



PeopleTools 8.12 Portal Technology PeopleBook

PeopleTools 8.12 Portal Technology PeopleBook

SKU MTPOr8SP1B 1200

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.

Copyright © 2001 by PeopleSoft, Inc. All rights reserved.

Printed in the United States of America.

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

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

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

PeopleSoft, the PeopleSoft logo, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, Vantive, and Vantive Enterprise are registered trademarks, and *PeopleTalk* and "People power the internet." are trademarks of PeopleSoft, Inc. All other company and product names may be trademarks of their respective owners.

Contents

About This PeopleBook

Audience	ix
Before You Begin.....	x
Related Documentation	x
Documentation on the Internet.....	x
Documentation on CD-ROM	xi
Hardcopy Documentation	xi
Typographical Conventions and Visual Cues.....	xi
Comments and Suggestions.....	xiii

Chapter 1

Introducing the PeopleSoft Portal

General Characteristics of Portals.....	1-1
The PeopleSoft Portal	1-2
Portal Architecture	1-2
Portal Templates and Template Pagelets	1-3
Personal Homepages and Pagelets	1-3
Portal Registry.....	1-4
Portal Servlet.....	1-5
Navigation.....	1-5
Search Functionality.....	1-7
Portal Architecture—Putting It All Together.....	1-7
Types of Portals Available.....	1-8
PeopleSoft Portal.....	1-9
Single Signon	1-10
How does PIA Navigation relate to Portal Navigation?.....	1-11
Portal Terminology	1-12

Chapter 2

Understanding the Portal Registry

Portal Registry Architecture	2-1
Portal Registry Properties.....	2-3
Folders	2-3

Folder Properties	2-3
Content References	2-4
Content Reference Properties	2-4
Content Providers	2-5
Content Provider Properties	2-5
Attributes	2-5
Security	2-5
Maintaining the Portal Registry	2-6

Chapter 3

Understanding the Portal Servlet

Setting Up the Portal Servlet	3-1
The Page Assembly Process for Page-Based Templates	3-2
Proxying	3-3
Converting Relative URLs to Absolute URLs	3-3
Old Tag (Relative)	3-4
New Tag (Absolute)	3-4
Converting Anchor Tags	3-4
Old Anchor Tag	3-4
New Anchor Tag	3-4
Converting Form Tags	3-4
Old Form Tag	3-4
New Form Tag	3-4
Converting JavaScript	3-5

Chapter 4

Designing Portal Templates

Template Types	4-1
Page-Based and Frame-Based Templates	4-1
Static and Dynamic Templates	4-2
Template Examples	4-2
Example of a Static, Page-Based Template	4-2
Example of a Static, Frame-Based Template	4-3
Example of a Dynamic, Frame-Based Template	4-4
Developing Portal Templates	4-9
Template Pagelets Based on Pages	4-9
Inheritance of Style Classes	4-10
Template Pagelet Tags	4-10
IClientComponent	4-10
Source	4-10

TargetContent.....	4-11
Considerations for Non-PeopleSoft Content Providers.....	4-11
Considerations for Frame-Based Templates	4-11

Chapter 5

Developing Pagelets

Designing Pagelets Based on PIA Pages	5-1
Narrow and Wide Pagelets	5-2
Narrow Pagelet.....	5-3
Wide Pagelet	5-3
Designing Pagelets Based on iScripts.....	5-4
Developing a Pagelet Based on an iScript.....	5-4
Storing iScripts.....	5-10
Style Sheets and Styles.....	5-10
Pagelet Design Considerations	5-11
General Design Considerations	5-11

Chapter 6

Using Portal Administration Features

Managing Portals	6-1
Managing Folders and Content References	6-3
Managing Folders.....	6-3
Managing Content References	6-7
Moving Folders and Content References	6-13
Managing General Portal Settings	6-16
Managing Content Providers.....	6-17
Importing a Menu Group into the Portal Registry	6-18
Selecting Components to Include in Portal Navigation	6-22
Building a Search Index.....	6-23
Managing Base Portal Data	6-24
Managing Templates	6-24
Changing the “Look and Feel” of a Portal.....	6-28
PORTAL_HP_USER_TEMPLATE	6-28
PORTAL_HP_3COL_LAYOUT	6-29
PORTAL_HP_2COL_LAYOUT	6-29
PORTAL_HP_COMPONENT	6-29
PORTAL_HP_ACTIONBAR.....	6-29
PORTAL_HP_ACTIONBARIMAGE.....	6-29
PORTAL_UNI_HEADER_NEW	6-29
Registering Queries as Target Pages in the Portal	6-30

Chapter 7

Using Portal Navigation Features

Menu Navigation	7-1
Example of Three Column Navigation	7-2
Breadcrumbs	7-3
Favorites	7-4
Search	7-4
My Profile	7-5

Chapter 8

Working with Homepages

Creating a Personal Homepage	8-2
Customizing Content	8-2
Customizing Layout	8-3

Chapter 9

Building and Using Portal Search Indexes

Search Architecture Overview	9-1
PeopleSoft Portal Technologies	9-2
Verity Technologies	9-2
Verity File Locations	9-3
Building a Portal Search Index	9-5
Building a Search Index for a Portal Application	9-5
Submitting Search Requests	9-7
What Happens When a User Clicks “Go!”	9-8
Customizing Search Settings	9-9
Search Tips	9-10
Limitations to Portal Search Functionality	9-11

Chapter 10

Miscellaneous Portal Information

Upgrade/Copy Functionality in Application Designer	10-1
Using the Portal Over HTTPS	10-1
HTTPS Support Through a Proxy Server	10-2
For Apache	10-2
For WebLogic	10-3
Combining Forms and Frames	10-3
Specifying a Pagelet Name Instead of a URL in a Template	10-3
Posting from an HTML Template to a Frame Template	10-3
Maintaining PIA State	10-4

Cookie Handling	10-4
Setting Hyperlinks in the Portal	10-5
Ensuring that Server Names and/or IP Addresses are Consistent	10-6
Single-Signon	10-6
Order of Precedence of Refresh Tags	10-7
Using a Hardware SSL Accelerator	10-7
iQuery Pagelets	10-7
Linking Back to the Portal from a Different Database	10-7
Controlling the “Error Getting Content” Message	10-8
Requirements for Portal Help to Function Properly	10-8
Overriding Page Encoding	10-9
Events Requiring Re-Saving Personalized Homepages	10-9

Chapter 11

Pagelet SDK Information

Portal Pagelet Samples	11-1
------------------------------	------

Index

ABOUT THIS PEOPLEBOOK

This book describes the PeopleSoft Portal and related technology used for creating and managing a PeopleSoft Portal.

Audience

This book is intended for technical users, system administrators, and programmers who will be implementing, maintaining, or developing applications for your PeopleSoft system. To take full advantage of the information covered in this book, we recommend that you have a basic understanding of how to use PeopleSoft applications, system administration, and basic client/server and internet architecture. You should know how to navigate through the system and how to add, update, and delete information using PeopleSoft tables and pages. You should also have a basic familiarity with relational database concepts, SQL, and HTML.

Introducing the PeopleSoft Portal introduces the basics of portals in general, and the PeopleSoft portal in particular—including an architecture overview.

Understanding the Portal Registry gives an overview of the portal registry, the central repository of content references for use with the portal.

Understanding the Portal Servlet discusses the functions performed by the portal servlet, which include page assembly and proxying.

Designing Portal Templates provides information to help you understand how PeopleSoft portal templates are designed. Includes discussion and examples of page-based templates, frame-based templates, static templates, and dynamic templates.

Developing Pagelets addresses the design of pagelets, from a developer's perspective.

Using Portal Administration Features covers the various topics related to managing a PeopleSoft portal, including management of folders, content references, and content providers. Also documented is the menu import feature, for importing PeopleSoft menu group definitions into a portal registry.

Using Portal Navigation Features introduces the fundamental navigation features of the portal, including the universal navigation header, menu navigation, breadcrumbs, and the use of favorites.

Working with Homepages shows portal users how to customize the content and layout of their personal homepages.

Building and Using Portal Search Indexes discusses the Verity search engine, how to build search indexes from the portal registry or other data sources, and how to use the search functionality once a search index has been created.

Miscellaneous Portal Information addresses some technical details that may be of use to developers or those implementing a PeopleSoft portal.



This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Before You Begin

To benefit fully from the information covered in this book, you need to have a basic understanding of how to use PeopleSoft applications. We recommend that you complete at least one PeopleSoft introductory training course.

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

Related Documentation

To add to your knowledge of PeopleSoft applications and tools, you may want to refer to the documentation of the specific PeopleSoft applications your company uses. You can access additional documentation for this release from PeopleSoft Customer Connection (www.peoplesoft.com). We post updates and other items on Customer Connection, as well. In addition, documentation for this release is available on CD-ROM and in hard copy.



Important! Before upgrading, it is *imperative* that you check PeopleSoft Customer Connection for updates to the upgrade instructions. We continually post updates as we refine the upgrade process.

Documentation on the Internet

You can order printed, bound versions of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM. You can order additional copies of the PeopleBooks CDs through the Documentation section of the PeopleSoft Customer Connection Web site: <http://www.peoplesoft.com/>

You'll also find updates to the documentation for this and previous releases on Customer Connection. Through the Documentation section of Customer Connection, you can download files to add to your PeopleBook library. You'll find a variety of useful and timely materials, including updates to the full PeopleSoft documentation delivered on your PeopleBooks CD.

Documentation on CD-ROM

Complete documentation for this PeopleTools release is provided in HTML format on the PeopleTools PeopleBooks CD-ROM. The documentation for the PeopleSoft applications you have purchased appears on a separate PeopleBooks CD for the product line.

Hardcopy Documentation

To order printed, bound volumes of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM, visit the PeopleSoft Press Web site from the Documentation section of PeopleSoft Customer Connection. The PeopleSoft Press Web site is a joint venture between PeopleSoft and Consolidated Publications Incorporated (CPI), our book print vendor.

We make printed documentation for each major release available shortly after the software is first shipped. Customers and partners can order printed PeopleSoft documentation using any of the following methods:

Internet

From the main PeopleSoft Internet site, go to the Documentation section of Customer Connection. You can find order information under the Ordering PeopleBooks topic. Use a Customer Connection ID, credit card, or purchase order to place your order.

PeopleSoft Internet site: <http://www.peoplesoft.com/>.

Telephone

Contact Consolidated Publishing Incorporated (CPI) at **800 888 3559**.

Email

Email CPI at callcenter@conpub.com.

Typographical Conventions and Visual Cues

To help you locate and interpret information, we use a number of standard conventions in our online documentation.

Please take a moment to review the following typographical cues:

`monospace font`

Indicates PeopleCode.

Bold

Indicates field names and other page elements, such as buttons and group box labels, when these elements are documented below the page on which they appear. When we refer to these elements elsewhere in the documentation, we set them in Normal style (not in bold).

We also use boldface when we refer to navigational paths, menu names, or process actions (such as **Save** and **Run**).

Italics

Indicates a PeopleSoft or other book-length publication. We also use italics for *emphasis* and to indicate specific field values. When we cite a field value under the page on which it appears, we use this style: ***field value***.

We also use italics when we refer to words as words or letters as letters, as in the following: Enter the number *0*, not the letter *O*.

KEY+KEY

Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press W.

Jump Links

Indicates a jump (also called a link, hyperlink, or hypertext link). Click a jump to move to the jump destination or referenced section.

Cross-references

The phrase For more information indicates where you can find additional documentation on the topic at hand. We include the navigational path to the referenced topic, separated by colons (:). Capitalized titles in *italics* indicate the title of a PeopleBook; capitalized titles in normal font refer to sections and specific topics within the PeopleBook. Cross-references typically begin with a jump link. Here's an example:

For more information, see [Documentation on CD-ROM](#) in *About These PeopleBooks*: Related Documentation.

- Topic list

Contains jump links to all the topics in the section. Note that these correspond to the heading levels you'll find in the Contents window.



Name of Page or
Dialog Box

Opens a pop-up window that contains the named page or dialog box. Click the icon to display the image. Some screen shots may also appear inline (directly in the text).



Text in this yellow bar indicates information that you should pay particular attention to as you work with your PeopleSoft system. If the note is preceded by **Important!**, the note is crucial and includes information that concerns what you need to do for the system to function properly.



Text in this gray bar indicates For more information cross-references to related or additional information.



Text within this bar outlined in red indicates a crucial configuration consideration. Pay very close attention to these warning messages.

Comments and Suggestions

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

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

Or send comments by email to the authors of the PeopleSoft documentation at:

DOC@PEOPLESOFT.COM

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions. We are always improving our product communications for you.

CHAPTER 1

Introducing the PeopleSoft Portal

A portal is a web site that helps you navigate to other web-based applications and content. As far as users are concerned, they often consider a portal their “entry point”—the place they typically visit first after launching their web browser.

When one considers the volume of web-based content available, and the dynamic nature of it, the utility offered by portals quickly becomes apparent, even for advanced users. Any portal’s primary purpose is to guide a user from what appears to be the “top level” of the information heap to some very specific content somewhere else on the web. In fact, due to the nature of the web, there is no “top level” of information, no built-in hierarchy, no built-in navigation system. A portal helps to create this kind of context for users. With a portal, users can easily find information of interest when beginning from a known, reliable starting point.

A portal must accomplish this objective of “guiding users to content” equally well for everyone, whether they are casual or advanced users. If one group of users finds that a portal is too complicated, and another group finds it too limited to be of value, both groups will simply find another way to navigate to the information of interest to them. Therefore, if a portal is to be successful in its purpose, it must offer a great degree of flexibility and appeal to users at all levels.

There are numerous portals available to anyone on the web, and you’ve undoubtedly used at least one portal, if not several different ones. Some common “general purpose” portals are Yahoo, Excite, and Netscape’s Netcenter. To understand what a portal is, from the perspective of today’s web user, you need only point your browser to one of these sites to see what portals look like and how to interact with them.

General Characteristics of Portals

It’s clear that a portal is a unique kind of web site. The most unique characteristic of a portal, in comparison to other web sites, is that it really does not contain much, if any, unique content. Instead, it provides a simple way for you to get to the content that interests you, no matter what or where that content is. Although portals may appear to be relatively simple to the user (after all, that is the point—to simplify access to other content), they must be carefully designed to accomplish a handful of tasks well. Portals typically provide at least the following features:

- **A default homepage.** This is the first page a new user will see. It contains all the basic features of the portal.
- **A customizable homepage.** This is a variation of the default page on which the user can add, remove, and arrange content to their liking. Examples include My Yahoo and My Excite.
- **Category-based navigation.** Portals usually classify content references into a hierarchical list

of categories to help the user drill down to specific information. The categories are presented more like “channels” than traditional menus.

- **Links to related content.** The portal attempts to assist users by providing links that may be of interest based on what the user is currently viewing.
- **A search engine.** Typically, a portal enables users to enter queries, and displays a set of search results.

The PeopleSoft Portal

The PeopleSoft portal is a *business portal*. It is much like the general purpose portals described above, except that its main purpose is to help users be more effective in accessing information related to performing their jobs. The portal’s “look” will change based on a number of variables. PeopleSoft portals have default homepages that resemble the following example.



A PeopleSoft portal

Portal Architecture

Using PeopleSoft portal design tools, developers can build a portal specifically for use with PeopleSoft applications, as well as other web-based applications and content. The main pieces of the portal architecture are as follows:

- Portal templates and template pagelets
- Personal homepages and pagelets
- Portal registry
- Portal servlet
- Navigation features

- Search functionality

Portal Templates and Template Pagelets

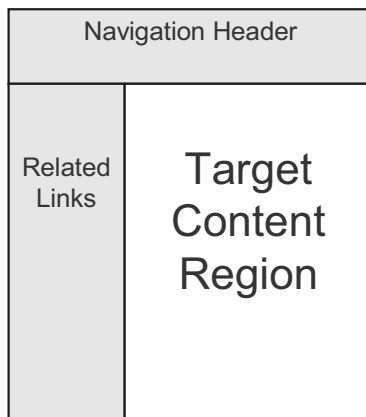
Developers create portal templates in Application Designer. At runtime, the portal template is constructed into a web page by the PeopleSoft portal. Each portal template is made from various template pagelets.

Each portal template (and each template pagelet) is designed individually and stored as an HTML object in the portal database. PeopleSoft 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.” Although not shown in the above example, a target page contains the specific HTML page that a user requested. For example, if a user were running a PeopleSoft Financials application, the page they’re currently accessing would appear 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.

The portal template in the example below is comprised 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 would be filled by the HTML returned by the target page, as would the other template pagelet regions.



Different parts of a portal template

Personal Homepages and Pagelets

A *personal homepage* is the user’s entry point into all PeopleSoft applications and related content. It is what they see first when they log into the PeopleSoft portal. There is a default personal homepage accessible by everyone, constructed from some pre-existing portal template. However, each user can easily define a “customized” homepage to replace the default one. A

personalized homepage allows the user to add, remove, and rearrange blocks of content in the same fashion as My Yahoo or My Excite.

Each “block of content” on the homepage is called a *pagelet*. It displays summarized information within a small rectangular area on the page. Pagelets are intended to provide features such as an integrated work list, high level navigation, graphs summarizing EPM statistics, and so on.

The example below shows what a personal homepage may look like in a PeopleSoft portal.



A personal homepage

Portal Registry

The portal registry is a tree-like structure in which content references are organized, classified, and registered. It is stored in a set of tables within a PeopleSoft database made specifically for hosting portal registries. Within each portal registry are folders and content references. *Folders* are similar to nodes on a tree, and, as such, they can be nested to create a multi-level hierarchy. Folders contain *content references*, which are URLs that have been “registered” in the portal registry.

In addition to specifying a URL, each content reference includes additional information, such as who created it, dates when it is effective, when it expires, and so on. The URL can point to any web site that responds to HTTP requests with an HTML response—in other words, some type of static or dynamic web page. One example of a content reference is a URL that points to a page in a PeopleSoft application. 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 permission lists assigned when the content reference is created.

All application pages in the various PeopleSoft product line databases are registered, secured, and described in the portal's registry. Additional content may be registered within the portal registry from sources such as the company's intranet and reporting systems.

The hierarchical structure of the portal registry enables a portal administrator to create a classification and navigation system in which content references can be registered and managed for all portal users. PeopleSoft provides a web-based Portal Administration utility for this purpose. Additionally, a registry API is provided for accessing each portal registry from PeopleCode, COM, or C programs.

Portal Servlet

The portal servlet is a Java servlet that runs on the portal web server. Its primary purpose is to intercept user requests for HTML pages, retrieve the requested page, wrap additional content around it, and then send the resulting page to the user's browser. In other words, the portal servlet acts like an "invisible browser" that sits between the user's browser and requested content. What's the purpose of this intermediate step? There are several main reasons for it:

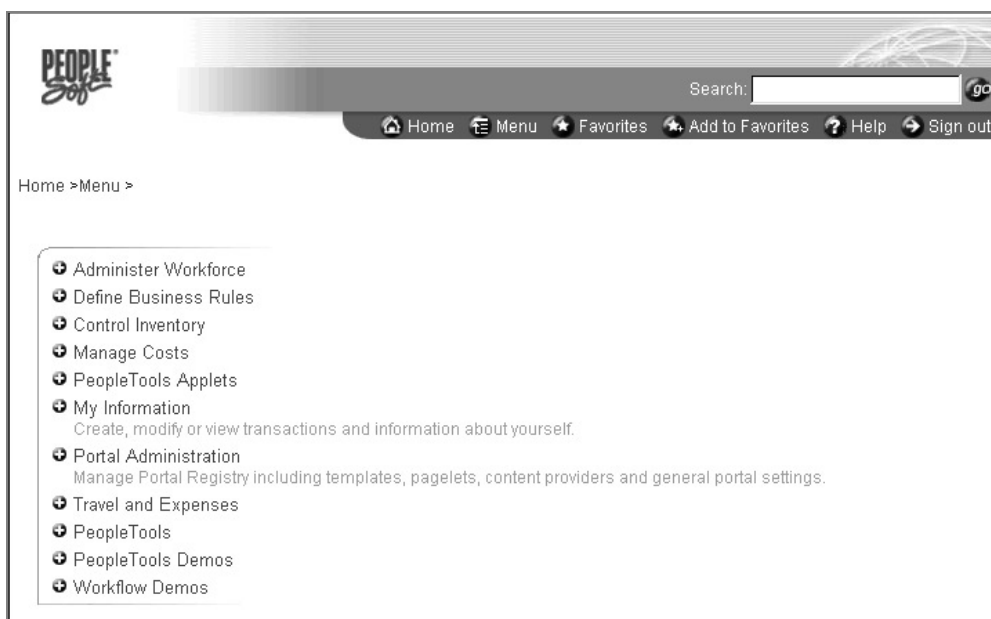
- **To provide a consistent user interface.** The portal servlet checks some of the properties associated with each content reference, including the name of a portal template. When a user accesses content through the portal, the portal servlet wraps the target page with the portal template specified in the content reference. This template provides a consistent user interface.
- **To ensure that PeopleSoft-specific tags get 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 any other HTML content, at runtime. The portal servlet then sends the resulting page to a browser as a single HTML document.
- **To ensure enforcement of PeopleSoft security.** The portal servlet ensures that the portal displays only the pages to which a user has access.

Navigation

Navigation is provided through template pagelets that allow the user to move through the portal registry visually, using folder drilldowns. Users can view the registry from a number of vantage points, including menu groupings and a "favorites" list.

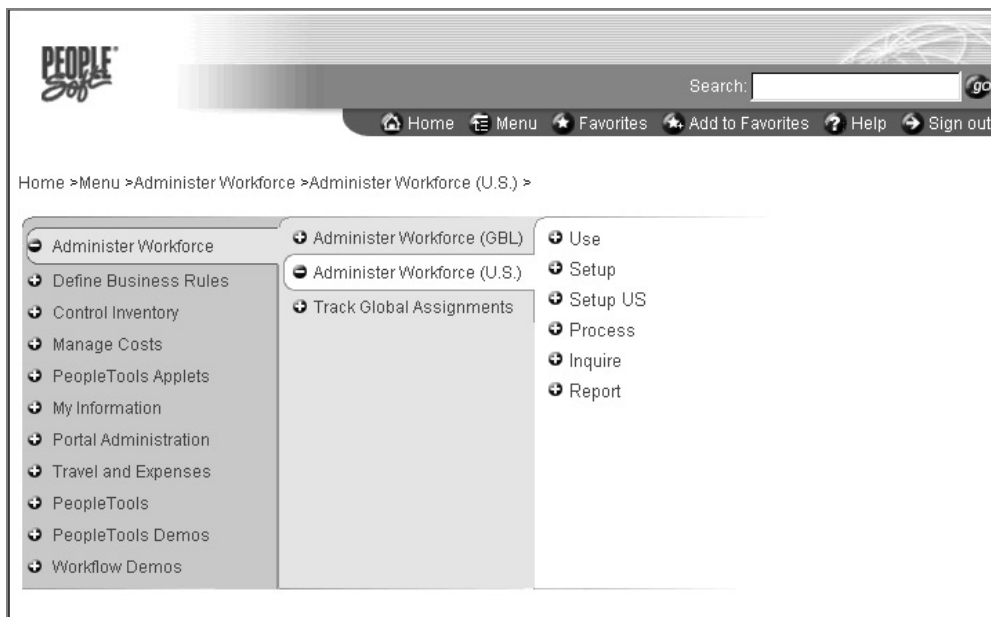
Menu Navigation

Product-oriented navigation is provided through the **Menu** button and the "three column" navigation feature. When **Menu** is selected, users see a list of applications retrieved from the first level of the portal registry. The portal applies security to filter the set of links displayed, so users see only the folders to which they have access. In the example below, the user has access to only one application—**Administer Workforce**. The **Portal Administration** option is available to users who have been granted appropriate permissions for managing the portal registry. **My Profile** is available to all users.



Menu navigation in a PeopleSoft Portal

Selecting one of the listed applications from the menu, such as Administer Workforce, presents users with another menu column, from which they can select additional options. The portal displays up to three levels of the registry at once, as shown below.



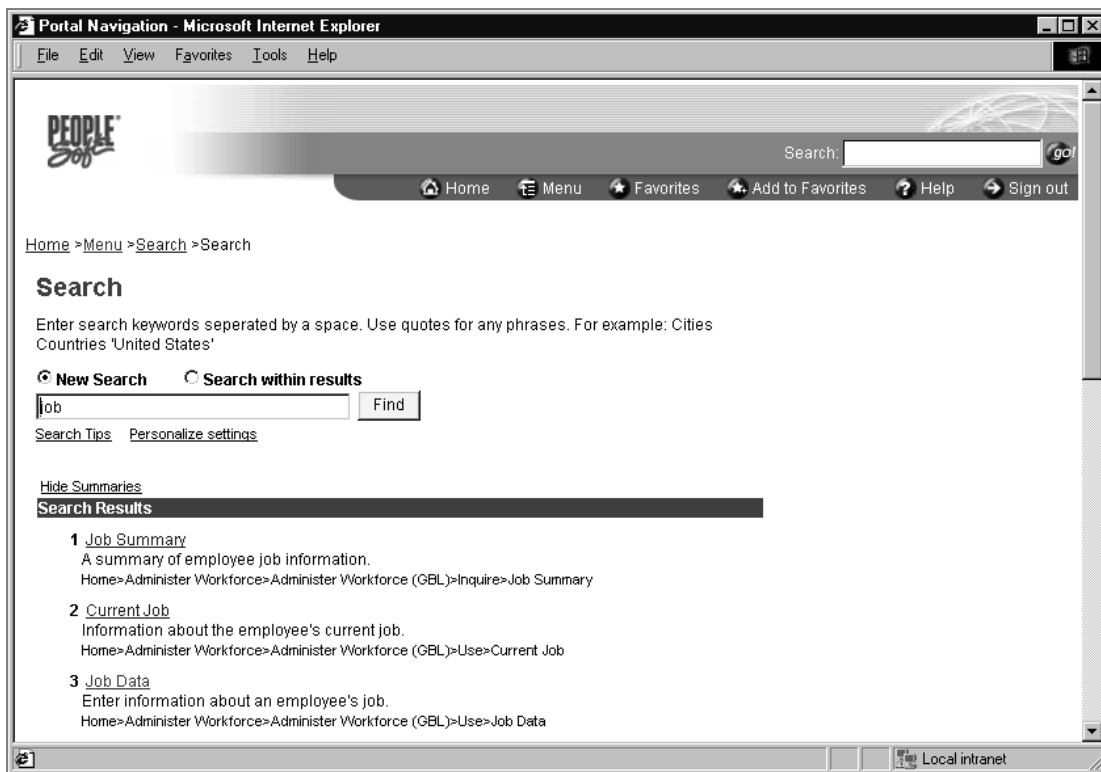
Using the three column navigation feature to navigate the portal

Favorites

The **Favorites** button enables users to create and manage their own lists of links to folders and content references. A user can add new links using the **Add to Favorites** button. Clicking on any of these links takes the user directly to the selected page.

Search Functionality

A search engine is built in to the PeopleSoft Portal. The search functionality depends on two main sets of technology: PeopleSoft-built Portal technology (such as the portal registry, APIs, and so on) and the Verity search engine. The end result of combining these technologies is that portal users can easily and efficiently search for any content references registered in the portal registry.



