ORACLE
PeopleSoft

PeopleTools 8.62: Portal Technology

December 2025

ORACLE

Documentation Accessibility

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

Access to Oracle Support

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

Contents

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

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

Hosted PeopleSoft Online Help

You can access the hosted PeopleSoft Online Help on the Oracle Help Center. The hosted PeopleSoft Online Help is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support. The hosted PeopleSoft Online Help is available in English only.

To configure the context-sensitive help for your PeopleSoft applications to use the Oracle Help Center, see Configuring Context-Sensitive Help Using the Hosted Online Help Website.

Locally Installed PeopleSoft Online Help

If you're setting up an on-premises PeopleSoft environment, and your organization has firewall restrictions that prevent you from using the hosted PeopleSoft Online Help, you can install the online help locally. Installable PeopleSoft Online Help is made available with selected PeopleSoft Update Images and with PeopleTools releases for on-premises installations, through the Oracle Software Delivery Cloud.

Your installation documentation includes a chapter with instructions for how to install the online help for your business environment, and the documentation zip file may contain a README.txt file with additional installation instructions. See *PeopleSoft 9.2 Application Installation* for your database platform, "Installing PeopleSoft Online Help."

To configure the context-sensitive help for your PeopleSoft applications to use a locally installed online help website, see Configuring Context-Sensitive Help Using a Locally Installed Online Help Website.

Downloadable PeopleBook PDF Files

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

Common Help Documentation

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

- Application Fundamentals

- Using PeopleSoft Applications

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

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

Field and Control Definitions

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

Typographical Conventions

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

Typographical Convention	***Description***
Key+Key	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For **Alt+W,** hold down the **Alt** key while you press the **W** key.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

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

ISO Country and Currency Codes

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

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

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

Region and Industry Identifiers

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

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

Region Identifiers

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

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

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

- USF (U.S. Federal)

- E&G (Education and Government)

Translations and Embedded Help

PeopleSoft 9.2 software applications include translated embedded help. With the 9.2 release, PeopleSoft aligns with the other Oracle applications by focusing our translation efforts on embedded help. We are not planning to translate our traditional online help and PeopleBooks documentation. Instead we offer very direct translated help at crucial spots within our application through our embedded help widgets. Additionally, we have a one-to-one mapping of application and help translations, meaning that the software and embedded help translation footprint is identical—something we were never able to accomplish in the past.

Using and Managing the PeopleSoft Online Help

Select About This Help in the left navigation panel on any page in the PeopleSoft Online Help to see information on the following topics:

- Using the PeopleSoft Online Help.
- Managing hosted Online Help.
- Managing locally installed PeopleSoft Online Help.

PeopleTools Related Links

PeopleTools 8.62 Home Page

PeopleSoft Search and Insights Home Page

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

PeopleSoft Online Help

PeopleSoft Information Portal

PeopleSoft Spotlight Series

PeopleSoft Training and Certification | Oracle University

My Oracle Support

Oracle Help Center

Contact Us

Send your suggestions to psoft-infodev_us@oracle.com.

Please include the applications update image or PeopleTools release that you're using.

Follow Us

Icon	*Link*
	Watch PeopleSoft on YouTube
	Follow @PeopleSoft_Info on X.
	Read PeopleSoft Blogs
	Connect with PeopleSoft on LinkedIn

Chapter 1

Getting Started with Portal Technology

Portal Technology Overview

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

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

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

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

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

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

Portal Technology Implementation

To implement PeopleTools portals, you need to:

- Design and configure the portal environment.
- Install PeopleSoft Pure Internet Architecture.
- Configure your web profile.
- Configure application security.
- Set up the portal.
- Configure performance features, such as caching.

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

Related Links

"PeopleTools Overview" (Getting Started with PeopleTools)

Designing and Configuring the Portal Environment

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

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

During implementation, you should research and understand how to:

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

Installing PeopleSoft Pure Internet Architecture

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

Step	*Reference*
Run the PeopleSoft Pure Internet Architecture setup program for your database.	See *PeopleSoft 9.2 Application Installation* for your database platform.

Configuring Your Web Profile

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

Step	***Reference***
Configure your selected web profile.	See Configuring Web Profiles. See “Understanding WebLogic” (System and Server Administration).

Configuring Application Security

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

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

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

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

Setting Up the Portal

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

Step	***Reference***
Manage portal definitions.	See Administering Portal Definitions.
Define folders and set folder security.	See Administering Folders.
Define content references, create related links, and set content reference security.	See Administering Content References.
Add templates and set template security.	See Managing Portal Objects.

Step	*Reference*
Create, configure, and secure homepages and dashboards.	See Understanding Homepages and Dashboard Pages.

Configuring Performance Features

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

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

Chapter 2

Understanding PeopleSoft Pure Internet Architecture

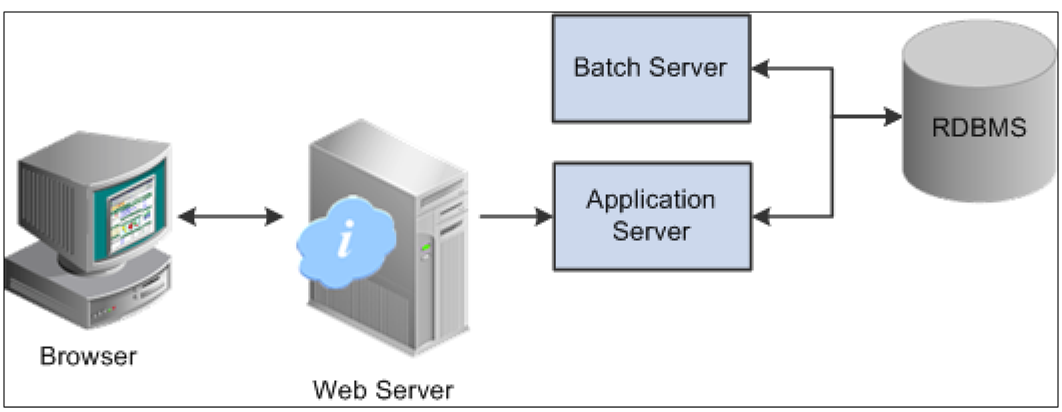
PeopleSoft Pure Internet Architecture Fundamentals

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

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

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

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



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

Database Server

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

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

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

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

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

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

Application Servers

This section discusses:

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

Application Servers

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

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

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

Related Links

Configuration and Deployment Options

Domains

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

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

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

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

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

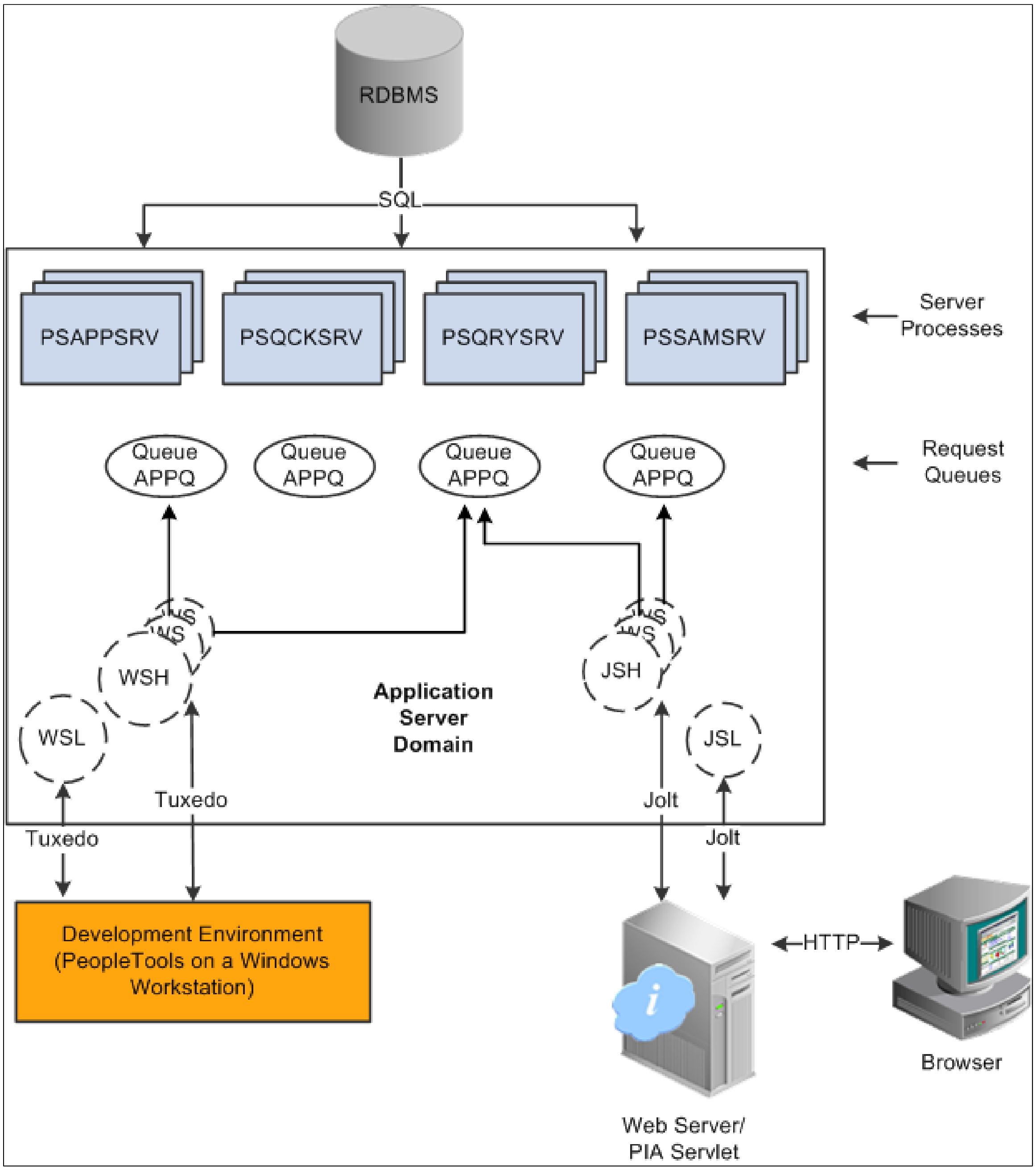
Starting an application server domain starts the set of server processes associated with that domain, such as PSAPPSRV, PSQCKSRV, and PSSAMSRV. Each server process establishes a persistent connection to a PeopleSoft database, and this connection acts as a generic SQL pipeline that the server process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

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

Listeners, Handlers, and Queues

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

The following diagram illustrates application server processes:



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

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

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

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

PeopleSoft Server Processes

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

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

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

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

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

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

- PSAPPSRV

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

- PSQCKSRV

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

- PSQRYSRV

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

- PSSAMSRV

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

- PSOPTENG

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

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

- PSMSGDSP
- PSMSGHND
- PSPUBDSP
- PSPUBHND
- PSSUBDSP
- PSSUBHND

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

Services

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

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

Related Links

“Using PSADMIN Configuration Files” (System and Server Administration)

Oracle Middleware

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

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

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

See *PeopleSoft 9.2 Application Installation* for your database platform.

Database Connectivity

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

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

Batch Server Environment

This section discusses:

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

Batch Server Environment

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

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

Related Links

“PeopleSoft Process Scheduler” (Process Scheduler)

Batch Server Support

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

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

PeopleSoft Process Scheduler and the Application Server

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

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

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

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

Web Server

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

This section discusses:

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

Server Software Elements

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

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

- Web services.

