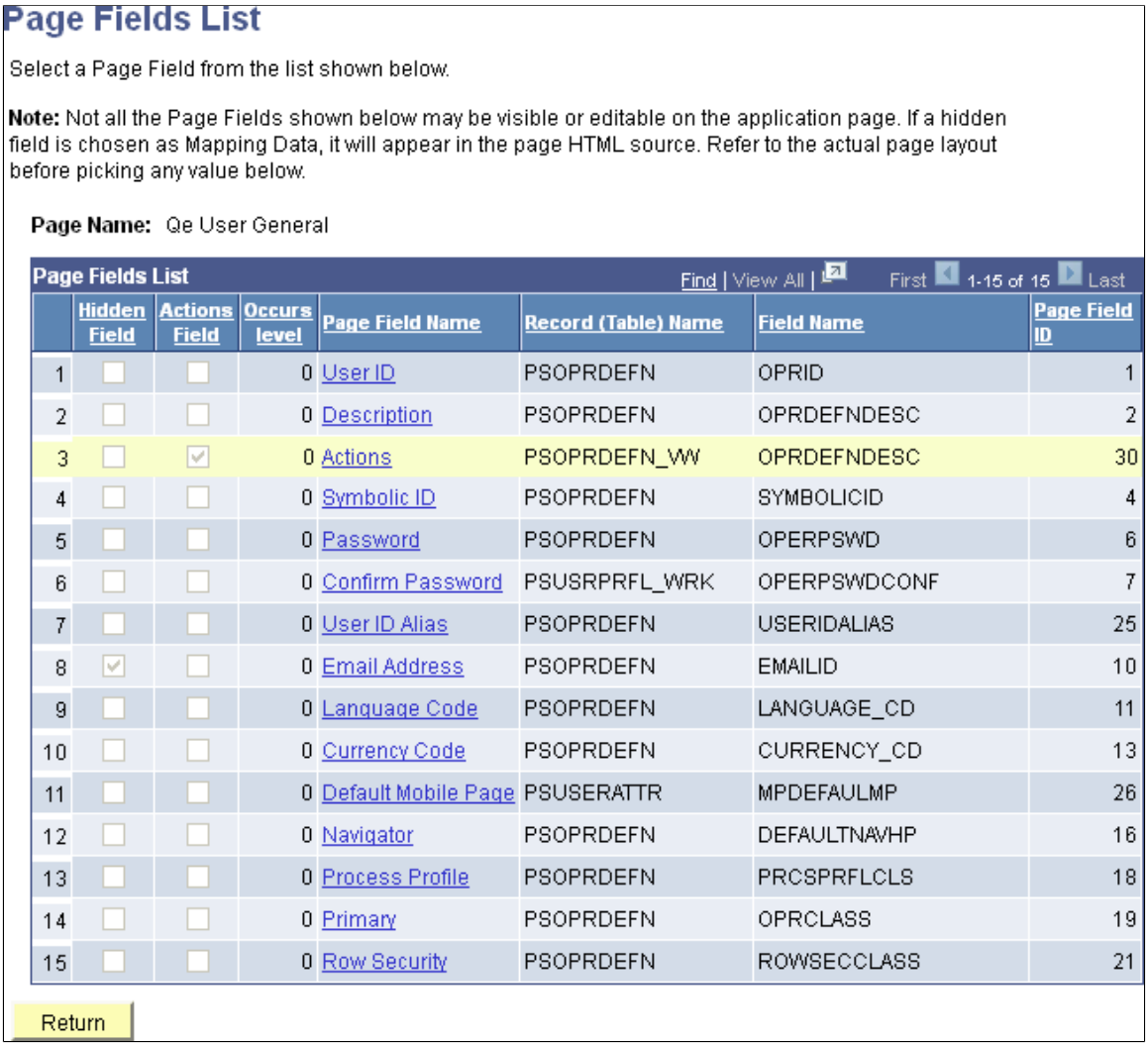
   - Select the Display in field menu option for the field that provides access to the contextual pop-up menu.

     Note: When the Display in field menu option is selected, then the Select button in the Menu Options group box is disabled.

   - Click the Select button in the Menu Options group box to select a page field different from the one that is mapped to the service parameter; the Page Fields List page is displayed:

     Image: Page Fields List page

     This example illustrates the fields and controls on the Page Fields List page:



Page Fields List

Select a Page Field from the list shown below.

Note: Not all the Page Fields shown below may be visible or editable on the application page. If a hidden field is chosen as Mapping Data, it will appear in the page HTML source. Refer to the actual page layout before picking any value below.

Page Name: Qe User General

Page Fields List — Find | View All | First 1-15 of 15 Last

	Hidden Field	Actions Field	Occurs level	Page Field Name	Record (Table) Name	Field Name	Page Field ID
1	☐	☐	0	User ID	PSOPRDEFN	OPRID	1
2	☐	☐	0	Description	PSOPRDEFN	OPRDEFNDESC	2
3	☐	☑	0	Actions	PSOPRDEFN_VW	OPRDEFNDESC	30
4	☐	☐	0	Symbolic ID	PSOPRDEFN	SYMBOLICID	4
5	☐	☐	0	Password	PSOPRDEFN	OPERPSWD	6
6	☐	☐	0	Confirm Password	PSUSRPRFL_WRK	OPERPSWDCONF	7
7	☐	☐	0	User ID Alias	PSOPRDEFN	USERIDALIAS	25
8	☑	☐	0	Email Address	PSOPRDEFN	EMAILID	10
9	☐	☐	0	Language Code	PSOPRDEFN	LANGUAGE_CD	11
10	☐	☐	0	Currency Code	PSOPRDEFN	CURRENCY_CD	13
11	☐	☐	0	Default Mobile Page	PSUSERATTR	MPDEFAULMP	26
12	☐	☐	0	Navigator	PSOPRDEFN	DEFAULTNAVHP	16
13	☐	☐	0	Process Profile	PSOPRDEFN	PRCSPRFLCLS	18
14	☐	☐	0	Primary	PSOPRDEFN	OPRCLASS	19
15	☐	☐	0	Row Security	PSOPRDEFN	ROWSECCLASS	21

Return

Use this methodology when you want to place the page field contextual menu on a field that is different from the one mapped to the service parameter. For example, the service parameter could

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

be mapped to the USERID value, but this methodology would allow you to place the page field contextual menu on the DESCRIPTION field.

In addition, use this methodology to add the related action to the menu on the free-standing Actions button on a page.

4. (Optional) Enter the package, path, and class ID of the service filter.

   Note: If the *App Class Required* option was selected on the service definition, then a service filter is not optional and must be identified to be able to save the service configuration.

5. Select the security option.

6. Click the OK button to accept the configuration and return to the configuration page.

Related Links

"Specifying Type Properties for Push Buttons or Links" (PeopleTools 8.54: Application Designer Developer's Guide)

Defining Related Content Services

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Menus

1. Access the Configure Layout page. (On the Configure Related Content page or the Configure Related Actions page, select the Configure Layout tab.)

   Image: Configure Layout page

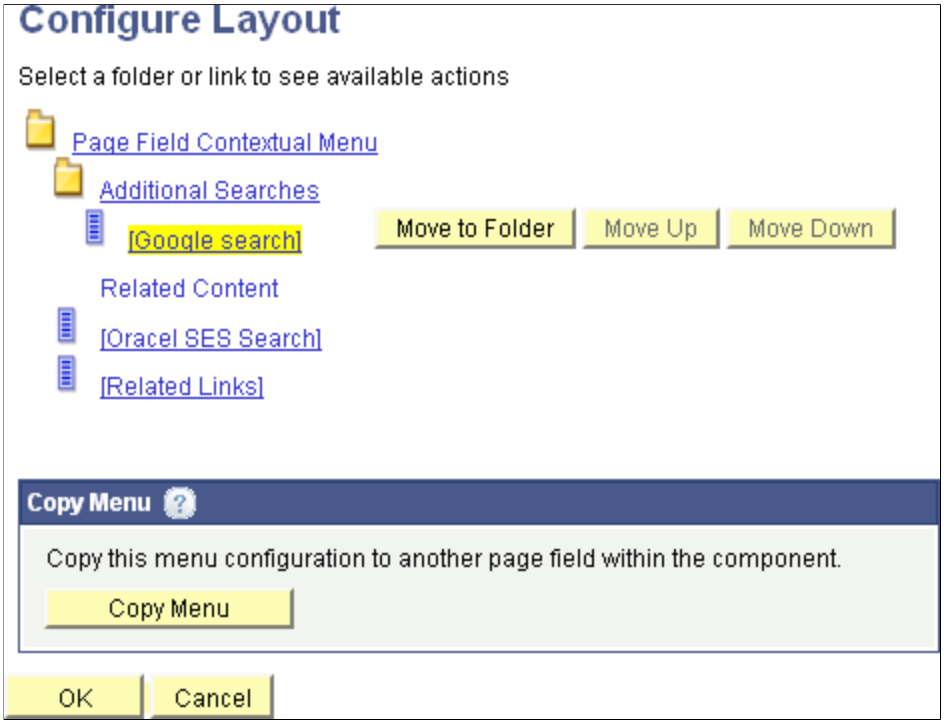
   This example illustrates the fields and controls on the Configure Layout page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

2. On the Configure Layout page, click an active Configure Layout button to configure a menu layout.

   Image: Configure Layout secondary page

   This example illustrates the fields and controls on the Configure Layout secondary page. You can find definitions for the fields and controls later on this page.



3. For related actions menus and page field contextual menus, you can add and move folders in the menu.

   Important! You cannot add or move folders in a related content menu.

   When you select a folder, these options appear:

   Note: Not all options appear for all folders.

Add Folder	Click to add a folder one level below the selected folder. If this option does not appear, you cannot add a subfolder.
Delete Folder	Click to delete the folder If this option does not appear, you cannot delete the folder.
Move Up	Click to move the folder up within the list of folders in the same level.
Move Down	Click to move the folder down within the list of folders in the same level.

4. For all menu types, you can move service links within the menu structure.

   When you select a service link, these options appear:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: Not all option appear enabled for all services.

Move to Folder	Click to access the Select a Folder page, which enables you to move the service to another folder. If this option does not appear, you cannot move the service link to a different folder.
Move Up	Click to move the service link up within the list of services in the same folder.
Move Down	Click to move the service link down within the list of services in the same folder.

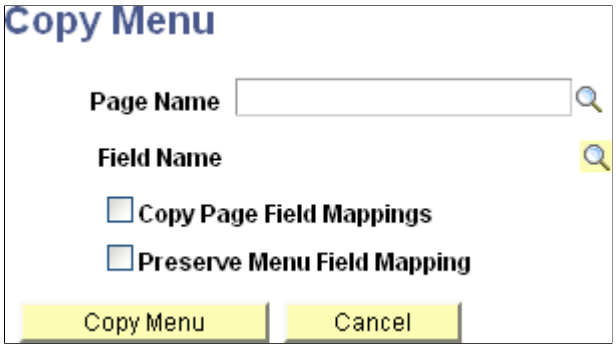
5. Click the OK button.
6. Click the Save button.

In addition, you can use the Copy Menu page to copy a page field contextual field menu layout from one page field to a different page field on the same page or on a different page within the same component. See Copying Menu Layouts.

Copying Menu Layouts

Image: Copy Menu page

This example illustrates the fields and controls on the Copy Menu page. You can find definitions for the fields and controls later on this page.



Use the Copy Menu page to copy a page field contextual field menu layout and services configuration from one page field to a different page field on the same page or on a different page within the same component.

The following values are copied to the new page field:

- Service target.
- Service label.
- Key field mappings.
- Fixed value mappings.
- System variable mappings.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Page field mappings (see note).
- Menu layout.
- Menu to which the copied service is attached.

Important! The Copy Page Field Mappings option must be selected to copy page field mappings. However, no error checking is performed to ensure that the page field mappings are still correct. Therefore, select this option only in circumstances that provide equivalent mapping of page fields.

The Preserve Menu Field Mapping option is valid only when the Copy Page Field Mappings option is also checked. Use the Preserve Menu Field Mapping option only when the menu location is moved to the new field. For example, a service parameter is mapped to the User ID field, which is marked as the menu display field as well. When this menu is copied to a new page field, Description, we want to keep the original parameter mapping to the User ID field. In this instance, we would also select the Preserve Menu Field Mapping option.

After the menu layout and service configuration is copied to the new page field, additional manual configuration might be required depending on the original configuration and how it was copied:

- If the services copied do not contain any page field mappings or if the Copy Page Field Mappings option was selected, the services will automatically be marked active and will be assumed as completely configured.
- If the services copied contain page field mappings *that were not copied,* the services will automatically be marked inactive by default. You must manually map these service parameters to page fields. See Configuring Service Parameters, Filters, and Security.
- After you manually map unmapped service parameters, the PeopleSoft Related Content Framework will check other services in the same menu to see if they contain a service parameter by the same name. If so, you will be prompted to copy this mapping to the other services as well. Choosing yes will apply the same mapping to any other inactive services in the same menu that also contain the same service parameter. The framework will not update any active service configurations that also contain the same service parameter.
- After the framework applies this service parameter mapping to other inactive services, an additional check will be made to determine if all service parameters are now mapped. If they are, you will be prompted to mark all inactive services as active in the new page field menu.

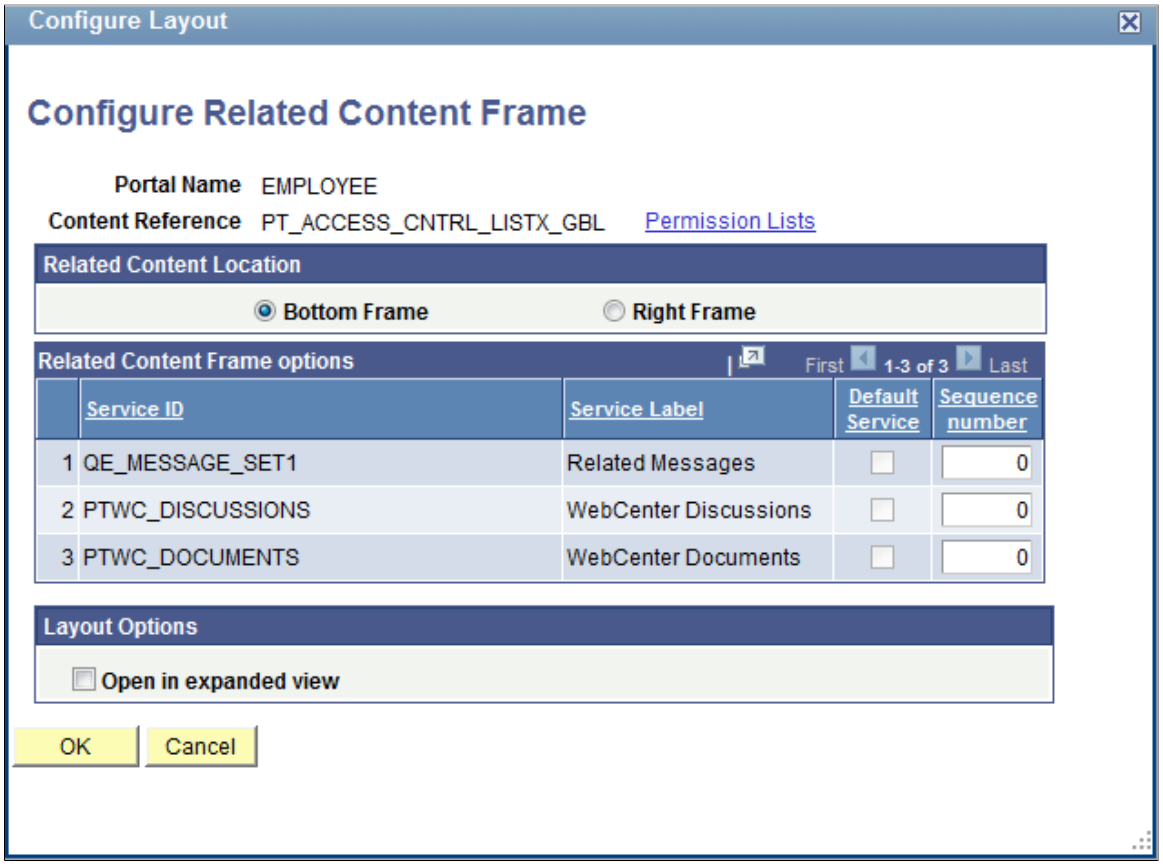
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring the Related Content Frame

Access the Configure Related Content Frame page (On the Configure Layout page, click the Configure Layout button in the Related Content Frame group box).

Image: Configure Related Content Frame page

This example illustrates the fields and controls on the Configure Related Content Frame page. You can find definitions for the fields and controls later on this page.



1. Select the frame location:

   - *Bottom Frame* – Select to have related content services appear as tabs in the frame beneath the transaction page.

   - *Right Frame* – Select to have related content services appear as pagelets in the frame to the right of the transaction page.

2. Configure the order and initial state of the services that appear in the Related Content frame:

   - *Default Service* – Select the service to be displayed by default when the Related Content frame is opened automatically at the bottom of the page.

     Note: You must select the Open in expanded view option first in order to select a default service.

   - *Initially Minimized* – Select so that the service appears in the closed state when the user initially accesses the Related Content frame at the right of the page.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Sequence number* – Specify the sequence for the services.

3. Select the frame layout options:
   - *Open in expanded view* – Select to open the Related Content frame automatically when the user accesses the target application page rather than after the user selects a service.
   - *Max no of minimized pagelets* – Enter the number pagelets that can appear minimized when the Related Content frame appears at the right of the page. If a user minimizes more than this number, the pagelets move to the More drop-down list at the bottom of the frame.

Configuring Display of PeopleCode FieldChange Event Based Related Actions

Configuring Display of PeopleCode FieldChange Event Actions

You can configure the display of PeopleCode FieldChange event based related actions in PIA. The display configuration should be completed in Application Designer; not in PIA. The following table describes the various options:

Display Option	***Configuration***
Show FieldChange event actions under a related actions menu only.	In Application Designer, access the Push Button/Hyperlink Properties dialog box, and in the Related Content Event group box, select Display in Related Action Menu. **Note:** The Display in Related Action Menu check box is available only if the field is of type Push Button or Hyperlink and the destination is PeopleCode Command. The Action Widget drop-down list displays all the related actions widget configured for the page. When you select an Action Widget, the FieldChange event of this field is associated as a menu item in the widget. Related action widgets in the drop-down list will only display the level at which the field is selected.
Show FieldChange event actions under a related actions menu and on a page.	In Application Designer, access the Push Button/Hyperlink Properties dialog box, and in the Related Content Event group box, select Display in Related Action Menu and Show in Page.

See "Using Push Buttons and Links" (PeopleTools 8.54: Application Designer Developer's Guide).

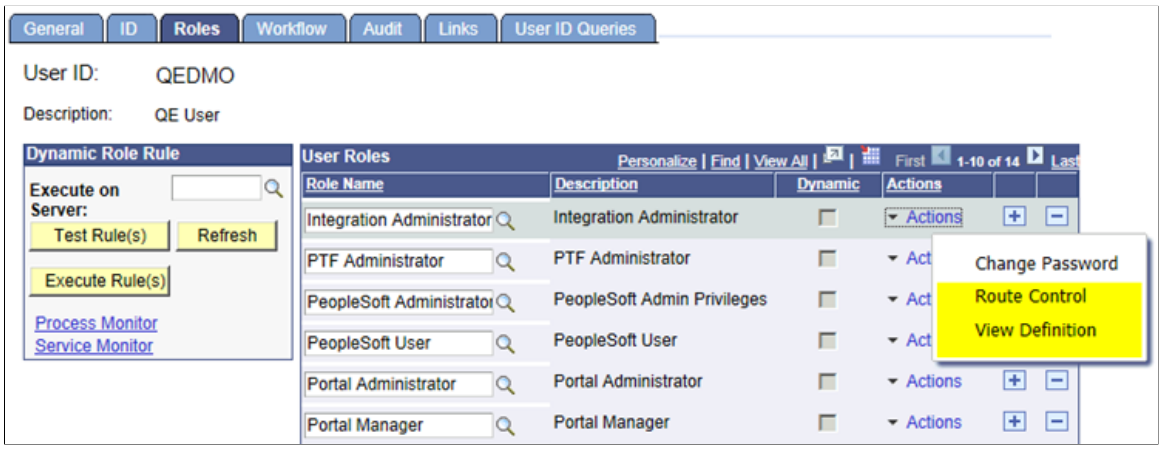
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

FieldChange Event Action in Related Action Menu

In this example, the Display in Related Action Menu check box is selected.

Image: Show in Related Action Menu option

This example illustrates the display of FieldChange event action in related action menu.

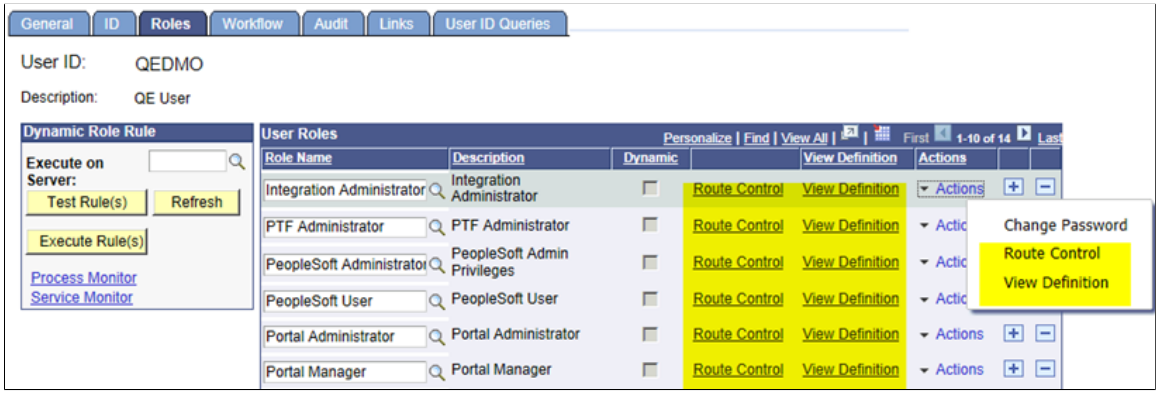


FieldChange Event Action Shown in Page and Related Action Menu

In this example, the Display in Related Action Menu and the Show in Page check boxes are selected.

Image: Show in Page option

This example illustrates the display of FieldChange event action on a page and in related action menu.



Configuring Display of Single Action on a Page

When a related action widget includes only a single action for a user, you can configure the display of that single action by selecting either action widget, link, or button.

To configure the display of single action on a page, in Application Designer, access the Push Button/ Hyperlink Properties dialog box, and in the Action Widget Settings group box, select an option (Action Widget, Link, or Button) from the Display Single Action Menu as drop-down list.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Display of Single Action on a Page

In this example, the single action of a related action widget is shown as a link on a page.

Image: Single action as link

This example illustrates the display of single action of related action widget as a link.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 18

Creating and Configuring WorkCenter Pages

Understanding WorkCenter Pages

WorkCenter pages (or WorkCenters for short) are a type of configurable PeopleSoft page that enable you to offer access to related transactions, analytics data, query results, and other PeopleSoft resources from one central location. You can organize the elements on WorkCenter pages based on processes, procedures, roles, or other business requirements that are specific to your organization. When carefully designed, WorkCenter pages bring together the PeopleSoft elements that your users access most when performing the tasks that comprise their primary responsibilities. As a result, WorkCenter pages can promote productivity by minimizing the time that users spend navigating between tasks.

WorkCenter pages can bring these different elements together because WorkCenter pages use a portal template that provides flexibility in the way that you access components and pages and also in the way the application displays these elements. The WorkCenter portal template differs from the standard portal template in two primary ways:

- The WorkCenter portal template partitions the browser window into two frames: an adjustable pagelet area and target a content area.
- The two areas of the WorkCenter portal template are designed to interact by passing contextual information back and forth by using an event-driven model. The actions in the pagelet area determine the content in the other.

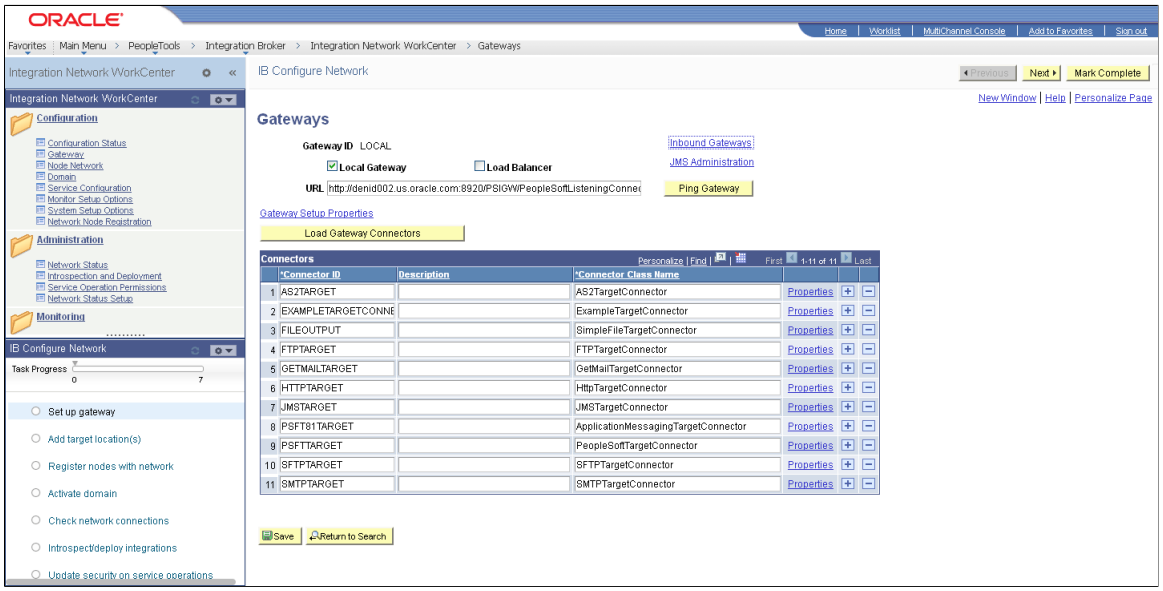
When users access WorkCenter pages, they are presented with pagelets that contain links to components, analytics, query results, and other PeopleSoft resources. The WorkCenter pagelet area is designed to extract and leverage values that are necessary to complete the transaction in the target area. When users click a link in the pagelet area, it triggers the appropriate transaction, and passes values from the pagelet

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

area so that the application can display the appropriate content in the target content area or in a new browser window.

Image: Example of a WorkCenter page

The following example illustrates the main parts of a WorkCenter page. On the left, the pagelet area displays two pagelets configured to display in this WorkCenter. On the right, the target area displays the current transaction page as selected in a WorkCenter pagelet. The standard portal header, including the drop-down menu and navigation breadcrumbs, is displayed at the top of the WorkCenter page.



The Pagelet Area

The WorkCenter pagelet area (also labeled the Task Panel when no pagelets have been configured) consists of three parts:

- The header.
- The pagelet area.
- The footer.

In the header, the title, the Home button, and the pagelet area controls drop-down menu are visible. You click the Home button to return to the WorkCenter starting page. You click the pagelet area controls drop-down menu to access up to three links that enable you to reload, personalize, and configure the WorkCenter page.

The pagelets that you assign to the WorkCenter page appear in the main body of the pagelet area. Each pagelet you assign to the area is a self contained unit that you control independently of other pagelets. Pagelets appear one above the other, either in an expanded or collapsed state based on your configuration. Expanded pagelets divide the available space evenly until the user resizes a pagelet by dragging the pagelet handle. The system automatically adjusts the size of remaining pagelets to fill the available space.

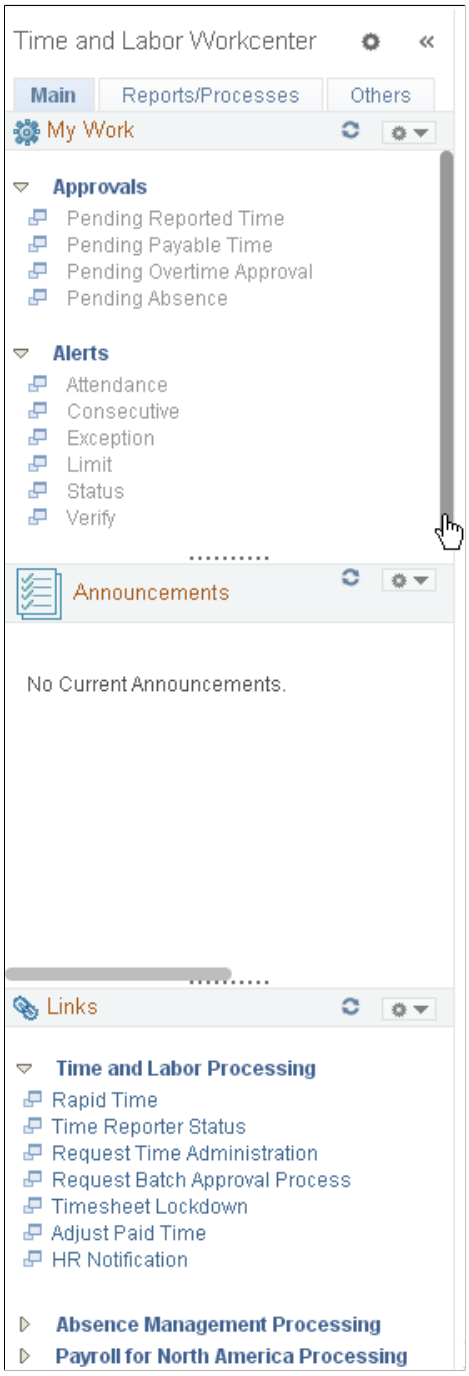
At the bottom of the pagelet area, the More drop-down menu appears in the footer. You can configure the WorkCenter page to *hide* any pagelet that is minimized instead of collapsing the pagelet into its header. The hidden pagelet is minimized to the More menu. When you click the drop-down button, the titles of the minimized pagelets appear in a list. Select a pagelet from the list and the systems restores the pagelet to its original size and positions it at the bottom of any other expanded pagelets in the pagelet area.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The pagelet area is adjustable in width and its configuration is flexible. You can organize the pagelet area by creating pagelet groups. Each group appears as a separate tab. If you configure multiple pagelet groups, a tab appears at the top of the each page to identify the pagelet group and provide a familiar vehicle for accessing the pagelets. You can define options such as the initial size of the pagelet area, the pagelet group order, and the pagelet group that appears by default when you access a WorkCenter page.

Image: WorkCenter pagelet area showing tabbed pagelet groups

The following example of a WorkCenter pagelet area shows three, tabbed pagelet groups:



After creating pagelet groups, you assign pagelets to the groups. You specify options such as the order of the pagelets, which pagelet receives focus by default, whether the pagelet is collapsed or expanded as

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

its initial state, and the maximum number of pagelets that remain visible when in a minimized state. The pagelet area also supports user personalizations to the extent that you enable them.

Generally, pagelets that you assign to WorkCenter pages are narrow format pagelets that are specifically designed to appear in the resizable pagelet area. The pagelet area supports homepage and template pagelets that are registered in either the Pagelets or the Template Pagelets folder in the portal registry.

Note: You access these folders by selecting PeopleTools, Portal, Structure & Content and then clicking the Portal Objects link.

The pagelet area supports iScript-based pagelets, component-based pagelets, and even external URL-based pagelets. You can use Pagelet Wizard to create and publish iScript-based template and homepage pagelets such as those that use navigation collections and queries as data sources. You can also manually create and register component-based or other pagelets.

Related Links

Understanding WorkCenter Pagelets

The Target Content Area

The primary purpose of the target content area is to display the transaction page, query results, analytics, or other resources that the application user requests as the result of clicking a link in the pagelet area. You use target content area to search, view, edit, and complete your transactions. The target content area can display any type of content that a standard PeopleSoft transaction page can display. In addition, the target content area can display a special type of page known as a WorkCenter dashboard, which allows you to incorporate dashboard functionality into your WorkCenters.

You can also set up and configure pages and pagelets to use interwindow communication, which enables pagelets to communicate in any direction: from pagelet area to target content area, from target content area to pagelet area, and even between elements that occupy the same section.

You can target resources to appear not only in the target content area, but to a new browser window or to a current browser replacing the entire WorkCenter page with the resource.

Example of Setting the Target to Open in a New Window

You can set the target to open in a new window when you need to view both the current page and the new page simultaneously. For example, an accounts payable representative is completing an invoice and needs to view customer history at the same time. You could design a pagelet so that the link opens the Customer History page in a new window.

Example of Setting the Target to Replace the Current Window

You can set the target to replace the current window entirely. For example, an account representative is reviewing her largest account and then wants to view a dashboard that shows all accounts in relation to one another. You could design a pagelet with a link that opens the Accounts Dashboard and replaces the current window.

WorkCenter Page Layout Templates

WorkCenter pages use an iframe-based portal template that supports four sections:

- The portal header and drop-down style menu.
- The pagelet area.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The target content area.
- The related content frames (optional).

The portal header and drop-down style menu, when enabled, appear at the top of the page above the pagelet and target content areas. The adjustable pagelet and target content areas occupy the middle of the page. The related content frames appear either below or beside the target content area when the current transaction page has related content configured.

You can change the layout of WorkCenter pages by using page layout templates. Page layout templates use Application class PeopleCode to determine the location of the pagelet area and target content area. WorkCenter pages deliver application class PeopleCode that renders the pagelet area on either the left side or the right side of the page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

WorkCenter Page Layout Template Examples

Image: Example WorkCenter page showing the Related Content frame

In this example, which uses the default WorkCenter page layout template, you see the drop-down menu, the pagelet area on the left, and the Related Content frame to the right of the target content area:

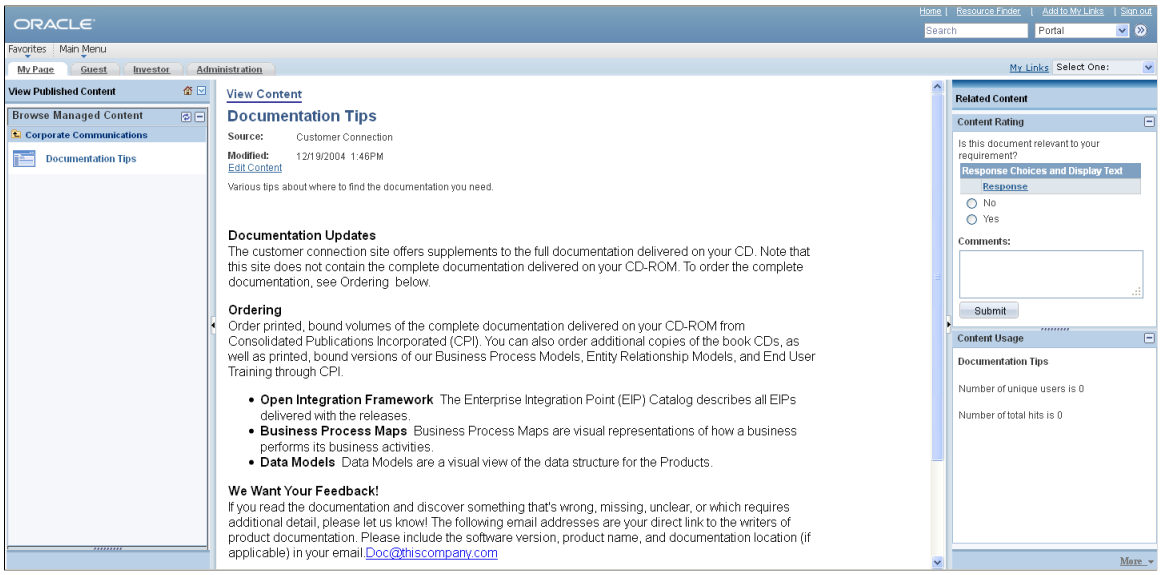
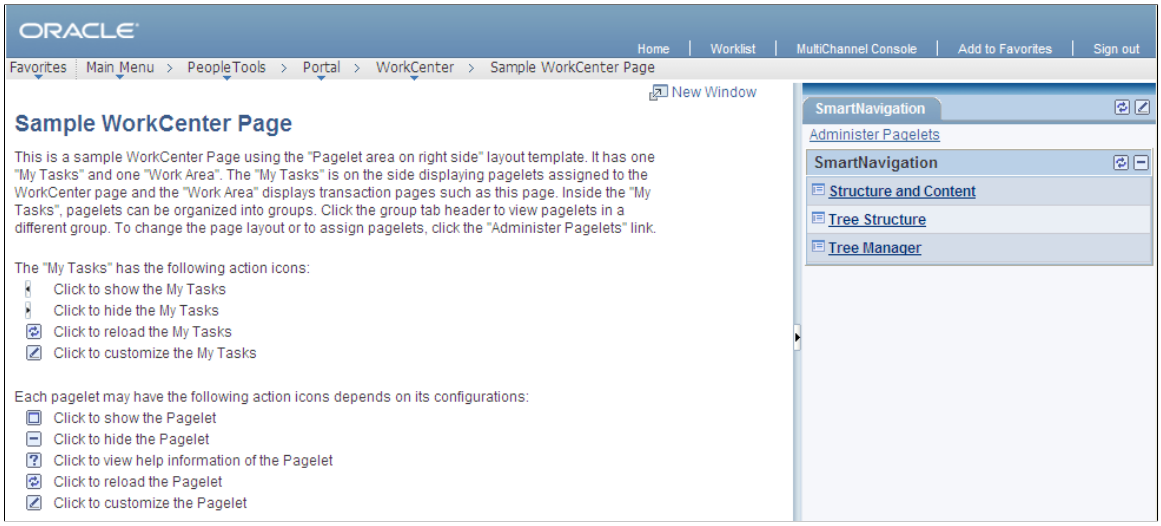


Image: Example WorkCenter page that uses an alternate page layout template

In this example, you see the pagelet area on the right side of the WorkCenter page as specified by a custom page layout template:



Related Links

Defining WorkCenter Page Layout Templates

Additional Considerations When Customizing Pages

Page Design and Processing Considerations for WorkCenter Pages

WorkCenters have different characteristics from standard PeopleSoft components and pages. This section discusses design and processing differences that you should consider when you work with WorkCenters, WorkCenter starting pages, and pagelets.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The areas of the WorkCenter are organized in a left-to-right, or top-to-bottom fashion. The pagelet area on the left presents the navigation links for the WorkCenter. The target content area in the middle provides the work area of the WorkCenter. The optional Related Content frame, on the right or at the bottom, presents additional information related to the current transaction.

Starting Page Considerations

When you create a WorkCenter starting page, consider the following differences and page design recommendations, which accommodate those differences:

- WorkCenter starting pages are determined by the menu item associated with the content reference that you select when you create a new WorkCenter.

  While the page that appears initially in the target content area can be any page, you might want to start with a summary page showing the status of the tasks to be completed in the WorkCenter. The Unified Navigation WorkCenter employs this approach. Alternatively, you might want to display the first task to be completed. WorkCenters that have designated an activity guide pagelet as the in-focus pagelet employ this approach. In addition, you could choose a WorkCenter dashboard as a starting page to display homepage pagelets on a special dashboard within the target content area.

- Typically, the component should consist of only one page, but no tabs or page links should appear on the starting page.

  Deselect the Multi-Page Navigation check boxes on the Internet tab.

- The elements that make up the tool bar and page bar are not applicable and should not appear on the WorkCenter starting page.

  Select the Disable Toolbar and Disable Pagebar check boxes on the Internet tab.

- WorkCenter starting pages generally require no save functionality.

  Select the Disable Saving Page check box on the Use tab.

- Do not use a search page as a WorkCenter starting page.

  If necessary, select a “dummy” record—one that has no keys or contains only one row of data—as the component search record on the Use tab.

  The INSTALLATION record is an example of a “dummy” search record.

Processing Considerations

Note these points about WorkCenter page processing:

- WorkCenter pages, like homepages and dashboards in your PeopleSoft applications, use deferred loading to load pagelets by default. When a WorkCenter page uses deferred loading, each pagelet appears independently from one another. When the data for the pagelet has been loaded, the pagelet appears. (This default behavior can be overridden be setting the PTPGLTDEFERLOAD content reference attribute to false for a specific pagelet.)

- WorkCenter pages and assigned pagelets support interwindow communication. Use the Interwindow Communication page to configure interactivity among pagelets that share a WorkCenter page.

  See Understanding Interwindow Communication.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- WorkCenter pages can process contextual parameters that you include in the WorkCenter page URL. The WorkCenter template accesses the parameters and passes them to the pagelets, which use the values to determine the records to retrieve and display. For example, you can create a WorkCenter page with pagelets that display customer information. You can pass the customer ID to the WorkCenter page URL as a context parameter so that all pagelets retrieve and display information about the same customer.

  The context parameter, CONTEXTPARAMS, must use the NAME:Value format and be passed as a query parameter string. Use this format when you build the WorkCenter page URL:

  ```
  http://myserver.example.com:8920/psp/ps/EMPLOYEE/EMPL/c/PTAL_ADMIN.PTAL_SAMPLE⇒
  _PAGE.GBL?CONTEXTPARAMS=USERPROFILE:VP1& CONTEXTPARAMS=EMPLOYEEID:1242.
  ```

  Using the previous example URL, the WorkCenter template retrieves the query string parameters, which it then sends to each pagelet in the following query string format ?USERPROFILE=VP1& EMPLOYEEID=1242.

  When you develop WorkCenter pagelets, use the **%Request.getParameter** PeopleCode method to retrieve and make use of the context parameter values.

Limitations

WorkCenters cannot be consumed as WSRP content.

Related Links

Limitations on PeopleSoft Content for WSRP Consumption

Additional Considerations When Customizing Pages

When you customize the look and feel of your PeopleSoft system, there are some additional considerations with respect to WorkCenters and WorkCenter pages.

If you make style customizations to the background color for pages that appear in the WorkCenter target content area, or if you make style customizations to the background color for WorkCenter pagelet area, or if you make style customizations to the background color for the portal homepage, you will want to make similar style changes to the following additional style classes:

- ptalPgltAreaResizeBar
- ptalPgltAreaHorizontalResize

In PeopleSoft Application Designer, make these changes in the active style sheet or to each of these style sheets:

- PTAL_SUBSTYLE_VL
- PTAL_SUBSTYLE_VL_SWAN
- PTAL_SUBSTYLE_VL_TANGERINE

Understanding WorkCenter Pagelets

You can create iScript-based and component-based homepage pagelets and template pagelets for use with WorkCenter pages. You use Pagelet Wizard to create iScript-based pagelets. Pagelet Wizard pagelets can be based on many possible data sources, such as activity guides, navigation collections, search records, HTML, and queries. You use PeopleSoft Application Designer to create component-based pagelets

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

that are based on PeopleTools definitions. This section presents overviews of pagelets based on these PeopleSoft data sources:

- Activity guides
- Navigation collections
- PeopleSoft query
- Components

Related Links

Understanding Pagelet Wizard Data Sources
Developing iScript-Based Pagelets

Pagelet Development Overview

This table summarizes the steps to be performed and references the corresponding documentation that you should review for developing WorkCenter pagelets.

Prerequisite	***Cross-Reference***
Create an activity guide template and generate an activity guide pagelet.	See Creating and Maintaining Activity Guide Templates and Generating an Activity Guide Pagelet.
Create navigation collections that you want to assign to the pagelet area.	See Creating and Maintaining Navigation Collections.
Create PeopleSoft query definitions that you want to use in the pagelet area.	See "Creating New Queries" (PeopleTools 8.54: Query).
Create PeopleSoft page definitions, add and configure hyperlink page controls, and then define and register the components that you want to use as pagelets.	See "Understanding Page Design and Development" (PeopleTools 8.54: Application Designer Developer's Guide). See Understanding Pagelet Development. See "Using Push Buttons and Links" (PeopleTools 8.54: Application Designer Developer's Guide). See "Defining Components" (PeopleTools 8.54: Application Designer Developer's Guide).
Use Pagelet Wizard to create and publish pagelets that use your navigation collections and query definitions as data sources.	See Understanding Pagelet Wizard. See Selecting a PeopleSoft Query Data Source. See Selecting a Navigation Collection Data Source.
Manually add component-based template pagelets to the template pagelets portal objects folder.	See Defining Content References.

Activity Guide Pagelets

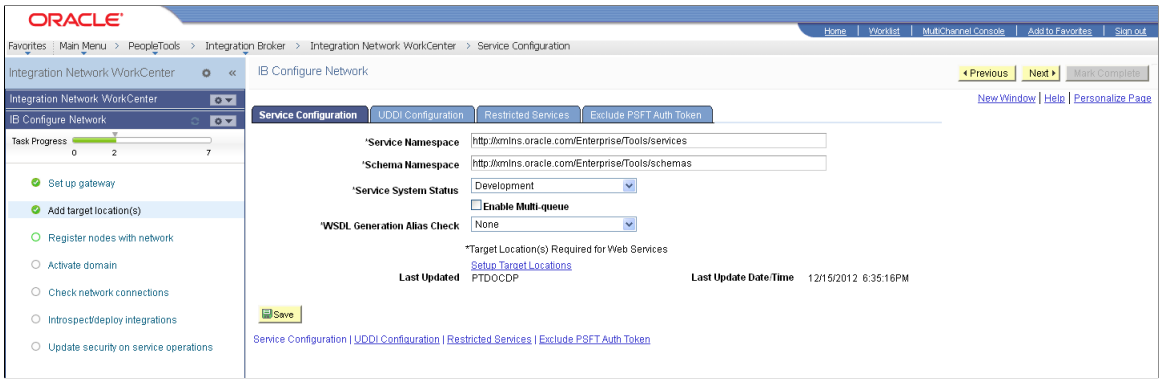
Activity guides are typically deployed on a WorkCenter page, which is where they can be accessed by users. Prior to creating a WorkCenter page where the activity guide will be deployed, you must first

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

use Pagelet Wizard to generate a pagelet for the pagelet area of the WorkCenter page. This pagelet will present the action items to the user in list form.

Image: WorkCenter with an activity guide pagelet and target content

In this example, the IB Configure Network activity guide pagelet is displayed in the pagelet area of the WorkCenter page with the Service Configuration page in the target content area:



Depending on whether you want to tie the activity guide pagelet to a specific activity guide instance or to dynamically create the activity guide instance, there are three methodologies for generating the activity guide pagelet in Pagelet Wizard. For more information, see Generating an Activity Guide Pagelet.

Navigation Collection-Based Pagelets

Navigation collections are portal objects that you use to create your own groupings of existing folders and content references, which you can then deploy or publish to different users or groups of users. Navigation collections enable you to create links to content that you already have in the portal registry and to organize the links in ways that specifically address your users' business needs.

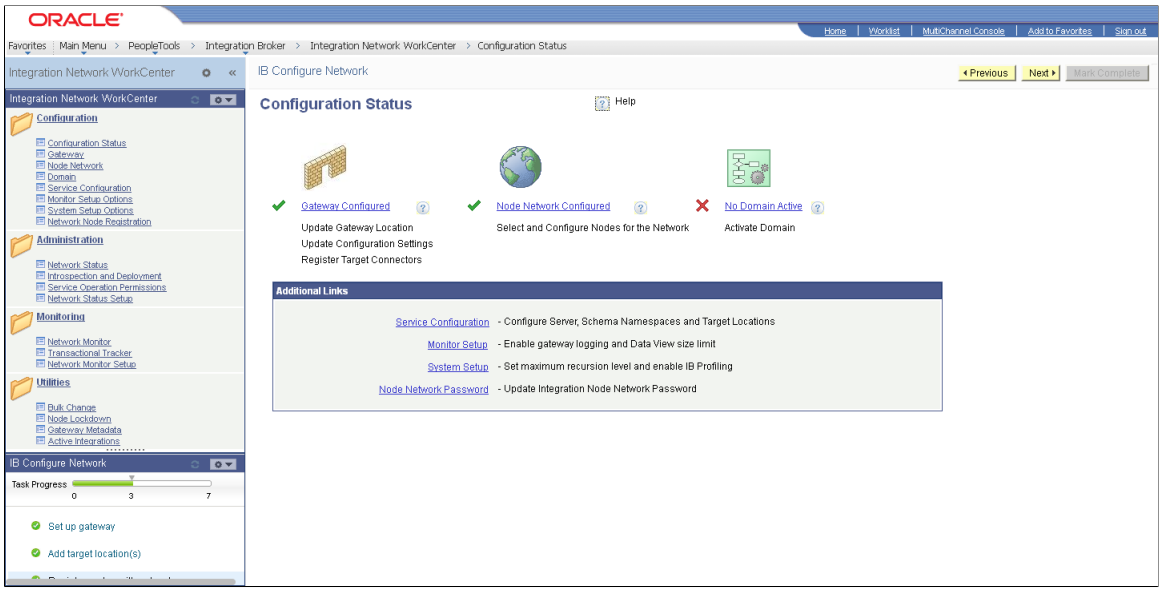
Navigation collection-based pagelets appear in the pagelet area as folders and links. When users click a folder, the folder contents appear in the pagelet. When users click a link, the resource opens in the target content area by default. Because navigation collections are designed as alternate menu structures, you navigate through folders to find links, which generally reference components. When users click the link, the component search page appears in the target content area unless the search record is configured so that

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

the search page is bypassed. In such cases, the component requires no search parameters and immediately opens in the target content area.

Image: WorkCenter with a navigation collection pagelet and target content

In this example, the Integration Network WorkCenter navigation collection pagelet is displayed in the pagelet area of the WorkCenter page with the Configuration Status page in the target content area:



Publishing Navigation Collection-Based Pagelets for Use With WorkCenter Pages

You create and maintain navigation collections using the navigation collections component. Using the pages in this component, you set the properties that determine the order in which the folders and links appear, and also the location where the target appears. If you decide to create pagelets based on navigation collections, you should review the link properties to ensure that neither replace the current page or open in a new browser window is selected. This will allow the content to open in the target content area, which is typicallythe desired behavior in a WorkCenter context.

You can use Pagelet Wizard to create pagelets based on any *published* navigation collection. At step two of the pagelet creation process in Pagelet Wizard, you must specify your navigation collection as the data source. Choose from any of your custom navigation collections or the PeopleSoft-delivered navigation collections. Continue through the process to step 6, where you specify *Homepage Pagelet* or *Template Pagelet* as a publishing option. When you save the pagelet, Pagelet Wizard creates the pagelet and you can then assign it to the pagelet area of a WorkCenter page.

Specific details for creating and publishing navigation collections and using Pagelet Wizard to publish them as pagelets are located in this documentation.

See Creating or Editing a Navigation Collection.

See Selecting a Navigation Collection Data Source.

See Step 6: Specifying Pagelet Publishing Options.

PeopleSoft Query-Based Pagelets

PeopleSoft Query is a reporting tool that enables you to extract your data by using visual representations of your PeopleSoft database, without writing SQL statements. The queries that you create can be as

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

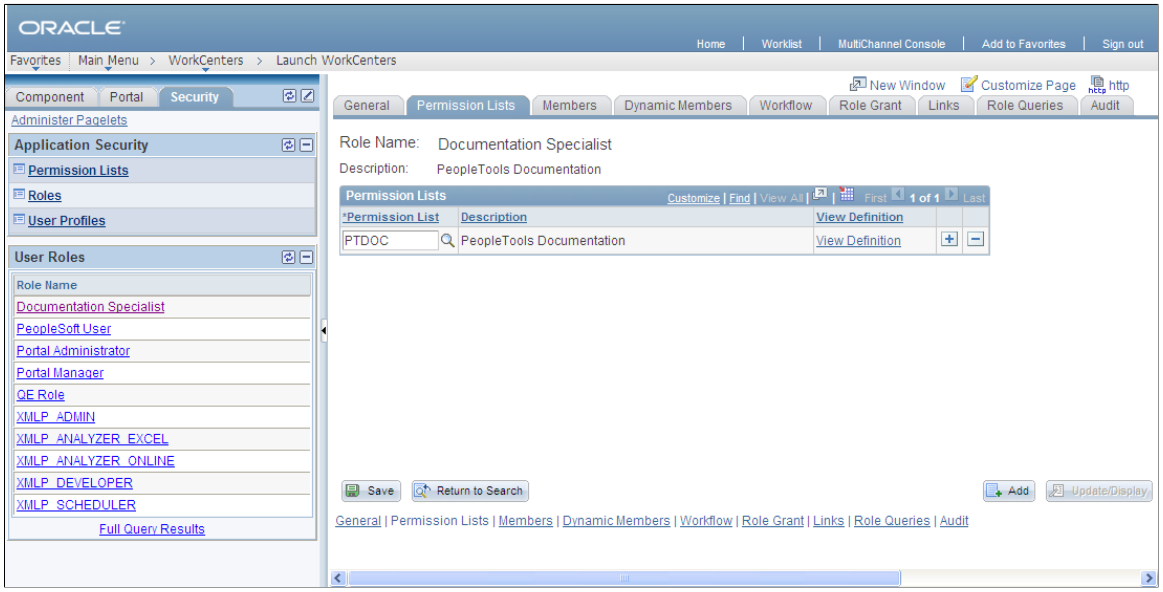
simple or as complex as necessary. You can design queries that return basic lists of static text. You can also use more advanced features, such as *drilling URLs*. By configuring drilling URLs and associating them with query fields, the values that the query retrieves appear as links, which you click to access both internal and external content.

Query-based pagelets can take advantage of queries, whether they are simple or complex. You might base a pagelet on a query that retrieves informational text to display to the user. However, query-based pagelets also have access to the more advanced query metadata, such as the drilling URLs. If you link fields in query-based template pagelets to these drilling URLs, the values that the pagelet retrieves appear as links as they do in PeopleSoft Query. Unlike PeopleSoft Query, however, you can configure the links to open the resource in the target content area of the WorkCenter page.

Because you can configure the links so that the associated resources open in the WorkCenter page target content area, query-based pagelets that are configured to display the retrieved values as active links are ideal for using with WorkCenter pages.

Image: WorkCenter with a PeopleSoft query pagelet and target content as specified by a drilling URL

In the pagelet area of this WorkCenter page, the User Roles pagelet is based on the *PT_SEC_USER_ROLES* query definition, which returns the roles that are assigned to the current user. Each role name in the list is a link to the Roles component. When users click a role name such as *Documentation Specialist,* that role name value passes to the component as a search parameter. Using the value, the Roles component opens to the *Documentation Specialist* role.



Publishing Query-Based Pagelets for Use With WorkCenter Pages

You create and maintain queries using the Query Manager component.

You can use Pagelet Wizard to create pagelets based on any query. At step two of the template pagelet creation process in Pagelet Wizard, you must specify your query as the data source. At step 5, you use the link icon to access the Link Details page, where you can configure fields to use the drilling URLs and appear as links. At this step, you also specify whether a link appears in the target content area or a new window. Continue through the process to step 6, where you specify *Homepage Pagelet* or *Template*

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Pagelet as a publishing option. When you save the pagelet, Pagelet Wizard creates the pagelet and you can then assign it to the pagelet area of a WorkCenter page.

Specific details for creating and maintaining queries are located in *PeopleTools: Query*.

Specific details for using Pagelet Wizard to publish queries as template pagelets are located in this PeopleBook.

Related Links

"Drilling URL in PeopleSoft Query" (PeopleTools 8.54: Query)

Selecting a PeopleSoft Query Data Source

Specifying Data Link Details

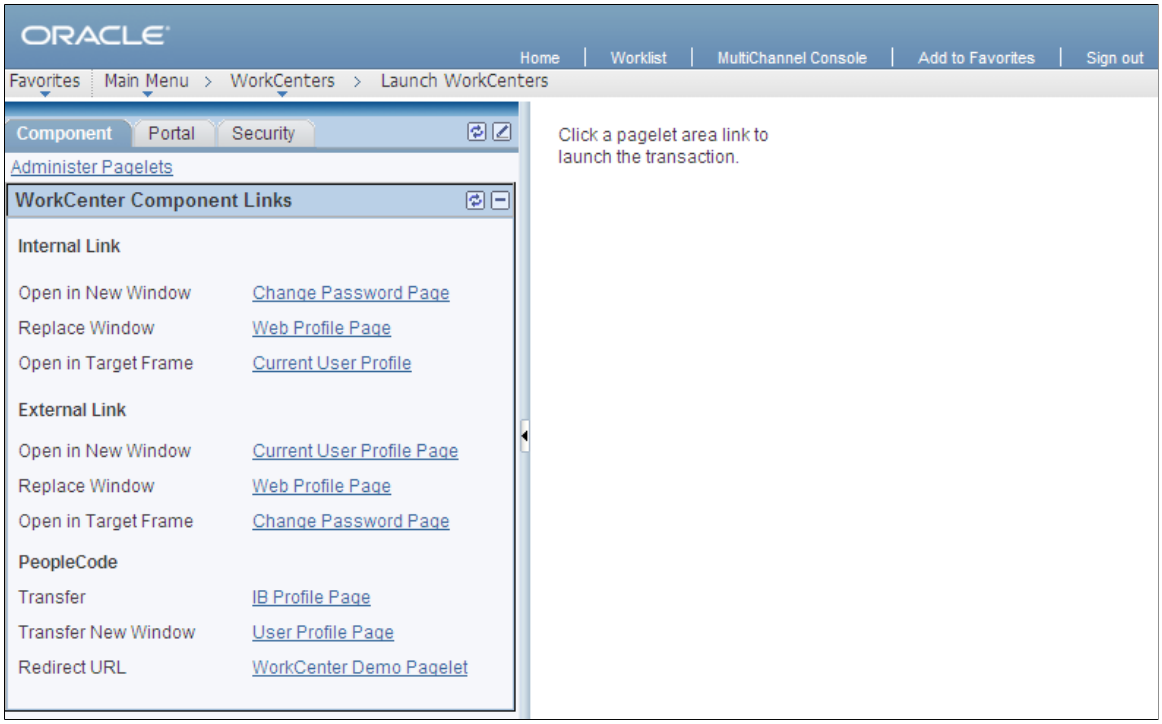
Step 6: Specifying Pagelet Publishing Options

Component-Based Pagelets

Component-based pagelets are PeopleSoft component definitions that you create and configure in PeopleSoft Application Designer and then register in the PeopleTools portal as pagelets. With minor differences, the process that you use to create, configure, and assemble the definitions that compose this type of component definition follows closely the application development process. While the *process* is similar to that which you use to create standard component definitions, some definition *properties* are different. In addition to these definition properties, other portal-related differences can exist depending on whether you use an existing component or create a new one.