The Search Results page



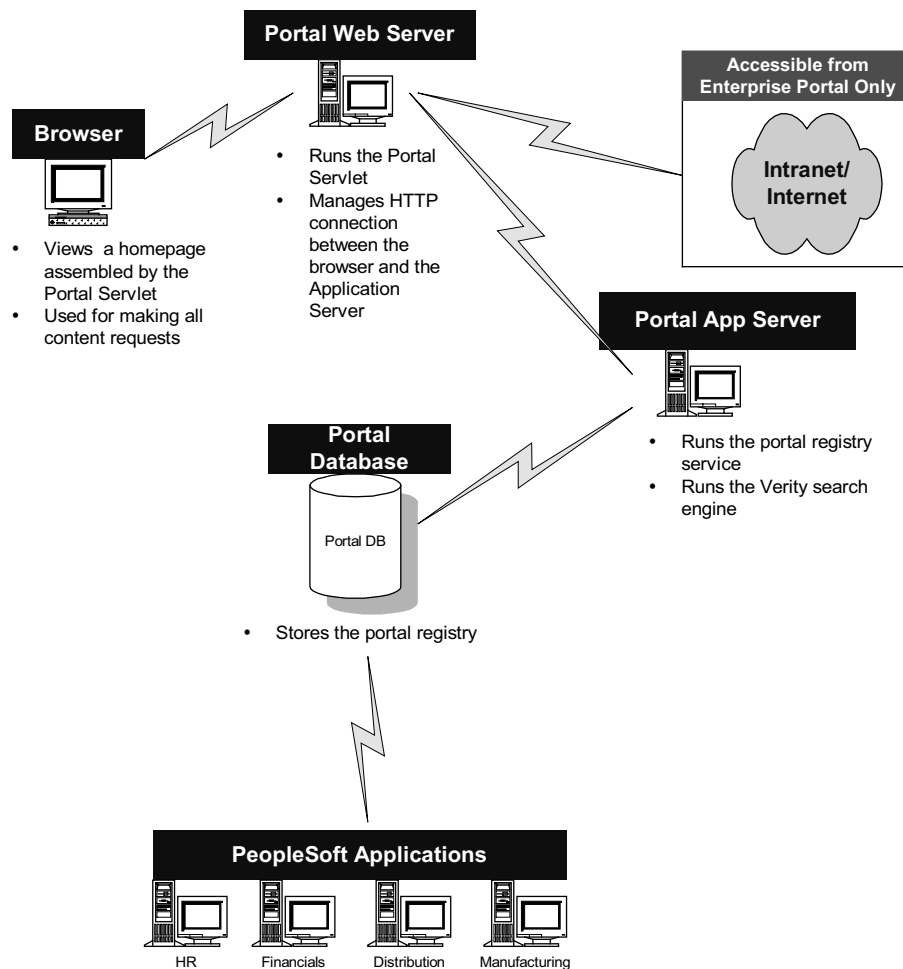
For more information on search functionality, see Building and Using Portal Search Indexes.

Portal Architecture—Putting It All Together

When we speak of the “PeopleSoft portal,” we are referring to all of the technologies above working together. The portal is always positioned between a user’s browser and the content they

are requesting. The portal intercepts requests for content, retrieves the content, and then reformats it (through the portal servlet) before sending it back to the user's browser, as shown below.

The various parts of the portal architecture are brought together using established pieces of the PeopleSoft Internet architecture. These include a browser (such as Netscape Communicator or Microsoft Explorer) for displaying portal homepages, a web server, an application server, and a PeopleSoft portal database (for storing the portal registry). The portal registry contains the content references to specific pages in PeopleSoft applications. Additionally, if either the Enterprise Portal product or a Portal Solution product is used, content references from non-PeopleSoft sites can be stored in the portal registry, and can be retrieved and formatted by the portal servlet. The figure below shows the relationship among the different pieces of the architecture.



Portal architecture

Types of Portals Available

There are three broad categories of PeopleSoft portals available:

- PeopleSoft portal
- Enterprise portal
- Portal Solutions

All of these portal types are built with exactly the same set of portal technologies delivered by PeopleTools. The differences are in the target audience, type of content that can be accessed, and pricing structure—not in the technology used. This section discusses the basic features of the PeopleSoft portal delivered with PeopleSoft Applications.



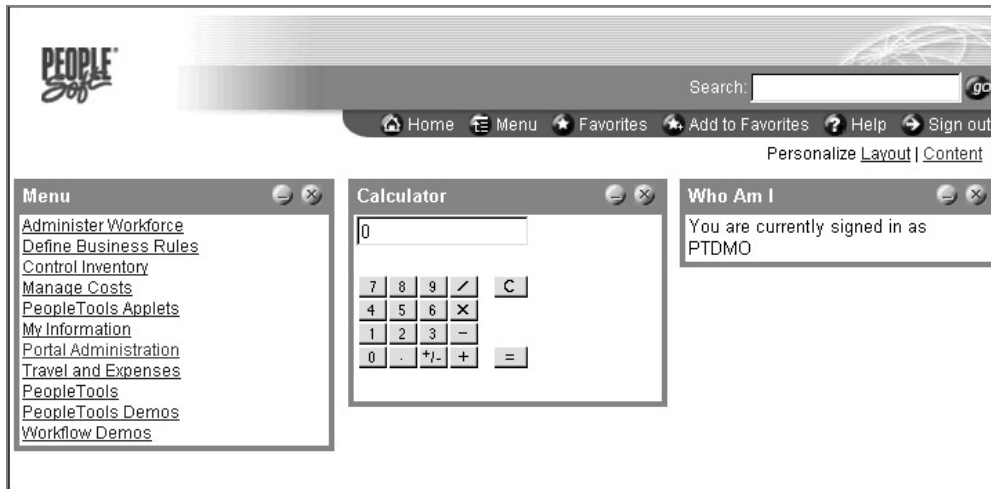
For more information on the PeopleSoft Enterprise Portal and PeopleSoft Portal Solutions, see the documentation for those products.

PeopleSoft Portal

The basic PeopleSoft portal is bundled with every PeopleSoft 8 application. It provides the following functionality:

- **A simple menu-based navigation system.** The menu navigation system looks the same as that provided with PeopleSoft Internet Architecture applications (those not running through the portal). However, the portal menu system retrieves its information from the portal registry, whereas PIA applications retrieve their menu information directly from menu group definitions.
- **A personalized web experience.** The ability of users to customize their homepages is an integral feature of the PeopleSoft portal.
- **Integrated access to licensed PeopleSoft applications.** Users are provided single-signon ability to all PeopleSoft applications.

Navigation to content outside of PeopleSoft applications is not provided with the basic portal, but you can add content to the portal that is related to the applications you have licensed.



PeopleSoft Portal



From the perspective of users, the “menu navigation” available in the portal by clicking the **Menu** button looks identical to the “menu navigation” employed when users log on to a PeopleSoft Internet Architecture application *without the portal*. This is a potential point of confusion. The main difference in appearance is that the portal provides users with the “universal navigation header” that includes some additional buttons not included with menu navigation. See “How does PIA Navigation relate to Portal Navigation?” below for more information.

Single Signon

PeopleSoft supports single signon for use with the PeopleSoft Internet Architecture (PIA) configuration. 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 a second PeopleSoft application server 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. Recall that each PeopleSoft product line, such as PeopleSoft HRMS or Student Administration, resides in its own database.

After the first application server/node authenticates a user, PeopleSoft delivers a web browser cookie containing an authentication token. PIA uses web browser cookies to store a unique access token for each user 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 re-authenticate the user behind the scenes so they don’t have to go through the signon process again.

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. Users need to signon once and be able to navigate freely without encountering numerous signon screens. Because single signon is so integral to the portal, you always need to configure it before deploying the portal.



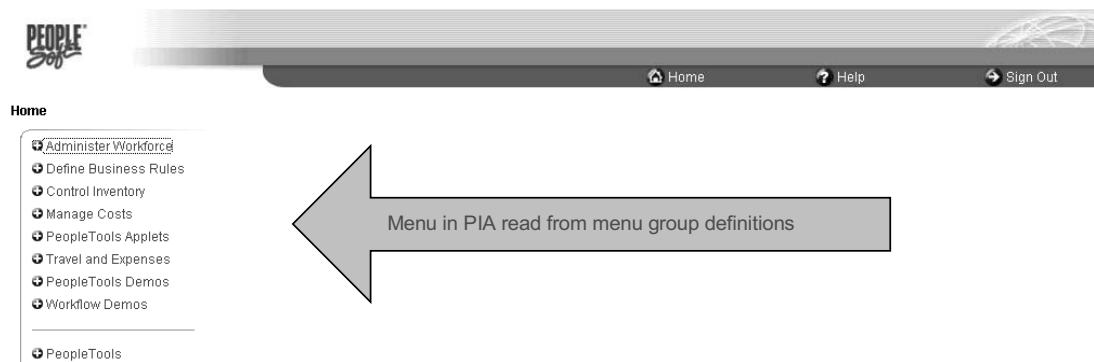
For more information, see Security

How does PIA Navigation relate to Portal Navigation?

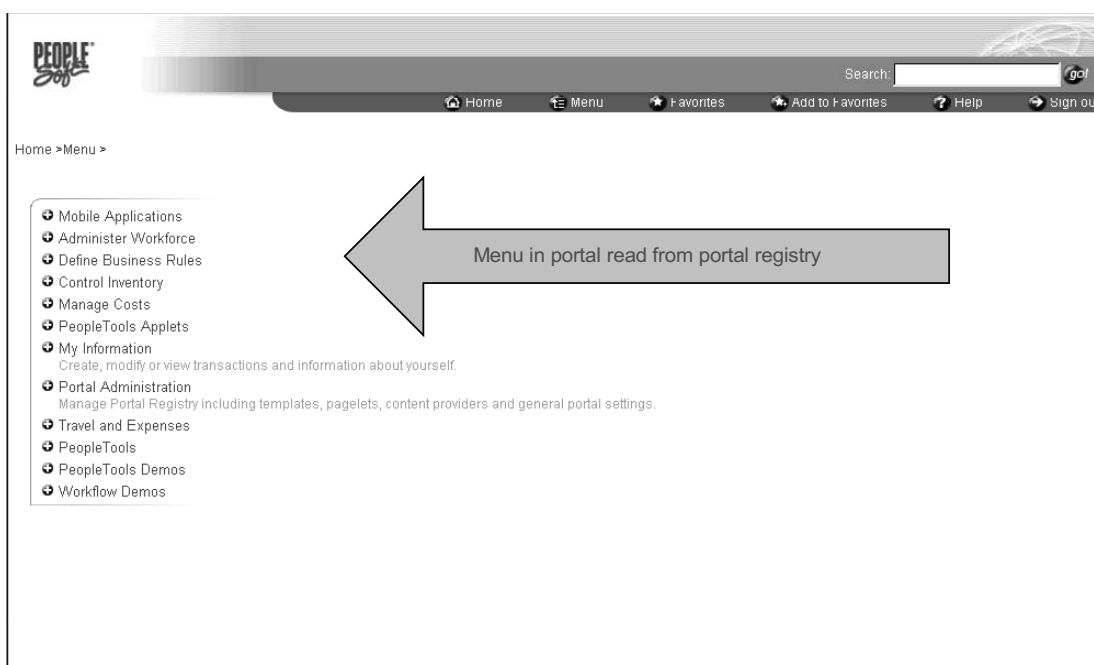
Menu navigation is the default navigation system employed when a user logs on directly to an individual PeopleSoft Internet Architecture (PIA) application—as opposed to logging on via the portal. The menu-like presentation is nearly identical to the navigation system provided in the portal menu. However, there are two main differences between the menu navigation and the portal menu.

- **Portal menus are read from the portal registry, while the PIA menus are read directly from menu definitions.** The portal registry is a series of tables in a database that stores every content reference (typically, a URL) available through the portal. Although you can import PeopleSoft menu groups into a portal registry, the registry is completely independent of the menu definitions stored in your PeopleSoft database. Suppose you import an existing menu group definition into the portal registry. Once the menu group has been imported, any further changes you make to the menu group definition, menus, menu items, and menu security through Application Designer *will not be reflected in the portal registry* (unless you import the menu group again). On the other hand, the menu navigation provided with PIA is *directly* based on the menu definitions stored in your PeopleSoft application. These definitions are read at runtime to create the menu structure dynamically, much like the panel processor used to do for the PeopleSoft Windows client. Any changes you make to the menu definitions are immediately reflected in the menu navigation. While the PIA's menu navigation system is simpler for existing applications based on our old-fashioned menu definitions, keep in mind that the portal registry is far more flexible and—unlike the PIA menu navigation—can readily handle links to external content.
- **The portal uses a universal navigation header, while PIA uses a limited navigation header.** The navigation header provided within PIA is limited to three navigation buttons: **Help**, **Home**, and **Signoff**. The portal's universal navigation header includes a number of additional buttons (**Favorites**, **Add to Favorites**, and **Help**).

The illustrations below show the main differences between the PIA menu navigation system and the portal navigation system.



Limited navigation header within a PIA application



Universal navigation header within the PeopleSoft Portal

Another easy way to determine if you're logging on to an individual PIA application is that its URL typically ends with **signon.html**. If you're logging on to the portal, the URL typically ends with **signonportal.html**.

Portal Terminology

Term	Definition
------	------------

Application Server	The application server is the centerpiece of PeopleSoft's three-tier architecture. It utilizes Tuxedo, BEA Systems' transaction monitor, to manage client transactions and provide the business rules and workflow capabilities of PeopleSoft's enterprise applications.
BIF file	This is the bulk insert file (input.bif) used with the Verity search engine to specify the documents to be submitted to a collection (search index). It contains a unique key, document size (in bytes), field names and values, and document location in the file system.
Breadcrumbs	<p>Breadcrumbs show the navigation path to the current web page location. As you drill down through the different levels of the registry, a “breadcumb trail” appears that shows the path you’ve selected. Each registry level is separated by an angled brace (>), and you can select any level to navigate directly back to that level.</p> <p>A typical Breadcrumb would look like this:</p> <p style="margin-left: 40px;">Home > HR > Administer Workforce > Benefits</p>
Collection	To make a set of documents available for searching in Verity, you must first create one or more collections. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents matching various search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Since a collection can only store information for a single locale, PeopleSoft maintains a set of collocations (one per language code) for each search index object.
Content Provider	A content provider is a name that you can use to refer to some source of HTML content. In more practical terms, a content provider is a URI string that defines the database and server to be used when the portal servlet attempts to retrieve content, proxy addresses, and assemble pages. The use of content providers also simplifies PeopleCode programming, since long URLs can be referred to quite easily by the appropriate content provider name. Some content provider names (HRMS, EPM, SA, FDM, and CRM) are preset in your portal. You can add additional content providers as necessary.

Content Reference	Content references are pointers to some kind of content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three broad categories: target content, templates, and template pagelets.
Cookie	A cookie is information that a Web site puts on your hard disk so that it can remember something about you at a later time. (More technically, it is information for future use that is stored by the server on the client side of a client/server communication.) Typically, a cookie records your preferences when using a particular site. Using the Web's Hypertext Transfer Protocol (HTTP), each request for a Web page is independent of all other requests.
DAT file	A text file (input.dat) used with the Verity search engine that contains all of the information from documents that will be searchable but not returned in the results list.
Enterprise Portal	The PeopleSoft enterprise portal is a separate product offering purchased independently of any other PeopleSoft applications. It can be used with or without any PeopleSoft application. It can be used as a standalone corporate portal that does not access PeopleSoft data at all.
Homepage	A homepage is the first page a user sees upon logging into a portal. Users can define and store their own personal homepage in which they specify their preferences for layout and content.
HTTP	HTTP is short for HyperText Transfer Protocol, the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. The other main standard that controls how the World Wide Web works is HTML, which covers how Web pages are formatted and displayed.
mkvdk	Verity's command-line tool used to index a collection, insert new documents, perform simple maintenance tasks like purge and delete a collection, and control indexing behavior/performance.
Page	<ol style="list-style-type: none">1. A page defined in Application Designer as part of a PeopleSoft Internet Architecture application.2. Any web page.

Page Assembly	Page assembly is one of the functions of the portal servlet. Page assembly involves intercepting the user's content request, retrieving the content, and properly formatting it using a pre-defined portal template. To complete the page assembly process, the portal servlet merges content from any HTML documents that it retrieves along with the defined template HTML. The assembled page is then sent back to the user's web browser as a single HTML document.
Pagelet	A page designed to appear on a customized homepage. A pagelet is smaller than the typical page dimensions in many PeopleSoft applications. It can be based on either a page designed in Application Designer or on an iScript.
PeopleSoft Portal	The portal bundled with every PeopleSoft 8 application. It provides a simple navigation system, based on existing menu definitions that have been imported into the portal registry. Navigation to content outside of PeopleSoft applications is not provided.
PIA	PeopleSoft Internet Architecture. This is the fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of an RDBMS, an application server, a web server, and a browser.
Portal	A portal is a web site that helps you navigate to other web-based applications and content. Users often consider a portal their "entry point"—the place they typically visit first after launching their web browser.
Portal Registry	The portal registry is a tree-like structure in which content references are organized, classified, and registered. It is a central repository that defines both the structure and content of a portal through a hierarchical, tree-like structure of "folders" useful for organizing and securing content references.
Portal Registry API	The Registry API is provided for accessing each portal registry from PeopleCode, COM, Java, or C programs. Providing the same kind of registry management capability as the online administration pages, it can be used by external systems to update the registry to reflect changes in the content reference URL, taxonomy, and effective dates. The Registry API is fully described in the PeopleCode documentation.
Portal Servlet	A Java servlet that runs on a web server. The portal servlet intercepts user requests for content, retrieves content, and builds a single HTML document to be displayed in the user's browser.

Portal Solutions	Portal Solutions are separate product offerings from PeopleSoft that consist of pre-built, packaged solutions focused at different audiences (customers, suppliers, and employees). Because they are both pre-built, supported application products, Portal Solutions can be deployed swiftly and easily, saving significant resources when compared to other custom-built solutions.
Query String	A query string is the text following a “?” in a URL.
RDBMS	Relational Database Management System.
SearchIndex	A set of objects that give the programmer the ability to create, delete, insert, and update a search index and the items within it. Search index items contain a set of statistics about the document that has been indexed (keywords, number of occurrences, proximity to other words, and so on) as well as a key that can be used to point to the document (a URL, database key, or file path).
SearchQuery	A set of objects that allow the programmer to pass a query string and operators to the search engine and receive a set of matching results with keys to the source documents from the search index in return.
Single Signon	This refers to the process by which a user can, after being authenticated by one PeopleSoft application server, access a second PeopleSoft application server without entering a user ID or password.
Style File (Verity)	Collection style refers to a set of configuration options that are used to create the indexes associated with a collection. A collection has one collection style and it is defined in a set of style files before creating the collection.
Template	A portal template is simply HTML code, associated with a web page, to define the style and layout of the page. Templates allow a developer to build an HTML page by combining HTML from a number of sources. Templates do two basic things: define the layout of the page, and define where to get HTML for each part of the page.
Template Pagelet	One piece of an overall template. For example, in a given template, there may be one template pagelet for the universal navigation header and one template pagelet for the target content.

Universal Navigation Header	Every PeopleSoft portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like Home , Menu , Favorites , and so on) the universal navigation header can also display a welcome message for each user.
URI	A URI is a part of any complete URL, but does not include the query string (the text following a ?) associated with some URLs. You can think of it as a subset of the URL that points to the resource, but does not include any parameters being passed to that resource. In comparison to the URL example below, the URI portion of the URL is as follows: <code>http://serverx:port/servlets/iclientservlet</code>
URL	In this document, the term URL refers to an entire URL, including the query string. The following is an example of a URL: <code>http://serverx:port/servlets/iclientservlet?ICType=Script&ICScriptProgramName=WEBLIB_BEN_401k.PAGES.FieldFormula.iScript_Home401k</code>
VdkVgwKey	A key within a Verity BIF file for every document to be indexed. VdkVgwKey values must be unique across all collections that will be searched in any one application.
Verity	The third-party search engine integrated with the PeopleSoft Portal.
Verity Fields	Verity fields are stored in the collection for retrieval and searching, and can be returned on a results list. Fields are defined in the BIF file and stored in the collection for retrieval and searching, and can be returned on a results list. Fields, like date and numeric fields can be used with the comparison operators (<,<=,>,>=).
Verity Thesaurus	The custom thesaurus consists of lists of synonyms defined in a synonym control file and can be used for synonym searching. After defining synonym lists in the control file, you use the mksyd utility to create a custom thesaurus (a control file which has the .syd extension) that the search engine uses.

Verity Topics	Verity applications can provide end users with predefined search criteria called <i>topics</i> . A topic is a named object that represents a concept, or subject area and can be used for synonym searching. It consists of words and phrases grouped together using the Verity query language in a tree-like structure. When provided, topics can be shared by all users.
Verity Zones	Zones are specific regions of a document to which searches can be limited. When the zone filter is used, the Verity engine builds zone information into the collection's full-word index. The index, enhanced with zone information, permits quick and efficient searches over zones. Searching a zone is faster than field searching. Zones are defined in the DAT file. The contents of a zone cannot be returned in the results list of an application.
Web Server	The PeopleSoft Internet Architecture requires a web server that supports Java servlet execution. A specific version of the Apache web server binaries are delivered with PeopleSoft 8.

CHAPTER 2

Understanding the Portal Registry

The portal registry is a central repository that defines both the structure and content of a portal. It provides a hierarchical, tree-like structure of “folders” useful for organizing and securing content references. A content reference is simply a reference to a given URL. Once a content reference has been entered in the portal registry, it is then defined by a number of attributes (name, description, and so on) that are useful within the portal environment, and is considered to be “registered.”

Although the PeopleSoft portal can work with either registered or unregistered content references, registered content is advantageous for several reasons:

- **It can be secured.** Registered content can be secured in the portal registry.
- **It can be associated with a specific portal template.** Every registered content reference has certain attributes, including the name of the associated portal template. If an associated portal template is not provided, then the portal template associated with content provider is used. If a template for the content provider has not been specified, then the default template for the portal is used. With unregistered content, the default template is always used.

Every component delivered with PeopleSoft applications is pre-registered within the portal registry. However, because of the dynamic nature of web-based information, there needs to be an easy way to add, change, or delete content references within the portal registry. This ability is provided through a set of administrative pages delivered with the PeopleSoft portal. Using these administrative pages, a user with appropriate access can manage the registry, including folder management, content reference management, and security administration.

Additionally, to provide programmatic access to the portal registry, PeopleSoft delivers a portal registry API. This API provides the same kind of registry management capability as the administration pages described above, and is fully described in the PeopleCode documentation.

Portal Registry Architecture

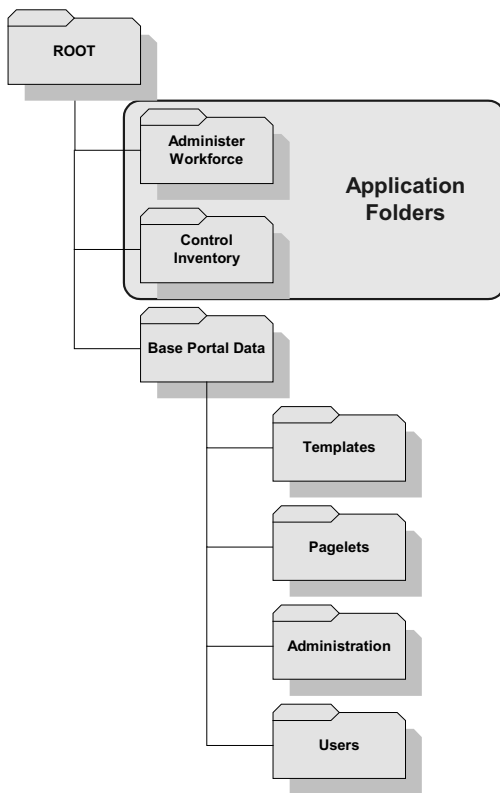
Each PeopleSoft portal is defined by one portal registry. The portal registry is stored in a PeopleSoft database, and can exist within an existing PeopleSoft application database or as an independent PeopleSoft database. A single portal database can support multiple portal registries, and therefore multiple portals, but only one portal registry is associated with any single portal.

PeopleSoft applications are delivered with the portal registry existing within the PeopleSoft application database. Enterprise portals are delivered as a separate PeopleSoft database.

The registry is constructed of a set of records within a PeopleSoft database forming a tree-like structure. Each portal registry consists of the following objects:

- Folders
- Content References
- Content Providers

The above items define the basic structure, content, and characteristics of the portal registry. In addition, every portal registry contains a set of standard folders. These include the root folder and the Base Portal Data folder. The Base Portal Data folder includes the following folders: Templates, Pagelets, Administration, and Users. In addition to these standard folders, there typically will be one folder per PeopleSoft application located directly off of the root folder. These application folders contain the folders and content references associated with each PeopleSoft application you've licensed.



The portal registry



Although it's appropriate to think of the portal registry as a tree-like structure, PeopleSoft Tree Manager is not used to manage this hierarchy.

The portal registry is stored in a PeopleSoft database. However, to improve performance, a copy of the registry also is stored on the Application Server in cache files; this is how the portal servlet accesses the registry at runtime.

Portal Registry Properties

Each portal registry has the following properties that can be set through the portal administration pages or through the portal registry API.

Name. A unique value for the registry.

Long Description. A description of the registry.

Default Portal Template. The default portal template associated with the portal registry. Used by the portal servlet to determine how to lay out the page if it cannot find a template associated with a specific content reference.

Template Collection. A collection of template objects defined for the portal registry.

Content Providers. Each content provider is a name that you can use to refer to some source of HTML content.

Folders

The portal registry includes “folders,” which are useful for grouping and organizing content references into a hierarchy. Except for the root folder, each folder has a parent folder, and each folder can contain content references, as well as other folders.

Folder Properties

Each folder has the following properties that can be set through the portal administration pages or through the portal registry API:

Name. A unique value within this folder’s parent folder.

Label. A short description used when displaying the folder.

Folder ID. A unique value within each registry allowing a folder to be identified without traversing the whole tree.

Parent ID. The folder ID of this folder’s parent.

Long Description. A description of the folder.

PeopleSoft Product. The PeopleSoft product in which this folder can be found. This provides a logical way of categorizing folders by product, so that registry operations can be performed against a set of associated folders.

Author. The PeopleSoft user who created the folder.

Created Date. Date the folder was created.

Valid From Date. A date that can be used by applications to determine when this content reference will be available to users. Note that this date is ignored by the portal, but is provided for use with PeopleSoft applications that may have use for it.

Valid To Date. A date that can be used by applications to determine when this content reference will be available to users. Note that this date is ignored by the portal, but is provided for use with PeopleSoft applications that may have use for it.

Attribute Collection. Contains name/value pairs that can be set on the folder. For more information, see the “Attributes” section.

Folder Collection. Contains any child folders in the folder.

Content Reference Collection. Contains any content references in the folder.

Permission List. Determines which permission lists have access to this folder.

Content References

Content references are pointers to some kind of content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three broad categories: target content, templates, and template pagelets.

Content Reference Properties

Each content reference has the following properties that can be set through the portal administration pages or through the portal registry API:

Name. A unique value within this folder’s parent folder.

Label. A short description used when displaying the content reference.

Content Reference ID. A unique value within each registry allowing a content reference to be identified without traversing the whole tree.

Long Description. A description of the content reference.

PeopleSoft Product. The PeopleSoft product in which this content reference can be found. This provides a logical way of categorizing content references by product, so that registry operations can be performed against a set of associated content references.

Author. The PeopleSoft user who created the content reference.

Created Date. Date the content reference was created.

Valid From Date. A date that can be used by applications to determine when this content reference will be available to users. Note that this date is ignored by the portal, but is provided for use with PeopleSoft applications that may have use for it.

Valid To Date. A date that can be used by applications to determine when this content reference will be available to users. Note that this date is ignored by the portal, but is provided for use with PeopleSoft applications that may have use for it.

URL. The URL address of the content.

Portal Template. The portal template associated with the content reference. Used by the portal servlet to determine how to lay out the page.

Attribute Collection. Contains name/value pairs that can be set on the content reference. For more information, see the “Attributes” section.

Permission List. Determines which permission lists have access to this content reference.