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

- Servlet engine.

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

- Java servlets.

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

PeopleSoft Servlets

The following PeopleSoft servlets reside on the web server:

- Portal servlet.

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

- Integration gateway servlet.

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

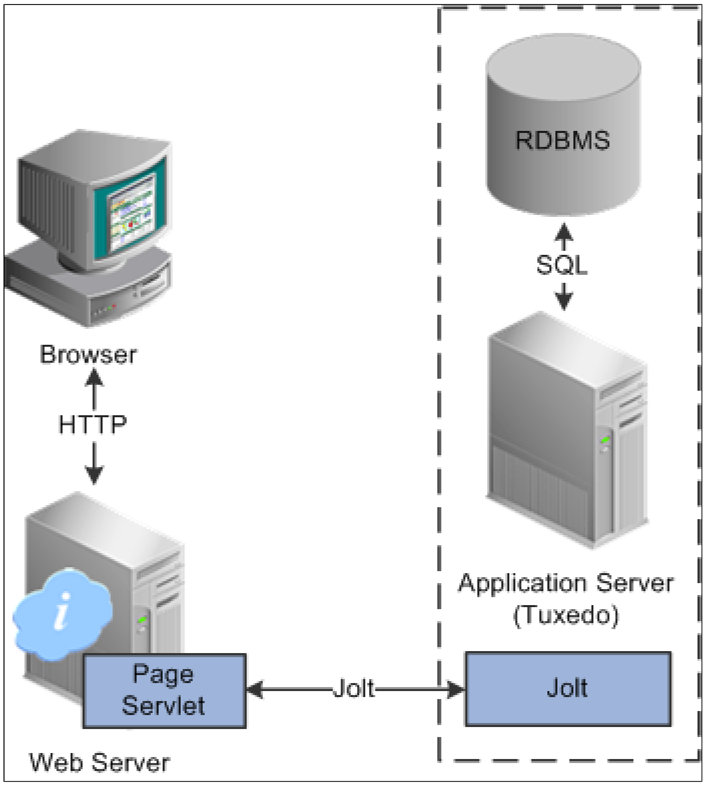
- Report repository servlet.

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

Oracle Jolt

The PeopleSoft servlets on the web server transmit requests and data through a connection to Oracle Jolt, which runs on the application server. Oracle Jolt extends Oracle Tuxedo capabilities to the communication layer between the web-based environment and the C++ environments. You configure the servlets to direct requests from the web server to a predefined Oracle Jolt port on the application server.

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



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

Web Browser

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

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

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

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

- HTML
- XML

- JavaScript
- Cookies

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

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

See PeopleTools Browser Compatibility Guide (Oracle Support Document 704492.1) on My Oracle Support for more information.

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

Web Browser Cookies Setting

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

To enable session cookies:

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

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

Configuration and Deployment Options

This section discusses:

- Configuration options.
- Deployment options.

Configuration Options

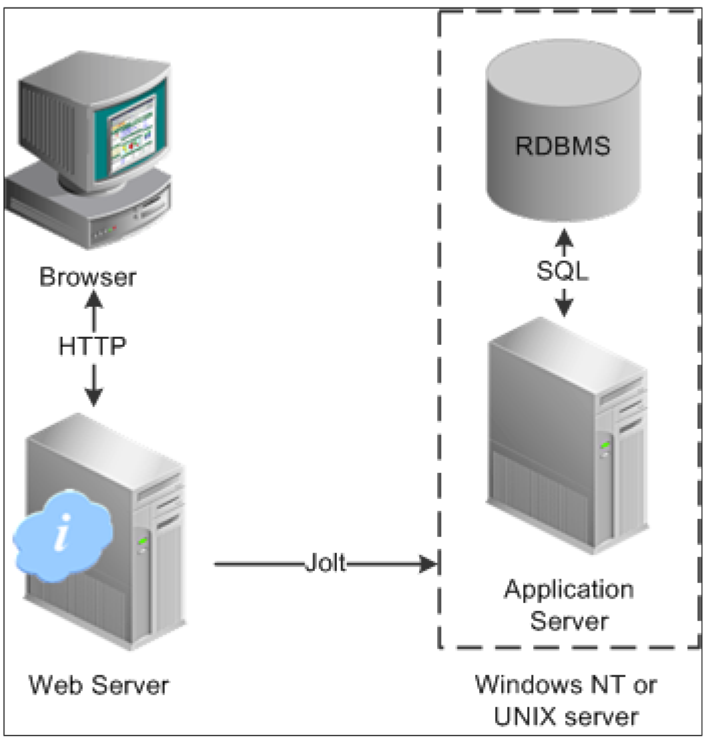
You can configure your environment to support either a physical or a logical application server configuration. In some cases, the PeopleSoft standard installation procedure recommends one or the other depending on the combination of database and operating system at your site.

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

Logical Application Server Configuration

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

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



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

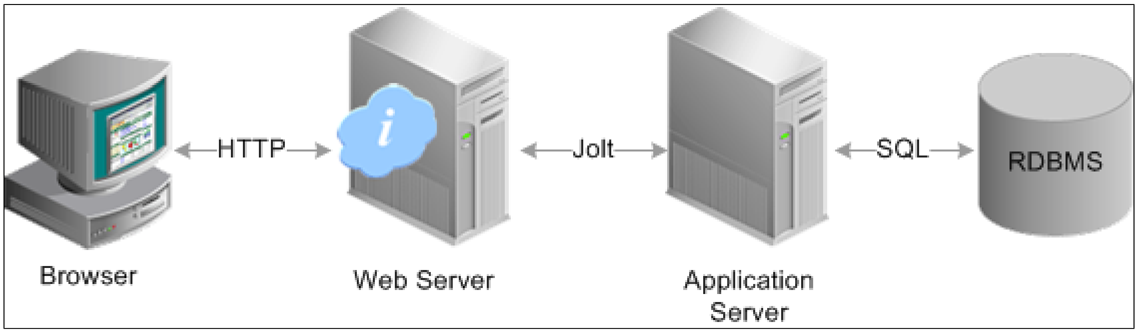
Although this diagram depicts the application server and the database server sharing the same machine, the web server could also reside on the same machine with both the application server and the database server. The only requirement is that each component be supported by the underlying operating system.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

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

Physical Application Server Configuration

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

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



Deployment Options

A variety of user deployment options exist.

PeopleTools Portal

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

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

Development Environment

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

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

Integration Technologies

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

- PeopleSoft Integration Broker.

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

- PeopleSoft Component Interfaces.

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

- PeopleSoft Business Interlinks.

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

Related Links

“Introduction to PeopleSoft Integration Broker” (Integration Broker)
“Understanding Component Interfaces” (Component Interfaces)

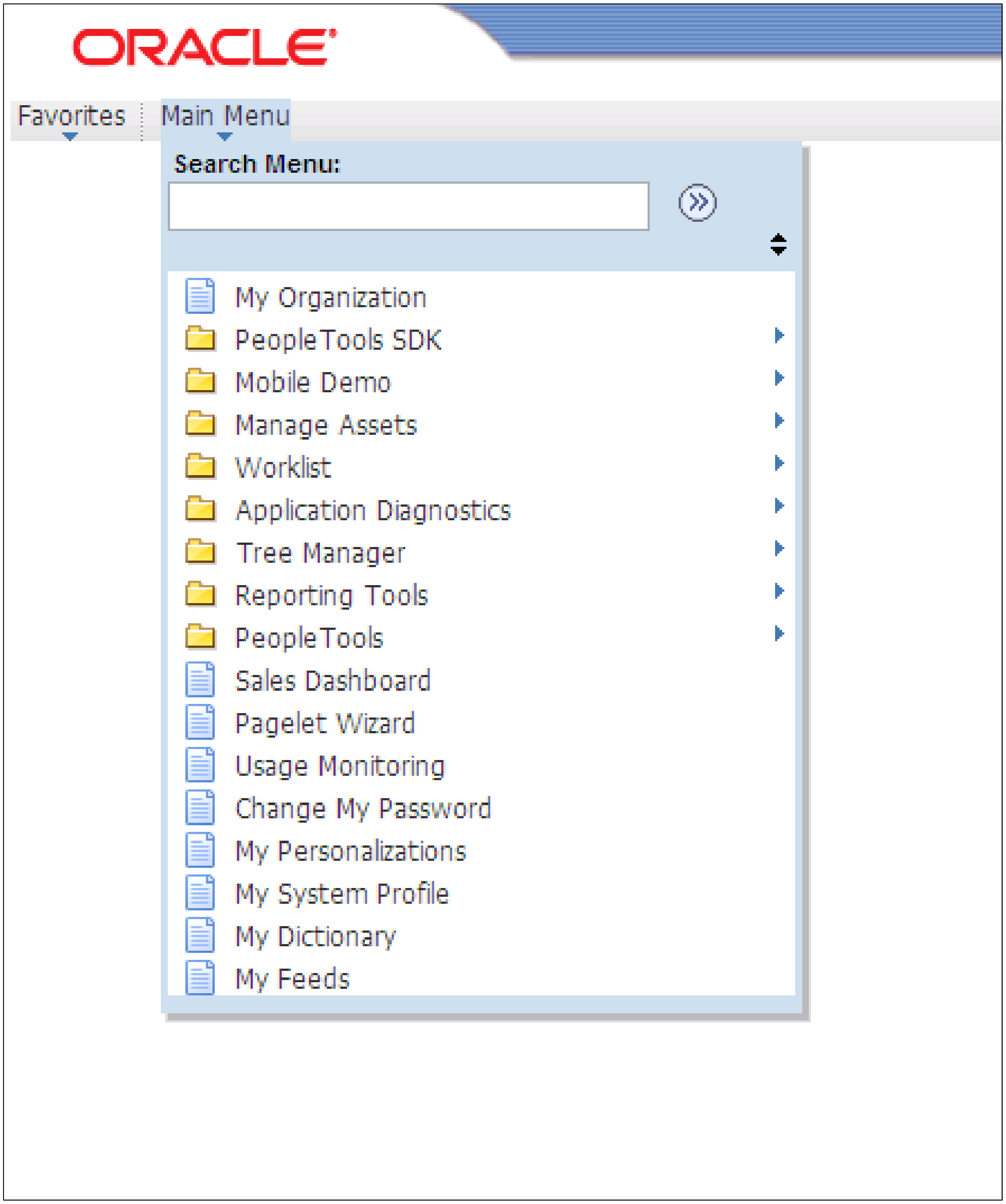
Chapter 3

Understanding Portal Technology

Portal Architecture

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

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



The main features of the portal architecture are:

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

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

Node Definitions

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

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

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

Related Links

"Node Class" (PeopleCode API Reference)
"Understanding Nodes" (Integration Broker Administration)

Portal Templates and Template Pagelets

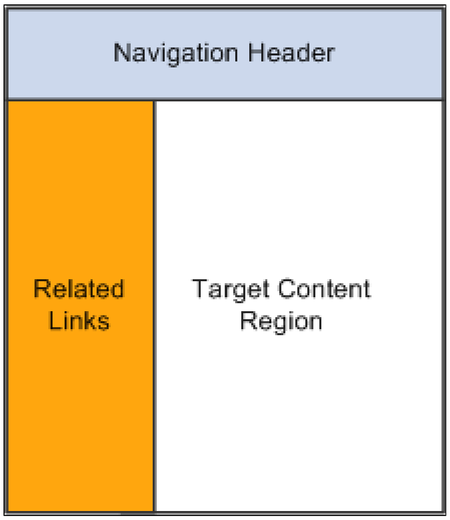
Developers create portal templates in PeopleSoft Application Designer. At runtime, a template is transformed into a web page by PeopleTools portal technology. Each template is made from various template pagelets.