Image: WorkCenter with a component-based template pagelet

This section discusses the important differences in settings between standard component definitions and components that you use in component-based pagelets. All examples in this section use an example component-based pagelet, titled *WorkCenter Component Links,* which you see in the pagelet area of this WorkCenter page:



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Record Definition Settings

Record definitions that support component-based pagelets typically are not set to the SQL table record definition type because the pagelet area is not intended to write data to tables. Create or use record definitions with a record type of SQL View or Derived/Work.

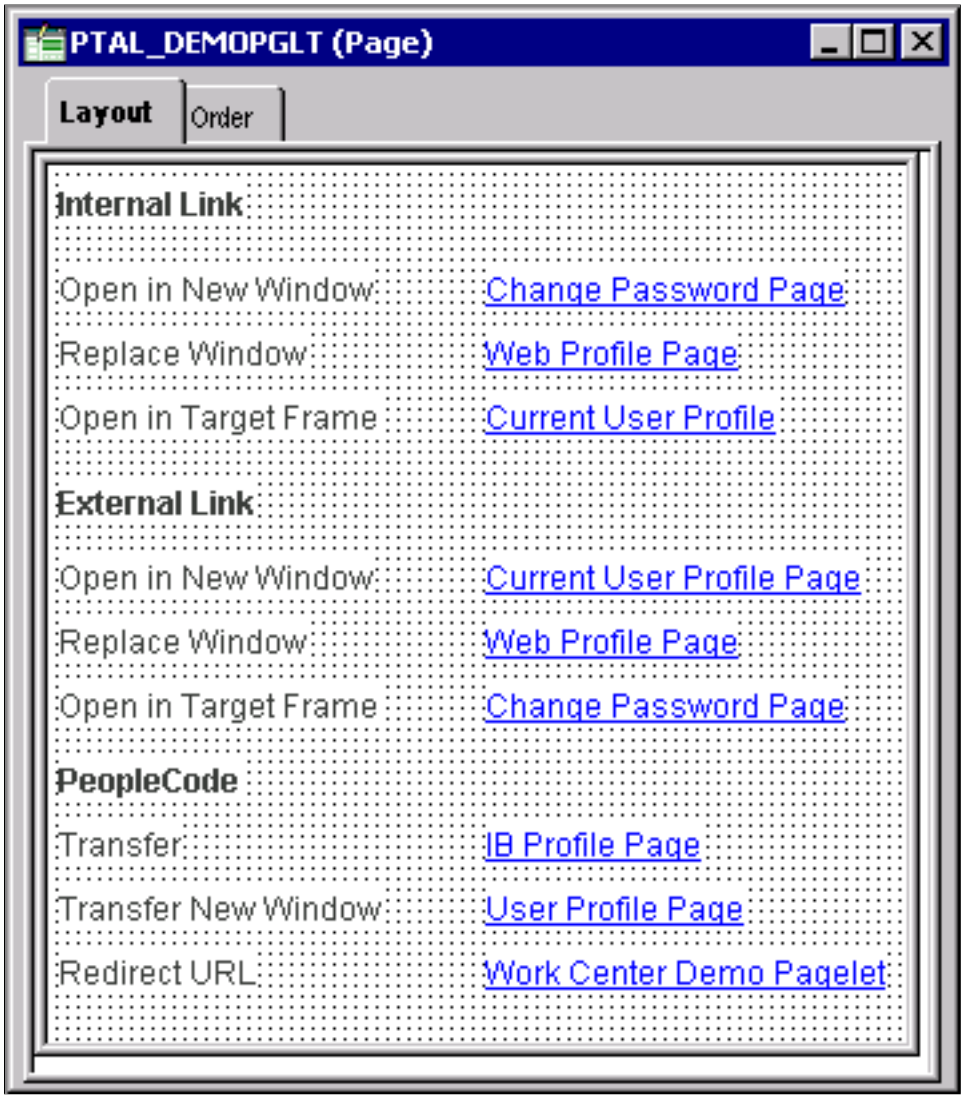
See "Setting Record Properties" (PeopleTools 8.54: Application Designer Developer's Guide).

Page Field Definition Settings

All pagelets should be designed to produce interaction between the pagelet area and the target content area. When defining pages that are part of component-based pagelets, the page developer can most easily facilitate this interaction by inserting push button/hyperlink page controls and then configuring properties that are inherent to that type of control. The developer can also write PeopleCode programs that accomplish the same pagelet area to content area interaction.

Image: Application Designer page definition for a component-based template pagelet

The WorkCenter Component Links pagelet is based on the PTAL_DEMOPGLT page definition, which includes nine push button/hyperlink controls:



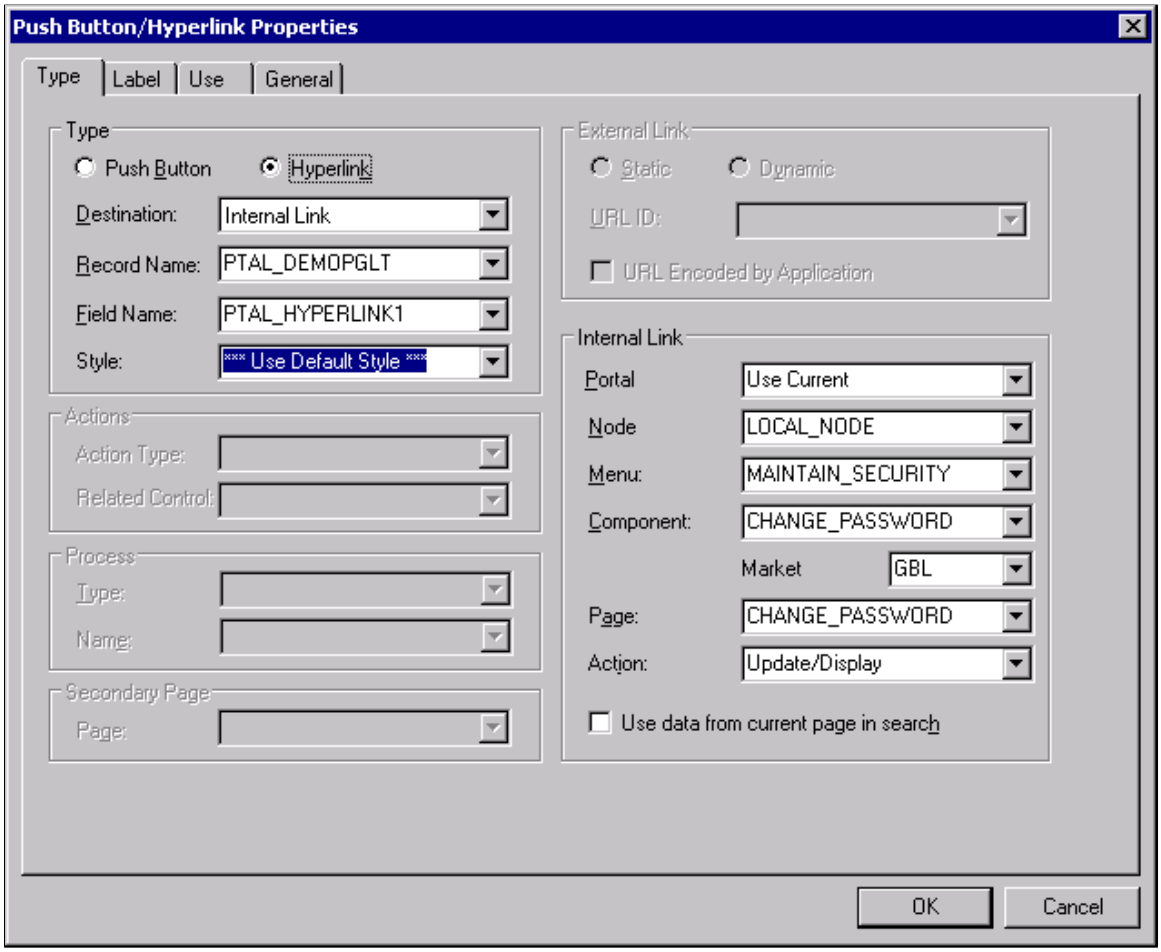
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See "Understanding Page Design and Development" (PeopleTools 8.54: Application Designer Developer's Guide).

By setting the push button/hyperlink page control Type and Field Use properties at design time, the developer determines the resource that appears when the user clicks the link and also where that the resource appears, such as in the target content area, a modal window, or a new window. Secondary pages, PeopleCode, internal links and external links only are valid destination types for push button/hyperlink controls when you use them on component-based template pagelets.

Image: Push Button/Hyperlink Properties dialog box: Type tab showing Internal Link properties

The Push Button/Hyperlink Type properties specify an internal link type. The internal link properties specify the location of the CHANGE_PASSWORD page:



See "Using Push Buttons and Links" (PeopleTools 8.54: Application Designer Developer's Guide).

Component Definition Settings

Standard components and the components that you use as component-based pagelets have different purposes, which require that the developer configure the types of components differently. The following list describes the recommended property settings for components that you use as component-based pagelets on WorkCenter pages:

- WorkCenter pages should not generate a search page or start the search process when the user clicks the content reference to access the page. The user should immediately see the initial page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select a “dummy” record—one that has no keys or contains only one row of data—as the component search record on the Use tab.

The INSTALLATION record is an example of a “dummy” search record.

- Component-based pagelets generally require no save functionality.

  Select the Disable Saving Page check box on the Use tab.

- Typically, the component should consist of only one page, but no tabs or page links should appear on the pagelet.

  Deselect the Multi-Page Navigation check boxes on the Internet tab.

- The elements that make up the tool bar and page bar are not applicable and should not appear on the page.

  Select the Disable Toolbar and Disable Pagebar check boxes on the Internet tab.

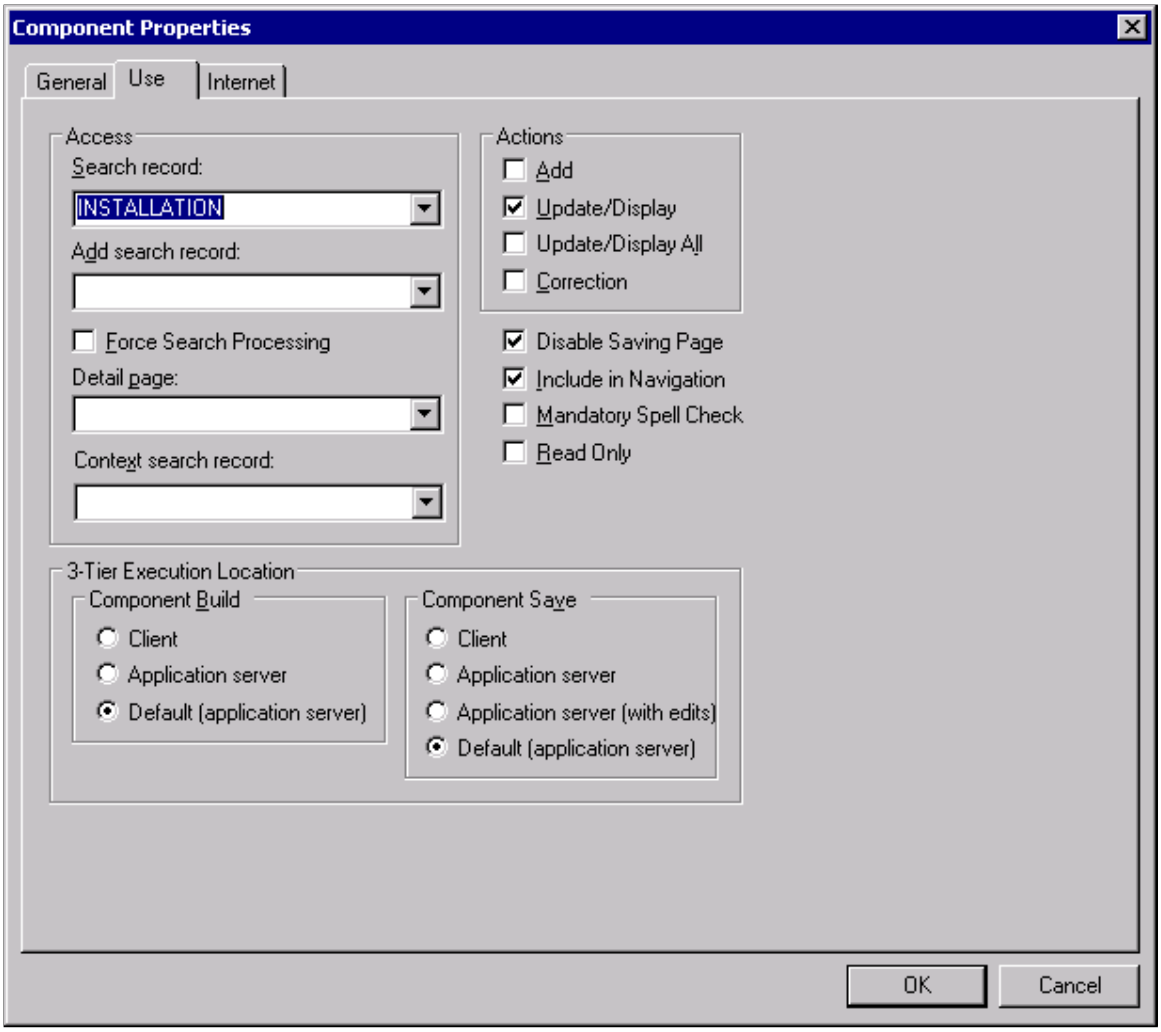
Notice the settings of the PTAL_DEMOPGLT component in this example:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The search record is *INSTALLATION* and the Disable Saving Page check box is selected:

  Image: Component Properties dialog box: Use tab settings for a component used as a component-based template pagelet

  The following example shows the Use tab settings for the Component Properties dialog box for a component-based template pagelet:

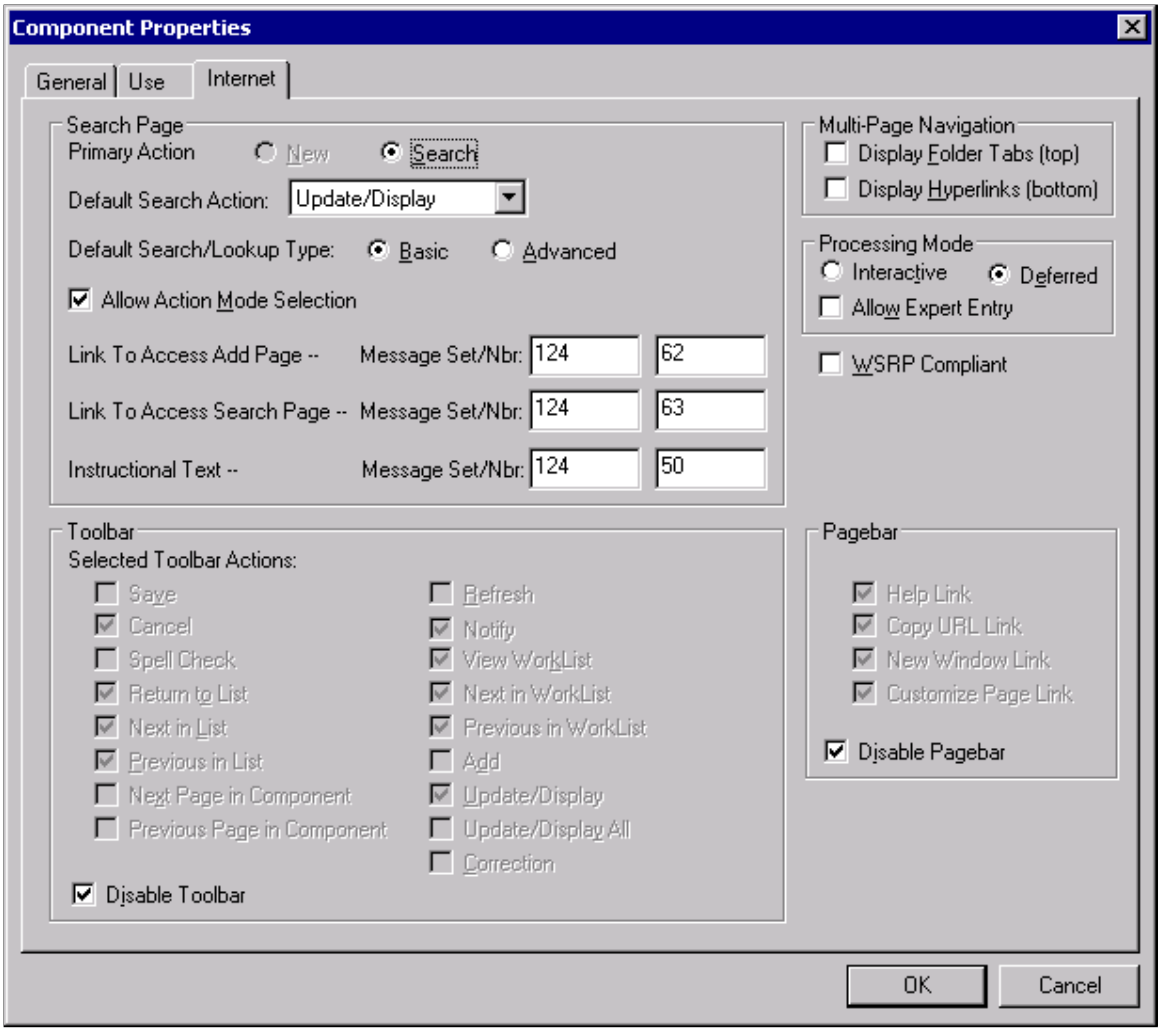


Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The Multi-Page Navigation check boxes are deselected and the Pagebar and the Toolbar are disabled:

  Image: Component Properties dialog box: Internet tab settings for a component used as a component-based template pagelet

  The following example shows the Internet tab settings for the Component Properties dialog box for a component-based template pagelet:



See "Setting Component Properties" (PeopleTools 8.54: Application Designer Developer's Guide).

Registering Components as Pagelets for Use as WorkCenter Pagelets

After creating the component definition, you must register the component as a homepage pagelet or template pagelet in the PeopleTools portal. The PeopleTools portal stores content references based on

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

their URL information. As a result, the combination of elements that make up the URL of every content reference in a portal must be unique.

Image: Content Ref Administration page for a component-based pagelet showing additional parameters

If a component is already registered as a content reference in the portal, then you must enter a “dummy” value in the Additional Parameters field when you register that same component as a template pagelet. Adding this field value ensures that the two content references are unique. A sample value that works well is *name1=value1,* as shown in this example:

See Defining Content References.

Related Links

"Main Application Development Steps" (PeopleTools 8.54: Application Designer Developer's Guide)

Understanding the WorkCenter Development and Implementation Process

The WorkCenter development and implementation process consists of these steps:

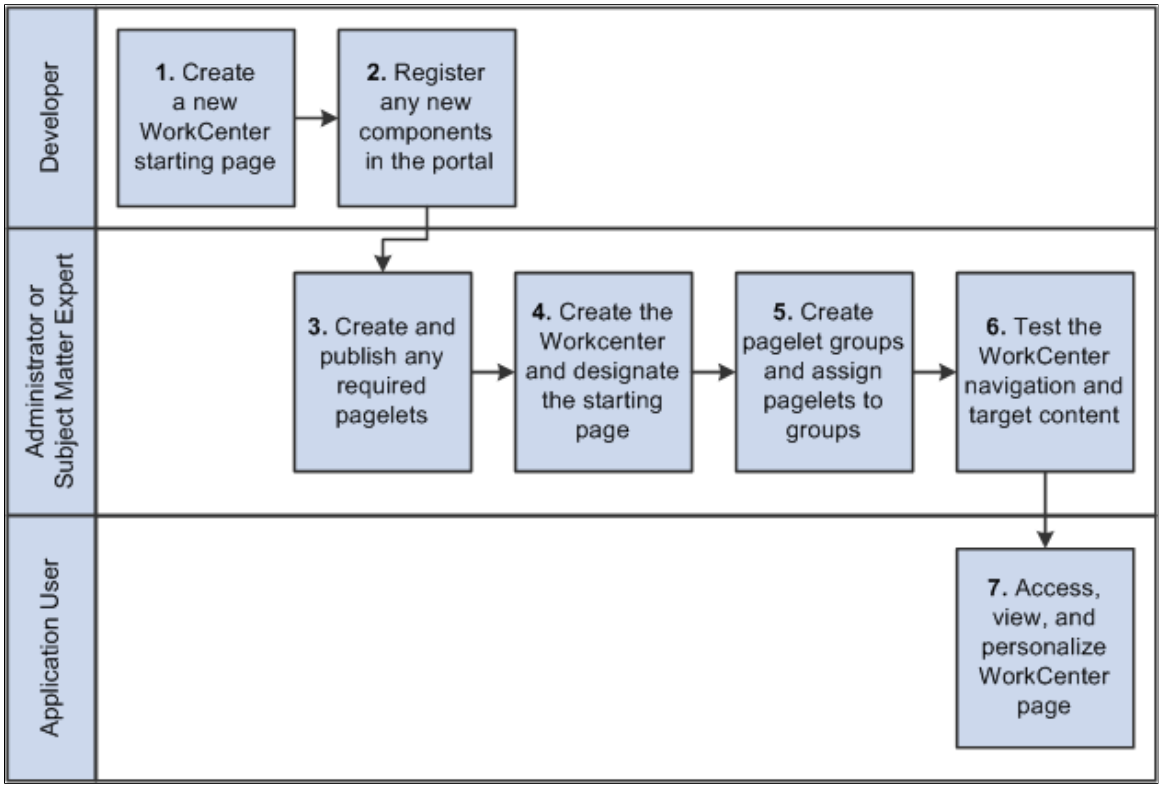
1. Create a new WorkCenter starting page, which can be based on a:
   - Content reference
   - Content reference link
   - WorkCenter dashboard
2. If necessary, register any new components (and pages) in the portal.
3. Create and publish any required pagelets.
4. Create the WorkCenter and designate the starting page.
5. Create pagelet groups and assign pagelets to groups.
6. Test the WorkCenter page including navigation and generation of target content.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

7. Once testing is complete, users can access and personalize the WorkCenter page for their use.

Image: WorkCenter page development and implementation process flow

This diagram presents an overview of the process for developing, implementing, and using WorkCenter pages:



Managing WorkCenter Pages

This section discusses how to:

- Administer WorkCenter pages.
- Select the WorkCenter starting page.
- Administer WorkCenter starting pages.
- Configure WorkCenter pages.

Administering WorkCenter Pages

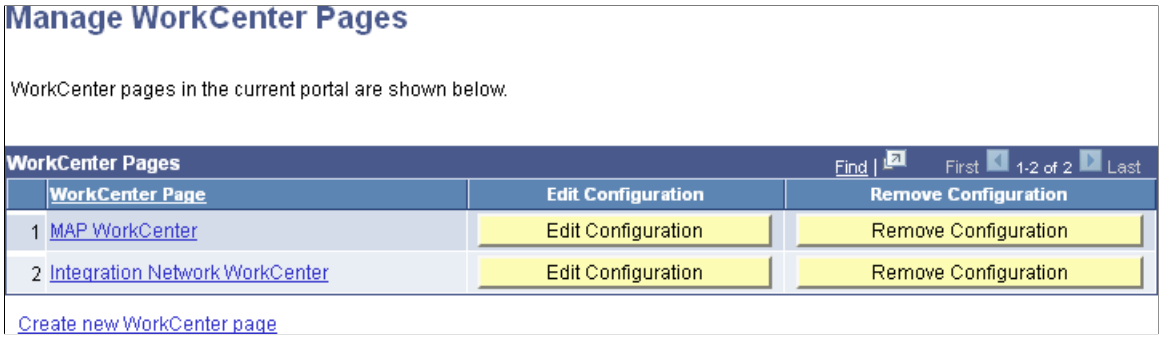
Access the Manage WorkCenter Pages page. (Select PeopleTools, Portal, WorkCenter, Manage WorkCenter Pages.)

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Create New WorkCenter Page Click to access the Select the WorkCenter Starting Page page and begin the process of creating a WorkCenter page.

Image: Manage WorkCenter Pages page

This example illustrates the fields and controls on the Manage WorkCenter Pages page. You can find definitions for the fields and controls later on this page.



WorkCenter Page Displays the content reference name as an active link. Click to access the content reference and view the WorkCenter.

Edit Configuration Click to access the Configure WorkCenter Page page.

Remove Configuration Click to delete the WorkCenter page, all pagelet groups, and all pagelet configurations associated with the content reference.

Note: Removing WorkCenter page configurations neither deletes the content reference upon which it is based nor does it delete the pagelets.

Create New WorkCenter Page Click to access the Create WorkCenter Page page to begin the process of creating a WorkCenter by selecting a WorkCenter starting page.

Selecting the WorkCenter Starting Page

To select the WorkCenter starting page:

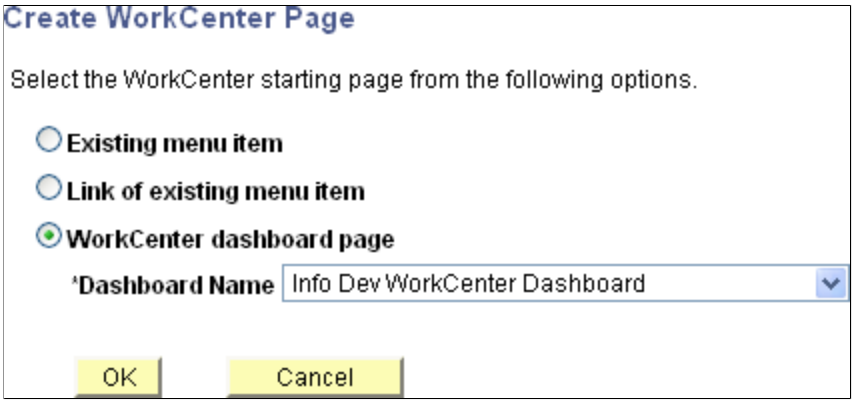
1. Select PeopleTools, Portal, WorkCenter, Manage WorkCenter Pages.
2. Click the Create new WorkCenter Page link.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The Create WorkCenter Page page is displayed:

Image: Create WorkCenter Page page

This example illustrates the fields and controls on the Create WorkCenter Page page:



3. On the Create WorkCenter Page page, select the type of starting page:

   - Existing menu item – Select this option to *convert* an existing content reference into a WorkCenter.

     Important! When users navigate to the menu item for this content reference, they will now open the WorkCenter once it is converted in this manner.

   - Link of existing menu item – Select this option to *create a link* to an existing content reference. The link will be used as the WorkCenter starting page.

     Note: When users navigate to the menu item for this content reference, they will continue to open the content, and not the WorkCenter.

   - WorkCenter dashboard page – Select this option to use an existing WorkCenter dashboard page as the WorkCenter's starting page.

     Note: When no WorkCenter dashboards exist in the system, this option is not displayed. In addition, the Dashboard Name field is displayed only after you select this option.

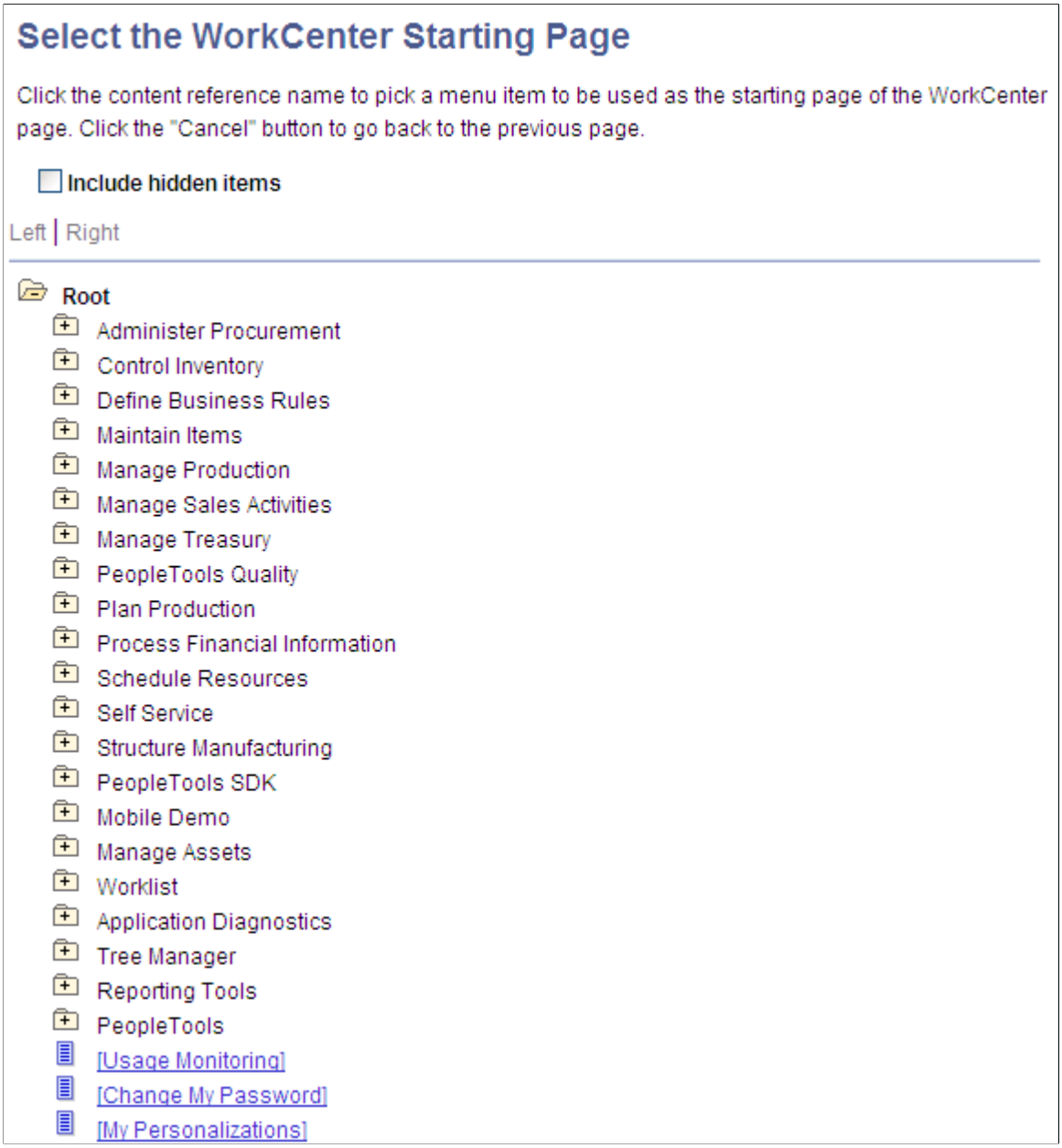
     After selecting the WorkCenter dashboard page, continue by configuring the WorkCenter. See Configuring WorkCenter Pages

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. If you selected either Existing menu item or Link of existing menu item, continue by selecting the content reference that represents that menu item:

   Image: Select the WorkCenter Starting Page page

   This example illustrates the fields and controls on the Select the WorkCenter Starting Page page:



5. Continue by:

   - If you selected Existing menu item, then configure the WorkCenter. See Configuring WorkCenter Pages.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- If you selected Link of existing menu item, then you must administer the starting page properties first. See Administering WorkCenter Starting Pages. Then, configure the WorkCenter. See Configuring WorkCenter Pages.

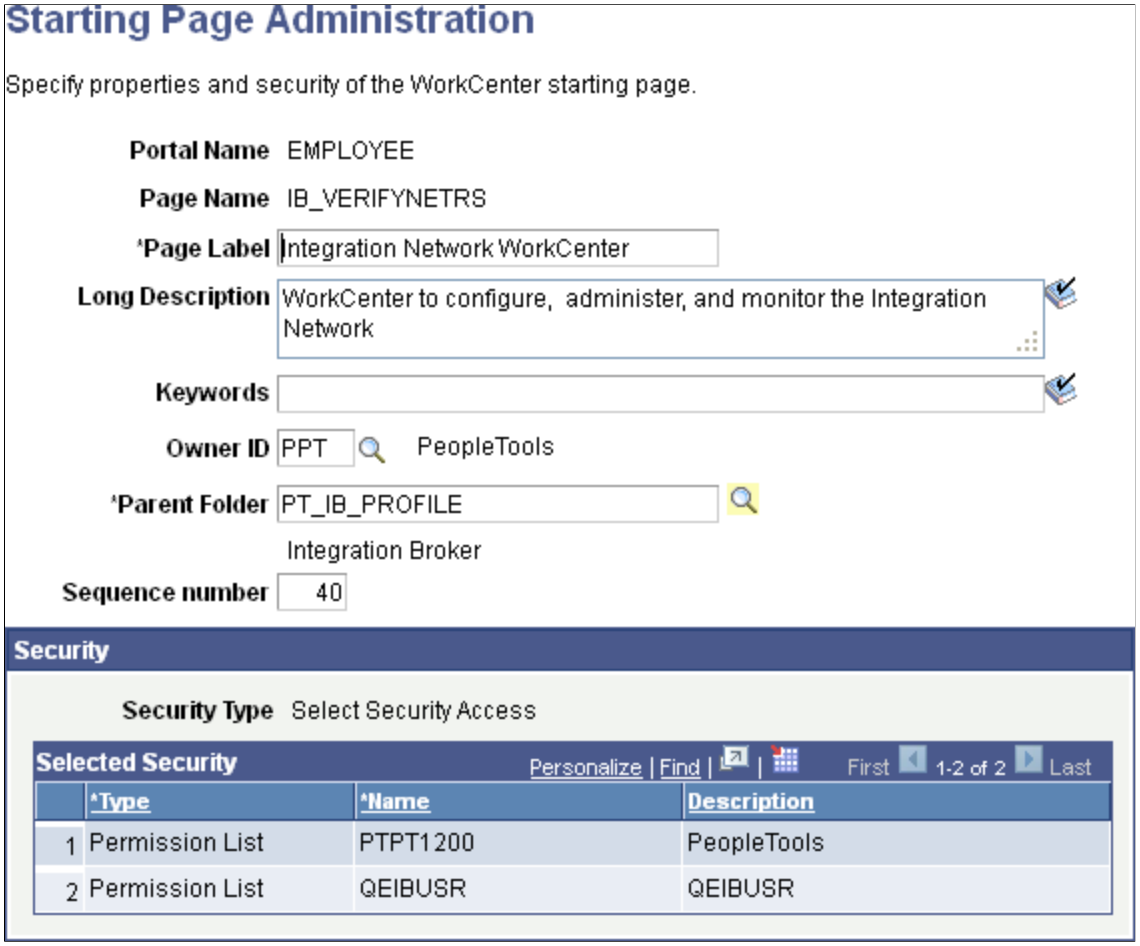
Administering WorkCenter Starting Pages

Access the Starting Page Administration page:

- On the Configure WorkCenter Page page, click the Properties button.
- When linking to an existing menu item as the WorkCenter starting page, click the content reference that represents the menu item. This page is displayed automatically

Image: Starting Page Administration page

This example illustrates the fields and controls on the Starting Page Administration page. You can find definitions for the fields and controls later on this page.



Portal Name Displays the portal in which the content reference is registered.

Page Name Enter the name for the new page when *linking* to an existing content reference only.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: Otherwise, for existing starting pages, this field is display only.

Page Label Enter the menu item title for the WorkCenter as it will appear in the portal menu and breadcrumbs.

Long Description Enter a description for the WorkCenter starting page.

Note: This value also appears in tool tip text.

Keywords Enter a keyword or keyword phrase to be stored as a content reference attribute.

Note: Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language).

See Defining Content References for more information on content reference attributes.

Owner ID Select the object owner ID.

Parent Folder Specify the parent folder for the menu item representing the WorkCenter.

Sequence number Specify a sequence number for this menu item within its parent folder.

Security

The Security group box displays the selected security settings for this content reference. When the WorkCenter starting page is a WorkCenter dashboard, the fields in the Security group box are editable. Otherwise, when the starting page is either a link or an existing content reference, these fields are display only.

Security Type Select one of the following security types:

- *Public* – Select this option to allow all users access to the content reference for this WorkCenter.
- *Select Security Access* – Select this option to specify which permission lists have access to the content reference for this WorkCenter.

Selected Security Select which permission lists have access to the content reference for this WorkCenter.

Configuring WorkCenter Pages

Access the Configure WorkCenter Page page by using one of these navigation paths:

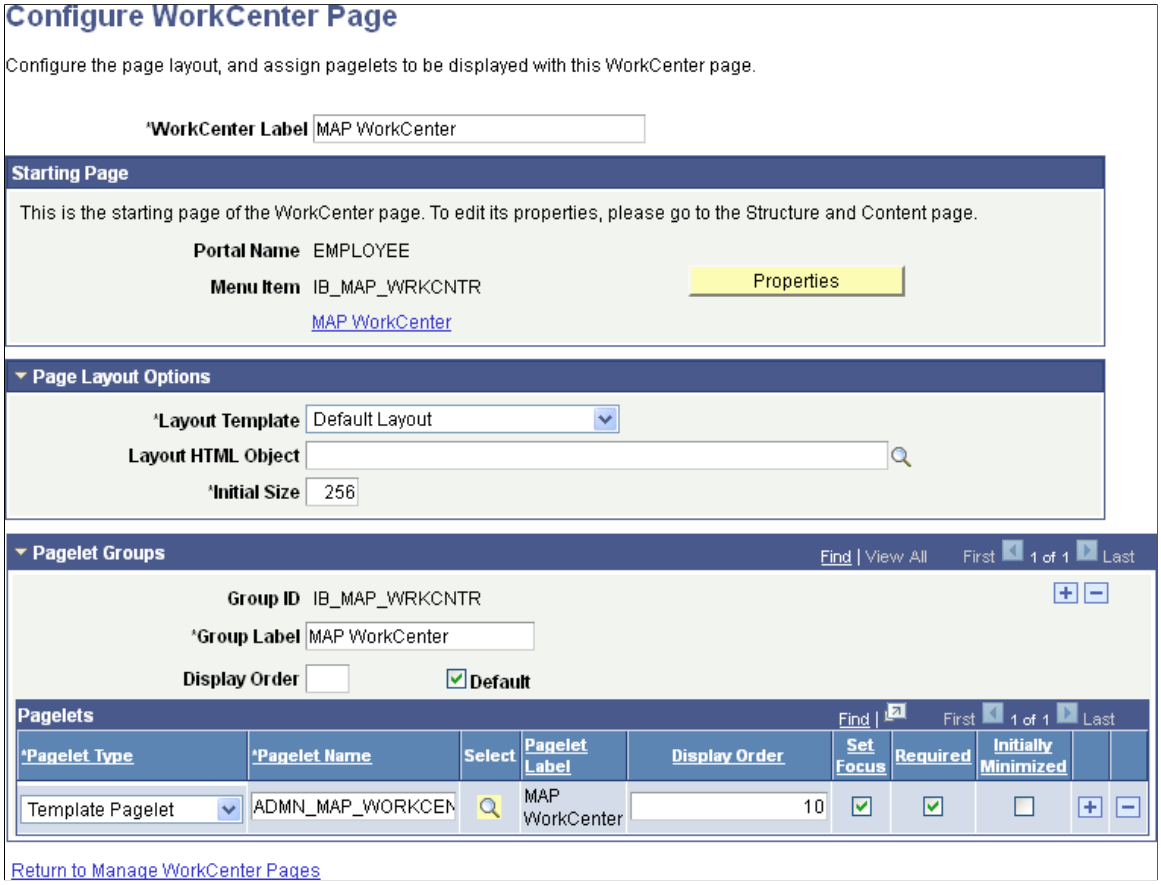
- On the Manage WorkCenter Pages page, click the Edit Configuration button for a WorkCenter.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- In the title bar of the WorkCenter pagelet area, click the Configure link.

Image: Configure WorkCenter Page page

This example illustrates the fields and controls on the Configure WorkCenter Page page. You can find definitions for the fields and controls later on this page.



WorkCenter Label

Enter the name that appears on the Manage WorkCenter Pages page and in the header of the WorkCenter pagelet area. This field can contain up to 30 characters.

Note: The default value is the name of the content reference, content reference link, or WorkCenter dashboard that you select as the starting page. This value also appears on the User Personalization page if one exists.

Return to Manage WorkCenter Pages

Click to access the Manage WorkCenter Pages page.

Starting Page

Portal Name

Displays the current portal definition name.

Menu Item

Displays the menu item name that is associated with the selected content reference or WorkCenter dashboard or the page name

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

that you specified for a content reference link on the Starting Page Administration page.

Properties Click to administer properties of the WorkCenter starting page on the Starting Page Administration page. See Administering WorkCenter Starting Pages.

<Starting Page Label> Click to access the WorkCenter.

Page Layout Options

Use these options to change the appearance and character of the WorkCenter page.

Changes that you make to the page layout options are visible after you save the page and click the Refresh button on your *browser*.

Layout Template Select a layout template from the list of all active page layout templates. The name of the default page layout template appears here when you create a new WorkCenter page.

Layout HTML Object Select a custom layout, which you define in an HTML definition in PeopleSoft Application Designer. Only HTML definitions that begin with *PTAL_LAYOUT_* appear in this prompt field.

Initial Size Enter an integer between 200 and 9999 that determines the width of the pagelet area when the user initially accesses the WorkCenter page. The default value is *256*.

Pagelet Groups

Use the fields in this scroll area to organize the pagelets in the pagelet area into tabbed groups.

Group ID Enter a value that identifies the group. Each value must be unique to the WorkCenter page and can contain no more the 20 characters. After you initially add this value, the field becomes display-only. This field is required.

Group Label Enter a meaningful name of up to 20 mixed-case characters for the pagelet group. The label appears on a tab in the pagelet area and is visible to the user. This field is required.

Display Order Enter a number between 0 and 999 to manually sequence the order in which the pagelet groups appear. This field is empty by default and pagelet groups appear in the same order as the row sort order in the Pagelet Groups scroll area. If you omit the Display Order value then the system considers it to be *0* and appear before groups for which the field contains a value.

Default Select to display the group by default when the WorkCenter page is initially accessed. Only one group can be the default.

Note these points about pagelet groups:

- If you create only one pagelet group, no tab appears to the user.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- If you delete a pagelet group, the application deletes all subordinate pagelet data and personalizations.

Pagelets

Use the fields in this grid to assign the pagelets to the group.

Pagelet Type	Select *Template Pagelet* or *Homepage Pagelet.*
Pagelet Name	Select the pagelet from a list of registered pagelets. Only registered pagelets appear in the list. A pagelet can be assigned to *only one* group on a WorkCenter page.
Select	Click the Select button and select the pagelet from the Pagelet Search page to add to the WorkCenter definition. See Searching for and Importing Pagelets for more information on searching for pagelets on local and remote nodes.
Pagelet Label	Displays the label that appears in the pagelet header.
Display Order	Enter a number between 0 and 999 to specifically determine the position of the pagelets in the group. This field is empty by default and pagelet groups appear in the same order as the row sort order in the Pagelets grid. Pagelets for which you leave the Display Order field blank are considered to have a value of *0* and appear before groups for which the field contains a value.
Set Focus	Select to move the cursor to the pagelet when the user initially accesses the WorkCenter page. You can select this check box for only *one* pagelet in the same group.
Required	Select to prevent the application user from hiding the pagelet in the group. The user can still minimize the pagelet.
Initially Minimized	Select to display the pagelet in the minimized state when the user initially accesses the WorkCenter page.

Creating and Managing WorkCenter Dashboards

This section discusses how to:

- Manage WorkCenter dashboards
- Create WorkCenter dashboards
- Configure the layout of WorkCenter dashboards
- Select a WorkCenter dashboard as a starting page

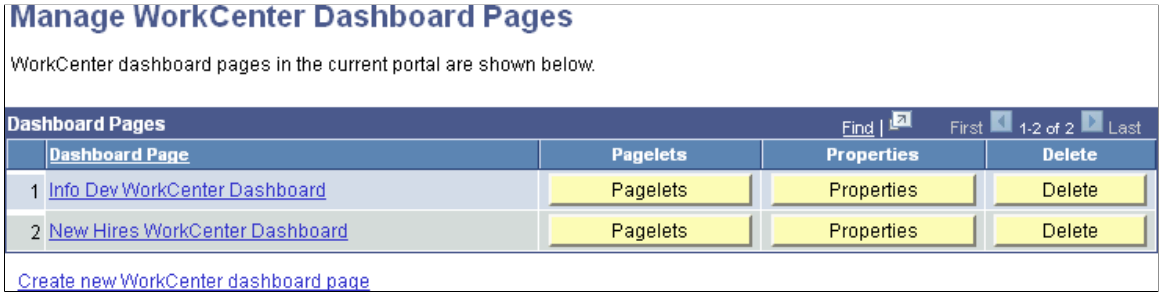
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Managing WorkCenter Dashboards

Access the Manage WorkCenter Dashboard Pages page. (Select PeopleTools, Portal, WorkCenter, Manage WorkCenter Dashboards.)

Image: Manage WorkCenter Dashboard Pages page

This example illustrates the fields and controls on the Manage WorkCenter Dashboard Pages page. You can find definitions for the fields and controls later on this page.



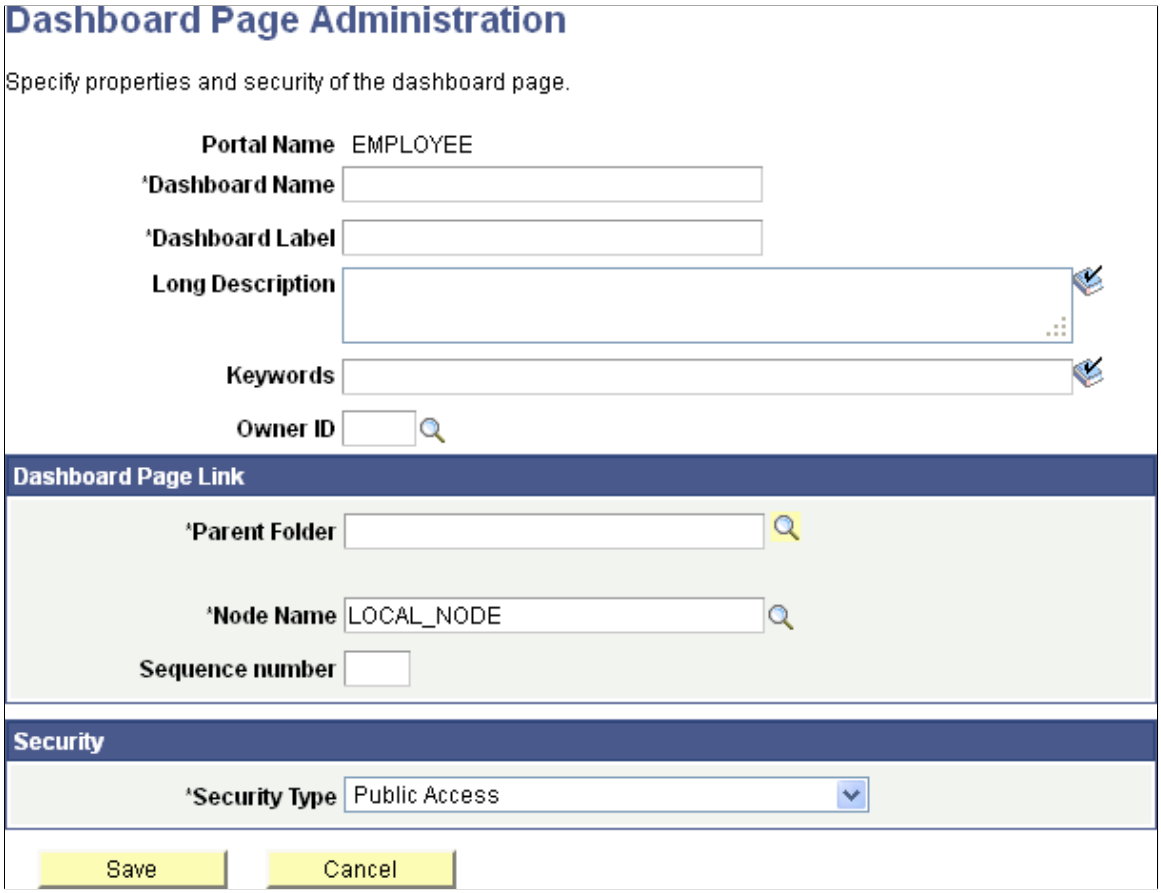
Dashboard Page	Displays the content reference name as an active link. Click to access the content reference and view the WorkCenter dashboard page.
Pagelets	Click the Pagelets button to configure the layout of the dashboard. See Configuring the Layout of WorkCenter Dashboards.
Properties	Click the Properties button to administer the properties for the dashboard page. See Creating WorkCenter Dashboards.
Delete	Click the Delete button to delete the dashboard page.
Create new WorkCenter dashboard page	Click the link to create a new WorkCenter dashboard. See Creating WorkCenter Dashboards.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating WorkCenter Dashboards

Image: Dashboard Page Administration page

This example illustrates the fields and controls on the Dashboard Page Administration page. You can find definitions for the fields and controls later on this page.



Portal Name

Displays the portal in which the content reference is registered.

Dashboard Name

Enter the name for the new WorkCenter dashboard page.

Dashboard Label

Enter the menu item title for the WorkCenter dashboard as it will appear in the portal menu and breadcrumbs.

Long Description

Enter a description for the WorkCenter dashboard page.

Note: This value also appears in tool tip text.

Keywords

Enter a keyword or keyword phrase to be stored as a content reference attribute.

Note: Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language).

See Defining Content References for more information on content reference attributes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Owner ID Select the object owner ID.

Dashboard Page Link

Parent Folder Specify the parent folder for the menu item representing the WorkCenter dashboard.

Node Name Specify the portal host node for the WorkCenter dashboard.

Sequence number Specify a sequence number for this menu item within its parent folder.

Security

The Security group box displays the selected security settings for this content reference.

Security Type Select one of the following security types:

- *Public* – Select this option to allow all users access to the content reference for this WorkCenter dashboard.
- *Select Security Access* – Select this option to specify which permission lists have access to the content reference for this WorkCenter.

Selected Security Select which permission lists have access to the content reference for this WorkCenter dashboard.

Configuring the Layout of WorkCenter Dashboards

To specify the layout of WorkCenter dashboards:

1. On the Manage WorkCenter Dashboard Pages page, click the Pagelets button for a WorkCenter dashboard.
2. On the Tab Content page, select the pagelets that you want to appear on the WorkCenter dashboard. See Specifying the Pagelets That Appear on Dashboard Pages.
3. On the Tab Layout page, arrange the layout of the pagelets. See Arranging the Pagelets on Dashboard Pages.

Note: For WorkCenter dashboards, only the two-column layout is available.

Selecting a WorkCenter Dashboard as a Starting Page

When a WorkCenter dashboard page is created, it is effectively already the starting page for its own WorkCenter, although it cannot yet be managed on the Manage WorkCenter Pages page.

From the Manage WorkCenter Pages page, you can create a new WorkCenter and designate a WorkCenter dashboard as the starting page for its own WorkCenter. Additionally, you can designate a WorkCenter dashboard as the starting page for a different WorkCenter. See Selecting the WorkCenter Starting Page.

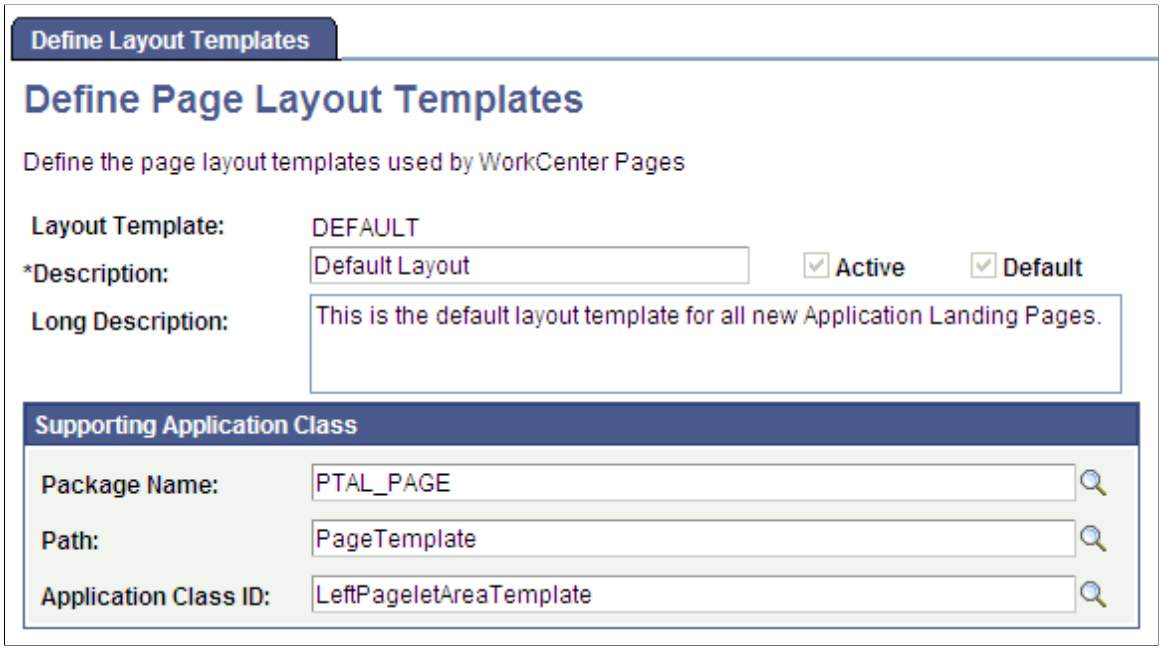
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Defining WorkCenter Page Layout Templates

Access the Define Page Layout Templates page (Select PeopleTools, Portal, WorkCenter Pages, Define Page Layout Templates.)

Image: Define Page Layout Templates page

This example illustrates the fields and controls on the Define Page Layout Templates page. You can find definitions for the fields and controls later on this page.



Layout Template View the template ID, which is the name as entered on the Add New Value page. This field is display-only.

Description Enter text that briefly describes the purpose of the layout template. You can enter up to 30 characters.

Active Select to enable the template. If you select this check box, you must enter the supporting application class. This check box is disabled if the template is designated as the default template because the default template must be active.

Default Select to specify the template as the default template for all new WorkCenter pages. If you select this check box, you must designate the template as active.

The application allows only one default template. To replace the current default template, you must create a new template and designate it as the active, default template. When you save the page, the application makes the Default and Active check boxes display-only.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Supporting Application Class

After you define the application class that creates the WorkCenter template in PeopleSoft Application Designer, you enter the class-identifying information in these fields:

Package Name

Enter the application class package name that contains the PeopleCode that defines the template.

Path

Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method that defines the template. If the class is defined in the top-level package, the Path lookup search results display a colon. Select the colon.

Application Class ID

Enter the name of the application class that determines the layout of the page. The class must exist in the application package and subpackage (path) that you specify.

LeftPageletAreaTemplate and *RightPageletAreaTemplate* application classes come with all PeopleSoft applications.

Note: If the combination of the preceding fields creates an invalid application class path when you attempt to save the component, then the system displays an error message and prevents you from saving an invalid template. The error message specifies alternatives that you can take to eliminate the condition that causes the error.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 19

Developing and Deploying Activity Guides

Understanding Activity Guides

This section provides an overview of:

- Activity guides.
- Activity guide templates and instances.
- Action items.
- Activity guide pagelets.
- A procedure for developing and deploying activity guides.
- Using the Setup Activity Guide WorkCenter.
- Prerequisites.
- Limitations.

What Is an Activity Guide?

Activity guides are a PeopleTools feature that allow you to define guided procedures for a user or group of users to complete. Users are presented with a list of actions, or tasks, that need to be completed to finish the procedure. Activity guides can be useful in a number of procedural scenarios. For example:

- The on-boarding of a new employee.
- Benefits enrollment.
- A life change event, such as a marriage or the birth of a child.
- Period end closing.
- System or feature configuration.

An activity guide lists the tasks (referred to as action items) that need to be completed by a user or group of users. These tasks are linked to specific local or remote transactions or external URLs. The tasks can be assigned to a specific user by user ID or to a group of users by PeopleTools role. Activity guides provide a hierarchical organization of sequential, and optionally dependent, tasks.

Typically, activity guides are generated from reusable definitions known as activity guide templates. These activity guide templates and action items are defined and managed through pages available under the PeopleTools, Portal menu. At run time, an activity guide instance can be instantiated from a template by providing context data. The state of the entire procedure as well as that of individual tasks is maintained between sessions, allowing a user to return later to complete the activity.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Important! PeopleTools activity guides are very different from workflow activity guides that are created from workflow activities within PeopleSoft Application Designer. See "Understanding Workflow Activity Guides" (PeopleTools 8.54: Workflow Technology) for more information on workflow activity guides.

Activity Guide Templates and Instances

Typically, activity guides are defined by the properties and characteristics of activity guide templates. An *activity guide template* provides a reusable definition of the important aspects of an activity guide: its properties, participants, contextual data, list of action items and to whom they are assigned, and so on. Then, at run time, an *activity guide instance* can be created from the definitions stored in the template. The instance will be generated with the contextual data that uniquely differentiates one instance from another. For example, for a benefits enrollment activity guide, the contextual data would likely include the employee ID, the country and locale (state, province, and so on) where the employee works, and any other key data required to initiate the benefits process.

While activity guide instances are typically created dynamically through user interaction with the system, they can also be created manually from a template.