Content Providers

When a content references is created, the specific URI (the web server, port, and so on) can be specified in the content reference. However, this can be a drawback. For example, whenever a new portal is set up or something is changed on the web server, all content references that reference that web server must be updated to reflect the change. Having a “content provider” solves this problem. A content provider is nothing more than a logical name for a specific web server address. So, instead of specifying a specific URI for each content reference you create, you can instead specify a content provider. Each content provider is defined just once, so if you change the specific URI that the content provider references, all associated content providers are updated automatically.

Content Provider Properties

Name. A unique value.

Long Description. A description of the content provider.

Portal URI Text. The location (URI) of the content provider. At runtime, this location will replace the content provider name in any content references that reference this content provider.

Default Template Name. The portal template associated with the content provider. Used by the portal servlet at runtime. If the content reference does not have specify a template, then the template for its content provider is used. If there is not a content provider template, then the portal default template is used.

Attributes

Any folder or content reference can have as many attributes added as required. Attributes are simply name/value pairs that can be defined and used by portal-aware applications, functions, and so on.

Security

The same security mechanism is used for both folders and content references. Folders and content references can be marked as public, owner accessible, or they can have explicit PeopleSoft permission values set, including cascading the permissions to child folders and content references.

When marked as “public,” the folder or content reference is accessible by any user. The “public” permission cannot be automatically inherited by children.

When marked as “owner accessible,” a folder or content reference is only accessible by the user who created the folder or content reference. The “owner accessible” permission cannot be automatically inherited by children.

Folders and content references can be marked as accessible by PeopleSoft permissions defined in Maintain Security. This means that if the user is a member of a role and the role contains that permission, then the user has access to the folder or content reference. Additionally, when applied to folder permissions, the permission can be inherited, and any child of that folder—including a content reference—automatically has that same permission added to its permission list.

Maintaining the Portal Registry

Access to the portal registry is provided through three separate means:

- **A menu import feature.** This feature enables you to import an existing menu group definition to the portal registry. This is a convenient method for initially creating a portal registry, based on the menu group definitions already stored in a PeopleSoft application.
- **Portal administration pages.** These pages are convenient for adding, changing, or deleting folders and content references from a portal registry.
- **The portal registry API.** This API provides programmatic access to the registry.

Understanding the Portal Servlet

The portal servlet is at the heart of the PeopleSoft portal runtime architecture. It is what enables the portal to integrate content from a wide variety of sources and present it on a single page in a coherent, consistent fashion in a user's browser. The portal servlet has several main functions, which change slightly depending on whether it is handling page-based or frame-based templates:

- **It 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.
- **For page-based templates, it assembles pages for the browser.** The portal servlet intercepts 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. We call this process *page assembly*. The portal servlet uses a template to “wrap” the contents of the requested page into the context of the site (headers, footers, bread crumbs, 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.
- For page-based templates, it assures that all URL references in the HTML on the assembled page are references back to the portal servlet itself. The portal servlet assembles pages for the browser, and these pages typically point to numerous other web pages from different sources on the Internet. Since 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. Therefore, 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, it updates the frameset with the target content and sends it to the browser.** When working with a frame-based template, the portal servlet inserts a URL into each frame and sends it along to the browser, rather than retrieving documents for the browser as it does with page-based templates.

Setting Up the Portal Servlet

The portal servlet is a Java servlet that runs in the web server environment. It must be set up properly on a web server before the PeopleSoft portal can service user requests. The web server and portal servlet environment should be configured and tested during the installation of your PeopleSoft applications.

For more information on the initial installation of PeopleSoft applications, including setting up the portal servlet, see the *PeopleSoft Installation and Administration* book for your products. For

more details related tailoring the portal servlet environment for your specific requirements, see *PeopleSoft Administration Tools*.

The Page Assembly Process for Page-Based Templates

When a user is viewing a page-based template, and clicks a hyperlink on the page, a certain processing flow takes place to ensure that the user receives the requested content, *and* that the content is properly displayed within the context of the portal framework. It is the job of the portal servlet to intercept the user's request, retrieve the content, properly format it using a pre-defined portal template, and send the assembled page back to the user's browser. The portal servlet also ensures that user has the proper security permissions and is allowed to view the page.

The portal servlet uses two processes, called proxying and page assembly, to display content. Proxying is a process by which URL references in the HTML content are altered to refer to the portal servlet itself, providing the original reference in a querystring parameter.

The processing steps that occur during the page assembly process are as follows.

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

The browser sends an HTTP request for target content in the form of a URL. The receipt of the request at the portal web server invokes the portal servlet. The portal servlet identifies the target content by looking at the querystring parameter that was put there when the referring page was proxied.

2. The portal servlet checks the portal registry to see if the content reference has been registered there.

If the user does not have access to the content reference, the portal servlet responds with an error message to that effect. 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 is then responsible for getting the content for each frame in the frameset, in the usual way. If the content reference has been registered as having no template, the servlet simply 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.

Each content reference registered in the portal registry can have a portal template associated with it. Therefore, 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. Otherwise, the portal servlet uses the template for the content provider associated with the content reference. If there is no content provider associated with the content reference, then the default template for the portal is used.

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 template pagelet used in the template.

5. The portal servlet completes the page assembly process, and sends a complete HTML document back to the user's browser.

To complete the page assembly process, the portal servlet merges content from any HTML documents that it retrieved into the template HTML, proxying all the content. The completed page is then sent back to the user's web browser as a single HTML document. The template HTML may contain special PeopleSoft tags, including the TargetContent tag and the IClientComponent tag. Each TargetContent tag gets replaced with whatever content is retrieved when fulfilling the request for target content. And each IClientComponent tag is replaced with the proxied content specified by that tag.

If there is a stylesheet associated with the template, it will be used. If not, the stylesheet associated with the target content will be used. Stylesheets included in template pagelets will be 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.

Proxying

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. Proxying is required to ensure that familiar portal features, such as the universal navigation header, do not suddenly disappear when a user clicks on a link. In other words, proxying ensures that, when the user clicks on a portal page, the user is still in the portal.

When a user logs in to a PeopleSoft portal, they log in to a web server on which the portal servlet is running. It is the job of the portal servlet to ensure that all requests for content come through the portal. The portal servlet does this by processing all the HTML that it returns to the browser, converting all URL references so that they point to the portal web server, rather than the original URL. Of course, the original URL is still necessary to retrieve the requested content. So, the original URL 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 discussed below.

As an example, imagine that a user requests a page from an external web site 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 appropriate web server (whichever web server is associated with the requested page). It then 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 example, this is often done when a web page includes links to other content on the same web server that it is on. This situation works fine when a browser communicates directly with a web server, because there is no ambiguity about where the relative URL points to. 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 example below shows the both the original, "relative" version of an HTML tag, and the rewritten, "absolute" version created by the portal servlet.

Old Tag (Relative)

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

New Tag (Absolute)

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

Converting Anchor Tags

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 if the target URL should be “wrapped” with a template. Here is an example of a rewritten anchor tag.

Old Anchor Tag

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

New Anchor Tag

```
<a src=http://portalserver/servlets/psportal/peoplesoft8/?URL="http
%3a%2f%2fserver%2ftargetpage.html %3fAction%3dNew">
```

Converting Form Tags

As with 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. Here is an example of a rewritten form tag.

Old Form Tag

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

New Form Tag

```
<form action=http://portalserver/servlets/psportal>
  <input type=hidden name=URL>
    http://server/targetpage.html
```

</input>

Converting JavaScript

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

CHAPTER 4

Designing Portal Templates

One of the main goals of the PeopleSoft portal is to integrate content from numerous sources and present the merged content on a single web page in a coherent, consistent fashion that keeps users within the portal framework. When constructing this web page, the portal servlet depends on a *portal template* for information on what content to place on the assembled page, and where each piece of content should be placed.

A portal template is simply HTML code. The portal servlet interprets the portal template in much the same way as a browser interprets any HTML document. However, in addition to standard HTML tags, the portal template can include several special PeopleSoft tags used for giving instructions to the portal servlet during Page Assembly. A normal browser would not be able to interpret these special tags, but the portal servlet understands them, follows any instructions they contain, and then removes them before passing the final page back to the user's browser.

The content in a template falls into three categories:

- The HTML required for the template itself.
- The HTML of the main content ("target content").
- Any other HTML required (such as for the navigation header).

Template Types

There are several types of portal templates that you can design and use with the PeopleSoft portal. Each portal template will be either a page-based or a frame-based template, and either a static template or a dynamic template. Thus, there are four possible combinations of template types:

- Page-based static templates.
- Page-based dynamic templates.
- Frame-based static templates.
- Frame-based dynamic templates.

Page-Based and Frame-Based Templates

Every portal template is either page-based or frame-based. A page-based template is one which uses HTML tables to generate a page with content placed in each cell defined by the template.

With a page-based 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 different parts. The page assembly process used for page-based templates is not required. Instead, the portal servlet constructs the appropriate URL required for each frame, and sends the entire frameset to the browser. Each frame then retrieves its own content.

Static and Dynamic Templates

Every portal template is either static or dynamic. A static template is based on HTML code entered into the HTML area of a "template" content reference (a content reference that exists in the Base Portal Data\Templates folder of the portal registry). Bind variables cannot be used within the HTML area. The HTML includes any HTML required for the template itself, plus tags that specify the URLs for template pagelets. The contents of a static template can be viewed and updated in the Portal Administration pages.

A dynamic template is retrieved from a URL, rather than stored in the database with the content reference. Unlike a static template, a dynamic template can use bind variables. The HTML contents of the dynamic template cannot be viewed directly in the Portal Administration pages. Instead, the Portal Administration pages enable you to identify the URL which provides the template content. If the dynamic template is implemented by an iScript, you can use Application Designer to navigate to the record and field where the iScript exists. The contents provided by the URL is the HTML used for the dynamic template at runtime.

Template Examples

Example of a Static, Page-Based Template

Below is an example of a static template that combines a universal navigation header with some target content. It is based on HTML tables, not frames. This template is comprised 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 example below by all the text that is not either bold or underlined. This HTML code remains in the assembled page 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 **bold** in the example below. At runtime, the portal servlet will get the content specified by the IClientComponent's Source tag, and substitute it for that tag in the template for the final assembled page.
- **A template tag for target content.** At runtime, the portal servlet will replace the TargetContent tag—the second block of **bold** HTML code in the example below—with the main content that was requested content (whatever content the user requested by clicking a link or button).

```

<html>

<head>

</head>

<body>


<table>

<tr><td>


<IClientComponent Name="UniversalNavigation">

    <Source
Product="Portal">ICType=Script&ICScriptProgramName=WEBLIB_PORTTEST.PAGES.Fie
ldFormula.IScript_UniversalNavigation</Source>

</IClientComponent>


</td></tr>

<tr><td>


<TargetContent Name="TransactionContent"></TargetContent>


</td></tr>

</table>


</body>

</html>

```

Example of a Static, Frame-Based Template

Below is an example of a static template based on frames. In this example, we show the Content Reference registration page to give you a better idea of 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 makes use of HTML tables, the **Usage Type** would have been set to *HTML template*.

The screenshot shows the PeopleSoft Portal Administration interface. The breadcrumb trail is: Home > Menu > Portal Administration > Structure and Content > Base Portal Data > Templates >. The page title is 'Content Ref Administration'. The left sidebar contains a list of content references: PORTAL_ADMIN, PORTAL_STMP_HOMEPAGE, PORTAL_HP_DEFAULT_APP, PORTAL_HP_DEFAULT_ENT, PORTAL_ADMIN_NOSIDENAV, PortalNavigationTemplate, PORTAL_DEFAULT, PORTAL_RELATEDLINKS, and UniWrapperTemplate. The main content area is divided into two tabs: 'Content Ref Administration' (selected) and 'Content Reference Security'. The 'Content Ref Administration' tab shows the details for a template named 'PORTAL_STATIC_FRAME'. The 'Name' field is 'PORTAL_STATIC_FRAME' and the 'Label' is 'Static Frame Template'. The 'Long Description' is 'Sample Static Frame Template based on the PORTAL_TRANSACTION_DYN HTML object'. The 'Usage Type' is 'Frame template' and the 'Storage Type' is 'Local (in HTML Catalog)'. The 'Author' is 'PTDMO' and the 'Parent Folder' is 'Templates'. The 'Product' is empty, 'Valid from date' is '11/17/2000', 'Sequence number' is empty, 'Valid to date' is empty, and 'Creation Date' is '11/17/2000'. The 'Portal URL' is '<TEMPLATE>PORTAL_STATIC_FRAME'. The 'HTML Area' contains the following code:


```

<HTML>
<HEAD>
<TITLE>Transaction Frame Template Dynamic New</TITLE>
</HEAD>

<frameset rows="144,*">

  <FRAME name=UniversalHeader frameborder = no noresize src=>
    <IClientComponent Name="UniversalNavigation">
      <Source
        Product="Portal">iCType=Script&amp;iCScriptProgramName=WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula.iScript_UniHead
        er_Frame&amp;isFrame=true</Source>
      </IClientComponent>
    <FRAME name=TargetContent frameborder = no noresize src=>
      <TargetContent>X</TargetContent>
    </FRAMESET>
  </HTML>
    
```

 Below the HTML area is the 'Content Reference Attributes' section. It has fields for 'Name', 'Label', and 'Attribute value'. There is a 'Translate?' checkbox (checked), an 'Attribute Information' link, and a 'Delete' button. An 'Add' button is at the bottom of the section. At the bottom of the page, there is a 'Save' button and the text 'Content Ref Administration | Content Reference Security'.

A static, frame-based template

Example of a Dynamic, Frame-Based Template

All dynamic portal templates are retrieved from a URL, rather than a static 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 in the Portal Administration pages. However, instead of including specific HTML content, the dynamic template references a specific iScript. The iScript is associated with a specified field in a specified record.

In example below, we'll look at a template named PORTAL_DEFAULT in the Base Portal Data folder. It is a dynamic, frame-based template.

PEOPLE
Soft

Search: go

Home Menu Favorites Add to Favorites Help Sign out

Home > Menu > Portal Administration > Structure and Content > Base Portal Data > Templates

Siblings for Templates

[Pagelets](#)
[Users](#)
[Home Page](#)

Structure and Content

* Click the folder label to view the child folders and content references for that folder
* Click the "Edit" link to edit the folder definition

▼ Folders First 1 of 1 Last

Add Folder

* Click the "Edit" link to edit the content reference definition

▼ Content References First 1-9 of 9 Last

Label	Edit	Sequence number
PORTAL_ADMIN	Edit	Delete
PORTAL_STMP_HOMEPAGE	Edit	Delete
PORTAL_HP_DEFAULT_APP	Edit	Delete
PORTAL_HP_DEFAULT_ENT	Edit	Delete
PORTAL_ADMIN_NOSIDENAV	Edit	Delete
PortalNavigationTemplate	Edit	Delete
PORTAL_DEFAULT	Edit	Delete
PORTAL_RELATEDLINKS	Edit	Delete
UniWrapperTemplate	Edit	Delete

Add Content Reference

Save

Selecting a template

You can identify this as a dynamic template because, as shown in the example below, the **Content Ref Storage Type** is set to *Remote by URL*. This setting is required for dynamic templates. Additionally, you can see that there is no HTML area that displays the associated HTML code for the template, as you would expect with a static template. Instead, since the **URL Type** for this example is set to *iScript*, an **iScript Parameters** area appears, in which you can define the exact **Record Name**, **Field Name**, **PeopleCode Event Name**, and **PeopleCode Function Name** that specify the iScript you want to use for the template. The iScript that dynamically generates this particular template is found on the WEBLIB_PORTAL record on the PORTAL_NAV field. Additionally, the iScript function is associated with FieldFormula PeopleCode event. The specific PeopleCode function name is IScript_Portal_Trans_Dyn.

PEOPLE'Soft

Search:

Home Menu Favorites Add to Favorites Help Sign out

Home > Menu > Portal Administration > Structure and Content > Base Portal Data > Templates >

Sibling Content References

- PORTAL_ADMIN
- PORTAL_STMP_HOMEPAGE
- PORTAL_HP_DEFAULT_APP
- PORTAL_HP_DEFAULT_ENT
- PORTAL_ADMIN_NOSIDENAV
- PortalNavigationTemplate
- PORTAL_RELATEDLINKS
- UniWrapperTemplate

Content Ref Administration

Author: PTDMO

Parent Folder: Templates

*Name: DEFAULT_TEMPLATE

*Label: PORTAL_DEFAULT

Long Description: Portal default template (254 Characters)

Usage Type: Frame template

Storage Type: Remote by URL

URL Type: PeopleSoft Script

*Content Provider: Portal

Product: PRTL *Valid from date: 05/18/2000

Sequence number: Valid to date:

Creation Date: 05/18/2000

iScript Parameters

*Record (Table) Name: WEBLIB_PORTAL *Field Name: PORTAL_NAV

*PeopleCode Event Name: FieldFormula *PeopleCode Function Name: iScript_Portal_Trans_Dyn

Content Reference Attributes

Name: ☒ Translate? [Attribute Information](#)

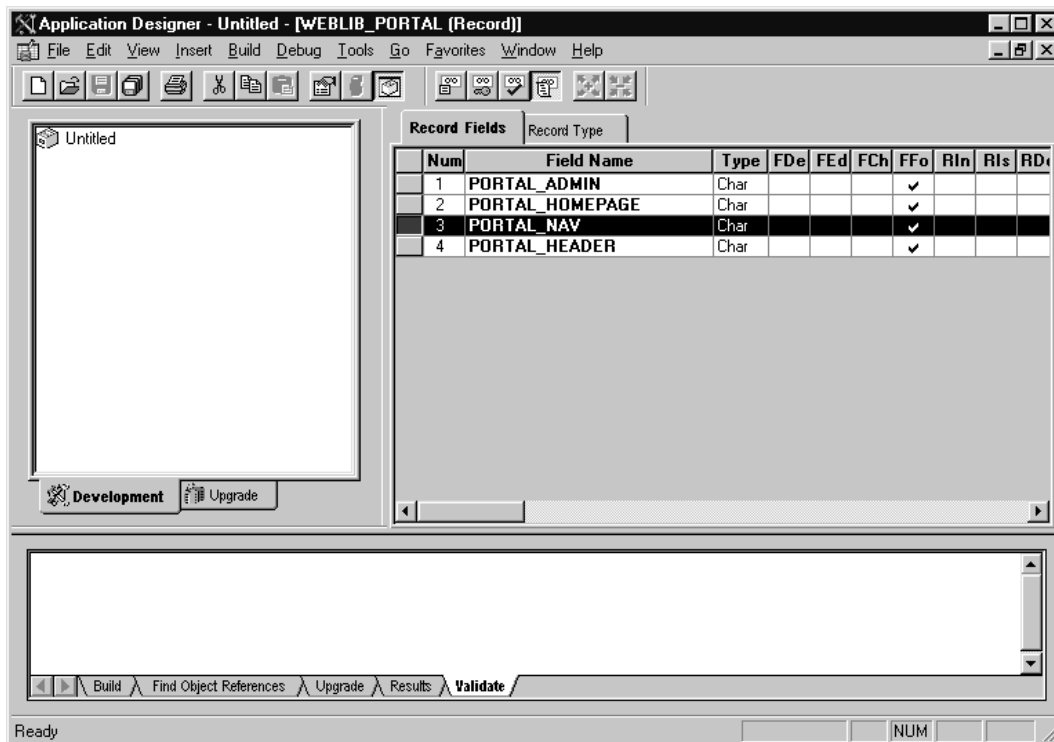
Label:

Attribute value:

Content Ref Administration | [Content Reference Security](#)

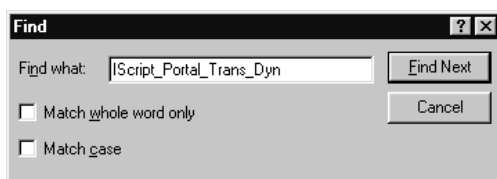
Defining a dynamic template in the Portal Administration pages

To view the iScript on which this dynamic template is based, open the record (WEBLIB_PORTAL) in Application Designer and select the appropriate field (PORTAL_NAV).



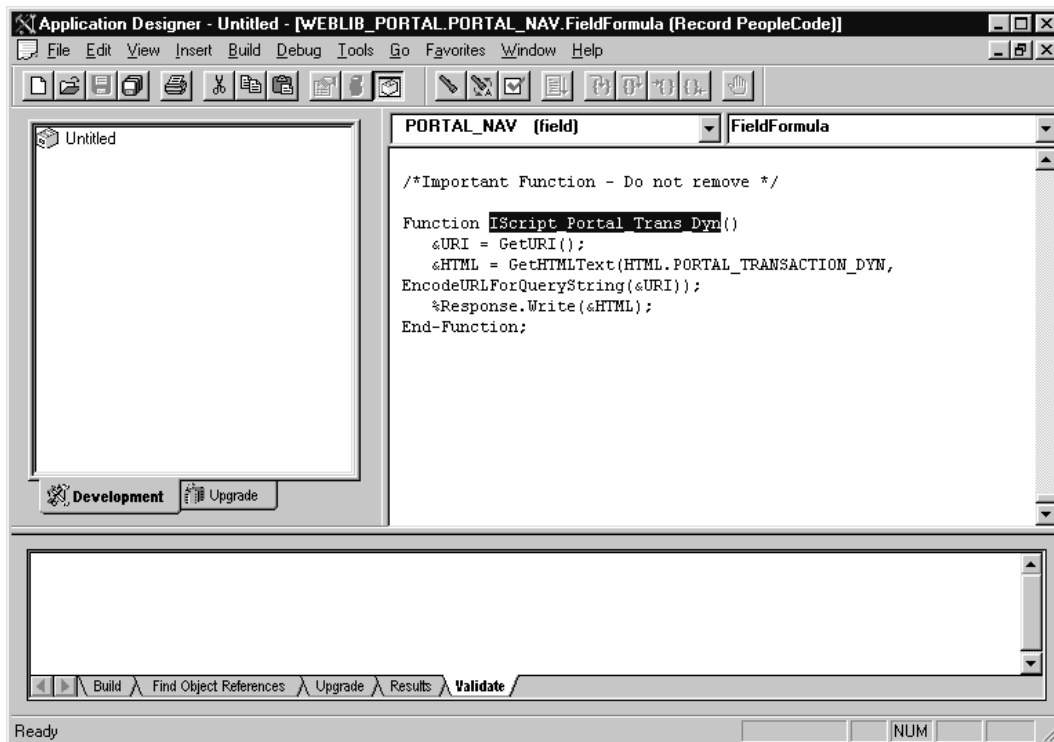
Opening the record and field associated with a dynamic template

Once you've opened the appropriate record and field (PORTAL_NAV) in Application Designer, you can then view the FieldFormula PeopleCode. You'll see the iScript referenced by the **PeopleCode Function Name** in the template's definition. The iScript name is iScript_Portal_Trans_Dyn. An easy way to find the exact reference is simply to copy the **PeopleCode Function Name** from the template definition and paste it into the **Find** dialog of the PeopleCode editor, once you've opened the corresponding record and field.



Finding the exact iScript name associated with a dynamic template

You now can see the code of the iScript referenced by the portal template. It is the function referenced in the Portal Administration pages: IScript_Portal_Trans_Dyn.



The iScript upon which the dynamic template is based

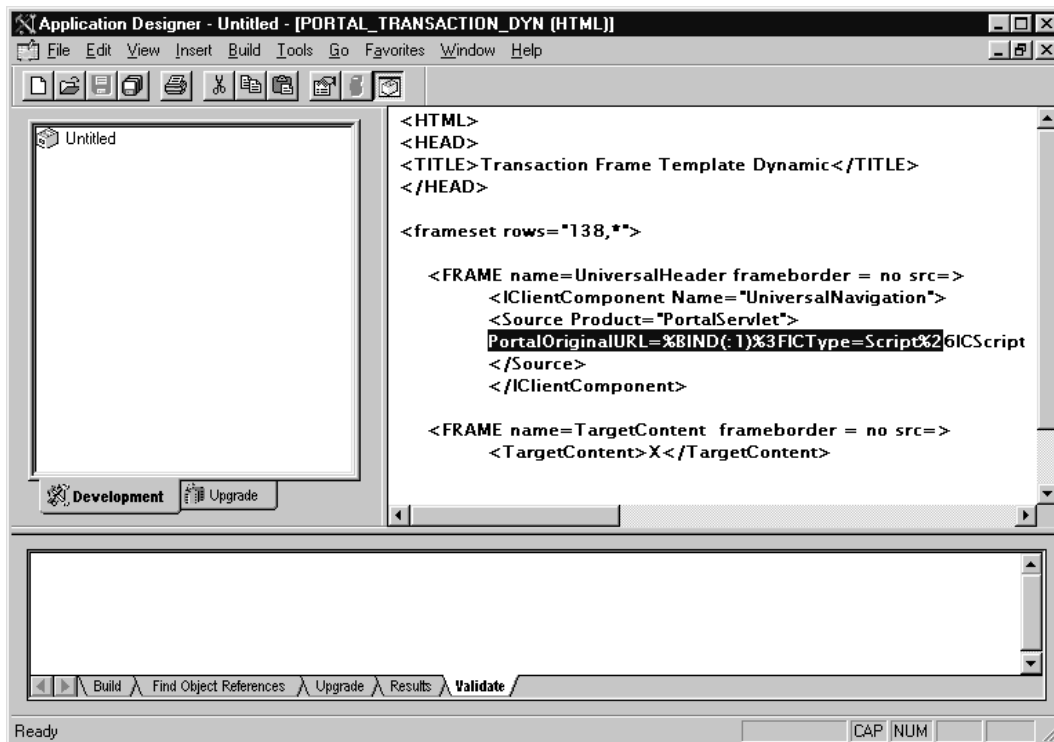
The following code calls an HTML object called PORTAL_TRANSACTION_DYN from the HTML catalog and passes it a bind variable (EncodeURLForQueryString(&URI)). The resulting HTML code forms the basis for the dynamic portal template at runtime.

```

    &HTML = GetHTMLText (HTML.PORTAL_TRANSACTION_DYN,
    EncodeURLForQueryString(&URI));

```

You can open the HTML object PORTAL_TRANSACTION_DYN in Application Designer, and see where the bind variable is passed in. Also note that this is a frame-based template, since FRAME tags are present.



Using bind variables with a dynamic template



For more information, see PeopleCode Reference

Developing Portal Templates

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, you need to add the query string parameter **&target=componentname** at the end of the query string parameter list for the template pagelet in the template HTML. This is to prevent the PeopleSoft Internet Architecture from assuming the same default name ("main") for each page. Failure to specify the target parameter will result in a JavaScript error when the portal servlet tries to load the template. If there is more than one form with the same name ("main") on the same page, the JavaScript on that page cannot determine which "main" component is the "main" component for the assembled page. Therefore, you should add the target parameter to the Source specification of each page-based template pagelet that is not meant to be the target content, in the template HTML.

Refer to the example below. Note the code in **bold**.

```
<tr>
```

```

<td>

    <iclientcomponent name="Related Links">

        <source
product="Portal">IType=Panel&Menu=PORTAL_ADMIN&Market=GBL&PanelGroupName=PORTAL
_RLNK_3COL&target=relatedlinks</source>

    </iclientcomponent> </td>

</tr>

```

Inheritance of Style Classes

When the portal servlet assembles portal pages based on Application Designer pages, the assembled pages inherit the style classes defined for them in Application Designer. It's possible for you to develop a template that includes pagelets built with different style sheets in Application Designer. Because you can combine different Application Designer pages on a portal page, it's possible to have conflicting style classes in your template. For example, if you have a style class named "Big" in page X, and you also have a style class named "Big" in page Y, there may be conflicting styles in the resulting page. Therefore, a simple rule of inheritance is used to determine which style sheet "wins" when conflicting style classes exist: When conflicting style classes exist, the style sheet associated with the page used for the target content will always take precedence over competing style sheets in the template.

Template Pagelet Tags

Template HTML can contain two Peoplesoft-specific tags that are used as portal servlet directives to assemble content into the page.

IClientComponent

- **Attributes:** Name (used to identify the component in a comment in final assembled page).
- **Elements:** Source tag.
- **Contents:** None.