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

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

Each portal template reserves space for a target page, which contains the specific HTML page that a user requested. For example, if a user is running a PeopleSoft Financial application, the page that the user is currently accessing appears in the target region, in addition to the other template pagelets displayed elsewhere on the page. The target region is typically the largest area of the template.

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



Navigation

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

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

The universal navigation header includes several links.

Term	***Definition***
Search	Enter a term or phrase in the **Search** field and click the **Go** button to search for a registered content reference within the system.
Home	Click to return to the homepage.
Worklist	Click to access the Worklist page, which provides summary information about all items on your personal worklist. This page also provides links enabling you to view additional details about the work, perform the indicated work, and reassign work items. This is an optional setting that your system administrator enables; it might not appear to all users. See “Sending and Receiving Ad Hoc Notifications” (Applications User’s Guide).

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

Related Links

“Understanding PeopleSoft Navigator” (Applications User’s Guide)

“Using the Universal Navigation Header” (Applications User’s Guide)

Portal Registry

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

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

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

Note: The PeopleSoft Administrator role and the Portal Administrator role give full access to all folders and content reference definitions in the portal registry—that is, a user with either of these roles can navigate to all menu folders and menu items in the portal registry even if the portal registry security settings do not specifically include this user or role. For more information on the PeopleSoft Administrator role, see “Defining the PeopleSoft Administrator Role” (Security Administration).

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

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

- **Homepage**
- **Navigation Collections**
- **Pagelets**
- **Template Pagelets**
- **Templates**

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

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

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

Maintaining the Portal Registry

Several tools exist to access and maintain the portal registry:

- Use the portal administration pages to manage portal and node definitions, folders, content references, templates, pagelets, navigation collections, and portlets.
- Use other portal administration pages to view menu security and structure.
- Use the portal registry API to programmatically maintain objects in the portal registry.
- Use the component registration wizard in PeopleSoft Application Designer to register and assign permissions to component-based content references.
- Use the Menu Import utility to import custom menu group definitions.

- Use the security synchronization process to reinstate the correct security relationships between objects in the portal registry after you copy a project that contains portal objects.

Related Links

Importing Menu Groups into the Portal Registry
"Understanding the Portal Registry" (PeopleCode API Reference)
"Understanding the Registration Wizard" (Application Designer Developer's Guide)
"Using the Portal Registry API" (PeopleCode API Reference)
Node Definitions
Defining Portal Nodes
"Node Class" (PeopleCode API Reference)

Portal Servlets

This section discusses:

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

Portal Servlets

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

The portal servlet performs these functions:

- Retrieves user-requested content.

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

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

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

- Provides a consistent user interface.

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

- Ensures that PeopleSoft-specific tags are processed correctly.

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

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

Page-Based Template Handling

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

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

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

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

See Proxying and URL Conversion.

Frame-Based Template Handling

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

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

iFrame-Based Template Handling

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

Portal Servlet Setup

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

See *PeopleSoft 9.2 Application Installation* for your database platform.

Page Assembly for Page-Based Templates

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

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

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

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

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

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

3. The portal servlet retrieves the appropriate template.

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

4. The portal servlet issues HTTP requests for content.

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

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

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

   If a style sheet is associated with the template, it is used. If not, the style sheet associated with the target content is used. Style sheets included in template pagelets are used if they are present.

Additionally, the portal servlet ensures that cookies and headers returned in the responses for the template pagelets are merged into the main response.

Portal URL Formats

This section discusses:

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

Basic Portal URL Format

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

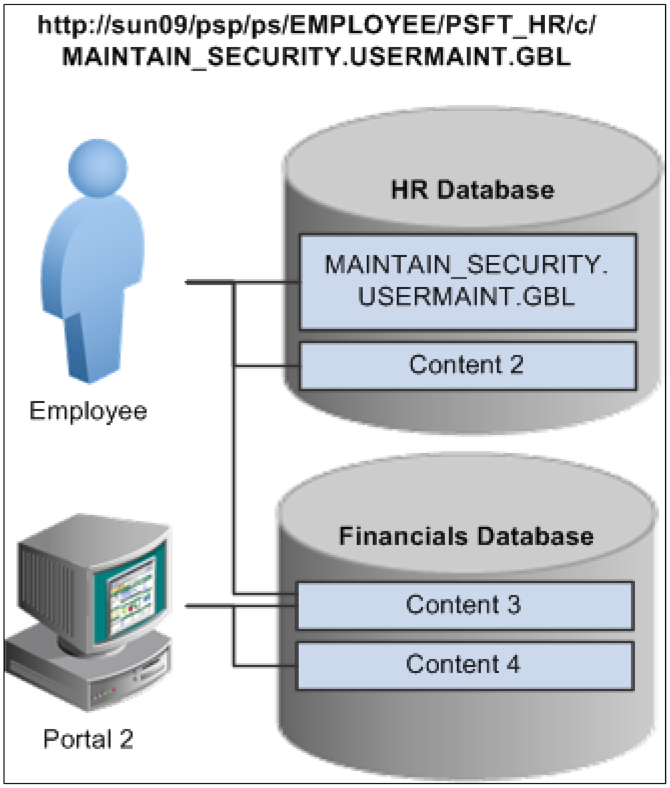
- The name of the portal through which content is being accessed.
- The node that hosts the content.
- The type and ID of the content.

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

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

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

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



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

Here's what a portal URL looks like:

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

The following table describes the different sections of the URL:

Section	***Description***
http://server/	Scheme (HTTP / HTTPS) and web server name. **Important!** The server name and port in this section must not exceed 30 characters in length. For example, *http://mybiglongservername.example.com:8080,* not including the *http://* prefix, is 36 characters—six characters too long.
servlet_name/	The name of the physical servlet that the web server invokes to handle the request.

Section	***Description***
site_name/	The site name specified during PeopleSoft Pure Internet Architecture setup. This enables you to set up multiple sites on one physical web server. The site name is ultimately mapped by the web server to the appropriate configuration.properties file. **Important!** The site name can include underscores (_), but an underscore cannot be followed by a numeric character or the string newwin (for example, *my_site_3* or *my_newwin_ site)*.
portal_name/	Name of the portal to use for this request. The portal definition contains metadata that describes how to present the content (template, pagelets, and so on).
node_name/	Name of the node that contains the content for this request.
content_type/	Type of the content for this request.
content_id	The identification of the content. The ID and type ensure that the correct content is retrieved.
?content_parm	Query string parameters (name value pairs) for the content.

URL Format for PeopleSoft Pure Internet Architecture Content Types

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

Content Type	***URL Format***	***Example***
Component	/c/*menu.component.market*/?Page=*page*&Action=*action* &*key_id*=*key _value*. . .	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/c/ MAINTAIN_ SECURITY.USERMAINT.GBL?page=view&view=narrow
Script	/s/*RECORDNAME. FIELDNAME. Event. Function*/?*parm_id*=*parm_value*. . .	http://sun09/psp/ps/EMPLOYEE/PSFT_HR/s/ WEBLIB_PORTAL. PORTAL_HEADER.FieldChange.Iscript _DoSomething
External	/e/?url=*URL*	http://sun09/psp/ps/EMPLOYEE/PSFT _HR/e/?url=http%2f%3a%3awww. example.com

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

URL Format for Unwrapped PeopleSoft Pure Internet Architecture Content

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

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

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

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

Pagelet URLs

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

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

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

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

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

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

System URLs

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

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

URL	*Description*
http://sun09/psp/ps/?cmd=logout	Closes the current session and returns to the sign-in page.
http://sun09/psp/ps/?cmd=login	Tells the servlet to return to the sign-in page.

Proxy Architecture and Relative URLs

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

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

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

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

Examples of Relative PeopleSoft URLs

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

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

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

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

Here is what the final URL looks like:

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

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

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

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

The absolute URL looks like this:

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

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

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

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

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

Related Links

Portal Technology Overview

Page-Based Template Proxying

This section discusses:

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

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

See Portal URL Formats.

Proxying and URL Conversion

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

When users sign in to a PeopleSoft portal, they sign in to a web server on which the portal servlet is running. The portal servlet processes all the HTML that isn't in the simple URL format, converting all URL references to point to the portal web server rather than the original URL. The original URL is still necessary to retrieve the requested content; it is stored in the new URL in the URL query string parameter.

The portal servlet proxies all links and all form actions in this manner. Additionally, it converts any relative URLs into absolute URLs.

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

Converting Relative URLs to Absolute URLs

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

The following example shows a relative tag:

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

The following example shows an absolute tag:

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

Anchor Tag Conversions

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

Old Anchor Tag

The following example shows an old anchor tag:

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

New Anchor Tag

The following example shows a new anchor tag:

```
<a src=http://portalserver/psp/ps/EMPLOYEE/NODE/e/?url=http%3a%2f%2fserver%2ftarget⇒

page.html%3fAction%3dNew>
```

Form Tag Conversions

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

Old Form Tag

The following example shows an old Form tag:

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

New Form Tag

The following example shows a new Form tag:

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

JavaScript Conversions

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

Chapter 4

Administering Portals

Understanding Portal Administration

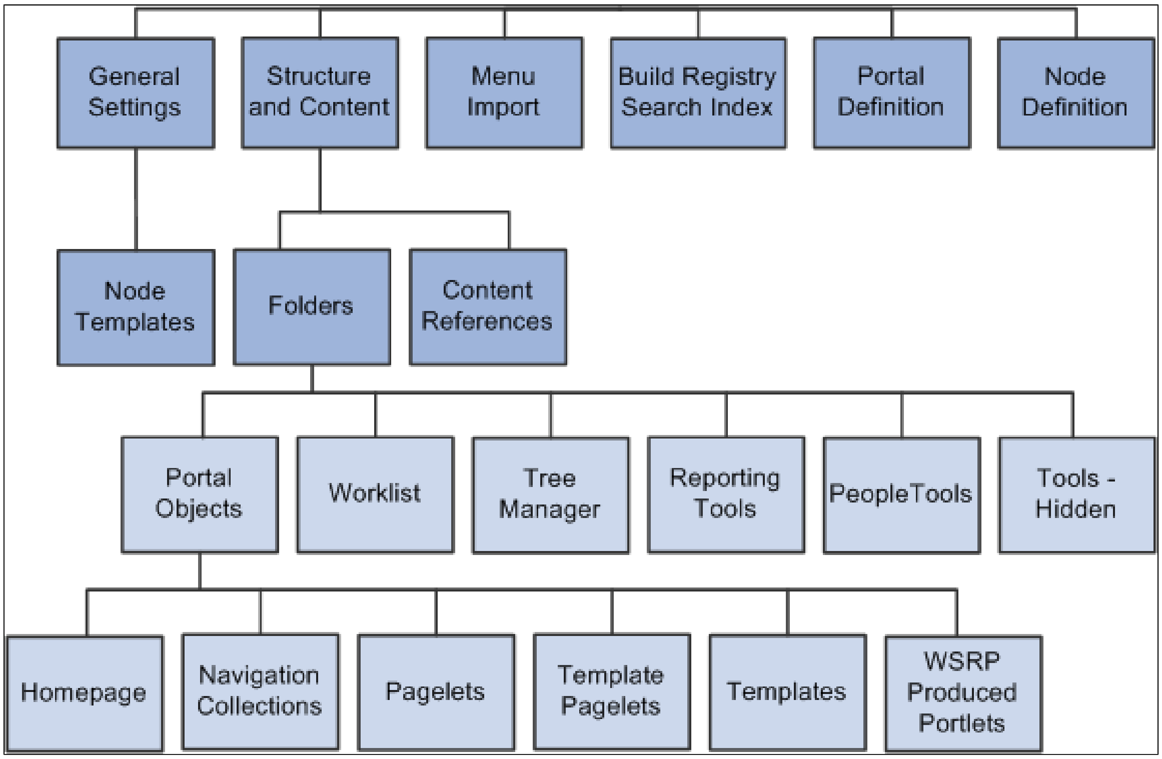
Every component delivered with PeopleSoft applications is registered as a content reference in the portal registry . Common administrative tasks include adding, deleting, and renaming these definitions in the portal. If you have portal administrator access, you can use the administration pages to manage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