Action Items

Action items are the tasks that are assigned for completion within an activity guide. These tasks can be linked to specific local or remote components, PeopleSoft queries, PeopleCode programs (an application class or an iScript), self-service process definitions, or external URLs. Within an activity guide, action items are organized into an ordered list; however, in many simple activity guides, action items can be completed in a non-sequential order. Action items can be organized hierarchically with a parent (summary) action item composed of child (detail) action items. Furthermore, an action item can be set to be dependent on the status of another action item.

Action items can be created and maintained within activity guide templates or within activity guide instances. However, typically you will define action items and their details at the template level only. Once an activity guide instance is created, updates to the action items themselves occur programmatically or through actions of the user.

Activity Guide Pagelets

An activity guide pagelet is the interface that presents the activity guide and its action items to a user. The pagelet lists tasks in sequence order and can optionally include a status bar indicating completion progress.

Procedure for Developing and Deploying Activity Guides

To develop and deploy an activity guide, follow this high-level procedure:

1. Determine the tasks to be completed by the users and the order of the tasks.

2. Determine whether task groupings exist—that is, summary tasks serve as parents to detailed tasks.

3. Determine which PeopleSoft component (transaction) corresponds to each task. Alternatively, a task can be associated with a PeopleSoft query, a non-PeopleSoft URL or a process.

4. Determine which user or role is to be assigned to each task.

5. Determine whether each task requires any pre-processing or post-processing logic.

   For example, post-processing logic is needed to change the status of an action item to complete.

6. Determine whether the activity guide instance is to be created dynamically from a template or manually from a template.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

7. Use the Setup Activity Guide WorkCenter to:
   a. Create the activity guide template.

      See Creating and Maintaining Activity Guide Templates.

   b. Create a definition for each action item in the template.

      See Creating and Maintaining Action Items.

   c. Generate the activity guide pagelet.

      See Deploying Activity Guides.

   d. Configure a WorkCenter for the activity guide.

      See Configuring a WorkCenter Page for the Activity Guide.

8. (Optional) Generate an activity guide instance from the template.

   See Creating New Activity Guide Instances Manually.

Using the Setup Activity Guide WorkCenter

Use the Setup Activity Guide WorkCenter to maintain an activity guide template definitions, maintain the action item definitions in the template , and to configure the deployment of the activity guide.

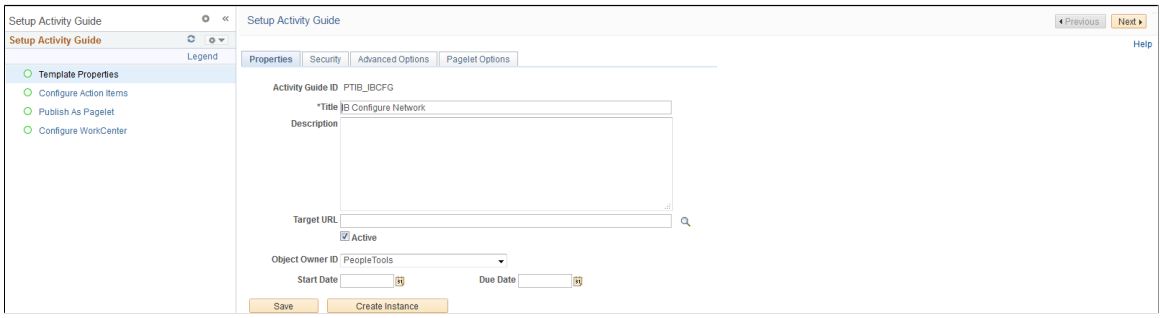
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

1. PeopleTools, Portal, Activity Guide, Manage Templates
2. On the Manage Activity Guide Templates page do one of the following:
   - Click the Create Template button to create a new activity guide template and open the WorkCenter.
   - Click the Properties link for an existing template to open the WorkCenter on the Template Properties task.
   - Click the title link for an existing template to open the WorkCenter on the Configure Action Items task.

Image: Setup Activity Guide WorkCenter

This example illustrates the Setup Activity Guide WorkCenter and the tasks in the Setup Activity Guide pagelet.



The Setup Activity Guide pagelet (itself an activity guide) presents the four tasks you must complete to define and deploy an activity guide:

1. Template Properties –

   Use the Template Properties task to define the activity guide template including its properties, members and privileges, advanced options, and pagelet options. See Creating and Maintaining Activity Guide Templates.

2. Configure Action Items –

   Use the Configure Action Items task to create a definition for each action item in the template. See Creating and Maintaining Action Items.

3. Publish as Pagelet –

   Use the Publish as Pagelet task to generate the activity guide pagelet. See Generating an Activity Guide Pagelet.

4. Configure WorkCenter –

   Use the Configure WorkCenter task to configure a WorkCenter for the activity guide. See Configuring a WorkCenter Page for the Activity Guide.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Prerequisites

Action items can be linked to specific local or remote components, PeopleSoft queries, PeopleCode programs (an application class or an iScript), self-service process definitions, or external URLs. Except for self-service process definitions, these links are made through the PeopleSoft Related Content Framework. You must create a related content service definition for each action item first, before you can complete the action item definition.

Related Links

Defining Related Content Services

Limitations

Please note the following limitations on developing and using activity guides:

- If the user accesses and updates a transaction directly (for example, through standard menu navigation) rather than through the action item link in the activity guide pagelet, then any status change effected there will not be reflected in the activity guide pagelet.
- If an action item is linked to an external URL (that is, a non-PeopelSoft URL), then you must provide a custom button to allow the user to mark the action item as complete.
- Do not link an action item to a mutli-step transaction such as a transaction that represents a wizard (for example, Pagelet Wizard).
- Activity guides support a two-level hierarchy only—that is, only one level of detail is supported beneath a summary action item.
- Do not use think-time functions such as MessageBox, WinMessage, or others in the methods that define custom navigation frame buttons. See Developing Custom Navigation Buttons.
- Do not configure a remote activity guide pagelet for use in a WorkCenter defined on the local node. Instead, define both the WorkCenter and activity guide pagelet on the remote node. Then, using unified navigation or another mechanism, access the content reference for the remote WorkCenter from the local node.
- Activity guide pagelets cannot be consumed as WSRP content.

Related Links

Limitations on PeopleSoft Content for WSRP Consumption

"Understanding Restrictions on Method and Function Use" (PeopleTools 8.54: PeopleCode Developer’s Guide)

Creating and Maintaining Activity Guide Templates

This section provides an overview of activity guide templates and discusses how to:

- Manage activity guide templates.
- Define activity guide template properties.
- Define activity guide template security.
- Define advanced options.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Define pagelet and navigation options.
- Copy activity guide templates.

Understanding Activity Guide Templates

Activity guide templates provide a mechanism for defining the properties, security, action items, contextual data, and other options for activity guides once. Then, individual activity guide instances can be generated from these reusable templates. Once an activity guide has been instantiated from a template, the instance cannot be associated with a different template. In addition, while the activity guide template can be modified, any active or inactive instances already generated from that template will not be altered by the modifications.

Typically,, activity guide properties, security, advanced options, and other attributes are created and maintained at the activity guide template level only. However, you can also maintain these attributes within activity guide instances; the same pages with the same fields are used in both cases. While the examples in this section illustrate creating and maintaining these attributes within an activity guide template, these examples could apply equally to activity guide instances.

Managing Activity Guide Templates

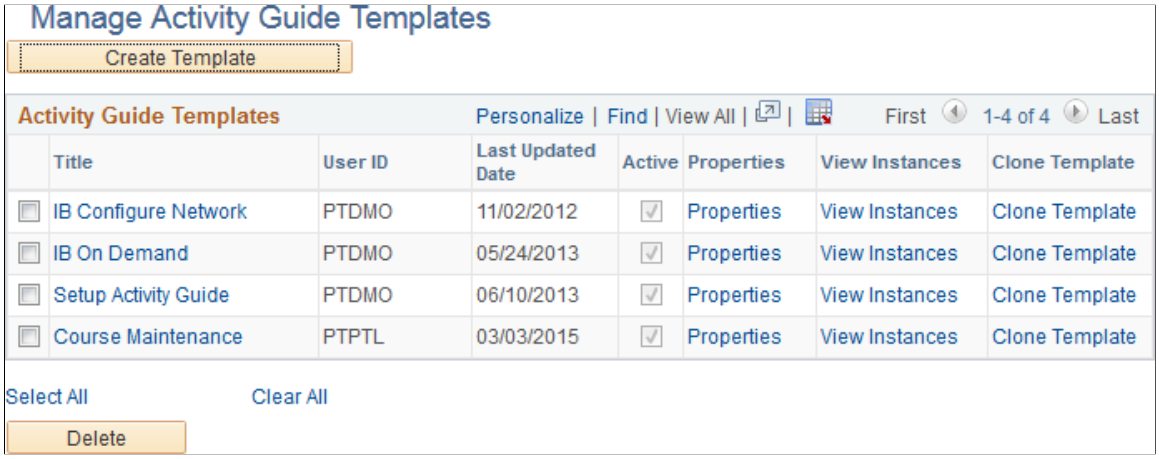
Use the Manage Activity Guide Templates page to manage the activity guide templates.

Navigation

PeopleTools, Portal, Activity Guide, Manage Templates

Image: Manage Activity Guide Templates page

This example illustrates the fields and controls on the Manage Activity Guide Templates page. You can find definitions for the fields and controls later on this page.



Create Template

Click the Create Template button to create a new activity guide template.

Clicking this button opens Setup Activity Guide WorkCenter with the Template Properties step selected.

Title

Click a title link to manage the action items for this template.

Clicking this button opens Setup Activity Guide WorkCenter with the Configure Action Items step selected.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Properties	Click the Properties link to manage the properties for this template. Clicking this button opens Setup Activity Guide WorkCenter with the Template Properties step selected.
View Instances	Click the View Instances link to view the Manage Activity Guide Instances page displaying all instances created from this template.
Clone Template	Click the Clone Template link to make a copy of this activity guide template.
Select All	Click to select all templates.
Clear All	Click to clear the selection of any templates.
Delete	Click to delete the selected templates.

Related Links

Using the Setup Activity Guide WorkCenter
Defining Activity Guide Template Properties
Creating and Maintaining Action Items
Managing Activity Guide Instances
Copying Activity Guide Templates

Defining Activity Guide Template Properties

Use the Template Properties: Properties page to define the properties for the activity guide template.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

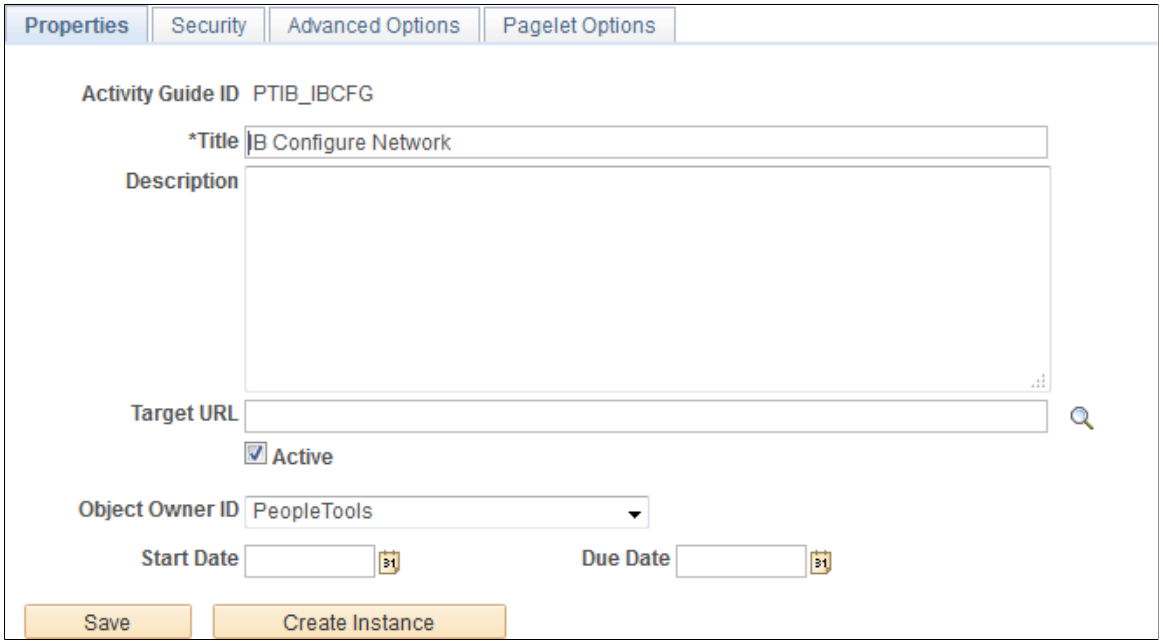
- On the Manage Activity Guide Templates page, click the Create Template button to create a new template definition.
- On the Manage Activity Guide Templates page, click the Properties link to modify an existing template.

  Note: In addition, on the Manage Activity Guide Instances page, click the Properties link for an existing instance.

- In the Setup Activity Guide WorkCenter, click the Template Properties task.

Image: Template Properties: Properties page

This example illustrates the fields and controls on the Template Properties: Properties page. You can find definitions for the fields and controls later on this page.



Activity Guide ID

Enter up to 10 characters as an identifier for the template. When you save the template, the system will fill characters up to the 10-character maximum.

Important! You must ensure that each template ID is unique.

Title

Enter a name for the template.

Description

Enter a description for the template.

Target URL

Select the URL for the WorkCenter that provides access to this activity guide. Defining the URL here provides access from the Activity Guides - In Progress homepage pagelet to the WorkCenter.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Important! To ensure that the URL is valid, select the URL from the lookup list, rather than typing or pasting a URL.

Active Select this option to make the template active. If this option is cleared, the template definition remains in the system, but is not available to be used to create new activity guide instances.

Start Date Select a start date.

Due Date Select a due date.

Note: The start and due dates specified at the template level do not have any effect on start and due dates at the action item level.

Create Instance Use the Create Instance button to create an activity guide instance from this template.

Related Links

Creating New Activity Guide Instances Manually

Working With the Activity Guides - In Progress Pagelet

Defining Activity Guide Template Security

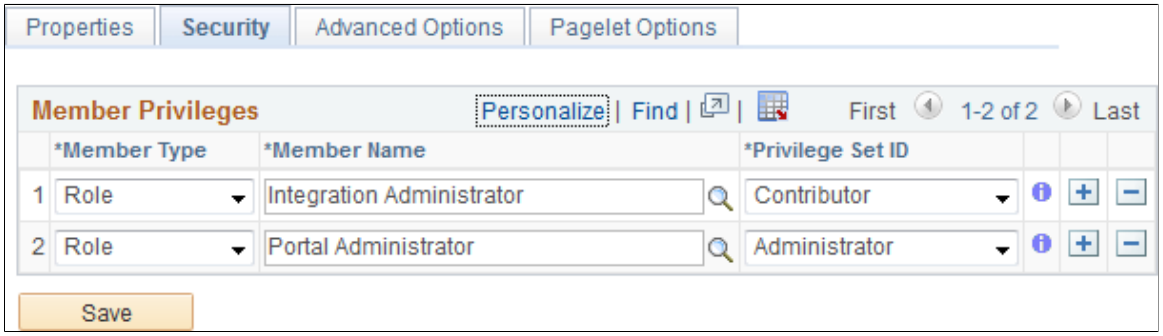
Use the Template Properties: Security page to manage participants and privileges for the template.

Navigation

1. In the Setup Activity Guide WorkCenter, click the Template Properties task.
2. Select the Security page.

Image: Template Properties: Security page

This example illustrates the fields and controls on the Template Properties: Security page. You can find definitions for the fields and controls later on this page.



Use the *Administrator* privilege set for the roles (or users) who will be responsible for maintaining this template and any activity guide instances generated from it. Use the *Contributor* privilege set for the users (or roles) who will be assigned to complete the action items within an active instance.

Member Type Select the type of participant you want to add to the activity guide template. Available values include:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Role* — Select to be able to select a role in the Member Name field. Available names are derived from PeopleSoft roles defined in the Roles component.

  See "Defining Role Options" (PeopleTools 8.54: Security Administration).

- *User* — Select to be able to select a user in the Member Name field. Available users are derived from PeopleSoft users defined in the User Profiles component.

  See "Specifying User Profile Attributes" (PeopleTools 8.54: Security Administration).

Member Name Specify the user ID or role name for the participant.

Privilege Set ID Select a privilege level you want to assign to the participant:

- *Administrator* — Administrators can make modifications to the template and any instances generated from the template including modifying properties, modifying participants, and maintaining action item definitions.

  Oracle recommends that you assign the administrator privilege to a system-defined administrative role such as Portal Administrator, PTAI_ADMINISTRATOR, and so on.

- *Contributor* — Contributors are the only participants who can complete assigned action items within an active instance that is displayed within a WorkCenter. In addition, the Activity Guides - In Progress homepage pagelet will display activity guide instances only for users who have been assigned the contributor privilege.

  Oracle recommends that you assign the contributor privilege to user ID such as an employee, a customer, a partner, and so on.

- *Viewer* — Viewers have read-only access to the activity guide instances and their action items.

  Important! The *Viewer* privilege does not have any practical usage in the current implementation.

Defining Advanced Options

Use the Template Properties: Advanced Options page to define advanced options for this template.

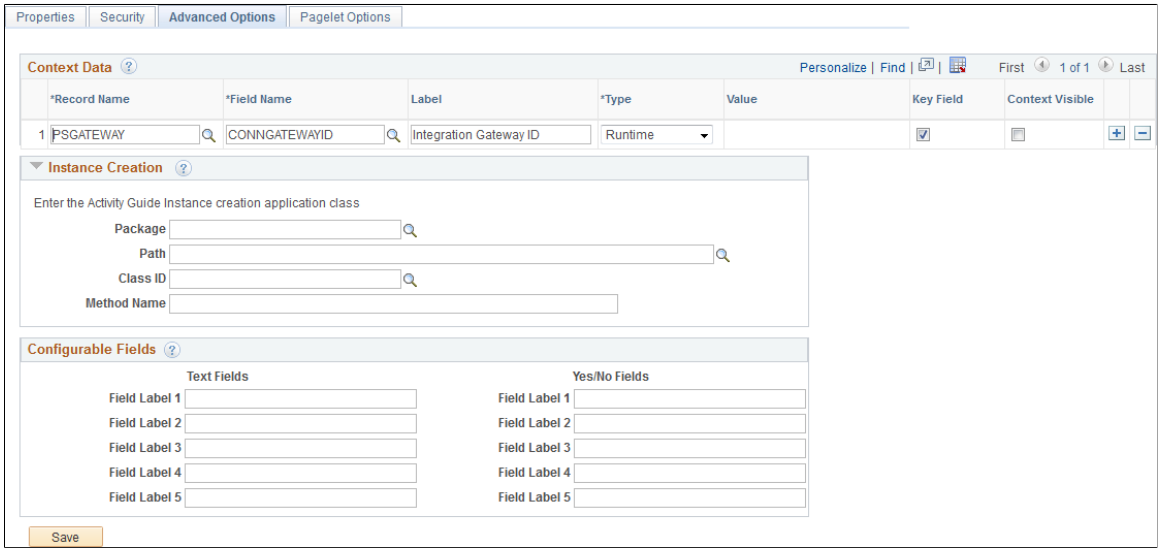
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

1. In the Setup Activity Guide WorkCenter, click the Template Properties task.
2. Select the Advanced Options page.

Image: Template Properties: Advanced Options page

This example illustrates the fields and controls on the Template Properties: Advanced Options page. You can find definitions for the fields and controls later on this page.



Context Data

Important! If multiple activity guide instances will be generated from the same template (for example, a life events activity guide template), then you must define at least one key context field in order to identify and differentiate these activity guide instances.

You can define contextual data that will be passed to the activity guide instance when it is created. Contextual data defined as key fields will be used to uniquely identify each activity guide instance. For example, if the employee ID were defined as a key contextual field, then each activity guide instance will be created for a specific employee ID. Contextual data defined as non-key fields is not used to differentiate an instance, but can be passed to the instance nevertheless. For example, non-key data can be displayed in the activity guide pagelet by specifying the Context Visible option for the field. In this way, instead of displaying the employee ID, you could display the employee name in the pagelet.

Record Name Specify the record name that is the source for the contextual field.

Field Name Specify the name of the source contextual field.

Label Enter an optional label for the contextual field. This label can be displayed along with the data value in the activity guide pagelet on the WorkCenter page.

Type Select one of the following context types:

- *Fixed Value*: Select to enter a fixed value as the context data.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Runtime*: Select to have the system determine the context data value at runtime.
- *System Variable*: Select to specify a system variable as the context data value. The value of the system variable is determined at runtime.

Value

Enter the value when the context type is fixed value. Alternatively, select the system variable from the look up list when the context type is system variable.

Note: No value is entered when the context type is runtime.

Key Field

Specify whether the contextual field is a key field.

Note: Keys fields are used to uniquely identify an activity guide instance.

Context Visible

Specify whether to display the contextual data and labels in the activity guide pagelet on the WorkCenter page.

Instance Creation

You can identify a method that will be executed whenever an activity guide instance is created from this template. When an instance is created from the template, the instance properties and action item details are initially set based on the template definitions. Then, use this method to initialize instance properties, participants, and action item properties (such as status, due date, assignment, and so on), or even to add or delete action items.

Package

Enter the name of the application package that contains the method to be invoked at instance creation.

Path

Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Class ID

Enter the name of the application class that contains the method to be invoked at instance creation.

Method Name

Enter the name of the method to be invoked to initialize the activity guide instance.

Configurable Fields

You can also define labels for fields that will appear on the Related Data page for an action item. In the action item definition, use these additional fields to associate data that will be used by the related content service at run time.

For example, a configurable field could be used to pass a run control ID to the related content service (and its underlying component, a run control page). A different action item definition—in the same template even—could be defined to supply a different run control ID to the same related content service.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Text Fields Enter the label for up to five configurable text fields.

Yes/No Fields Enter the label for up to five configurable yes/no fields.

Related Links

Initializing an Activity Guide Instance

Defining Pagelet and Navigation Options

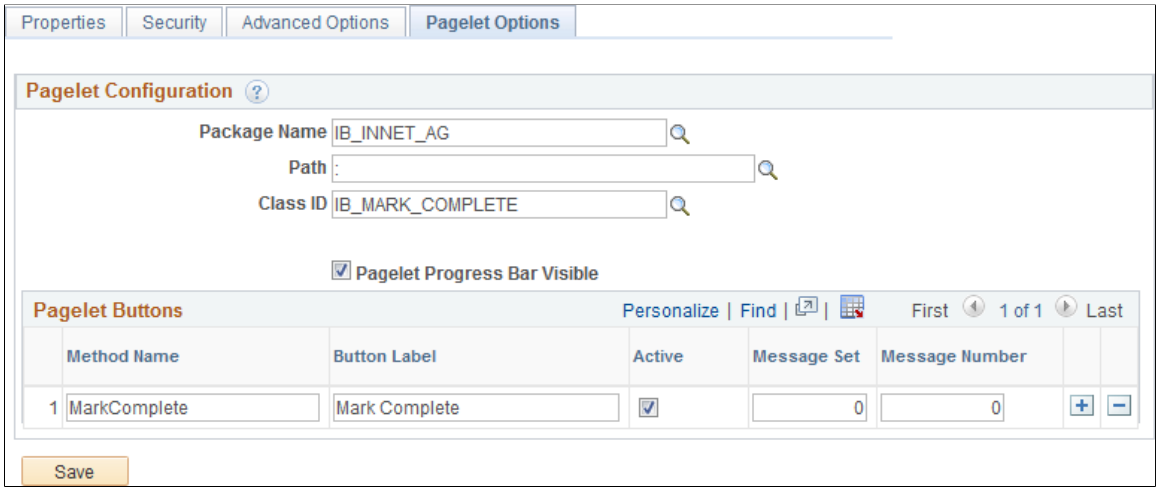
Use the Setup Activity Guide - Pagelet Options page to add custom navigation frame buttons and to define pagelet properties including pagelet pre-processing.

Navigation

1. In the Setup Activity Guide WorkCenter, click the Template Properties task.
2. Select the Advanced Options page.

Image: Template Properties: Pagelet Options page

This example illustrates the fields and controls on the Template Properties: Pagelet Options page. You can find definitions for the fields and controls later on this page.



Note: The navigation frame always includes Next and Previous buttons. Oracle recommends that you add no more than two additional buttons to the navigation frame.

Package Enter the name of the application package that contains the method (or methods) to be invoked to create custom navigation frame buttons.

Path Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Class ID Enter the name of the application class that contains the method (or methods) to be invoked to create custom navigation frame buttons.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

In addition to or in lieu of navigation frame buttons, this application class can define a pagelet pre-processing method that will be executed every time the pagelet is loaded or reloaded in the WorkCenter pagelet area. This method *must* be named PageletPreProcess.

See Performing Pre-Processing for the Activity Guide Pagelet for an example.

Pagelet Progress Bar Visible

Indicate whether the progress bar will be displayed in the activity guide pagelet on the WorkCenter page.

Pagelet Buttons

Method Name

Enter the name of the method to be invoked to create the additional navigation button.

Important! The method named here *must* exist in the selected class. No validation is performed to ensure this is the case when the template definition is saved.

Important! Mark Complete is a special type of custom navigation button that can be implemented to manually mark an action item as complete. This method *must* be named MarkComplete. After the MarkComplete method is executed, the system will reload the activity guide pagelet in the WorkCenter pagelet area.

Button Label

Enter the label for the button as it will appear in the navigation frame.

Active

Select this option to retain the button definition as active.

Message Set

Specify the message set containing an optional message to be displayed after the user clicks the button.

Message Number

Specify the message number containing an optional message to be displayed after the user clicks the button and before invoking the specified method.

For example, if the user clicks a Cancel button to cancel an activity guide instance, this could display a confirmation message.

Related Links

Developing Custom Navigation Buttons

Copying Activity Guide Templates

Use the New Activity Guide Template Details page to create a copy of the selected activity guide template.

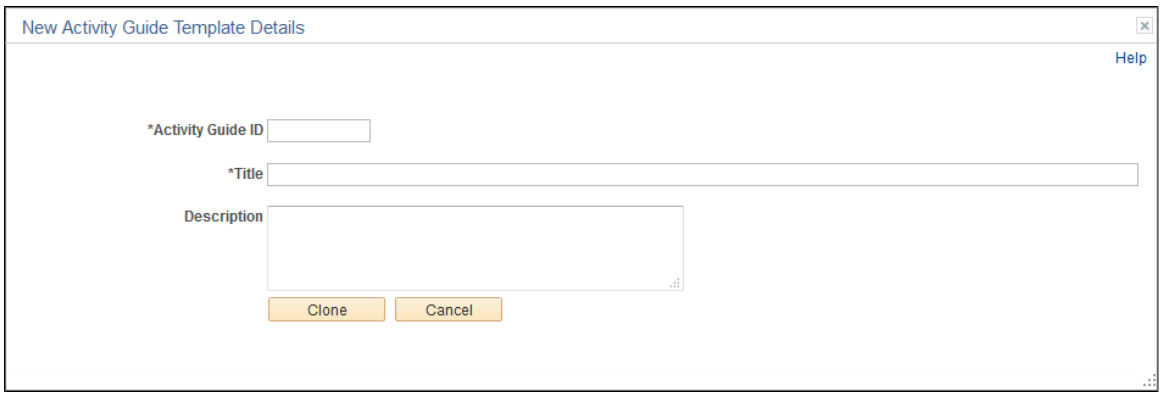
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

1. PeopleTools, Portal, Activity Guide, Manage Templates
2. Click the Clone Template link for an activity guide template.

Image: New Activity Guide Template Details page

This example illustrates the fields and controls on the New Activity Guide Template Details page. You can find definitions for the fields and controls later on this page.



Activity Guide ID	Enter an unique identifier for the template. When you save the template, the system will fill characters up to the 10-character maximum.
Title	Enter a name for the template.
Description	Enter a description for the template.
Clone	Click the Clone button to create a new activity guide template.

Creating and Maintaining Action Items

This section provides an overview of creating and maintaining action items and describes how to:

- Maintain the list of action items.
- Define action item details.
- Create a component-based service definition.
- Configure related data.
- Configure additional context data.
- Select a self-service process.
- Configure component interface parameters.

Understanding Creation and Maintenance of Action Items

Action items are the tasks that are assigned for completion within an activity guide. These tasks can be linked to specific local or remote components, PeopleSoft queries, PeopleCode programs (an application class or an iScript), self-service process definitions, or external URLs. Within an activity guide, action

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

items are organized into an ordered list; however, in many simple activity guides, action items can be completed in a non-sequential order. Action items can be organized hierarchically with a parent (summary) action item composed of child (detail) action items. Furthermore, an action item can be set to be dependent on the status of another action item.

Except for self-service process definitions, these links are made through the PeopleSoft Related Content Framework. You must create a related content service definition for each action item first, before you can complete the action item definition.

Typically, action items are created and maintained at the activity guide template level only. However, you can also create action items within activity guide instances; the same pages with the same fields are used in both cases. Once an activity guide instance is created, updates to the action items themselves would occur programmatically or through actions of the user rather than through these administrative pages. While the examples in this section illustrate creating and maintaining action items within an activity guide template, these examples could apply equally to activity guide instances.

Sequence Numbers

You must designate a sequence number for each action item, which determines its order within its designated level—that is, at the root level or as a detail action item beneath a summary item. In addition, if you want to make an action dependent on another action item, that action item must be a predecessor —that is, it must have a lower sequence number than the dependent action item. Each sequence number must be unique within an activity guide template or instance.

While sequence numbers define the order in which action items are presented to a user, they do not in and of themselves dictate that tasks be completed in sequence. Additional attributes, such as dependencies, must be defined to ensure that tasks be completed in a specific sequence when that is required. Therefore, the simplest activity guide definitions allow for non-sequential completion of tasks.

Note: Activity guides support a two-level hierarchy only—that is, only one level of detail is supported beneath a summary action item.

With a little planning, you can specify action item sequence numbers that clearly specify the order of tasks and that allow modifications to the list at a latter date—for example, to add or delete action items. Oracle recommends that you specify initial sequence numbers in units of 10 to permit insertion of new action items into an existing sequence. For example, for a simple activity guide with no summary action items, the initial sequence would be:

```
10, 20, 30, 40 ...
```

If your activity guide contains summary as well as detail items, then increment the root-level items by 100 instead of by 10. For example:

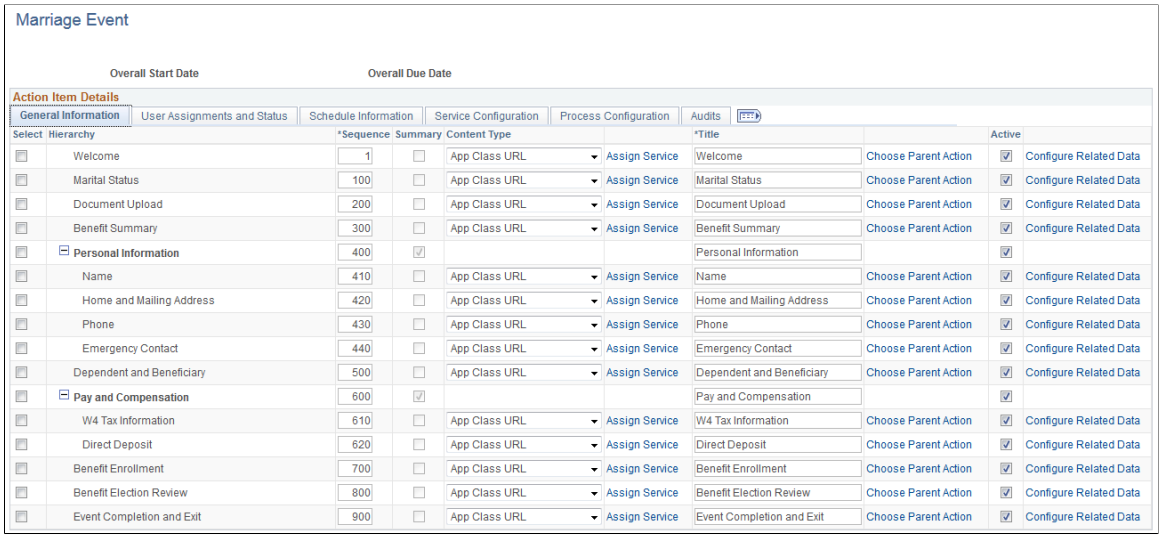
```
100 - Summary item 1
  110, 120, 130, 140, ...
200
300
400 - Summary item 2
  410, 420, 430, 440, ...
500
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Designating the initial sequence numbers in this manner will allow you to easily convert an existing root-level action item into a summary item without having to renumber the rest of the sequence.

Image: Example of summary and detail action items

The following example shows sequence numbers for summary and detail action items. In this list, items 400 and 600 are parent, or summary, items:



Marriage Event

Overall Start Date Overall Due Date

Action Item Details

General Information | User Assignments and Status | Schedule Information | Service Configuration | Process Configuration | Audits

Select	Hierarchy	*Sequence	Summary	Content Type		*Title		Active	
	Welcome	1		App Class URL	Assign Service	Welcome	Choose Parent Action	✓	Configure Related Data
	Marital Status	100		App Class URL	Assign Service	Marital Status	Choose Parent Action	✓	Configure Related Data
	Document Upload	200		App Class URL	Assign Service	Document Upload	Choose Parent Action	✓	Configure Related Data
	Benefit Summary	300		App Class URL	Assign Service	Benefit Summary	Choose Parent Action	✓	Configure Related Data
	Personal Information	400	✓			Personal Information		✓	
	Name	410		App Class URL	Assign Service	Name	Choose Parent Action	✓	Configure Related Data
	Home and Mailing Address	420		App Class URL	Assign Service	Home and Mailing Address	Choose Parent Action	✓	Configure Related Data
	Phone	430		App Class URL	Assign Service	Phone	Choose Parent Action	✓	Configure Related Data
	Emergency Contact	440		App Class URL	Assign Service	Emergency Contact	Choose Parent Action	✓	Configure Related Data
	Dependent and Beneficiary	500		App Class URL	Assign Service	Dependent and Beneficiary	Choose Parent Action	✓	Configure Related Data
	Pay and Compensation	600	✓			Pay and Compensation		✓	
	W4 Tax Information	610		App Class URL	Assign Service	W4 Tax Information	Choose Parent Action	✓	Configure Related Data
	Direct Deposit	620		App Class URL	Assign Service	Direct Deposit	Choose Parent Action	✓	Configure Related Data
	Benefit Enrollment	700		App Class URL	Assign Service	Benefit Enrollment	Choose Parent Action	✓	Configure Related Data
	Benefit Election Review	800		App Class URL	Assign Service	Benefit Election Review	Choose Parent Action	✓	Configure Related Data
	Event Completion and Exit	900		App Class URL	Assign Service	Event Completion and Exit	Choose Parent Action	✓	Configure Related Data

Maintaining the List of Action Items

Use the action item list page (Configure Action Items task) to maintain the list of action items.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

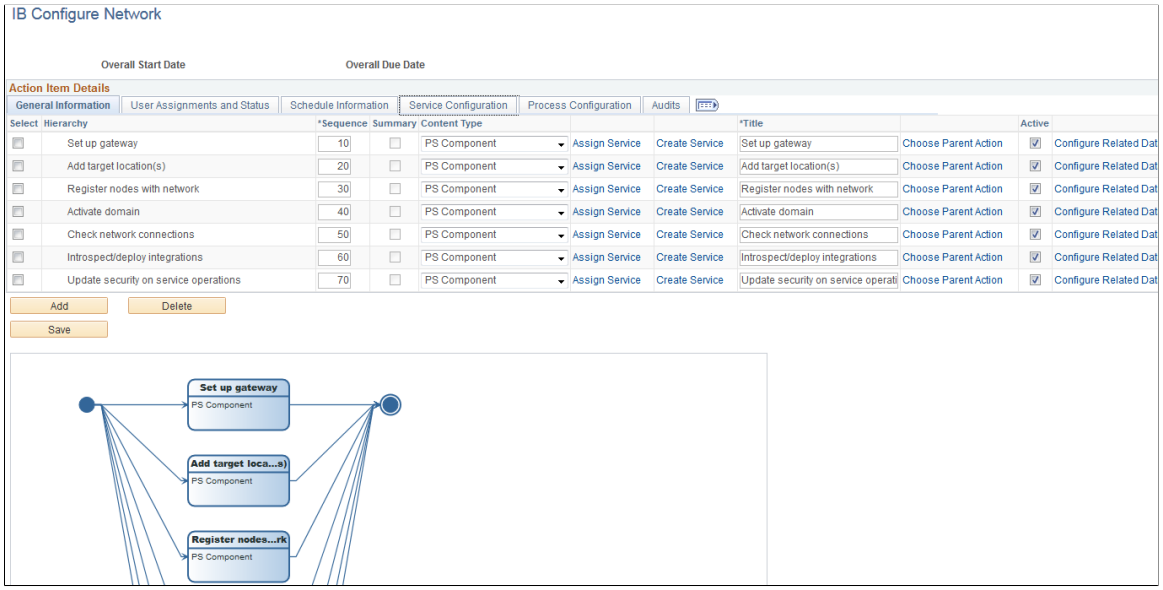
- On the Manage Activity Guide Templates page, click the title link for an existing template.

  Note: In addition, on the Manage Activity Guide Instances page, click the title link for an existing instance.

- In the Setup Activity Guide WorkCenter, click the Configure Action Items task.

Image: Example action item list for an activity guide template

This example illustrates the fields and controls on the action item list page for an activity guide template. You can find definitions for the fields and controls later on this page.



Overall Start Date Displays the start date of the activity guide template (as defined on the Properties page).

Overall Due Date Displays the due date of the activity guide template (as defined on the Properties page).

Select Select an action item to delete. Selecting a summary (parent) action item also selects any detail (child) action items associated with it.

Add Click the Add button to access the Add an Action Item page to create a new action item.

Delete Click the Delete button to delete the selected action items.

Note: There are no checks to prevent you from deleting an action item from an activity guide instance no matter its current status or relationship to other action items in the instance.

Save Click the Save button to save updates to any action item details.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Adding Action Items

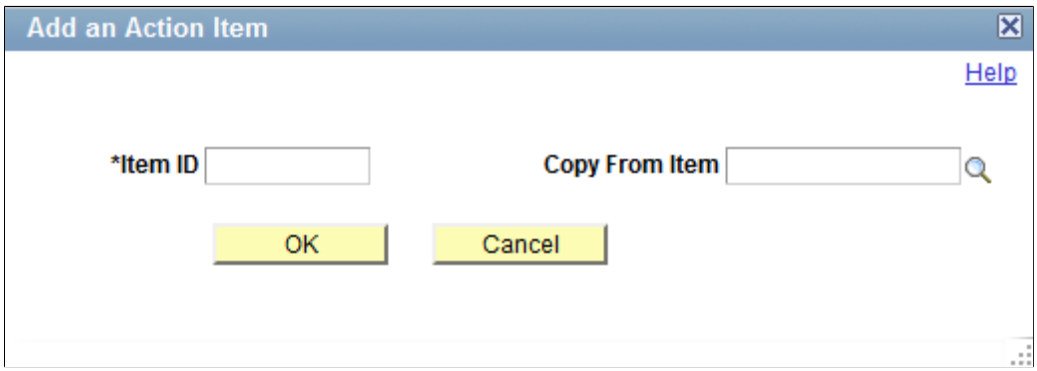
Use the Add an Action Item page to add a new action item.

Navigation

1. In the Setup Activity Guide WorkCenter, click the Configure Action Items task.
2. Click the Add button.

Image: Add an Action Item page

This example illustrates the fields and controls on the Add an Action Item page. You can find definitions for the fields and controls later on this page.



Item ID

Enter up to 10 characters as an identifier prefix.

When you save the action item, the system will append digits to ensure that this identifier is unique within the template.

Important! If you specify an identifier prefix that is already in use in another template, an error is generated.

Copy From Item

Select the ID for an existing action item within this template.

This allows you to select an existing action item from this template to clone as a new action item. You can then modify the properties of the new action item.

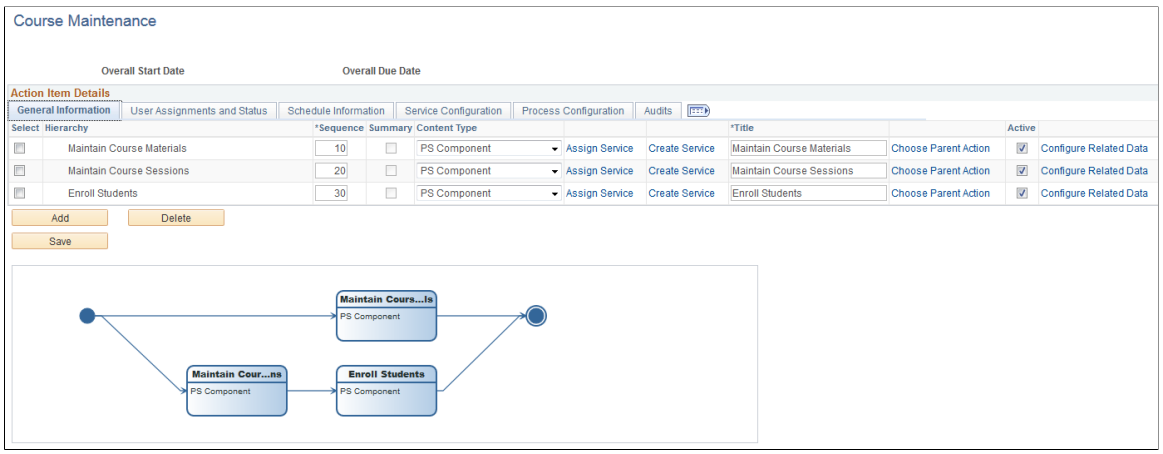
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Graphical Representation of Action Item Dependencies

A graphical representation of the action item dependencies is displayed beneath the Action Item Details grid.

Image: Example action item list with dependencies

In the following example, the third action item, Enroll Students, has been set to be dependent on the completion of its predecessor, Maintain Course Sessions.



Defining Action Item Details

Use the following tabs of the Action Item Details grid to define the details for each action item:

- Use the **General Information tab** to specify the title of the action item, assign a related content service (which can also be assigned on the Service Configuration tab), choose a parent action, and configure related data.
- Use the **User Assignments and Status tab** to maintain the action item's status and to whom it is assigned.
- Use the **Schedule Information tab** to define the start date and time and due date and time for each action item.
- Use the **Service Configuration tab** to associate related content service definitions with action items.
- Use the **Process Configuration tab** to define the self-service process details for action items defined as a *Process* type.
- Use the **Audits tab** to review audit information for each action item including a change log.

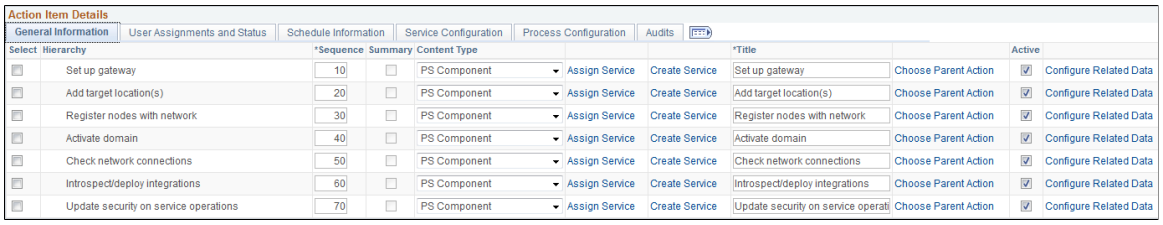
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

General Information Tab

Use the General Information tab to specify the title of the action item, assign a related content service (which can also be assigned on the Service Configuration tab), choose a parent action, and configure related data.

Image: Action Item Details - General Information tab

This example illustrates the fields and controls on the Action Item Details - General Information tab for the activity guide template. You can find definitions for the fields and controls later on this page.



Hierarchy

Displays the parent-child relationship of the action items in the template or instance.

Sequence

Enter up to a four-digit sequence number to indicate this item's position in the ordered list of action items. A sequence number must be unique within an activity guide template or instance.

See "Sequence Numbers" in Understanding Creation and Maintenance of Action Items.

Summary

Select this option to indicate that this is a summary action item.

Important! Important! You must select this option first and then save the action item. If you save the action item without selecting this option it will be created as a child action. Once an action item has been saved, it cannot be changed from a summary item to a child item, or vice versa.

Content Type

Select the content type for the action item. This column is duplicated on the Service Configuration tab and the content types are documented there.

Note: The column is also editable on the Process Configuration tab.

Assign Service

Click this link to select an existing related content service definition of the specified content type.

Note: This function is duplicated by the lookup button for the Service ID field on the Service Configuration tab

---

Create Service

Click the this link to create a new related content service definition for a component-based related content service only.

See Creating a Component-Based Service Definition for more information.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: This link is also displayed on the Service Configuration tab.

Title

Enter the title for the action item.

Note: The default value is the name of the selected related content service.

Choose Parent Action

Click this to access the Look Up Parent Action Item page to change the parent-child relationship of an action item within the hierarchy.

Active

Indicates that the action item is active.

Clear this option to make an action item inactive. The action item definition remains in the template, and is copied to any instances generated from this template. However, an inactive action item will not be displayed to users in an activity guide pagelet.

Configure Related Data

Click this link to configure dependencies, related data, and pagelet button overrides for this action item.

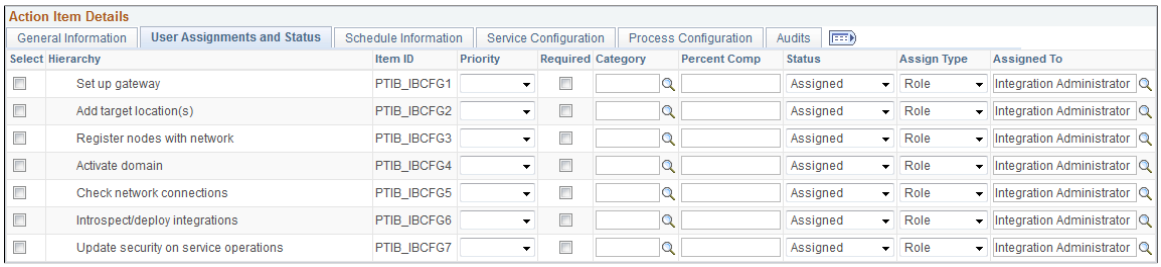
See Configuring Related Data for more information.

User Assignments and Status Tab

Use the User Assignments and Status tab to maintain the action item's status and to whom it is assigned.

Image: Action Item Details - User Assignments and Status tab

This example illustrates the fields and controls on the Action Item Details - User Assignments and Status tab. You can find definitions for the fields and controls later on this page.



Item ID

Displays the ID for the action item.

Priority

Select a priority for the action item: *High*, *Low*, or *Medium*.

Required

Select this option to indicate that an action item is required.

For the status of an activity guide instance to be designated as complete, all required action items must have a complete status. Similarly, for a summary action item to be designated as complete, all required detail action items under it must have a complete status.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Category

Enter an optional category for this action item.

Note: If the category does not exist already, it will be created for you when you save the action item.

Percent Comp

Specify a percent completion for this action item.

Status

Select a value for the status of the action item. Typically, in an activity guide template, the initial value should be *Assigned*.

Note: When viewing the action item list in an activity guide instance, the status for action items will be updated dynamically through user actions.

Assign Type

Select the type of participant that you want to assign the action item to. Available values include:

- *Role* — Select to be able to select a role in the Assigned To field.
- *User* — Select to be able to select a user in the Assigned To field.

Assigned To

Select an assignee from the list of all users or roles in the system.

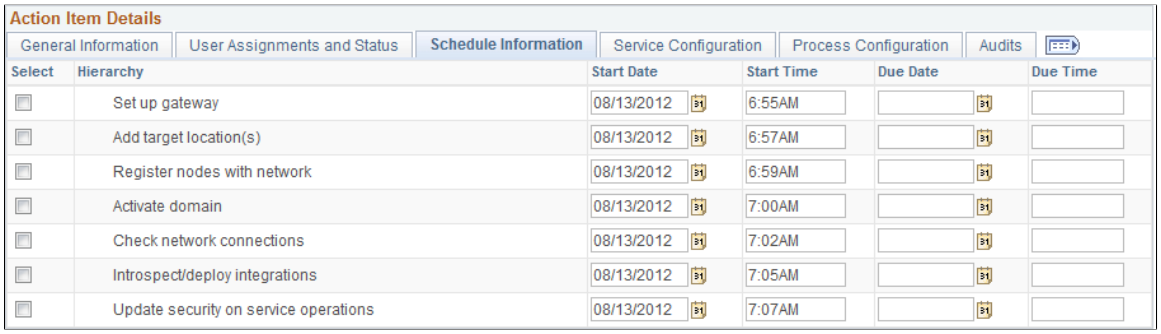
Important! While you can select any user or role, only a user or role that has also been assigned *Contributor* privileges on the template or instance Security page will be allowed to complete the action item. The system will not validate your selection against the users or roles defined on the Security page.

Schedule Information Tab

Use the Schedule Information tab to define the start date and time and due date and time for each action item.

Image: Action Item Details - Schedule Information tab

This example illustrates the fields and controls on the Action Item Details - Schedule Information tab. You can find definitions for the fields and controls later on this page.



Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Start Date Select a start date from the calender, or enter a start date in the format specified by your user personalizations.

Start Time Enter a start time.

Due Date Select a due date from the calender, or enter a due date in the format specified by your user personalizations.

Due Time Enter a due time.

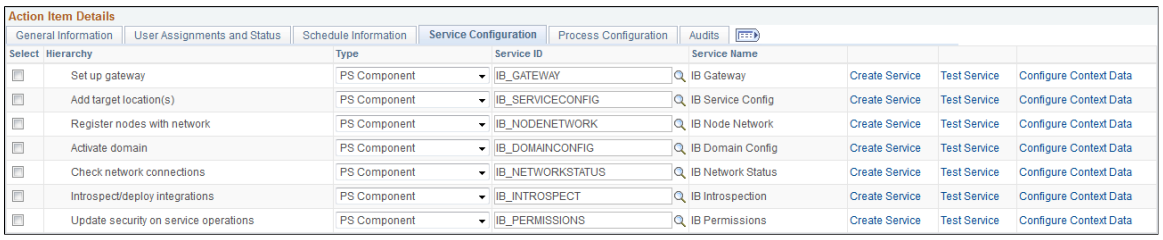
Service Configuration Tab

Use the Service Configuration tab to associate related content service definitions with action items.

Important! Except for component-based related content service definitions, which can be created by clicking the Create Service link in the Action Item Details grid, you *must* explicitly use the Define Related Content Service page to create any other related content service definition prior to associating that service definition with an action item. See Defining Related Content Services for more information.

Image: Action Item Details - Service Configuration tab

This example illustrates the fields and controls on the Action Item Details - Service Configuration tab. You can find definitions for the fields and controls later on this page.



Type

Note: This column is also editable on the General Information tab and the Process Configuration tab.

Select one of the following types:

- *App Class URL –*

  Use this type when you wish to perform any pre-processing prior to displaying the transaction to the user. Also, use this type when the transaction associated with the action item is not known at design time and needs to be determined at runtime. In both cases, the method specified in the service definition must redirect to the URL of the target transaction.

  See Performing Pre-Processing for Action Items for coding guidelines and an example method.

- *Non PeopleSoft URL –*

  Use this type when the user's task must be performed on a non-PeopleSoft system. In this case, because there is no

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

callback mechanism from the non-PeopleSoft system, the user will have to manually mark the action item as complete.

Note: The External URL in the Define Related Content Service page for a non-PeopleSoft URL must contain the additional parameter, *PTAI=T*. For example, `http://example.com/2014/09/15/technology/mobile/moto-x/index.html?iid=TL_Popular&PTAI=T`.

Important! You will need to define a custom button that the user can click to manually mark this action item complete.

- *PS Component –*

  Use this type when the action needs to be performed on a PeopleSoft transaction page. The page can reside on either a local or remote node.

- *PS Query –*

  Use this type when the action to be performed is running a PeopleSoft query.

  For example, a manager may be assigned an action item to run a query to determine which issues are open on a project. In this case, the manager will have to manually mark the action item as complete.

  Important! You will need to define a custom button that the user can click to manually mark this action item complete.

- *PS Script –*

  Use this type when the transaction associated with the action item is not known at design time and needs to be determined at run time. As well, you can use this type when you wish to perform any pre-processing prior to displaying the transaction to the user. In both cases, the iScript specified in the service definition must specify and transfer the user to a target transaction.

- *Process –*

  Use this type to execute a self-service process automatically. Self-service processes can be local or remote. See the following “Process Configuration Tab” section for more information on configuring process type action items.

Service ID

Specify the ID for an *existing* related content service that will be executed for this action item.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: The related content service definitions available for selection are filtered by the type selected.

Note: This field is equivalent to the Assign Service link on the General Information tab.

Service Name Displays the label of the service.

Create Service Click the this link to create a new related content service definition for a component-based related content service only.

See Creating a Component-Based Service Definition for more information.

Note: This link is also displayed on the General Information tab.

Test Service Click this link to test the related content service.

See Testing Related Content Service Definitions for more information.

Configure Context Data Click this link to define additional contextual data that will be passed to the related content service at runtime.

See Configuring Additional Context Data.

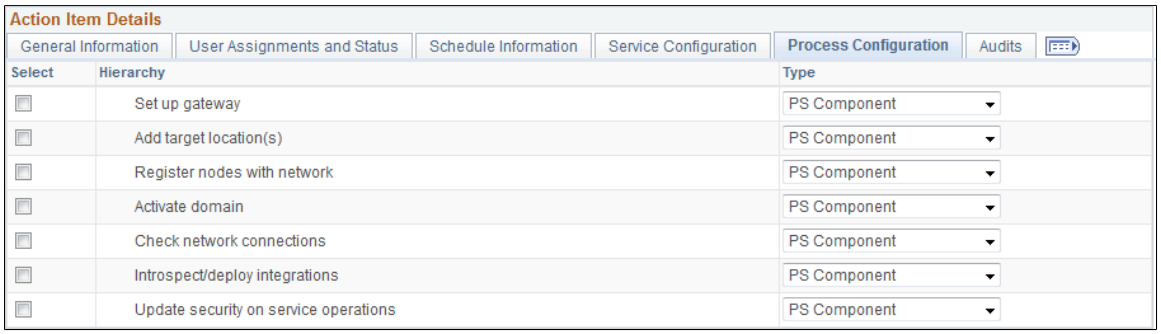
Process Configuration Tab

Use the Process Configuration tab to define the self-service process details for action items defined as a *Process* type. Self-service processes are executed automatically either at instance creation or when all predecessor action items configured with a finish-to-start dependency are marked complete.

For more information about creating, selecting, and configuring self-service processes, see Selecting and Configuring Self-Service Processes.

Image: Action Item Details - Process Configuration tab

This example illustrates the fields and controls on the Action Item Details - Process Configuration tab. You can find definitions for the fields and controls later on this page.



Type Select *Process* to configure a self-service process as an action item.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

	Note: The other values for this column are documented on the Service Configuration tab. This column is also editable on the General Information tab.
Assign Process	Click this link to select a process. See Selecting and Configuring Self-Service Processes for more information.
Target Node	Displays the source node for the process.
Schedule Name	Displays the schedule name associated with the action item.
Process Run Status	Displays the status of the process.
Pause Before	Select this option to pause the process before scheduling the process. For example, you may pause the process to validate or modify the input parameters before scheduling the process.
Pause After	Select this option to pause the process after the completion of process. For example, You may validate the process output before proceeding to the next.
Skip on Error	Select this option to enable display of the process status as Complete even if a process item runs into Error or No Success.
Resume on Error	Select this option to allow the users to reschedule process if it had stopped due to an error.
Configure Input Parameters	Select this option to enable the end-users to modify the run control parameters. This activates the Configure CI Parameters link that enable you to access the Configure CI Parameters page. **Note:** This option is valid only if the Pause Before option is selected.
Configure CI Parameters	Click this link to configure the component interface parameters. See Selecting and Configuring Self-Service Processes for more information.

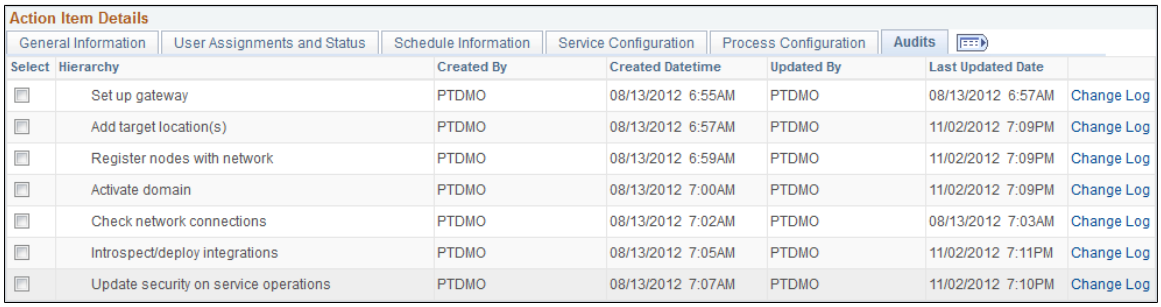
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Audits Tab

Use the Audits tab to review audit information for each action item including a change log.

Image: Action Item Details - Audits tab

This example illustrates the fields and controls on the Action Item Details - Audits tab. You can find definitions for the fields and controls later on this page.



Select	Hierarchy	Created By	Created Datetime	Updated By	Last Updated Date	
	Set up gateway	PTDMO	08/13/2012 6:55AM	PTDMO	08/13/2012 6:57AM	Change Log
	Add target location(s)	PTDMO	08/13/2012 6:57AM	PTDMO	11/02/2012 7:09PM	Change Log
	Register nodes with network	PTDMO	08/13/2012 6:59AM	PTDMO	11/02/2012 7:09PM	Change Log
	Activate domain	PTDMO	08/13/2012 7:00AM	PTDMO	11/02/2012 7:09PM	Change Log
	Check network connections	PTDMO	08/13/2012 7:02AM	PTDMO	08/13/2012 7:03AM	Change Log
	Introspect/deploy integrations	PTDMO	08/13/2012 7:05AM	PTDMO	11/02/2012 7:11PM	Change Log
	Update security on service operations	PTDMO	08/13/2012 7:07AM	PTDMO	11/02/2012 7:10PM	Change Log

Created By Displays the user who created the action item definition.

Created Datetime Displays the date and time when this action item was created.

Updated By Displays the user who last updated the action item definition.

Last Updated Date Displays the date and time when the action item was last updated.

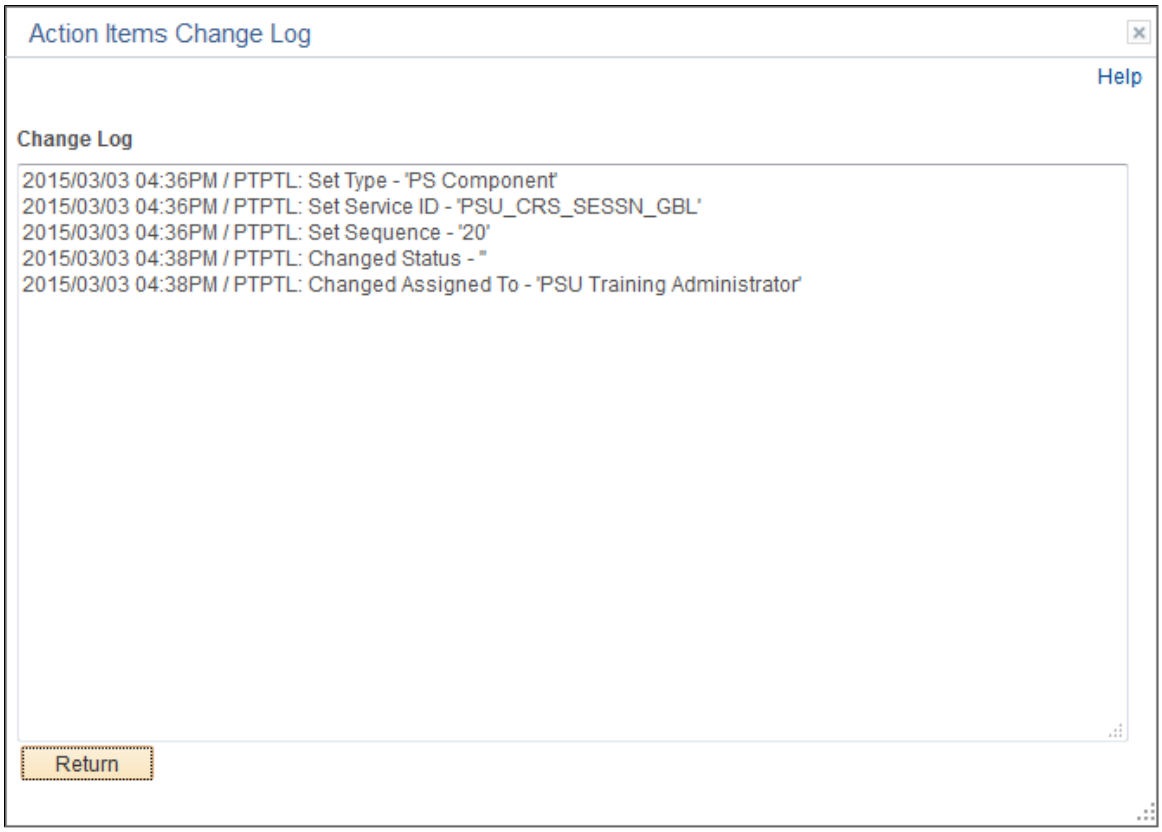
Change Log Click this link to view the action item's change log.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Use the Action Items Change Log page to view the changes to the action item definition.

Image: Action Items Change Log page

This example illustrates the fields and controls on the Action Items Change Log page.



Creating a Component-Based Service Definition

To create a component-based service definition:

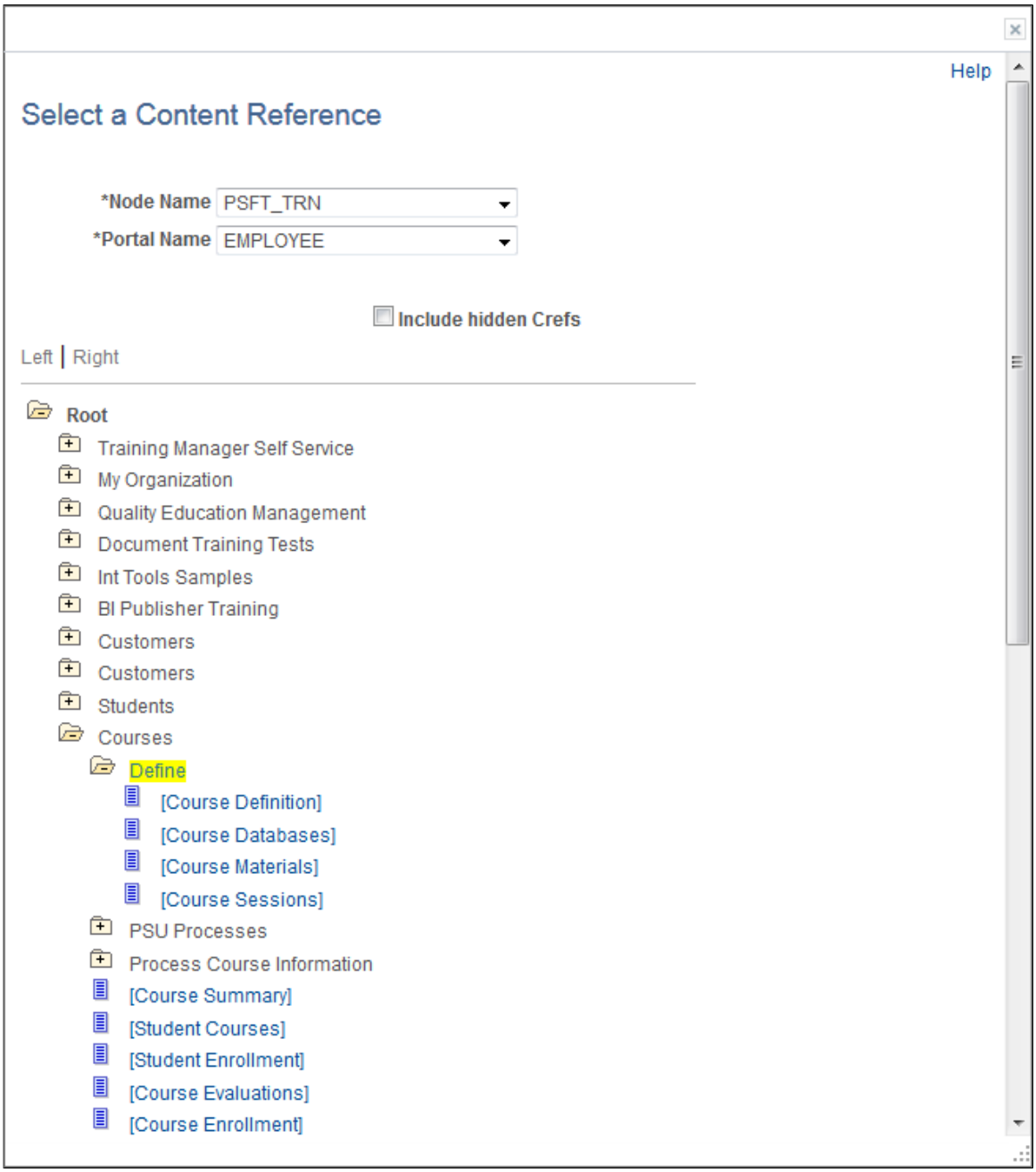
1. In the Action Item Details grid, select *PS Component* as the content type.
2. Click the Create Service link.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

3. Use the Select a Content Reference page to first select the component.

Image: Select a Content Reference page

This example illustrates the fields and controls on the Select a Content Reference page.

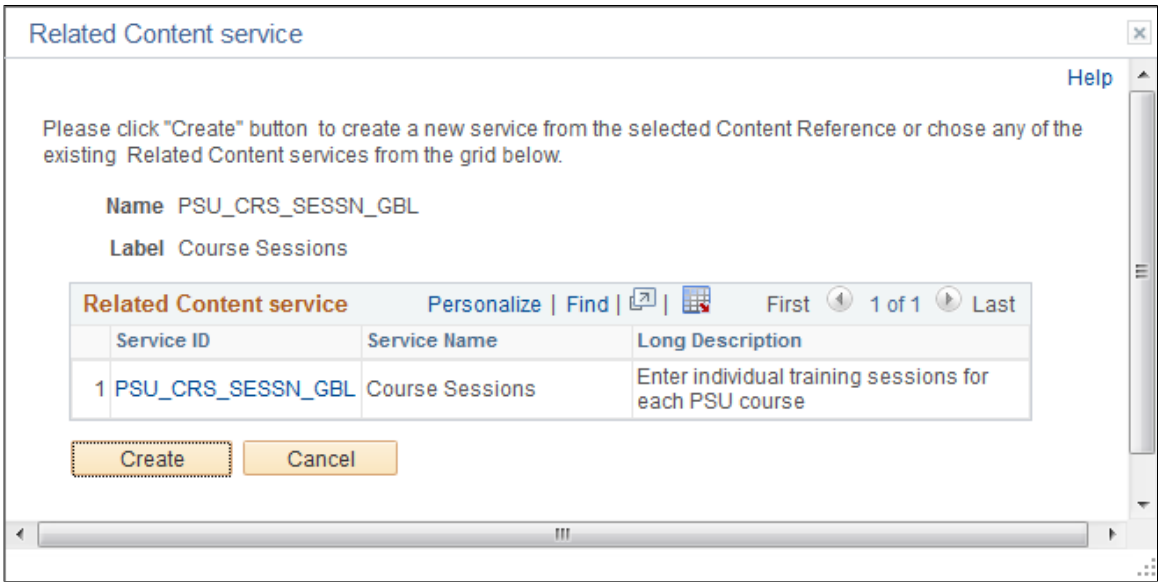


 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. Then, use the Related Content service page to confirm your selection and create the service definition.

   Image: Related Content service page

   This example illustrates the fields and controls on the Related Content service page.



Configuring Related Data

Use the Configure Related Data page to specify dependencies to other action items, to specify post-processing, to override pagelet buttons for this action item, and to specify values for related data fields.

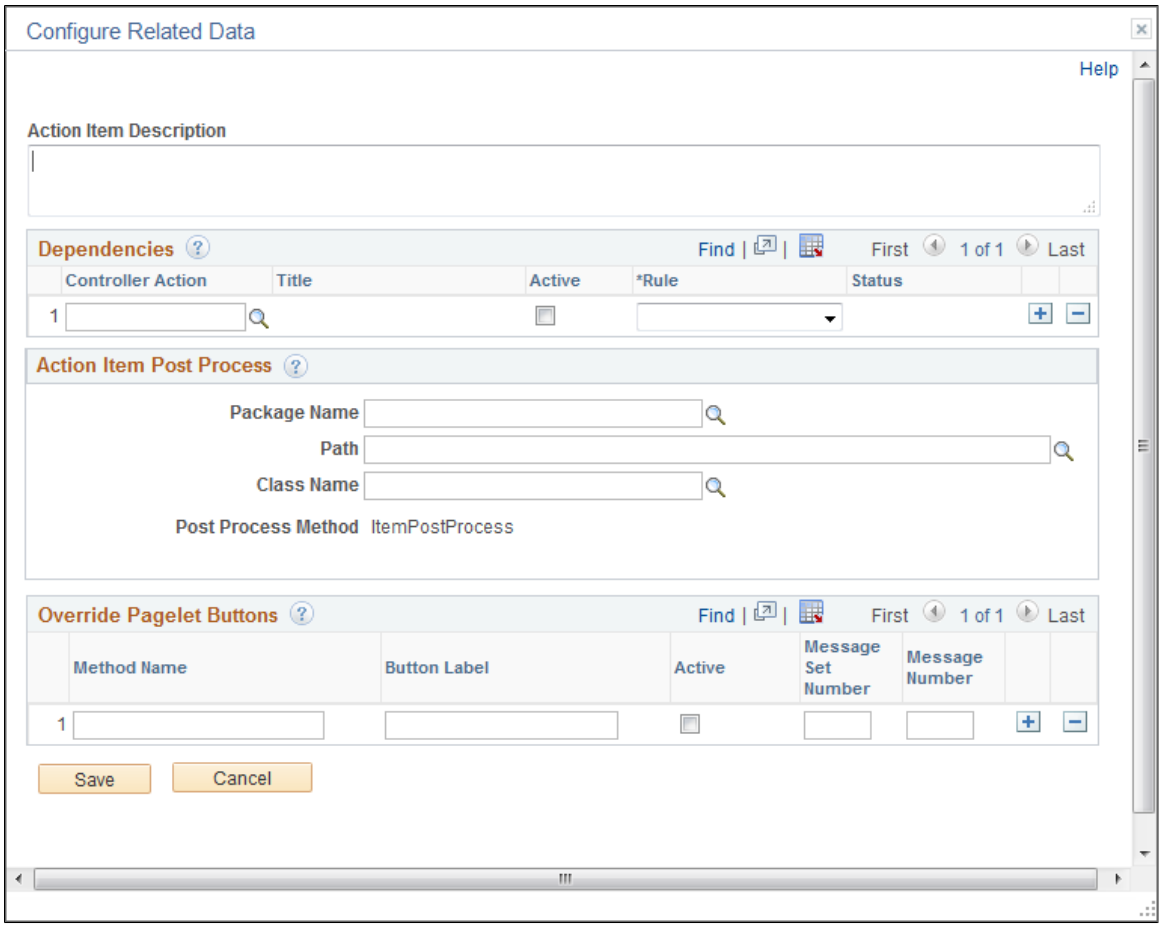
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

Click the Configure Related Data link on the General Information tab of the Action Item Details grid.

Image: Configure Related Data page

This example illustrates the fields and controls on the Configure Related Data page. You can find definitions for the fields and controls later on this page.



Action Item Description Enter any additional description about the action item.

Dependencies

Use the Dependencies grid to designate controller action items upon which this action item is dependent. The finish-to-start dependency rule means that the controller action item must be completed before this (the dependent) action item is allowed to start.

Controller Action Select a controller action item. The list of action items available for selection is filtered to include only those action items with a lower sequence number than the current action item and to exclude completed or cancelled action items.

Title Displays the title for the controller action item.

Active Select this option to retain the dependency as active.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Rule

Select the Finish to Start rule to specify that the controller action item must finish before this action item is allowed to start.

Status

Note: Currently, this column is not used.

Action Item Post Process

Use the Action Item Post Process group box to specify an application package and class that contains the required method to be executed as post-processing when a user completes the transaction—for example, to update the status of the action item to complete. In addition, this application class can optionally define custom navigation buttons on a per action item basis.

Post-processing can occur on the local system or the remote system if that's where the transaction is hosted. Post-processing is executed through component save processing only. If you have implemented custom save buttons on the page to save the component, the PeopleCode executed by these buttons must include either the DoSave or DoSaveNow built-in functions to trigger the component save events; otherwise, the post-processing specified here will not be executed. After the post-processing is executed, the pagelet frame of the WorkCenter page is reloaded to get the latest status of all the action items in the pagelet.

Package Name

Enter the name of the application package that contains the method to be invoked for post-processing and the methods to be invoked for any optional, custom navigation buttons.

Path

Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Class Name

Enter the name of the application class that contains the method to be invoked for post-processing and the methods to be invoked for any optional, custom navigation buttons.

Important! This class should extend the PT_RCF:ServiceAGInterface interface class and must implement a method named ItemPostProcess, which is the method that will be executed for post-processing. If you do not require any post-processing but you want to implement custom navigation buttons for the action item, you must implement still implement a stub method named ItemPostProcess. Otherwise, an error will occur and save processing will be blocked for this action item.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Data

The Related Data group box and fields are displayed only when configurable field labels have been defined for the activity guide template or instance. The displayed field labels allow you to associate application data with an action item at either the template level or the instance level.

Image: Example of the Related Data group box

In the following example, a configurable field labeled Session was defined for the activity guide template. Here, at the action item level, the Session field is displayed along with instance-specific data (the value *801*, providing one methodology for passing data from one action item to the next in the instance:

Override Pagelet Buttons

The Override Pagelet Buttons group box allows you to create custom navigation buttons on a per action item basis. If the name of any of these buttons is the same as a button defined at the template level, then the effect is to override the definition with the method defined here at the action item level. If the name of any of these buttons differs from buttons defined at the template level, then the button is created in addition to those defined at the template level.

Method Name

Enter the name of the method to be invoked to create the additional navigation button.

Important! The method named here *must* exist in the selected class. No validation is performed to ensure this is the case when the template definition is saved.

Important! Mark Complete is a special type of custom navigation button that can be implemented to manually mark an action item as complete. This method *must* be named MarkComplete. After the MarkComplete method is executed, the system will reload the activity guide pagelet in the WorkCenter pagelet area.

Button Label

Enter the label for the button as it will appear in the navigation frame.

Active

Select this option to retain the button definition as active.

Message Set

Specify the message set containing an optional message to be displayed after the user clicks the button.

Message Number

Specify the message number containing an optional message to be displayed after the user clicks the button and before invoking the specified method.

For example, if the user clicks a Cancel button to cancel an activity guide instance, this could display a confirmation message.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Additional Context Data

Use the Action Item Context Data page to define service parameters for the related content service.

Navigation

Click the Configure Context Data link on the Service Configuration tab of the Action Item Details grid.

Context data that is defined as a key field at the template (or instance) level is automatically passed to each related content service. However, an activity guide definition can include additional non-key context data. Use the Action Item Context Data page to map these additional context data fields to related content service parameters as needed on a per action item basis.

Image: Action Item Context Data page

This example illustrates the fields and controls on the Action Item Context Data page. You can find definitions for the fields and controls later on this page.

Parameter Name Enter the field name (as specified on the Template Properties: Advanced Options page) for a context data item.

Service Parameter Select the service parameter (typically, a component key) for the related content service.

Type Select one of the following context types:

- *Fixed Value*: Select to enter a fixed value as the context data.
- *Runtime*: Select to have the system determine the context data value at runtime.
- *System Variable*: Select to specify a system variable as the context data value. The value of the system variable is determined at runtime.

Value Enter the value when the context type is fixed value. Alternatively, select the system variable from the look up list when the context type is system variable.

Note: No value is entered when the context type is runtime.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Selecting and Configuring Self-Service Processes

Self-service scheduling allows users to execute processes and generate reports using context-specific information. You can configure execution of self-service processes from any transaction page where self-service scheduling is enabled. Scheduling in this manner retrieves mapped application data from the transaction page and uses it for the process execution. Once the process is completed, the authorized user can access the report from the dynamically generated notification from any PeopleSoft page.

Within an activity guide, an action item can be defined as a *Process* type, which allows you to associate that action item with an already defined self-service schedule. Use the Process Configuration tab of the Action Item Details grid to complete configuring an action item as a process type:

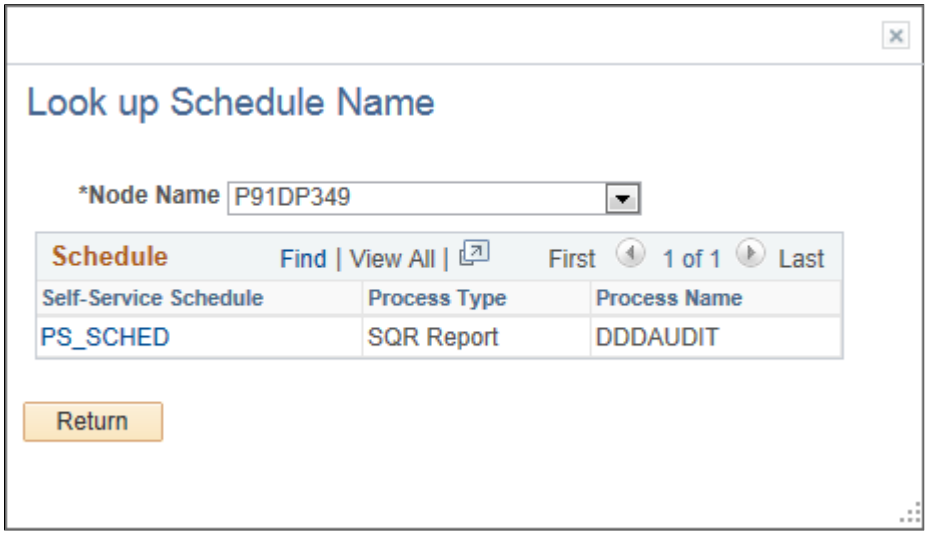
1. Click the Assign Service link to Look up Schedule Name page.
2. If modifiable component interface parameters have been defined for the self-service schedule, click the Configure CI Parameters link to map activity guide data to the component interface.

Look up Schedule Name Page

Use the Look up Schedule Name page to select the already defined self-service schedule.

Image: Look up Schedule Name

This example illustrates the fields and controls on the Look up Schedule Name page.



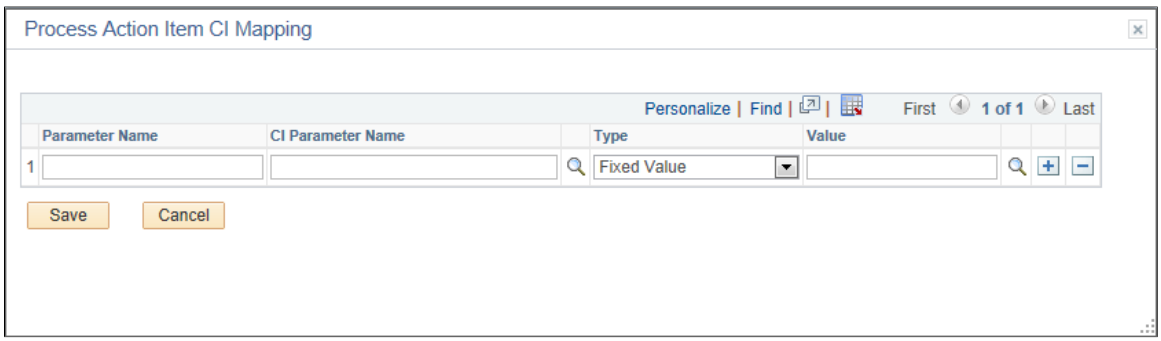
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Process Action Item CI Mapping Page

Use the Process Action Item CI Mapping page to map activity guide context data to any modifiable parameters defined for the component interface for the self-service schedule.

Image: Process Action Item CI Mapping page

This example illustrates the fields and controls on the Process Action Item CI Mapping page. You can find definitions for the fields and controls later on this page.



Parameter Name Enter the field name (as specified on the Template Properties: Advanced Options page) for a context data item.

CI Parameter Name Select the modifiable parameter defined for the component interface.

Type Select one of the following context types:

- *Fixed Value*: Select to enter a fixed value as the context data.
- *Runtime*: Select to have the system determine the context data value at runtime.
- *System Variable*: Select to specify a system variable as the context data value. The value of the system variable is determined at runtime.

Value Enter the value when the context type is fixed value. Alternatively, select the system variable from the look up list when the context type is system variable.

Note: No value is entered when the context type is runtime.

Related Links

"Understanding Self-Service Scheduling" (PeopleTools 8.54: Process Scheduler)

Deploying Activity Guides

This section discusses how to:

- Generate an activity guide pagelet.
- Configure a WorkCenter page for the activity guide.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Modify access to the WorkCenter.

Generating an Activity Guide Pagelet

Activity guides are typically deployed on a WorkCenter page, which is where they can be accessed by users. Prior to creating a WorkCenter page where the activity guide will be deployed, you must first generate a pagelet for the pagelet area of the WorkCenter page. This pagelet will present the action items to the user in list form.

Depending on whether you want to tie the activity guide pagelet to a specific activity guide instance or to dynamically create the activity guide instance, there are three approaches for generating the activity guide pagelet:

- Provide the template ID, but not the instance ID, which allows the instance to be obtained or generated at runtime. At runtime, the required context data is passed as *query string parameters* in the WorkCenter URL as name-value pairs. If an activity guide instance exists for this context data, then it is retrieved. Otherwise, a new activity guide instance will be created dynamically.

  You can use the Publish as Pagelet task in the Setup Activity Guide WorkCenter to generate an activity guide pagelet of this type, which is typically all that is required in most circumstances. Alternatively, you can use Pagelet Wizard to generate this type of activity guide pagelet.

- Provide both the template ID and the activity guide instance ID, which creates a pagelet tied to an existing activity guide instance.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. The template ID is specified as a data source identifier in step 2 of Pagelet Wizard; the instance ID is specified as a data source parameter in step 3.

- Do not provide either the template ID or the activity guide instance ID, which generates an empty, generic activity guide pagelet that can be used to generate any instance from any activity guide template at run time.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. At run time, the template ID is passed as a data source parameter to the pagelet and the context data is provided as query string parameters in the WorkCenter URL as name-value pairs. This will create the activity guide instance dynamically.

Using the Setup Activity Guide WorkCenter to Generate an Activity Guide Pagelet

Use the Publish as Pagelet page to publish an activity guide pagelet with a template ID, but without an instance ID.

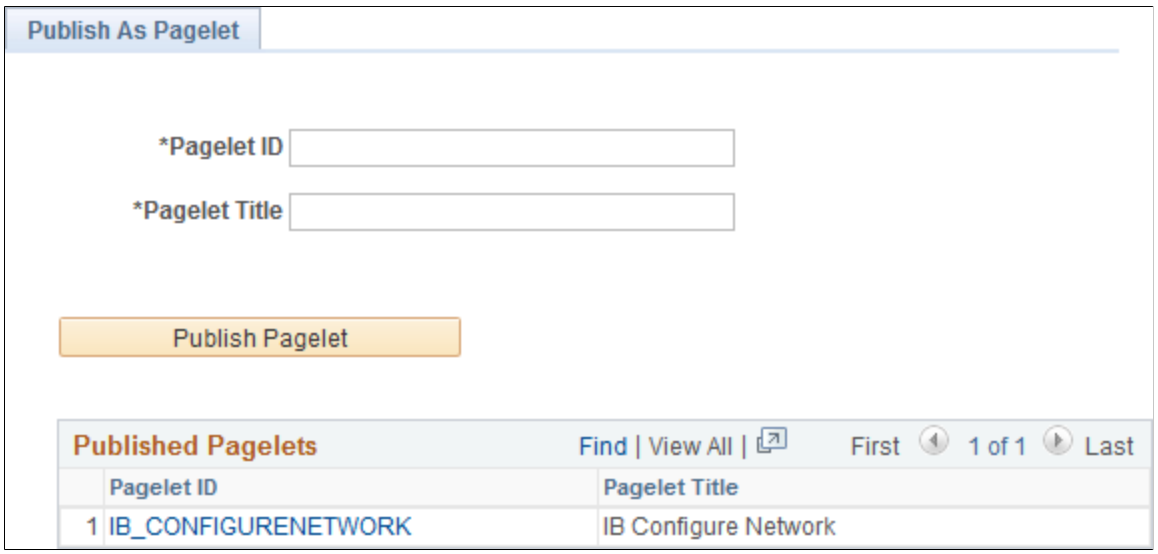
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Navigation

1. PeopleTools, Portal, Activity Guide, Manage Templates
2. Open the template in the Setup Activity Guide WorkCenter:
   - Click the title link for the template.
   - Click the Properties link for the template.
3. Click the Publish as Pagelet task.

Image: Publish as Pagelet page

This example illustrates the fields and controls on the Publish as Pagelet page. You can find definitions for the fields and controls later on this page.



Pagelet ID	Enter an ID for the pagelet definition.
Pagelet Title	Enter a title for the pagelet, which will appear in the pagelet's title bar.
Publish Pagelet	Click the Publish Pagelet button to create the pagelet definition.
Pagelet ID (link)	Click the Pagelet ID link to open the pagelet definition in Pagelet Wizard in a new browser window (or tab).

Using Pagelet Wizard to Generate an Activity Guide Pagelet

You must use Pagelet Wizard to generate an activity guide pagelet for these two scenarios:

- When you need to provide both the template ID and the activity guide instance ID, which creates a pagelet tied to an existing activity guide instance.
- When you do not want to provide either the template ID or the activity guide instance ID, which generates an empty, generic activity guide pagelet that can be used to generate any instance from any activity guide template at runtime.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

When specifying a display format in step 4 of Pagelet Wizard, select the Passthru format. When specifying publication options in step 6 of Pagelet Wizard, always publish an activity guide pagelet as a *template pagelet.* Activity guide pagelets should not be made available or used as homepage pagelets. In addition, activity guide pagelets cannot be consumed as WSRP content.

Related Links

Step 2: Selecting a Pagelet Data Source

Step 3: Specifying Pagelet Data Source Parameters

Step 4: Selecting a Pagelet Display Format

Step 6: Specifying Pagelet Publishing Options

Configuring a WorkCenter Page for the Activity Guide

Use the Configure WorkCenter task in the Setup Activity Guide WorkCenter to configure a WorkCenter page definition for the activity guide. You must use the Activity Guide Layout (PTAILAYOUT) WorkCenter template, which displays the pagelet area is on the left, and the target content area is on the right. In addition, the template displays an activity guide navigation frame above the target content area providing Next and Previous navigation buttons by default. You must also add the activity guide pagelet to the WorkCenter definition.

To configure a WorkCenter page for the activity guide, follow the standard WorkCenter development process:

1. Create a starting page for the WorkCenter that will display the activity guide pagelet.
2. If necessary, register any new components (and pages) in the portal.
3. Create the WorkCenter and designate its starting page. You can use any of the three options as the starting page:
   - Existing menu item.
   - Link of existing menu item
   - WorkCenter dashboard page.
4. Specify the Activity Guide Layout template as the layout template for this WorkCenter page.

   Important! If the default WorkCenter layout is used for the WorkCenter page instead, then the activity guide navigation frame will not be displayed above the target content area and action item statuses will not be updated in the activity guide pagelet.

5. Add the activity guide pagelet to the WorkCenter pagelet area.

Important! Do not configure a remote activity guide pagelet for use in a WorkCenter defined on the local node. Instead, define both the WorkCenter and activity guide pagelet on the remote node. Then, using unified navigation or another mechanism, access the content reference for the remote WorkCenter from the local node.

After you have configured the WorkCenter for the activity guide, you can update the template's Target URL property. See Defining Activity Guide Template Properties for more information.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Links

Understanding the WorkCenter Development and Implementation Process
Selecting the WorkCenter Starting Page
Configuring WorkCenter Pages

Modifying Access to the WorkCenter

You must ensure that users can only access the WorkCenter in the proper manner:

- If the activity guide pagelet was defined with a specific instance ID, then standard menu navigation to the WorkCenter can be allowed.
- If the activity guide requires unique context data to identify each instance, you must ensure that context data is defined prior to accessing the WorkCenter:
  - Use a landing/page questionnaire approach to pass query string parameters to the WorkCenter.
  - Use a query pagelet to pass query string parameters to the WorkCenter.
  - In both cases, hide the WorkCenter's content reference from menu navigation.

If the activity guide requires context data to retrieve or generate the correct activity guide instance, then you must ensure that the WorkCenter is accessed in a manner that provides the context data as query string parameters on the WorkCenter URL. (Additional parameters at the end of a URL are typically referred to as *query string parameters* and sometimes simply as *URL parameters.*) Some PeopleSoft applications achieve this by creating a landing page with a questionnaire that sets the contextual data prior to launching the WorkCenter page. In other cases, you might use a query and query pagelet to set the context data on the WorkCenter URL. In either case, you must also ensure that users cannot access the WorkCenter directly, because doing so would not set required context data on the WorkCenter URL. In some cases, this means hiding the WorkCenter content reference from direct menu navigation.

Managing Activity Guide Instances

This section provides an overview to creating and managing activity guide instances and discusses how to:

- Administer activity guide instances.
- Create new activity guide instances manually.
- Work with the Activity Guides - In Progress pagelet.

Understanding Creation and Management of Activity Guide Instances

The fields and properties of activity guide instances match those of activity guide templates. Therefore, this data can be maintained in much the same way as activity guide templates. In fact, the underlying pages to maintain this data are exactly the same in most cases. However, because activity guide instances can and often will be created programmatically, there is not the same need to maintain activity guide instance data manually.

Activity guide instances can be created programmatically or manually in the following manners:

- Automatically by accessing a WorkCenter displaying an activity guide pagelet.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Manually by clicking the Create Instance button on the Template Properties: Properties page for an activity guide template.

Important! Once an activity guide instance has been created and saved, which template is associated with the instance cannot be changed. Moreover, changes to the template are not propagated to any active instances.

Administering Activity Guide Instances

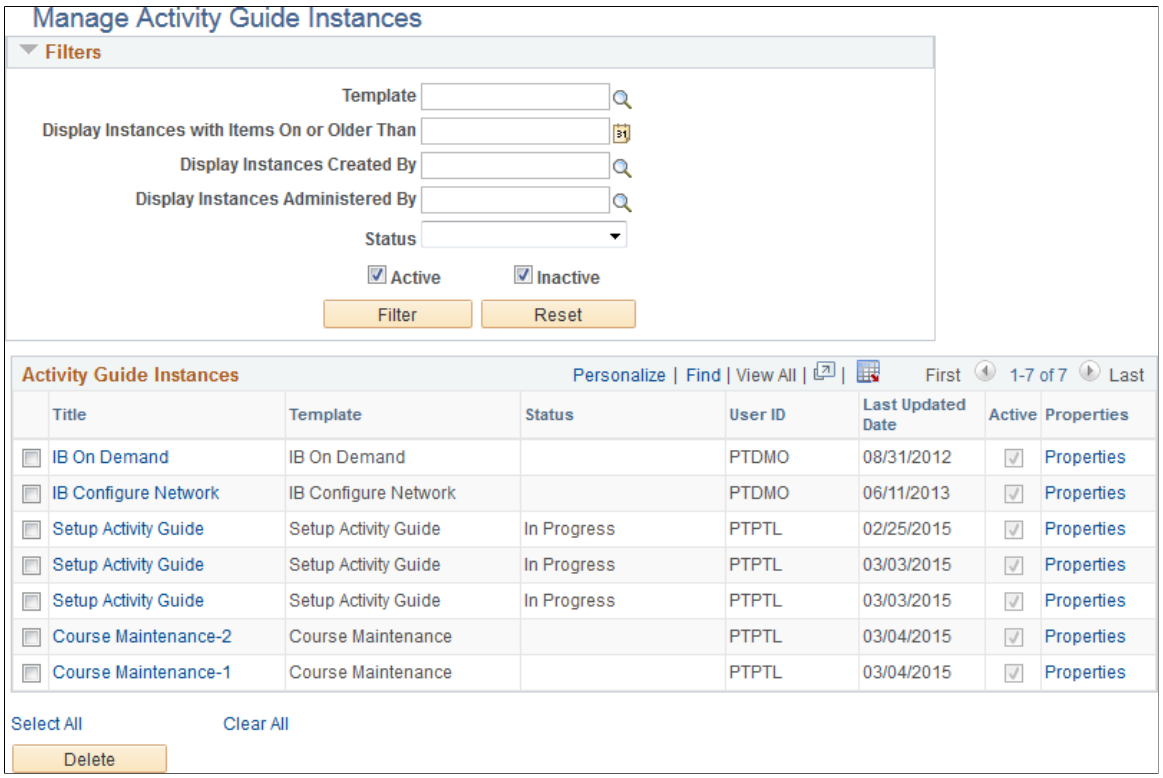
Use the Manage Activity Guide Instances page to manage activity guide instances.

Navigation

PeopleTools, Portal, Activity Guide, Manage Instances

Image: Manage Activity Guide Instances page

This example illustrates the fields and controls on the Manage Activity Guide Instances page. You can find definitions for the fields and controls later on this page.



Filters

Template	Select an activity guide template as a filter criteria.
Display Instances with Items On or Older Than	Select a date to filter by.
Display Instances Created By	Select a user ID as a filter criteria.
Display Instances Administered By	Select a user ID as filter criteria.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Status	Select a status as a filter criteria.
Active	Select whether to include active instances.
Inactive	Select whether to include inactive instances.
Filter	Click to filter the list by the selected criteria.
Reset	Click to clear all filter criteria.

Activity Guide Instances

Title	Click a title link to manage the action items for that instance.
Properties	Click the Properties link to manage the properties for that instance.
Select All	Click to select all instances.
Clear All	Click to clear the selection of any instances.
Delete	Click to delete the selected instances.

Creating New Activity Guide Instances Manually

Use the Create Activity Guide Instance page to specify required context data in order to generate an instance manually.

Navigation

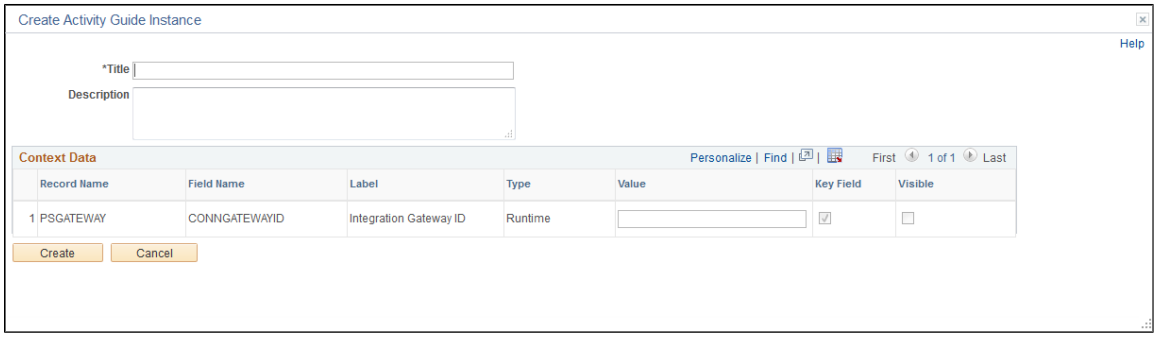
1. PeopleTools, Portal, Activity Guide, Manage Templates
2. Click the Properties link to modify an existing template.

   Note: Alternatively, in the Setup Activity Guide WorkCenter, click the Template Properties task.

3. Click the Create Instance button.

Image: Create Activity Guide Instance page

This example illustrates the fields and controls on the Create Activity Guide Instance page. You can find definitions for the fields and controls later on this page.



Title	Specify the title for this activity guide instance.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Description	Specify an optional description.
Value	If contextual data has been defined for the activity guide template, then you must specify a value for each contextual field.
Create	Click to create the instance.

When you create an activity guide instance in this manner, action items along with template characteristics such as participants, contextual data fields, and pagelet and navigation options are copied from the template to the instance.

Working With the Activity Guides - In Progress Pagelet

Access the Activity Guides - In Progress homepage pagelet (click the Content link on the portal homepage and select the *Activity Guides - In Progress* option to display the pagelet on your homepage).

Image: Activity Guides - In Progress pagelet

This example illustrates the fields and controls on the Activity Guides - In Progress pagelet. You can find definitions for the fields and controls later on this page.

The Activity Guides - In Progress homepage pagelet can be used to access active activity guide instances to which the user is a contributor.

Activity Guide	Click a link to go to the WorkCenter page for that activity guide instance.

Note: While a hyperlink is always displayed, the target URL property must be defined on the Properties page for the activity guide instance in order for the link to work properly.

Related Links

Defining Activity Guide Template Properties

Enhancing Activity Guides

This section provides an overview to where PeopleCode can be used to enhance activity guides and discusses how to:

- Create an activity guide instance dynamically.
- Initialize an activity guide instance.
- Perform pre-processing for action items.
- Perform post-processing for action items.
- Develop custom navigation buttons.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Perform pre-processing for the activity guide pagelet.

Understanding Where PeopleCode Can Be Used

The activity guide framework provides multiple methodologies for using PeopleCode to implement and customize activity guide usage in your system including:

- Creating an activity guide instance dynamically.

  While the browser interface provides pages that allow you to manually create activity guide instances, more typically, you will create activity guide instance dynamically from activity guide templates as the result of user actions.

- Initializing activity guide instances.

  Whether an activity guide is created programmatically or manually, each instance created from a template can be initialized through a PeopleCode method that you define specifically for this purpose.

- Performing pre-processing for action items.

  You can define a PeopleCode application class and method to perform pre-processing on an action item before the target transaction is displayed to the user. In addition to performing this pre-processing, the method *must* also redirect to the target transaction. In addition, Oracle recommends that you develop this application class and method by implementing and extending the PT_RCF:ServiceAGInterface interface class.

  Alternatively, pre-processing can be performed in an iScript that must also specify and transfer the user to the target transaction.

- Performing post-processing for action items.

  You can define another PeopleCode method to perform post-processing on an action item after the user has completed the task. Post-processing can occur on the local system or the remote system if that's where the transaction is hosted.

- Developing custom navigation buttons.

  The navigation frame provides default navigation buttons to allow a user to move to the next or previous action item. You can create additional navigation buttons—for example, to allow a user to manually mark an action item as complete or to cancel the activity guide instance altogether.

- Performing pre-processing for the activity guide pagelet.

  You can define a PeopleCode method to perform pre-processing that will be executed every time the activity guide pagelet is loaded or reloaded in the WorkCenter pagelet area.

Creating an Activity Guide Instance Dynamically

While the browser interface provides pages that allow you to manually create activity guide instances, more typically, your PeopleSoft system will create activity guide instance dynamically from activity guide templates as the result of user actions. The activity guide pagelet that is displayed on the WorkCenter page provides one methodology for dynamically generating activity guide instances from supplied context data.

In this example, the GetActivityGuideInstance method creates an activity guide instance using contextual data:

```
/***
* Following method generates the instance id using the template id and context data
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
* Context Data field populated on template properties could be OPRID, EMPID etc for⇒
 which the
* the values can be retrieved using system variables like %UserID and %EmployeeId.
* If the template has context field defined, check if instance is already available⇒
 for
* %UserId or %EmployeeId context. If there, return the instance id else create a ne⇒
w instance
* the context field value.
***/
method GetActivityGuideInstance
   /+ &QueryString as String, +/
   /+ &TemplateID as String +/
   /+ Returns String +/
   Local PTAI_COLLECTION:Collection &oCtxt = create PTAI_COLLECTION:Collection();
   Local PTAI_ACTION_ITEMS:List &oList = create PTAI_ACTION_ITEMS:List();
   Local array of string &asKeyVal = CreateArrayRept("", 0);
   Local array of array of string &aasKeyVal = CreateArrayRept(CreateArrayRept("", ⇒
0), 0);
   Local number &i, &j;
   Local string &Instance_Id;

   %This.Template_Id = &TemplateID;
   &oList.open(&TemplateID);
   &oCtxt = &oList.GetContext();

   If %This.IsContextDataAvail(&oCtxt) Then
      &Instance_Id = %This.CanLoadInstFromContext();
      If None(&Instance_Id) Then
         &Instance_Id = %This.CreateInstFromCtxtData(&oCtxt);
      End-If;
   End-If;

   Return &Instance_Id;
end-method;


/***
* Check if the context field is defined on template. If yes, use the system variabl⇒
e to get the context field value.
* Form the SQL that results the instance id for the populated context field value.
* Notice the %UserId is used in the following method for context field OPRID define⇒
d on the template.
***/
method IsContextDataAvail
   /+ &oCtxt as PTAI_COLLECTION:Collection +/
   /+ Returns Boolean +/



   Local string &criteria;
   Local SQL &sqlGetCtxtParam;
   Local number &j, &pos;
   Local number &i = 1;
   Local boolean &boolCtxtParamNotAvail = True;

   &Sql = "Select A.PTAI_LIST_ID from PS_PTAI_LIST A where ";

   While &i <= &oCtxt.Count
      If (&oCtxt.Item(&i) As PTAI_ACTION_ITEMS:ContextData).ctxKey Then
         If &criteria <> "" Then
            &criteria = &criteria | " and ";
         End-If;
         &criteria = &criteria | "exists (select 'x' from  PS_PTAI_CONTEXT B where ⇒
A.PTAI_PARENT_TMPL = '" | &Template_Id | "' and A.PTAI_LIST_ID = B.PTAI_LIST_ID and⇒
 B.FIELDNAME = '" | (&oCtxt.Item(&i) As PTAI_ACTION_ITEMS:ContextData).fieldname | ⇒
"' AND B.PTAI_KEY_VALUE = '" | %UserId | "' )";
      End-If;

      (&oCtxt.Item(&i) As PTAI_ACTION_ITEMS:ContextData).keyValue = %UserId;
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
      &i = &i + 1;
   End-While;


   If All(&criteria) Then
      &Sql = &Sql | &criteria;
   Else
      &Sql = "";
   End-If;
   &IsCtxtAvailable = True;
   Return True;

end-method;

/***
* Returns the instance id if there is an instance already available for the context⇒
 data
***/
method CanLoadInstFromContext
   /+ Returns String +/
   Local string &instance;
   Local string &exist;
   SQLExec("select 'X' from PS_PTAI_CONTEXT where PTAI_LIST_ID= :1", %This.Template⇒
_Id, &exist);
   If None(&exist) Then
      Return "";
   End-If;
   If &Sql <> "" Then
      SQLExec(&Sql, &instance);
      If All(&instance) Then
         Return &instance;
      Else
         Return "";
      End-If;
   End-If;

end-method;


/***
* Creates and returns a new instance id for the passed context field value.
***/
method CreateInstFromCtxtData
   /+ &oCtxt as PTAI_COLLECTION:Collection +/
   /+ Returns String +/
   Local PTAI_ACTION_ITEMS:ContextData &ctxData;
   Local PTAI_ACTION_ITEMS:List &newList = create PTAI_ACTION_ITEMS:List();
   Local PTAI_COLLECTION:Collection &ctxDataColl = create PTAI_COLLECTION:Collectio⇒
n();
   Local integer &i, &nbr;
   Local string &label, &descr, &InstanceId;
   try
      For &i = 1 To &oCtxt.Count
         &ctxData = (&oCtxt.Item(&i) As PTAI_ACTION_ITEMS:ContextData);
         If &ctxData.ctxKey And
               &ctxData.keyValue = "" Then
            Return "";
         End-If;
         &ctxDataColl.InsertItem(&ctxData);
      End-For;

      &newList.open(&Template_Id);
      SQLExec(SQL.PTAI_GET_INSTANCES, &newList.ListId, &nbr);
      &nbr = &nbr + 1;
      &label = &newList.Label | "-" | &nbr;
      &InstanceId = &newList.createInstance(&label, &newList.DescrLong, &ctxDataCol⇒
l).ListId;
      Return &InstanceId;
   catch Exception &e
      Return "";
   end-try;
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
end-method;
```

Related Links

Generating an Activity Guide Pagelet

"List Class Methods" (PeopleTools 8.54: PeopleCode API Reference)

Initializing an Activity Guide Instance

Whether an activity guide is created programmatically or manually, each instance created from a template can be initialized through a PeopleCode method that you define specifically for this purpose. This method is identified on the Advanced Options page for the activity guide template.

In the following example, the InstanceIDCreation class implements and extends the delivered PTAI_ACTION_ITEMS:AGInterface class. This InstanceCreation method implements the delivered abstract method.

```
import PTAI_ACTION_ITEMS:*;
import PTAI_ACTION_ITEMS:AGInterface;
import PTAI_COLLECTION:Collection:*;
import PTAI_ACTION_ITEMS:Member:*;


class InstanceIDCreation implements PTAI_ACTION_ITEMS:AGInterface
   method InstanceCreation(&list As PTAI_ACTION_ITEMS:List);
end-class;


/**
* Sets the Instance items status to Assigned and Assigned to Operator ID to %User i⇒
d.
**/
method InstanceCreation
   /+ &list as PTAI_ACTION_ITEMS:List +/
   /+ Extends/implements PTAI_ACTION_ITEMS:AGInterface.InstanceCreation +/

   Local array of string &childItemIds;
   Local integer &i;
   Local boolean &return;
   Local string &ptai_mbr_name, &ptai_mbr_type, &ptai_privset_ID, &ListID, &templat⇒
eID, &templatelabel;

   Local PTAI_ACTION_ITEMS:ActionItem &item;
   Local PTAI_COLLECTION:Collection &members, &buttons, &contextfields;
   Local PTAI_ACTION_ITEMS:Member &member;
   Local PTAI_ACTION_ITEMS:Privilege &privileges;
   Local PTAI_ACTION_ITEMS:PgltButtons &button;

   &list.Status = "IP";
   SQLExec("Select PTAI_PARENT_TMPL From PS_PTAI_LIST Where PTAI_LIST_ID = :1 ", &l⇒
ist.ListId, &templateID);

   rem get the template label;
   SQLExec("Select PTAI_LABEL From PS_PTAI_LIST Where PTAI_LIST_ID = :1 ", &templat⇒
eID, &templatelabel);

   &list.Label = &templatelabel;
   &return = &list.save();

   <*
   rem the button label and method;
   &buttons = create PTAI_COLLECTION:Collection();

   &button = create PTAI_ACTION_ITEMS:PgltButtons("Cancel", "CancelAG");
   &buttons.InsertItem(&button);
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
   &button = create PTAI_ACTION_ITEMS:PgltButtons("Continue Later", "ExitAG");
   &buttons.InsertItem(&button);

   rem leave the code here if the Ben Admin decides to use the mark task complete p⇒
ushbutton;
   rem the label can be changed from Mark Task Complete' to another label based on ⇒
the Ben Admin needs;
   rem the parameter MarkComplete must NOT be removed. This is the app method that ⇒
executes code for the button;

   rem &button = create PTAI_ACTION_ITEMS:PgltButtons("Mark Task Completed", "MarkC⇒
omplete");
   rem &buttons.InsertItem(&button);

   &list.SavePgltBtn(&buttons);


   *>

   Rem change the status of the action items;
   &childItemIds = &list.getActionItems();

   For &i = 1 To &childItemIds.Len
      &item = create PTAI_ACTION_ITEMS:ActionItem();
      &item.open(&childItemIds [&i]);
      &item.Status = "1";
      /*
         &item.PackageRoot = "W3EB_LIFE_EVENT";
         &item.QualifyPath = ":";
         &item.AppClassID = "ItemStatus";
      */
      &item.AssignedToOprid = %OperatorId;
      &item.AssignType = "USER";
      &return = &item.save();

   End-For;

   Rem Save the logged in UserID for security and privileges;

   &members = create PTAI_COLLECTION:Collection();

   &members = &list.getMembers();
   &ptai_mbr_name = %UserId;
   &ptai_mbr_type = "USER";

   SQLExec("Select PTAI_PRIVSET_ID From PS_PTAI_PRIVSET Where DESCR = 'Contributor'⇒
 and PTAI_FEATURE = 'AI' AND ACTIVE_FLAG = 'A' AND PTAI_SYSTEM_ITEM = 'Y' ", &ptai_⇒
privset_ID);


   &member = create PTAI_ACTION_ITEMS:Member(&ptai_mbr_name, &ptai_mbr_type);
   &member.PrivilegeSetID = &ptai_privset_ID;
   &members.InsertItem(&member);
   &list.saveMembers(&members);

end-method;
```

Related Links

Defining Advanced Options

Performing Pre-Processing for Action Items

You can define a PeopleCode application class and method to perform pre-processing on an action item before the target transaction is displayed to the user. In addition to performing this pre-processing, the application class and method *must* redirect to the target transaction. Alternatively, you can implement this functionality in an iScript instead of in an application class.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To implement pre-processing for an action item in an application class and method:

1. Define an application class that extends the PT_RCF:ServiceAGInterface.
2. Define the execute method, which implements and extends the execute method in the base class.
3. Set the %This.StrRedirectUrl property to redirect to the target transaction.
4. Append the %This.StrAGQueryParameter property to the end of the generated component URL.
5. Create a related content service definition for this application class.
6. In the Action Item Details grid, select *App Class URL* as the link type and select the related content service definition.

The following example shows the PTAI_AGPKG_UNINAV:TestPreProcessing, which implements and extends the PT_RCF:ServiceAGInterface base class:

```
import PT_RCF:ServiceAGInterface;
import PTAI_ACTION_ITEMS:*;

class TestPreProcessing extends PT_RCF:ServiceAGInterface
   method execute();
end-class;

method execute
   /+ Extends/implements PT_RCF:ServiceInterface.execute +/

   Local PTAI_ACTION_ITEMS:ActionItem &ObjAItem;
   Local boolean &boolSaved;
   Local string &market;

   &ObjAItem = create PTAI_ACTION_ITEMS:ActionItem();
   &ObjAItem.open(%This.StrActionItemId);
   If &ObjAItem.Status <> "4" Then
      &ObjAItem.Status = "2";
   End-If;
   &boolSaved = &ObjAItem.save();

   &market = %Market;
   If None(&market) Then
      &market = "GBL";
   End-If;

   %This.StrRedirectUrl = GenerateComponentContentURL(%Portal, %Node, MenuName.PTUN⇒
_MENU, &market, Component.PTUN_REMOTENODECFG, Page.PTUN_REMOTENODECFG, "") | %This.⇒
StrAGQueryParameter;
end-method;
```

Performing Post-Processing for Action Items

You can define a PeopleCode application class and method to perform post-processing on an action item after the user has completed the transaction—for example, to update the status of the action item to complete. For post-processing, the method name must be called ItemPostProcess. Post-processing can occur on the local system or the remote system if that's where the transaction is hosted.

Important! Post-processing will be executed through component save processing only. If you have implemented custom save buttons on the page to save the component, the PeopleCode executed by these buttons must include either the DoSave or DoSaveNow built-in functions to trigger the component save events; otherwise, this post-processing will not be executed.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Example 1

In the first example, the class has been developed by implementing and extending the PTAI_ACTION_ITEMS:AGInterface interface class, which is optional and not required. The ItemPostProcess method defines the post-processing that will occur for this action item.

```
import PTAI_ACTION_ITEMS:AGInterface;
import PTAI_ACTION_ITEMS:ActionItem;
import PTAI_ACTION_ITEMS:Constants;

class IB_GATEWAY implements PTAI_ACTION_ITEMS:AGInterface
   method IB_GATEWAY();
   rem method InstanceCreation(&list As PTAI_ACTION_ITEMS:List);
   rem method PageletPreProcess(&list As PTAI_ACTION_ITEMS:List);
   rem method ItemPreProcess(&item As PTAI_ACTION_ITEMS:ActionItem);
   method ItemPostProcess(&list_id As string, &item_id As string, &Nodename As stri⇒
ng);
end-class;

/* constructor */
method IB_GATEWAY
end-method;

method ItemPostProcess
   /+ &list_id as String, +/
   /+ &item_id as String, +/
   /+ &Nodename as String +/
   /+ Extends/implements PTAI_ACTION_ITEMS:AGInterface.ItemPostProcess +/
   Local PTAI_ACTION_ITEMS:Constants &PTAI_CONSTANTS = create PTAI_ACTION_ITEMS:Con⇒
stants();
   Local PTAI_ACTION_ITEMS:ActionItem &item = create PTAI_ACTION_ITEMS:ActionItem()⇒
;
   Local boolean &ret;
   Local string &connurl;
   Local string &defaulturl = "http://machinename:port/PSIGW/PeopleSoftListeningCon⇒
nector";
   Local number &count;

   /*** Debug Test ***/
   rem MessageBox(0, "", 0, 0, "I'm being triggered");

   &item.open(&item_id);
   /*** Add logic to check for step completion ***/
   SQLExec("select connurl from psgateway where local_flag='Y'", &connurl);
   SQLExec("select count(*) from psconn", &count);

   If All(&connurl) And
         (&connurl <> &defaulturl) And
         &count > 0 Then
      &item.Status = &PTAI_CONSTANTS.STATUS_COMPLETED;
   Else
      &item.Status = &PTAI_CONSTANTS.STATUS_IN_PROGRESS;
   End-If;

   /*** Save the Action Item State ***/
   &ret = &item.save();
end-method;
```

Example 2

In this second example, the ItemPostProcess method includes conditional logic that handles execution differently depending on whether the transaction is on the local node or a remote node. If the ItemPostProcess method is executing on the remote node for the remote transaction, then it uses the PTAI_UPDATEITEM service operation to update status on the local node.