Source

- **Attributes:** Product (used to identify which PIA web server to route to). The Product tag value should be name of the registered content provider for the content.
- **Elements:** None.
- **Contents:** Query string to access page/iScript implementing the component.
- **Notes:** Contents must be "escaped." Ampersands must be written as "&". The

IClientComponent tag can also be used to specify non-PeopleSoft Internet Architecture content. 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 contents, and the product attributes can be omitted.

TargetContent

- **Attributes:** Name (used to identify the component in a comment in final assembled page).
- **Elements:** None.
- **Contents:** None.

Considerations for Non-PeopleSoft Content Providers

One of the main challenges of the page assembly process is to figure out which template is associated with the requested content. The normal strategy is to do this by looking up the content reference associated with the target content using the portal registry API by the URL of the target content. The problem with this is that the same URL can supply different content pages, depending on the context. For example, with PeopleSoft application pages, the URL for submit buttons always looks the same. For example:

```
http://server/servlets/iclientservlet/peoplesoft8
```

It's reasonable to expect different templates based on different PeopleSoft application pages, but the portal servlet cannot figure out exactly which template to use based solely on the URL.

Because of this problem, the target content can specify its "registered URL" using a custom response header, "PortalRegisteredURL." When the target content supplies this response header, the portal servlet uses it to identify the URL registered in the portal, rather than using the actual URL used to get the content. All of the PeopleSoft Internet Architecture (PIA) technologies (pages, iScripts, queries, and so on) supply this URL.

In order for web content to be displayed in the correct template, it can include in its response a custom header named "PortalRegisteredURL." The value of this header must be the string with which the page should be registered. Note that it's up to the developer registering the content to register it with the identical string that it will return as this header.

If content does not return the header information, the portal servlet will identify the content by the URL that was in the request submitted to the portal.

Considerations for Frame-Based Templates

In frame-based templates, proxying does not occur automatically. Therefore, there are some special considerations, particularly if you want to create a link within a template pagelet that refers to another page assembled by the portal servlet. In essence, you need to construct a proxied link "by hand." In such cases, you should create the link using the "PortalServlet" content provider. Additionally, you must specify the URL for the target content in the "URL" parameter.

You must use `EncodeForQueryString()` to encode the value of this parameter. Refer to the following example:

```
&strHREF = EncodeURLForQueryString(Substring(&URL, &StartPos + 1, Len(&URL) -  
&StartPos));
```

```
&LINK = &Portal.GetQualifiedURL("PortalServlet", "URL=" | &strHREF);
```

The URL parameter value must always be an absolute URL. Using a relative URL will produce unpredictable results.

CHAPTER 5

Developing Pagelets

When a user first logs into a PeopleSoft portal they see a homepage. The homepage displays pagelets (small, rectangular pages). Which pagelets are displayed and where they are displayed depends on several factors, including the type of portal the customer has purchased, the permissions granted to the user, and how the user has customized his or her homepage. The objective of the pagelet is to provide the user with a display-only snapshot of PeopleSoft or non-PeopleSoft content.

Pagelets are simply “small” pages that follow a basic set of rules so that they can be displayed properly in a PeopleSoft portal homepage. The size of pagelets corresponds to the homepage layout specified by users. A user can customize the layout of their homepage so that it displays either a 2-column or a 3-column layout. In the 3-column layout, all three columns are “narrow” and of equal width. In the 2-column layout, there is one narrow column (1/3 of the display width) and one wide column (2/3 of the display width). Any pagelet you design must conform to the dimensions of the narrow column and—optionally—for the wide column. The only specific dimensions that must be adhered to are the width of the pagelet. The length of the pagelet can vary, but good design principles suggest keeping them as short as possible.

You will typically create pagelets in Application Designer just as you would create any other PIA page. However, you can also base the design of a pagelet on an iScript. This option is recommended only if, for some reason, it is not possible to accomplish the same task with a PIA page. This may be the case if you are developing a pagelet that uses “external” content (content that does not originate in a PeopleSoft application).

Designing Pagelets Based on PIA Pages

To design a page-based pagelet

1. Open a page in Application Designer.
2. Select **File, Object Properties** and click the **Use** tab.
3. Select the appropriate **Page Size**.
4. Design the pagelet using similar design techniques that you would use for a page.

For more information on designing pagelets, as opposed to pages, see “Pagelet Design Considerations” below.

5. Save the page.

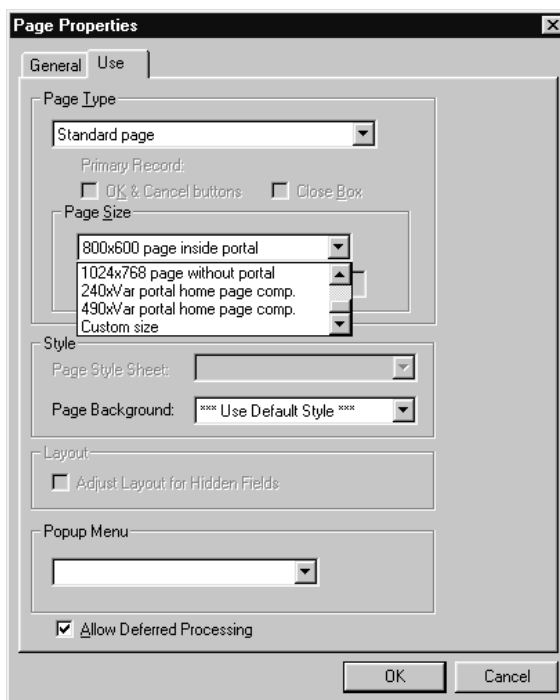
6. Register the pagelet in the portal registry.

Narrow and Wide Pagelets

You can create two sizes of pagelets:

- **Narrow:** You design “narrow” pagelets by selecting a page size of “240xVar” in the page properties in Application Designer. The pagelet itself is 240 pixels wide. Subtracting the border and the internal margin leaves 218 pixels for the content. This pagelet size provides a predefined width, but no explicit height.
- **Wide:** You design “wide” pagelets by selecting a page size of “490xVar” in the page properties in Application Designer. The pagelet itself is 490 pixels wide. Subtracting the border and the internal margin leaves 468 pixels for the content. This pagelet size provides a predefined width, but no explicit height.

In addition to these page sizes, you can also select the “standard” portal page size. This is done by selecting a page size of “800x600 page inside portal” in the page properties in Application Designer. Unlike the small pagelets, this page size provides both a predefined width *and* a predefined height. However, if necessary the page height can be extended to form a “long page.”



Setting the page size on the Page Properties dialog



For more information, see Application Designer

Narrow Pagelet

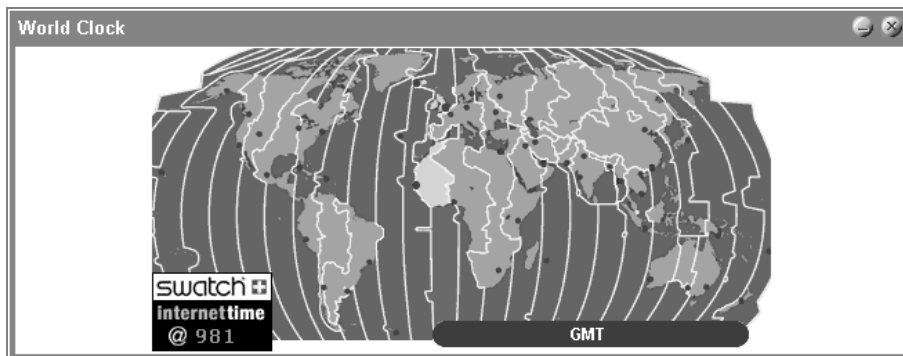
A “narrow” pagelet will display in a 1/3 column of the portal's homepage. It is for this reason that the amount of key data will be limited in the narrow pagelet. The objective of the narrow pagelet is to provide the user with a succinct and clear list of values they can quickly traverse. You should select the minimum important pieces of data that would best encapsulate what the user is to review. A narrow pagelet will only fit a single 30-character field. Use wrapping in fields to accommodate more than one field in a grid.



Example of a narrow pagelet

Wide Pagelet

A “wide” version of the pagelet is optional. This will display in the 2/3 column of the portal's homepage. Define the page size field to use "490xVarPortal Homepage Component". The objective of the wide pagelet is to, again, provide the user with a succinct and clear list of values they can quickly traverse. However, the wide pagelet gives you more freedom because the wider pagelet can accommodate more columns of data. Keep in mind that even though the user's visibility will increase, the data must remain meaningful.



Example of a wide pagelet



There is a request parameter sent to the pagelet when it is requested that indicates if the pagelet is going to be in a narrow or wide column. This allows you to build one component with two page sizes—narrow and wide. On the “load” event, PeopleCode can then go to the appropriately sized page. See the PeopleCode documentation for more information.

Designing Pagelets Based on iScripts

Cases may arise where it is not convenient to base a pagelet on a page designed in Application Designer. In these cases, you will want to use an iScript instead of a page for your pagelet design. Using an iScript is advised only for pagelets that reference non-PeopleSoft data or if a PIA page does not provide the necessary functionality.

If you develop a pagelet based on an iScript, you should understand that some of the functionality provided by Application Designer will not be automatically included in your iScript, and thus becomes your responsibility to develop. Currency codes, language support, and multiple browser support are examples of areas that become your responsibility.

Developing a Pagelet Based on an iScript

Developing basic pagelets based on iScripts is not difficult. As with any development, some tasks require more coding than others, but creating a basic pagelet from an iScript is straightforward. The important thing to note is that there are several conventions to be followed. This section addresses these conventions. If you are comfortable developing with HTML, Active Server Pages (ASP), or JavaScript, you'll find that using iScripts is quite similar to such environments. Keep in mind, however, that this section does not attempt to teach you how to code iScripts; it merely explains how to develop a pagelet based on an iScript.

This section provides an example of developing an iScript-based pagelet that uses content from an external site for display in the PeopleSoft portal. Here's a summary of what occurs in the following procedure: We begin by copying the relevant HTML code from an external source and paste it into a new HTML object in Application Designer. We then update an existing web library with an iScript that references the new HTML object. After the iScript is created, we register the new pagelet by defining the appropriate iScript name in the portal registry. Finally, we test that the pagelet works as expected.

In the end we'll have a "dictionary search" pagelet that resembles the following. The pagelet is based on a public-use form provided at www.onelook.com.



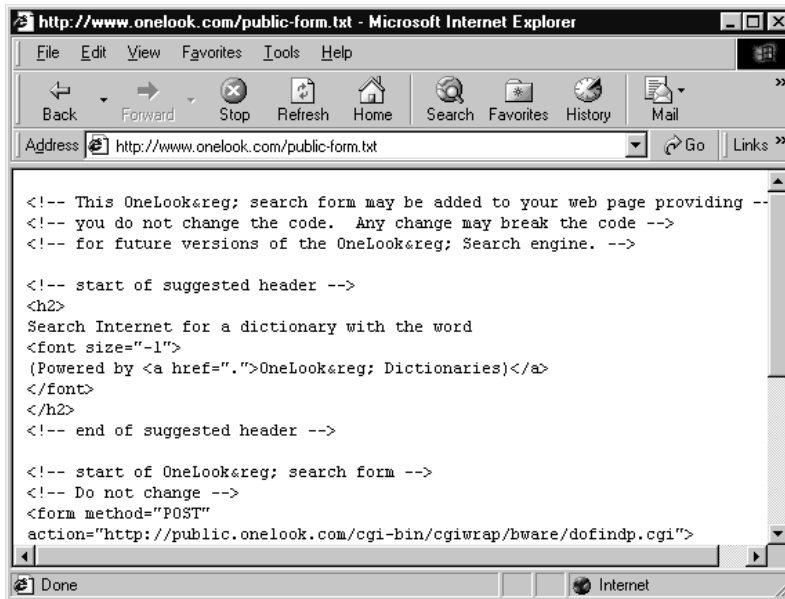
Example of a pagelet based on an iScript

Remember, the following procedure assumes that you have a basic understanding of web libraries and iScripts.

To create a pagelet based on an iScript

1. Create the HTML code you want to use for the pagelet.

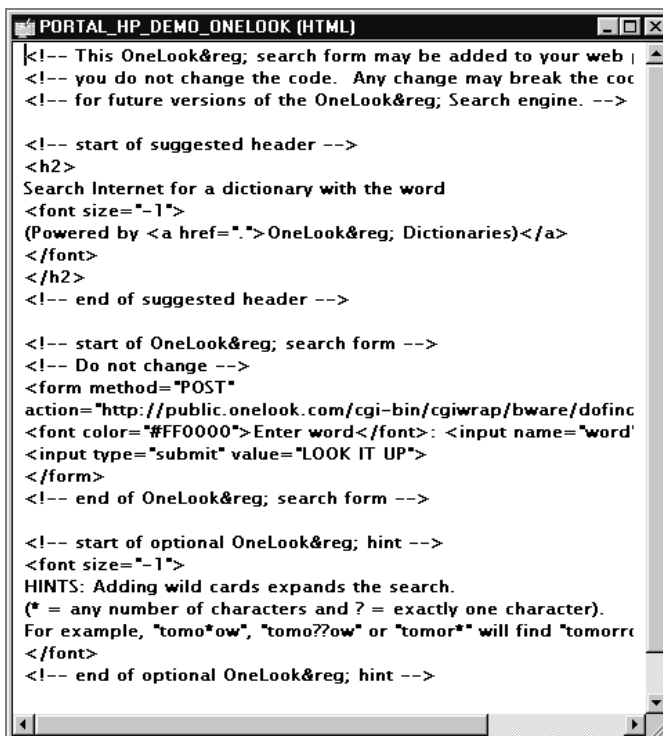
In many cases, you can simply navigate to the URL that you want to turn into a pagelet, and then copy the HTML associated with that web page, as we've done below. This is simpler than creating the HTML from scratch. For form-based web pages, you'll find that you can simply copy all of the HTML code between the <FORM> tags (all the HTML beginning and ending with the form tags) from an existing HTML document.



The HTML code used for a pagelet based on an iScript

2. Store the new HTML code as an HTML object in Application Designer.

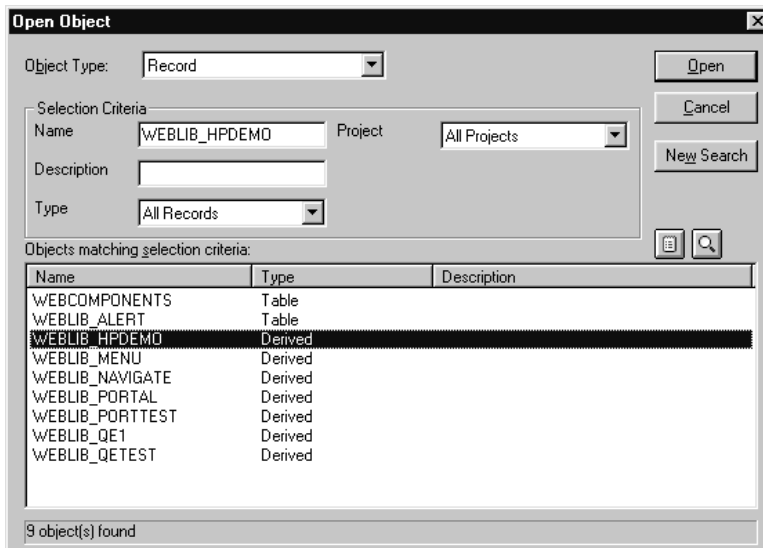
Paste the HTML from the previous step into a new HTML object in Application Designer and save the HTML object.



Pasting HTML code into Application Designer for use with an iScript

3. Within a web library, create an iScript that calls the new HTML object.

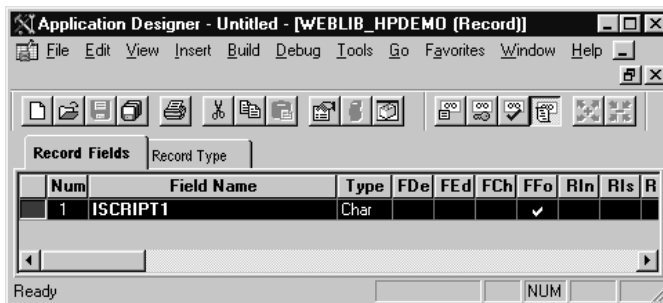
A web library is nothing more than a derived record created in PeopleTools with express purpose of storing iScripts. The name of all web library records begin with “WEBLIB_”. In our example, we will open a web library named WEBLIB_HPDEMO. We will then create a new iScript within this web library that calls the OneLook dictionary search page.



Placing an iScript in a web library

Once you’ve opened the appropriate web library, you must navigate to the appropriate field name and edit the FieldFormula PeopleCode associated with the field. This is where you’ll create an iScript that calls the HTML object you created in the previous step.

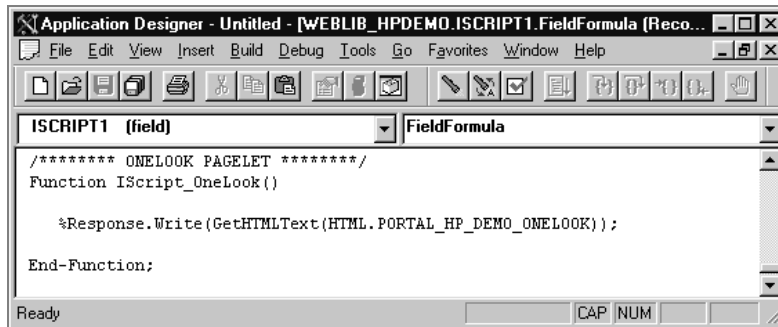
In our example, we’ll open the ISCRIP1 field in the WEBLIB_HPDEMO record.



Selecting a field in which to place an iScript

Once we’ve selected the appropriate field, the PeopleCode editor will appear, and we’ll create a new iScript and give it a descriptive name. All iScript names must begin with “iScript_”. We’ll name ours “iScript_OneLook.” (An easy way to begin a new iScript is to copy and paste an existing iScript in the web library, and then make appropriate changes to it.)

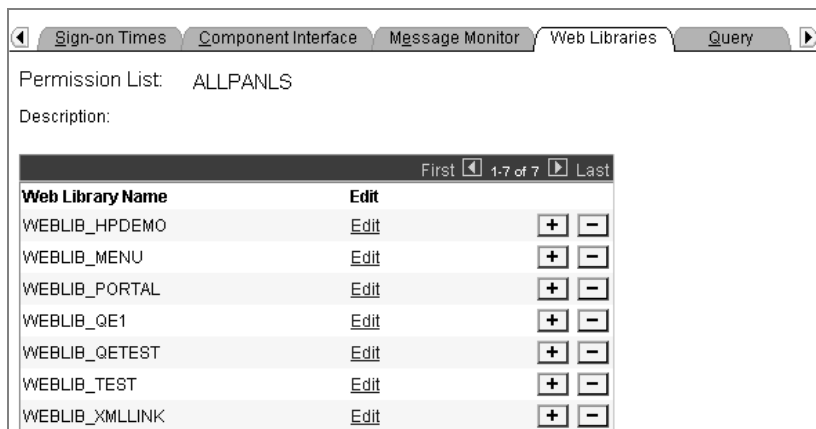
Below is an example of what our new iScript looks like. Note that `GetHTMLText()` is used to retrieve the HTML object we previously created, and then `%Response.Write` is used to display the results of the HTML. This is the essence of how the pagelet is created. After creating the new iScript, save the web library.



Viewing an iScript in Application Designer

4. Enable security for the new iScript.

Select **Maintain Security, Use, Permission Lists** and open the ALLPANLS permission list. Select the **Web Libraries** tab, and click the **Edit** link next to the appropriate web library.



Selecting a web library on which permissions will be set

In keeping with our example, we'll select the WEBLIB_HPDEMO web library to edit. We'll then grant full access to the OneLook iScript and save the permission list, as shown below.

Home > Menu > PeopleTools > Maintain Security > Use > Permission Lists

Weblib Permissions (48,191)

WEBLIB_HPDEMO

Function	*Access Permissions
ISCRIP11.FieldFormula.IScript_BART	Full Access
ISCRIP11.FieldFormula.IScript_Calculator	Full Access
ISCRIP11.FieldFormula.IScript_Calendar	Full Access
ISCRIP11.FieldFormula.IScript_CurrConvert	Full Access
ISCRIP11.FieldFormula.IScript_Dictionary	Full Access
ISCRIP11.FieldFormula.IScript_ExciteNews	Full Access
ISCRIP11.FieldFormula.IScript_PTObalance	Full Access
ISCRIP11.FieldFormula.IScript_TechnologyNews	Full Access
ISCRIP11.FieldFormula.IScript_WorldClock	Full Access
ISCRIP11.FieldFormula.IScript_YahooStockQuote	Full Access
ISCRIP11.FieldFormula.IScript_ZAGAT	Full Access
ISCRIP11.FieldFormula.IScript_OneLook	Full Access

First 1-12 of 12 Last

Full Access (All)

No Access (All)

OK Cancel

Setting permissions for a web library

5. Register the new pagelet in the portal registry.

The new pagelet has now been created with an iScript and users have been granted access to the iScript. However, the pagelet cannot be accessed through the portal until it is registered in the portal registry. To register the new pagelet, select **Menu, Portal Administration, Structure and Content**. Then select **Base Portal Data, Pagelets** and click on the name of the folder in which you want to add the new pagelet. (We'll select the "Miscellaneous" folder for the OneLook pagelet.) Select **Add Content Reference**, and define the Name and Long Description (we'll enter "OneLook" for both of these). Set the **Usage Type** to *Pagelet*, and the **URL Type** to *PeopleSoft Script*, and the **Content Provider** to *Portal*. You then must set the **iScript Parameters** to point to the appropriate iScript. Note that the iScript parameters shown in the example below correspond to the record name, field name, PeopleCode event name, and PeopleCode function name we used to create the iScript.

Home > Menu > Portal Administration > Structure and Content > Base Portal Data > Pagelets > Miscellaneous >

Content Ref Administration **Content Reference Security**

Content Ref Administration Author: PTDMO Parent Folder: Miscellaneous

*Name: ONELOOK
 *Label: OneLook
 Long Description: OneLook Dictionary (254 Characters)
 Usage Type: Pagelet Product: Valid from date: 11/16/2000
 Storage Type: Remote by URL Sequence number: Valid to date:
 URL Type: PeopleSoft Script Creation Date: 11/16/2000
 *Content Provider: Portal

iScript Parameters
 *Record (Table) Name: WEBLIB_HPDEMO *Field Name: ISCRIPT1
 *PeopleCode Event Name: FieldFormula *PeopleCode Function Name: IScript_OneLook

Pagelet Attributes
 Default Column: Column 1 Content provider name:
 Edit URL:

Content Reference Attributes
 Name: Translate? Attribute Information Delete
 Label:
 Attribute value:
 Add

Save
 Content Ref Administration | Content Reference Security

Defining a content reference to use an iScript-based pagelet

This step includes only the information that is pertinent to registering the pagelet in this example.

6. Test the new pagelet in the portal.

Storing iScripts

All iScripts should be coded in the Field Formula PeopleCode event of a derived/work record called WEBLIB_XX, where XX is the product code. Only functions that start with "iScript" (such as iScript_IntegratedTasklist) can be invoked from a URL.

Style Sheets and Styles

Developers using iScripts should always use the styles (also known as classes) defined in Application Designer style sheets to specify the attributes of objects referenced in iScripts.

Pagelet Design Considerations

When you develop a pagelet, you should assume that untrained users will be accessing it. Therefore, the primary design goal should be ease of use. The use of pagelets should require no training. Pagelets should allow shortcut navigation and quick display of frequently accessed business applications, as well as common types of pagelets that display external content like news or weather.

General Design Considerations

Here are some basic guidelines you should consider when developing pagelets.

- **Use the default style sheets.** Unless otherwise required, you should always use the “Use Default Style Sheet” option when creating the page and the "Use Default Style" option for any controls on the page. The style sheet rendered by the portal follows the following order of precedence.

1. Template
2. Target Component
3. Template Pagelet

The portal first searches for the style sheet class in the style sheet defined for the template. If the style sheet class is found in the template’s style sheet, then every occurrence of that style sheet class rendered by the portal servlet will use the template’s style sheet definition for that style sheet class. If the style sheet class is not defined in the template’s style sheet, then the portal servlet searches for it in the target component’s defined style sheet. The key point to note is that the portal servlet applies the first style sheet class definition found to every occurrence using a given style sheet class.

All pagelets are rendered using the defined template for the specific portal’s homepage. To achieve a different look and feel for different portal homepages (such as the Workforce portal versus the Customer portal), the template associated with these homepages should be different. Just assigning a different style sheet to a group of pagelets is not sufficient.

- **Set your display size to 800x600.** PeopleSoft 8 applications are designed and developed for an 800x600 monitor resolution. All portal pagelets should be designed and tested using this resolution.
- **Use the appropriate page size.** Portal pagelets fall into two page sizes—narrow and wide. Select an appropriate size.
- **Avoid horizontal scrolling.** This is true with pagelets, just as with full-sized application pages. Numerous usability studies have shown that applications should avoid horizontal scrolling, regardless of the type of user.
- Adhere to object naming standards.
- **Add all pagelets to the PORTAL_COMPONENTS menu on the homepage bar item.** This

same menu exists in each application database. However, the pagelets on this menu vary according to the pagelets specific to the application database. Consequently, the menu contents will be different for each different database. These menus should not be synchronized.

- **Do not use help text.** There should be no help text for pagelets. The help on the homepage is for help in homepage usage, not for individual pagelets. Try to create pagelets with ease of use in mind, so that help text is not required.
- **Do not use search keys or defaulted search key values.** All pagelets and template pagelets must use a search record with no search keys or defaulted search key values. That is, the page must come up without requiring any search values. The most common search record is the INSTALLATION record. If the user ID is needed for the pagelet or template pagelet, then an alternative such as OPRID_VW can be used (a view of PSOPRDEFN as used in ERP).
- **Do not use page titles for pagelets.** (The name of this content reference in the portal registry is used as a default for the pagelet's title.) Pagelet instructions should not be necessary.
- **Use "Unlimited occurs" on all grids and scroll areas.** Control the number of retrieved rows with the PeopleCode function StopFetching() in the RowSelect event of the selected record. Match the StopFetching() when the user specified or defaulted value of the 'Maximum Displayed Number of Rows' is reached. Pagelets should display no more than five rows by default.
- **Use deferred processing.** Deferred processing on pagelets is not an issue because any FieldChange/Prompts/FieldEdits other than pushbutton/hyperlinks page transfers are not allowed for pagelets. These types of processing require a trip to the application server, and the portal has been designed to redisplay any pagelet as a target page (wrapped in the default template) after any trips to the server. This means that such processing would result in the pagelet being displayed outside the context of the homepage. This is why deferred processing is required for pagelets.
- **Do not provide any filter, sort, or refresh buttons.** The reason for this is the same as for using deferred processing above.
- **View internet options in Application Designer.** Selecting **View, Internet Options** ensures that you have access to all the styles available.
- **Leave more than 20% spacing between field labels and field edit boxes.** The styles are rendered larger in the browser than they appear in Application Designer.
- **View your pagelet in Application Designer before bringing it into the portal.** To do this, select **Layout, View in Browser** in Application Designer. This gives you an opportunity to see if the pagelet is designed properly before you bring it into the portal environment.
- Do not overlap group boxes.
- Left-align field labels and field edit boxes.
- Uncheck "Show Borders" on group boxes and scroll areas.

CHAPTER 6

Using Portal Administration Features

Portal administration refers to the management of objects in the portal registry. These objects include folders, content references, templates, and content providers. Common administrative tasks include adding, deleting, and renaming portal objects. Additionally, there are many properties associated with every portal object, and all of these properties can be accessed and modified by an appointed portal administrator who has been given appropriate permissions through PeopleSoft Security.