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

You use the administration pages to:

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

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



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

Related Links

"Understanding Nodes" (Integration Broker Administration)

"Using the Portal Registry API" (PeopleCode API Reference)

Understanding Folder Navigation

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

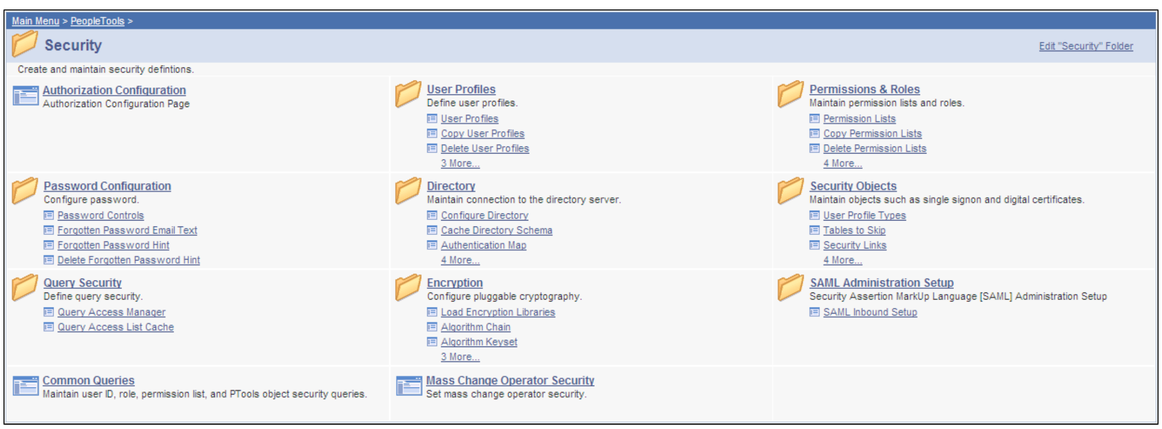
Important! Standard navigation pages (DFANs), custom navigation pages (CFANs), SmartNavigation, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see "Working with Fluid Pages and Controls" (Applications User's Guide).

Navigation Pages

Note these points about the characteristics of navigation pages:

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

The following example is the standard navigation page for the Security folder:



Traversing Menus Without Engaging Navigation Pages

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

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

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

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

Related Links

Adding and Editing Portal Definitions
Managing General Settings for Portals
Understanding Navigation Pages
“Understanding Permission Lists” (Security Administration)
“Understanding Roles” (Security Administration)

Common Elements Used in This Chapter

Field or Control	***Description***
Chart Navigation Object Name	At the portal definition and general settings levels, specify the portal object that creates the general appearance and character of SmartNavigation folders. The system default value is *PT_PORTAL_DEFAULT_CHART_NAV,* which is a hidden iScript-based content reference that renders the basic SmartNavigation page. At the individual folder level, specify an object that overrides the system-level chart navigation object. If you don't specify a value at the folder level, the portal uses the system-level chart navigation object.
Copy object	Click to access the Copy Object to a Different Portal Registry page, which enables you to copy an existing folder to a different portal registry. This button appears only when you're editing a folder.
Default Chart Navigation Page	Select to replace the standard navigation page with chart-based SmartNavigation page. This field is disabled by default. To enable this field, select a value in the **Data Source** field. If you select this check box, then the application uses *PT_ABN_ORGCHART* component as the default PeopleTools chart navigation page. If you deselect this check box, then you must enter your own value in the **Folder Navigation Object Name** field.

Field or Control	***Description***
Folder Navigation Object Name	At the portal definition and general settings levels, specify the content reference that the system launches when the user clicks a folder icon. The system default value is *PT_PTPP_SCFNAV _BASEPAGE_SCR,* which is a hidden iScript-based content reference that renders the standard navigation page. At the individual folder level, specify an object that overrides the system-level default. If you don't specify a value at the folder level, the portal uses the system-level folder navigation object. **Important!** You must use the definitions associated with the *PT_ABN_ORGCHART* component as the basis of any custom SmartNavigation folder object. **Note:** If you select a content reference that's hidden from portal navigation, the folder navigation occurs, but the menu name that you click remains highlighted.
Hide From Portal Navigation	Select to make this folder invisible to users.
Is Folder Navigation Disabled	Select to override the system-level default setting and disable folder navigation for individual folders. **Note:** If you disable and then re-enable folder navigation at the system level, this field is automatically cleared.
Is Folder Navigation Enabled	Select to enable folder navigation for all folders in the current portal registry or selected portal definition. When folder navigation is enabled, if the user clicks the folder name in the menu structure, the navigation page or a content reference that you specify in the **Folder Navigation Object Name** field appears in the target area.
Label	Enter the text to appear as the link for this folder.
Long Description	Enter a description for this folder or content reference. **Note:** In classic forms of navigation (which have been desupported), this description appears as hover text for the link for this folder or content reference. In fluid forms of navigation, such as the NavBar menu, no hover effects are displayed. In addition, no hover effects are displayed for links in fluid navigation collections.

Field or Control	*Description*
Name	Enter a name for the folder or content reference. **Note:** The name can consist of letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.
(Object) Owner ID	Select the owner of the portal object such as portal, folder, or content reference. The owner ID helps determine the team that created or modified the object. You can define unique owner IDs for your organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer. Do not use PeopleSoft owner IDs.
Parent Folder	The name of the current parent folder. This field is display-only.
Product	(Optional) Portal-aware applications can use this field for group processing. All PeopleTools objects have the product name *PT.*
Select New Parent Folder	Click to copy the folder to a new location within the current portal registry. This button appears only when you're editing a folder.
Sequence number	Enter a sequence number to define where this folder appears in its parent folder, and therefore where its associated menu entry appears within its parent menu. If you leave the sequence number blank, the folder is added to the top of the list of folders and is displayed in alphabetical order. You can change the sequence only one item at a time, and each time you do so, you must save your changes to see the result.
Valid from date and **Valid to date**	Enter a valid from date. The valid to date is optional and is omitted by default. These dates can be used for application-specific processing by portal-aware applications and are used by the portal navigation to determine visibility.

Administering Portal Definitions

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

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

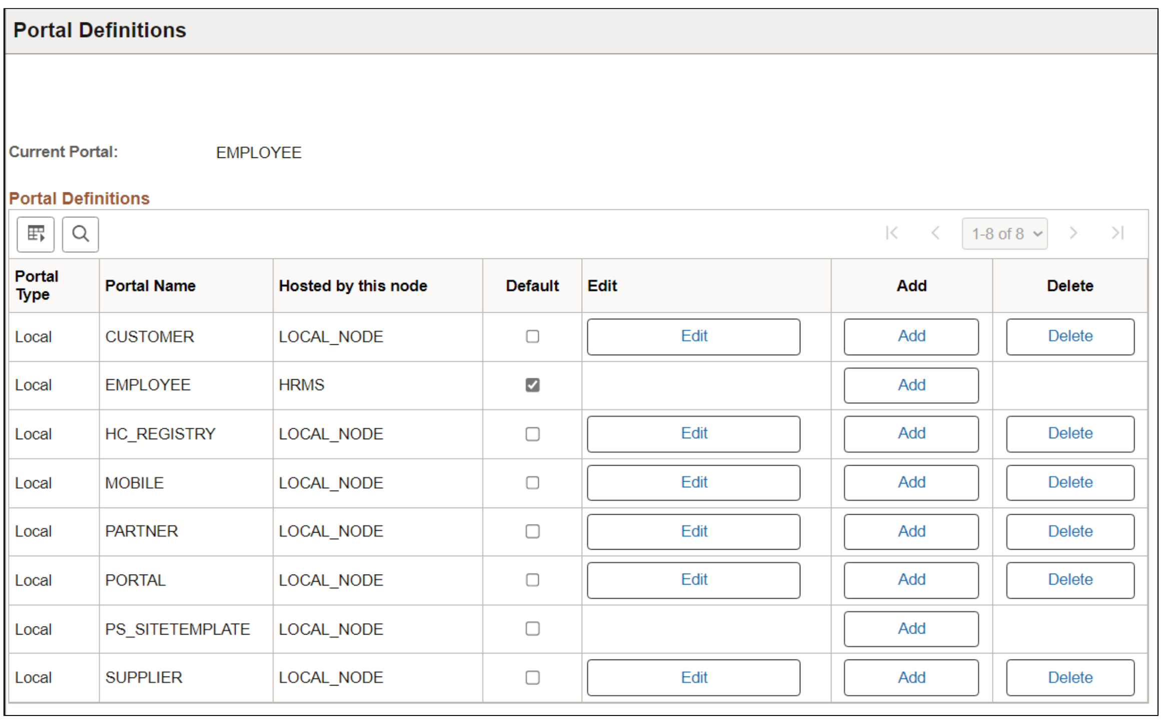
Common Elements Used in This Section

Field or Control	***Description***
Hosted by this node	The node where the portal is defined. This is either a remote or local node. The node must have been defined previously on the Node Definitions page.
Portal Name	The name to appear as the portal label. This value is used for the browser window title when a long description has not been entered.

Defining Portals

Access the Portal Definitions page (**PeopleTools** > **Portal** > **Portal Definitions**).

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


Portal Definitions

Current Portal: EMPLOYEE

Portal Definitions

1-8 of 8

Portal Type	Portal Name	Hosted by this node	Default	Edit	Add	Delete
Local	CUSTOMER	LOCAL_NODE	☐	Edit	Add	Delete
Local	EMPLOYEE	HRMS	☑		Add	
Local	HC_REGISTRY	LOCAL_NODE	☐	Edit	Add	Delete
Local	MOBILE	LOCAL_NODE	☐	Edit	Add	Delete
Local	PARTNER	LOCAL_NODE	☐	Edit	Add	Delete
Local	PORTAL	LOCAL_NODE	☐	Edit	Add	Delete
Local	PS_SITETEMPLATE	LOCAL_NODE	☐		Add	
Local	SUPPLIER	LOCAL_NODE	☐	Edit	Add	Delete

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

Field or Control	***Description***
Portal Type	Displays the attribute of the selected node. Possible types are *Local* and *Remote.*

Field or Control	***Description***
Default	Select to use this as the default node. Only local nodes can be the default. If you change the default, be sure to click the **Save** button.
Edit	Click to update the portal definition. **Note:** This button does not appear for the site template or the local node that is currently in use. To modify the current portal, use the General Settings page. See Managing General Settings for Portals.
Add	Click to add a new portal definition. The new portal is listed in alphabetical order.
Delete	Click to delete a portal definition. You cannot delete the site template or node that is currently in use.