```
import PTAI_ACTION_ITEMS:*;
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
Declare Function SecuritySyncMessage PeopleCode PTUN_NODECFGDVW.PTUN_BTN FieldFormu⇒
la;
Declare Function IsLocalNode PeopleCode PSPTCSSRVDEFN.PTCS_SERVICEURLTYP FieldFormu⇒
la;

method ItemPostProcess
   /+ &list_id as String, +/
   /+ &item_id as String, +/
   /+ &Nodename as String +/
   Local boolean &boolSaved;
   Local PTAI_ACTION_ITEMS:ActionItem &ObjAItem;
   Local PTAI_ACTION_ITEMS:Constants &AIConstants;
   Local string &Status, &Priority;
   Local string &Required;
   Local number &PercCompl, &Seqno;

   &AIConstants = create PTAI_ACTION_ITEMS:Constants();
   /* Test values */
   &Status = &AIConstants.STATUS_COMPLETED;
   &Priority = "1";
   &Required = "Y";
   &PercCompl = 100;
   rem &Seqno = 35;
   If Not IsLocalNode(&Nodename) Then
      Local Message &Req_Msg, &Response;
      Local XmlDoc &requestDoc;
      Local number &i1;
      Local XmlNode &currentParamsNode;
      Local array of XmlNode &SuccessNode, &errorNode;
      Local string &ResponseXml;
      Local SOAPDoc &TransformedDoc;

      &Req_Msg = CreateMessage(@("Operation." | "PTAI_UPDATEITEM"), %IntBroker_Requ⇒
est);
      &requestDoc = CreateXmlDoc(GetHTMLText(HTML.PTAI_UPDATEITEM_REQ, %UserId, Get⇒
HTMLText(HTML.PTAI_UPDATEITEM_REQ_PARAMS, &item_id, &list_id, &Nodename, &Priority,⇒
 &Status, &Required, &PercCompl, &Seqno)));
      &Req_Msg.SetXmlDoc(&requestDoc);

      &Response = SecuritySyncMessage(&Req_Msg, &Nodename, "PTAI_UPDATEITEM");
      If &Response <> Null Then
         &ResponseXml = &Response.GenXMLString();
         &TransformedDoc = CreateSOAPDoc(&ResponseXml);
         &SuccessNode = &TransformedDoc.DocumentElement.GetElementsByTagName("SAVES⇒
UCCESS");
         &errorNode = &TransformedDoc.DocumentElement.GetElementsByTagName("ERRORMS⇒
G");

         If &SuccessNode <> Null And
               &SuccessNode.Len = 1 Then
            rem do nothing;
         End-If;
         If &errorNode <> Null And
               &errorNode.Len = 1 Then
            Error "Error from Application hosting Activity Guide : " | &errorNode [⇒
1].NodeValue;
         End-If;
      Else
         Error "No Response from IB";
      End-If;
   Else
      &ObjAItem = create PTAI_ACTION_ITEMS:ActionItem();
      &ObjAItem.open(&item_id);
      &ObjAItem.Status = &Status;
      &ObjAItem.Priority = &Priority;
      If &Required = "Y" Then
         &ObjAItem.Required = True;
      Else
         &ObjAItem.Required = False;
      End-If;
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
      &ObjAItem.PercCompl = &PercCompl;
      If All(&Seqno) Then
         &ObjAItem.Sequence = &Seqno;
      End-If;
      &boolSaved = &ObjAItem.save();
   End-If;
end-method;
```

Developing Custom Navigation Buttons

The navigation frame provides default navigation buttons to allow a user to move to the next or previous action item. You can create additional navigation buttons—for example, to allow a user to manually mark an action item as complete or to cancel the activity guide instance altogether. Since, the navigation panel always includes Next and Previous buttons. Oracle recommends that you add no more than two additional buttons to the navigation panel. Buttons for the navigation panel are defined on the Pagelet Options page for a template or instance and can be overridden per action item on the Configure Related Data page.

Important! Do not use think-time functions such as MessageBox, WinMessage, or others in the methods that define custom navigation frame buttons.

Mark Complete Button

Important! Mark Complete is a special type of custom navigation button that can be implemented to manually mark an action item as complete. This method *must* be named MarkComplete. After the MarkComplete method is executed, the system will reload the activity guide pagelet in the WorkCenter pagelet area.

In the following example, the MarkComplete method sets the status of the current action item to complete.

```
method MarkComplete
   /+ &list_id as String +/
   Local PTAI_ACTION_ITEMS:List &ObjList;
   Local PTAI_ACTION_ITEMS:ActionItem &ObjItem;
   Local string &strCurrItem;
   Local boolean &boolSaved;
   &ObjList = create PTAI_ACTION_ITEMS:List();
   &ObjList.open(&list_id);
   &strCurrItem = &ObjList.getCurrentItem();
   &ObjItem = create PTAI_ACTION_ITEMS:ActionItem();
   &ObjItem.open(&strCurrItem);
   &ObjItem.Status = &PTAI_CONSTANTS.STATUS_COMPLETED;
   &boolSaved = &ObjItem.save();
end-method;
```

Exit Button

In another example, the Exit button that invokes this method exits the activity guide and the WorkCenter and returns the user to the default portal homepage:

```
method ExitAG
   /+ &list_id as String +/
   /+ Returns String +/
   <* Local PTAI_ACTION_ITEMS:List &ObjList;
   Local string &strCurrItem;
   &ObjList = create PTAI_ACTION_ITEMS:List();
   &strCurrItem = &ObjList.getCurrentItem(); *>
   Return GenerateHomepagePortalURL(%Portal, %Node, "");
end-method;
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

One alternative to the GenerateHomepagePortalURL built-in function would be to use the GenerateComponentPortalURL built-in function.

Related Links

Defining Pagelet and Navigation Options

"Understanding Restrictions on Method and Function Use" (PeopleTools 8.54: PeopleCode Developer's Guide)

"GenerateHomepagePortalURL" (PeopleTools 8.54: PeopleCode Language Reference)

"GenerateComponentPortalURL" (PeopleTools 8.54: PeopleCode Language Reference)

Performing Pre-Processing for the Activity Guide Pagelet

You can define a pagelet pre-processing method that will be executed every time an activity pagelet is loaded or reloaded in the WorkCenter pagelet area. This method *must* be named PageletPreProcess. The application class containing this method is identified on the Pagelet Options page.

In the following example for the Unified Navigation pagelet, each time the pagelet is reloaded, the PageletPreProcess method creates a link for Manage Related Content for each remote system configured under unified navigation.

```
import PTAI_ACTION_ITEMS:ActionItem;
import PTAI_ACTION_ITEMS:List;
import PTAI_ACTION_ITEMS:AGInterface;
import PTAI_ACTION_ITEMS:Constants;

class UninavPageletPreprocessor implements PTAI_ACTION_ITEMS:AGInterface
   method PageletPreProcess(&oList As PTAI_ACTION_ITEMS:List);
end-class;


<*
 method PageletPreprocess() - This method will be called everytime UniNav pagelet i⇒
s rendered. This function creates a new action item in the UniNav pagelet
for all the remote RC services. If an earlier created remote service is deleted the⇒
n the corresponding action item also will be deleted.
*>
method PageletPreProcess
   /+ &oList as PTAI_ACTION_ITEMS:List +/
   /+ Extends/implements PTAI_ACTION_ITEMS:AGInterface.PageletPreProcess +/

   Local SQL &sqlGetRemoteSvc;
   Local string &svcId, &svcName, &svcDescr, &svcParamname;
   Local boolean &boolStatus;
   Local integer &seq, &i;
   Local array of string &arrChildItems;;
   Local PTAI_ACTION_ITEMS:ActionItem &oItem = CreateObject("PTAI_ACTION_ITEMS:Acti⇒
onItem");
   Local PTAI_ACTION_ITEMS:ActionItem &oNewItem = CreateObject("PTAI_ACTION_ITEMS:A⇒
ctionItem");
   Local PTAI_ACTION_ITEMS:Constants &PTAI_CONSTANTS = CreateObject("PTAI_ACTION_IT⇒
EMS:Constants");

   <* Every time the pagelet is opened child items under the node 'Related Content ⇒
Setup' will be deleted and recreated. Nodes are deleted and added again
      everytime because, if some remote services are deleted then it will get refle⇒
cted in the tree immediately and to prevent the addition of duplicate nodes *>
   try
      &oItem.open("PAPP_UNITM8_PAPP_UNINA1003");
      &arrChildItems = &oItem.getChildActions();
      If All(&arrChildItems) Then
         For &i = 1 To &arrChildItems.Len
            &oItem.open(&arrChildItems [&i]);
            If Not &oItem.ItemId = "PAPP_UNITM9_PAPP_UNINA1003" Then
               try
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
                  &boolStatus = &oItem.delete();
               catch Exception &e1
               end-try;
            End-If;
         End-For;
      End-If;

      SQLExec("Select MAX(PTAI_SEQ) from PS_PTAI_ITEM where PTAI_LIST_ID = :1", &oL⇒
ist.ListId, &seq);
      &sqlGetRemoteSvc = GetSQL(SQL.EPPAI_GET_REMOTE_SVC);
      While &sqlGetRemoteSvc.Fetch(&svcId, &svcName, &svcDescr)
         &oNewItem.new(&svcId, &svcName, &oList.ListId);
         &oNewItem.DescrLong = &svcDescr;
         &oNewItem.ServiceId = &svcId;
         &oNewItem.ParentId = "PAPP_UNITM8_PAPP_UNINA1003";
         &oNewItem.Status = "1";
         &oNewItem.CreatedDateTime = %Datetime;
         &oNewItem.CreatedByOprid = %UserId;
         &oNewItem.LastUpdatedDateTime = %Datetime;
         &oNewItem.LastUpdatedByOprid = %UserId;
         &oNewItem.Target = "T";
         &oNewItem.Type = "C";
         &oNewItem.AssignType = &PTAI_CONSTANTS.MEMBER_TYPE_ROLE;
         &oNewItem.AssignedToOprid = "Portal Administrator";
         &seq = &seq + 10;
         &oNewItem.Sequence = &seq;
         &boolStatus = &oNewItem.save();
      End-While;
      &sqlGetRemoteSvc.Close();
   catch Exception &e
   end-try;
end-method;
```

Related Links

Defining Pagelet and Navigation Options

Migrating Activity Guide Data

You use Data Migration Workbench to migrate projects containing activity guide data between databases. With Data Migration Workbench, you can migrate:

- Activity guide template definitions (with or without associated action item definitions).
- Activity guide instance definitions (with or without associated action item definitions).
- Action item definitions.

For example, to migrate an activity guide template definition and its associated action item definitions:

1. Select PeopleTools, Lifecycle Tools, Migrate Data, Data Migration Workbench.
2. Click the Add a New Value link.
3. Enter a project name and description.
4. On the Project Definition page, select *Activity Guide List* as the Data Set Name.
5. Go to the Data Set Content page.
6. Click the Insert Content button.
7. Search for the activity guide template that you want to migrate.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: Search results can display both templates and instances depending on your search criteria. If you don't know the ID for the activity guide template or instance, select the *All* operator.

8. Select the activity guide template that you want to migrate.
9. Click the Insert and Return button to insert the selected items.
10. Return to the Project Definition page.
11. Add a row to the Project Content grid and select *Activity Guide Item* as the Data Set Name.
12. Return to the Data Set Content page.
13. Ensure that *ACTIVITY_GUIDE_ITEM* is selected and click the Insert Content button.
14. Search for only the action items that correspond to the template that you want to migrate:

Note: Search results will display action items from both templates and instances depending on your search criteria.

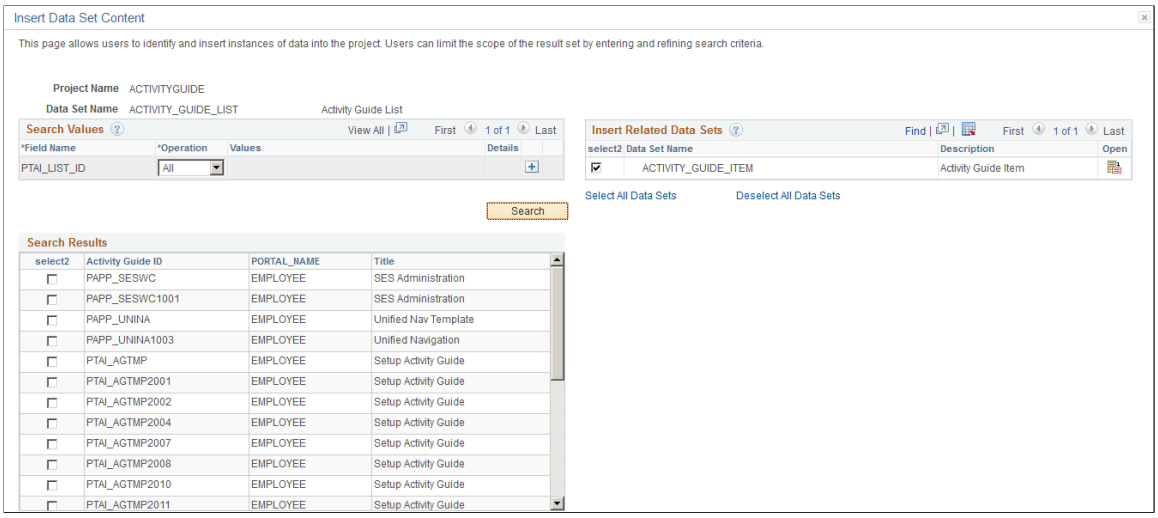
a. For the PTAI_ITEM_ID field, select the *All* operator.

b. Add a row to the search values and select PTAI_LIST_ID as the field name.

c. Select the = and specify the template ID.

Image: Insert Data Set Content

The follow example illustrates search values on the Insert Data Set Content page.



15. Select the appropriate actions items or click the Select All button.
16. Click the Insert and Return button to insert the selected items.
17. Click the Save button to save the data set.
18. Click the Copy To File button to copy the data set to a file.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

19. Select the location for the file.

20. Click Run.

21. On the Process Scheduler Request page, click OK to start the copy process.

Once the project has been copied to a file, use Data Migration Workbench on the target database to compare the project data and copy it into the database. See "Copy Compare Overview" (PeopleTools 8.54: Data Migration Workbench).

Related Links

"Defining Data Migration Project" (PeopleTools 8.54: Data Migration Workbench)
"Defining Data Set Content" (PeopleTools 8.54: Data Migration Workbench)
"Inserting Data Content" (PeopleTools 8.54: Data Migration Workbench)
"Copying Project to File" (PeopleTools 8.54: Data Migration Workbench)

Upgrading Activity Guide Data Using ADS Definitions

Using ADS definitions for upgrading or migrating activity guide data, PeopleSoft enables you to preserve configurations and personalizations customer data. With this enhancement, you can:

- Define intra object and inter object relationships.
- Define Activity Guide configurations.
- Merge and organize activity guide.

Activity guide uses the ADS relationship framework to create relationships to get the related objects and to validate the referential integrity.

Activity guide consists of the following objects:

Migrated with the ADS Objects	***Migrated with the Managed Objects***
Activity Guide	Application Classes
Activity Guide items	Permission Lists
Related Content Services	PS Query
WorkCenter	Component
Pagelet Wizard	Message Catalog
Scheduler Object	Iscript
	Process definitions
	Component Interface

Note: With activity guide ADS definition, you can activity guide, activity guide items, related content service and scheduler object. WorkCenter and pagelet wizard can be migrated using separate ADS projects.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Use Data Migration Workbench to migrate activity guide data. See *PeopleTools: Data Migration Workbench*.

PeopleSoft recommends that you:

- Do not delete any PeopleSoft delivered objects.
- Restrict your configurations to configurable fields specified for each table because these configurations are preserved
- Avoid modification to any field that is in the non-reconcilable section because any modification to these fields is considered to be a customization and will not be restored.

This table lists the configurable and non-reconcilable changes associated with the activity guide template:

Configurable Changes	***Non-reconcilable Changes***
Modifying by adding navigation push buttons.	The button package and Application class information.
Modifying the attributes of an action item, such as title, due date, assignments, additional context data, and so on.	Modifying by changing the related content service of a delivered action item.
Adding navigation buttons, such as *Mark Complete*, *Cancel* and so on.	Modifying styles.
Modifying by adding new action item.	Modifying context data.
Modifying by removing an action item.	
Modifying sequence number .	
Modifying by adding a custom instance creation application class.	
Modifying the post processing properties of an action item.	
Modifying by changing the summary item of an action item.	

The following sections list the database tables.

PTAI_LIST (Activity Guide List)

The following table lists the configurable and the non-reconcilable fields:

Configurable Fields	***Non-reconcilable Fields***
PTAI_LABEL	URL – Target URL
PTAI_DESCRLONG	OBJECTOWNERID – Object Owner
PTAI_ACTIVE_FLG	
PTAI_PBVISIBLE	

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configurable Fields	*Non-reconcilable Fields*
PACKAGEROOT	
QUALIFYPATH	
APPCLASSID	
APPCLASSMETHOD	
PTAI_INST_PKGROOT	
PTAI_INST_QUALPATH	
PTAI_INST_CLASSID	
All label fields	
PTAI_NOTIFY_TYPE	

Related Links

"Defining Data Migration Project" (PeopleTools 8.54: Data Migration Workbench)
"Defining Data Set Content" (PeopleTools 8.54: Data Migration Workbench)
"Inserting Data Content" (PeopleTools 8.54: Data Migration Workbench)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Part V

Working with Portlet Technologies

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 20

Understanding Web Services for Remote Portlets (WSRP)

Understanding WSRP

WSRP is a web services protocol that is used to bring together content and interactive web applications from remote sources. WSRP incorporates standards such as XML, Simple Object Access Protocol (SOAP), and Web Service Description Language (WSDL) to serve as a foundation, while allowing for the implementation of evolving standards.

A WSDL document is an XML file that contains information about the interface, semantics, and other details of a call to a web service. When someone wants to use a service, he or she requests a WSDL document to determine the location of the service, the associated function calls, and how to access them. The administrator then uses this information in the WSDL to form a SOAP request and send it via HTTP to the external system endpoint.

Oracle provides support for:

- Consuming portlets using WSRP technology.
- Producing WSRP portlets through Pagelet Wizard.
- Producing WSRP portlets for PeopleSoft pages and iScripts.
- Implementing WS-Security for single signon with third -party portals that support WSRP and WS-Security.

The terms *producer* and *consumer* are used to describe parties implementing the WSRP protocol.

Producer: A web service that offers one or more portlets and implements various WSRP interfaces. A producer may offer just one portlet or provide a container for deploying and managing several portlets.

Consumer: A web service client that invokes producer-offered WSRP web services and provides an environment for users to interact with portlets offered by one or more producers.

Most PeopleSoft pages, portlets, and iScripts are WSRP-compliant and available for consumption in WSRP-compliant portals; this applies to existing definitions and newly created definitions. While these PeopleSoft items, by default, are available for WSRP consumption, an administrator controls which items are actually produced, or exposed, for WSRP consumption.

This functionality enables you to incorporate PeopleSoft application pages, pagelets, and iScripts into a portal of your choice. When users in the non-PeopleSoft portal interact with the WSRP portlet, depending on how the portlet is configured, they can interact directly within the portlet or they can be transferred to the PeopleSoft portal where the portlet originated.

To enable components for WSRP production, you select the WSRP Compliant option on the Internet tab of the Component Properties dialog box.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See "Setting Internet Properties" (PeopleTools 8.54: Application Designer Developer's Guide).

WSRP Interfaces and Operations

A producer must implement the following WSRP interfaces:

Service Description Interface: Provides metadata of itself and the list of portlets that it offers. The consumer invokes the `getServiceDescription` operation of this interface to obtain the metadata.

Markup Interface: Generates markup and processes interaction requests. The consumer invokes the `getMarkup` operation of this interface to obtain the portlets markup. These portlets then invoke the `performBlockingInteraction` operation to process user interactions to the producer.

WSRP also specifies the following optional interfaces:

Registration Interface: Provides an in-band mechanism for a consumer to register with a producer. It enables the producer to customize its behavior for each consumer based on registration information. WSRP also allows out-of-band registrations and no registration.

Portlet Management Interface: Enables consumers to clone or destroy portlets as well as customize portlets by changing any associated properties.

Note: The registration interface and portlet management interfaces are not used by the PeopleSoft producer. Consumers are therefore not required to register with the PeopleSoft producer.

WS-Security

By implementing the WS-Security standard, Oracle enables you to leverage emerging XML security technologies to address web services security requirements. WS-Security provides:

- A way for applications to construct secure SOAP message exchanges.
- A general-purpose mechanism for associating security tokens with SOAP messages.
- XML message integrity and confidentiality.

With WS-Security capabilities, you can leverage the standard set of SOAP extensions that you use when building secure web services to implement message content integrity and confidentiality. WS-Security provides a way to insert and convey security tokens in SOAP messages. The ability to leverage WS-Security standards provides for better interoperability and improved usability, enabling the implementation of robust security within a WSRP-capable environment. The solutions being provided through the PeopleSoft WS-Security implementation include:

- Single signon solution between WSRP consumer and producer.

  The web services consumer passes the appropriate identification to a producer as part of the SOAP message so that the producer can verify the identity to run requested web services on behalf of the user without requiring a user to sign in. Single signon between the web services consumer and producer is currently supported in PeopleSoft WSRP Portal, PeopleSoft Integration Broker, and the BPEL product.

- SOAP message integrity. This ensures that no one has tampered with the messages.
- SOAP message confidentiality. This guarantees that messages are protected against eavesdroppers.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

WS-Security UsernameToken Profile defines a standard way to associate user ID and password information in the SOAP messaging for web services interoperability.

The Security Assertion Markup Language (SAML) Token defines assertions, protocols, bindings, and profiles.

The PeopleSoft portal solution provides support for WS-Security for single signon with third-party applications, limited to user authentication using user name and password or user authentication using user name and digital signature through the use of Web Services Security: Username Token Profile and SAML Token.

Note: Oracle provides multiple levels of security for WSRP. These levels, or options, are discussed in the following topic. Oracle recommends that you determine the level that is appropriate for your needs before implementing WS-Security. Using SSL (Secure Sockets Layer) connections to secure transmissions may be sufficient.

See Improving Same-Server Performance Under SSL, Understanding the PeopleSoft WS-Security for WSRP.

Terminology

ASF	Apache Software Foundation.
Java Portlet	A Java software module that conforms to the Portlet API.
JCP	*Java Community Process:* Established for the development of Java technology.
JSR	*Java Specification Request:* Each submission to the JCP gets assigned a unique JSR number.
JSR 168	The JCP specification that describes the Portlet API.
SAML	*Security Assertion Markup Language:* An XML standard for exchanging authentication and authorization data between entities. SAML provides a standard security token—a SAML assertion—that can be used with standard web services security frameworks. See http://www.oasis-open.org/glossary/index.php
SOAP	*Simple Object Access Protocol:* An XML-based messaging protocol framework for building and exchanging distributed, structured information in a decentralized and distributed environment.
WSDL	*Web Service Description Language:* An XML language for describing web services; it defines the core language that can be used to describe web services based on what the services offer.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

WSRP *Web Services for Remote Portlets:* A web services protocol for bringing together content and interactive web applications from remote sources.

WSS *Web Services Security Language:* Supports security mechanisms, each using implementation and language-neutral XML formats, which include the use of:

- XML signature to provide SOAP message integrity.
- Use of XML encryption to provide SOAP message confidentiality.
- Attaching and referencing security tokens and associating signatures with security tokens.

WSRP4J The ASF WSRP reference implementation project.

XML *Extensible Markup Language:* Describes data and focuses on what data is. XML is designed to structure, store, and send information.

Implementing WSRP Protocol Scenario

This scenario depicts interactions between two companies: Oracle America (a WSRP producer), and Kane Consulting (a WSRP consumer).

In this scenario, Kane Consulting is an online company providing personalized financial services to clients by subscription. Oracle America wants to host a number of financial applications, including a web-based inventory planning application. Kane Consulting wants to offer this application to its clients via its web pages.

Without WSRP, to offer the inventory planning application to clients, Oracle America and Kane Consulting must agree on the following procedure:

1. Oracle America makes the metadata of the inventory planning application available to Kane Consulting. Kane Consulting uses this metadata to create a page that clients can use to manage their plans.
2. A client visits Kane Consulting's web site and clicks a link to the inventory planning application.
3. Kane Consulting then transmits a request to Oracle America to obtain the initial view of the application. Oracle America responds by returning HTML markup that represents the first page of the application.
4. Kane Consulting processes the returned markup and prepares it for conversion. If the returned markup has links, Kane Consulting transforms the markup so that when the links are activated, they return to the Kane Consulting web site.
5. Kane Consulting converts the markup into a web page, writes it into the response of the browser's connection, and transmits the page to the client's browser.
6. The client reviews the page and finds a form to submit a new vendor ID. The client then enters the ID number and other details and submits the entry.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

7. Kane Consulting receives the request containing the new data. Upon determining that the request is for the inventory planning application, Kane Consulting transmits another request to Oracle America to process the client transaction.
8. Oracle America processes the transaction, adds the vendor ID to the clients plan, and returns a new state for the plan.
9. Kane Consulting then sends a request to get the changed markup based on the current state of the plan. Oracle America generates the markup and returns it to Kane Consulting.
10. Kane Consulting then repeats steps 4 and 5.
11. The client receives a new page containing the updated plan.

Instead of developing a protocol to achieve the preceding procedure, Oracle America and Kane Consulting can use WSRP as the protocol. Oracle America is a WSRP producer offering portlets, and Kane Consulting is a WSRP consumer consuming portlets and aggregating portlets for clients to access aggregated portlet pages. The inventory planning application is a portlet offered by the WSRP producer.

To implement the preceding procedure, Oracle America and Kane Consulting use WSRP to define various interactions, with Oracle America implementing the following required WSRP interfaces and operations.

Service Description Interface: Provides metadata of itself and the list of portlets it offers. Kane Consulting invokes the `getServiceDescription` operation of this interface to obtain this metadata step 1 of the procedure).

Markup Interface: To generate markup and to process requests, Oracle America implements the markup interface specified by WSRP. Kane Consulting, invokes the `getMarkup` operation to obtain the portlet's markup (steps 3 and 9), and invokes `performBlockingInteraction` to generate the client's interactions to Oracle America (Step 7).

By implementing the preceding interfaces and agreeing to conform to WSRP, both Oracle America and Kane Consulting can use a standard process to offer and consume portlets. In addition, Oracle America can offer the same portlets to company X as long as company X adheres to WSRP, and Kane Consulting can consume additional portlets offered by company Y provided that company Y also implements WSRP interfaces.

Considerations for WSRP and Server Cluster Configuration

Support for web server clustering with session replication is supported. However, the open source products that the application uses for WSRP functionality, Apache Pluto and WSRP4J, do not support serialization. If you intend to implement web server clustering with session replication *and* WSRP functionality, you must set up two separate web server environments. The web server environment servicing WSRP functionality cannot have session replication enabled.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 21

Configuring WS-Security for WSRP Consumption and Production

Determining Security Requirements

Before implementing WSRP, you first need to determine the level of security that you require. If you are consuming remote portlets, the level that you select *must* be the same as the producer. If you are producing portlets, consider whether you need to secure only the transmission channel or the transmission channel and the messages. You should also consider how security will affect the performance of your servers.

Including the *No Security* option, six types of token security options are available for WS-Security (set in the server-config.wsdd file):

Important! The security option used by the producer *must* be an exact match to the security option used by the consumer.

- *None:* No security token is required.
- *Username Token:* The Username token is in clear text format.

  Important! If you require a secure channel between the producer and consumer, the .HTTPS protocol *must* be used.

- *Authentication Token as Username Token with full security:* Username token is encrypted and messages are digitally signed.

  Note: For this option, either the HTTP or HTTPS protocol can be used to communicate between the producer and consumer.

- *Username token with no password and digital signature:*
- *Authentication Token as SAML Token:* The SAML token messages are digitally signed.

  Note: SAML token messages are digitally signed by default.

- *Authentication Token as SAML Token with full security:* The SAML token is encrypted and messages are digitally signed.

In addition, PeopleSoft features a WSRP WSS Enabled Response option. When specified in an outbound consumer request, the producer response must include a WSS header.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Understanding the PeopleSoft WS-Security for WSRP

Configuring WS-Security for WSRP Overview

Important! The procedures for configuring WSRP consumption and production must be performed in the order set out here.

Complete configuration of WS-Security for WSRP in the following order:

1. Set up WSRP security options. See Setting Up WSRP Security Options.
2. Set up Security Assertion Markup Language (SAML). See Setting Up SAML.
3. Configure WS-Security for PeopleSoft as a WSRP producer. See Configuring WS-Security for PeopleSoft as a WSRP Producer.
4. Define nodes with WS-Security. See Defining Nodes with WS-Security.

PeopleSoft WS-Security

WSRP involves passing SOAP messages between the WSRP consumer and producer. To safely use WSRP, Oracle provides message-level security between the consumer and the producer by incorporating WS-Security.

WS-Security is an extension to the concept of the SOAP envelope header that enables applications to construct secure SOAP message exchanges. It also provides a means for associating security tokens with messages.

WS-Security provides three main mechanisms:

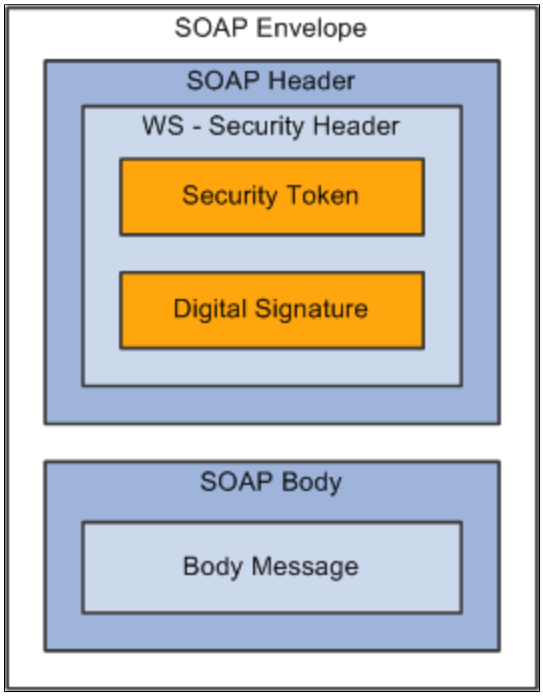
- *Integration solution between the WSRP consumer and producer:* The consumer passes their identification to the producer as part of the SOAP message so that the producer can verify the identity and produce WSRP content without requiring the user to sign in.
- *Message integrity:* Ensures that messages have not been tampered with.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- *Message confidentiality:* Guarantees that messages are protected.

Image: WS-Security SOAP message structure

This diagram shows the SOAP envelope, SOAP header, and SOAP body and how WS-Security embeds the security token in the SOAP messages:



When the PeopleSoft portal is a WSRP consumer, the user ID and password of the user signed in to the portal is placed into a Username or SAML token in the SOAP header, and the WSRP portlet consumes it. It is presented to each portlet during the initial markup request. The security handler performs WS-Security token generation, digital signature generation, and token encryption before the PeopleSoft system sends the SOAP request message and the WS-Security SOAP header.

Important! The PeopleSoft portal authentication information (both the user ID and password) must be the same as the producer's.

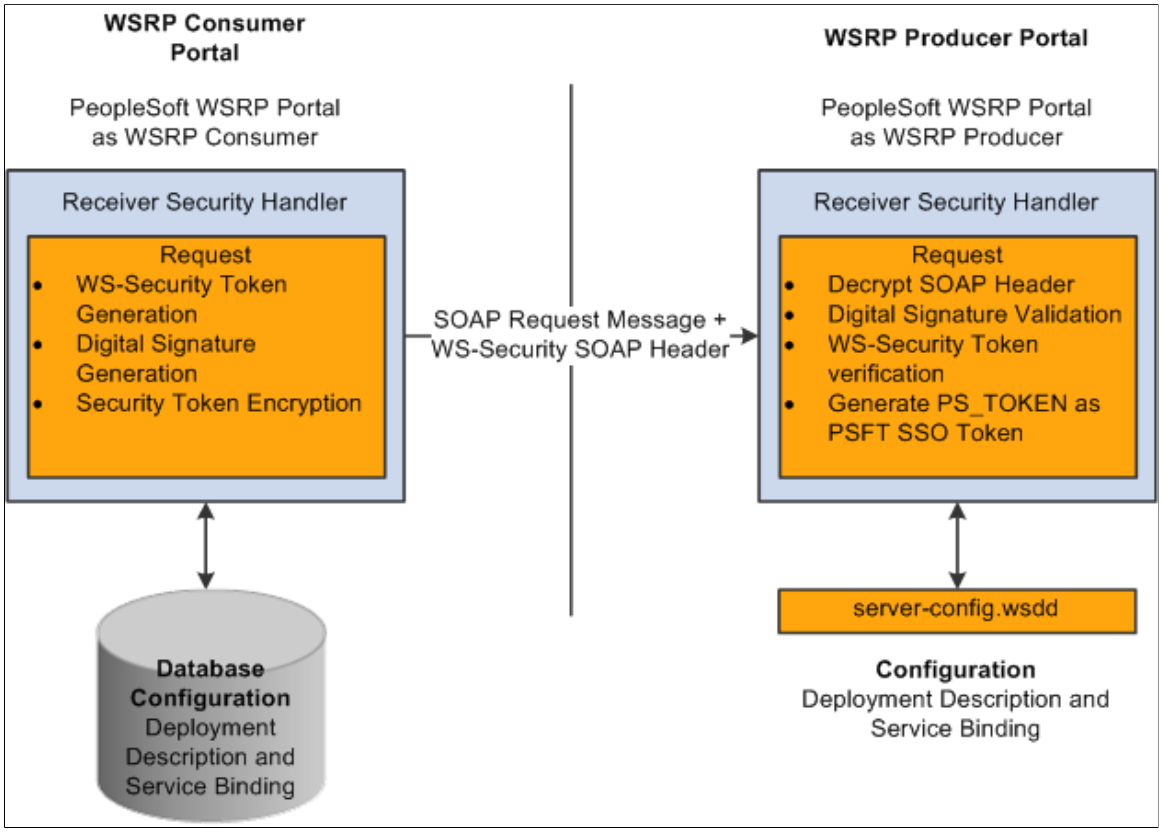
When the PeopleSoft system is the producing system, the consuming system sends PeopleSoft the SOAP request message and the WS-Security SOAP header. The portal accepts the Username token in the WS-Security header from the remote portal, assuming that the ID and password are acceptable to the PeopleSoft system. The receiver security handler decrypts the SOAP header, validates the

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

digital signature, verifies the WS-Security token, and generates a PS_TOKEN cookie, the PeopleSoft authentication token.

Image: PeopleSoft Portal as WSRP consumer and producer

This diagram shows the PeopleSoft portal as both WSRP consumer and WSRP producer:



The WS-Security support provided by PeopleSoft applications includes:

- Username Token Profile.
- Security Assertion Markup Language (SAML) Token Profile.

Note: PeopleSoft applications support SAML 1.1.

UserNameToken Security

With Username Token support, a consumer can supply a UsernameToken as a means of identifying the requestor by username, and optionally using a password to authenticate that identity to the web service provider.

This is an XML example of a UserName Token with password:

Note: The password is in clear text.

```
<soapenv:Header>
 <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/w⇒
ss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
  <wsse:UsernameToken>
   <wsse:Username>QEDMO</wsse:Username>
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
   <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-use⇒
rname-token-profile-1.0#PasswordText">QEDMO</wsse:Password>
  </wsse:UsernameToken>
 </wsse:Security>
</soapenv:Header>
```

This is an XML example of a UserName Token with digital signature:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/ws⇒
s/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"><ds:Signature xmlns:ds="http:⇒
//www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#id-18871350">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>Do+wViC4mSHBWYmRMWKaRDF8xmU=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#id-11549664">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>4NVgNpjX16B4+Wrmw+7bnGfeJSs=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
QFXQuStu8sLAvsDIgNaZHuVWBIes9GpgY6fGb9KLP209hzNi+cPp3NahsF1mVxcXzCjrui2Vc3vq
wLtYA9kaFsM7EMY8TdJC9cmiepr07pR2iXXMmDYYp01dOQkz+3SeBg9F2qPOBAg/jTrWgppys2rS
ES9pV6OIKRk0try8BcU=
</ds:SignatureValue>
<ds:KeyInfo Id="KeyId-1049443">
<wsse:SecurityTokenReference wsu:Id="STRId-14721926" xmlns:wsu="http://docs.
oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<ds:X509Data><ds:X509IssuerSerial>
<ds:X509IssuerName>CN=PeopleTools TEST root CA,DC=peoplesoft,DC=com,OU=
PeopleToolsDevelopment,O=PeopleSoft Inc,L=Pleasanton,ST=CA,C=US</ds:X509IssuerName>

<ds:X509SerialNumber>566474176348341487536757</ds:X509SerialNumber>
</ds:X509IssuerSerial>
</ds:X509Data></wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
<wsse:UsernameToken wsu:Id="id-11549664" xmlns:wsu="http://docs.oasis-open.
org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"><wsse:Username>
QEDMO</wsse:Username><wsse:Password Type="http://docs.oasis-open.org/wss/2004
/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">QEDMO</wsse:
Password></wsse:UsernameToken></wsse:Security></soapenv:Header>

<soapenv:Body wsu:Id="id-18871350" xmlns:wsu="http://docs.oasis-open.org/wss
/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"><ns1:testMethod xmlns:
ns1="http://axis/service/security/test9/LogTestService9"/></soapenv:Body>
</soapenv:Envelope>
```

SAML Token Security

A SAML token makes statements about a principal. All SAML tokens include the following common information:

- Issuer ID
- Issuance timestamp

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- NotBefore or NotOnOfAfter conditions
- Subject
- Name
- Subject confirmation
- Conditions under which the assertion is valid

This is an XML example of an SAML token:

```
<Assertion AssertionID="d9aeaa4c1126df5ee0c6df64fdf961b1" IssueInstant=
"2008-05-14T18:18:47.246Z" Issuer=".peoplesoft.com" MajorVersion="1"
MinorVersion="1"xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:saml=
"urn:oasis:names:tc:SAML:1.0:assertion" xmlns:samlp="urn:oasis:names:tc:
SAML:1.0:protocol">
  <Conditions NotBefore="2008-05-14T18:18:47.184Z" NotOnOrAfter="2008-05-
  14T18:28:47.184Z"/>
  <AuthenticationStatement AuthenticationInstant="2008-05-14T18:18:47.
  215Z"AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
   <Subject>
    <NameIdentifier NameQualifier=".peoplesoft.com">QEDMO</NameIdentifier>
     <SubjectConfirmation>
       <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-
     vouches</ConfirmationMethod>
     </SubjectConfirmation>
    </Subject>
  </AuthenticationStatement>
</Assertion>
```

This is an XML example of an SAML token with digital signature

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
- <soapenv:Header>
- <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.
org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
- <Assertion AssertionID="ede14876b3389b653824f0456e07676d" IssueInstant="
2008-05-13T22:54:17.417Z" Issuer=".peoplesoft.com" MajorVersion="1" MinorVersion
="1"xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:saml="urn:oasis:names:tc:
SAML:1.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
  <Conditions NotBefore="2008-05-13T22:54:17.386Z" NotOnOrAfter="2008-05-
  13T23:04:17.386Z" />
- <AuthenticationStatement AuthenticationInstant="2008-05-13T22:54:17.
  401Z" AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
- <Subject>
  <NameIdentifier NameQualifier=".peoplesoft.com">QEDMO</NameIdentifier>
- <SubjectConfirmation>
  <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</
  ConfirmationMethod>
  </SubjectConfirmation>
  </Subject>
  </AuthenticationStatement>
  </Assertion>
- <wsse:SecurityTokenReference wsu:Id="STRSAMLId-11733267" xmlns:wsu="
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
<wsse:Reference URI="#ede14876b3389b653824f0456e07676d" ValueType="http:
//docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-
  1.0#SAMLAssertion-1.1" />
  </wsse:SecurityTokenReference>
  <wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/
  2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
  ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-pro⇒
file-1.0#X509v3"
 wsu:Id="CertId-28365241" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-2⇒
00401-wss-wssecurity-utility-1.0.xsd">MIIElzCCBACgAwIBAgIKQrSKyQAAAAAL+DANBgkqhkiG9⇒
w0BAQUFADCBvjELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAkNBMRMwEQYDVQQHEwpQbGVhc2FudG9uMRcwFQYD⇒
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
VQQKEw5QZW9wbGVTb2Z0IEluYzEgMB4GA1UECxMXUGVvcGxlVG9vbHMgRGV2ZWxvcG1lbnQxEzARBgoJkia⇒
Jk/IsZAEZFgNjb20xGjAYBgoJkiaJk/IsZAEZFgpwZW9wbGVzb2Z0MSEwHwYDVQQDExhQZW9wbGVUb29scy⇒
BURVNUIHJvb3QgQ0EwHhcNMDcwODE2MTc1MDIzWhcNMDgwODE2MTgwMDIzWjBxMQswCQYDVQQGEwJVUzETM⇒
BEGA1UECBMKQ2FsaWZvcm5pYTETMBEGA1UEBxMKUGxlYXNhbnRvbjEPMA0GA1UEChMGT3JhY2xlMRQwEgYD⇒
VQQLEwtQZW9wbGVUb29sczERMA8GA1UEAwwIcWVfaWJ0Z3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoG⇒
BAMrvvgIBLkotkSm+tnbxjyNQlTi3h3p8h44WhIJ1Ikdvhgtwk890doxdF1HkDaX0Zz7/9IcW4wc5l3z/C/⇒
r0kchcq95ToRxKZ5UC50Sx2wbT9/GL6GVUP/EBtGqcaZSeE4G9OHiVmeXxe7opb3InyyAxU/dbuLj/bMpOv⇒
ABgbmdhAgMBAAGjggHmMIIB4jAdBgNVHQ4EFgQUZ1KpPrdYQY+mEdNk/YWKe4Iqn+wwHwYDVR0jBBgwFoAU⇒
/jeXdgwdjCBwIIOg3c+t5n06BswwgawGA1UdHwSBpDCBoTCBnqCBm6CBmIZMaHR0cDovL3B0bnRhczEyLnB⇒
lb3BsZXNvZnQuY29tL0NlcnRFbnJvbGwvUGVvcGxlVG9vbHMlMjBURVNUJTIwcm9vdCUyMENBLmNybIZIZm⇒
lsZTovL1xccHRudGFzMTIucGVvcGxlc29mdC5jb21cQ2VydEVucm9sbFxQZW9wbGVUb29scyBURVNUIHJvb⇒
3QgQ0EuY3JsMIHwBggrBgEFBQcBAQSB4zCB4DBwBggrBgEFBQcwAoZkaHR0cDovL3B0bnRhczEyLnBlb3Bs⇒
ZXNvZnQuY29tL0NlcnRFbnJvbGwvcHRudGFzMTIucGVvcGxlc29mdC5jb21fUGVvcGxlVG9vbHMlMjBURVN⇒
UJTIwcm9vdCUyMENBLmNydDBsBggrBgEFBQcwAoZgZmlsZTovL1xccHRudGFzMTIucGVvcGxlc29mdC5jb2⇒
1cQ2VydEVucm9sbFxwdG50YXMxMi5wZW9wbGVzb2Z0LmNvbV9QZW9wbGVUb29scyBURVNUIHJvb3QgQ0EuY⇒
3J0MA0GCSqGSIb3DQEBBQUAA4GBABaZwo6xhKZFRbESi3ICewrBTKFjtDbmJvVqBaB0pConBMRGDJ0bQf9R⇒
wo6/Ucm/BoUEEP/dzBkLYM0NEuEqoQvF4ZHRD73qwNV9CCUHU3nlwfnL5K54qrda4V2CoBvgpHEU7EVdt47⇒
YV2E8HUAUfyDaXZ0prRMB6I2KtKaaYBNI</wsse:BinarySecurityToken>
- <ds:Signature Id="Signature-22949069" xmlns:ds="http://www.w3.org/2000/09/xmldsig⇒
#">
- <ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />⇒

  <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
- <ds:Reference URI="#STRSAMLId-11733267">
- <ds:Transforms>
- <ds:Transform Algorithm="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-⇒
soap-message-security-1.0#STR-Transform">
- <wsse:TransformationParameters>
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />⇒

  </wsse:TransformationParameters>
  </ds:Transform>
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>TK2a7xf+ldF9MkI1XYut8g5RG+A=</ds:DigestValue>
  </ds:Reference>
- <ds:Reference URI="#id-78219">
- <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>wTVh0pHi6NrTKDWnyXbX/WNCw68=</ds:DigestValue>
  </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>LGWmXxak++duS8IfY+/4BphfPJW+Ka6a8DxUfYmUGUZ57d1HOmTs0fLR0RoqdI⇒
OKJtSD33qRyU6p 7lufZXsRoiJD5iEUJr+El7KgBEmFPFV5hDx2a+dnHN8Zd9A1DRh7qzr1ewKcRpdBDdoS⇒
2mJnqjcz mlLOU6aPQqKAjgRtZtg=</ds:SignatureValue>
- <ds:KeyInfo Id="KeyId-15595312">
- <wsse:SecurityTokenReference wsu:Id="STRId-24840600" xmlns:wsu="http://docs.oasis⇒
-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <wsse:Reference URI="#CertId-28365241" ValueType="http://docs.oasis-open.org/wss/⇒
2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
  </wsse:SecurityTokenReference>
  </ds:KeyInfo>
  </ds:Signature>
  </wsse:Security>
  </soapenv:Header>
- <soapenv:Body wsu:Id="id-78219" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01⇒
/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <ns1:testMethod xmlns:ns1="http://axis/service/security/test9/LogTestService9" />⇒

  </soapenv:Body>
  </soapenv:Envelope>
```


Note these points about PeopleSoft SAML assertions:

- The PeopleSoft SAML token is concerned with the authentication statement *only.*
- The PeopleSoft SAML token supports SAML with digital signature and encryption. SAML tokens without digital signatures are not supported.
- The PeopleSoft SAML profile of WSS: SOAP Message Security requires that systems support *sender-voucher* methods of subject confirmation.
- The SAML Assertion validity or condition by default is set to 10 minutes. However, you can override the default time by adding `org.apache.ws.security.saml.AssertValidMins=15` in the wssSAML.properties file, which is located in the \\WEB-INF\classes\wssSAML.properties directory.

Setting Up WSRP Security Options

This section discusses how to:

- Use HTTPS protocol to communicate between producer and consumer.
- Install certificates for encryption and signature.

Using HTTPS Protocol to Communicate Between Producer and Consumer

If you choose the security option Authentication token as Username token in clear text format, Oracle recommends that you use HTTPS to protect the username and password. For you to use HTTPS protocol, both the producer and consumer must first configure and enable SSL for HTTPS on their web servers.

Before the consumer imports the producer \pspc\wsdl\wsrp4j_service.wsdl, the producer must modify the file by setting it to HTTPS mode and changing the port number from the HTTP port number to the HTTPS port.

Note: The system automatically inserts the hostname and port number using the values entered during installation.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:oasis:names:tc:wsrp:v1:wsdl"
       xmlns:bind="urn:oasis:names:tc:wsrp:v1:bind"
       xmlns="http://schemas.xmlsoap.org/wsdl/"
       xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
       xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

 <import namespace="urn:oasis:names:tc:wsrp:v1:bind"
          location="wsrp_v1_bindings.wsdl"/>

   <wsdl:service name="WSRPService">
      <wsdl:port binding="bind:WSRP_v1_Markup_Binding_SOAP" name="WSRPBaseService">
        <soap:address location="https://<Producer Hostname>:<SSL port>/pspc/wsrp4j/⇒
WSRPBaseService"/>
      </wsdl:port>
      <wsdl:port binding="bind:WSRP_v1_ServiceDescription_Binding_SOAP" name="WSRPS⇒
erviceDescriptionService">
     <soap:address location="https://<Producer Hostname>:<SSL port>/pspc/wsrp4j/WSR⇒
PServiceDescriptionService"/>
      </wsdl:port>
        </wsdl:service>
       </wsdl:definitions>
```

Note: In the preceding example, <Producer Hostname> is the hostname of the producer web server and <SSL port> is the SSL port number of the producer web server.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

If you are using an SSL accelerator in your PeopleSoft system, and you want to use the HTTPS protocol for WSRP to communicate with PeopleSoft Pure Internet Architecture, you need to use the following properties file:

<PIA_HOME>\webserv\peoplesoft\applications\peoplesoft\pspc\WEB-INF\config\services \PIAService.properties.

Note: This is similar to using Web Profile Default Addressing fields.

See Configuring Web Profiles.

In this properties file, specify the default protocol, host, and port to override the incoming request protocol, host, and port. The settings that WSRP checks in the PIAService.properties file are:

usePIAConfig = true	Checks whether settings in the PIAService.properties file should be used. If you set usePIAConfig to true, the incoming request is overridden by values set in the PIAService.properties file. Otherwise, the system uses the incoming request values for protocol (scheme), host, and port.
portalHost=<your webserver host>	The default host to use.
portalPort=<your port>	The default port to use.
SSLToPIA=true	If you set SSLToPIA to true, the system uses HTTPS; otherwise, it uses the HTTP scheme.

Related Links

"Implementing WebLogic SSL Keys and Certificates" (PeopleTools 8.54: System and Server Administration)

"Setting Up SSL For WebSphere" (PeopleTools 8.54: System and Server Administration)

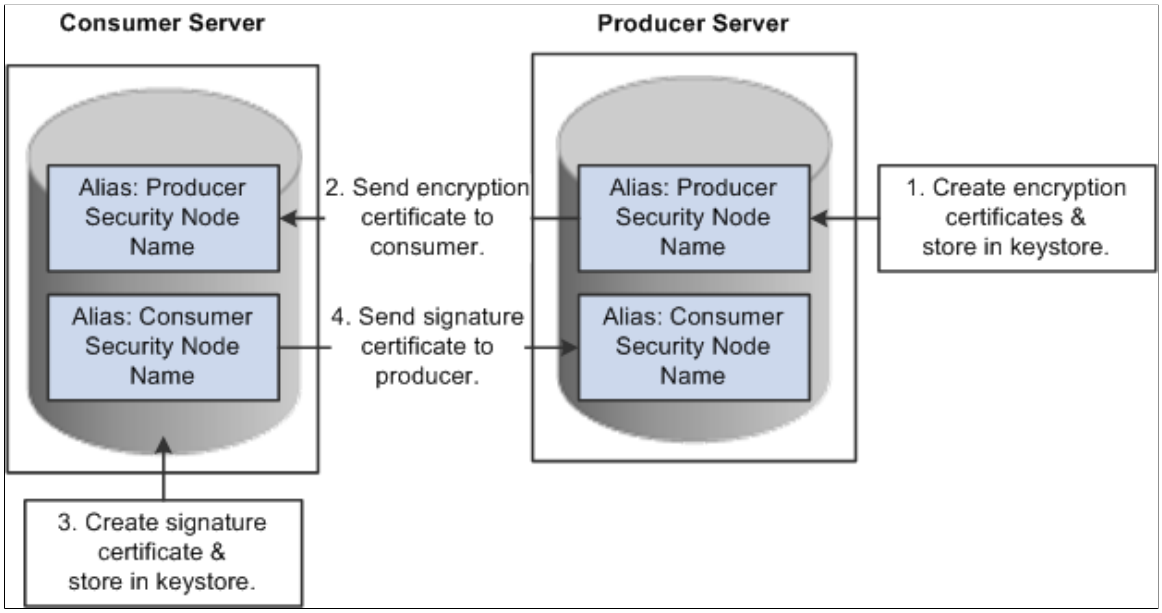
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Installing Certificates for Encryption and Signature

When you select the *Authentication token as Username token with full security* security option, both the producer and the consumer must install the certificates that are used to enable the encryption and signature.

Image: Process for installing certificates and enabling encryption and signature

This diagram illustrates the process flow for installing certificates and enabling encryption and signature:



First use the following reference for instructions on how to create and request your encryption and signature certificates.

See "Implementing WebLogic SSL Keys and Certificates" (PeopleTools 8.54: System and Server Administration),"Setting Up SSL For WebSphere" (PeopleTools 8.54: System and Server Administration).

Enabling Encryption

Producer:

The encryption key and certificate are created on the producer and stored in the web server predefined keystore.

Note: Make sure that when you create the entry to hold your encryption keys and certificate in your keystore, the entry's alias name must be the security node name of your producer server.

Send the encryption certificate to the consumer. You can do this by using an out-of-band method, such as file copy or email.

Consumer:

Install the encryption certificate on the consumer server by running `pskeymanager.cmd -import`.

Note: Make sure that when you import the certificate, the entry's alias name must be the security node name of your producer server.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Enabling Message Signature

Consumer:

Create the signature key and certificate on the consumer and store in the web server predefined keystore.

Note: When you create the entry to hold your signature keys and certificate in your keystore, the entry's alias name must be the security node name of your consumer server.

Send the signature certificate to the producer. You can do this by using an out-of-band method, such as file copy or email.

Producer:

Install the signature certificate on the producer server by running `pskeymanager.cmd -import`.

Note: When you import the certificate, the entry's alias name must be the security node name of your Producer server.

Encrypting Keystore Password Using PSCipher

A copy of the wss.properties file, which contains PKI java keystore information such as keystore file location and password for the keystore can be found on both the consumer and producer servers.

The location of the wss.properties file on the consumer web server is:

...\PORTAL\WEB-INF\classes

The location of the wss.properties file on the producer web server is:

....\pspc\WEB-INF\classes

Here is the sample wss.properties file:

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto
org.apache.ws.security.crypto.merlin.keystore.type=jks
org.apache.ws.security.crypto.merlin.keystore.password=UWZzB57U6SE=
org.apache.ws.security.crypto.merlin.file=keystore/pskey
```

To define the location of the keystore file, the portal administrator needs to define the proper keystore file to replace above the`org.apache.ws.security.crypto.merlin.file` property.

To encrypt the keystore password using Java program PSCipher:

1. Run PSCipher.bat (.sh for UNIX) to encrypt the password

   ```
   % PSCipher.bat <password>
   ```

   For example, `PSCipher interop` provides output as `UWZzB57U6SE=`

2. Update the encrypted password in the wss.properties file for `org.apache.ws.security.crypto.merlin.keystore.password` with the output from PSCipher.

See "Understanding the Triple Data Encryption Standard (DES) Encryption Implementation" (PeopleTools 8.54: Security Administration).

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Setting Up SAML

This section discusses how to:

- Create the SAML administrator.
- Import digital certificates.
- Configure the SAML inbound setup
- Run the RedeployWSRP.cmd executable.

Note: You must perform all of the tasks in the order presented to correctly implement the use of the SAML token.

Creating the SAML Administrator

The SAML administrator must have access to the SAML pages. You grant access to the SAML pages through the *PTPT1000* permission list.

To create the SAML administrator:

1. Access the User Profile page (PeopleTools, Security, User Profiles, User Profiles).
2. Add a new user or select an existing user who will be the SAML administrator.
3. Access the Roles page and insert a role that contains the *PTPT1000* permission list.
4. Save the user profile.

Related Links

"Working With User Profiles" (PeopleTools 8.54: Security Administration)

Importing Digital Certificates

To implement SAML, you must import the digital certificate of the sender and store it in the keystore of participating PeopleSoft applications.

See "Configuring Digital Certificates" (PeopleTools 8.54: Security Administration).

Configuring the SAML Inbound Setup

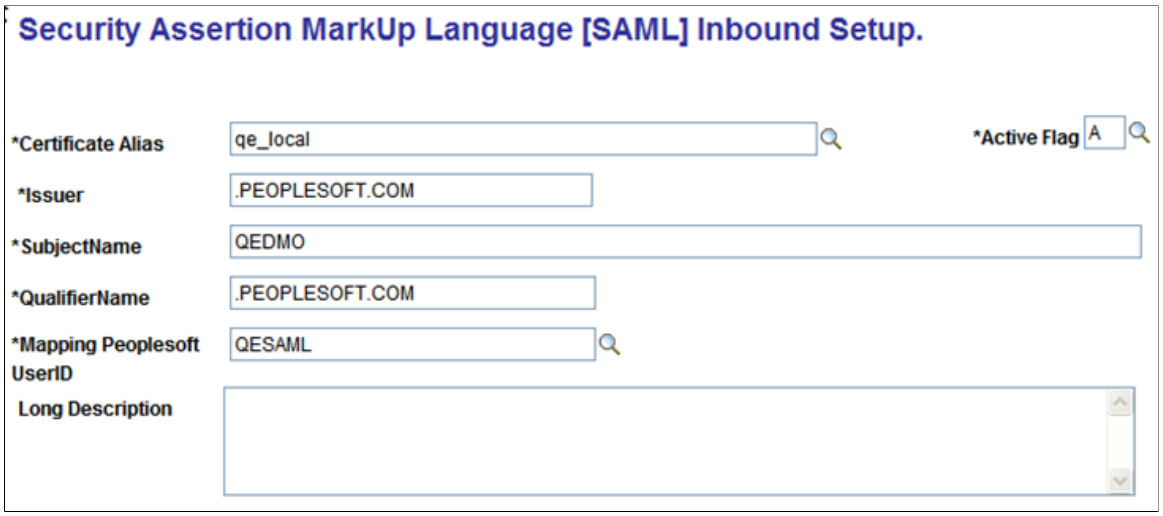
The SAML Inbound Setup page creates an InBound web service in the producer site that maps the one PeopleSoft user ID to one SAML assertion subject and links the subject with the sender's digital certificate (public key). The SAML administrator sets up a web service for each external user who accesses the PeopleSoft system and who is using the SAML security option. This information should be configured by the SAML administrator—someone who understands the external requirements and how these requirements map to the component permissions that are necessary for the user to accomplish the business task.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Security Assertion Markup Language [SAML] Inbound Setup page (PeopleTools, Security, SAML Administration Setup, SAML Inbound Setup).

Image: Security Assertion Markup Language [SAML] Inbound Setup page

The following screenshot illustrates about configuring the SAML Inbound setup on the Security Assertion Markup Language [SAML] Inbound Setup page.



Certificate Alias

Enter the sender's public key, which you imported in the previous step (Importing Digital Certificates).

Note: This key must be base-64 encoded.

Issuer

Enter the domain name of the issuing entity.

SubjectName

Enter a user ID or email address.

QualifierName

Enter the domain name of the issuing entity.

Mapping PeopleSoft UserID

Enter the user ID to map to the SubjectName. This field sets the PeopleSoft internal permissions for the external user and prevents cross-site vulnerability.

Note: This user ID does not have to be the user ID of the sender, but must be a valid PeopleSoft user in the PSOPRDEFN table.

Note: This field is internal to the PeopleSoft application and is hidden from all consumer sites and third-party systems.