There are two ways to update the portal registry. You can update the registry manually, through the Portal Administration pages, or programmatically, through the portal registry API. The tool you select will depend on the type of updates required, and your organization will likely find uses for both methods of updating the portal registry. This section focuses on the use of the Portal Administration pages. Of course, you should be familiar with the design of the portal registry before attempting to manage its contents.

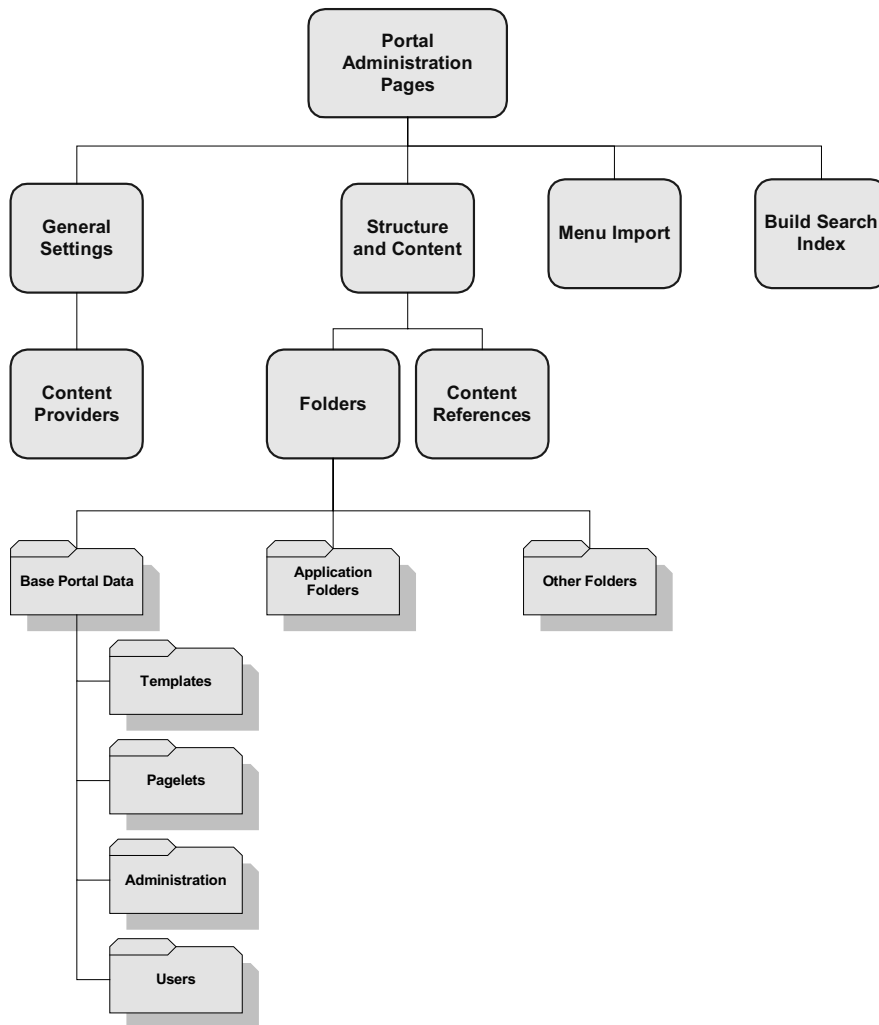
Managing Portals

From the Portal Administration pages, you can accomplish a number of administrative tasks for any installed portal:

- **Manage the structure and content of a portal.** This refers to adding, editing, and deleting folders and content references.
- **Manage general settings for a portal.** This option enables you to enter a description for an entire portal, and to set the default template. You can also add, edit, and delete content providers for the portal.
- **Import menu definitions into a portal registry.** This is a simple way to populate a registry using existing PeopleSoft menu definitions.
- **Build a Search Index.** This is an easy way to generate a Verity search index (or “collection”) for use with your portal, based on the contents of the portal registry.

To access the Portal Administration pages, you must have been given the appropriate permissions in PeopleSoft Security.

The following diagram shows a map of the Portal Administration pages, corresponding to the bulleted items above. Note the location of the special “Base Portal Data” folder. This folder contains folders and content references for all templates, pagelets, and users associated with a single portal.

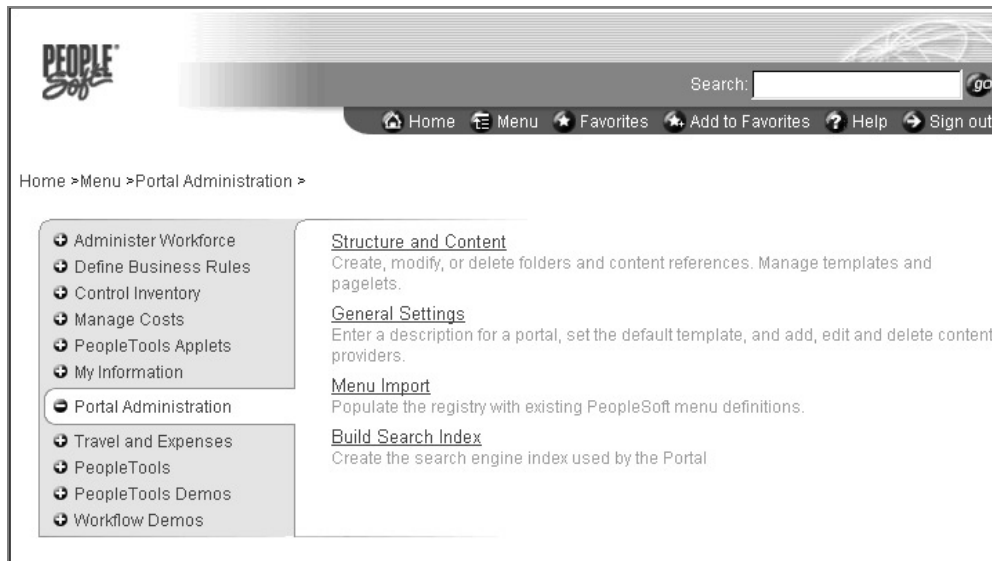


The Portal Administration pages

To use the Portal Administration pages

1. Click the **Menu** button, and then select **Portal Administration**.

The following menu appears. (Note that if you have not been given appropriate permissions in Maintain Security, the **Portal Administration** option will not be visible, and you will not have access to the Portal Administration pages).



Navigating to the Portal Administration pages

2. Select the administration page you want to use.

The options are Structure and Content, General Settings, Menu Import, and Build Search Index.

Managing Folders and Content References

Managing Folders

To edit, add, or delete an existing folder

1. Navigate to the Portal Administration pages.

You can do so by clicking the **Menu** button in the universal navigation header, and then selecting **Portal Administration**.

2. Select **Structure and Content**.

A page resembling the following appears. You can collapse the list of folders by clicking **► Folders**.

Home > Menu > Portal Administration > Structure and Content

Root Child Folders

[Base Portal Data](#)
[Administer Workforce](#)
[Define Business Rules](#)
[Control Inventory](#)
[Manage Costs](#)
[PeopleTools Applets](#)
[My Information](#)
[Portal Administration](#)
[Travel and Expenses](#)
[PeopleTools](#)
[PeopleTools Demos](#)
[Workflow Demos](#)

Structure and Content

* Click the folder label to view the child folders and content references for that folder
 * Click the "Edit" link to edit the folder definition

Label	Edit	Sequence number	
Base Portal Data	Edit		Delete
Administer Workforce	Edit	1	Delete
Define Business Rules	Edit	2	Delete
Control Inventory	Edit	3	Delete
Manage Costs	Edit	4	Delete
PeopleTools Applets	Edit	5	Delete
My Information	Edit	100	Delete
Portal Administration	Edit	9900	Delete
Travel and Expenses	Edit	9999	Delete
PeopleTools	Edit	9999	Delete
PeopleTools Demos	Edit	9999	Delete
Workflow Demos	Edit	9999	Delete

[Add Folder](#)

* Click the "Edit" link to edit the content reference definition

Content References

[Add Content Reference](#)

[Save](#)

Navigating to a folder

3. Continue navigating to the folder you want by “drilling down” through the folder names.

Each time you click a folder name, you “drill down” to the next level of folders. Continue drilling down until the folder that you’re looking for shows up in the list of folders.

4. Select the appropriate action (add, delete, or edit).

To delete an existing folder, click the **Delete** button next to the folder you want to delete. A confirmation dialog appears. Click **OK** only if you’re sure that you want to delete the folder. Deleting a folder has global impact on your portal. Click **Cancel** if you’re not sure if you want to delete the content provider.



All child folders and child content references will be deleted when you delete the selected folder. You must make absolutely sure that no critical dependencies exist for these objects before deleting their parent folder.

To add a folder, click the **Add Folder** link. To edit an existing folder, click the **Edit** link to the right of the folder name you want to edit. A page resembling the following appears. Note that the bar on the left side of the page provides navigation to sibling folders.

The screenshot shows the PeopleTools 8.12 Portal Administration interface. The top navigation bar includes links for Home, Menu, Favorites, Add to Favorites, Help, and Sign out. The breadcrumb trail indicates the current location: Home > Menu > Portal Administration > Structure and Content >.

The left sidebar contains a list of links under the heading "Sibling Folders":

- Base Portal Data
- Administer Workforce
- Define Business Rules
- Control Inventory
- Manage Costs
- PeopleTools Applets
- My Information
- Portal Administration
- Travel and Expenses
- PeopleTools
- PeopleTools Demos
- Workflow Demos

The main content area is titled "Folder Administration" and includes a "Folder Security" tab. The form contains the following fields:

- Name:** NEWFOLDER
- Label:** New Folder
- Long Description:** New Folder Example (254 Characters)
- Parent Folder:** Root
- Product:** (empty)
- Valid from date:** 11/16/2000
- Creation Date:** 11/16/2000
- Sequence number:** (empty)
- Valid to date:** (empty)
- Author:** PTDMO
- ☐ **Hide from portal navigation?**

Below the main form is a section titled "Folder Attributes" with the following fields:

- Name:** (empty)
- Label:** (empty)
- Attribute value:** (empty)
- ☒ **Translate?**
-
-

At the bottom of the form, there is a "Save" button and a link to "Folder Security".

Changing a folder's properties

5. Define the folder **Name** and **Long Description**.

Enter a **Name**, **Label**, and **Long Description** for the folder. A name is required, but the label and description are optional. The name will appear as the link in the browser.

6. Define the **Valid from date** and **Valid to date** if your application will use them.

The **Valid from date** is optional, and defaults to the same date as the **Creation Date**. The **Valid to date** is also optional, and is blank by default. These dates may be used for application-specific processing by portal-aware applications.

7. Enter the **Product** with which this folder is associated if your application will use this field.

Entering a **Product** is optional. Portal-aware applications can use this field for group processing.

8. Change the **Sequence number** of the folder as desired.

The **Sequence number** determines the order in which this folder appears in the current folder collection when viewed by users. If you're creating a new folder, you can leave this field blank and it will automatically take the highest sequence number, so that it will appear last. You can only change the sequence of a single folder at a time, and each time you do so, you must save your changes to see the results.

9. Enable the **Hide from portal navigation** checkbox if desired.

This option makes this particular folder invisible to users.

10. Define **Attribute Names, **Labels**, and **Attribute values** for the folder.**

These optional names, labels, and values may be used by portal-aware applications for application-specific processing. If you enable the **Translate** option, then the **Label** and **Attribute values** are translatable.

11. Click the **Add Folder link if you want to add a new folder as a child of the current folder.**

If you click **Add Folder**, a new Folder page appears. Define any child folders as required.

12. Click **Save.**

After you save a folder, you must set security permissions for it. Follow the procedure below.

To set permissions for a folder

1. From the Folder definition page, click the **Folder Security tab.**

The Folder Security page appears.

The screenshot shows the 'Folder Security' page in the PeopleTools portal. The breadcrumb trail is 'Home > Menu > Portal Administration > Structure and Content >'. The left sidebar lists various portal administration tasks. The main content area has two tabs: 'Folder Administration' and 'Folder Security', with 'Folder Security' being the active tab. The 'Folder Security' section includes a 'Label' field with the value 'New Folder'. Below it is the 'Access Type' section with two radio buttons: 'Permission List' (selected) and 'Public'. The 'Author Access' section has a checked checkbox. The 'Permissions' section shows a table with one row containing 'Permission List' and a 'Cascade?' checkbox. The 'Inherited Permissions' section also shows a table with one row containing 'Permission List'. At the bottom, there is a 'Save' button and a breadcrumb trail 'Folder Administration | Folder Security'.

Setting permissions for a folder

2. Set the **Access Type to either **Permission List** or **Public**.**

Typically, you will select **Permission List** and then define which permission list(s) will have access to this folder. You can select **Public** if you'd like to make the folder available to all users without manually adding each permission list.

3. Select whether to enable **Author Access.**

Enabling this option means that you (the currently logged on user and the author of the content reference) will be able to access the folder in the future, regardless of how other permissions are set.

4. Select the **Permission List** (or lists) that will have access to this folder.

This step only applies if you set the **Access Type** to **Permission List**. You can click the lookup button to see a list of permission lists. To give additional permission lists access to this folder, click **Add** and select additional permission lists. To remove access to this folder from a permission list, click **Delete** next to the associated permission list.

Note that **Inherited Permissions** are read only because they show you permissions that have been inherited from folders above. You can change them by going to the folder that defined the cascading permission.

5. Click **Save**.

None of your changes to the folder are final until you save them.

6. Test your changes in the portal.

After creating the folder and setting up permissions, test the folder reference by clicking **Home** and navigating to it.



For more information, see Security

Managing Content References

To edit, add, or delete a content reference

1. Navigate to the Portal Administration pages.

You can do so by clicking the **Menu** button in the universal navigation header, and then selecting **Portal Administration**.

2. Select **Structure and Content**.

A page resembling the following appears. You can collapse the list of content references by clicking ► **Content References**.

Home > Menu > Portal Administration > Structure and Content > Base Portal Data

Structure and Content

* Click the folder label to view the child folders and content references for that folder
 * Click the "Edit" link to edit the folder definition

▼ Folders		
Label	Edit	Sequence number
Templates	Edit	Delete
Pagelets	Edit	Delete
Users	Edit	Delete
Home Page	Edit	Delete

[Add Folder](#)

* Click the "Edit" link to edit the content reference definition

▼ Content References		
Label	Edit	Sequence number
PORTAL_HOME	Edit	Delete
Portal Expire	Edit	Delete

[Add Content Reference](#)

[Save](#)

Navigating to a content reference

3. Continue navigating to the content reference you want by “drilling down” through the folder names.

Each time you click a folder name, you “drill down” to the next level of folders. Continue drilling down until you reach the folder in which you want to add, delete, or edit a content reference.

4. Select the appropriate action (add, delete, or edit).

To delete an existing content reference, click the **Delete** button next to the content reference you want to delete. A confirmation dialog appears. Click **OK** only if you’re sure that you want to delete the content reference. Click **Cancel** if you’re not sure if you want to delete the content reference.



Do not delete the PORTAL_HOME content reference. This is the default content reference, which has special significance to your portal. Changing this setting could make your portal inoperative.

To add a content reference, click the **Add Content Reference** link. To edit an existing content reference, click the **Edit** link to the right of the content reference name you want to edit. A page resembling the following appears. Note that the bar on the left side of the page provides navigation to sibling content references.

Home > Menu > Portal Administration > Structure and Content > Administer Workforce > Administer Workforce (GBL) > Use >

Content Ref Administration

Author: PTDMO
Parent Folder: Use [Select New Parent Folder](#)

*Name: PERSONAL_DATA1
*Label: Personal Data
Long Description: (254 Characters)
Usage Type: Target
Storage Type: Remote by URL
URL Type: PeopleSoft Component
Template Name: DEFAULT_TEMPLATE
*Content Provider: Portal

Product: DEMO *Valid from date: 10/24/2000
Sequence number: 4 Valid to date:
Creation Date: 10/24/2000
☐ No Template

[Add Content Reference](#) [Test Content Reference](#)

Component Parameters

*Menu Name: ADMINISTER_WORKFORCE_GBL *Market: GBL *Component: PERSONAL_DATA1

☐ Hide from portal navigation?

Content Reference Attributes

Name: ☒ Translate? [Attribute Information](#) [Delete](#)
Label:
Attribute value:
[Add](#)

[Save](#)

Content Ref Administration | [Content Reference Security](#)

Changing the properties of a content reference

5. Define the **Name** and **Long Description**.

Enter a **Name**, **Label**, and **Long Description** for the content reference. A name is required, but the label and description are optional. The name will appear as the link in the browser.

6. Define the **Usage Type**, **Storage Type**, and **URL Type**.

You can set the **Usage Type** to to *Frame template*, *HTML template*, *Homepage template*, *Pagelet*, *Target*, or *Template Component* (a *template component* is just a *template pagelet*). For a typical PeopleSoft application page, you'll set the **Usage Type** to *Target* and ensure that the **URL Type** is set to *PeopleSoft Component*.

The **Storage Type** can be set to either *Remote by URL* or *Local (in HTML Catalog)*. The availability of both of these options depends on the **Usage Type** selected.

If the **Usage Type** is set to *Pagelet*, you'll see the **Pagelet Attributes** area appear. Set the **Default Column**, **Content provider name**, and **Edit URL** as appropriate. Pagelets must be defined in the Base Portal Data\Pagelets folder. Otherwise, they will not be recognized as pagelets by the portal.

Pagelet Attributes	
Default Column:	Column 1
Content provider name:	
Edit URL:	

The **URL Type** can be either *Non-PeopleSoft URL*, *PeopleSoft Component*, or *PeopleSoft Script*. The availability of these options depends on which **Usage Type** is selected.

If the **URL Type** is set to *PeopleSoft component*, you'll see the **Component Parameters** area appear. Select the **Menu Name**, **Market**, and **Component** name that describe the particular component you want to use for this content reference.

Component Parameters		
*Menu Name:		*Market: GBL
		*Component:

If the **URL Type** is set to *PeopleSoft Script*, you'll see the **iScript Parameters** area appear. Select the **Record (Table) Name**, **PeopleCode Event Name**, **Field Name**, and **PeopleCode Function Name** that describe the particular iScript you want to use for this content reference.

iScript Parameters			
*Record (Table) Name:	WEBLIB_HPDEMO	*Field Name:	ISCRPT1
*PeopleCode Event Name:	FieldFormula	*PeopleCode Function Name:	IScript_YahooStockQuote



For more information on Usage Types and URL Types, see “Using Content References” in the *PeopleCode Reference* documentation.

7. Define the **Template name**.

Set the **Template name** to the name of the portal template you want to use for this page. If you leave it blank, the portal servlet will default it to the content provider template at runtime. If a content provider has not been specified, then it will use the portal's default template. If you select **No Template**, the target content (your application page) will not get “wrapped” with a portal template at runtime, meaning that, by viewing your page, the user will effectively be taken outside of the portal environment (typically not desirable). Another effect of selecting **No Template** is that the **Template name** field becomes hidden on this page.

8. Define the **Content Provider**.

This defaults to the name of the portal. In this case, the **Content Provider** already is set to *Portal* because that is the name of the portal.

9. Enter a **Product** name if desired.

Entering a **Product** is optional. Portal-aware applications can use this field for group processing.

10. Set a **Sequence** number if desired.

You can set a **Sequence** number to adjust where this content reference will appear in the list of content references for this folder. If you leave the **Sequence** number blank, then this content reference will simply be added to the end of the list of content references within this folder. You can only change the sequence of a single item at a time, and each time you do so, you must save your changes to see the results.

11. Define the **Valid from date** and **Valid to date** if your application will use them.

The **Valid from date** is optional, and defaults to the same date as the **Creation Date**. The **Valid to date** is also optional, and is blank by default. These dates may be used for application-specific processing by portal-aware applications.

12. Enable the **Hide from portal navigation** checkbox if desired.

This option makes this particular content reference invisible to users.

13. Define **Attribute names**, **Labels**, and **Attribute values** for the content reference.

These optional names, labels, and values may be used by portal-aware applications for application-specific processing. If you enable the **Translate** option, then the **Label** and **Attribute values** are translatable.

14. Enter **iScript Parameters**, **Component Parameters**, and **Pagelet Attributes** as required.

15. Click the **Add Content Reference** link if you want to add a new content reference.

If you click **Add Content Reference**, a new Content Reference page appears. Define any additional content references as required.

16. Click **Save**.

After you save a content reference, you must set security permissions for it. Follow the procedure below.



Once you've added a content reference for a component in the portal registry, information from the menu definition in Application Designer no longer applies. That is, if you change something in the menu definition, the menu group definition, or security for the menu group definition, these changes will *not* be reflected in the portal. This is because the portal navigation system uses the portal registry, not Application Designer definitions, to construct its menus.

To set permissions for a content reference

1. From the Content Reference definition page, click the **Content Reference Security** tab.

The Content Reference Security page appears.

PEOPLE
Soft

Search: go

Home Menu Favorites Add to Favorites Help Sign out

Home > Menu > Portal Administration > Structure and Content > Administer Workforce > Administer Workforce (GBL) > Use >

Content Ref Administration Content Reference Security

Content Reference Security

Label: Personal Data

Access Type

☒ Permission List ☐ Public

☐ Author Access

Permissions First 1-10 of 10 Last

Permission List	Delete
ALLPANLS	Delete
CCADMIN	Delete
CCDVLP	Delete
CCREST	Delete
HRPNLS	Delete
PTALL	Delete
QEMGR	Delete
QETSTR	Delete
REPRTSUF	Delete
WEBPANLS	Delete

Add

Inherited Permissions First 1 of 1 Last

Permission List

Save

Content Ref Administration | Content Reference Security

Setting permissions for a content reference

2. Set the **Access Type** to either **Permission List** or **Public**.

Typically, you will select **Permission List** and then define which permission list(s) will have access to this content reference. You can select **Public** if you'd like to make the content reference available to all users without manually adding each permission list.

3. Select whether to enable **Author Access**.

Enabling this option means that you (the currently logged on user and the author of the content reference) will be able to access the content reference in the future, regardless of how other permissions are set.

4. Select the **Permission List** (or lists) that will have access to this content reference.

This step only applies if you set the **Access Type** to **Permission List**. You can click the lookup button to see a list of permission lists. To give additional permission lists access to this content reference, click **Add** and select additional permission lists. To remove access to this content reference from a permission list, click **Delete** next to the associated permission list.

Note that **Inherited Permissions** are read only because they show you permissions that have been inherited from folders above. You can change them by going to the folder that defined the cascading permission.

5. Click **Save**.

None of your changes to the content reference are final until you save them.

6. Test your changes in the portal.

After creating the content reference and setting up permissions, test the new content reference by clicking **Home** and navigating to it or click **Test Content Reference**.




For more information, see Security

Moving Folders and Content References

This section describes how to move folders and content references within the portal registry.



When navigating, do not click the folder name (unless it's the name of the new parent folder you'd like to select). Once you click a folder name, as opposed to the  icon, the folder name you selected becomes the new parent folder.

To move a folder or content reference to a new parent folder

1. From the folder page or the content reference page click **Select New Parent Folder**.

PEOPLE
Soft

Search:

Home Menu Favorites Add to Favorites Help Sign out

Home > Menu > Portal Administration > Structure and Content > Administer Workforce > Administer Workforce (GBL) > Use >

Sibling Content References

- Hire
- Add Concurrent Job
- Job Data
- Current Job
- Pay Rate Change
- Contract Data
- Absence History
- Bank Accounts
- Business Expenses
- Citizenship/Passport Data
- Company Property
- Disciplinary Action
- Driver's License Data
- Emergency Contact
- Employee Checklist
- Employee Photo
- General Comments
- Names
- Prior Work Experience
- Visa/Permit Data
- Volunteer Activities
- Hot Key Test

Content Ref Administration **Content Reference Security**

Content Ref Administration

Author: PTDMO

Parent Folder: Use

*Name: PERSONAL_DATA1

*Label: Personal Data

Long Description: (254 Characters)

Usage Type: Target

Storage Type: Remote by URL

URL Type: PeopleSoft Component

Template Name: DEFAULT_TEMPLATE

*Content Provider: Portal

Product: DEMO

*Valid from date: 10/24/2000

Sequence number: 4

Valid to date:

Creation Date: 10/24/2000

☐ No Template

[Add Content Reference](#) [Test Content Reference](#)

Component Parameters

*Menu Name: ADMINISTER_WORKFORCE_GBL

*Market: GBL

*Component: PERSONAL_DATA1

☐ Hide from portal navigation?

Content Reference Attributes

Name: ☒ Translate? [Attribute Information](#)

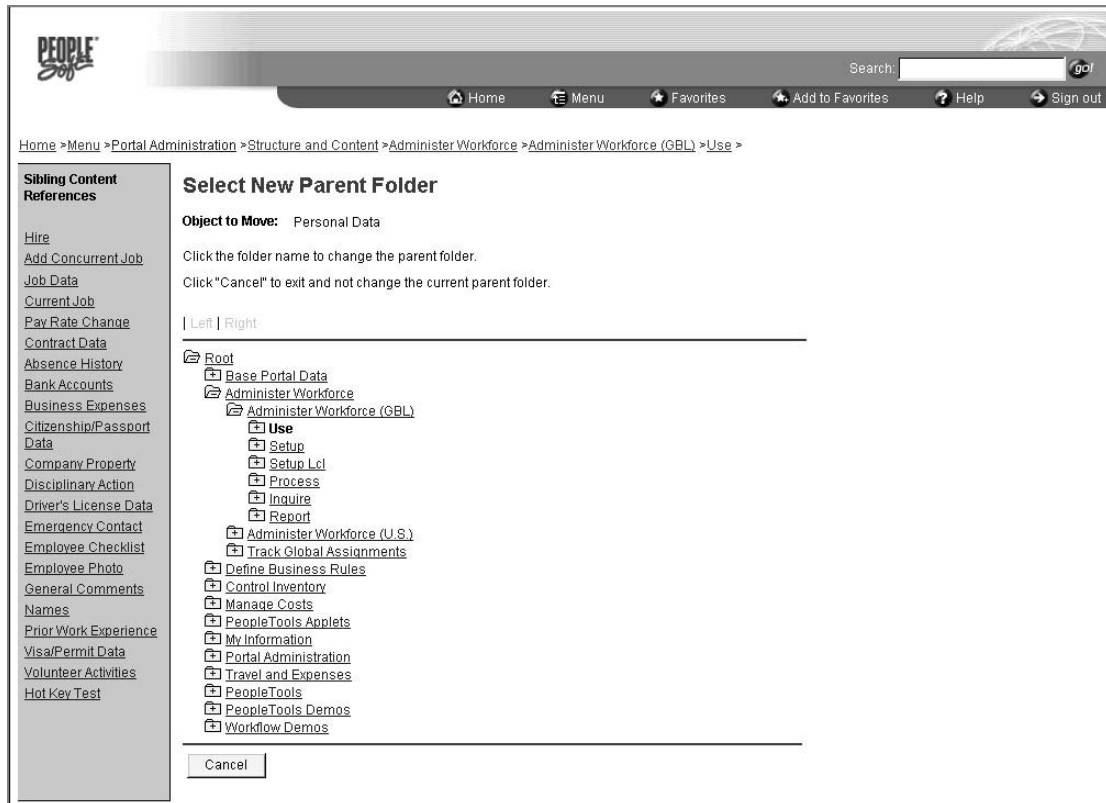
Label:

Attribute value:

Content Ref Administration | [Content Reference Security](#)

Getting ready to move a folder or content reference

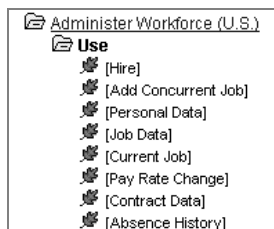
The Select New Parent Folder page appears, as shown below.




Selecting a new parent folder


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, by default.

You can expand the tree to show content references. Content references are shown as “tree leaves” and are shown in normal



Viewing content references

2. Navigate to the new parent folder by clicking the  icon.

When navigating, do not click the folder name (unless it's the name of the new parent folder you'd like to select). Once you click a folder name, as opposed to the  icon, the folder name you selected becomes the new parent folder.