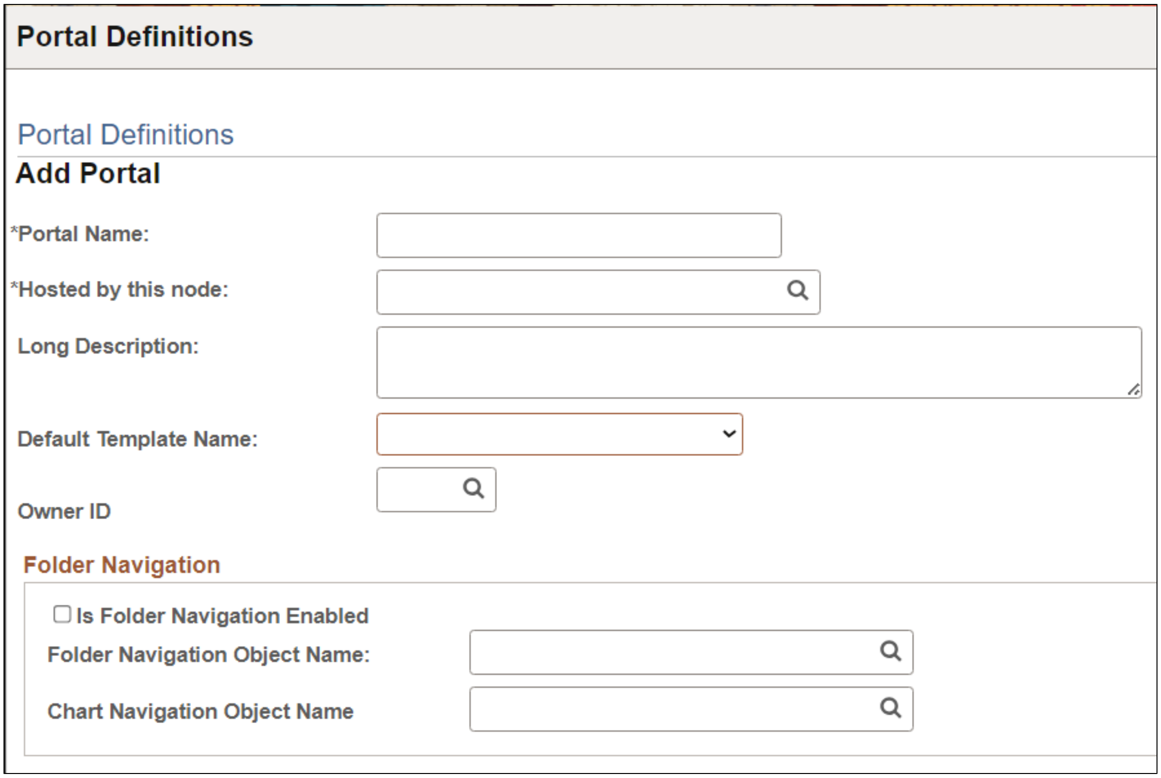
Related Links

Managing General Portal Settings

Adding and Editing Portal Definitions

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

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



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

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

Folder Navigation

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

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

See Managing General Settings for Portals.

Changing Default Portals

You can change the default portal in two ways:

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

Changing the Default Portal in the Portal Definition

To change the default portal:

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

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

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

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

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

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

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

Administering Folders

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

- Define folders.
- Set folder permissions.

Understanding Folder Administration

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

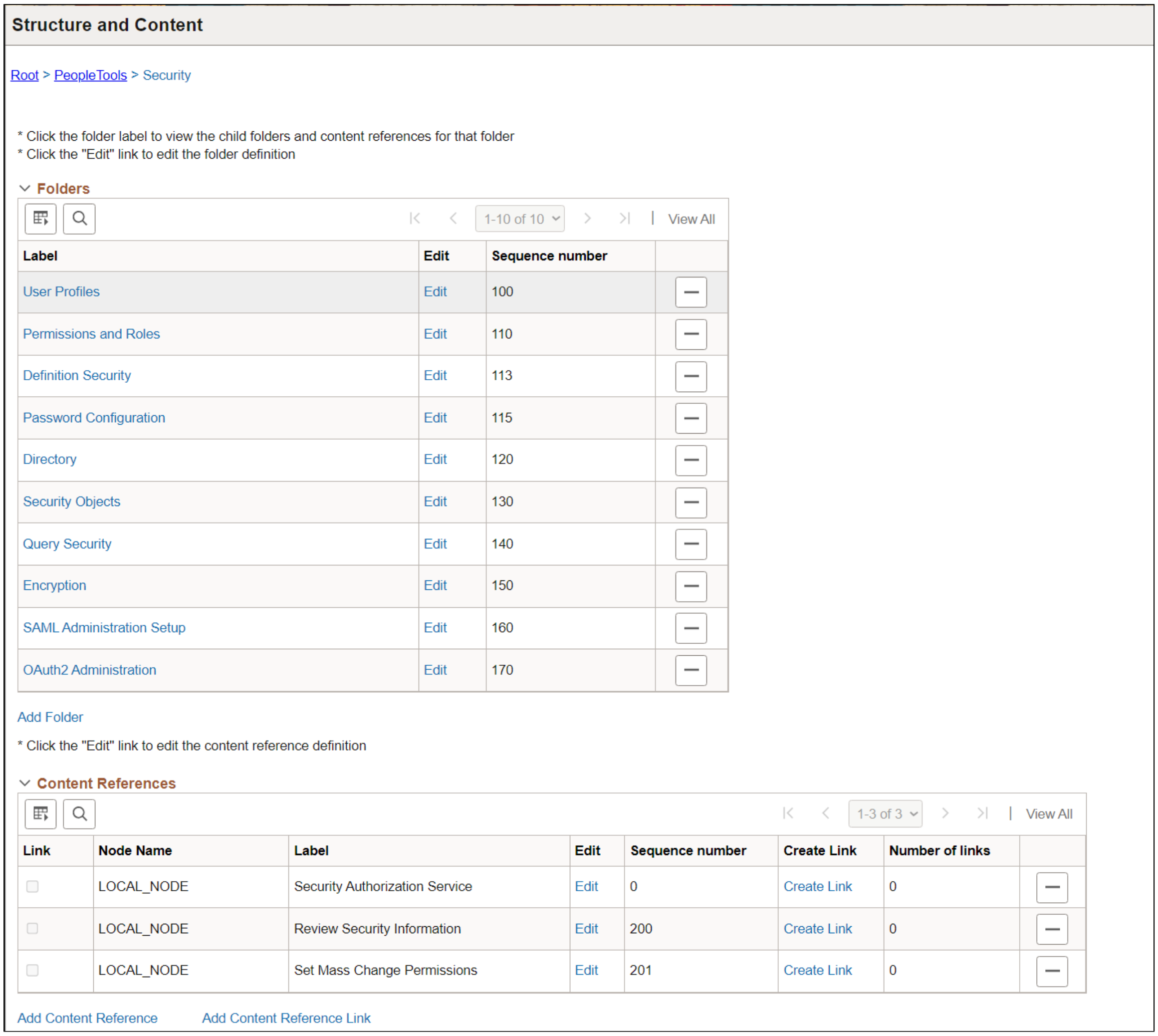
Organization of the Portal Registry

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

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

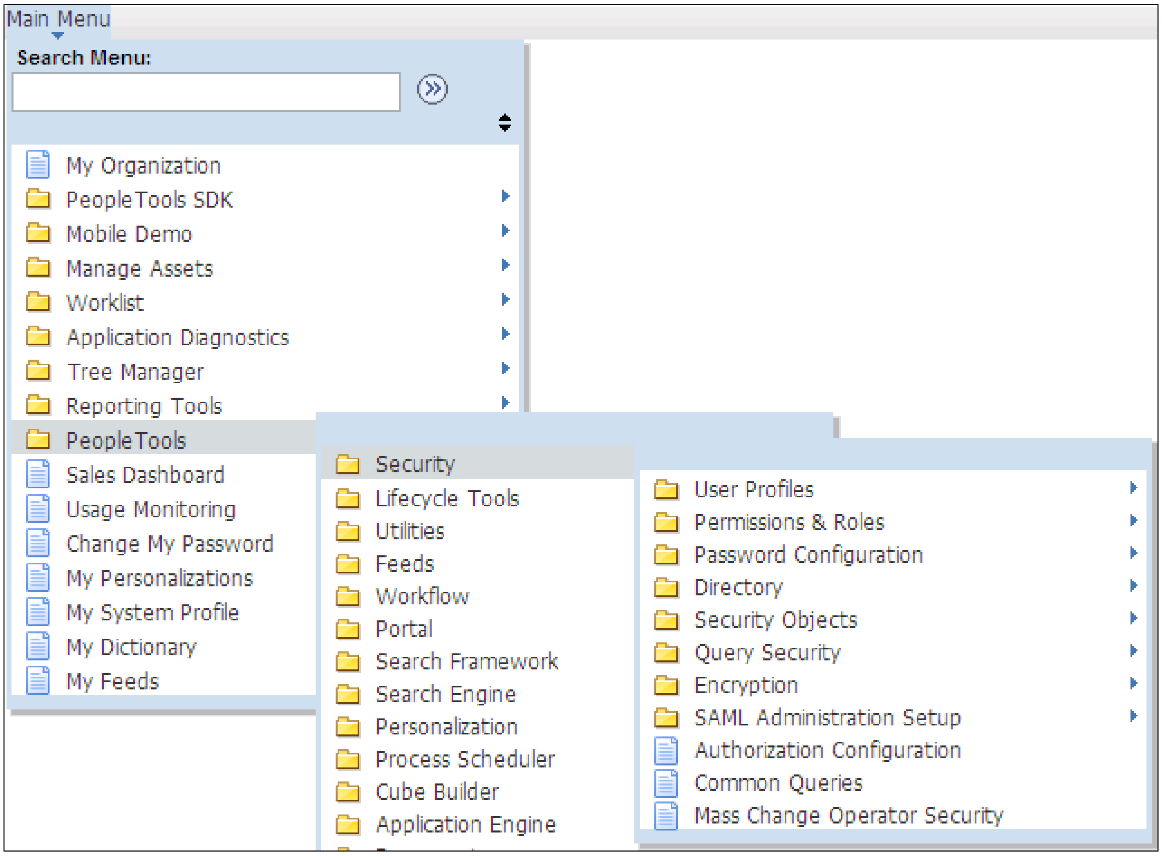
As you navigate through the hierarchy of folders, breadcrumbs appear at the top of the page directly beneath the Main Menu navigation bar. The breadcrumbs appear to visually indicate your current position in hierarchy of the portal registry. Click a link in the breadcrumbs to return to the folder. In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

The following example illustrates the Structure and Content page showing the breadcrumb path, and the folders and content references that appear in the Security folder.