Running the RedeployWSRP.cmd Executable

To run the RedeployWSRP.cmd executable:

1. Navigate to the producer web server folder, for example *PSHOME/Webserver/bin.*
2. Double-click the file to launch the program.
3. Select *Option 5:* Redeploy WSRPBaseService with the SAMLToken Security Option.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring WS-Security for PeopleSoft as a WSRP Producer

The *...peoplesoft\pspc\WEB-INF* directory on the producer web server contains a server-config.wsdd file, which contains the WSRPBaseService definition plus the following variations. Each one of the variations is associated with a security option:

Security Option	***Supported with WebCenters***	***Filename***
1. None	Yes.	server-config.wsdd.none
2. UsernameToken in ClearText	No.	server-config.wsdd.usernametoken
3. Authentication Token as Username Token with full security	Not supported out-of-the-box, but can customize.	server-config.wsdd.fullsecurity
4. UsernameToken, No Password, Digitally Signed	No.	server-config.wsdd.untnp
5. SAMLToken Full Security	No.	server-config.wsdd.samltoken
6. SAMLToken Full Security Option (timestamp)**	Yes.	server-config.wsdd.samltokentimestamp
7. UsernameToken Full Security Option With WSS Response*	Yes.	server-config.wsdd.fullsecurityR
8. UsernameToken, No Password Full Security Option With WSS Response*	Yes.	server-config.wsdd.untnpR
9. SAMLToken Full Security Option With WSS Response*	No.	server-config.wsdd.samltokenR
10. SAMLToken Full Security Option (timestamp) With WSS Response* **	Yes.	server-config.wsdd.samltokentimestampR

* The response message must be signed and encrypted.

** A PeoplesSoft SAML token request does not contain a timestamp. However PeopleSoft can accept third-party SAML requests that contain timestamps. No configuration on the PeopleSoft system is required.

PeopleSoft applications provide two options for modifying the security constraint on the WSRPBaseService. Which option you should use depends on whether your web server is currently running. If your web server is running, you can modify WSRPBaseService without rebooting the server.

Note: The *PIA_HOME\webserv\%DOMAIN_NAME%* directory contains the redeployWSRP batch file.

Modifying WSRPBaseService Without Rebooting

To modify WSRPBaseService without having to reboot the web server:

1. Using the command prompt, change to the directory containing the redeployWSRP file:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
cd %PIA_HOME%\webserv\%DOMAIN_NAME%\
```

2. Enter the command redeployWS <option #>

   For example:

   ```
   redeployWSRP 2
   ```

   This will redeploy WSRPBaseService using UsernameToken in the clear text security option. The web service is undeployed and redeployed dynamically without having to reboot the web server.

Modifying WSRPBaseService and Rebooting

To modify WSRPBaseService if the web server is not up and running:

1. Using the command prompt, change to the ...\peoplesoft\pspc\WEB-INF directory:

   For example:

   ```
   cd %PIA_HOME%\webserv\%DOMAIN_NAME%\applications\peoplesoft\pspc
   \WEB-INF\
   ```

2. Copy <desired server-config.wsdd.option> to replace the server-config.wsdd file.

3. Reboot the WSRP Producer Portal web server to allow the newly installed server-config.wsdd file to deploy web services for the producer.

Defining Nodes with WS-Security

This section discusses how to configure WS-Security for WSRP consumers.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Configuring WS-Security for WSRP Consumers

Access the WS Security page. (Select PeopleTools, Portal, Node Definitions.Select the WS Security page.)

Image: Node Definition - WS Security page

This example illustrates the fields and controls on the Node Definition - WS Security page. You can find definitions for the fields and controls later on this page.

Use the Node Definition – WS Security page to select the authentication token type, as well as encryption, digital signature, and WSS response options.

Select Authentication Token Type options based on the options specified in the server-config.wsdd file.

Security Option	***Authentication Token Options***
None	Select *None* as the authentication token type, and deselect all check boxes.
Authentication Token as Username Token	Select *Username Token* as the authentication token type, and deselect all check boxes.
Authentication Token as Username Token with full security	Select *Username Token* as the authentication token type, and select both the Encrypted and Digitally Signed check boxes. **Note:** With this setting, the Username token is encrypted and messages are digitally signed.
Authentication Token as Username Token with no password and digital signature	Select *Username Token, no password* as the authentication token type, and select the Digitally Signed check box.
Authentication Token as SAML Token with digital signature	Select *SAML Token* as the authentication token type. **Note:** By default, SAML tokens are digitally signed.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Security Option	***Authentication Token Options***
Authentication Token as SAML Token with full security	Select *SAML Token* and select the Encrypted check box. **Note:** With this setting, the SAML token is encrypted and by default messages are digitally signed.
Authentication Token as Username Token with full security and WSS Response*	Select *Username Token* as the authentication token type, and select the Encrypted, Digitally Signed, and *WSRP WSS Enabled Response* check boxes. **Note:** With this setting, the Username token is encrypted and messages are digitally signed.
Authentication Token as Username Token , no password, with full security and WSS response*	Select *Username Token, no password* as the authentication token type, and select the Digitally Signed, and *WSRP WSS Enabled Response* check boxes.
Authentication Token as SAML Token with full security and WSS response*	Select *SAML Token* as the authentication token type, and select the Encrypted and *WSRP WSS Enabled Response* check boxes. **Note:** With this setting, the SAML token is encrypted and by default messages are digitally signed.

* The response message must be signed and encrypted.

Related Links

Importing Producer Information
Determining Security Requirements
Defining Portal Nodes

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 22

Using WSRP to Consume and Produce Remote Portlets

Understanding Use of WSRP to Consume and Produce Remote Portlets

These topics discuss how to:

- Set up the local gateway URL (Consumer).
- Use the WSRP Consumption Wizard (Consumer).
- Display remote portlets on a homepage (Consumer).
- Produce remote portlets (Producer).

Note: Use the following link to the primer for more information about Web Services for Remote Portlets (WSRP), a web services protocol for aggregating content and interactive web applications from remote sources: Web Services for Remote Portlets 1.0 Primer

---

Related Links

Understanding Homepages and Dashboard Pages
Managing Homepages
Managing Dashboard Pages

Setting Up the Local Gateway URL (Consumer)

This section discusses how to set up the local gateway URL.

The local gateway is used for administrative tasks, such as registering producers.

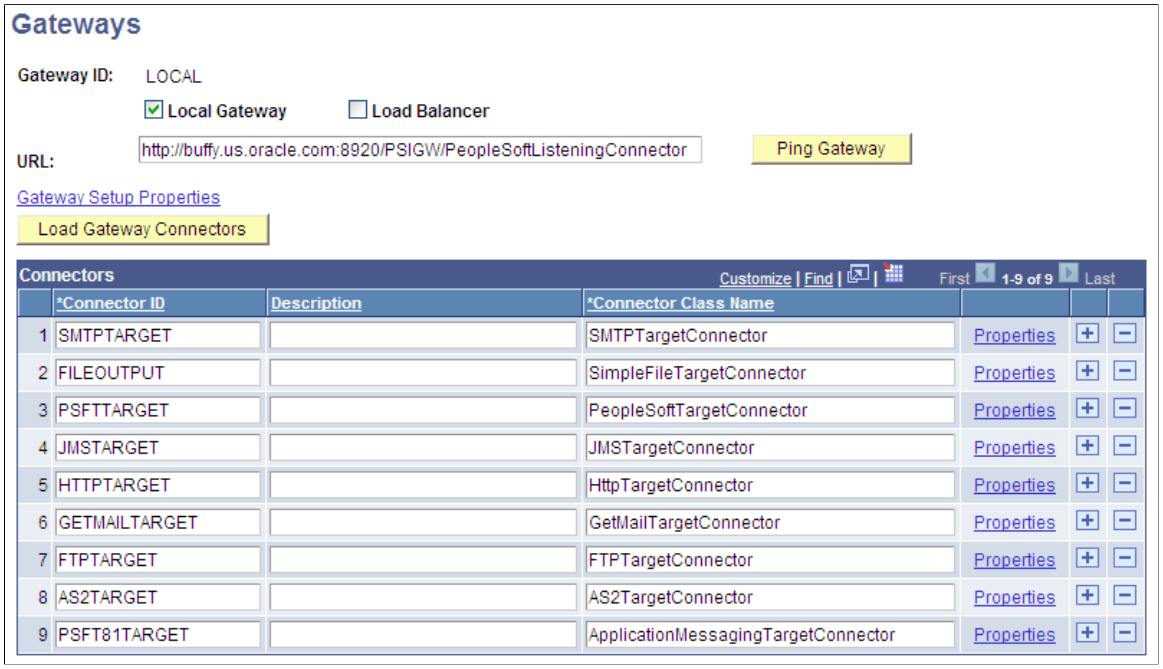
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Setting Up the Local Gateway URL

Access the Gateways page. (Select PeopleTools, Integration Broker, Configuration, Gateways.)

Image: Gateways page

The following screenshot illustrates setting up of the local gateway URL on the Gateways page.



To set up a web server for WSRP administration tasks:

1. Change the URL to *http://<web server name:port>/PSIGW/PeopleSoftListeningConnector*

   Note: If you omit the port number, the system presumes a value of *9000.*

2. Click Save.

3. Click Load Gateway Connectors.

   The connectors appear in the Connectors grid.

4. Click Save.

Using the WSRP Consumption Wizard (Consumer)

This section provides an overview of the WSRP Consumption Wizard data, lists common elements used in this section, and discusses how to:

- Import producer information.
- Register producers.
- Create content references.

Understanding the WSRP Consumption Wizard Data

The Consumption Wizard is a tool used to easily import producer information, register producers, and create content references (CRefs) for the producer-offered portlets. Once a producer has been registered,

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

the offered portlets become available for use by any of your portals. Although the automatically created CRefs are only available to users of the current portal, an administrator can sign in a different portal and manually create CRefs for any of the producer-offered portlets.

Related Links

Administering Content References

Common Elements Used in This Section

Click a number to go to the corresponding page within portal wizard.

1: Import Producer Information.

2: Registration.

3: Producer Offered Portlets.

The different background colors signify the status of the page:

- *Pale Blue:* The page has been updated.
- *Orange:* Current page.
- *Dark Blue:* The page has not been updated.

Previous and Next Click to navigate page by page.

Producer Title and Description Enter meaningful titles and descriptions.

Note: The producer title is also used as the default folder label for the content references created based on the producer's offered portlets.

Creation Date/TimeandLast Update Date/Time Indicates the date and time that the remote portlet information was first entered in the PeopleSoft system and most recently updated.

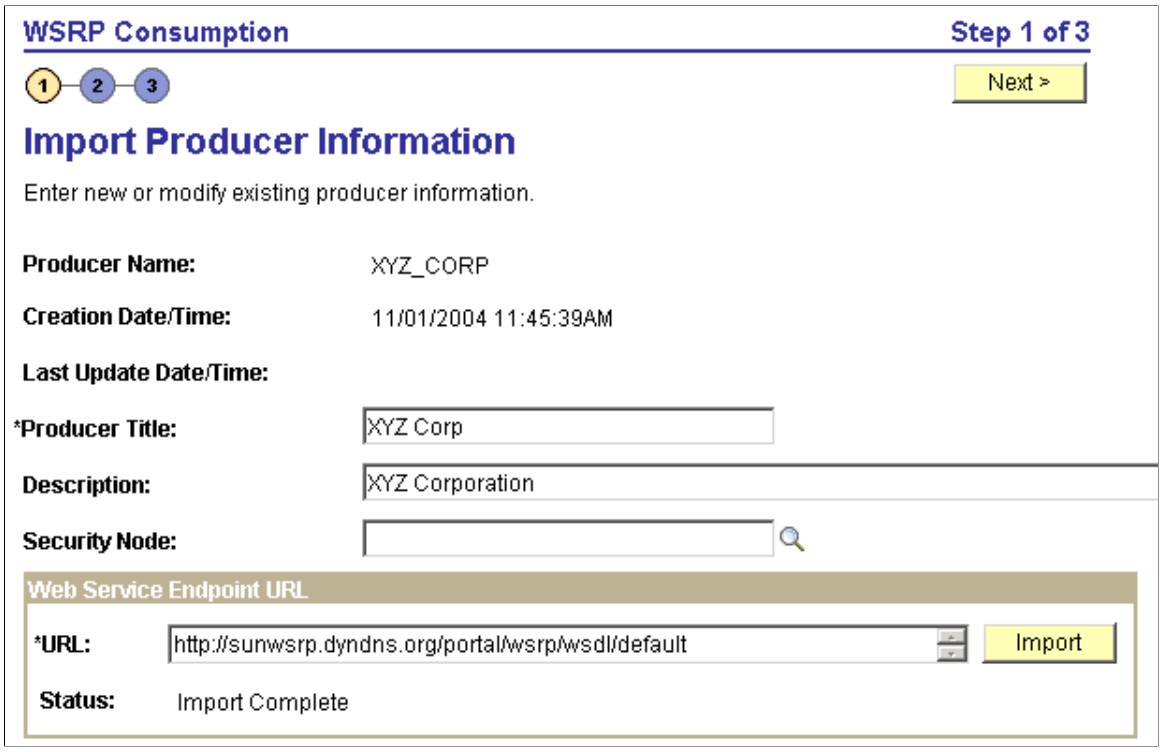
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Importing Producer Information

Access the Import Producer Information page. (PeopleTools, Portal, WSRP Consumption. Click the Add New Producer or Edit button.)

Image: Import Producer Information page

This example illustrates the fields and controls on the Import Producer Information page.



Producer Name Enter a meaningful name for this producer.

The value entered here is also used as part of the default folder name for content references that are created based on this producer's offered portlets.

Security Node (Optional) Select the node definition that was created for WS-Security.

See Understanding the PeopleSoft WS-Security for WSRP.

Web Service Endpoint URL

URL Enter the web service WSDL URL for this producer.

Note: The producer provides this information.

Import Click to import the required producer information.

Status If the WSDL URL is correct, the status will be updated automatically from *No Data Imported* to *Import Complete.* If the URL is incorrect or the producer is unavailable, you will

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

receive a warning message with information about how to make corrections.

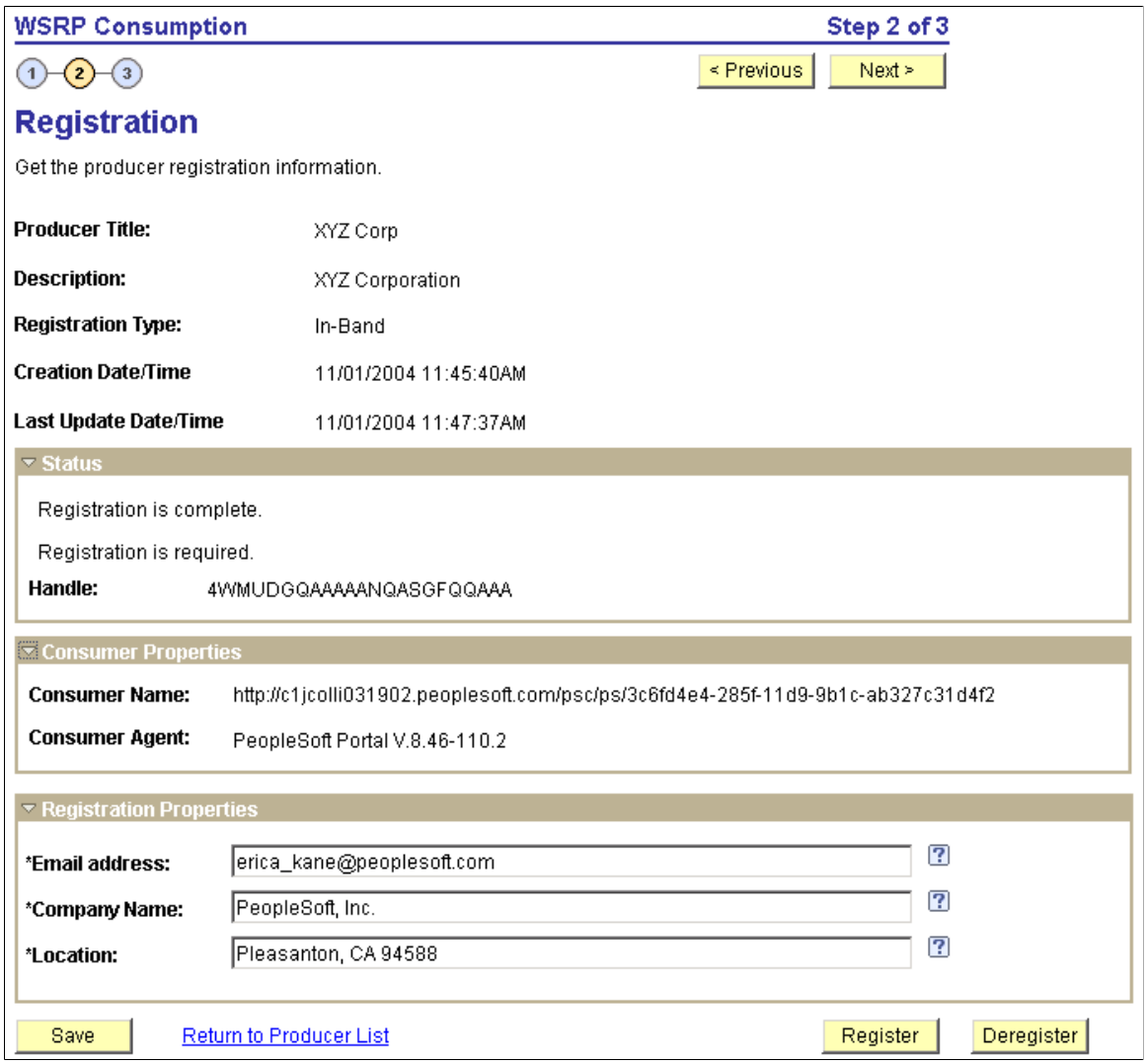
Save Click to save the producer information and access the Registration page.

Registering Producers

Access the Registration page. (Click the Next button on the Import Producer Information page.)

Image: Registration page

The following screenshot illustrates about registering producers on the Registration page.



Registration Type Choose a registration type:

In Band: In this process, the consumer sends a request to register with the producer, along with any properties required by the producer.

Out of Band: In this process, the producer and consumer specify manual processes such as email or phone conversations to establish registration.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

None: Registration is not required. No further information is required. Continue to the next page.

Status

Handle

Out of band: Enter the handle value conveyed to you by the producer.

In Band: Read-only. The value appears automatically when you click Register.

Validate

Click to check the validity of the handle.

Note: This button appears only when the registration type is *Out of band.*

Handle Status

Indicates the validity of the handle.

Note: This field appears only when the registration type is *Out of band.*

Note: Messages regarding registration status are also displayed within this group box.

Consumer Properties

Consumer Name and Consumer Agent

Displays the consumer information, (URL and software version), which is required by and sent to the producer.

Registration Properties

This group box appears only when the registration type is *In-Band.*

Note: The fields contained in this group box may vary according to each producer's requirements.

Register

Click to register the producer.

If registration is successful, the registration message in the Status group box is updated and the Deregister button appears.

Important! When you deregister a producer, the handle becomes invalid and all user portlet personalizations are removed. You will need to obtain a new handle before attempting a reregistration.

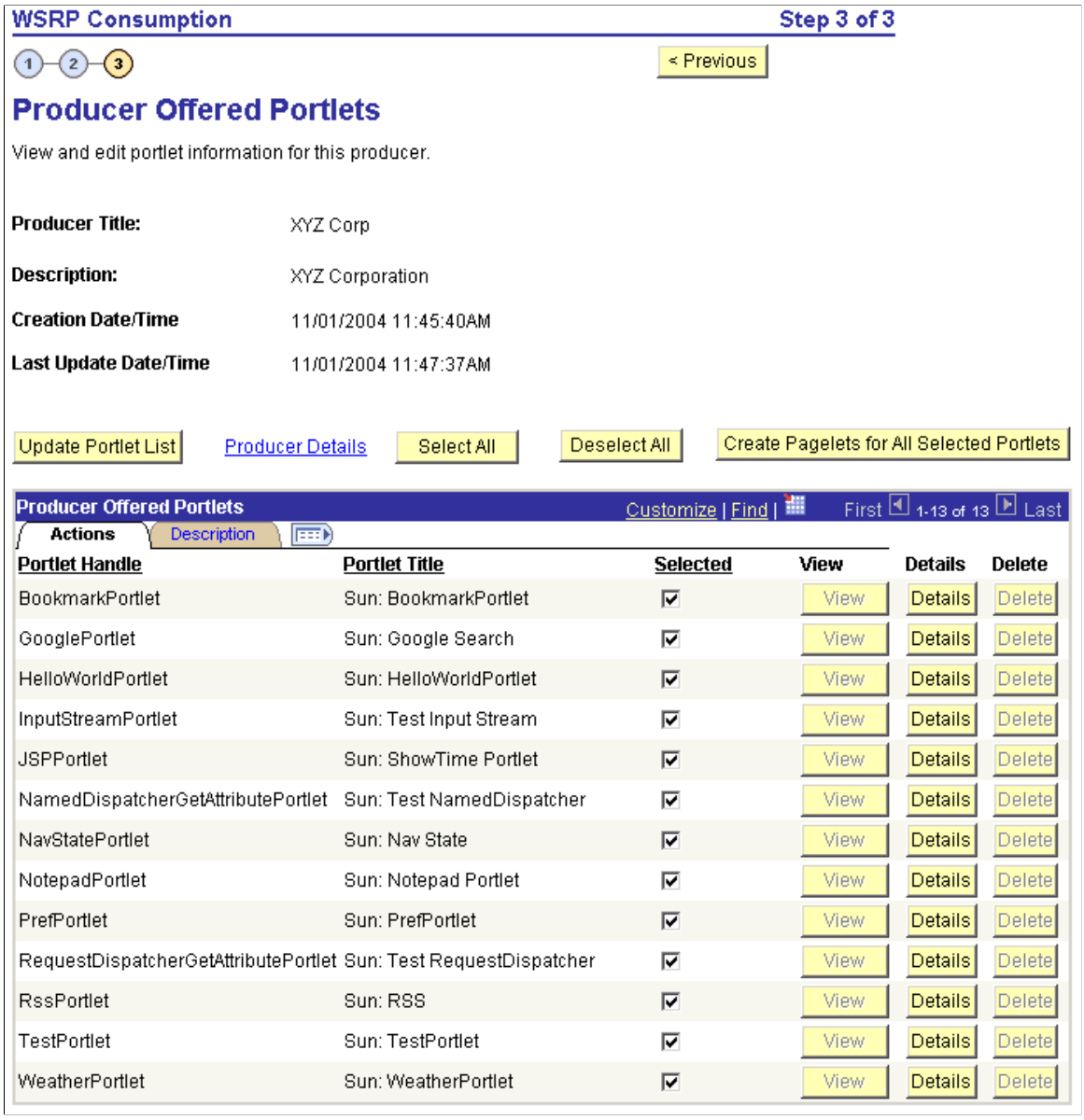
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating Content References

Access the Producer Offered Portlets page. (Click the Next button on the Registration page.)

Image: Producer Offered Portlets page

The following screenshot illustrates how to create content references on the Producer Offered Portlets page.



Update Portlet List — Click to refresh the list of offered portlets.

Important! When previously listed portlets are no longer offered by a producer, the Delete button associated with those portlets becomes active.

Producer Details — Click to go to the Producer Details page and view the producer's actual ServiceDescriptionResponse.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Create Pagelets for All Selected Portlets

Click to create content references for the selected portlets.

These content references can be located in a folder created automatically by means of the values entered for this producer name and title for its name.

This will make the portlets available when end users are personalizing their homepages.

Note: You can also create the content references manually in any portal using the Structure and Content page.

See Administering Content References.

Producer Offered Portlets

Selected

Select the portlets that you want to consume, then click the Create Pagelets for All Selected Portlets button to create content references.

You can also use the Select All and Deselect All buttons.

View

Click to go to the Portlet Content References page to view the newly created content reference in a new browser window. The page displays the portal where the new portlet resides, the name and label of the portlet, and a link to the content reference.

Note: This button is disabled until a content reference is created.

Details

Click to go to the Portlet Details page to view the XML that was returned from the producer for this portlet.

Delete

Click Update Portlet List to refresh the list of portlets that are being offered by this producer. The Delete button will become active for portlets that are no longer being offered.

Click the Delete button to remove the portlet and associated content reference.

Displaying Remote Portlets on a Homepage (Consumer)

This section discusses how to:

- Review content references.
- Display remote portlets on a homepage.

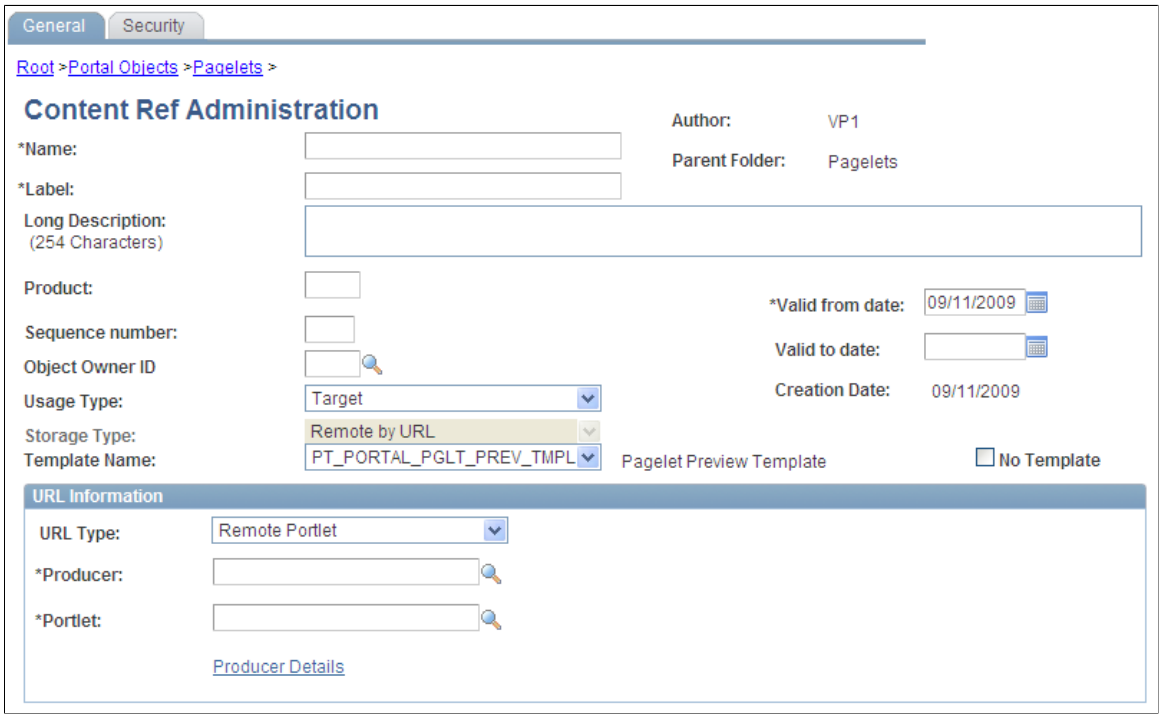
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Reviewing Content References

Access the Content Ref Administration page for a remote portlet. (Select PeopleTools, Portal, Structure and Content, Portal Objects, Pagelets. Click the desired folder name, then click Edit for the desired portlet.)

Image: Content Ref Administration page for a remote portlet

The following screenshot illustrates about reviewing content references on the Content Ref Administration page for a remote portlet.



The field values that are automatically entered are based on the values entered for the producer via the portal wizard. These include:

Name	The defaults is WSRP_*<portlet handle>*. This field is read-only and cannot be changed.
Label	Enter a meaningful label description for this portlet. The default is the remote portlet short title.
Parent Folder	The default is the producer's title. Click the Select New Parent button to move the content reference to a different folder.
Usage Type	The default is *Pagelet.*
Storage Type	The default is *Remote by URL.* This field is read-only and cannot be changed.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

URL Information

URL Type	The default is *Remote Portlet.*
Producer	The default is the producer's name.
Portlet	The default is the portlet handle.
Producer Details	Click to go to the consumption wizard's Import Producer Information page for this producer.

Note: You create content references manually for offered portlets update fields by using the same default values explained in this section.

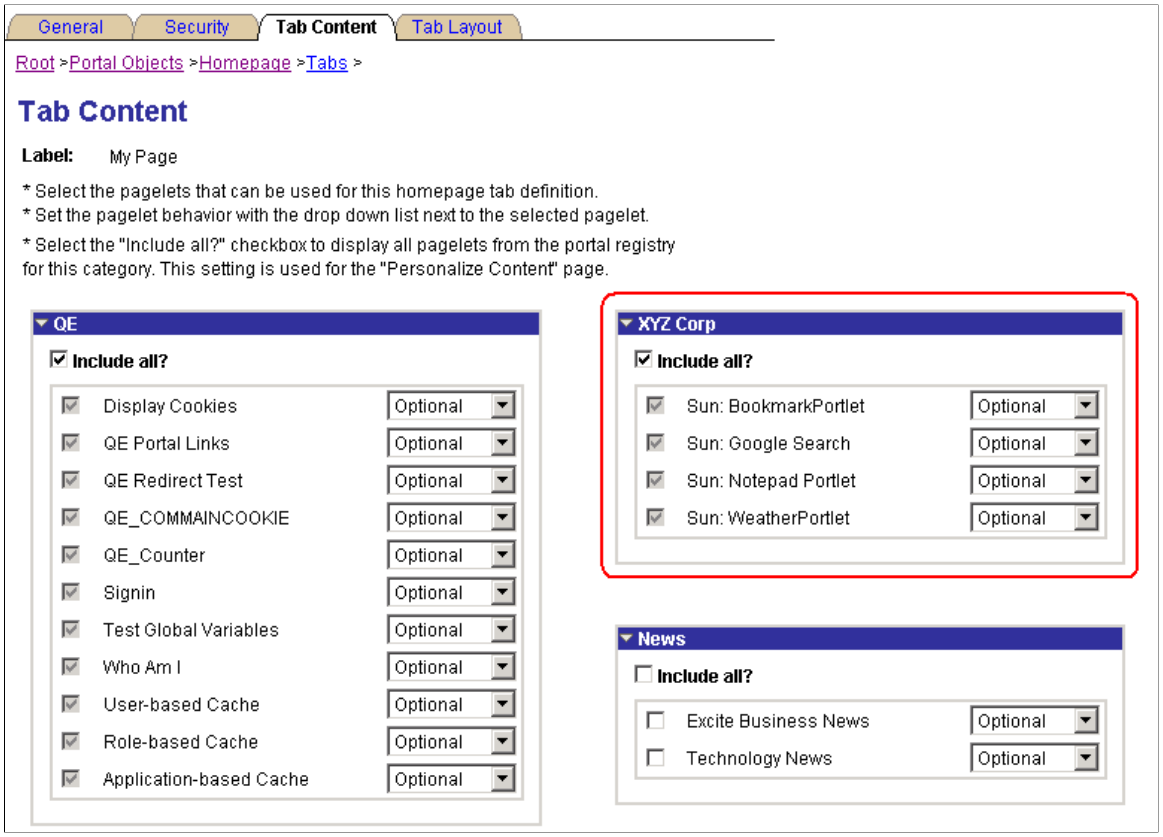
See Defining Content References.

Displaying Remote Portlets on Homepages

Access the Tab Content page. (Select PeopleTools, Portal, Structure and Content, Portal Objects, Homepage, Tabs. Click the Edit button associated with the My Page content reference. Select the Tab Content tab.)

Image: Tab Content page for remote portlets

The following screenshot illustrates remote portlets on the Tab Content page for remote portlets homepage.



Note: This procedure is performed by the administrator.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To make newly created remote portlets available to end users:

1. On the Tab Content page, select the required portlets, or select the Include all? check box to display all portlets from the portal registry for remote content group box.
2. Click Save.

   The selected portlets are now available for end users to display on their homepages.

To display the newly created remote portlets on a homepage:

1. On the homepage, click Personalize Content.
2. Select the remote portlets that you want to appear on your homepage.
3. Click Personalize Layout to organize your portlets.
4. Click Save to save your selections and return to your homepage.

Producing Remote Portlets (Producer)

This section discusses:

- Limitations on PeopleSoft content for WSRP consumption.
- Specify PeopleSoft content for WSRP production.
- Offer WSRP portlets.
- Work with WSRP content references.
- Set WSRP display mode.
- Map PeopleSoft styles to WSRP styles.
- Validate WSRP-compliance.
- Use PeopleSoft WSRP-produced content.

Limitations on PeopleSoft Content for WSRP Consumption

Please note the following limitations of producing PeopleSoft content to be consumed as a remote WSRP portlet:

- Related content and related actions cannot be consumed as WSRP content.
- WorkCenters cannot be consumed as WSRP content.
- Activity guide pagelets cannot be consumed as WSRP content.
- Gantt class, OrgChart class, and RatingBoxChart class objects cannot be consumed as WSRP content.
- While pages or pagelets with long text fields that have the rich text editor enabled can be consumed as WSRP content, the rich text editor itself will not be active in the WSRP content.

Specifying PeopleSoft Content for WSRP Production

This section discusses how to:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Specify pagelets for WSRP production.
- Specify components for WSRP production.
- Specify content references for WSRP production.

Specifying Pagelets for WSRP Production

Access the Pagelet Wizard - Specify Publishing Options page.

Use Pagelet Wizard to create pagelets that you want to publish as WSRP portlets. On the Specify Publishing Options page, you can select WSRP Portlet for Hompage pagelets and Template pagelets. This enables the pagelet's WSRP Producible option and causes the pagelet to appear on the Producer Offered Portlets page so that an administrator can expose it for remote, WSRP portal consumption. Also consider the Enable Caching option on the Specify Publishing Options page. If you select to enable caching, also select Caching Type and Minutes to Cache.

Note: When you create a pagelet using Pagelet Wizard and select the WSRP Portal option, the system automatically enables the pagelet for consumption by Oracle portals using the JPDK specification.

Specifying Components for WSRP Production

To enable a component for WSRP production, select the WSRP Compliant option on the Internet tab of the Component Properties dialog box.

See "Setting Internet Properties" (PeopleTools 8.54: Application Designer Developer's Guide).

Specifying Content References for WSRP Production

While you can also specify pagelets for WSRP production using Pagelet Wizard, you specify PeopleSoft components and iScripts intended for WSRP production using content reference options.

Note: Content references for pagelets (portlets) also control whether the pagelet can be WSRP-produced. For example, selecting the WSRP Producible option for a pagelet's content reference is equivalent to selecting the WSRP Portlet option on the Specify Publishing Options page in Pagelet Wizard. This applies also to component-based and iScript-based pagelets

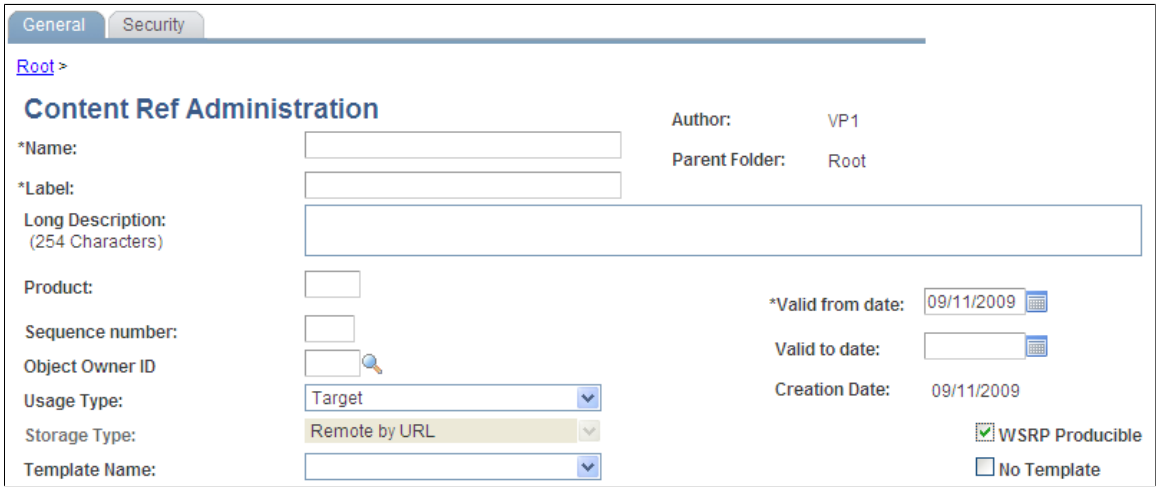
---

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Access the Content Ref Administration page (PeopleTools, Portal, Structure and Content..)

Image: Content Ref Administration page: WSRP Producible option

The following screenshot illustrates the WSRP Producible option on the Content Ref Administration page.



The WSRP Producible option appears only for content references with the usage type set to *Target* or *Pagelet,* with the URL type set to one of these values:

- *PeopleSoft Component*
- *PeopleSoft Generic URL*
- *PeopleSoft Script*
- *PeopleSoft Worklist URL*

The WSRP Producible check box appears just to the right of the Storage Type drop-down list box and, by default, it is not selected. Selecting this option causes the current content reference to appear on the Producer Offered Portlets page so that an administrator can choose to expose it for remote, WSRP portals. In turn, deselecting the option causes the current content reference to be removed from the list on the Producer Offered Portals page.

Because WSRP-producible content must originate from the local node, if a content reference is modified to reference a remote node, the WSRP Producible check box will not appear, and the content reference will not be able to be exposed to remote, WSRP portlets.

WSRP-producible content references possess a set of optional, WSRP-specific attributes, differentiating them from other content references. The following table contains descriptions of the WSRP content reference attributes:

Attribute	***Attribute Label***	***Description***	***Values***
WSRP_TITLE	WSRPTitle	Allows the exposed WSRP title to be exposed to remote WSRP portals.	Entered text.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Attribute	*Attribute Label*	*Description*	*Values*
WSRP_DESCR	WSRPDescription	Allows the exposed WSRP description to be different from the default content reference description.	Entered text.
WSRP_WINDOWSTATE	WSRPWindowState	Enables you to override the default window state behavior for portlets or target pages. The default window state without the content reference attribute override for portlets is *normal.* The default window state without the content reference attribute override for target content is *maximize.*	normal maximize

Content reference links created from content references displaying the WSRP Producible check box also display the WSRP Producible check box. By default, the option is not selected when you create a new content reference link, even if the WSRP Producible check box is selected for the associated content reference. As with the content reference, if you select the WSRP Producible option on the content reference link, it will appear on the Producer Offered Portlets page.

When working with WSRP-producible content references, also consider the following points:

- Portal registry folders do not display the WSRP Producible check box.
- Context Manager template pagelets, considered by the system to be a type of portlet, are WSRP-producible. If keys or parameters expected by a supporting transaction are unavailable, the content will appear the same way as keys being unavailable in the PeopleSoft portal.
- If a WSRP-producible content reference is moved to a different location in the portal registry, it still retains its ability to be WSRP-produced.
- The Test Content Reference link only tests how the content reference will appear in the PeopleSoft portal, not WSRP.
- If you have a content reference that should be available only for WSRP and not through the PeopleSoft portal, hide the content reference from the PeopleSoft navigation system.

Offering WSRP Portlets

Access the Producer Offered Portlets page. (Select PeopleTools, Portal, WSRP Production.)

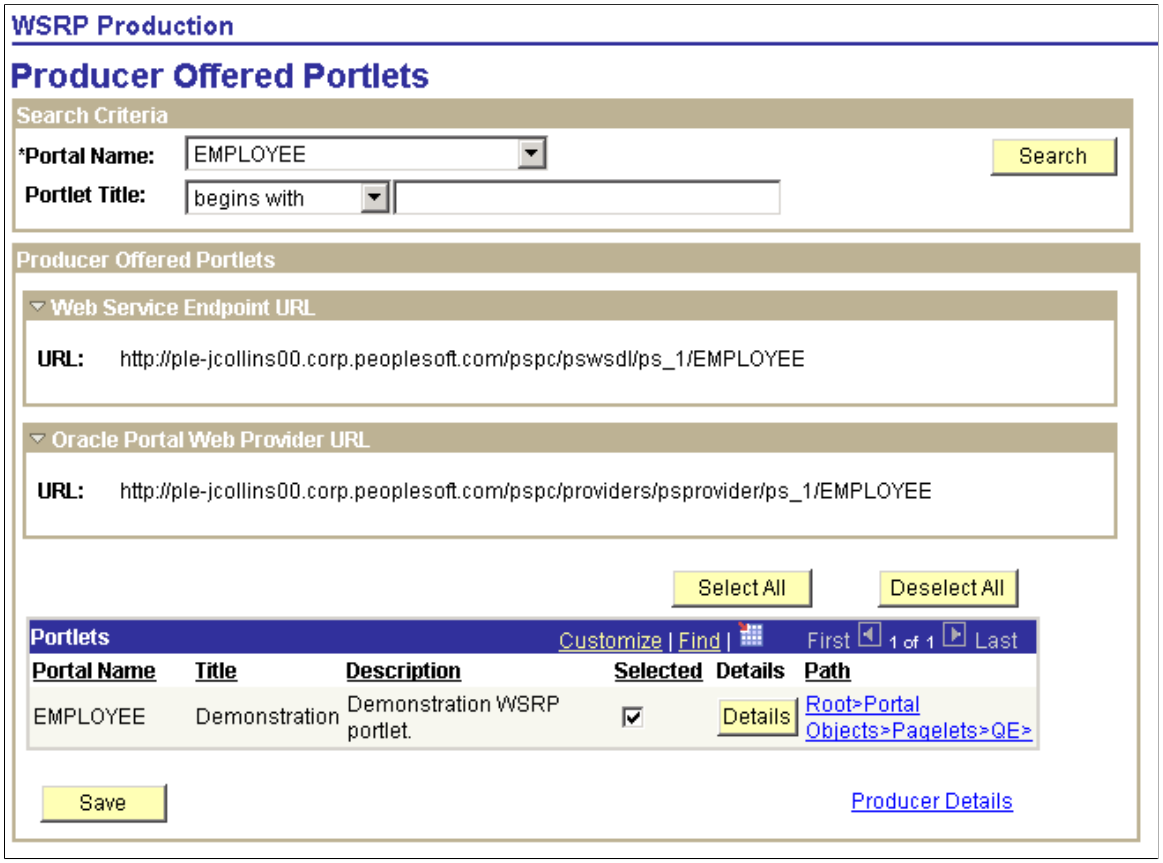
The Producer Offered Portlets page displays PeopleSoft provider content that has been selected for WSRP production. That is, any pagelet appearing on this page has the WSRP Portlet option selected in Pagelet

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Wizard, and any content reference that appears on this page has the WSRP Producible option selected on the Content Ref Administration page.

Image: Producer Offered Portlets page

The following screenshot illustrates WSRP properties on the Producer Offered Portlets page.



Portal Name

Choose the portal where the content you want to offer is located, and click Search.

Note: To display portlets within all portals, select *All Portals* and omit the Portlet Title value.

Portlet Title

Enter the portlet title or a portion of it for increased search options. Use the % character for wild card values.

Web Service Endpoint URL

Displays the URL that the WSRP consumer portal requires to import the offered content.

Oracle Portal Web Provider URL

Displays the URL that the JPDK-compliant Oracle Portal requires to import the offered content.

Note: This value applies only for Pagelet Wizard pagelets.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Portlets

This section of the page is named Portlets, despite the fact that the content being exposed could be components, pages, or iScripts. Regardless, the content is exposed and discovered by WSRP-consuming portals as portlets.

Selected	Select the portlets that you want to offer to consumers. Click the Select All button to offer all of the portlets listed. Click the Deselect button to remove all of the selections. If a content item in the list is not selected, it cannot be exposed to consumer portals. **Note:** By default, each new item appearing in the portlets list will have the Selected check box selected.
Details	Click to access the Portlet Details page, which displays the generated XML that is sent to a consumer.
Path	Click to go to the Content Ref Administration page to view the CREF information for this portlet.
Producer Details	Click to go to the Producer Details page to view the service description XML that is sent to the consumer for all portlets in this portal.

Note: If a WSRP-producible content reference is deleted from the portal registry, the system removes it from the Producer Offered Portlets page.

Working with WSRP Content References

Use the Content Ref Administration page to view WSRP content reference information.

Important! Use the Content Ref Administration page only to select a new parent folder or update the security for the pagelet.

Setting WSRP Display Mode

You can set the WSRP display mode using the system-wide, WSRP Display Mode option on the PeopleTools Options page (PeopleTools, Utilities, Administration, PeopleTools Options). 'This option enables you to control how PeopleSoft content is rendered when user interaction is initiated from within a WSRP-consuming portal. This may override the WSRP-compliant setting within the component. You can select these settings:

Display as Portlet: Renders content accessed from a link in a portlet inline the same way as a portlet, provided that the component has the WSRP Compliant property enabled. As with PeopleSoft applications and new windows, opening a new window, versus replacing the window, is not possible due to session and state management issues.

Replace Existing Window: All content that is navigated to from a portlet will be rendered in the same browser window but will take the user out of the consuming portal and into the PeopleSoft portal to render the content.

If a pagelet is WSRP-produced, and the end user clicks a link on the pagelet to take the user to a different page from the PeopleSoft system, the following numbered list describes the logic flow that determines whether:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The link is displayed inline in the portlet window of the WSRP-consuming portal.
- The end user is taken out of the WSRP-consuming portal and into the PeopleSoft portal to render the PeopleSoft content.

1. Determine WSRP display mode.
   - If set to *Replace Existing Window,* display the PeopleSoft content in the PeopleSoft portal within the same window.
   - If set to *Display as Portlet,* determine whether content is WSRP-compliant.
2. Determine WSRP compliance status.
   - If WSRP compliance is not selected, display PeopleSoft content in the PeopleSoft portal within the same window.
   - If WSRP compliance is selected, display the PeopleSoft content inline as a portlet.

Related Links

"PeopleTools Options" (PeopleTools 8.54: System and Server Administration)

Mapping PeopleSoft Styles to WSRP Styles

For WSRP content, PeopleSoft style classes are mapped to corresponding WSRP styles. PeopleSoft applications provide PeopleSoft WSRP-produced markup references with WSRP style names only, not PeopleSoft styles. If a particular PeopleSoft style is not mapped to a corresponding WSRP style, the style cannot be generated in the WSRP markup. The appearance and character of the style in the WSRP-consuming portal is determined by the WSRP CSS style sheets on the consuming portal.

PeopleSoft-to-WSRP styles adhere to WSRP specification 1.0 (10.6). To modify the WSRP mapping, use the WSRP Style drop-down list box on the Miscellaneous tab for a style definition in Application Designer.

Note: Compared to PeopleSoft style options, WSRP style options are limited.

Validating WSRP Compliance

To ensure that PeopleSoft pages, iScripts, and portlets comply with WSRP standards, use the WSRP validation options in PeopleSoft Application Designer. You can validate a single project or an entire database. Results of the validation appear in the Validate output tab, as with any other validate operation. The first line of the WSRP validation results is "Begin Validate WSRP Compliance," with the last line being "End Validate WSRP Compliance."

The system validates WSRP compliance against the following definitions:

- HTML areas within pages.
- HTML objects.
- JavaScript objects.
- Templates.
- Application classes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- PeopleCode.
- PeopleCode iScripts.

The validation causes errors after identifying the existence of any of the following keywords. The use of these keywords is not recommended:

- href="http://
- frameset
- src="http://
- .location
- action="http://
- top.
- parent.
- parent.frames
- window.
- IFRAME
- ILAYER

Performing WSRP Validation for a Project

To perform WSRP validation for a project:

1. In Application Designer, open the project containing the WSRP definitions.
2. Select Tools, Options and select the Validate tab.
3. On the Validate tab, select the Validate WSRP compliance check box, and click OK.

   Note: By default, this option is not selected.

4. Select Tools, Validate project.

Performing WSRP Validation for a Database

To perform WSRP validation for a database:

1. Launch Application Designer, and sign in to the appropriate database.
2. Select Tools, Validate Database for WSRP.

Using PeopleSoft WSRP-Produced Content

This section discusses the differences between the behavior of PeopleSoft content within the PeopleSoft portal and a consuming WSRP portal:

- The main PeopleSoft menu portlet does not provide navigation of WSRP-consumed content. The PeopleSoft WSRP content needs to be discovered through the consumer portlet's navigation.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Hot keys (such as Ctrl + J for the System Information page) are available as long as they do not conflict with any hot keys defined on the consuming system.
- Pagebar functionality is not included in PeopleSoft WSRP-produced content.

  For example, the New Window link or the Multi-Language drop-down list box will not appear (so multilanguage functionality is *not* supported for WSRP content). The Copy URL functionality is available on the System Information page only.
- Personalizations, such as Tab-Over, made through the My Personalizations page apply to WSRP content.
- Only PeopleSoft portlets and iScripts support Edit capability, not components.

Note: PeopleSoft portlets, including an Edit URL, do not support multiple instances of the of the portlet in the consuming portal.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Chapter 23

Working with JSR 168 Portlets

Understanding JSR 168

JSR 168 is a standard for portals running in a Java 2 Platform, Enterprise Edition (J2EE) architecture. It specifies content definitions much like pagelets, called portlets, that present portal content defined according to the JSR 168 standard. Portlets are Java-based web components that you can register in the PeopleSoft portal just as you register pagelets. Both HTML templates and frame templates support portlet content.

Portlets generate fragments of markup (such as HTML, Extensible HyperText Markup Language, and Wireless Markup Language). The PeopleSoft portlet engine manages and invokes portlets by using Java programs. This combines markup fragments that are generated by different portlets into a portal page.

You develop your own Java-based JSR 168-compliant portlets, and then register them by using the PeopleSoft portal registration components.

You expose your locally installed portlets to remote WSRP consumers by creating entries in the portal entity registry.

See Describing the Portlet Entity Registry.

Understanding PeopleSoft Compliance with JSR 168

This section discusses:

- Java Portlet API.
- Window states.
- Content caching.
- Portlet modes.

Java Portlet API

Portlets obtain their context via the Portlet API functions such as access to user profile information for the current user, participation in the portal window and action event model, access to web client information, sharing of information with other portlets, and a standard way of storing and retrieving per-user/per-instance portlet data persistently. PeopleTools portlet container supports the Java Portlet API 1.0. The Java Portlet API provides a URL-rewriting mechanism for creating links to trigger actions within a portlet without requiring knowledge about how URLs are structured in the particular web application. Portlets are grouped in a portal application by bundling them in a single web application archive file (WAR) with a portlet deployment descriptor file. In addition, the API provides a means of sharing data among portlets of the same portal application.

Window States

The PeopleSoft JSR 168 engine accepts all of the required window states as well as any custom window states. However, although the engine allows any window state, PeopleSoft JSR 168 does not react to them. Portlets always appear to have a *Normal* amount of screen space.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Content Caching

In the current release, the PeopleSoft JSR 168 engine doesn't support the caching of portlet content based on caching elements that are embedded in the portlet; any caching tags in the portlet are ignored. However, the engine does support caching by using the PSCACHECONTROL content reference attribute.

See Implementing Pagelet Caching.

Portlet Modes

The PeopleSoft JSR 168 engine supports all of the required portlet modes, which are:

- VIEW
- EDIT
- HELP

Configuring the Portlet Container

If the *logClientIPOnMalformedRequest* initialization parameter is set to *true,* it will log the address of clients making failed attempts to access the Java portlets (due to malformed URLs).

If the *AccessJavaPortletsFromPeopleSoftPortalOnly* initialization parameter is set to *true,* only clients who have their REMOTE_USER setting will be allowed access. By default, all accesses from the user's homepage supply the REMOTE_USER setting.

Note: You can secure all of the Java portlets on your system using J2EE security and role-based permissions within your portlet logic.

The parameter can be found in the PortletContainerServlet definition in the web.xml of the /pspc application.

The following code is the relevant section from the /pspc/WEB-INF/web.xml within the domain:

```
<servlet>
   <servlet-name>PortletContainerServlet</servlet-name>
   <display-name>PeopleSoft PortletContainer Servlet</display-name>
   <servlet-class>com.peoplesoft.pt.portlet.container.ContainerServlet</servlet-cla⇒
ss>
       <!-- <load-on-startup>0</load-on-startup> -->
   <init-param>
       <param-name>logClientIPOnMalformedRequest</param-name>
       <param-value>true</param-value>
   </init-param>
   <init-param>
      <param-name>AccessJavaPortletsFromPeopleSoftPortalOnly</param-name>
      <param-value>false</param-value>
   </init-param>
</servlet>
```

Describing the Portlet Entity Registry

The portlet entity registry is the source for both the PeopleSoft portlet content references and the portlets produced through WSRP.

Choosing Portlets to Produce for WSRP

You can choose which portlets to produce for WSRP by including them as <portlet> elements in portletentityregistry.xml.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To produce portlets for WSRP:

1. Edit portletcontainer/WEB-INF/data/portletentityregistry.xml to add an <application> element for your new portlet application.

   The following code is the portletentityregistry.xml file as delivered, showing the production of four portlets for WSRP consumption, WSRPTestPortlet of the wsrptest portlet application, TestPortlet1 and TestPortlet2 of the testsuite portlet application, and the HelloPortlet of the helloportletapp portlet application.

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-entity-registry>
    <application id="">
        <definition-id>helloportletapp</definition-id>
        <portlet id="">
            <definition-id>helloportletapp.HelloPortlet</definition-id>
        </portlet>
    </application>
    <application id="">
        <definition-id>wsrptest</definition-id>
        <portlet id="">
            <definition-id>wsrptest.WSRPTestPortlet</definition-id>
        </portlet>
        <portlet id="">
            <definition-id>wsrptest.WSRPTestPortlet</definition-id>
        </portlet>
    </application>
    <application id="">
        <definition-id>testsuite</definition-id>
        <portlet id="">
            <definition-id>testsuite.TestPortlet2</definition-id>
        </portlet>
        <portlet id="">
            <definition-id>testsuite.TestPortlet1</definition-id>
        </portlet>
    </application>
</portlet-entity-registry>
```

   The portletentityregistry.xml requires that:

   - Your portlet application must have a new <application> element in which the <definition-id> is the name of your portlet application directory and the ID is unique within the file.
   - Any portlets that you want to produce from your portlet application must have their own <portlet> elements within your new <application> element. Use the delivered portletentityregistry.xml as a guide.

     Note: Each portlet must have a unique ID within the <application>. For example, refer to the same definition multiple times—each having different preferences and its own unique entity ID.

   - The <definition-id> of each portlet is a period-separated, fully qualified name for the portlet of the form: “<porletApplicationName>.<portletName>” as shown. The portletApplicationName portion must match the <definition-id> of the <application>. The portletName portion must match the <portlet-name> element for the <portlet> definition in your portlet.xml file.

2. Restart your web server. Your Java portlets are now produced via WSRP technology. All WSRP consumers that you may use will need to retrieve the new service description, which will include the new Java portlets.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix A

Understanding Changes in Portal Configuration Settings

Understanding Changes in Portal Configuration Settings

In the current release, PeopleSoft portal configuration has moved from a file-based environment to a PeopleSoft Pure Internet Architecture interface. This topic provides a complete listing that maps PeopleTools 8.43 and earlier portal settings to the current PeopleTools web profile settings. Within each of several broad functional categories, the release 8.43 properties are listed alphabetically, alongside the corresponding web profile fields that replace them.

The portal configuration files still exist in the current PeopleTools release, but they now contain only a small number of properties, and they're not meant to be edited. Don't modify any of the following files unless directed to do so by a PeopleSoft representative:

- Configuration.properties
- Pstools.properties
- Browserprops.xml
- Cookierules.xml

Related Links

Configuring Web Profiles

General Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - General page:

Release 8.43 Configuration Property	***Corresponding Web Profile General Field***
AuthTokenDomain	Authentication Domain
compressCacheFiles	Compress Response References
compressResponse	Compress Responses
compressMimeTypes	Compress MIME Types (compress Multipurpose Internet Mail Extensions types)
CompressReportOutput	Compress Report Output field in the Reports group box.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Release 8.43 Configuration Property	*Corresponding Web Profile General Field*
CompressReportOutputNetscape	Compress Report Output field in the Reports group box.
enableNewWindow	Enable New Window
enableProcessingWait	Enable Processing Message
enableReportRepository	Enable Report Repository field in the Reports group box.
helpUrl	Help URL
physicalpath	Non-standard Base Path
ReportRepositoryPath	Report Repository Path field in the Reports group box.
saveConfirmDisplayTime	Save Confirmation Display Time
singleThreadNS	Single Thread Netscape
threadDelay	Single Thread Delay

Related Links

Configuring General Portal Properties

Security Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Security page:

Release 8.43 Configuration Property	*Corresponding Web Profile Security Fields*
byPassSignOn	Allow Public Access field in the Public Users group box.
defaultPWD	Password field in the Public Users group box.
defaultUSERID	User ID field in the Public Users group box.
defaultXMLLinkUSERID	User ID field in the XML Link group box.
defaultXMLLinkPWD	Password field in the XML Link group box.
portalUseHttpForSameServer	PIA use HTTP Same Server (PeopleSoft Pure Internet Architecture use HTTP same server).
sessionTimeout	Authenticated Users - Inactivity Logout field in the Authenticated Users group box.
SSLRequired	Secured Access Only field in the SSL group box.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Release 8.43 Configuration Property	*Corresponding Web Profile Security Fields*
timeoutWarningScript	Timeout Warning Script and Override fields in the Authenticated Usrs group box.
userIDCookieAge	Days to Autofill User ID
UseSecureCookieWithSSL	Secure Cookie with SSL field in the SSL group box.
warningTimeout	Inactivity Warning field in the Authenticated Users group box.
XMLLinkUseHttpForSameServer	XML Link Use HTTP Same Server field in the XML Link group box.

You can now set these properties from the pstools.properties file by using the Web Profile Configuration - Security page:

Release 8.43 Pstools Property	*Corresponding Web Profile Security Field*
tuxedo_network_disconnect_timeout	Disconnect Timeout field in the Web Server Jolt Settings group box.
tuxedo_receive_timeout	Receive Timeout field in the Web Server Jolt Settings group box.
tuxedo_send_timeout	Send Timeout field in the Web Server Jolt Settings group box.