3. Select the new parent folder by clicking on the folder name.

When you click the folder name, the .

4. Click **Save**.

Changes are not apparent until you click **Save**.

Managing General Portal Settings

To manage general portal settings

1. Navigate to the Portal Administration pages.

You can do so by clicking the **Menu** button in the universal navigation header, and then selecting **Portal Administration**.

2. Select **General Settings**.

A page resembling the following appears.

The screenshot shows the 'General Settings' page in the PeopleSoft Portal Administration interface. The page has a navigation header with links for Home, Menu, Favorites, Add to Favorites, Help, and Sign out. Below the header, the breadcrumb trail is 'Home > Menu > Portal Administration > General Settings'. The main content area is titled 'General Settings' and contains several form fields: 'Portal name' (text box with 'PORTAL'), 'Long Description' (text area with 'PORTAL'), and 'Default Template Name' (dropdown menu with 'DEFAULT_TEMPLATE' selected). Below these fields, there is a section for 'Content Providers' with a table listing 'Portal' and 'PortalServlet', each with 'Edit' and 'Delete' buttons. At the bottom of the page, there is a 'Save' button.

Managing general portal settings

3. Update the **Description**.

The **Description** allows a longer description than the portal name provides. Note that the portal name cannot be changed. This description appears with the list of portals on the main Portal Administration page.

4. Define the **Default template**.

This is the portal template that will be used to wrap all registered content that has not been associated with another template, as well as all unregistered content.



You should not change the default template, unless you are absolutely sure of the consequences this change will have.

5. Edit, add, or delete content providers as required.

Click **Edit** next to the content provider you want to edit, or click **Add Content Provider** to add a new content provider. You can collapse the list of content providers by clicking **▼ Content Providers**.



You should not change the “Portal” or “PortalServlet” content providers. These are default content providers of special significance to your portal. Changing these settings could make your portal inoperative. If, for some reason, you change the default content provider, you will have to sign out of the portal and sign back in before your changes will take effect.

6. Click **Save**.

The portal registry is not updated until you click **Save**.

Managing Content Providers

The PeopleSoft Portal enables you to define content providers. A content provider is a name that you can use to refer to some source of HTML content. In more practical terms, a content provider is a URI string that defines the database and server to be used when the portal servlet attempts to retrieve content, proxy addresses, and assemble pages. The use of content providers also simplifies PeopleCode programming, since long URIs can be referred to quite easily by the appropriate content provider name. Some content provider names (HRMS, EPM, SA, FDM, and CRM) are preset in your portal. You can add additional content providers as necessary.

Two default content providers come with each portal: Portal and PortalServlet. The Portal content provider defines the location (URI) of the portal administration pages. The PortalServlet content provider defines the location (URI) of the portal servlet.

To edit, add, or delete a content provider

1. Navigate to the Portal Administration pages.

You can do so by clicking the **Menu** button in the universal navigation header, and then selecting **Portal Administration**.

2. Select **General Settings**.
3. Select the appropriate action that you want to do with a content provider (add, delete, or edit).

To delete an existing content provider, click the **Delete** button next to the content provider you want to delete.



Do not delete the default content providers. They have special significance to your portal. Changing these settings could make your portal inoperative.

To add a content provider, click the **Add Content Provider** link. To edit an existing content provider, click the **Edit** link to the right of the content provider name you want to edit. If you are adding or editing a content provider, a page resembling the following appears.

The screenshot displays the 'Content Provider Administration' interface. On the left, a sidebar lists 'Content Providers' and 'PortalServlet'. The main content area has a title 'Content Provider Administration' and a breadcrumb trail: 'Home > Menu > Portal Administration > General Settings > Content Providers'. Below the title, there are several input fields: 'Portal name' (set to PORTAL), 'Name' (Portal), 'Long Description' (Default Content Provider, with a 254-character limit), 'Portal URI Text' (http://mlee2021800/servlets/iclientservlet/peoplesoft8/), and 'Default Template Name' (DEFAULT_TEMPLATE). A 'Save' button is located at the bottom left of the form area.

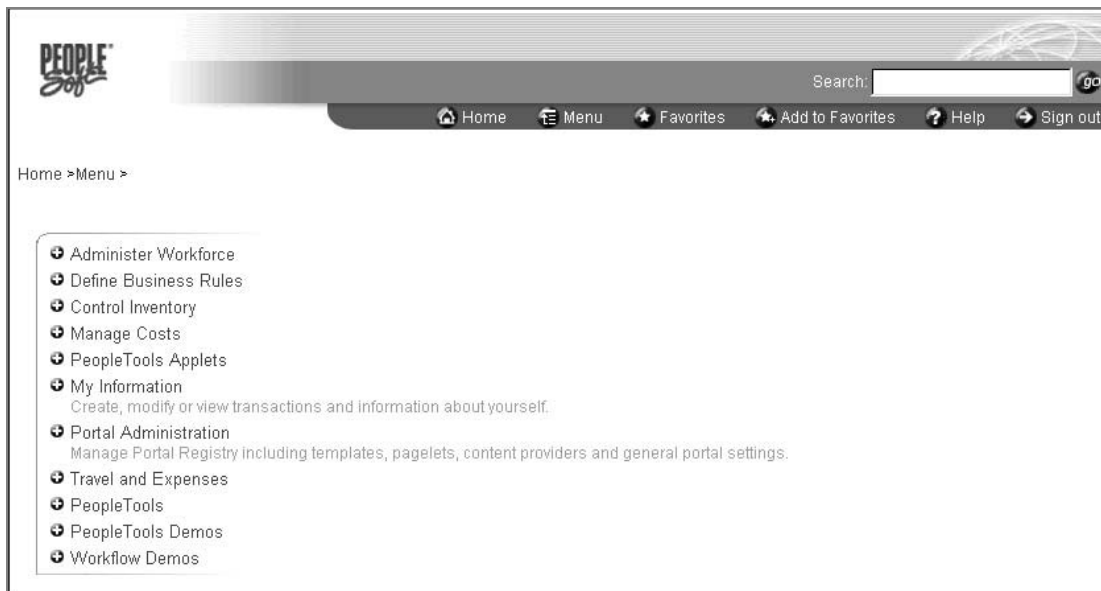
Managing content providers

4. Define the **Portal Name** and **Long Description**.
5. Define the **Portal URI Text**.
6. Define the **Default Template Name**.
7. Click **Save**.

Importing a Menu Group into the Portal Registry

This section shows you how to import an entire menu group into the portal registry. The menu import process creates content references for all components and pages that belong to the menu group. We'll start with a portal example that has had some menu groups imported, and will add a new menu group. At this point, it resembles the example below. At the end of this process, there

will be a new folder, called “Mobile Applications,” added on the same level as the other top-level folders. By default, it will appear at the end of the list, after “Workflow Demos.”



The menu view of registered applications



The menu pagelet, showing registered applications

To import a menu group in the portal registry

1. Navigate to the Portal Administration pages.

You can do so by clicking the **Menu** button in the universal navigation header, and then selecting **Portal Administration**.

2. Select **Menu Import**.

The PeopleSoft Menu Import page appears.

PEOPLE
Soft

Search: go

Home Menu Favorites Add to Favorites Help Sign out

Home > Menu > Portal Administration > Menu Import

Menu Import

General Settings

*Portal name:

*Content provider name:

Template Name: PORTAL_DEFAULT

Menu Group Selection

☐ All Menu Groups Menu Group:

Optional

*Valid from date: Product:

Valid to date:

 [Process Monitor](#)

Preparing to import a PeopleSoft menu group

3. Select the **Portal name**, **Content provider name**, and **Template name**.

These are all specified in the **General Settings** area, and you will typically accept the defaults. Both **Portal name** and **Content provider name** are required fields. Specifying a **Template name** is optional; if you do not specify a template, then the portal servlet will use the template associated with the content provider. If a template is not associated with the content provider, then the default template will be used for all content references created through this import.

4. Select the **Menu Group** to import or click the **All Menu Groups** checkbox.

We'll select the *Mobile Applications* menu group for our example.

Menu Group Selection

☐ All Menu Groups Menu Group:

5. Optionally select the **Valid from date** and **Valid to date**, and specify a **Product**.

These are all specified in the **Optional** area, and you will typically accept the defaults, unless you want to set specific valid dates or associate a product code with the imported menu group(s).

6. Click **Import** to begin the import process

When you click the **Import** button, Process Scheduler launches an Application Engine program that imports the selected menu group definition(s) into the portal registry. This process typically takes several minutes and sometimes requires 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 group(s) being imported. Remember, once the menu group definitions are imported into the portal registry, there is no dynamic connection between the menu group definition in Application Designer and the portal registry, nor is there a dynamic connection between security permissions set in Application Designer and those set in the portal registry. If a change is made in one of these areas, but not in the other, the menu definition and the portal registry will become “out of sync.”

7. Click **Process Monitor**.

This launches the Process Monitor so you can view the status of the menu import. When the PORTAL_MENU process has run successfully, move on to the next step, testing the content reference.

The screenshot shows the 'Process Monitor' window with the 'Process List' tab selected. It features a search bar with fields for User (PTDMO), Type, Last (1 Days), and a Refresh button. Below the search bar are fields for Server, Run Status, and Instance. A checkbox for 'View Job Items' is also present. The main area contains a table of process runs with columns: Instance, Seq., Process Type, Process Name, User, Run Date/Time, Run Status, and Details. Two rows are visible, both showing 'Success' status. At the bottom, there is a 'Save' button and a link to 'Server List'.

Instance	Seq.	Process Type	Process Name	User	Run Date/Time	Run Status	Details
6		Application Engine	PORTAL_MENU	PTDMO	11/16/2000 7:45:48PM PST	Success	Details
5		Application Engine	UPG81RPTG	PTDMO	11/16/2000 8:30:47AM PST	Success	Details

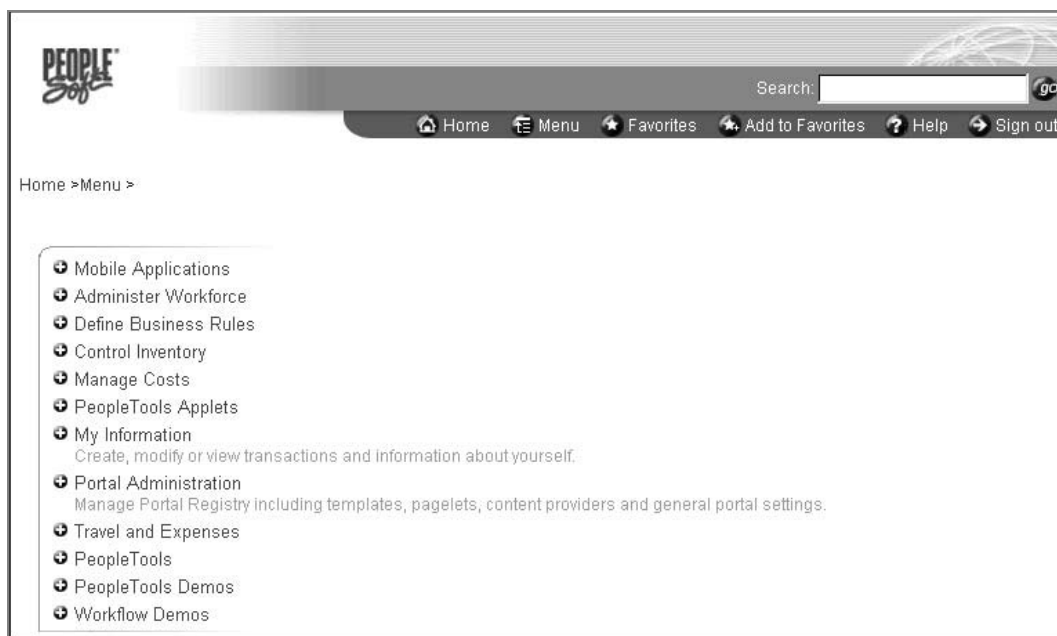
Viewing the status of menu group import in Process Monitor

8. Test your changes in the portal.

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.



If the menu group does not appear, open the folder definition for the menu group and define a label for the folder.



Viewing an imported menu in the portal menu



Viewing an imported menu in the menu pagelet

Selecting Components to Include in Portal Navigation

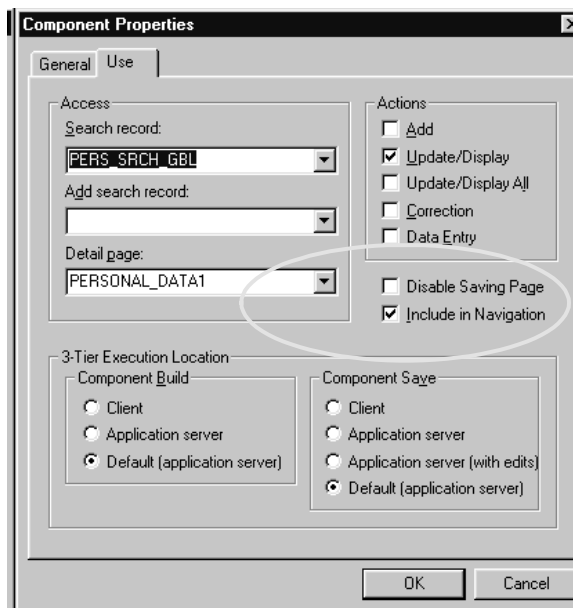
During the menu import process, it's possible to control which components are included in your portal navigation system.

To select components to include in portal navigation

1. From Application Designer, open the appropriate component.

2. Select File, Object Properties and click the Use tab.
3. Select the **Include in Navigation** checkbox.

Turning on the **Include in Navigation** option means that this component will be imported during a menu import operation and initially will be visible to users with appropriate permissions (of course, you can change the permissions for any content reference after the menu import as necessary). Turning off the **Include in Navigation** option means that this component is imported during a menu import, but will not be visible to users. This checkbox controls the setting of the PORTAL_HIDE_FROM_NAV attribute name for the related content reference in the portal. During a menu import, the attribute value is set to *FALSE* if the **Include in Navigation** option is turned on for this component. If it is turned off, the attribute value is set to *TRUE*.



Setting component properties in Application Designer

4. Click **OK**.

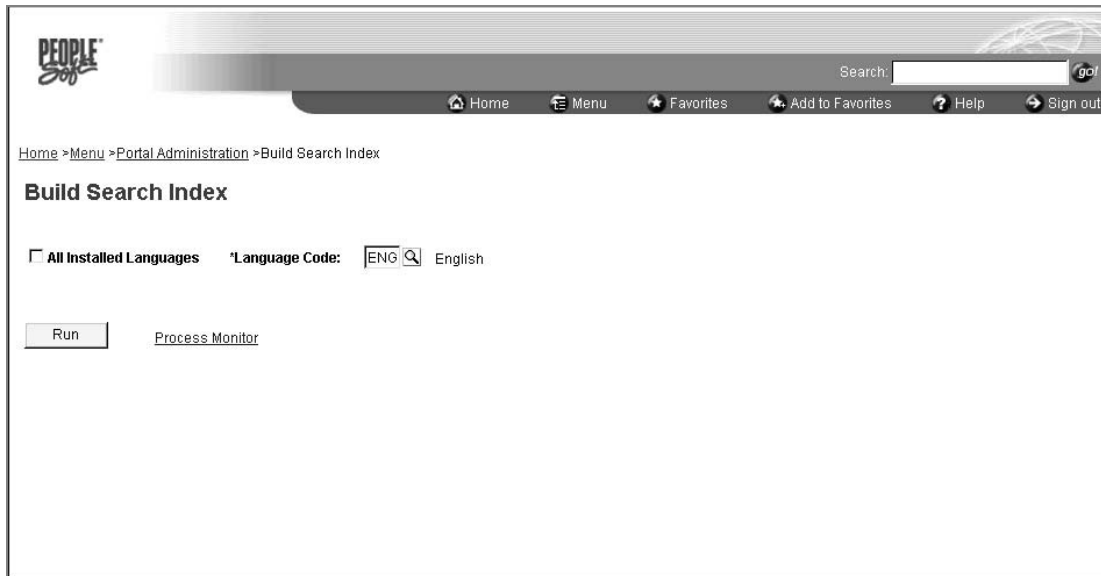
Remember that changes you make to the component in Application Designer will not be reflected in the portal until you do another menu import.



For more information, see Application Designer

Building a Search Index

You can build a search index from the Portal Administration pages, and you can even schedule the build process via Process Scheduler. Note that each time you build the search index, the index is entirely rebuilt, thus overwriting any previous entries.



The Build Search Index page



For more information on building search indexes for the PeopleSoft portal, see *Building and Using Portal Search Indexes*.

Managing Base Portal Data

Base Portal Data is a unique folder of special significance to the portal. Settings in Base Portal Data should not be changed, unless you know exactly what you are doing and what the consequences may be. It's entirely possible to make the portal inoperative by changing some of the settings in this folder.

Managing Templates

Templates are stored in the HTML catalog, but are edited through Portal Administration pages—they are not intended to be edited in Application Designer. For the portal to process templates properly, they must be referenced in Base Portal Data, specifically in the Templates folder. Having them stored in the HTML catalog enables them to be upgraded like other objects.

Templates are stored in the HTML catalog according to the following naming convention (they always beginning with the literal PORTAL_):

```
PORTAL_<PORTALNAME>_<CONTENT REFERENCE ID>
```

Because templates can be larger than the maximum size permissible in the HTML catalog, the template may be divided into several files in the HTML catalog. In this case, a letter is appended to each HTML object comprising the template, as follows:

PORTAL_PORTAL_123456789_A
 PORTAL_PORTAL_123456789_B

To edit, add, or delete an existing folder

1. Open the Portal Administration page.

If you haven't already done so, navigate to the Portal Administration page by clicking the **Home** button, and then selecting **Portal Administration** from the menu.

2. Click **Manage the structure and content for a portal**.
3. Select the **Base Portal Data** folder.
4. Select the **Templates** folder.

A page resembling the following appears.

Home > Menu > Portal Administration > Structure and Content > Base Portal Data > Templates

Structure and Content

* Click the folder label to view the child folders and content references for that folder
 * Click the "Edit" link to edit the folder definition

▼ Folders First 1 of 1 Last

[Add Folder](#)

* Click the "Edit" link to edit the content reference definition

▼ Content References First 1-9 of 9 Last

Label	Edit	Sequence number	Delete
PORTAL_ADMIN	Edit		Delete
PORTAL_STMP_HOMEPAGE	Edit		Delete
PORTAL_HP_DEFAULT_APP	Edit		Delete
PORTAL_HP_DEFAULT_ENT	Edit		Delete
PORTAL_ADMIN_NOSIDENAV	Edit		Delete
PortalNavigationTemplate	Edit		Delete
PORTAL_DEFAULT	Edit		Delete
PORTAL_RELATEDLINKS	Edit		Delete
UniWrapperTemplate	Edit		Delete

[Add Content Reference](#)

[Save](#)

Viewing portal templates

As is shown in the above example, templates are defined as content references.

5. Select the appropriate action (add, delete, or edit).

To delete an existing template, click the **Delete** button next to the template you want to delete. A confirmation dialog appears. Click **OK** only if you're sure that you want to delete the template. Click **Cancel** if you're not sure if you want to delete the template.



Do not delete the PORTAL_DEFAULT template. This is the default template, which has special significance to your portal. Changing this setting could make your portal inoperative. Do not delete any template unless you are fully aware of how it used within the portal.

To add a template, click the **Add Content Reference** link. To edit an existing template, click the **Edit** link to the right of the content reference name you want to edit. A page resembling the following appears. Note that the bar on the left side of the page provides navigation to sibling templates (content references).

The screenshot shows the PeopleSoft Portal Administration interface. The top navigation bar includes links for Home, Menu, Favorites, Add to Favorites, Help, and Sign out. The breadcrumb trail is: Home > Menu > Portal Administration > Structure and Content > Base Portal Data > Templates >.

The main content area is titled 'Content Ref Administration' and includes a 'Content Reference Security' tab. The form fields are as follows:

- Name:** ADMIN_TEMPLATE
- Label:** PORTAL_ADMIN
- Long Description:** Portal Administration Template (254 Characters)
- Usage Type:** HTML template
- Storage Type:** Local (in HTML Catalog)
- Product:** PRTL
- Valid from date:** 05/13/2000
- Sequence number:** (empty)
- Valid to date:** (empty)
- Creation Date:** 05/13/2000
- Parent Folder:** Templates (with a 'Select New Parent Folder' button)
- Portal URL:** <TEMPLATE>PORTAL_ADMIN
- HTML Area:** A large text area containing HTML code for the template, including <HTML>, <HEAD>, <TITLE>, <BODY>, <TABLE>, <TR>, <TD>, <ICCLIENTCOMPONENT>, and <SOURCE> tags.
- Content Reference Attributes:** A section with fields for Name, Label, and Attribute value, along with a 'Translate?' checkbox and buttons for 'Attribute Information' and 'Delete'.

At the bottom of the form, there is a 'Save' button and a status bar indicating 'Content Ref Administration | Content Reference Security'.

Defining a portal template

- Define the content reference for the template as you would for any other content reference.

Note that the **Usage Type** is set to *HTML Template* and the **Storage Type** is set to *Local (in HTML Catalog)*.

The **Portal URL** defaults to <TEMPLATE> followed by the template name. Because a template is really just a content reference, it requires a unique URL. You can change the **Portal URL** if desired, but it's recommended that you simply accept the default to avoid duplicate URLs.

The **HTML Area** is simply an area in which you can view and edit the HTML that describes the template.

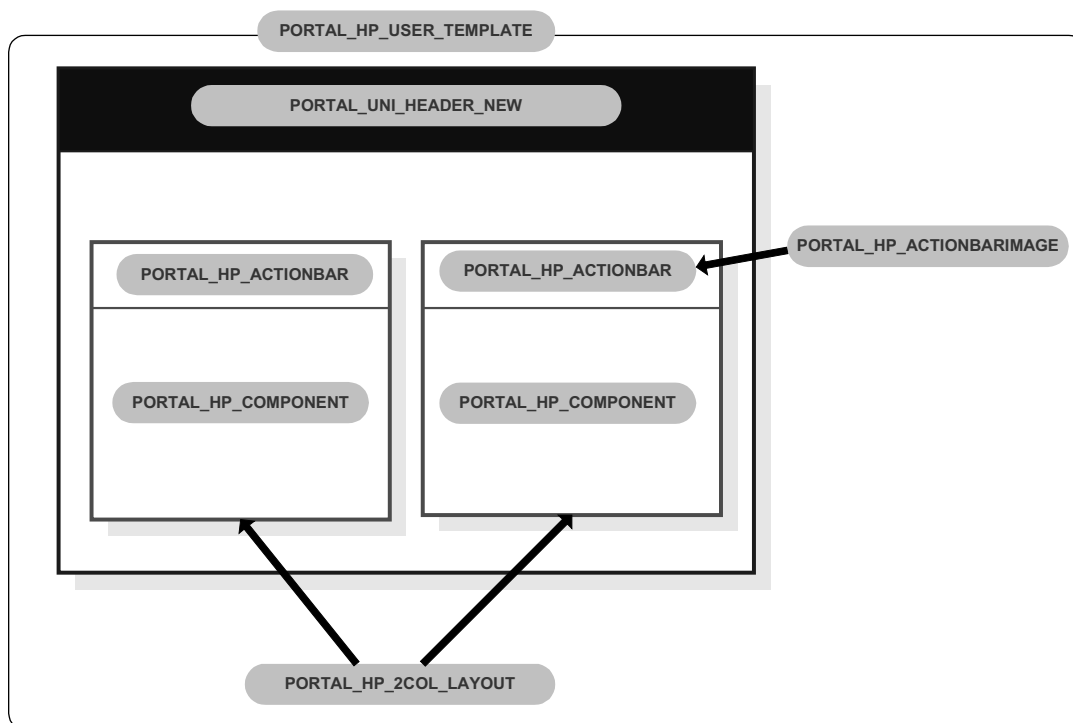
7. Click **Save**.

After you save a content reference, you must set security permissions for it. Follow the procedure above for setting permissions for content references.

Changing the “Look and Feel” of a Portal

There are a handful of HTML objects that you can edit in order to control the “look and feel” of your portal. These HTML objects are detailed below.

The following example shows a 2-column layout used to display two pagelets side-by-side. You can see which part of the page that each HTML object controls. For example, `PORTAL_HP_USER_TEMPLATE` controls the overall page (shown in blue outline), and `PORTAL_HP_COMPONENT` controls the look of each pagelet. Similarly, `PORTAL_UNI_HEADER_NEW` controls the navigation header (blue background). Each HTML object is discussed in more detail below.



The HTML objects comprising a template

PORTAL_HP_USER_TEMPLATE

Each user’s homepage is generated from this HTML object. This HTML object can be modified to add default components that will be displayed to all users.

PORTAL_HP_3COL_LAYOUT

This is the HTML object for the three-column user homepage. This HTML object's default column widths are set to 33% for each column.

PORTAL_HP_2COL_LAYOUT

This is the HTML object for the two-column user homepage. This HTML object's default column widths are set to 33% for column one and 66% for column two.

PORTAL_HP_COMPONENT

This HTML object is used to create all homepage pagelets. This HTML object can be modified to change the look and feel of a pagelet.



Do not move, delete, or change the first or last lines of this HTML definition:

```
<!-- Begin Pagelet=%BIND(:6) -->
.
.
.
<!-- End Pagelet=%BIND(:6) -->
```

These are special lines that must exist exactly as shown. You must also ensure that all bind variables exist and exactly match the bind variables used in the associated PeopleCode.

PORTAL_HP_ACTIONBAR

This HTML object is used to create the pagelet action bar. The default action bar icons are remove and minimize/maximize.

PORTAL_HP_ACTIONBARIMAGE

This HTML object is used to create the “customize” image icon for the pagelet action bar. This HTML object is added only if the user has defined an advanced personalization page.

PORTAL_UNI_HEADER_NEW

This HTML object controls the universal header for the portal.

With the exception of the PORTAL_UNI_HEADER_NEW HTML object, the above homepage HTML objects 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, IScript_HPCompExpand.

Registering Queries as Target Pages in the Portal

You can register either the PeopleSoft query page or a predefined query in the portal registry. To do so, follow the procedure below.

To register a new query page or a predefined query

1. Create a new content reference for the query page.
2. Set the **Usage Type** to *Target* and the **URL Type** to *Non-PeopleSoft URL*.
3. Define the **Content Provider**.
4. In the **Portal URL** field, enter the following:

```
ICType=Query
```

To specifically add an individual query (ADDRESSLIST, for example), change the URL field to the following:

```
ICType=Query&ICAction=ICQryNameURL=ADDRESSLIST
```

PEOPLE
Soft

Search: go

Home Menu Favorites Add to Favorites Help Sign out

Home > Menu > Portal Administration > Structure and Content > Mobile Applications >

Sibling Content References

Content Ref Administration Content Reference Security

Content Ref Administration

Name: QUERY TEST **Author:** PTDMO

Label: Query Test **Parent Folder:** Mobile Applications

Long Description: (254 Characters) Query Test

Usage Type: Target **Product:** ☐ **Valid from date:** 11/24/2000

Storage Type: Remote by URL **Sequence number:** ☐ **Valid to date:**

URL Type: Non-PeopleSoft URL **Creation Date:** 11/24/2000

Template Name: DEFAULT_TEMPLATE **PORTAL_DEFAULT** ☐ No Template

Content Provider: Portal

***Portal URL:** ICType=Query&ICAction=ICQryNameURL=ADDRESSLIST

☐ Hide from portal navigation?