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

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



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

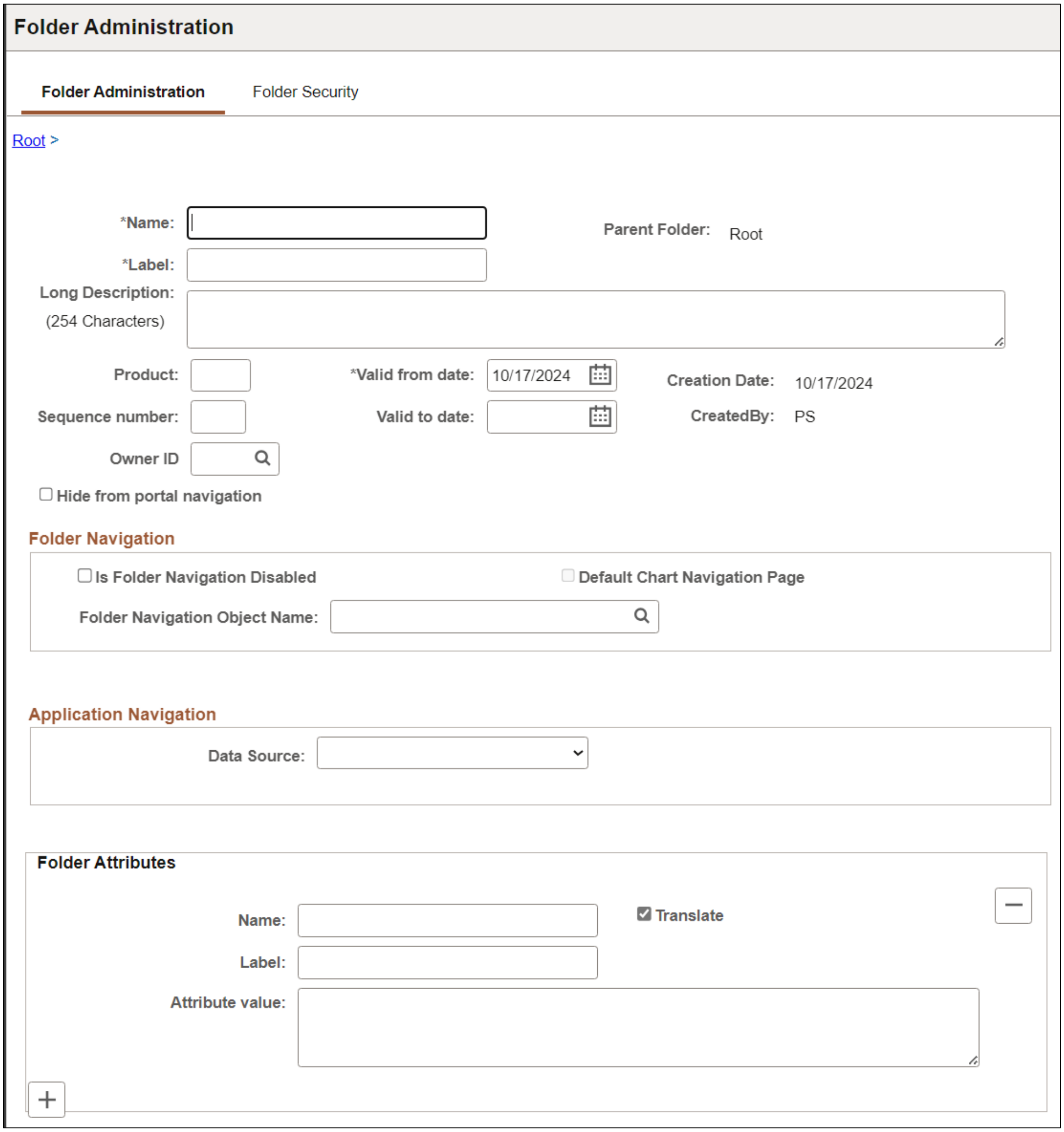
Related Links

Administering Folders

Defining Folders

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

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



1. Enter values in the **Folder Administration** fields.
2. (Optional) Enter values in the **Folder Attributes** fields.
3. Set the folder security.

   See Setting Folder Permissions.
4. Save the folder.

Folder Administration

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

Field or Control	*Description*
Name	Enter a unique identifier for the folder. **Note:** The name can consist of letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.
Label	Enter the text to appear as the link for this folder.
Copy object	Click to access the Copy Object to a Different Portal Registry page, which enables you to copy an existing folder to a different portal registry. This button appears only when you're editing a folder.
Parent Folder	The name of the current parent folder. This field is display-only.
Select New Parent Folder	Click to copy the folder to a new location within the current portal registry. This button appears only when you're editing a folder.
Long Description	Enter a description for this folder. **Note:** In classic forms of navigation (which have been desupported), this description appears as hover text for the link for this folder. In fluid forms of navigation, such as the NavBar menu, no hover effects are displayed.
Product	(Optional) Portal-aware applications can use this field for group processing. All PeopleTools objects have the product name *PT.*
Sequence number	Enter a sequence number to define where this folder appears in its parent folder, and therefore where its associated menu entry appears within its parent menu. If you omit the **Sequence number,** the folder is added to the top of the list of folders and is displayed in alphabetical order. You can change the sequence only one item at a time, and each time you do so, you must save your changes to see the result.
Valid from date and **Valid to date**	Enter a valid from date. The valid to date is optional and is omitted by default. These dates can be used for application-specific processing by portal-aware applications and are used by the portal navigation to determine visibility.

Application Navigation

Use these fields to create SmartNavigation folders.

Note: These fields have been retained for backwards compatibility only. SmartNavigation has been desupported.

Folder Attributes

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

1. Specify the attribute name.
2. (Optional) Specify the attribute label.
3. Enter the attribute value.
4. Select the **Translate** check box to make an attribute label and attribute value translatable.

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

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

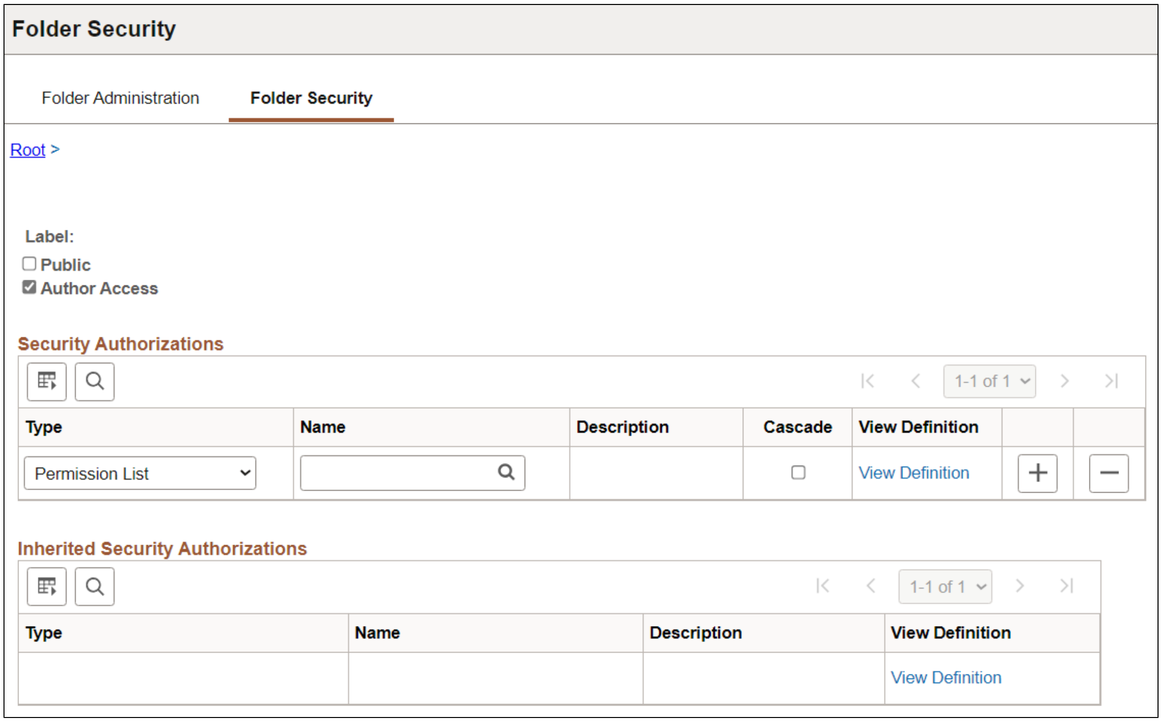
Related Links

"Using Menu Pagelets" (Applications User's Guide)
Administering Content References

Setting Folder Permissions

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

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



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

Security Authorizations

You can base folder security on roles or permission lists when security authorizations are not inherited. You can specify any combination of roles and permission lists to provide folder security.

Field or Control	*Description*
Type	Select from these types: *Permission List:* Enables access for users based on permission list membership. You then specify the permission lists that should have access to this folder. *Role:* Enables access for users based on role membership. You then specify the roles that should have access to this folder.

Field or Control	***Description***
Name	Select the permission list or role that provides access to this folder.
Cascade	Select to indicate that the permission list or role of the current row should also apply to any folder or content reference that is subordinate to this folder.
View Definition	Click to access the security administration component for the permission list or role specified on the current row.

Inherited Security Authorizations

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

Related Links

“Understanding Permission Lists” (Security Administration)

“Understanding Roles” (Security Administration)