Related Links

Configuring Portal Security

Virtual Addressing Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Virtual Addressing page:

Release 8.43 Configuration Property	*Corresponding Web Profile Virtual Addressing Fields*
defaultPort	Port field in the Default Addressing group box.
defaultScheme	Protocol field in the Default Addressing group box.
pswebservername	Name field in the Default Addressing group box.
relativeURL	Generate Relative URLs
RPS	Protocol, Host, HTTP, and HTTPS fields in the Reverse Proxy Server List group box.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Related Links

Configuring Virtual Addressing

Cookie Rules

You can now set these elements and attributes from the cookierules.xml file by using the Web Profile Configuration - Cookie Rules page:

Release 8.43 Cookierules.xml Element or Attribute	*Corresponding Web Profile Cookie Rules Field*
block domain	Cookies Not Passed to Server
cookie name	Cookie Pattern
delete_on_logout	Delete Cookie on Logout
forward domain	Cookies Passed to Server
proxied	Proxied
secure	Secure

Related Links

Configuring Cookie Rules

Caching Settings

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Caching page:

Release 8.43 Configuration Property	*Corresponding Web Profile Caching Field*
CachePurgeAllHitCount	Cache Purge All Hit Count field in the On the Web Server group box.
CacheTargetContent	Cache Target Content field in the On the Web Server group box.
chartdirphys	Chart Directory field in the On the Web Server - Directories group box.
chartdirweb	Chart Web Directory field in the On the Web Server - Directories group box.
cssdirphys	CSS Directory (cascading style sheet directory) field in the On the Web Server - Directories group box.
cssdirweb	CSS Web Directory (cascading style sheet web director) field in the On the Web Server - Directories group box.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Release 8.43 Configuration Property	*Corresponding Web Profile Caching Field*
enableBrowserCache	Cache Generated HTML field in the On the Browser group box.
enableNoVersion	Copy Image/CSS (No Versioning) (copy image/cascading style sheet [no versioning]) field in the On the Web Server - Directories group box.
imagedirphys	Image Directory field in the On the Web Server - Directories group box.
imagedirweb	Image Web Directory field in the On the Web Server - Directories group box.
jsdirphys	JavaScript Directory field in the On the Web Server - Directories group box.
jsdirweb	JavaScript Web Directory field in the On the Web Server - Directories group box.
maxSavedState	Number of States Supported field in the On the Browser group box.
PortalCacheHomepageOnBrowser	Cache Homepage field in the On the Browser group box.
PortalCacheObjects	Cache Portal Objects field in the On the Web Server group box.
PortalCacheStaleInterval	Cache Stale Interval field in the On the Web Server group box.
PortalHomepageStaleInterval	Homepage Stale Interval field in the On the Browser group box.
portalUseCachedProxiedJS	Cache Proxied JavaScripts field in the On the Web Server group box.

You can now set these elements and attributes from the browserprops.xml file by using the Web Profile Configuration - Caching page:

Release 8.43 Browserprops.xml Element or Attribute	*Corresponding Web Profile Caching Field*
property name="CacheHomePage"	Cache Home Page field in the Browsers group box.
useragent id	User Agent ID field in the Browsers group box.

Related Links

Configuring Caching

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Trace and Debug Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Debugging page:

Release 8.43 Configuration Property	*Corresponding Web Profile Debugging Field*
connectionInformation	Show Connection Information
debug_inlinejavascript	Show JavaScript Inline HTML
debug_inlinestylesheet	Show StyleSheet Inline HTML
debug_overlap	Show Overlapping Fields
debug_savefile	Create File from PeopleSoft Pure Internet Architecture HTML Page
debug_showlayout	Show Layout
enableDebugDumpFile	Write Dump File
enableTrace	Show Trace Link at Signon
testing	Generate HTML for Testing

Related Links

Configuring Trace and Debug Options

Look and Feel Settings

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Look and Feel page:

8.43 Configuration Property	*Corresponding Web Profile Look and Feel Fields*
authtokenenabled_page	Auth Token Enable Page field in the Other Pages group box.
chgPwdOnExpire	Change Password On Expire and Override fields in the Password group box.
chgPwdOnWarn	Change Password On Warning and Override fields in the Password group box.
cookiesrequired_page	Cookies Required Page field in the Other Pages group box.
exception_page	Exception Page field in the Error Pages group box.
expire_page	Page field in the Expire Page group box.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

8.43 Configuration Property	*Corresponding Web Profile Look and Feel Fields*
expirePage_ContentName	Content Name field in the Expire Page group box.
logout_page	Logout Page field in the Signon/Logout Pages group box.
mcfAuthFailure_page	MCF Auth Failure Page field in the Error Pages group box.
passwordexpired_page	Password Expired Page field in the Password group box.
passwordwarning_page	Password Warning Page field in the Password group box.
signon_page	Signon Page field in the Signon/Logout Pages group box.
signonError_page	Signon Error Page field in the Signon/Logout Pages group box.
signonresultdoc_page	Signon Result Doc Page field in the Signon/Logout Pages group box.
signontrace_page	Enable Trace Page field in the Other Pages group box.
sslrequired_page	SSL Required Page field in the Other Pages group box.
start_page	Page field in the Start Page group box.
startPageScript	Script and Override fields in the Start Page group box.
userprofile_page	User Profile Page field in the Other Pages group box.

You now specify the locale properties from the pstools.properties file as default language mappings on the Manage Installed Languages page. You can specify additional mappings from International Organization for Standardization (ISO) locale codes to PeopleSoft language codes by using the Web Profile Configuration - Look and Feel page.

Related Links

Configuring Trace and Debug Options

"Adding New Language Codes to the System" (PeopleTools 8.54: Global Technology)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix B

Configuring the Application for 8.4x Navigation

Understanding Configuration of PeopleSoft Applications for 8.4x Navigation

The 8.5x navigation is based on iframe templates and features drop-down navigation. The 8.4x navigation is based on frame (HTML) templates and uses menu pagelet navigation. These topics describe the two tasks that you must complete to configure an 8.5x application for 8.4x navigation.

These topics describe how to:

- Reconfigure the default template.
- Reconfigure the navigation pagelet.

Important! HTML templates are deprecated as of PeopleTools 8.50. Many features introduced or enhanced after release 8.50 require iframe templates to function as designed. Features that are not supported in HTML templates include: modal windows and prompts; drop-down navigation; related content; WorkCenter pages; and SmartNavigation.

Reconfiguring the Default Template

To reconfigure the default template:

1. Select PeopleTools, Portal, Structure and Content.
2. In the Folders group box, click Portal Objects, and then click Templates.
3. Click the Edit link for the 8.4x default template.
4. In the Field Name field, delete the existing value and enter *DUMMY.*

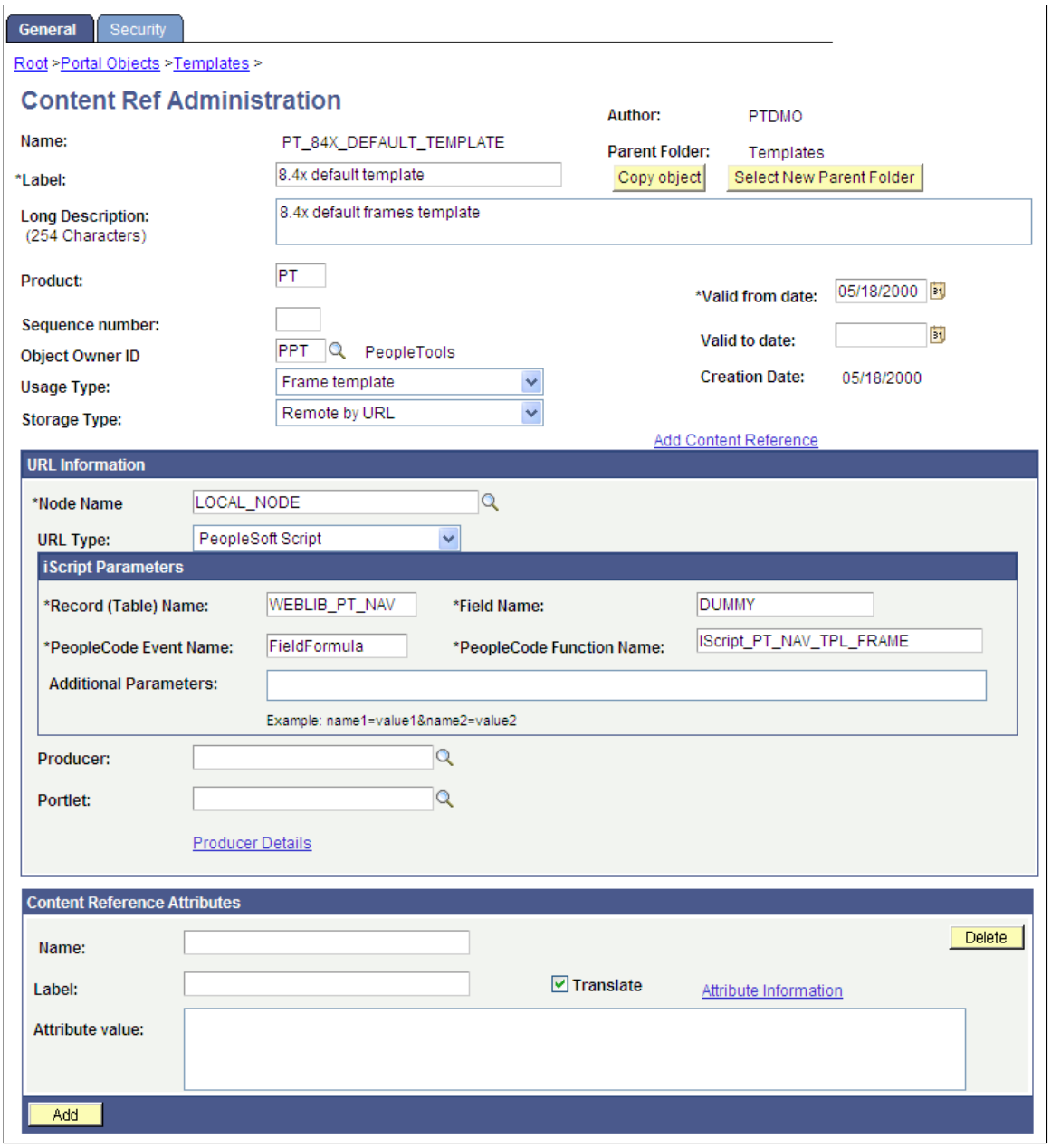
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

5. Save the component.

Image: 8.4X DEFAULT_ TEMPLATE template after reconfiguration

After completing the preceding reconfiguration steps, the 8.4X DEFAULT_ TEMPLATE template will appear similar to the following:

The following screenshot is an example of 8.4X DEFAULT_ TEMPLATE template.



6. In the navigation path, click the Templates link.

7. Click the Edit link for the 8.50 default template.

8. In the Field Name field, replace *ISCRIPT1* with *ISCRIPT2*.

9. Change the template Usage Type from *Inline frame template* to *Frame template*.

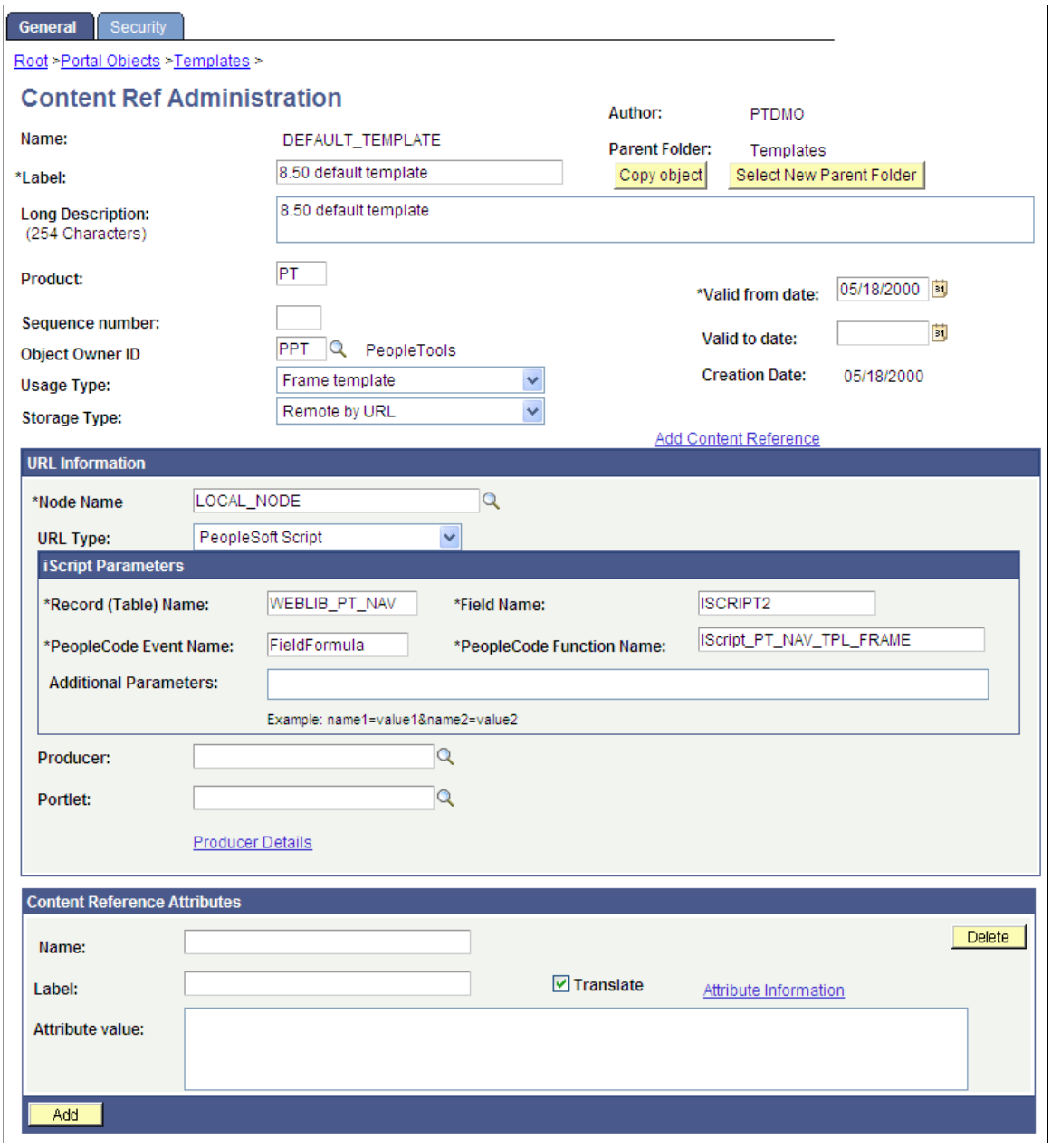
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

10. Save the component.

Image: 8.50 DEFAULT_ TEMPLATE template after reconfiguration

After completing the preceding reconfiguration steps, the 8.50 DEFAULT_ TEMPLATE template will appear similar to the following:

The following screenshot is an example of 8.50 DEFAULT_ TEMPLATE template.



Reconfiguring the Navigation Pagelet

To reconfigure the Menu navigation pagelet:

1. Select PeopleTools, Portal, Structure and Content.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

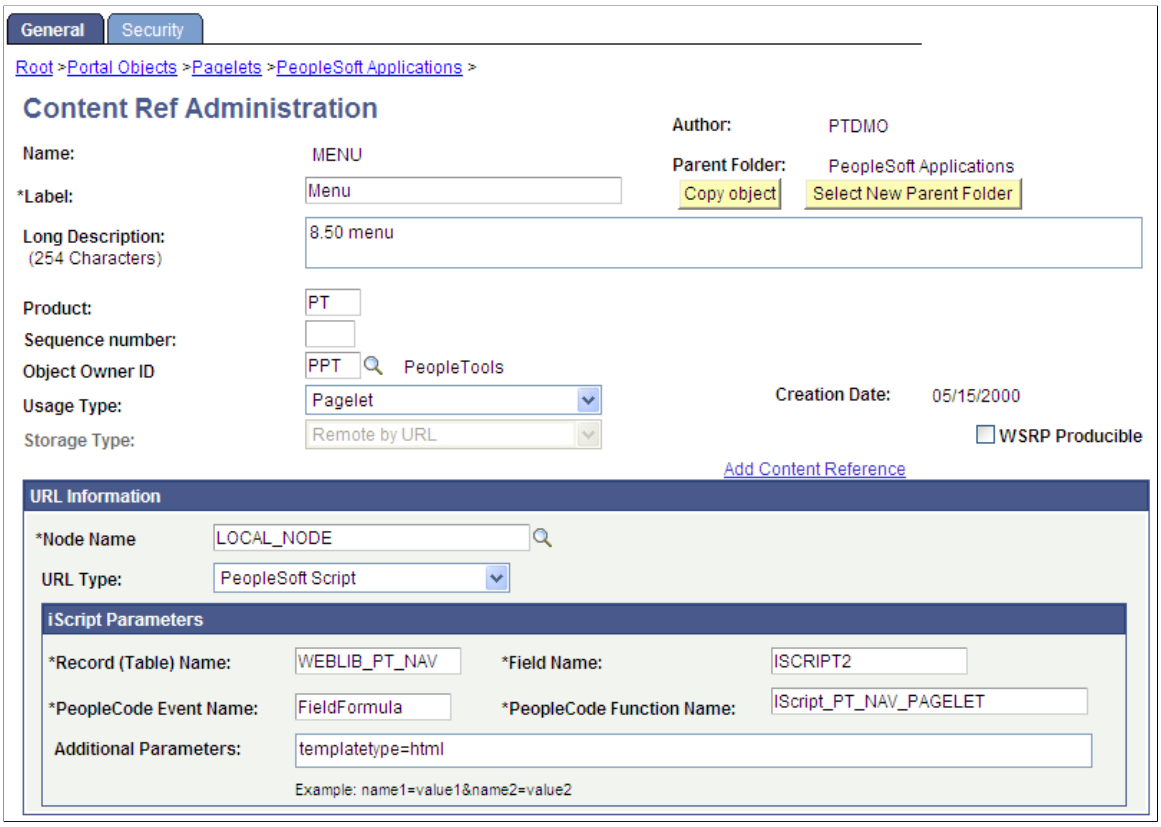
2. In the Folders group box, click Portal Objects, and then click Pagelets.
3. Click the PeopleSoft Applications link, and then click the Edit link for the Menu.
4. In the Field Name field, delete the existing value and enter *ISCRIPT2*.
5. Save the component.

   The content reference will look like this:

   Image: Menu pagelet after reconfiguration

   After completing the preceding reconfiguration steps, the Menu navigation pagelet will appear similar to the following:

   The following screenshot is an example of Menu navigation pagelet.



6. In the breadcrumbs, click the PeopleSoft Applications link.
7. Click the Edit link for Menu - Classic.
8. In the Label field, replace *Menu - Classic* with *Menu.*
9. In the Field Name field, replace *ISCRIPT2* with *ISCRIPT1*.
10. Save the component.

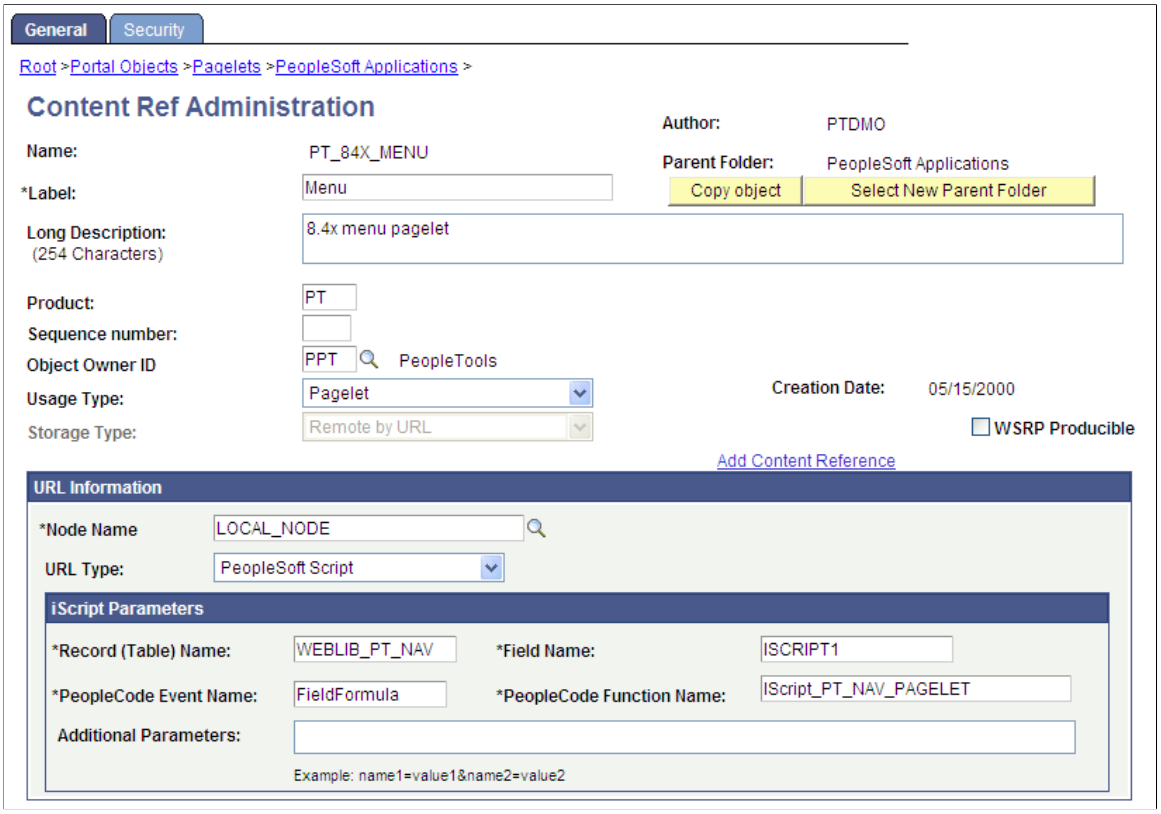
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The content reference will look like this:

Image: Menu - Classic pagelet after reconfiguration

After completing the preceding reconfiguration steps, the Menu - Classic navigation pagelet will appear similar to the following:

The following screenshot shows how a classic pagelet looks after reconfiguring the Navigation pagelet.



11. Stop and restart the application and web servers.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix C

Converting Frames-Based Templates to iFrame-Based Templates

Understanding Conversion of Frames-Based Templates to iFrame-Based Templates

This topic provides a detailed example of converting an existing frames-based template (eProcurement Shopping Cart template) to the 8.5x drop-down navigation template.

Converting a frames-based template to an iframe-based template enables you to take advantage of the new features in the iframe, drop-down navigation template: related content; AJAX-based, drop-down navigation; modal, key-specific favorites improvements; and the recently used menu functionality.

Important! The following conversion example assumes that the current frames-based template you are converting uses a storage type of *Local in HTML (catalog)*. If your frames template is a *Remote by URL* template, your steps will be slightly different.

Converting the eProcurement Shopping Cart Template

This section describes how to:

- Gather information about the current template.
- Create the iframe template iScript.
- Create the iframe template HTML definition.
- Set security for the template iScript.
- Convert the template content reference.
- Troubleshoot errors.

Gathering Information About the Current Template

To gather information about the current template:

1. Select PeopleTools, Portal, Structure and Content.
2. Within Structure and Content, click Portal Objects, then click Templates.
3. Click the Edit link next to the ePro Shopping Cart Template.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. Copy the text in the HTML Area field. Save it to a text file. After you have changed the template type, this HTML will be removed.

5. Click the Security tab and note the current security for this content reference.

   This step is important if specific permissions or roles have been set. Later, you apply these permissions to the new iScript that will be created.

Creating the iFrame Template iScript

To create the iframe template iScript:

1. Open the WEBLIB_PV.ISCRIPT.FieldFormula PeopleCode in edit mode.

2. At the bottom of the PeopleCode, enter the following code:

```
Declare Function ptIframeTemplateHNav PeopleCode WEBLIB_PT_NAV.ISCRIPT1 FieldF⇒
ormula;

/*
  * builds the inline frame template with the drop-down navigation
*/
Function IScript_PVShopCart_Template

   /* parameter 1: HTML object name that will be used to generate the inline f⇒
rame template */
       %Response.Write(ptIframeTemplateHNav("PV_SHOP_CART_HNAV_TEMPLATE"));

End-Function;
```

Note: For *Remote by URL* templates, use the existing web library iScript and just modify the PeopleCode as shown previously.

Creating the iFrame Template HTML Definition

To create the iframe template HTML definition:

1. In Application Designer, open the PT_HNAV_TEMPLATE HTML definition.

2. Clone the definition and save it as the value that was used in parameter 1 of the new iScript.

3. Open the text file that contains the original template HTML and locate the pagelets that you need to move to the new template, for example:

```
<frame name="CART" id="CART" frameborder="no" scrolling="auto" noresize src=""⇒
 style="border-top:ridge #CCCCCC 5px;">
     <IClientComponent Name="Links">
         <Source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT">dummy</Source>
     </IClientComponent>
</frame>
```

4. Change the HTML so that it looks like this:

```
<div id="pvcartpagelet">
     <iframe name="CART" id="CART" frameborder="no" src="">
          <pagelet name="Links">
                <source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT"/>
          </pagelet>
     </iframe>
 </div>
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: The iframe template does not support the `IClientComponent` template meta tag. These meta tags are deprecated and are replaced by the `pagelet` meta tag.

5. Locate where you want to position the Links pagelet in the PV_SHOP_CART_HNAV_TEMPLATE iframe template. If the pagelet is to exist to the right or left of the target content, place the pagelet inside of the iframe content container `<div>`. In this scenario, the pagelet will be located to the right of the target content.

```
                  <div id="ptifrmcontent">
  <div id="ptifrmtarget">
    <iframe id="ptifrmtgtframe" name="TargetContent" title="%bind(:6)" framebo⇒
rder="0" scrolling="auto" onload="ptrc.onLoadTC()" src="">
      <target Name="target" params="%bind(:7)"/>
    </iframe>
  </div><div id="pvcartpagelet">
    <iframe name="CART" id="CART" frameborder="no" src="">
      <pagelet name="Links">
        <source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT"/>
      </pagelet>
    </iframe>
  </div>
  %bind(:8)
  <script type="text/javascript" src="%bind(:9)"></script>
</div>
```

6. Add the style information for the new pagelet. Ideally, these styles should go into a style sheet to take advantage of style sheet caching on the web server and browser. For this example, the embedded style approach places the style in the head section of the template.

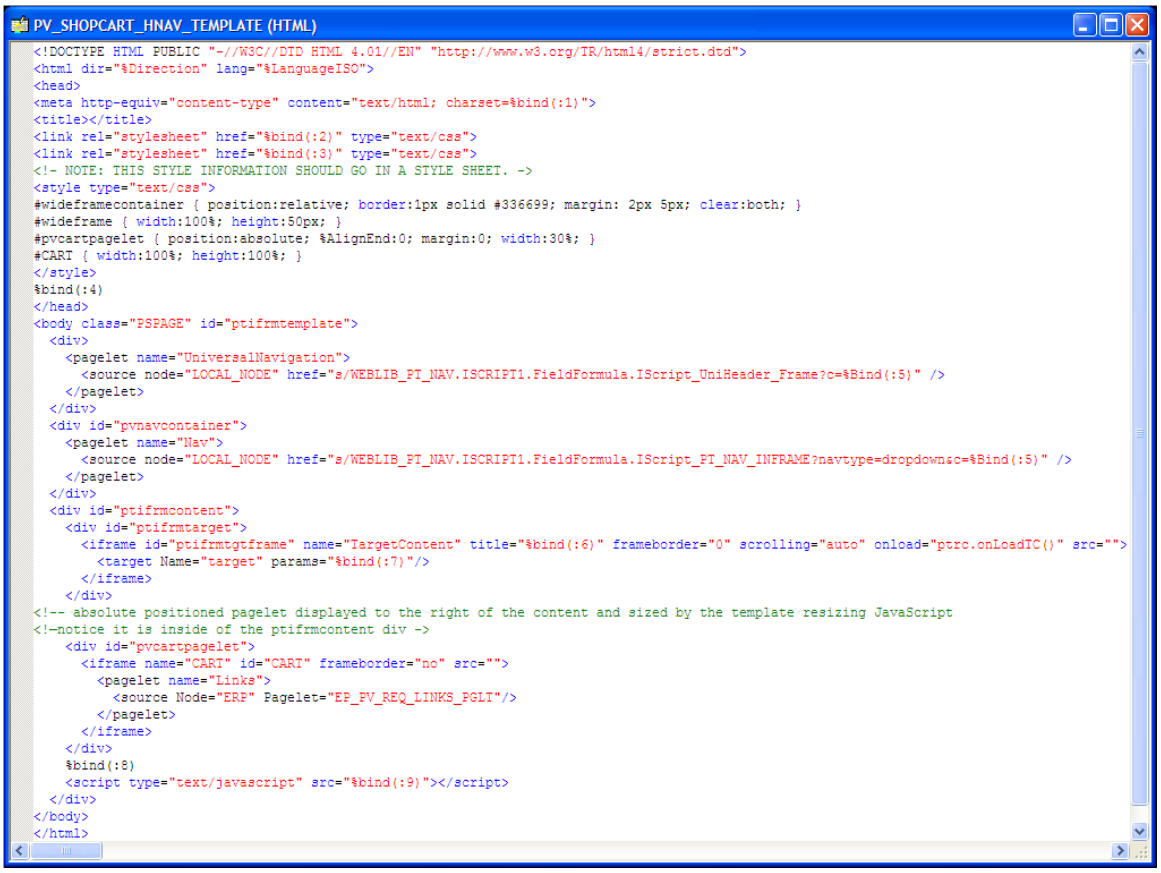
```
<style type="text/css">
/* style for the CART pagelet container div */
#pvcartpagelet {
  position:absolute; /* required: the position must be fixed so the iframe can⇒
 be sized properly */

  %AlignEnd:0;       /* meta CSS to position the location of the pagelet */
  margin:0;
  width:30%;         /* required:must be a percentage,value so the correct wid⇒
th is rendered when the browser is resized */
}

/* style for the CART iframe */
#CART { width:100%; height:100%; }
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
</style>
```

Image: Test HTML object: PV_SHOPCART_HNAV_TEMPLATE with shopping cart pagelet

The following screenshot is an example of full HTML definition that contains template with shopping cart pagelet.



If a pagelet needs to expand to the width of the page, the following HTML and style information can be used:

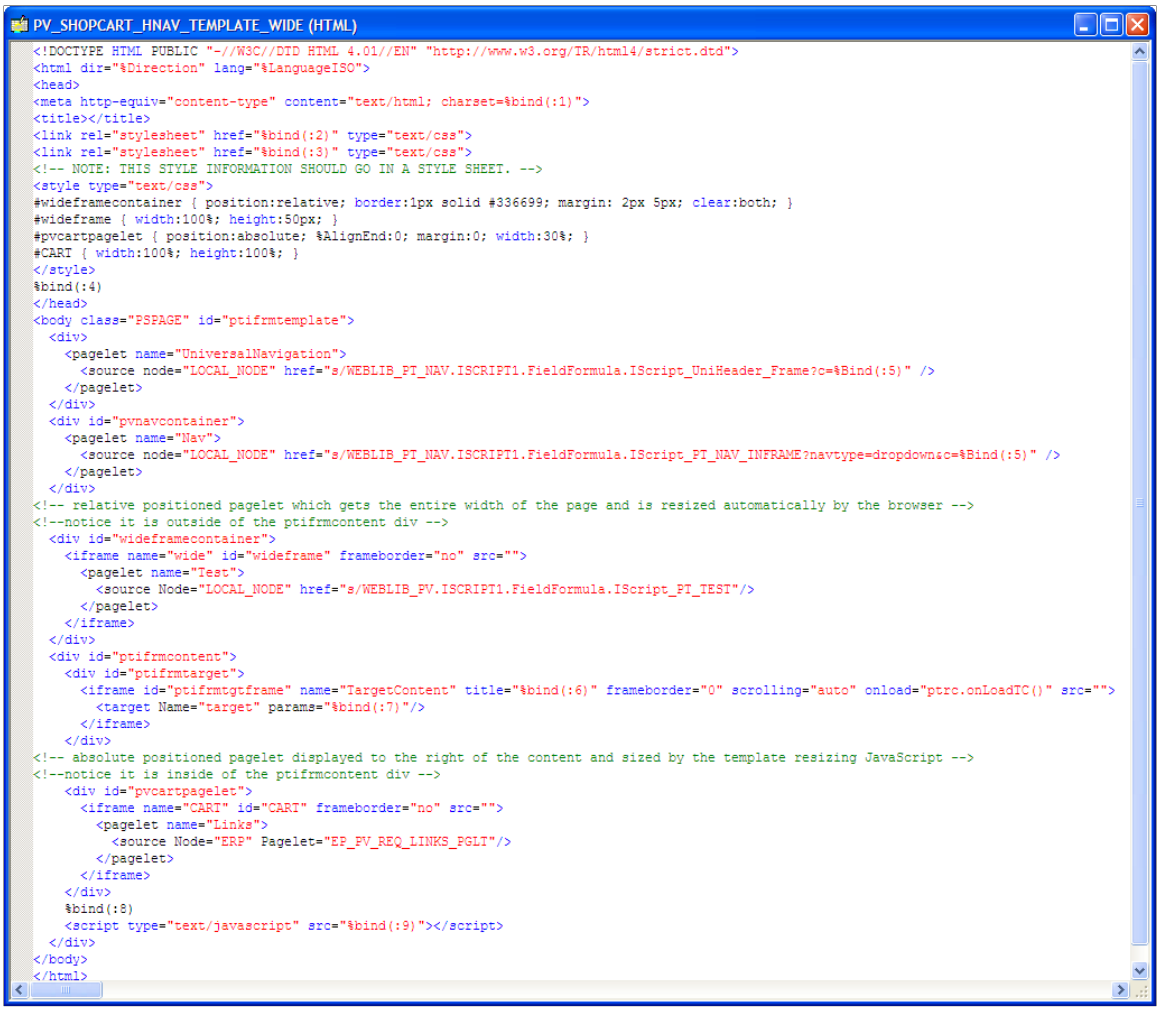
```
<style type="text/css">
#wideframecontainer {
  position:relative;                        /* required: */
  border:1px solid #336699;
  margin: 2px 5px;
  clear:both;
 }
#wideframe { width:100%; height:50px; }
</style>
...
<!--must be located outside of the content container div-->
<div id="wideframecontainer">
    <iframe name="wide" id="wideframe" frameborder="no" src="">
      <pagelet name="Test">
        <source Node="LOCAL_NODE" href="s/WEBLIB_PV.ISCRIPT1.FieldFormula.IScr⇒
ipt_PT_TEST"/>
      </pagelet>
    </iframe>
```

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
</div>
```

Image: Test HTML object: PV_SHOPCART_HNAV_TEMPLATE with shopping cart and wide pagelet

The following screenshot is an example of a full HTML definition that contains a wide pagelet:



Setting Security for the Template iScript

To set security for the template iScript:

1. Select PeopleTools, Security, Permissions & Roles, Permission Lists.
2. Enter the permission list that you noted when you gathered information about the template earlier.
3. Enable Web Library security for the permission list.
4. Continue until all permission lists and roles are added.

Note: If the template content reference security is public, only one permission list is needed.

Note: For *Remote by URL* templates, this step is optional because the iScript already exists.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Converting the Template Content Reference

To convert the template content reference:

1. Select PeopleTools, Portal, Structure and Content.
2. Within Structure and Content, click Portal Objects, then click Templates.
3. Click the Edit link next to the ePro Shopping Cart Template.
4. Change the usage type to *inline frame template.*
5. Change the storage type to *Remote By URL.*
6. Enter the iScript record, field, and PeopleCode function names.
7. Click the Save button.
8. Stop and restart the application and web servers.
9. Test the template.

Note: For *Remote by URL* templates, omit steps 5, 6, and 7.

Troubleshooting Errors

This table describes some common errors and possible causes:

Error	***Solution***
JavaScript errors	Check existing pagelet JavaScript. Make sure to correct statements like this: `parent.parent.frames["UniversalHeader"] /* no longer valid in an iframe template */ parent.frames[3]. /* use a iframe name or an iframe id */`

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix D

Replacing SmartNavigation Images

Understanding SmartNavigation Icons

SmartNavigation folder and content reference icons appear next to each folder and menu item element in the SmartNavigation menu hierarchy. The application generates the icons for the folders and menu items by using:

- Image definitions.
- Style sheet definitions.
- PeopleCode function calls.

Image Definitions

SmartNavigation folders and content references use these images by default:

Default folder image.

Default content reference image.

Note these points about the images and style sheets:

- The dimensions of the images must be 14 pixels by 14 pixels.
- You must use PeopleSoft Application Designer to save the images in the portal site.
- You must include a style class for each image—a folder image and a content reference image—in a free form style sheet.
- You must use PeopleSoft Application Designer to save the style sheet in the portal site.

Sample Style Sheet Code

Each style class defines the background style and specifies the image identifier.

```
.fldrdynimage {
background: transparent url(%Image(Folder_Image_ID)) no-repeat center center;
}

.crefdynimage {
background: transparent url(%Image(Cref_Image_ID)) no-repeat center center;
}
```

Note: You do not have to use the same class names that you see in the sample code.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Creating Replacement Images

To create replacement images:

1. Use the graphic application of your choice to create the replacement images.
2. Open PeopleSoft Application Designer in the portal site database and create an image definition for each of the images .

   See "Creating Image Definitions" (PeopleTools 8.54: Application Designer Developer's Guide).
3. Save each image definition.
4. Make note of the image name. You must reference the image name, or image ID, in the style sheet.

Defining New Style Classes

To define new style classes:

1. Open PeopleSoft Application Designer in the portal site database and create a free form style sheet definition.

   See "Understanding Style Sheets and Style Classes" (PeopleTools 8.54: Application Designer Developer's Guide), “Free Form Sub Style Sheet.”
2. Enter text to create the style sheet. Use the format that you see here:

```
.fldrdynimage {
background: transparent url(%Image(Folder_Image_ID)) no-repeat center center;
}

.crefdynimage {
background: transparent url(%Image(Cref_Image_ID)) no-repeat center center;
}
```

   Note: You must reference the images you created in PeopleSoft Application Designer.

   See Modifying Folder Settings.
3. Save the style sheet.

Modifying Folder Settings

To modify folder settings:

1. Access the Folder Administration page (Select PeopleTools, Portal, Structure and Content. Click the SmartNavigation folder that you want to change).
2. In the Style Sheet field, enter the name of the style sheet that you created in the previous step.

   See Understanding Folder Navigation.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

See Administering Folders.

See Managing Folders and Content References.

3. Save the page.

Writing PeopleCode

The code that you write depends upon the data source. This section discusses the PeopleCode for:

- Rowset data sources.
- Tree data sources.

Rowset Data Source

When you use a rowset as the SmartNavigation folder data source, use the following function call in the PeopleCode program to generate folder and content reference icons in the menu:

```
LoadABN(&rs_DataSource, &rs_ChartRowset, &rs_RelatedActions, GetABNNode(&reqParams)⇒
, GetABNInitialNode(&reqParams), "fldrdynimage", "crefdynimage");
```

This table describes the function parameters:

Parameter	***Description***
fldrdynimage	Folder image class ID (name)
crefdynimage	Content reference image class ID (name)

The class id parameters are optional. If you want to use the default icons, make one of the two following calls:

```
LoadABN(&rs_DataSource, &rs_ChartRowset, &rs_RelatedActions, GetABNNode(&reqParams)⇒
, GetABNInitialNode(&reqParams), "", "");
```

```
LoadABN(&rs_DataSource, &rs_ChartRowset, &rs_RelatedActions, GetABNNode(&reqParams)⇒
, GetABNInitialNode(&reqParams));
```

Related Links

"LoadABN" (PeopleTools 8.54: PeopleCode Language Reference)

"GetABNNode" (PeopleTools 8.54: PeopleCode Language Reference)

"GetABNInitialNode" (PeopleTools 8.54: PeopleCode Language Reference)

Tree Data Source

When you use a tree as the SmartNavigation folder data source and you also use PeopleCode, you use function calls in the program to generate:

- Images for tree nodes (folders).

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- Images for tree leaves (content references).

Images for Tree Nodes

When you generate the menu item of a tree node, you must pass both the folder image class name and the content reference image class name because a tree node without children appears as a content reference. Use the following function calls in the PeopleCode program to generate folder icons in the menu and create the HTML fragment that the portal consumes that creates the drop-down menu items:

```
&ChildNode.GenABNMenuElementWithImage(GetABNInitialNode(&reqParams), "fldrdynimage"⇒
, "crefdynimage");
GenHTMLMenu(&szLI, "folderdynimage");
```

This table describes the function parameters:

Parameter	***Description***
folderdynimage	Folder image class name
crefdynimage	Content reference image class name

Images for Tree Leaves

When you generate the menu item of a tree leaf, you pass the content reference image class name. Use the following function calls in the PeopleCode program to generate content reference icons in the menu and create the HTML fragment that the portal consumes that creates the drop-down menu items:

```
&ChildLeaf.GenABNMenuElementWithImage("crefdynimage");
GenHTMLMenu(&szLI, "crefdynimage");
```

This table describes the function parameters:

Parameter	***Description***
crefdynimage	Content reference image class name

Default Images

If you want the default icons to appear, make the following calls in the PeopleCode program to generate folders and content references in the menu and create the HTML fragment that the portal consumes that creates the drop-down menu items:

```
&ChildNode.GenABNMenuElementWithImage(GetABNInitialNode(&reqParams), "", "");

&ChildLeaf.GenABNMenuElementWithImage("");

GenHTMLMenu(&szLI, "");

GenHTMLMenu(&szLI);
```

You must pass an empty string as the parameter for both the folder image class name and the content reference image class name.

Note: You must pass parameter values for the class name into the Tree API methods because the Tree API is a psoftapi object; psoftapi object methods do not allow optional parameters.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

The class name parameter is optional in the GenHTMLMenu function. If you want the default icons to appear, make one of the two following function calls:

- `GenHTMLMenu(&szLI, "")`
- `GenHTMLMenu(&szLI);`

Oracle recommends that you use the first function call in the previous list because it offers more flexibility in the event that additional parameters are added to the GenHTMLMenu function.

Note: If you use a tree as the datasource *without* PeopleCode, the default images always appear as the folder and content reference elements in the menu.

Related Links

"GenHTMLMenu" (PeopleTools 8.54: PeopleCode Language Reference)

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix E

Related Content Reserved Words and Service Parameters

Reserved Words Default Mapping

When you have a service using any of the following reserved words as service parameters, they will have some default mapping done when you first access the Map Fields page. Here is the list of related content-specific reserved words, their service parameters, and the default mapping data.

This table lists the reserved words and their default mapping:

Reserved Word	***Description***	***Default Map Type***
PTCS_CREFNAME	Current content reference (cref) name for which the service assignment and mapping is being done.	Fixed value.
PTCS_CREFLABEL	Current related content transaction cref label.	Fixed value.
PTCS_NODENAME	%Node.	System variable.
PTCS_PORTALNAME	%Portal.	System variable.
PTCS_KEYFIELD1	Component level 0 key1 if available.	Key field.
PTCS_KEYFIELD2	Component level 0 key2 if available.	Key field.
PTCS_KEYFIELD3	Component level 0 key3 if available.	Key field.
PTCS_KEYFIELD4	Component level 0 key4 if available.	Key field.
PTCS_PAGENAME	%Page.	System variable.
PTCS_SERVICEID	Current related content service ID.	Fixed value.
PTCS_RELURL	Relative URL of the content reference.	Fixed value.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix F

Configuring UPK Topics as Related Content Services

Understanding Oracle UPK and Related Content

You can configure Oracle User Productivity Kit (UPK) modules and topics as component-level and page-level related content services. These sections provide a brief overview of Oracle UPK and related content and describe how to:

- Create UPK-based related content service definitions.
- Assign UPK-based services as related content.

Note: In these sections, the word *topic* is used to represent both UPK topics and UPK modules unless specified otherwise.

Oracle UPK Developer and content is an end-user training solution that provides a single-source development and customization tool to enable rapid content creation and customization. UPK also can provide system process documentation, web-based training materials, role-based performance support, and other types of documentation.

PeopleSoft Related Content Framework is a set of PeopleTools definitions and processes that enable developers and subject matter experts to link application pages with contextually relevant content. The primary goal of related content services is to provide immediate access to relevant, contextual information for any PeopleSoft transaction without requiring any additional user effort. PeopleSoft Related Content Framework acts as an enterprise mashup, tying together all types of content such as Oracle Business

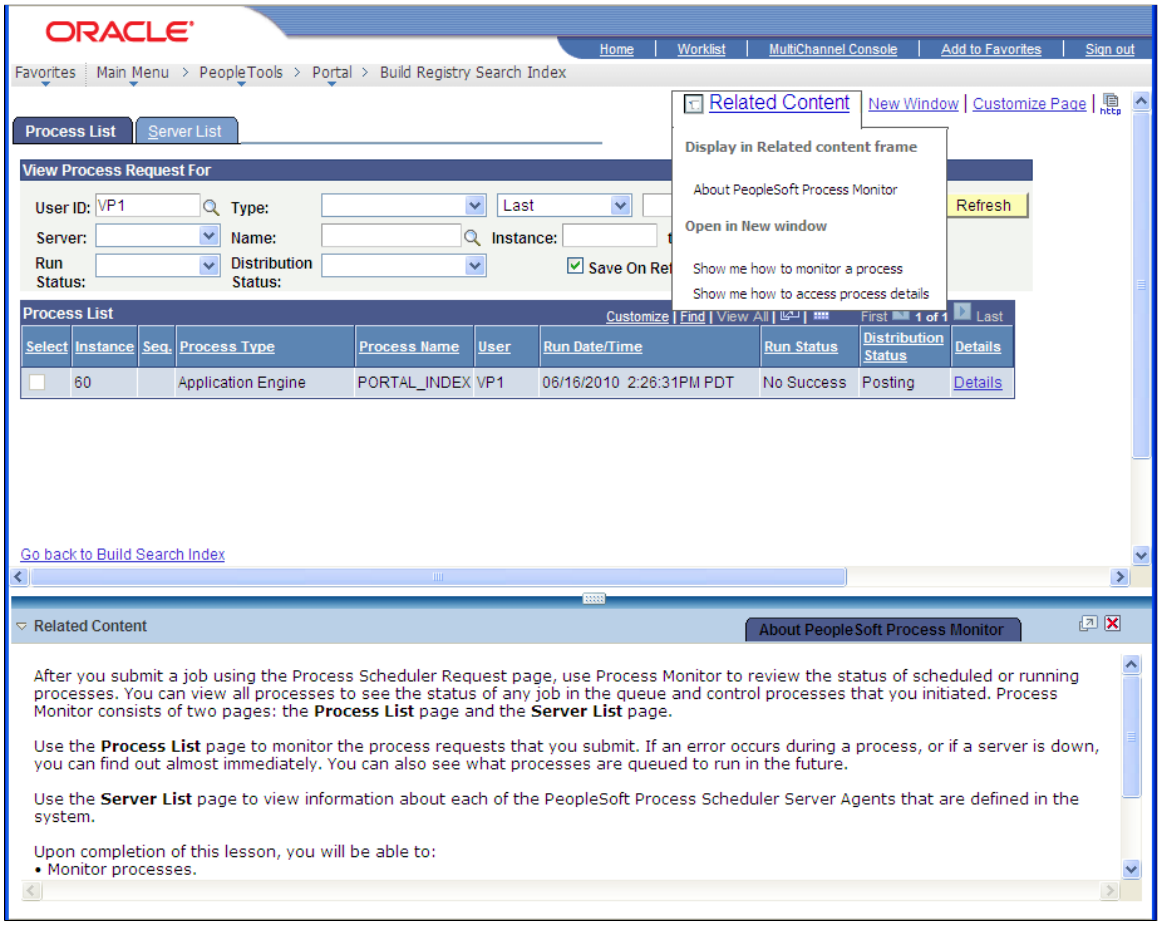
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Intelligence Enterprise Edition (OBIEE) analytics, queries, discussion threads, documentation, and relevant non-PeopleSoft data into a single tool—PeopleSoft application pages.

Image: UPK links under the Related Content contextual menu with a conceptual topic in the related content frame

You can use the related content framework to set up access to UPK modules and topics so that any UPK content is available at the page or component levels under the Related Content menu as shown here:

The following screenshot helps user understand Oracle UPK and related content in the related content frame.



Prerequisites

This table lists the necessary prerequisites for integrating UPK and related content and provides references to information about these prerequisites:

Prerequisite	*Reference*
Perform all related content configurations.	See Understanding Oracle UPK and Related Content.
Create, deploy, and configure UPK player package to your web server.	See Oracle's *PeopleSoft Enterprise Help Menu Integration* See Oracle's User Productivity Kit *In-Application Support Guide*

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

UPK Player Package Links

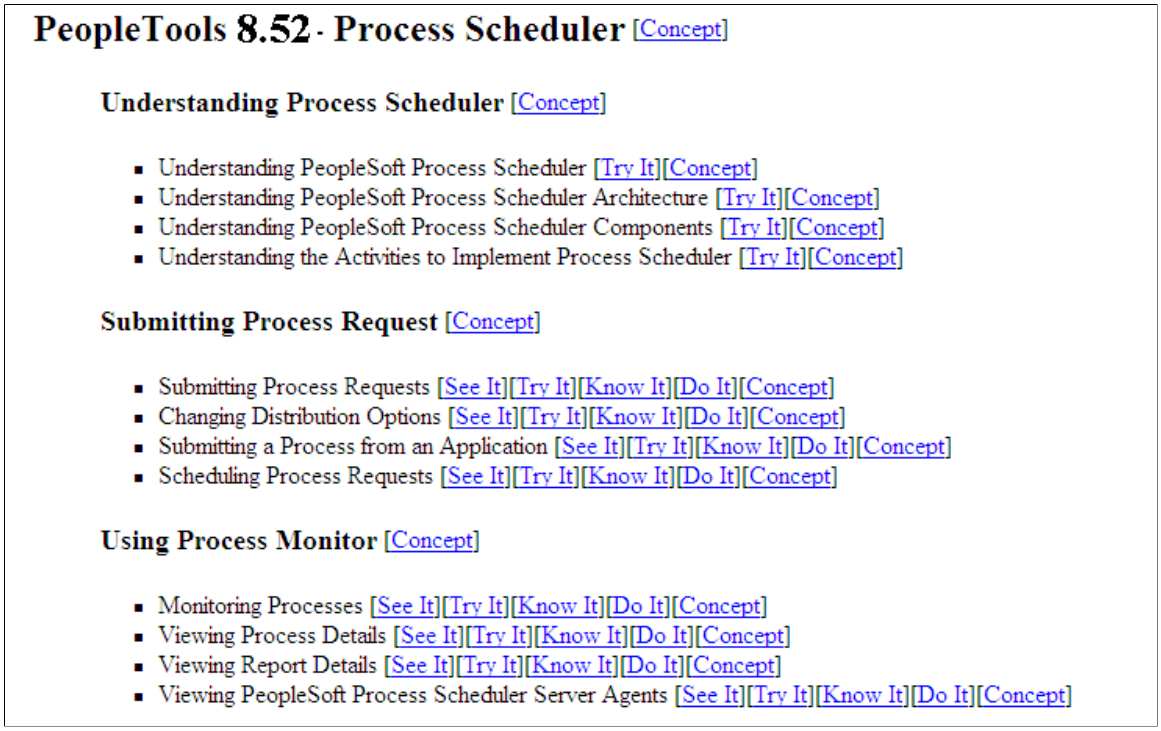
The UPK administrator will have access to the player package link (URL). The link should look similar to this:

http://*upkservername*/upk/*content_subject*/toc.html

If you change the player package link ending from *toc.html* to *kp.html* you can access the kp.html file.

Image: Sample kp.html page

This is a sample kp.html page:



PeopleTools 8.52 - Process Scheduler [Concept]

Understanding Process Scheduler [Concept]

- Understanding PeopleSoft Process Scheduler [Try It][Concept]
- Understanding PeopleSoft Process Scheduler Architecture [Try It][Concept]
- Understanding PeopleSoft Process Scheduler Components [Try It][Concept]
- Understanding the Activities to Implement Process Scheduler [Try It][Concept]

Submitting Process Request [Concept]

- Submitting Process Requests [See It][Try It][Know It][Do It][Concept]
- Changing Distribution Options [See It][Try It][Know It][Do It][Concept]
- Submitting a Process from an Application [See It][Try It][Know It][Do It][Concept]
- Scheduling Process Requests [See It][Try It][Know It][Do It][Concept]

Using Process Monitor [Concept]

- Monitoring Processes [See It][Try It][Know It][Do It][Concept]
- Viewing Process Details [See It][Try It][Know It][Do It][Concept]
- Viewing Report Details [See It][Try It][Know It][Do It][Concept]
- Viewing PeopleSoft Process Scheduler Server Agents [See It][Try It][Know It][Do It][Concept]

KP.HTML File Topic Modes

Each UPK installation includes a kp.html file that lists your UPK modules and topics. The kp.html file, which is located in the player package folder, enables you to manually link to a specific mode for a topic.

UPK modules can appear in five modes:

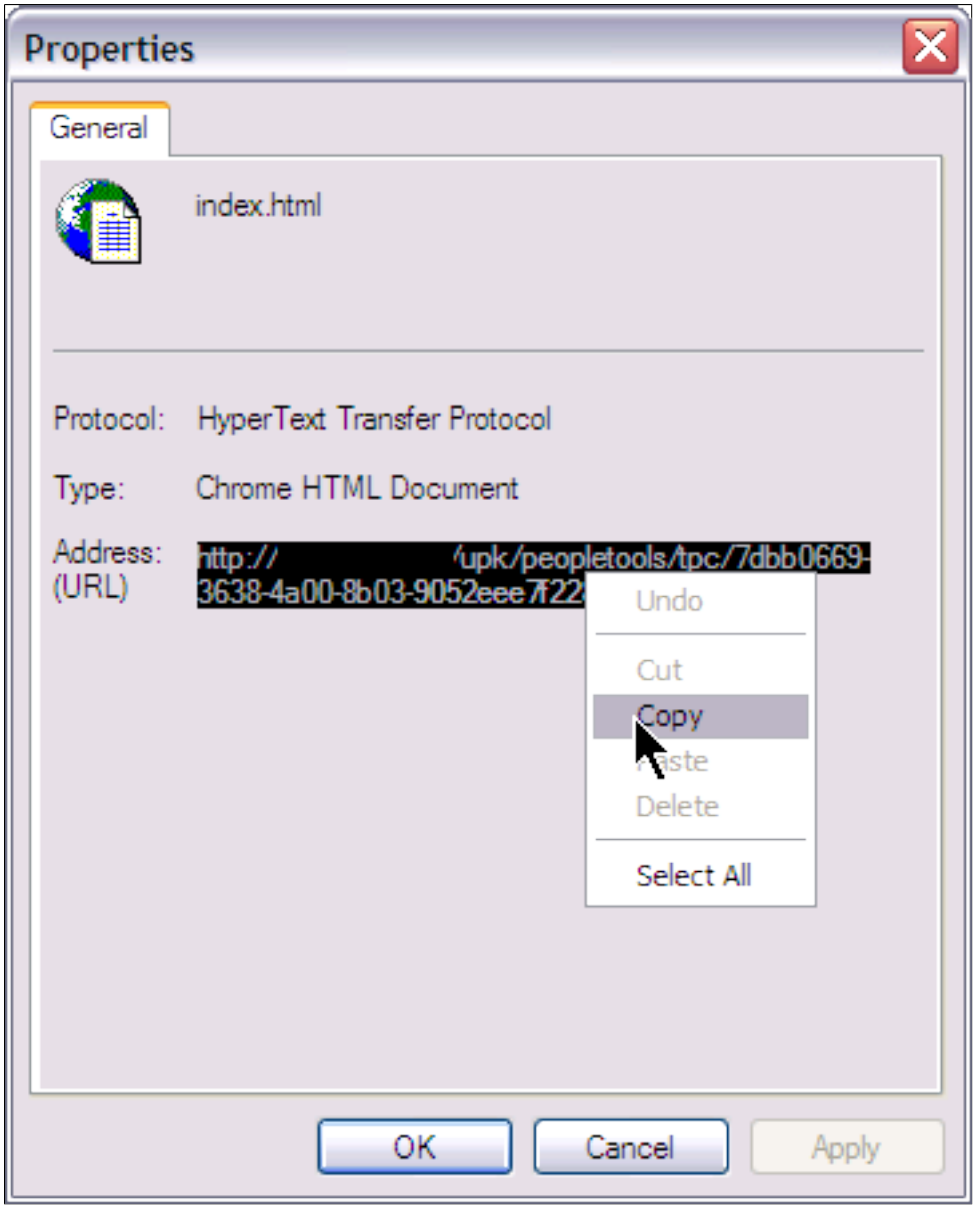
- Do It
- Know It
- Try It
- See It
- Concept

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Right-click the topic link and select Properties to access the topic-specific URL. If you are creating task-oriented related content, right-click the Do It link. If you are creating concept-oriented related content, right-click the Concept link.

Image: URL for a UPK index.html file

The Properties dialog box shows the URL in the Address field. Although the fields on the dialog box are display-only, you can highlight and copy the address as this example shows:



Note these points about the format of the URLs:

- For task-oriented information, the URL should look similar to this:

  http://upkserver/upk/peopletools/dhtml_kp.html?Guid=69f0376e-70fc-417c-a43c-ecd6afb72b29&*Mode=D*&Back

- For conceptual information, the URL should look similar to this:

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

http://upkserver/upk/peopletools/tpc/0350d9a9-6013-4cc9-9022-016f815029ba/Parts/*index.html*

Use this table to verify that the link accesses the correct mode:

Link Fragment	***Mode***
Mode=D	Do It
Mode=K	Know It
Mode=T	Try It
Mode=S	See It
index.html	Concept

For each URL, the `&Back` fragment is extraneous for related content purposes. You delete this part of the URL when you create the service definition.