Content Reference Attributes

Name: ☒ Translate? [Attribute Information](#)

Label:

Attribute value:

Content Ref Administration | [Content Reference Security](#)

Registering a query as target content

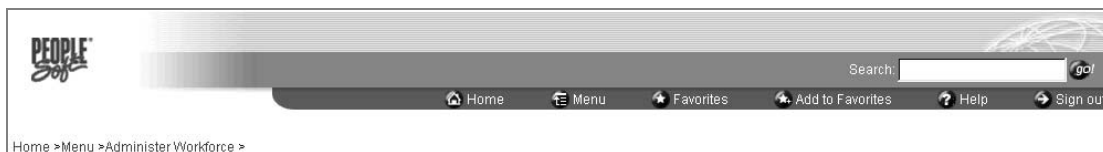
CHAPTER 7

Using Portal Navigation Features

The PeopleSoft portal provides a set of navigation tools to help you navigate the portal registry. The following navigation tools are included:

- Menu navigation
- Breadcrumbs
- Favorites
- Search

All of these navigation applications are included in a standard template pagelet delivered with PeopleTools called the *universal navigation header*, shown below.



The PeopleSoft universal navigation header

Every PeopleSoft portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like **Home**, **Menu**, **Favorites**, and so on) the universal navigation header can also display a welcome message for each user.

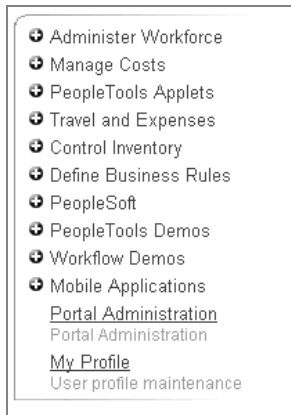
Menu Navigation

You can drill down through the registry using *menu navigation*. The “top” level of the menu corresponds to the first level of organization (directly off the root level) in the portal registry. When you click the **Menu** button on the universal navigation header, these top levels of the registry are shown. In addition to the menus provided for your other PeopleSoft applications, there is a “My Profile” link, as well as a “Portal Administration” option that takes you to the portal administration pages, if you have appropriate security privileges.

The “three column” navigation feature is used to show you where you are in the registry, as you drill down through a menu. Up to three levels of the registry are displayed at once. Each column presents you with a page containing the folders and content references within the selected product.

Example of Three Column Navigation

When you click the **Menu** button on the universal navigation header, you are presented with the top-level categories in the registry in a single column on the left side of the display, as shown below.



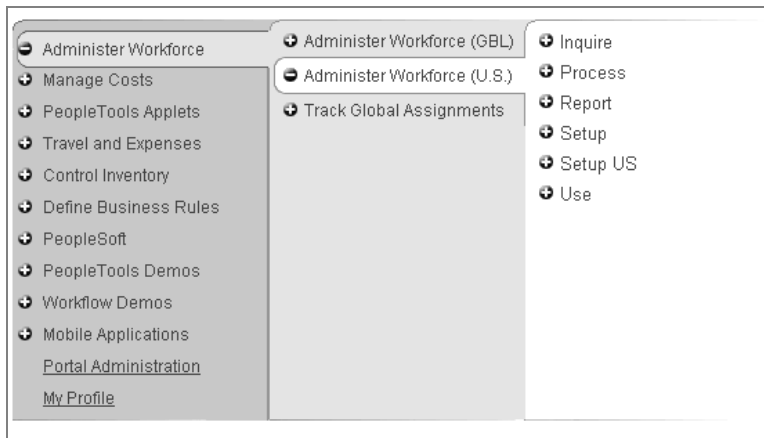
Top level categories in the registry

Clicking the “+” button expands the “Administer Workforce” category to the next level in the registry, and two levels are now shown in two columns, as shown below.



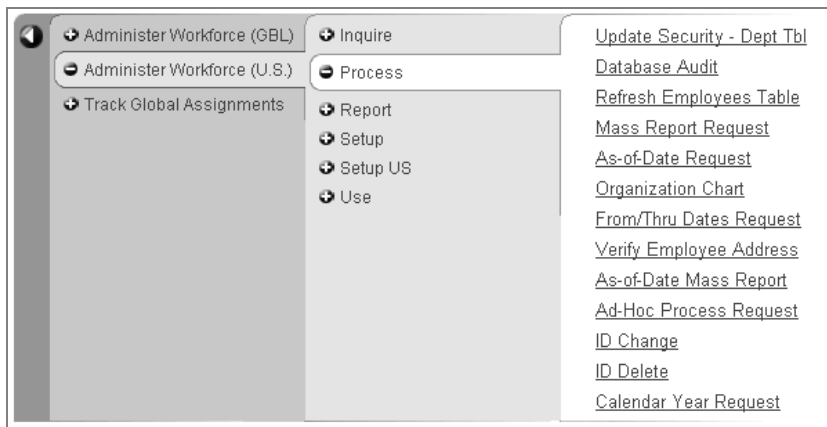
Top two levels of the registry

Clicking the “+” button expands the “Administer Workforce (U.S.)” category to the next level in the registry, and three levels are now shown in three columns, as shown below.



Top three levels of the registry

At this point, the maximum number of columns (three) are displayed, so subsequent clicks of the “+” button will not result in additional columns being displayed. Instead, the three column display shifts one column to the left each time a level is expanded. Note in the example below that “Administer Workforce (U.S.)” has moved left to occupy the left-most column, rather than the center column it previously occupied. Also note the left arrow button on the left edge of the display, indicating the existence of higher levels in the hierarchy not currently displayed.

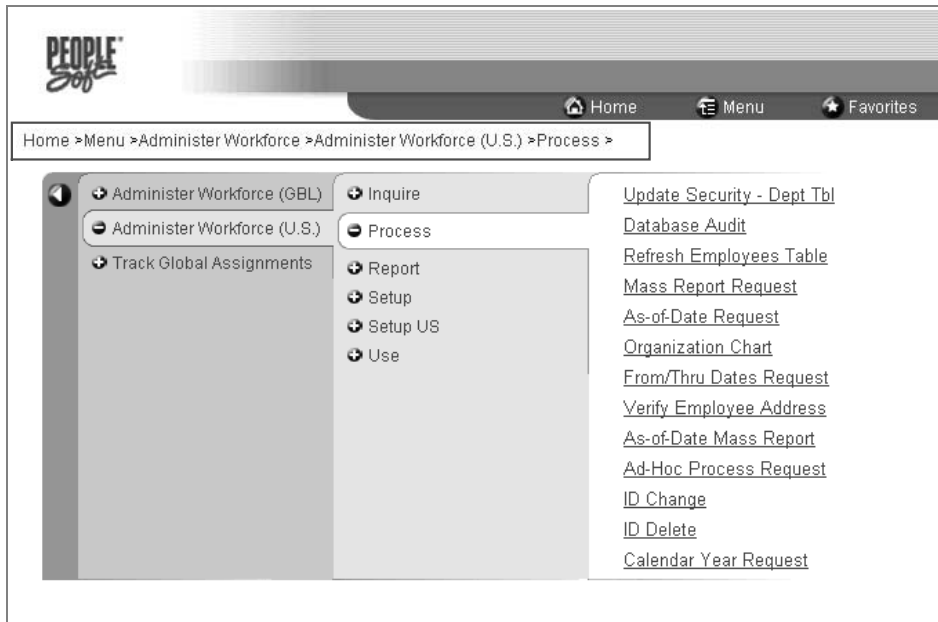


Second, third, and fourth levels of the registry

Continuing in this manner, you will eventually navigate to the application component of interest to you. When the component is displayed, the three column navigation display disappears.

Breadcrumbs

As you drill down through the different levels of the registry, a “breadcrumb trail” appears that shows the path you’ve selected. In the example below, we’ve highlighted the breadcrumb trail by outlining it in a red box. Each registry level is separated by an angled brace (>), and you can select any level to navigate directly back to that level.



Using breadcrumbs for navigation

Favorites

You can “bookmark” frequently accessed folders and content references through the **Add to Favorites** button on the universal navigation header. When you want to navigate quickly to a favorite link, simply click the **Favorites** button and to see all the links you’ve added. Clicking any of these links will take you directly to the desired page.

Search

A search engine is built in to the PeopleSoft Portal. To submit a search, you enter a query string into the Search box in the universal navigation header.



Entering a query string to search



For more information on search functionality, see Building and Using Portal Search Indexes.

My Profile

The PeopleSoft portal enables users to enter and change their personal “profile” information. By editing your profile, you can change your portal password, define your email address, and set up an “alternate user” to receive routings in your absence.

To change your profile

1. Select **Menu, My Information, My Profile** from the universal navigation header.

The General Profile Information page appears.

General Profile Information

Unger,Annette

Password

[Change password](#)

[Change or set up forgotten password help](#)

Personalization

Changes to Personalization settings require you to log off and log back on in order to take effect.

My preferred language for reports and email is: English

Currency Code:

[Set Personalizations](#)

Email

E-mail Address:

Alternate User

If you will be temporarily unavailable, you can select an alternate user to receive your routings.

Alternate User ID:

From Date: (example:12/31/2000)

To Date: (example:12/31/2000)

[Miscellaneous User Links](#)

Personalizing your profile

2. Change your **Password, Personalization, Email, or Alternate User** information as required.
3. Save your changes.

CHAPTER 8

Working with Homepages

From the perspective of users, one of the most interesting features of a portal is the ability to personalize it. With the PeopleSoft Portal, users can define and store their own portal homepage in which they specify their preferences for layout and content. In other words, they can choose which content they want to see, and where they want it displayed.

The first time a user logs on to a PeopleSoft portal, they will be presented with the default homepage for that portal. Users will continue to be directed to the default homepage upon each log on, until they define a personal homepage.

Creating a personal homepage is a simple process for any user, and yields many benefits for minimum effort. Any easy-to-use “Personalize” page is provided that makes customization very simple and fast.

Each personalized homepage template is stored as a portal template in the individual user’s folder in the portal registry (under the Base Portal Data folder). At runtime, the portal servlet determines which user is referencing a homepage and uses this template to construct the page. If no personal homepage template exists for the user, the default homepage template is retrieved and used.

The screenshot shows the myPlanet PeopleSoft corporate portal. At the top, there's a navigation bar with links: Home, Map, Contact Us, PeopleSoft.com, Help, Sign Off. Below this is a banner for 'People power the internet.' and a clock showing 4:00 pm on Friday, 17 November 2000. The left sidebar has a 'Welcome Dan' section with 'Personalize' and 'Layout' links. Below that is a search bar with 'Planet PeopleSoft' entered. Further down is the 'PlanetNavigator' section with a 'View A-Z Listings' and 'View Category Map' link, and a list of links: About PeopleSoft, Communities, Customers and Partners, Divisions and Departments, Media, Meetings, My Info, Policies and Guidelines, Products and Solutions, Purchases, Recruiting and Staffing, Resources and Services, Tech Support, Training. The main content area features a headline 'Spain Generates PeopleSoft 8 Audience Excitement' with a photo of a group of people. Below this is a 'PeopleSoft Headline News' section with several bullet points. On the right, there's an 'Events Calendar' and a 'BART Schedule' section. The footer includes copyright information and the PeopleSoft8 logo.

A personal homepage for a sample corporate portal

Creating a Personal Homepage

Click either the Personalize **Layout** link or the Personalize **Content** link at the bottom of the Universal Navigation Header.



Selecting to Personalize the Layout or Content of a Homepage

The Personalization page appears. Depending on whether you clicked the **Layout** link or the **Content** link, either the **Personalize Layout** page or the **Personalize Content** page will appear. You can select either one by clicking the appropriate link.

Customizing Content

Customizing content that appears on your own homepage is easy. Follow the procedure below to personalize content.

To customize content on your personal homepage

1. Click the Personalize **Content** link.

A page resembling the following appears.

PEOPLE

Search:

[Home](#) [Menu](#) [Favorites](#) [Add to Favorites](#) [Help](#) [Sign out](#)

Personalize Home Page Content

Choose Pagelets: Simply check the items that you want to appear on your homepage.
Remember to click "Save" when done.

Arrange Pagelets: Go to [Personalize Layout](#)

Welcome Message:

<p>QE</p> <p><input type="checkbox"/> Content Reference Query</p> <p><input type="checkbox"/> Current Content Provider</p> <p><input type="checkbox"/> Who Am I</p> <p><input type="checkbox"/> QE Portal Links</p> <p>Finance</p> <p><input type="checkbox"/> Yahoo Stock Quote</p>	<p>PeopleSoft Applications</p> <p><input checked="" type="checkbox"/> Menu</p> <p>Miscellaneous</p> <p><input type="checkbox"/> Bart Schedule</p> <p><input type="checkbox"/> Zagat Guide</p> <p><input type="checkbox"/> OneLook</p> <p><input type="checkbox"/> Calendar</p> <p><input type="checkbox"/> World Clock</p> <p><input type="checkbox"/> Calculator</p> <p><input type="checkbox"/> Dictionary</p> <p><input type="checkbox"/> Currency Converter</p>	<p>News</p> <p><input type="checkbox"/> Excite Business News</p> <p><input type="checkbox"/> Technology News</p>
--	---	---

[Return to Home](#)

Personalizing the Content of Your Homepage

2. Choose which pagelets you want to appear on your personalized homepage.

The selection of pagelets you can choose from is preset by your portal administrator.

3. Click the **Save** button.

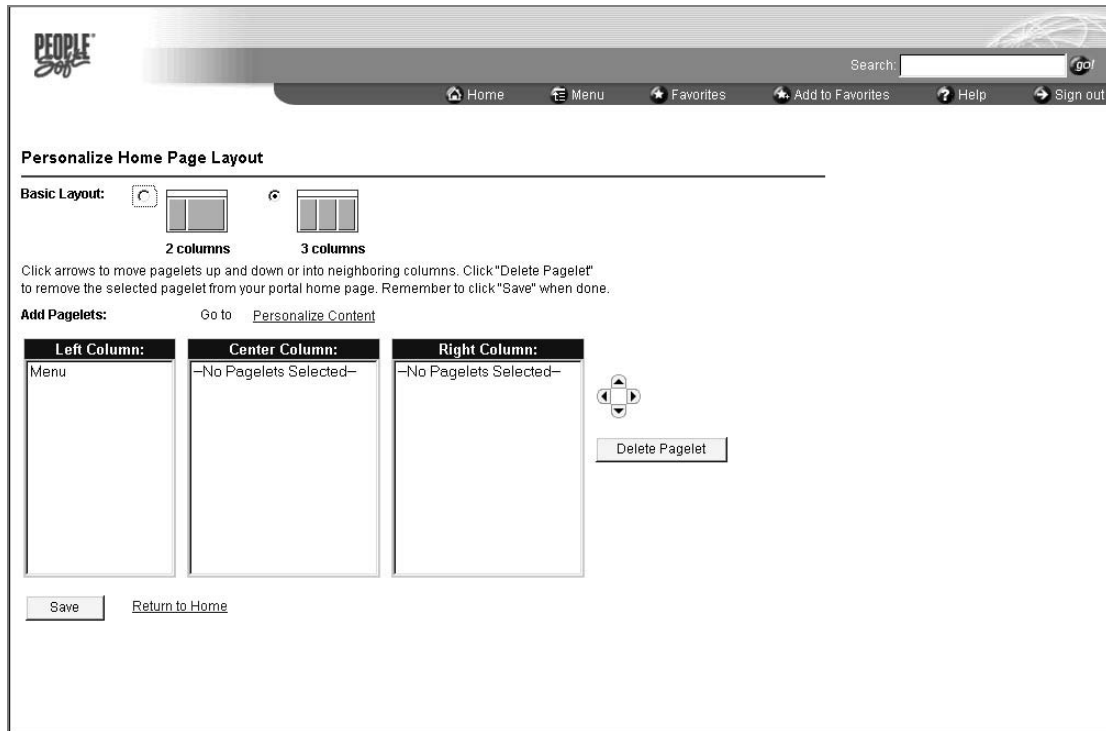
Customizing Layout

After selecting the exact content you want to see on your personal homepage, you'll probably want to arrange it according to your personal taste. As with customizing content, customizing the layout for the content homepage is very simple. Follow the procedure below to personalize the layout.

To customize layout of your personal homepage

1. Click the Personalize **Layout** link.

A page resembling the following appears.



Personalizing the Layout of Your Homepage

2. Select the **Basic Layout** (either **2 columns** or **3 columns**).

Selecting a two-column layout results in one narrow column on the left side of your homepage, and one wider column to the right. The left column takes one third of the horizontal display area, and the right column takes the remaining two thirds of the display area. Selecting a three-column layout results in three narrow columns of equal width. However, if you place a “wide” pagelet in a narrow column, the column automatically stretches to accommodate the wider pagelet. The layout page changes to reflect either a two-column or three-column layout, depending on your selection.

3. Arrange the pagelets.

You can easily delete or change the location of any pagelet on the page. To delete the selected pagelet(s) from your personalize homepage, select the pagelet that you want to delete. Then click the **Delete Pagelet** button. (Doing so does not remove the pagelet from the portal registry—it only removes it from whichever personal homepage you’re working with.) To move the pagelet, select the pagelet you want to move by clicking the appropriate checkbox. Then click the arrow buttons to the right of the column display to move the pagelet to the desired location:



Arrow Buttons

For example, to move the “My Portfolio” pagelet from the top row in the right column to the second row in the center column, first select the “My Portfolio” checkbox. Then click the left arrow button once to move the pagelet into the center column. The pagelet is added to the bottom row of the center column. Click the up arrow until “My Portfolio” appears in the second row.

4. Click the **Save** button.

CHAPTER 9

Building and Using Portal Search Indexes

This document explains how to build a search index from the portal registry, how to submit search requests, and how search requests are processed and results are displayed. It also discusses how to use the Verity search engine to index and search content other than that from the portal registry. It does not address detailed questions about the Search API and the Portal Registry API. For this information, see the PeopleCode documentation.

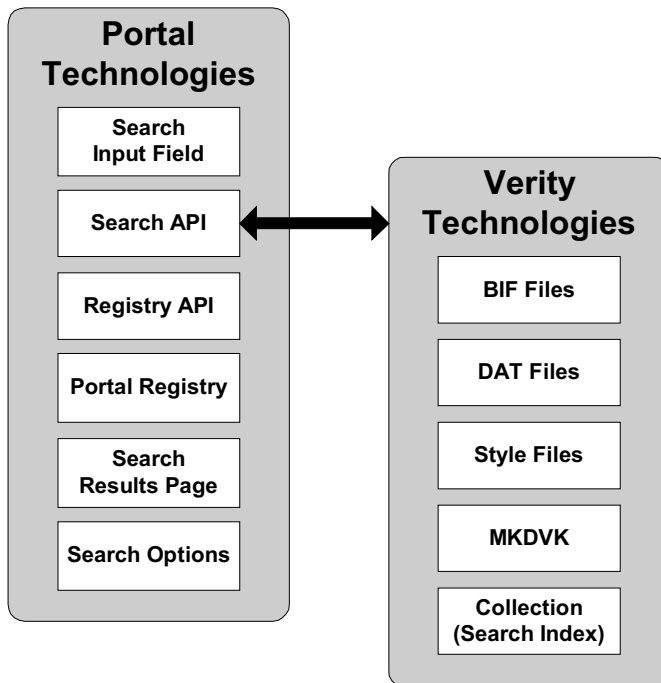
PeopleSoft Portal search functionality depends on two main sets of technology: PeopleSoft-built Portal technology and the Verity search engine. The end result of combining these technologies is that portal users can easily and efficiently search for any content references registered in the portal registry.



There currently is no method to index documents other than content references within the portal registry.

Search Architecture Overview

The following illustration provides an overview of the main pieces of the search architecture embedded in PeopleTools. As you can see, there are two main areas of technology—that provided by the PeopleSoft Portal and that provided by Verity. They are connected by the PeopleSoft search API. Each component is discussed in more detail below.



Components of the PeopleSoft Portal search architecture

PeopleSoft Portal Technologies

Search Input Field. Captures a query string entered by users in the portal header.

Search API. Passes the query string captured in the search input field to the Verity search engine.

Portal Registry API. Applies security to filter search results.

Portal Registry. Contains a repository of content references that can be searched.

Search Results Page. Formats and displays search results for the user.

Search Options. Gives individual users the ability to personalize search behavior and results.

Verity Technologies

The basic pieces of the Verity architecture incorporated in the PeopleSoft Portal search architecture are as follows:

- **Verity Collection.** This is the set of files forming a search index. When a user performs a search, the search is conducted against the Verity collection.
- **BIF File.** This is an intermediate file created in the process of building a Verity collection. The BIF file (or “bulk insert file”) is a text file used to specify the documents to be submitted to a collection. It contains a unique key, document size (in bytes), field names and values, and

document location in the file system.

- **DAT File.** This is another intermediate file created in the process of building a Verity collection. The DAT file is a text file named input.dat that contains all the information from the documents that will be searchable, but not returned in the results list. This information is stored in zones. Zones are specific regions of a document to which searches can be limited.
- **Style Files.** These files describe a set of configuration options used to create the indexes associated with a collection.
- **MKDVK.** This is Verity's command-line tool. It is used for several tasks:
 - To index a collection
 - To insert new documents into a collection
 - To perform simple maintenance tasks, like purging and deleting a collection
 - To control indexing behavior/performance

The various pieces comprising the Verity product are discussed in more detail below.

Verity File Locations

Verity schema files are stored on a file server at the following locations:

```
$PS_HOME\data\search\<<index_name>\<db_name>\<lang_cd>\style
```

Verity collection files will be stored on a file server at the following locations:

```
%ps_home%\data\search\<<index_name>\<db_name>\<lang_cd>\<subdir>
```

Where the following conditions exist:

- The <db_name> is the name of the data base
- The <index_name> is the name of the application (or portal name for portal registry) that the collection is serving
- The <lang_cd> is the PeopleSoft local (language) code. A single portal will have one collection per locale. These codes will come from PSLANGUAGES. This is currently the old language code such as CFR, DUT, ENG, ESP, FRA, INE, JPN, POR.
- The <subdir> in the following set:
 - Style: Contains the style files that define the collection schema
 - Parts: Contains the Verity collection index partitions
 - Topicdx: Contains the topics (precompiled queries for easy synonym processing)
- Other <subdir> directories include assists, morgue, pdd, temp, trans and work.

The .BIF and .DAT intermediate files can be placed in the <lang_cd> level of the directory tree. They can be deleted after construction or be retained for validation purposes.

The source style files are checked into source safe to be shipped with the product, and will rarely change. The style file most likely to change from application to application is style.ufl. The collection create encapsulated in command is as follows:

```
%PSVERITYDIR%\_nti40\bin\mkvdk -collection %coll% -create -style
%style_file_path% -charmap %codePage% -locale %locale%
```

- The `-locale` parameter specifies the locale code.
- The `-charmap` parameter specifies the code page name for the character set appropriate for the locale.
- The `%style_file_path%` parameter points to the source style files, and is defined as follows:

```
set style_file_path=%PS_HOME%\DATA\search\<index_name>\<db_name>\<lang_cd>\style
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.ddd
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.dft
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.did
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.ngm
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.pdd
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.prm
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.sfl
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.sid
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.ufl
```

```
c:\ptdvl\DATA\search\Portal\ptdmo\FRA\style\style.wld
```

^	^	^	^	^	^	^	^
							various style files
							style directory
							Language code
							Database name
							Index name
							Search data directory
							DATA directory

```
%ps_home%
```

The command to load the collection from a bulk insert file is:

```
%PSVERITYDIR%\_nti40\bin\mkvdk -collection %coll% -bulk -insert %coll%\input.bif
```

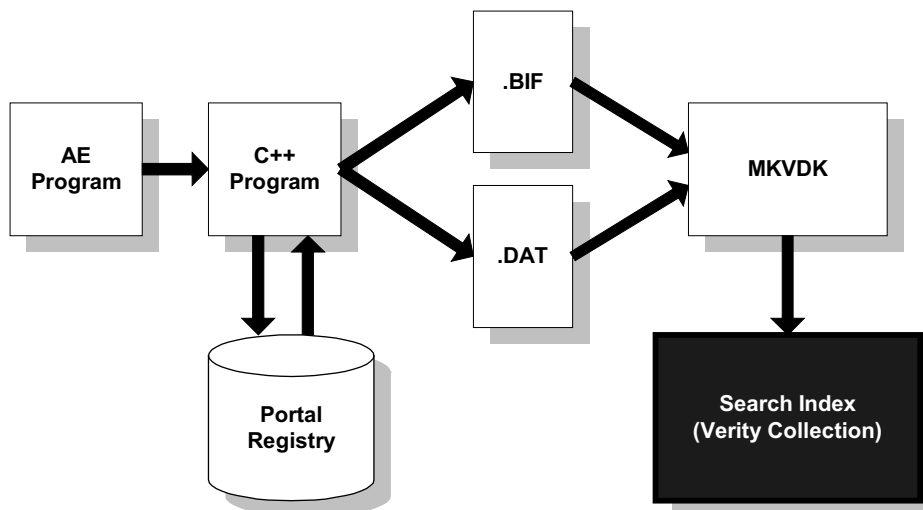
Building a Portal Search Index

To enable accurate, high-performance searches through the portal, the search engine must reference a comprehensive, up-to-date search index. Furthermore, to be useful, the search index must be easy to create and maintain, as content is likely to change frequently within the portal registry.

Building a Search Index for a Portal Application

If you want to build a search index for a portal application, the process is fairly easy because data in the portal registry is already in a known format which can be used to build the search index. Essentially, you just select to build the index from the Portal Administration pages, and the index gets built. Of course, there are number of steps that take place to build the index. But since the data is a known format, a number of PeopleSoft processes can take the manual work out of building the Verity index.

When you select to build a search index from the Portal Administration pages, an Application Engine program calls a C++ program that reads from the portal registry. The C++ program then generates the BIF and DAT files, which MKVDK takes as input to creating the search index. The illustration below represents this process.



Building a search index from the Build Search Index page in Portal Administration

The procedure below leads you through the exact steps required to build the search index from the Portal Administration pages.



Each time you build the search index you are overwriting the existing search index.

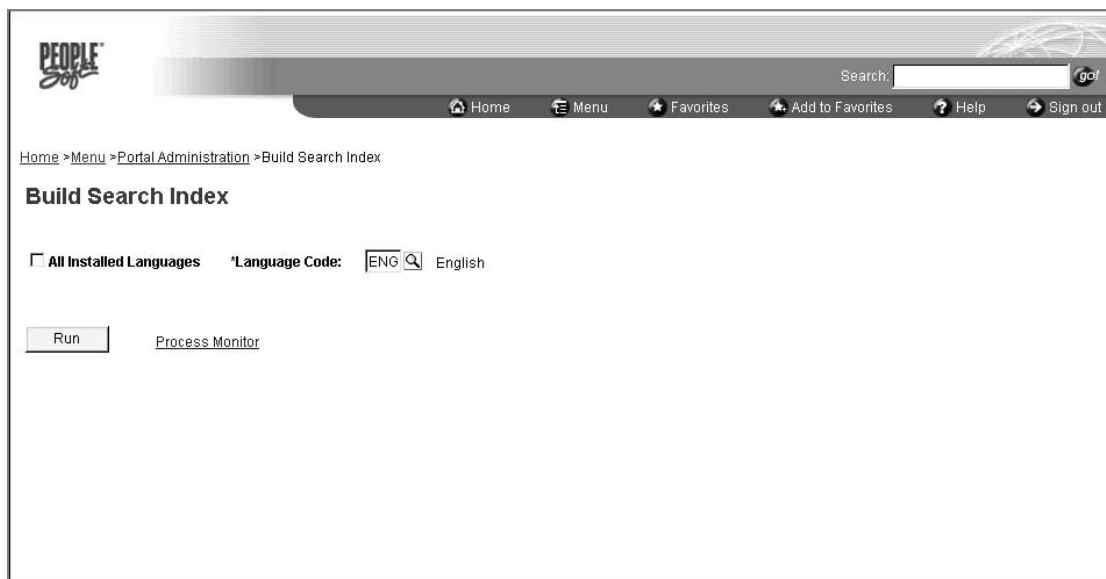


For more information on the PeopleSoft Search API and the PeopleSoft Portal Registry API, refer to the PeopleCode documentation.