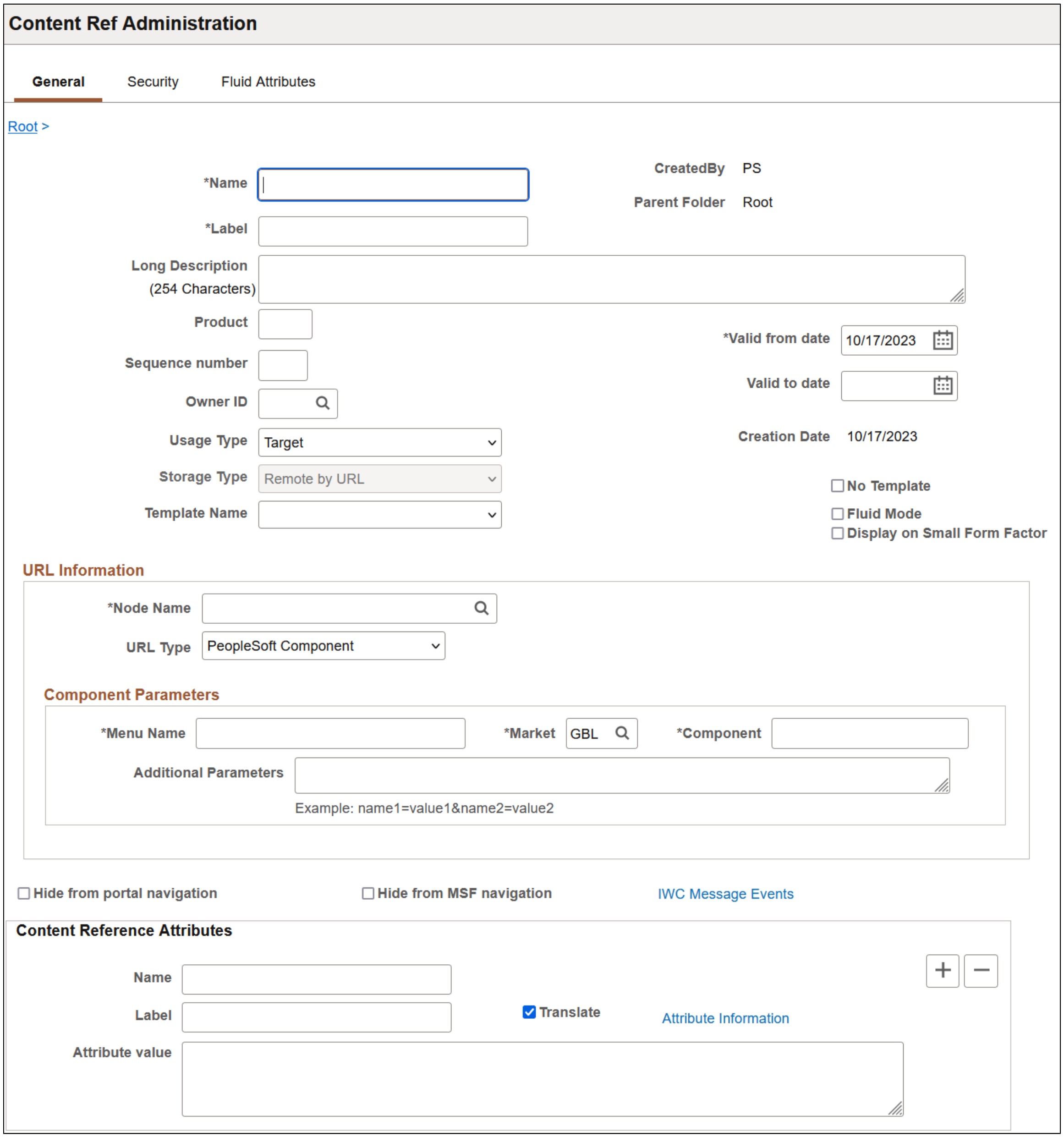
Administering Content References

This section discusses how to define and administer content reference definitions.

Defining Content References

Access the Content Ref Administration page (**PeopleTools** > **Portal** > **Structure and Content.** Navigate the portal registry hierarchy until you are in the parent folder for the content reference; then click the link for an existing content reference or click the **Add Content Reference** link.)

This example illustrates the fields and controls on the Content Ref Administration page. You can find definitions for the fields and controls later on this page.



Note: Many of the fields described in this section appear only when you enter certain field values.

Many of these fields are similar to fields on the Folder Administration page.

When you save a content reference that points to a PeopleSoft component or script on the default local node, the security settings for the content reference are inherited automatically from the menu or script security settings. Any permission lists or roles that you enter on the Content Reference Security page before saving the content reference are deleted and replaced by those of the component or script. These new security settings are also propagated up the folder hierarchy. This ensures that the content reference is visible and all folders are viewable, enabling users to navigate to the component.

Warning! Do not add internal component-based PeopleSoft content references using the Content Ref Administration page. Use the registration wizard instead so that information is consistent between Application Designer and the portal. Any changes made to the menu definition in Application Designer and not made to the portal content reference definition break the content reference. The portal navigation system uses the portal registry, not the internal location of menu definitions in Application Designer.

See “Understanding the Registration Wizard” (Application Designer Developer’s Guide).

This table describes the fields that appear on the Content Ref Administration page:

Field or Control	***Description***
Usage Type	Select from these usage types: • *Fluid Homepage.* • *Frame template.* • *Homepage tab.* • *HTML template.* • *Inline frame template.* • *Pagelet.* **Important!** The portal recognizes pagelets that are registered as portal objects in the Pagelets folder only. This restriction is not enforced by the portal registry; you can create a pagelet that is not recognized. The name that you specify for a new pagelet can consist of letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit. • *Section Template.* • *Target.* This is the default value. Based on the value that you select, the application shows, hides, disables, and enables the other fields on the page so that only the fields that are relevant to the usage type are available. For example, if you select a usage type of *Pagelet,* then **Pagelet Attributes** group box appears in which you can specify the default column, the refresh time, and other properties that apply specifically to pagelets. For a typical PeopleSoft application page, set the usage type to *Target* (the default value) and the URL type to *PeopleSoft Component.* With a usage type of target, the content will populate the target area on the page. **Note:** An additional group box titled Homepage tab attributes is displayed when either *Fluid Homepage* or *Homepage tab* is selected as the usage type. See Creating and Configuring Homepage Definitions for more information.

Field or Control	***Description***
Storage Type	This field is enabled when you selected a usage type of *Frame template* or *HTML template.* Select *Local (in HTML Catalog)* or the default value, *Remote by URL.*
HTML Area	This text area is enabled when you select a usage type of *HTML template* or *Frame Template* and *Local (in HTML Catalog)* as the storage type. An HTML template uses the HTML that is entered into the HTML area of a content reference that is located in the Templates folder, which you find within the Portal Objects folder. Bind variables cannot be used within the HTML area. The HTML area includes any HTML that is required for the template itself, plus tags that specify the URLs for template pagelets.
Template Name	Select the template for this page. If you leave this field blank, the portal servlet automatically uses the default template of the node at runtime. The node default template is set up on the General Settings page. If a node default template has not been specified, it uses the portal default template.
No Template	Select this option to have the target content not wrapped with a portal template (by the psp servlet) at runtime. This option is selected automatically for certain types of fluid content; manually select it for all other types of fluid content. For classic content, it is typically not desirable to select this option. When this check box is selected, the **Template Name** field is hidden. **Note:** This option appears only for usage type of *Target*.
Fluid Mode	Manually select this option only for: • iScript based URLs. • Generic PeopleSoft URLs. • *Remote* component-based fluid content. **Note:** This option appears only for usage type of *Target*. **Note:** For *local* component-based fluid content, this setting is maintained on the component definition in Application Designer and appears as display-only here.

Field or Control	*Description*
Display on Small Form Factor	Manually select this option only if this content reference has been designed to display and perform well on small form factor devices such as smart phones. If this option is deselected, any tile defined for this content reference will not be available on fluid homepages or fluid dashboards when the user is signed in on a small form factor device. **Note:** This option appears only for usage type of *Target*. **Note:** For *local* component-based fluid content, this setting is maintained on the component definition in Application Designer and appears as display-only here.
Create Content Reference Link	Click to create or edit a content reference link.
Add Content Reference	Click to add a content reference.
Test Content Reference	Click to test a content reference. See Creating Related Links.
Hide from portal navigation	Select to hide the content reference from menu navigation. **Note:** If this option is selected *and then* this content reference is added to a permission list, the permission list is not added to the parent folder. However, if the content reference is already added to a permission list *and then* this option is selected, the permission list is not removed from the parent folder.
Hide from MSF Navigation	MSF is not currently supported.
IWC Message Events	Click to access the Interwindow Communications page. See Publishing and Subscribing to IWC Messages.

URL Information

The fields that appear in this group box are based on the selection that you make in the **URL Type** field.

Field or Control	***Description***
Node Name	Specify the node that hosts this content reference. Enter *LOCAL_NODE* to associate the content reference with the default local node (as defined on the Node Definitions page). **Note:** In a clustered environment, when the Usage Type is set to *Pagelet,* the node name should not be defined as *LOCAL_NODE* and must be set to the actual portal host node for the pagelet.

Field or Control	***Description***

URL Type	Select from these URL types:
	
	• *Non-PeopleSoft URL:* Enter the URL in the **Portal URL** field that appears.
	
	This usage type can also be used in conjunction with a defined node. The result is the concatenation of the selected node's URI text and the portal URL that you entered.
	
	• *PeopleSoft Component:* The **Component Parameters** region appears. Select the menu name, market, and component name to use for this content reference.
	
	• *PeopleSoft Generic URL:* Enter the URL in the **Portal URL** field that appears.
	
	This usage type can also be used in conjunction with a defined node. The result is the concatenation of the selected node's URI text and the portal URL that you entered.
	
	• *PeopleSoft Homepage Tab:* No longer used. Do not select this option.
	
	• *PeopleSoft Script:* The **iScript Parameters** group box appears. Select the record (table) and field names, PeopleCode event name, and PeopleCode function name that specifies the iScript to use.
	
	• *PeopleSoft Worklist URL:* Enter the URL in the **Portal URL** field. Use the following format:
	
	`w/WORKLIST?ICAction=ICViewWorklist&⇒`
	
	`Menu=Worklist&Market=GBL&PanelGroup⇒`
	
	`Name=WORKLIST`
	
	• *Remote Portlet:*This option is desupported. Do not select this option.
	
	Note: The availability of these options depends on the selected usage type.

Additional Parameters	Enter query string parameters to be appended onto a PeopleSoft component or PeopleSoft script. For example, enter *emplid=8001*.

Pagelet Attributes

The **Pagelet Attributes** group box enables you to manage pagelet behavior and appearance.

See Using Attributes to Enhance Pagelets.

Content Reference Attributes

Content reference attributes provide a framework to specify and store free-form information about a content reference. For example, you can use content reference attributes to specify and store keywords, help information, related content properties, and related links for content references.

In addition, portal-aware applications can use content reference attribute information for application-specific processing. Portal-aware applications are those that:

- Know about the portal registry and API.
- Know how to get information from a content reference.
- Contain code that can access the properties of content references and do something meaningful with the information.

Field or Control	***Description***
Name	Enter an object name to identify the attribute programmatically. This name is not displayed to the user. **Note:** The name can consist of letters, digits, and underscores (_) only. It cannot contain any spaces and cannot begin with a digit.
Translate	Select to specify that the label and attribute values are translated.
Attribute Information	Click to create related links. You can create related links only after entering *RELLINK* in the **Name** field.
Label	Enter a descriptive name for the attribute to display to the user. The label is required only if **Translate** is selected.
Attribute value	Enter information about the attribute that the portal uses to apply the attribute to the content reference.

Links

This grid appears only if content reference links that connect to this content reference exist. Click **Link Definition** to view the definition of each listed content reference link.

Available Content Reference Attributes

This example illustrates the fields in the Content Reference Attributes group box using the KEYWORD attribute as an example.

This table describes the content reference attributes that your PeopleTools portal recognizes:

Content Reference Attribute Name	***Description***
BANNERTILEHANDLER	Use this attribute to identify the custom application class implementation of PTNUI:Banner:BannerTile that applies to this contextual fluid dashboard. See Configuring the Fluid Dashboard.
BCUPDATELABEL	Use this attribute to override the default label in the breadcrumbs. See Overriding Default Breadcrumbs Using Content Reference Attributes.
BCUPDATETYPE	Use this attribute to override the default breadcrumb update behavior. See Overriding Default Breadcrumbs Using Content Reference Attributes.
BGLOBALVALUES	Use this attribute to prevent pagelets from loading in parallel. Pagelets with global variables should not be loaded in parallel; such pagelets must be loaded sequentially. When the parallelLoading web profile property is set to true, you must also add the BGLOBALVALUES content reference attribute to the content reference definition for each pagelet that includes global variables; set the BGLOBALVALUES attribute to true. Then, these pagelets will be loaded sequentially while all other pagelets will be loaded in parallel. See Configuring Custom Properties.