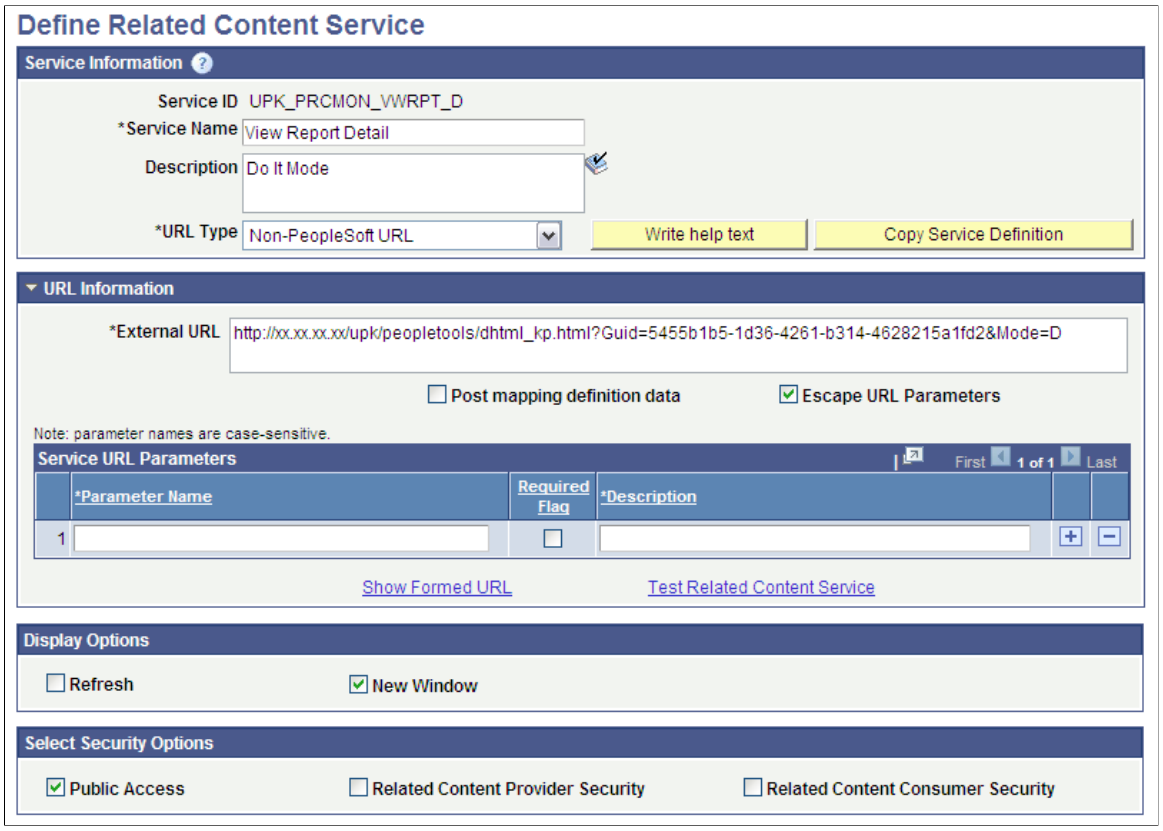
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Service Definitions

You configure UPK topics as non-PeopleSoft URL service definitions.

Image: Related content service definition for a UPK topic

The following screenshot illustrates the related content service definition for the View Report Detail UPK topic, which has been configured as a non-PeopleSoft URL type:



Related Content Menus and Frames

When you assign UPK services to content references, you assign the services as component-level related content only for these reasons:

- UPK service definitions do not require any parameters; therefore topics cannot be mapped at the field level.
- Users should be able to access UPK content from the Related Content drop-down menu. Only services assigned as related content appear in the Related Content drop-down menu.

Menu Item Labels

When you create a label for the service, consider the specific context of the service and use direct language. For example, if you want to add a service in which the UPK topic shows users how to submit a process in Do It mode, then *Show me how to submit a process* is an appropriate label.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

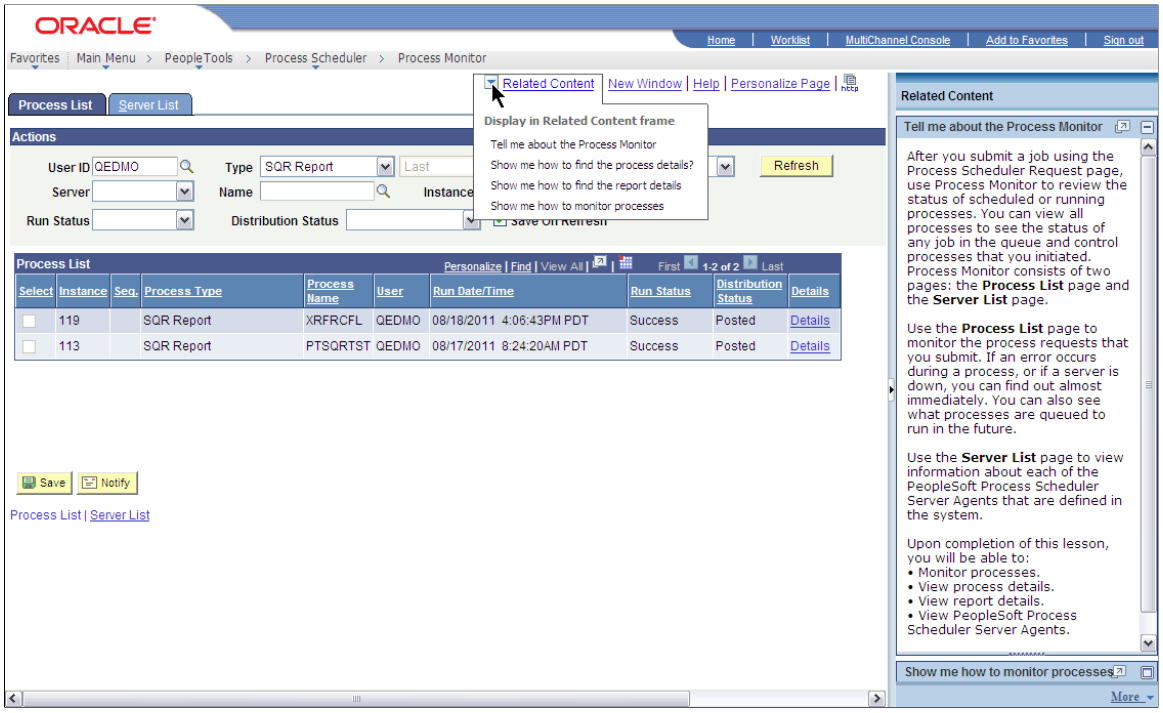
Related Content Frames

If you assign the UPK topics to appear in the right frame, users can access multiple help topics simultaneously because they appear in individual pagelets. If you assign UPK topics to the bottom frame, one topic only is visible at a time because they appear in tabs.

Example UPK Topic in Related Content Frame

Image: Example UPK topic as related content on the Process Monitor page

In this example, the UPK topic appears in the right related content frame:



Related Links

Understanding the PeopleSoft Related Content Framework
Configuring Related Content, Related Actions, and Layout
Understanding the PeopleSoft Related Content Framework

Creating UPK-Based Related Content Service Definitions

This section discusses how to:

1. Modify the UPK player package link.
2. Create the UPK-based service definition

Modifying the UPK Player Package Link

To modify the UPK player package link:

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

1. Contact your UPK system administrator to obtain the player package link (URL).
2. Change the file name *toc.html* to *kp.html* so that the URL looks similar to this:

   http://*upkservername*/upk/*content_subject*/kp.html
3. Paste the link into your browser and access the file.
4. Find the topic. Right-click the Concept or Do It link associated with the topic. If you are creating task-oriented related content, right-click the Do It link; otherwise, click the Concept link.
5. From the pop-up menu, select Copy Shortcut.

   Alternately, select Properties and then copy the URL that you see on the dialog box.
6. If you are creating multiple service definitions, you might consider creating a spreadsheet that contains information similar to that which exists in this table:

Topic	***URL***
Process Scheduler - Concept	http://*upkserver*/upk/peopletools/tpc/0350d9a9-6013-4cc9-9022-016f815029ba/Parts/index.html
Viewing Process Details - Do It	http://*upkserver*/upk/peopletools/dhtml_kp.html?Guid=69f0376e-70fc-417c-a43c-ecd6afb72b29&Mode=D&Back

7. For each task-oriented URL, remove the *&Back* at the end of the address.
8. Save your spreadsheet.

Creating the UPK-Based Service Definition

To create the UPK-based service definition:

1. Create a new related-content service definition.

   Choose a naming convention that makes UPK service definitions easily identifiable, for example *UPK_*<service_definition_name>
2. Enter the service name and description.
3. Select *Non-PeopleSoft URL* as the URL type.
4. In the External URL field, enter the URL to your kp.html file.

   It should look similar to this: *http://myupksite.com/upk/mycontent/dhtml_kp.html?Guid=5cd5ba9f-c3a8-441f-923b-a8aca7e95b6e&Mode=D*
5. Click the Test Related Content Service link.

   The Test Related Content Service page opens.
6. On the Test Related Content Service page, click the Test Related Content Service link.

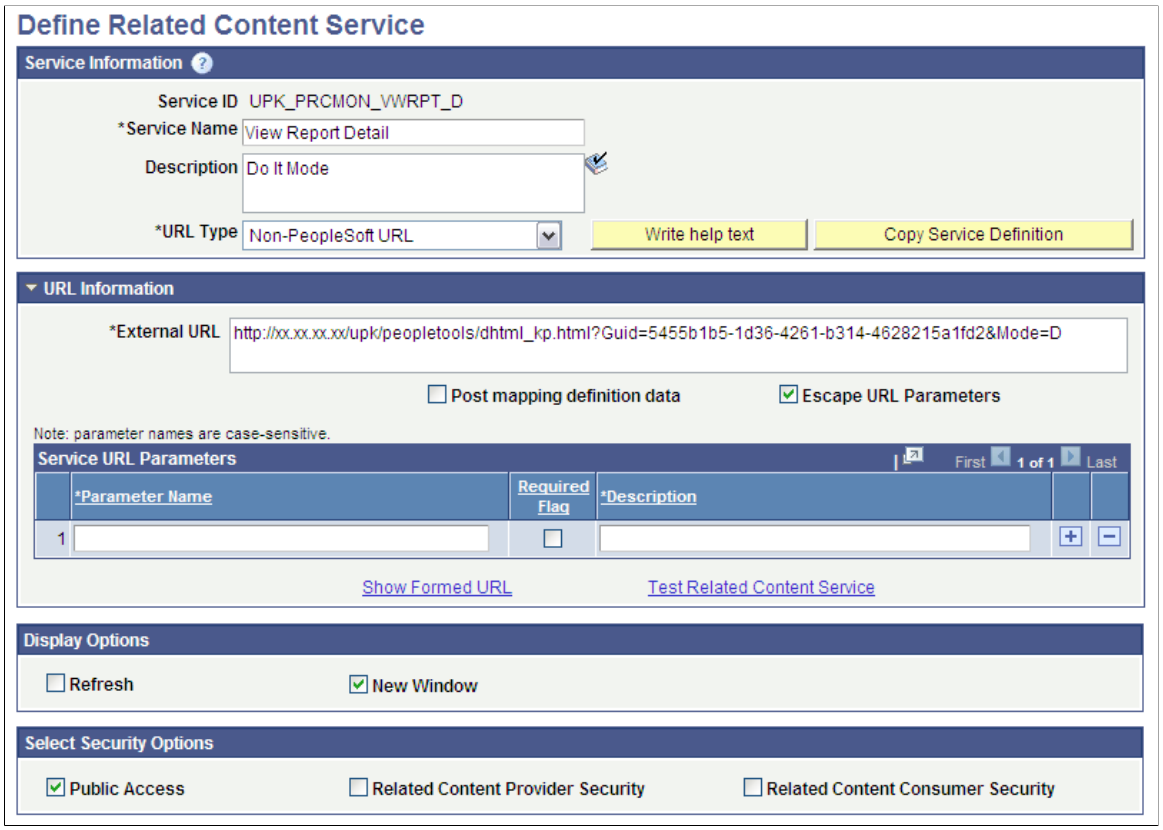
 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

7. Click the Return button to go back to the Define Related Content Service page.
8. For conceptual information, select the New Window check box to give the administrator the ability to enable the new window functionality for the user.
9. In the Select security options group box, select the Public Access check box.
10. Click the Save button.

Note: UPK-based service definitions do not require additional parameters.

Image: UPK related content service definition

The following screenshot illustrates how a completed service definition shows an external URL that accesses UPK content in Do It mode:



Note: You can use the Copy Service Definition button to clone this definition and save time creating new UPK-based service definitions.

Related Links

Creating Non-PeopleSoft URL Service Definitions

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Assigning UPK-Based Services as Related Content

Access the Manage Related Content Configuration page. (Select PeopleTools, Portal, Related Content Service, Manage Related Content Service.)

Image: Manage Related Content Configuration page

Use the Manage Related Content Configuration page to assign UPK-based services as related content:

The following screenshot illustrates an example of managing related content configuration page.

1. Click the Assign Related Content to Application Pages link.
2. Select the content reference.

   See Defining Related Content Services.
3. On the Related Content tab, add the UPK service as component-level related content.
4. Configure the related content menu.
5. Configure the related content frame.

   See Configuring Related Content, Related Actions, and Layout.
6. Click the Save button.
7. Click the content reference link to test your configuration.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix G

Configuring OBIEE-Based Related Content Services

OBIEE Configuration Prerequisites

You can configure related content services to display Oracle Business Intelligence Enterprise Edition (OBIEE) analytics reports. The OBIEE servlet must be installed and OBIEE must be configured to enable OBIEE-based related content. In addition to configuring the analytics servlet, you must complete certain prerequisites before you assign OBIEE-based related content services to application pages.

When using OBIEE analytics as related content services, note the following configuration requirements:

- The OBIEE database needs to be on release 10.1.3.3 or later.
- The PeopleSoft application database and the OBIEE database must be configured with the same users to access the reports.
- XML needs to be generated for the OBIEE report definition that you make available as related content. To automatically generate the XML, navigate to the advanced tab on the OBIEE report definition and save the report.
- The sizing of reports needs to be handled on the OBIEE database. The associated grid, width of the page, and centering of the objects cannot be adjusted.
- Set up the Integration Broker Gateway on the PeopleSoft database.
- Encrypt the *secureFileKeystorePasswd.*

  1. Select PeopleTools, Integration Broker, Configuration, Gateways and click the Gateway Setup Properties link.
  2. Enter *administrator* in the User ID field and *password* in the Password field and click the OK button.
  3. Click the Advanced Properties Page link and search for *secureFileKeystorePasswd.*
  4. Copy the password and encrypt it using the Password Encryption Utility at the bottom of the page.
  5. Replace the existing plain text password with the encrypted value.
  6. Click the OK button.
  7. Save the page and click the OK button.

- Test the deployment by entering this URL in your browser: *http://<webservername:port>/analytics*.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- After installing analytics.war, update the OBIEE server setup with the weblogic URI: http://<webservername:port>/analytics and reload the report definitions.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix H

Configuring PeopleSoft Applications to Use Oracle WebCenter Related Content Services

Understanding the Oracle WebCenter Integration

Oracle WebCenter is a collaborative portal environment that provides an infrastructure for enterprise-wide collaboration. Oracle WebCenter Spaces and Oracle WebCenter Services provide a comprehensive set of standards-based components that enrich your PeopleSoft applications with the most complete and open set of Enterprise 2.0 capabilities in the industry.

Oracle WebCenter Spaces allows users to work more effectively with project teams and work groups, including teams that span multiple geographies and include external members. Within the context of a space, Oracle WebCenter Services provide the capability to include wikis, blogs, communications, content collaboration and social networks. PeopleTools, by leveraging the PeopleSoft Related Content Framework, enables you to integrate WebCenter group spaces and many of the accompanying features with PeopleSoft transaction pages. The following WebCenter related content services are standard with your PeopleTools installation and are available for use with any PeopleSoft transaction page:

- Blogs
- Discussions
- Documents
- Links
- Wikis

Note: As of PeopleTools 8.54, integration with Oracle WebCenter through the delivered related content services has been deprecated. These service definitions may be removed from a future release of PeopleTools.

Understanding WebCenter Spaces

Within any company, large groups or departments often must create smaller groups to focus on a particular goal, project, or even topic. To help these groups organize themselves, Oracle WebCenter uses the concept of spaces as the nexus for bringing content together. Spaces provide a meaningful way to manage all project details involving any group of users. The WebCenter Spaces application creates spaces to help PeopleSoft application users share information and interact in a collaborative setting.

Templates

Oracle WebCenter Spaces application provides templates for building spaces that support different types of endeavors. When a user accesses a transaction page for the first time, WebCenter creates a new group space for the page or its transaction instance. The template specifications determine the services that

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

appear by default in the new group space. You can install and use the PeopleSoft template that your PeopleTools installation includes, you can use an out-of-the-box WebCenter template, or create and install your own custom template using an existing group space as the basis for the template.

When you configure the WebCenter options in your PeopleSoft applications, PeopleTools requires that you specify the default template for use with new PeopleSoft transaction group spaces. PeopleTools uses web services to retrieve group space template information from the WebCenter server, and then display the group space services in the bottom related content frame.

Understanding WebCenter Services

Oracle WebCenter exposes social networking and shared productivity features through a comprehensive set of services. The services are integrated so that you can use them together. For example, you can add a discussion forum to enable shared discussion, upload documents for shared review, and a blog to keep your personal notes regarding a particular PeopleSoft transaction. You can associate these Oracle WebCenter services with PeopleSoft application pages through the Manage Related Content component:

- Oracle WebCenter Blogs service
- Oracle WebCenter Discussions service
- Oracle WebCenter Documents service
- Oracle WebCenter Links service
- Oracle WebCenter Wikis service

While the related content service definitions are provided in PeopleTools, an active Oracle WebCenter database is required to provide the services and to store the related data.

Important! With the exception of the links service, each WebCenter service uses a set of corresponding WebCenter pages for managing the elements of the group space. The WebCenter application, not PeopleTools, controls the behavior of these pages.

See *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter,* Getting WebCenter Portal Applications Up and Running.

Oracle WebCenter Blogs Service

The Blogs service enables users to create instance specific or shared WebCenter blogs. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Blog posts to the group space become shared blogs of that transaction page, which means the blogs appear for all transaction instances of the transaction page. When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Blog posts of the group space appear when the user selects the specific transaction instance on the transaction page.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor privilege to the group spaces. If you specify a default moderator role on the WebCenter Options page, a moderator is also added.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance-level data security.

Oracle WebCenter Discussion Service

The Discussions service lets you expose discussion forums on your application pages, so users can create forums, post questions, and search for answers. For example, customers can share product reviews, or a customer service department can answer questions online. Discussion forums additionally provide the means to preserve and revisit discussions.

The Discussion service enables users to create instance specific or shared WebCenter discussions. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Discussions in the group space become shared discussions of that transaction page, which means the discussions appear for all transaction instances of the transaction page. When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Discussions posts of the group space appear when the user selects the specific transaction instance on the transaction page.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor privilege to the group spaces. If you specify a default moderator role on the WebCenter Options page, a moderator is also added.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance-level data security.

Oracle WebCenter Documents Service

The Documents service provides features for accessing, adding, and managing folders and files; configuring file and folder properties; and searching file and folder content in the connected content repositories. When you set up a content repository connection to integrate the Documents service, you can add content to the transaction pages of the PeopleSoft application.

The Documents service enables users to create instance specific or shared WebCenter discussions. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Documents in the group space become shared documents of that transaction page, which means the documents appear for all transaction instances of the transaction page.

When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Documents in of the group space appear when the user selects the specific transaction instance on the transaction page.

The Documents service uses corresponding WebCenter pages for actions such as, uploading new documents, listing and searching documents in the group space, and managing documents. The WebCenter application, not PeopleTools, controls the behavior of these pages.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor privilege to the group spaces. WebCenter also adds a moderator user group member, if you specify a default moderator role on the WebCenter Options page.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance-level data security.

Oracle WebCenter Links Service

The Links service provides a way to view, access, and associate related information. For example, in a list of project assignments, you can link to the specifications relevant to each assignment. In a discussion thread about a problem with a particular task, you can link to a document that provides a detailed description of how to perform that task.

Unlike other WebCenter related content services, the Links service is a PeopleSoft component-based page. The Links service enables users to search and create links to existing WebCenter group spaces. The service also enables users to add links to external web pages. When you click the links that appear on the service page, the target opens in a new browser window.

The Links service enables users to create instance-specific or shared WebCenter links to resources.

This service supports Related Content Security only. When the transaction page or instance includes other WebCenter related content services such as the WebCenter Document service or the WebCenter Discussion service, the group spaces created for those services appear as related links by default; you cannot remove these links. Conversely, the service does not alter user access to a linked service. When one user does not have access to a group space that is the target of a link added by other users, the group space link does not appear to the unauthorized user.

Note: For this service, related content security governs whether links appear. The service does not control the security of the target content.

Oracle WebCenter Wiki Service

A wiki is a collection of useful content or information that users can browse, update, and remove, sometimes without the need for registration. This ease of interaction and the variety of operations make wiki an effective tool for collaborative authoring, where multiple people create written content together.

The wiki service enables users to create instance specific or shared WebCenter wiki. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Wikis in the group space become shared wikis of that transaction page, which means the wiki page appear for all transaction instances of the transaction page.

When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Wikis in the group space appear when the user selects the specific transaction instance on the transaction page.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

privilege to the group spaces. WebCenter also adds a moderator user group member, if you specify a default moderator role on the WebCenter Options page.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance level data security.

Understanding Security Synchronization

To function correctly, the permissions of the group space must be the same as the associated content reference object of the transaction page. Your PeopleSoft applications use the WebCenter *SpaceWebService* to create and maintain group spaces security that is synchronized to PeopleSoft security. You must configure the web service security for the WEBCENTER node before using any of the WebCenter related content services. Currently, SAML token web service security is the only security type that the WebCenter application supports to consume WebCenter web services.

When a WebCenter service creates a groups space for a transaction page or its transaction instances, all user roles with access to the content reference object are added as user group members with the contributor privilege. WebCenter supports transaction data security at the role level only. Any user who has access to a transaction page is granted access to all group spaces that are created for that page or its transaction instances. If permissions change for a transaction page, the same changes must be reflected in the WebCenter permissions. You synchronize security by running the WebCenter Security batch process.

WebCenter Security Batch Process

The WebCenter Security sync batch process maintains security synchronization between PeopleSoft transactions and WebCenter services.

Important! You should run this process when the permissions of any transaction page that uses one of the WebCenter related content services changes.

The WebCenter Security synchronization process:

1. Retrieves a list of expected user group members of the group space.

   This list consists of the roles that have access based on the permission lists that appear on the Security tab of the content reference object for the transaction page.

2. Adds the user roles that do not exist in the list.

3. Adds a moderator user group member, if you specify a default moderator role on the WebCenter Options page.

4. Removes user group members that do not appear in the list.

5. Removes individual user members that do not have one of the listed user roles.

Note these points about the security synchronization process:

- The batch process should be run only once after a transaction page permissions are changed to update permissions of existing group spaces.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

- The batch process updates only those group spaces that the WebCenter related content services created.
- The batch process does *not* change the permissions of group spaces that are linked to a transaction page or its transaction instances by the WebCenter Links service.

Note: Links do not appear to users if permission to access the associated target is not granted.

Related Language Considerations

Note these points about the use of related language in the WebCenter integration:

- WebCenter and PeopleSoft applications support related language data differently.
- A related language item that you translate into the nonbase language appears to users of all languages.
- You cannot configure a translated item to appear to base language users only, instead of the base language.
- Both the base language and the nonbase item can coexist in the same system.

Important! WebCenter currently does not support Canadian French or UK English as a local language option.

Example

If you have an application that uses English as the base language and French as a related language:

- If a French-speaking user enters a related language item, then it appears to users of all languages.
- An English-speaking user can not provide an English translation of that item to show to English users only, instead of the French version.
- Both English and French items can appear simultaneously in the same system.

Prerequisites

These tables discuss the prerequisite steps to configuring Oracle WebCenter options in the PeopleSoft application.

Perform these tasks in the WebCenter application.

Prerequisite	***Reference***
Install Oracle WebCenter.	See Oracle® Fusion Middleware Installation Guide for Oracle WebCenter 11g Release 1 (11.1.1.5.0).
Import the PeopleSoft template into the WebCenter application.	See *Oracle® Fusion Middleware User's Guide for Oracle WebCenter Spaces 11g Release 1 (11.1.1.5.0),* Importing Space Templates.
Enable locale switching.	See Registering Provider Classes.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Prerequisite	*Reference*
Set the frame busting context parameter to *never*. **Note:** You set the parameter to *never* because the WebCenter pages appear as related content in the lower Related Content frame, which is an iframe.	See *Oracle® Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework 11g Release 1 (11.1.1.5.0),* Framebusting.
Modify the configuration of the PeopleSoft template to fit your business needs. (Optional)	See *Oracle® Fusion Middleware User's Guide for Oracle WebCenter Spaces 11g Release 1 (11.1.1.5.0),* Working with Space Templates.
Create new templates. (Optional) **Important!** When you create your own template, you must enable the Discussions and Documents services and include the default Wiki and Blog pages in your template. WebCenter provides no process or program to synchronize any template modifications; if you make more changes to the default template after your initial modifications, then you must recreate the custom template from the delivered template.	See *Oracle® Fusion Middleware User's Guide for Oracle WebCenter Spaces 11g Release 1 (11.1.1.5.0),* Working with Space Templates.

Perform these tasks in the PeopleSoft application.

Prerequisite	*Reference*
Configure the Integration Gateway.	See "Administering Integration Gateways" (PeopleTools 8.54: Integration Broker Administration).
Configure single sign on.	See "Implementing PeopleSoft-Only Single Signon" (PeopleTools 8.54: Security Administration).
Access the WEBCENTER node and enter the WebCenter server address in the Content URI Text field.	See Setting Portal Node Characteristics.
Access the WEBCENTER node and configure WS Security so that the node can consume the WebCenter web services. **Note:** Currently, only *SAML Token* is supported by WebCenter as the authentication token type.	See Understanding the PeopleSoft WS-Security for WSRP.

Related Links

"Understanding the Integration Network" (PeopleTools 8.54: Integration Broker Administration)
"Understanding Single Signon" (PeopleTools 8.54: Security Administration)

Configuring Oracle WebCenter Options

This section discusses how to configure WebCenter options:

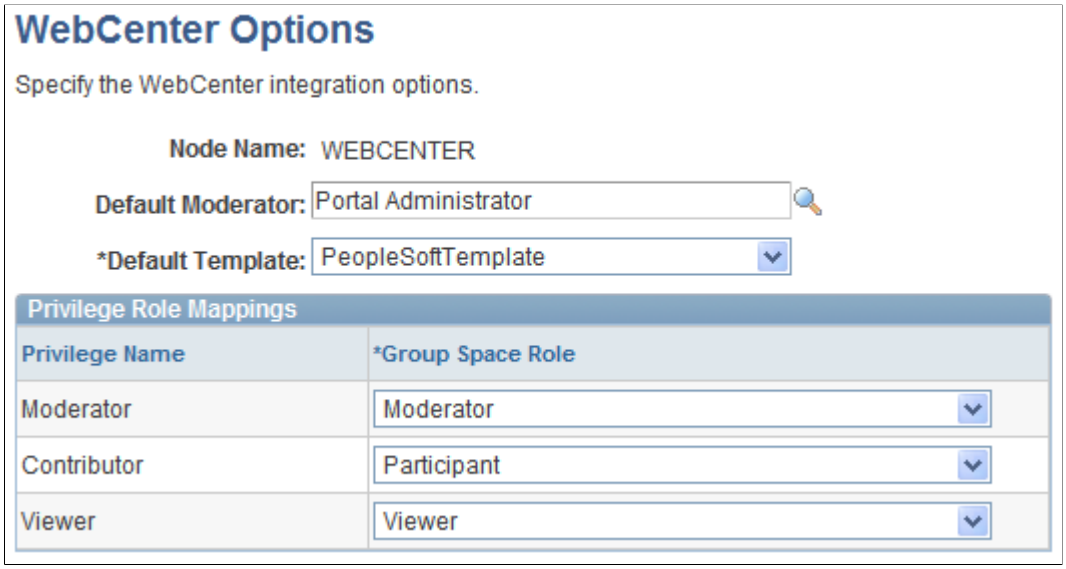
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Setting the WebCenter Integration Options

Access the WebCenter Options page (Select PeopleTools, Portal, WebCenter Integration, WebCenter Options).

Image: WebCenter Options page

The following screenshot illustrates about setting the WebCenter integration options on the WebCenter Options page.



WebCenter Options

Node Name Displays the name for the WebCenter application default node, which is *WEBCENTER.*

Default Moderator Specify the user role to be added as a user group member with the moderator privilege to any group spaces created by WebCenter related content services.

Default Template Select the template that the WebCenter related content service uses to create new group spaces for PeopleSoft transaction pages. All public group space templates that you have set up in WebCenter appear in the drop-down list. The default value is *PeopleSoftTemplate.* This field requires a value.

Privilege Role Mappings

Use the fields in the Privilege Role Mappings grid to correlate the PeopleSoft privilege to the WebCenter group space role. This grid appears when you select a valid default group space template. All fields in this grid are required when you select a default group space template.

Privilege Name Displays the PeopleSoft privilege.

Group Space Role Select from the list of privilege roles that exist in the specified default template. If you select a new default template, the values in drop-down list change to reflect the new default.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Select from these default values:

- *Moderator*
- *Participant*
- *Viewer*

This field requires a value.

Synchronizing Oracle WebCenter and PeopleSoft Application Security

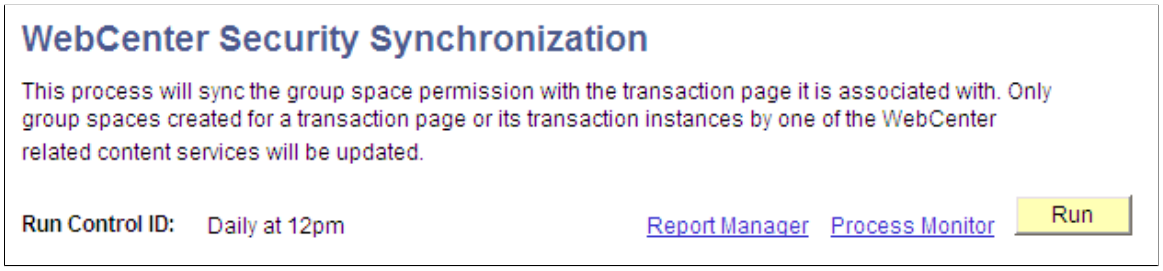
This section discusses how to run the WebCenter Security Sync batch process.

Running the WebCenter Security Sync Batch Process

Access the WebCenter Security Synchronization page. (Select PeopleTools, Portal, WebCenter Integration, WebCenter Security Sync).

Image: WebCenter Security Synchronization page

This example illustrates the fields and controls on the WebCenter Security Synchronization page.



To run the WebCenter Security Synchronization batch process:

1. Click the Run button.
2. Click the Process Monitor link to verify success.

Managing WebCenter Links

This section describes how to:

- Manage instance-specific links.
- Manage shared links.
- Add group space links.
- Add external URL links.

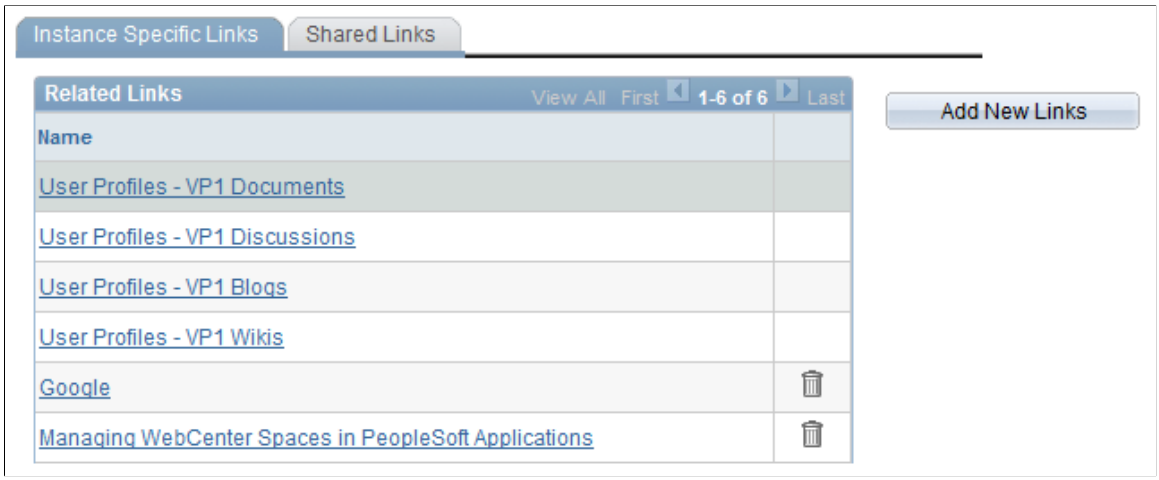
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Managing Instance-Specific Links

Access the Instance Specific Links page (Select the Instance Specific Links tab).

Image: Instance Specific Links page

The following screenshot illustrates about managing instance-specific links on the Instance Specific Links page.



To add new instance-specific links, click the Add New Links button. The Add New Links page appears.

See Adding Group Space Links, Adding External URL Links.

To delete instance-specific links, click the Delete Link(trash can) icon.

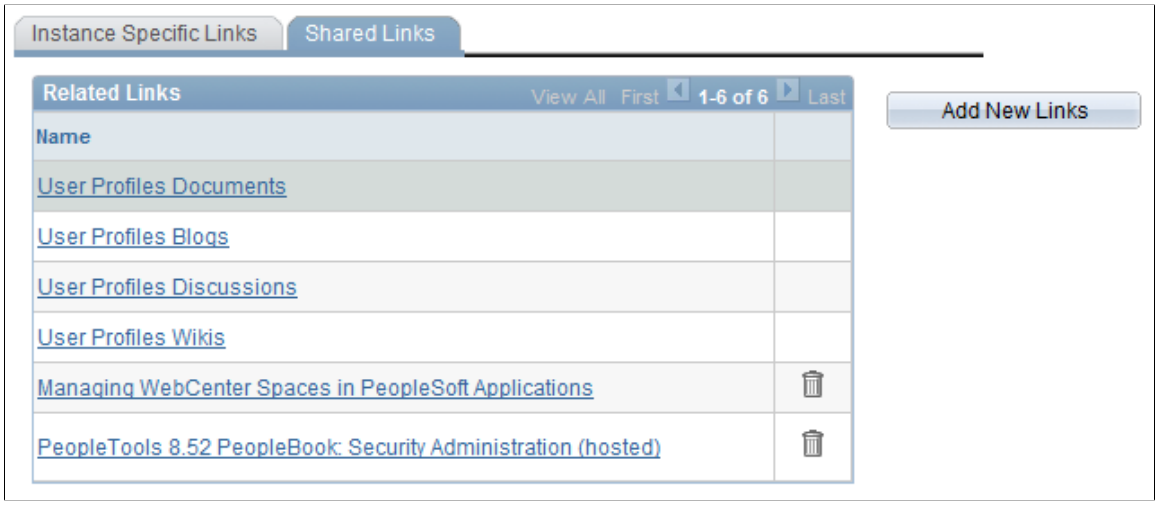
Note: Links to services within the current group space appear automatically and cannot be removed.

Managing Shared Links

Access the Shared Links page (Select the Shared Links tab).

Image: Shared Links page

The following screenshot illustrates about managing shared links on the Shared Links page.



 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

To add new shared links, click the Add New Links button. The Add New Links page appears.

See Adding Group Space Links, Adding External URL Links.

To delete shared links, click the Delete Link(trash can) icon.

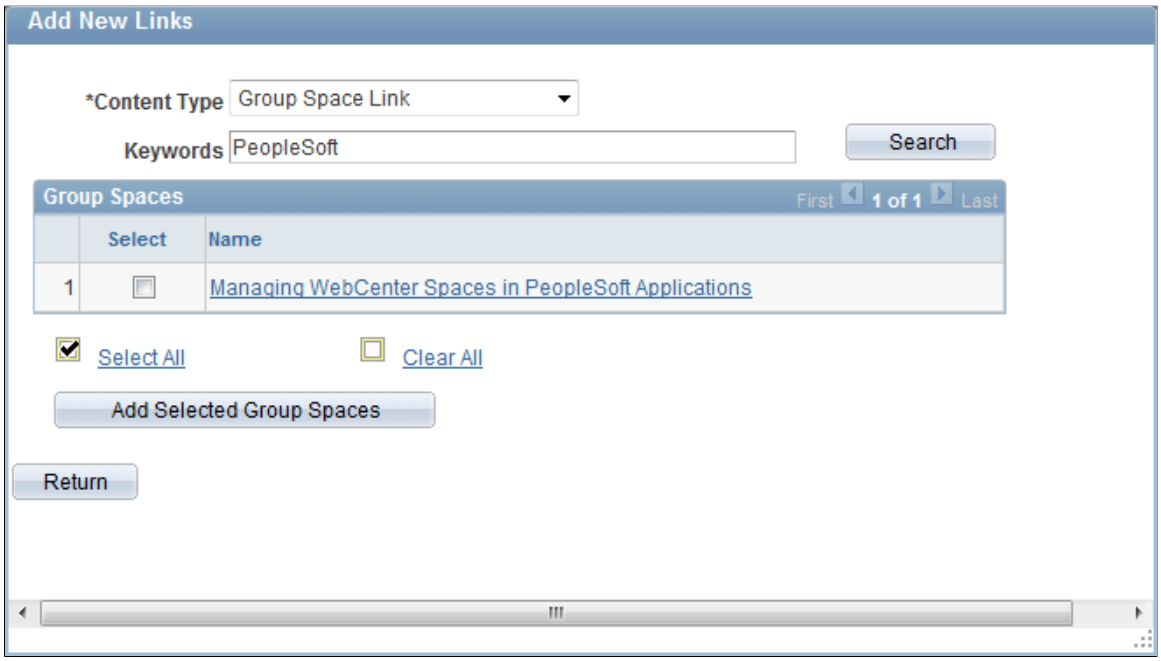
Note: Links to services within the current group space appear automatically and cannot be removed.

Adding Group Space Links

Access the Add New Links page (Click the Add New Links button).

Image: Add New Links page

The following screenshot illustrates adding group space links on the Add New Links page.



To add new group space links:

1. Select *Group Space Link* as the content type.
2. To search the group spaces that are available to you in your WebCenter application, enter keywords and click the Search button.

   Note: Keywords for each group space are managed by the WebCenter administrator in the WebCenter application

3. Select the group spaces for which you want to add links.
4. Click the Add Selected Group Spaces button.

   The application returns you to the WebCenter Links page.

Click the Return button to go back to the previous page.

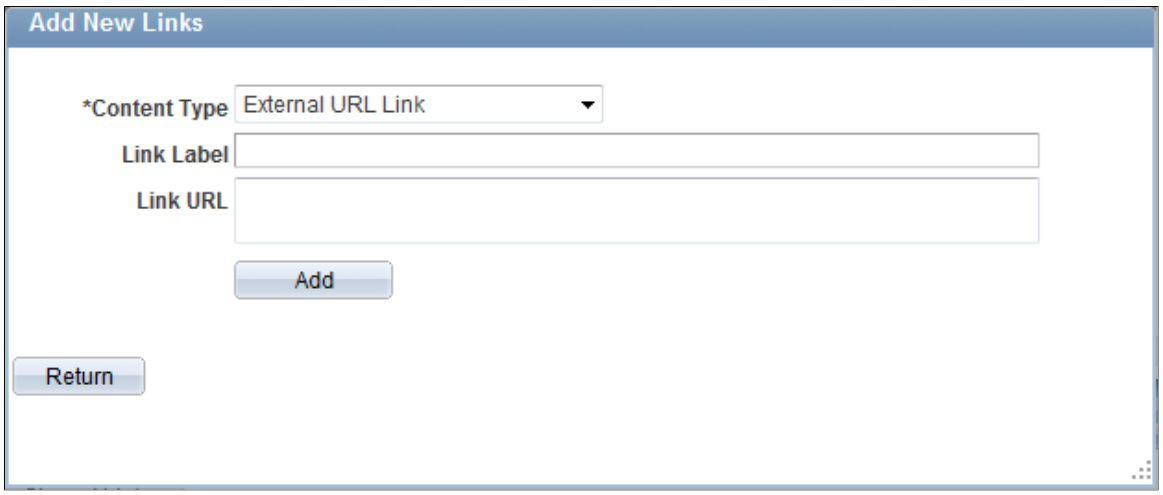
Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Adding External URL Links

Access the Add New Links page (Click the Add New Links button).

Image: Add New Links page

The following screenshot illustrates adding external URL links on the Add New Links page.



To add new external URL links:

1. Select *External URL Link* as the content type.
2. Enter the label that you want to appear in the Related Links grid on the WebCenter Links page.
3. Enter the target address of the external link in the Link URL field. Use this example as a model: *https://support.oracle.com*.
4. Click the Add button.

   The application returns you to the WebCenter Links page.

Click the Return button to go back to the previous page without saving your entry.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix I

Enabling Locale Switching Through Query Parameters in Oracle WebCenter

Registering Provider Classes

Note: As of PeopleTools 8.54, integration with Oracle WebCenter through the delivered related content services has been deprecated. These service definitions may be removed from a future release of PeopleTools.

To support locale switching by using the WCGS_PreferredLanguage=*LangCode* query parameter, you must manually register the URLBaseProvider class.

You can do that manually or by using a WebLogic Scripting Tool (WLST) command.

Manual Registration

To register the provider class manually:

1. Using a text editor, open the adf-config.xml file.
2. Find the adf-config element and enter the following xml inside the tag:

```
 <adf-webcenter-generalsettings-config xmlns="http://xmlns.oracle.com/webcente⇒
r/generalsettings/config">
   <provider-class>
     oracle.webcenter.generalsettings.model.provider.URLBaseProviderprovider
   </provider-class>
 </adf-webcenter-generalsettings-config>
```

3. Save the file.

WLST Registration

To use WLST to register the provider class:

1. Open the WebLogic Scripting Tool.
2. Run this command:

```
setWebCenterGeneralSettingsProvider('webcenter','oracle.webcenter.generalsetti⇒
ngs.model.provider.URLBaseProvider')
```

Note: The first parameter is the application name. The second parameter is the providerClass. Sometimes you must supply a third parameter.

or this command:

```
setWebCenterGeneralSettingsProvider('webcenter','oracle.webcenter.generalsetti⇒
```

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

```
ngs.model.provider.URLBaseProvider','WLS_Spaces1')
```

Note: The third parameter is the spaces cluster service name or spaces service name.

3. Restart the server.

See Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework 11g Release 1 (11.1.1.5.0), "Getting Started Using the Oracle WebLogic Scripting Tool."

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix J

Configuring OBIEE Admin User Permissions in Enterprise Manager 11g

Configuring OBIEE Server Admin User Permissions in Enterprise Manager 11g

This section discusses the necessary permission configuration for using OBIEE.

Note: For use with 11g only.

The user that you configure in the Server Setup page must have *impersonate* permissions in Enterprise Manager. In release 10g, the BI Administrator satisfies this requirement. In release 11g, the user must be assigned the BI System application role.

To enable the BI System application role in Enterprise Manager 11g:

1. Sign in to Webserver Enterprise Manager as an administrator.
2. In the left side navigation, click Business Intelligence, coreapplication.
3. When the new page appears, click Configure and manage Application Roles.
4. When the new page appears, click BI System.
5. In the users column, click Add User and add the same user that you entered as the Admin User on the Server Setup page.
6. Click the OK button to save your changes.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix K

Language Support for Consuming and Producing Remote Portlets

Language Support for Consuming Remote Portlets

Note: The ServiceDescription provides translations for the following strings: title, shortTitle, description, displayName, and keywords. The main language value is placed inside the <offeredPortlets> tag, other language strings are inside <resources> tags at the end of xml file. PeopleSoft applications store the mapping of three-character PeopleSoft language codes such as *ENG, FRA, CFR,* and *DUT,* as well as ISO locales such as *en, en-US, fr, fr-ca,* and *nl*.

When we consume remote portlets, we request service description XML from the remote producer. We store the definition of the remote producer in the PeopleSoft database. We then parse service description XML and store the definitions of the remote portlets. If multilanguage data is in the service description, we map ISO locales to PeopleSoft language codes and store the values in our database along with the correct PeopleSoft language codes.

Note: We are interested only in such locales that correspond to installed PeopleSoft language codes. All other locales are ignored.

When parsing the remote ServiceDescription, we look for matches between ISO locales and installed PeopleSoft language codes. We accept the string values when matches are found. Here are the rules:

- Accept if an exact match is found, for example, if *fr-ca* (CFR) is installed and *fr-ca* is provided in ServiceDescription.
- If the child is installed and they provide only the parent, we accept parent for child. For example, if *fr* (FRA) and *fr-ca* (CFR) are installed and only *fr* is provided, it is matched to both *fr* and *fr-ca.*
- If the parent locale is installed, but only children are provided, then accept the child for the parent. For example, if *fr (FRA)* is installed and *fr-ca* (CFR) is provided, then accept it for FRA.

Language Support Rules for Producing Remote Portlets

Note: The ServiceDescription provides translations for the following strings: title, shortTitle, description, displayName, and keywords. The main language value is placed inside the <offeredPortlets> tag, other language strings are inside <resources> tags at the end of xml file. PeopleSoft applications store the mapping of three-character PeopleSoft language codes such as *ENG, FRA, CFR,* and *DUT,* as well as ISO locales such as *en, en-US, fr, fr-ca,* and *nl*.

When we produce remote portlets, we look at the definitions of the appropriate content references and generate the ServiceDescripton. If any multilanguage strings (such as cref label, description, and

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

keywords) exist, we may have to include them into ServiceDescripton. The remote producer sends us the list of desired locales he is interested in and in response, we determine whether these desired locales have matching installed PeopleSoft language codes.

When generating the ServiceDescription, the PeopleSoft system tries to find matches for the desired locales to PeopleSoft installed languages using the following rules:

- If the locale is installed, the system returns it.
- If only the parent is installed, the system returns the parent. For example, if the desired locale is *fr-ca* and only *fr* is found, the system returns *fr*.
- If no match is found, the system returns *en* (English).

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix L

Overriding the Default Breadcrumb Behavior

Understanding the Default Breadcrumb Behavior

Breadcrumbs refer to the collective sequence of items that appear above the content area when you navigate to transaction pages or navigation pages. The breadcrumbs display the path you took to get to where you are—that is, the page you are on.

The PeopleSoft application dynamically generates the individual items in the breadcrumb list using the following criteria:

- The first item in the sequence is always the Main Menu.
- The intermediate items in the sequence are the names of each item that you click to access the resource; menu items and component names appear in the order that you click them.
- The last item in the sequence is:
  - The component label when you access a transaction page.
  - The folder label when you access a navigation page.
  - The WorkCenter or dashboard label when you access the WorkCenter or dashboard.
  - The value of the focus node when you access a SmartNavigation page.

Breadcrumbs can reveal both the navigational path to your current location and the hierarchical or parent-child relationships that exist among the items in the breadcrumb sequence.

The following example illustrates how breadcrumbs are constructed for a transaction page, and how those breadcrumbs are updated when you navigate to additional pages:

1. Access the User Profile page (select PeopleTools, Security, User Profiles, User Profiles).

   Image: Breadcrumb example on the User Profiles component

   The breadcrumbs display Main Menu > PeopleTools > Security > User Profiles > User Profiles as shown in this example:

2. Select a user profile—for example, PSADMIN.
3. Go to the Roles page.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Note: No breadcrumb update occurs because the Roles page is within the User Profiles component.

4. Click the View Definition link for a specific role—for example, for PeopleSoft User.

   Image: Breadcrumb example after navigating to the Roles component

   The breadcrumbs are updated to append the new component at the end of the list. The breadcrumbs display Main Menu > PeopleTools > Security > User Profiles > User Profiles > Roles as shown in this example:

   Main Menu > PeopleTools > Security > User Profiles > User Profiles > Roles

5. Go to the Permission Lists page.

   Note: No breadcrumb update occurs because the Permission Lists page is within the Roles component.

6. Click the View Definition link for a specific permission list—for example, for PTPT1000.

   Image: Breadcrumb example after navigating to the Permission Lists component

   The breadcrumbs are updated to append the new component at the end of the list. The breadcrumbs display Main Menu > PeopleTools > Security > User Profiles > User Profiles > Roles > Permission Lists as shown in this example:

   Main Menu > PeopleTools > Security > User Profiles > User Profiles > Roles > Permission Lists

7. Click Roles in the breadcrumbs.

   The breadcrumbs are updated to remove Permission Lists and display the role definition for the selected role.

See "Navigating Breadcrumbs" (PeopleTools 8.54: Applications User's Guide) for more information on working with breadcrumbs. This topic includes a list of limitations on when the breadcrumbs are not updated.

However, there might be situations in which you do not want the default breadcrumb behavior—that is, to append the next item to the breadcrumb list. In these situations, you can use either of two methodologies to modify the default, PeopleTools breadcrumb behavior:

- Use the CreateBreadcrumb built-in function to append a breadcrumb element containing application-specific data to the user's list of breadcrumbs. This function can be included in any PeopleCode event so an application-specific breadcrumb can be added whenever it is needed. See "CreateBreadcrumb" (PeopleTools 8.54: PeopleCode Language Reference) in *PeopleTools: PeopleCode Language Reference* for more information.
- Add content reference attributes to override the default breadcrumb behavior whenever the content reference is the target—that is, whenever it is navigated to. SeeOverriding Default Breadcrumbs Using Content Reference Attributes for more information.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Overriding Default Breadcrumbs Using Content Reference Attributes

You can use optional content reference attributes to override the default breadcrumb behavior for a target content reference when you want to do any of the following:

- Specify a different label as the text to either append to the breadcrumb list or to replace the last item in the list rather than using the label from the target component.

  Note: The default label is the label for the content reference.

- Skip the breadcrumb update—that is, the breadcrumbs remain unchanged.
- Replace the last item in the breadcrumb list rather than append to the list.
- Preserve the existing breadcrumbs *and* append the WorkCenter page that is the target to the existing list. (Typically, a transfer to a WorkCenter page would regenerate the entire breadcrumb list starting from the Main Menu.)
- Append to the breadcrumb list even if the target is a hidden content reference, in a hidden folder, or when the component is defined such that Display in Navigation is disabled.

Note: Alternatively, you can use the CreateBreadcrumb built-in function to append a breadcrumb element containing application-specific data to the user's list of breadcrumbs. This function can be included in any PeopleCode event so an application-specific breadcrumb can be added whenever it is needed. See "CreateBreadcrumb" (PeopleTools 8.54: PeopleCode Language Reference) in *PeopleTools: PeopleCode Language Reference* for more information.

Define the content reference attributes on the content reference that is being navigated to. You can specify one or both of the following attributes:

- Use the BCUPDATELABEL attribute to specify a message set and number that defines the label to be displayed in the breadcrumbs.
- Use the BCUPDATETYPE attribute to specify a numeric value that indicates:
  - **0** – Do not update the breadcrumbs.
  - **1** – Replace the last item in the breadcrumbs with the new element.
  - **2** – Preserve the existing breadcrumbs and append the WorkCenter page that is the target to the existing list.
  - **3** – Append hidden items to the breadcrumb list.

To modify the breadcrumb behavior for a content reference:

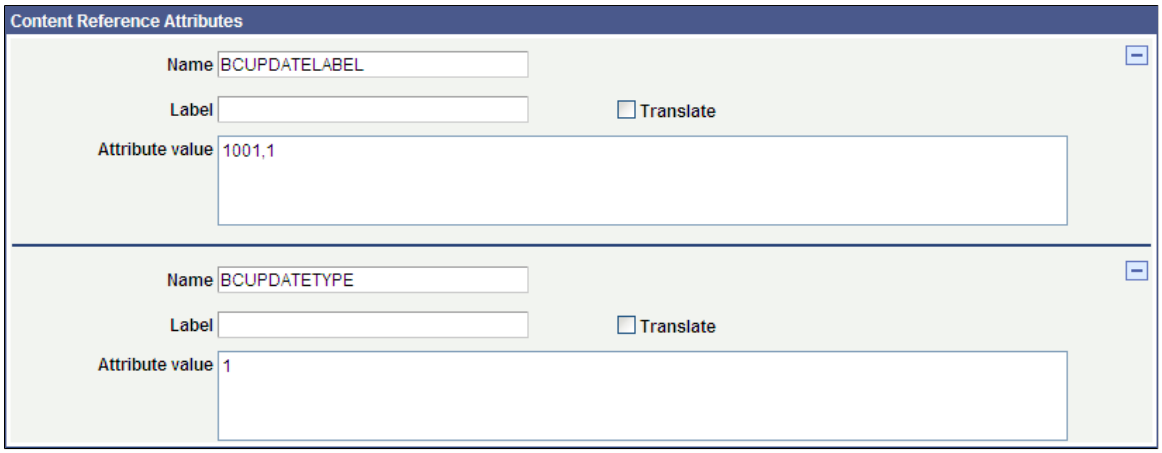
1. Go to PeopleTools, Portal, Structure and Content.
2. Navigate the menu folder hierarchy to the content reference that you want to modify.
3. Click the Edit link for the content reference.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

4. In the Content Reference Attributes group box, add an attribute for each aspect that you want to specify (BCUPDATELABEL, BCUPDATETYPE, or both).
5. For each attribute, deselect the Translate option.
6. Save the content reference definition.

Image: BCUPDATELABEL and BCUPDATETYPE content reference attributes

The following example shows both the BCUPDATELABEL attribute and the BCUPDATETYPE attribute defined for a content reference:



Related Links

Defining Content References

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Appendix M

PeopleTools Web Libraries

Understanding Web Libraries

A web library is a derived or work record the name of which starts with WEBLIB_. All PeopleSoft iScripts are embedded in records of this type. An iScript is a specialized PeopleCode function that generates dynamic web content. Administrators should make sure that users have the proper access to web libraries.

For example, the default navigation system for PeopleSoft Pure Internet Architecture users is implemented by means of a web library. If users do not have the proper authorization to the web library and its associated scripts, then they won't have proper access to the system. If users are not authorized for a particular web library or script, then they can't invoke it. After you add a web library, you set the access for each script function individually. Invoking an iScript requires the assembly of a URL. Developers assemble the URL by using PeopleCode.

PeopleTools Web Libraries

This table lists the primary PeopleTools web libraries:

Web Library Name	***Description***
WEBLIB_CTI	Contains iScripts for generating the Computer Telephony Integration (CTI) console and the CTI free-seat extension page.
WEBLIB_IB	Contains iScripts for define PeopleSoft Integration Broker functions.
WEBLIB_MCF	Contains iScripts for generating the Multichannel Framework (MCF) console, Agent to Customer chat window, Customer to Agent chat window, Agent to Agent chat window, and MCF tracer window.
WEBLIB_MSGWSDL	This is the record behind the page for the enterprise integration point (EIP) Web Services Description Language (WSDL) generation.
WEBLIB_OPT	Contains iScripts for generating Optimization Framework functions.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Web Library Name	*Description*
WEBLIB_PORTAL	Contains the following six fields with FieldFormula iScript PeopleCode, each of which relates to a functional area of the portal: • PORTAL_DYN_TEMP: Support for the dynamic template. • PORTAL_HEADER: Support for the header portion of the page and some generic routines. • PORTAL_HOMEPAGE: Support for homepage runtime interaction, including the homepage version of the menu navigation. • PORTAL_NAV: Main support routines for side navigation. • PORTAL_PGLT_PREV: Support for the pagelet preview functionality. • PT_WSRP: Used to sign in to the PeopleSoft application when it is defined as a producer.
WEBLIB_PPM	Contains iScripts for generating Performance Monitor functions.
WEBLIB_PTAF WEBLIB_PTAFEMC	Contains iScripts for generating Approval functions.
WEBLIB_PTCBD	Contains iScripts for displaying common Dashboard components.
WEBLIB_PTCXM	Contains iScripts for displaying Context Manager pagelets.
WEBLIB_PTDIAG	Contains iScripts for displaying PeopleTools Diagnostics pages.
WEBLIB_PTIFRAME	Contains iScripts for portal favorites and popup functionality.
WEBLIB_PTOBIEE	Contains iScripts for displaying Business Intelligence (Analytics) pagelets.
WEBLIB_PTPP_SC	Contains iScripts for generating navigation pages and the Main Menu navigation pagelet.
WELIB_PTPP_PGT	Renders navigation pagelets without Pagelet Wizard.
WEBLIB_PTPPB	Contains iScripts for displaying Pagelet Wizard pagelets as homepage pagelets and template pagelets
WEBLIB_PTPPB1	Contains an iScript for displaying Pagelet Wizard pagelets as WSRP portlets.
WEBLIB_PTRC	Contains iScripts for displaying related content.
WEBLIB_PTRTE	Contains iScripts for uploading inline images in rich text.

 Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

Web Library Name	***Description***
WEBLIB_PT_NAV	Contains iScripts for the menu pagelet, drop-down navigation, and left-hand navigation for transaction pages.
WEBLIB_PT_TPL	Contains iScripts for portal template processing.
WEBLIB_QUERY	Contains the iScripts for generating a URL for running queries.
WEBLIB_RPT	Contains iScript for the Run report to window output option. Supports access to the new browser window.
WEBLIB_SDK	Contains iScripts for generating software development kit functions.
WEBLIB_SDK_BI	Contains iScripts for generating sample code showing a call to the PeopleSoft Business Interlinks PSHttpEnable function to return information from a remote site that is formatted into the HTML display.
WEBLIB_SOAPTOCI	This is the entry point for ExcelToCI and the WSDL Discovery.
WEBLIB_TIMEOUT	Contains iScripts for generating an inactivity timeout warning script.
WEBLIB_XMLLINK	Contains iScripts for generating PeopleSoft Business Interlinks XML functions.

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.