To build a search index

1. Click the **Menu** button and select **Portal Administration, Build Search Index**.

The following page is displayed.



The Build Search Index page

2. Select whether or not to build the search index for **All Installed Languages** or not.

The default is to *not* build the index for all installed languages. Change this setting if necessary.

3. Select the **Language Code** for which you want to build the search index.

The language code defaults to the base language.

4. Click **Run** to launch the build process.

When you click **Run**, Process Scheduler initiates an Application Engine program named PORTAL_INDEX. You can click the **Process Monitor** hyperlink to view the status of the index build.

Submitting Search Requests

Submitting search requests in the PeopleSoft Portal is quite simple. This section discusses the basic procedure for executing a search.

To submit a search request

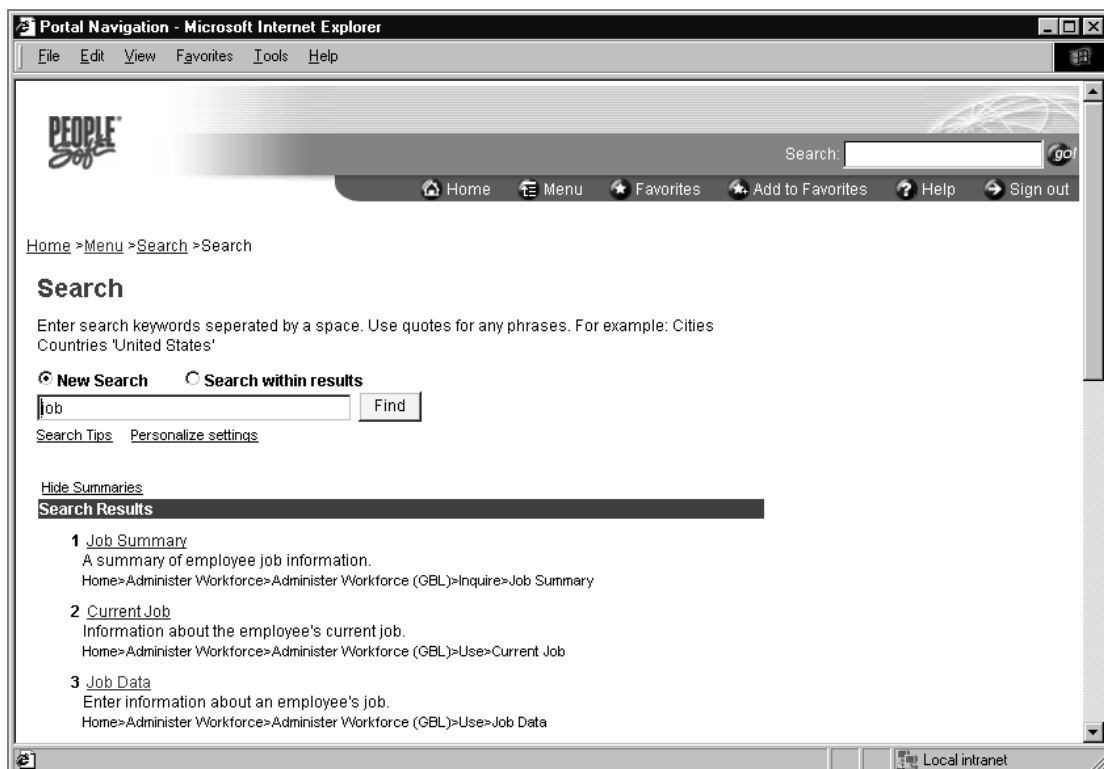
1. Enter a search string in the portal header and click **go!**



Submitting a search request

The search request is *not* case-sensitive. Any query sting you enter will be converted to all upper-case before it is submitted to the search engine. This helps to avoid mixed-case searches that return no results.

After you click **go!**, the portal processes your request and displays the search results page. If you click **go!** without having entered a query string, the search results page displays no results.



The Search Results page

The search results page changes the case of the text you enter to all upper case characters. It also filters out expired and hidden content references and ensures that you only see results for which you have been granted appropriate PeopleSoft security access.

The following fields are displayed for each content reference returned by the search results page:

- Content reference label (a hyperlink that takes you directly to the content reference)
- Description
- Path

There are a number of user-customizable settings that you can set for the search results page. For example, you can choose to hide the content reference descriptions by clicking **Hide Summaries** on the search results page or on the Customize Settings page.



For more information on customizing search settings, see Personalizing Search Settings.



In PeopleTools 8.12, in order to optimize performance, the total number of results is not calculated or displayed.

2. If necessary, refine your search or submit a new search request.

It's possible that your original search request either returned too many results to be practical for you to sort through, or that you want to try a different search altogether. To refine your search results, you can perform a search *within* the search results. To do this, click **Search within results**, enter a search query, and then click **Find**. A subset of the original search results will be displayed. (In this case, PeopleCode appends an <AND> operator, along with your new search request, to your original search request). To create a new request altogether, click **New Search**, enter a search query, and then click **Find**.

3. Select the content reference(s) for which you were searching.

Click the hyperlink corresponding to the desired content reference. The portal then navigates you directly to the content reference you select.

What Happens When a User Clicks “Go!”

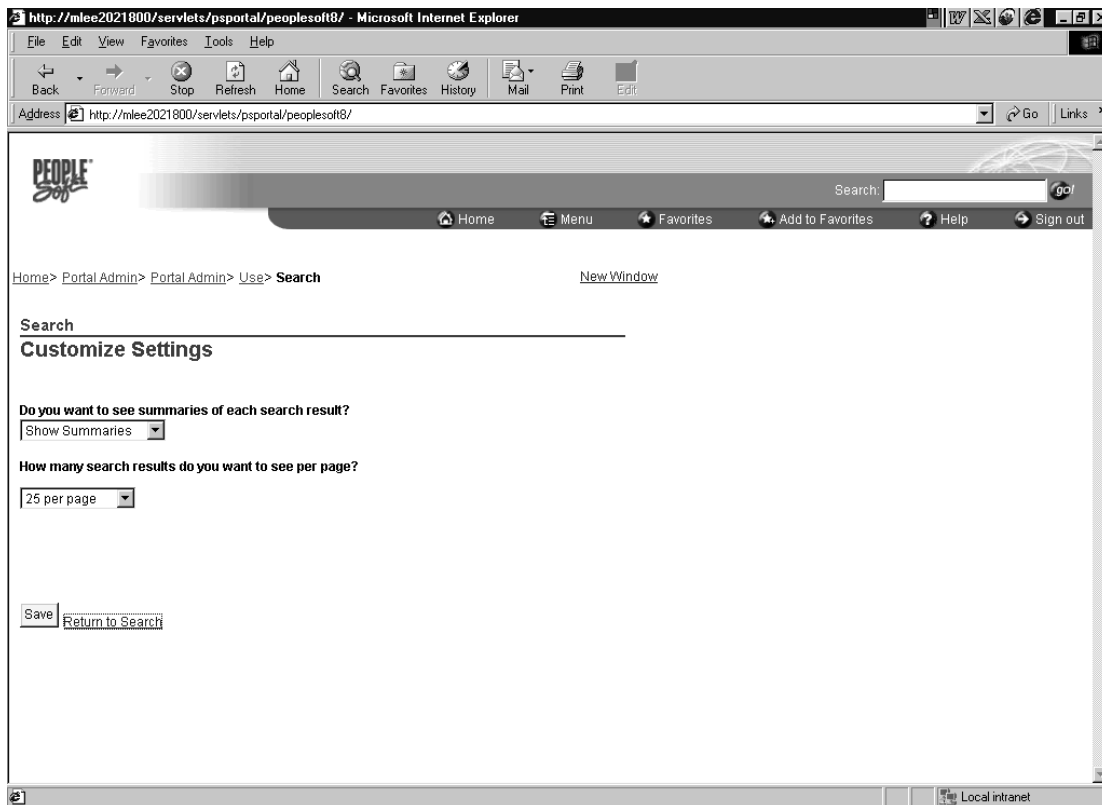
A user submits a search request by entering a search string into the search input form field in the portal header. The <form action=...> in the portal header is generated at runtime to link to a PIA page, and a Java script submits the form. The query string is passed to the Search API as a parameter named PortalSearchQuery to find matching results. Those results are filtered for security through PeopleCode by the Portal Registry API. The search results page echoes the original query string, and displays a list of content references that match the request. If the user clicks the *Go!* button but does not enter a search query, the search results page displays without any results.

The search results page performs the following steps:

- **Changes the case of the entered text to all upper case characters.** By default, the Verity search engine searches for all mixed-case variations when a query string is entered in all lower case or in all upper case. However, search queries entered in mixed-case automatically become case-sensitive. (For example, a query on *Apple* behaves as if the user had specified *Apple*—which would find only the precise string *Apple*—while a query on *apple* finds *APPLE*, *Apple*, and *apple*.) But the portal makes one important change: It changes the case of the query string to all upper-case, prohibiting users from truly executing case-sensitive searches. This avoids situations where mixed-case searches would otherwise return no results. On the search results page, however, the original case is echoed back to the user.
- **Formats the query string to pass to the Search API.** This includes filtering out expired and hidden content references, and content references that are not valid yet.
- **Calls the Search API.** This returns the query results.
- **Calls the Portal Registry API.** This is done to apply security filtering to the results. Security is applied in PeopleCode by checking the “Authorized” property.
- **Formats and displays search results.** This completes the user’s search request.

Customizing Search Settings

You can use the Customize Settings page to set whether or not summaries are displayed for each search result found, as well as the number of search results you’d like to view per page.



The Customize Settings page

Search Tips

The following tips may be useful to you when searching the portal.

Phrases: Use single quotes or double quotes around words that make up a phrase. For example:

`'Stock Option'`

Using single or double quotation marks: The rules for single and double quotes are as follows. The “stemming” feature is enabled for single quotes, but not for double quotes. Therefore, using double quotes means only the exact words will be searched for. For example, a recent search for 'definitions' returned 25 search hits, including results for both *definition* and *definitions*. A search for "definitions" yielded eight hits with only the exact phrase *definitions* included.

All words: Use an “and” to specify that all words must appear in the results. For example:

`'Stock Option' and grant`

Any words: Use an “or” to specify that any word must appear in the results. For example:

`'Stock Option' or bonus`

Limitations to Portal Search Functionality

- Verity does not run on IBM 390.
- Verity collections must reside on the PeopleSoft application server or be accessible from it through a shared drive.
- A maximum of one million documents is recommended.
- Verity collections are most efficient if you index large groups of data, rather than indexing one or two documents at a time. Small updates degrade the index and necessitate running a Verity-supplied cleanup utility.
- Asian locales are only available on Windows NT and Sun Solaris application servers.

CHAPTER 10

Miscellaneous Portal Information

This section contains miscellaneous information regarding portal development and implementation. If you are having issues that you're not sure how to resolve, consult this section for assistance.

Upgrade/Copy Functionality in Application Designer

For PeopleTools 8.12, Application Designer supports full Upgrade/Copy functionality of the portal registry.



For more information on using Application Designer's Upgrade/Copy functionality, see Upgrading with Application Designer. You may also refer to the *PeopleTools 8.12 Release Summary* for more detailed information on upgrade functionality and the portal.

Using the Portal Over HTTPS

If you're using the portal over HTTPs and using non-default ports for HTTP or HTTPS, you must edit the configuration.properties file for the web site to include the PortalHTTPPort and PortalHTTPSPort settings for the HTTP and HTTPS ports, respectively. Remove the "comment" from the two lines, and include the correct ports that the portal servlet will use for HTTP and HTTPS. This action must be repeated for the configuration.properties file for each web site that the portal servlet is serving.

Here's a sample section of the configuration.properties file:

```
#Define PortalHTTPPort and PortalHTTPSPort to be the ports that the portal
servlet runs on for HTTP and HTTPS.

#The values do not have to be defined if the default ports are used.

#PortalHTTPPort=80

#PortalHTTPSPort=443
```

HTTPS Support Through a Proxy Server

If you need your portal to issue requests using a proxy server, follow the instructions in this section.

The proxy server settings are controlled by the following Java system properties:

```
http.proxyHost  
http.proxyPort  
https.proxyHost  
https.proxyPort
```

To use a proxy server for the HTTP protocol, the following properties must be set:

```
http.proxyHost=<the proxy server's host name>  
http.proxyPort=<the proxy server's port>
```

To use a proxy server for the HTTPS protocol, the following properties must be set:

```
https.proxyHost=<the https proxy host's name>  
https.proxyPort=<the proxy server's port>
```

The method for setting the Java system parameters varies by web server (Apache or WebLogic). The properties must be passed to the Java interpreter as command line directives.

For Apache

If you're running under a single VM and allowing Apache to start the VM for you, edit the `jserv.properties` file to include lines like this:

```
wrapper.bin.parameters=-Dhttp.proxyHost=<proxyhostname>  
wrapper.bin.parameters=-Dhttp.proxyPort=<proxy port>  
wrapper.bin.parameters=-Dhttps.proxyHost=<proxyhostname>  
wrapper.bin.parameters=-Dhttps.proxyPort=<proxy port>
```

If you are starting the VM or VMs with a batch file, edit the Java command line as in this Apache example:

```
%JDKDIR%\bin\java.exe -Dhttp.proxyHost=<proxyhostname> -  
Dhttp.proxyPort=<proxyport> -Xms16m -Xmx16m org.apache.jserv.JServ  
"%JSERVDIR%\conf\JServ.8010.properties" > "%APACHEDIR%\logs\vm1.log"
```

For WebLogic

Alter the startWeblogic.cmd or startWeblogic.sh file, define the properties on the command line that starts the VM, like this:

```
set PROXY=-Dhttp.proxyHost=<proxyhostname> -Dhttp.proxyPort=<proxy port> -
Dhttps.proxyHost=<proxyhostname> -Dhttps.proxyPort=<proxy port>

%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
```

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 will be displayed in a frame (either in a frame template or in a non-template frame), then the form action cannot be "Post." The portal servlet will retrieve the page using "Get"—not "Post"—regardless of the form action. Therefore, form data will be lost if the form action was originally "Post." Making the form action "Get" resolves this problem.

Specifying a Pagelet Name Instead of a URL in a Template

You can specify a pagelet name in the template rather than specifying a URL. The syntax is as follows:

```
<IClientComponent Name="My Pagelet">

  <Source Product="Portal" Pagelet="MY_PAGELET">dummy</Source>

</IClientComponent>
```

If the portal server sees the "Pagelet" attribute in the Source tag, it will look up that pagelet name in the registry, and use the associated URL there.



The pagelet must be of *Remote* storage type, not *Local*. Like the TargetContent tag, the Source tag cannot be empty—a dummy string must be included between the <Source> and </Source> tags.

Posting from an HTML Template to a Frame Template

When posting data from a form in an HTML template, if the action of the form is in a frame template, then post data is lost. The solution for this is to make the frame source for the target be a special request to the portal to repost the form data. The HTML response to that request will

contain a form containing the posted field/value pairs in hidden input fields, and a script tag that submits that form. When the browser receives this request, the JavaScript will immediately submit the form to the original URL, causing the original desired post request to occur.

Actual implementation steps would be similar to the following:

1. Construct the frameset. The src of the target frame is the portal, with the querystring `cmd=framerepost&key=xxx` (or something similar).
2. Store the post data on the session object as a `PropertySet` (name/value pairs) for later retrieval, with the key. The key must uniquely identify the post data. The URL to post to can be kept in the `PropertySet`, or put on the querystring.
3. When the portal gets a `framerepost` request, it uses the incoming key to look up the stored `PropertySet`, then uses the `propertyset` to construct the HTML response, with the name/value pairs in hidden input fields. The form will also have a submit button. The HTML response will also contain an inline JavaScript (after the form), which submits the form.

Maintaining PIA State

If the PeopleSoft Internet Architecture (PIA) is installed on the same web server as the portal servlet, the portal servlet (specifically, the proxy process) sends a session cookie to PIA, so that they can share the same session. So, whenever a transition occurs between HTML and frame templates, PIA is using the same session as the portal was using to make the original request, so the state is not lost. The portal servlet will attempt to determine the correct cookie to send by matching the session ID value with the cookie value. If a match is made, then the cookie is passed to PIA.

In some cases, the above method will not work. If that is the case, then there is an optional configuration parameter that can be used for the session cookie name. If that parameter has been set in the configuration file, it will be used, and the above method will not be used to detect the cookie name automatically.

If PIA and the portal servlet are on different servers, or if the content in question is non-PIA, the PIA state still may be lost when transferring from HTML templates to frame templates.

Cookie Handling

How a cookie is handled, and when it's available, depends on the domain(s) and how they're set up. In the following table, four domain names are used, in order of least specific to most specific domains:

y.z

x.y.z

w.y.z

v.x.y.z

Domain	Cookie Availability
cookie set on x.y.z with null domain	x.y.z, not y.z
cookie set on x.y.z with domain .y.z	available on any domain ending with .y.z, by not y.z
cookie set on x.y.z with domain x.y.z	only available on x.y.z
cookie set on y.z with null domain	available on any domain ending with y.z, including y.z
cookie set on y.z with x.y.z domain	not set
cookie set on x.y.z with y.z domain	not set

Setting Hyperlinks in the Portal

There is a limitation of the `transfer()` and `redirectURL()` capability. Neither of these methods should be used when going between different template types (HTML and frame). Either of these methods will work when going between the same template type.

It was originally thought that the use of `RedirectURL` would solve the issue of going from an HTML template (like the homepage template) to a frame template transaction. This direction should be okay, but the use of `RedirectURL` does not do the correct proxying for a frame template to a HTML template.

From within a frame template to an HTML template the use of an HTML area is required. This HTML area must set the hyperlink to include the tag `"target=_top"` as well as the hand proxying with the use of the parameter `URL=`.

To get the hand proxying, use the following function found in `WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula` (or create your own function).

```
Function convertURLforPortal(&url) Returns string
    &PortalURI = GetPortalURI("PortalServlet");
    &url = &PortalURI | "?URL=" | EncodeURLForQueryString(&url);
    Return &url;
End-Function;
```

Here is an example of the code that sets the hyperlink value and the hyperlink label, where the `hyperlinkval` has already been proxied:

```
Rem This example is slightly adjusted for easier reading and it's from the
related links panel;

&LISTURL = "<A href=""" | &HYPERLINKVAL | ""target=_top>" | &HYPERLINKLBL |
"</A>";
```

Ensuring that Server Names and/or IP Addresses are Consistent

Once a user has logged on to a portal, it's possible to get the following error. This will occur if the server name and/or IP address you've defined for your portal is defined inconsistently in different locations:

```
Error getting portal (96,5) .
An error happened trying to get the definition for the portal.
```

In order for signon to work properly, the server name or IP address has to match in these three places:

- The URL used in the browser.
- The content provider.
- The pswebservername property in configuration.properties.

Single-Signon

When frame-based content is accessed across different machines, the user is forced to login again when accessing content on another web server. If the machines are on the same domain, single signon is supposed to work, and not display a login page.

Here are some different scenarios:

Test #1	Test #2
AppServer/WebServer #1 (Machine 1)	AppServer/WebServer #1 (Machine 1)
AppServer/WebServer #2 (Machine1)	AppServer/WebServer #2 (Machine 2)
Target content with html template	Target content with html template
Test Result: Works	Test Result: Works
Test #3	Test #4
AppServer/WebServer #1 (Machine 1)	AppServer/WebServer #1 (Machine 1)
AppServer/WebServer #2 (Machine1)	AppServer/WebServer #2 (Machine 2)
Target content with frame template	Target content with frame template
Test Result: Works	Test Result: Fails (multiple logins needed)

Single signon should work in frames/frame templates **if the URL's have the same domain**, as shown in the examples below:

```
xxx.peoplesoft.com
```

```
yyy.peoplesoft.com
```

It's not enough for the machines to be in the same domain—this has to be specified on the URL. So you'd have your portal content provider on xxx.peoplesoft.com and register your other content provider on yyy.peoplesoft.com.

Order of Precedence of Refresh Tags

The order of precedence of refresh tags in pagelets is as follows:

- Template
- Target content
- Pagelet

Among pagelets, the first one in the HTML to include a meta refresh tag will "win" (that is, subsequently found refresh tags will not be included in the HTML).

Using a Hardware SSL Accelerator

When the portal is being run through a hardware SSL accelerator device, it will probably be necessary to set the **portalUseHttpForSameServer** property in the configuration.properties file to "true." Setting this property to "true" may result in reducing the amount of time it takes to make a http request to the portal.

iQuery Pagelets

In order to ensure that your iQuery pagelets work properly, you need to put some "dummy" text into the iframe block of the HTML, like this:

```
<iframe frameborder=0 marginheight=0 marginwidth=0 name=iframeptest noresize  
scrolling=no src="http://www.yahoo.com">dummy</iframe>
```

If there is no "dummy" text in the iframe block, then the portal servlet assumes that it is empty and ignores it.

Linking Back to the Portal from a Different Database

You may have a need to create a page that links back to the portal homepage from a database other than the one which stores the portal. For example, you may have a pagelet on a portal homepage that links to another database, and the other database may have a link back to the portal. In these cases, you can use a custom portal header and the PortalServletURI parameter to pass the portal servlet's URI to the content. The pagelet would then be able to add this parameter to the link to the non-portal database. The page in the non-portal database would, in turn, use this parameter to construct its link back to the portal. A PortalURI parameter is also available.

Headers are passed to all content (target content and pagelets). Parameters are passed to pagelets only (as with all other special portal parameters).

Controlling the “Error Getting Content” Message

When a pagelet cannot be retrieved, an error message is displayed that resembles the following example.



Displaying the “Error getting content” message

Template developers have an option that enables them to specify whether they want this error displayed or not. If the error is not displayed, then the pagelet just “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, as follows.

```
<IClientComponent Name="Loser" DisplayError=false>

    <Source Product="">http://www.InvalidServer/InvalidPage.html</Source>

</IClientComponent>
```

If the portal servlet finds DisplayError=false, it will not display the error, but will add the empty pagelet comment tags. If DisplayError=true or if the attribute is not found, the error message is displayed.

Note that this feature works best in cases where the HTML for the template is able to flow around the empty content properly.

Requirements for Portal Help to Function Properly

There are a couple of requirements for ensuring that the Help link on the portal header works correctly. These are requirements for using the delivered IScript_UniHeader_Frame header pagelet.

The portal header finds the correct URL for the help link as follows. First, it looks for a form called "main" in a frame called "TargetContent." It then uses the value of the hidden form field "ICHelpUrl" as the URL for the help link. Thus, in to ensure that the correct URL is found, you must adhere to these requirements:

- The target content must be in a frame called “TargetContent.” The frame set should resemble this example:

```
<FRAME name=TargetContent...>
```

- In most cases, the URL of the target content should not contain the "target" querystring

parameter. However, if it does, the value of “target” must be "main." Note that this requirement refers to the PIA querystring parameter "target," and not the anchor tag, form tag, or base tag HTML attribute of that same name.

Overriding Page Encoding

It's possible to override the character set encoding for any given page registered in the portal. You can specify 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.

Events Requiring Re-Saving Personalized Homepages

There are some typical events that require re-saving personalized homepages before the desired change takes effect. This section is intended to document some of these events. Currently, there is only one event in this section:

- Resetting the help URL.

CHAPTER 11

Pagelet SDK Information

PeopleSoft has made available a portal pagelet Software Development Kit (SDK). The following table lists the components delivered with the SDK, an overview of their usage, and their location.

Component	Description	Path
Portal Pagelet – Samples	Portal Home Page (pagelet) uses iPage/iScript with JavaScript, HTML, Business Interlink, or URL to render pagelet display and in SDKDB development database	pspagelet\src\javascript\samples pspagelet\src\HTML\samples pspagelet\src\XML\samples pspagelet\bin\client\winx86\InterfaceDrivers

Portal Pagelet Samples

There are a total of five pagelet code samples delivered with the pagelet SDK, to be found in the “samples” directories shown in the table above.

- The first sample shows registration of an external URL to be called directly from a PeopleSoft pagelet.
- The second sample shows a PeopleSoft iScript calling a JavaScript that will render the HTML to be displayed.
- The third sample shows a PeopleSoft iScript calling an HTML object to render the HTML within the pagelet.
- The fourth sample is a PeopleSoft Internet Architecture page built upon a standard PeopleSoft application database component (panel group).
- The fifth and final sample contains a call to the Business Interlink PSHttpEnable to return the information from a remote site that is formatted into the HTML display.



For more information on this SDK, refer to the Integration SDK document available on Customer Connection.

Index

A

- absolute URLs 3-3
- anchor tags 3-4
- application portal 1-9
- architecture 1-2, 1-7
- attributes 2-5

B

- base portal data folder 6-24
- breadcrumbs 7-3

C

- CD-ROM
 - ordering xi
- changing the lookandfeel of a portal 6-28
- characteristics of portals 1-1
- content providers
 - managing 6-17
 - non-PeopleSoft content providers 4-11
- content providers and content provider
 - properties 2-5
- content references
 - adding, deleting, editing 6-7
 - security 6-11
 - setting permissions 6-11
- content references and content reference
 - properties 2-4
- converting anchor tags 3-4
- converting form tags 3-4
- converting JavaScript 3-5
- converting URLs 3-3

D

- design considerations for pagelets 5-11

F

- favorites 7-4
- folders
 - adding, deleting, editing 6-3
 - security 6-6
 - setting permissions for 6-6
- folders and folder properties 2-3

- form tags 3-4
- frame-based templates 4-11

G

- general characteristics of portals 1-1
- general portal settings 6-16

H

- HTML objects that control the portal 6-28

I

- importing menu groups into the registry 6-18
- inheritance of style classes 4-10
- iScripts
 - designing pagelets based on iScripts 5-4

J

- JavaScript 3-5

L

- lookandfeel of a portal 6-28

M

- managing portals 6-1
 - folders and content references 6-3
- menu groups
 - importing 6-18
- menu import 6-18
- menu navigation 7-1
- my profile 7-5

N

- narrow pagelets 5-2
- navigation 1-5, 7-1
 - breadcrumbs 7-3
 - favorites 7-4
 - my profile 7-5
 - three-column navigation 7-2
- navigation features 7-1

non-PeopleSoft content providers 4-11

P

page assembly process 3-2

pagelets 5-1

design considerations 5-11

designing pagelets based on PIA pages 5-1

developing 5-1

iScript-based pagelets 5-4

narrow 5-2

wide 5-2

PeopleBooks

CD-ROM, ordering xi

printed, ordering xi

permissions

for content references 6-11

for folders 6-6

portal

application portal 1-9

architecture 1-2

basics 1-1

general characteristics 1-1

navigation 1-5

portal servlet 1-5

registry 1-4, 2-1

single signon 1-10

terminology 1-12

types 1-8

portal administration 6-1

Portal Administration pages 6-1

portal HTML objects 6-28

portal servlet 3-1

basics 1-5

setting up 3-1

proxying 3-3

Q

queries

registering 6-30

R

registering queries 6-30

registry 2-1

basics 1-4

content providers and content provider

properties 2-5

content references and content reference

properties 2-4

folders and folder properties 2-3

maintaining 2-6

properties 2-3

registering queries 6-30

registry architecture 2-1

relative URLs 3-3

S

security 2-5

servlet 3-1

single signon with the portal 1-10

style class inheritance 4-10

T

template pagelet tags 4-10

template pagelets 4-9

templates 4-1

and style classes 4-10

default portal template 6-16

designing 4-1

developing 4-9

dynamic templates 4-2

examples 4-2

frame-based templates 4-1

HTML-based templates 4-1

non-PeopleSoft content providers 4-11

page-based templates 4-1

static templates 4-2

template pagelet tags 4-10

template pagelets 4-9

types of templates 4-1

terminology 1-12

three-column navigation 7-2

types of portals 1-8

W

wide pagelets 5-2