Content Reference Attribute Name	Description
KEYWORD KEYWORD_CUST	Use these attributes to enter keywords or keyword phrases to be indexed by the PeopleSoft Search Framework as part of the PTPORTALREGISTRY (the menu) search index. Keywords can be used to add alternative search terms not found in the content reference label or description. Separate search keywords or phrases by commas. Select the Translate check box when using these attributes. The example preceding this table illustrates the use of the KEYWORD attribute. The KEYWORD attribute is reserved for use by PeopleSoft application development teams. The KEYWORD_CUST attribute is for use by customers. See “Working With the Portal Registry Search” (Search Technology).
KEYWORD KEYWORD_*LNG*	These attributes are added by the system when keywords are added to specific portal definitions such as classic WorkCenters, classic dashboards, and classic pagelet definitions. Then, the keywords are available on the Pagelet Search page. See Searching for and Importing Pagelets and Tiles.
Name of a default homepage object or image	Use this attribute to override a default HTML object or image used for homepage components, such as tabs and pagelets, by providing the name of an alternative HTML object or image. **Note:** You can override only a specified set of homepage objects and images.
IMAGE_BIND_12 to IMAGE_BIND_26	Use these attributes (numbered from 12 to 26) to insert additional custom images on a homepage tab.
NAVNEWWIN	Use this attribute to specify that the component or page always opens in a new browser window. **Note:** This attribute applies to links on navigation collection template pagelets that appear in the pagelet area of a WorkCenter page.

Content Reference Attribute Name	*Description*
NAVTOPWIN	Use this attribute to specify that the component or page opens in the current browser window and replaces the current page completely. **Note:** This option affects only the link behavior when it is rendered in a navigation collection template pagelet inside the pagelet area of a WorkCenter page.
PSCACHECONTROL	Use this attribute to implement caching for individual pagelets and targets based on user, role, or application.
PORTAL_BANNER_PGLT	Use this attribute to create a banner pagelet, which can appear at the top of a homepage or dashboard page and spans the width of the page. **Note:** To use a pagelet as a banner pagelet, it has to be added to a homepage tab or dashboard page as a required-fixed pagelet in the first column. See Maintaining Pagelet Branding Attributes.
PORTAL_ENCODING_OVERRIDE	Use this attribute to override the character set encoding that is used by the content reference.
PORTAL_HP_COMPONENT	Use this attribute to hide pagelet attributes such as the title bar, the borders, or both. See Maintaining Pagelet Branding Attributes.
PSTIMEOUT	Use this attribute to specify how long the portal should wait for a pagelet to load before it considers the pagelet unavailable.
PT_PORTAL_PGLT_HDR_IMG	Use this attribute to add an icon to the header of the pagelet. The attribute value should be the name of the image definition that is stored in the image catalog using Application Designer.
PTPPB_PAGELET_HELP_LINK	Use this attribute to specify a URL that you want to associate with the help link for a pagelet. **Note:** This attribute takes priority over a PeopleSoft Help ID if both properties are set.

Content Reference Attribute Name	***Description***
RCF_SERVICE_PARAMETER	Use this attribute to provide related content service parameters in name/label pairs for the content reference. The presence of this attribute changes the default behavior of component-based URLs and supplants all parameters with those that you enter in the Value field. See Creating and Managing Related Content Service Definitions.
RELLINK	Use this attribute to create related links for the content reference.
TILEAPPFILTER	Use this attribute to specify a custom application class to hide or display this tile on fluid homepages and dashboards. See Hiding or Displaying Tiles Using an Application Class.
TILEAPPFILTER_REFRESH	Use this attribute to re-evaluate the code configured with the TILEAPPFILTER attribute every time the homepage is opened (invalidating all homepages that are cached). See Configuring Tiles for an Empty Section.
HIDE_IN_PERSONALIZE	Use this attribute to hide a tile on the Personalize Homepage page and to not list the tile on the Add Tile page. **Note:** Use this attribute for a tile only within a section on a homepage or dashboard. See Hiding Tiles in Sections on Homepage Personalization.
EMPTY_SECTION_TILE	Use this attribute to display the tile when no other tile is present in a section or the tiles' visibility in a section is controlled through PeopleCode. **Note:** Use this attribute for a tile only within a section on a homepage or dashboard. See Configuring Tiles for an Empty Section.

Content Reference Type Parameters

This table summarizes the valid combinations of usage type, storage type, and URL type or key HTML content supported for content references:

Usage Type	***Storage Type***	***URL Type or Key HTML Content***
Frame template	Remote by URL	PeopleSoft Script
Frame template	Remote by URL	Non-PeopleSoft URL
Frame template	Local (in HTML Catalog)	Target or IClientComponent element
Target	Remote by URL	PeopleSoft Component
HTML template	Local (in HTML Catalog)	Target, Pagelet, or Source element
HTML template	Remote by URL	PeopleSoft Script
HTML template	Remote by URL	Non-PeopleSoft URL
Homepage tab	Local (in HTML Catalog)	PeopleSoft Homepage tab
Inline Frame template	Local (in HTML Catalog)	Target or Pagelet component
Inline Frame template	Remote by URL	PeopleSoft Script
Inline Frame template	Remote by URL	Non-PeopleSoft URL
Pagelet	Remote by URL	Non-PeopleSoft URL
Pagelet	Remote by URL	PeopleSoft Component
Pagelet	Remote by URL	PeopleSoft Script
Pagelet	Remote by URL	PeopleSoft Generic URL
Pagelet	Remote by URL	PeopleSoft Worklist URL
Target	Remote by URL	PeopleSoft Generic URL
Target	Remote by URL	PeopleSoft Script
Target	Remote by URL	PeopleSoft Worklist URL

Usage Type	***Storage Type***	***URL Type or Key HTML Content***
Target	Remote by URL	Non-PeopleSoft URL

Related Links

Defining Folders
Understanding Portal Templates
Managing General Portal Settings
Configuring Pagelet Time-out Settings
Using Predefined Homepage HTML Definitions
Administering Server-Based Caching
Overriding Page Encoding
Maintaining Pagelet Branding Attributes
Searching for and Importing Pagelets and Tiles

Creating Related Links

You can create related links for a portal page using the RELLINK content reference attribute. You can create related links to external sites, to internal sites, and to other portal pages.

Note: Related links can be implemented with your choice of template on the content reference. Using the PeopleTools Related Links template displays the related links for the content reference, but not the portal menu navigation.

To create related links:

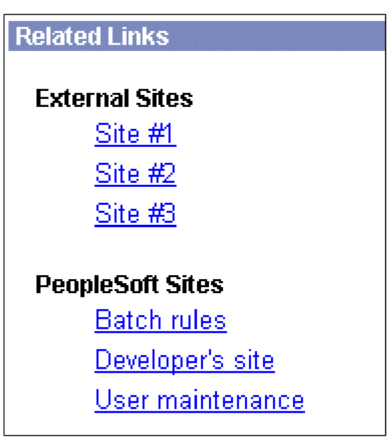
1. Define the RELLINK content reference attribute.

2. Create related links groups.

3. Create related links for each group.

4. Test the related links.

This example illustrates related links for a content reference.



This table describes the page elements shown in the sample page:

Page Element	***Description***
Related Links	RELLINK content reference attribute with the label Related Links.
External Sites	A related link group defined with the label External Sites.
Site #1, Site #2, Site #3	Related links for the External Sites group.
PeopleSoft Sites	A related link group defined with the label PeopleSoft Sites.
Batch rules, Developers' site, and User maintenance	Related links for the PeopleSoft Sites group.

Defining the RELLINK Content Reference Attribute

To define a related link content reference attribute:

1. Access the Content Ref Administration page (**PeopleTools** > **Portal** > **Structure and Content.** Navigate to the appropriate content reference; then click the **Edit** link.)
2. Enter *RELATEDLINKS_TEMPLATE* in the **Template Name** field.
3. In the **Name** field of the **Content Reference Attributes** region, enter *RELLINK.*

   After you enter this value and move out of the field, the **Attribute Information** link changes to **Edit Related Links.**
4. Verify that the **Translate** check box is selected.
5. Enter a label for the related links attribute.

Note: You do not need to enter any information in the **Attribute Value** field. After you create the related link group and the related links, the system populates this field with all defined attributes as a string.

Creating Related Link Groups

Access the Related Links Group (PORTAL_RLNK_ADM) page.

To create a related link group:

1. In the **Related Link Groups Label** field, enter a name.

   This name is the heading under which you place related links.
2. To create additional related link groups, click **Add** and repeat step 1.

Creating Related Links

To create a link:

1. In the **Related Links** region on the Related Links Group (PORTAL_RLNK_ADM) page, enter a name in the **Related Link URL Label** field.

   This name appears as a link under the group name.

2. If you're linking to an internal site, select a node name from the list.

   If you're linking to an external site, leave this field blank.

3. In the **Related Link URL** field, enter URL information for the link.

   If the link is to an internal site, enter the URL or URI. The URL or URI can be to an internal site, to a portal, and so on. If you specified a value for the **Portal URI Text** field on the node definition, you can enter a specific document name as the target. The name is appended to the URI text for a full URL.

   If the link is to an external site, enter the URL.

4. To define additional links for the group, click **Add** and repeat steps 2 and 3.

5. Click **OK** to save the entry and return to the Content Ref Administration page.

   The **Attribute Value** field is populated with an XML-type formatted string. This string contains all the related link data and is stored in the **PORTAL_ATTR_VAL** field of the PSPRSMATTRVAL record for the content reference specified in the PORTAL_OBJNAME field. The following example shows a generated XML-type formatted string:

```
<GRPLBL>Worklist Pages
  <URLLBL>Worklist Details</URLLBL>
  <CP>Portal</CP>  <URL>ICType=Panel&Menu=WORKLIST&Market=GBL&PanelGroupName=W⇒

ORKLIST_DETAILS
  </URL>
  <URLLBL>Worklist Monitor</URLLBL>
  <CP>Portal</CP>  <URL>ICType=Panel&Menu=WORKFLOW_ADMINISTRATOR&Market=GBL&Pa⇒

nelGroupName=WF_MONITOR_01
  </URL>
 </GRPLBL>

 <GRPLBL>Tasks
  <URLLBL>Task List Summary</URLLBL>
  <CP>Portal</CP>
  <URL>ICType=Panel&Menu=PORTAL_COMPONENTS&Market=GBL&PanelGroupName=EO_PE_TAS⇒

K_LIST  </URL>
</GRPLBL>
```

Testing Related Links

You should test related links to view how the page appears to users, ensure that the labels are correct, and verify that the links function properly. To test related links, click **Test Content Reference** on the Content Ref Administration page just above the **URL Information** region.

Note that:

- If the portal cache is enabled on the web server, the related link template does not appear for a content reference page accessed from the menu until the cache expires.
- Related link URLs are displayed only when a user has security access to the URLs.

To grant security access to a related link URL, register the related link URL as a separate content reference and associate permission lists with it.

If the related link is not separately registered, then it is treated as having public access.

If a related link group contains no URLs to which the user has security access, the related link group is hidden. If the user does not have access to any of the related links associated with the target page, then the related links section of the template is hidden.

If the target page does not display any related links, ensure that:

- The application and portal databases are registered as content providers.

  Also check for data-entry errors.

- The templates are registered in Application Designer with the correct names.

  If the templates are registered, check for data-entry errors.

- The correct related link template is specified for the target page.

  Also ensure that the value of the PORTAL_TEMPL_NAME field in the PSPRSMDEFN record displays a correct related link template

- The RELLINK string was saved on the target page in the content registry.

  Check whether an XML string appears in the **Attribute Value** field on the Content Ref Administration page. If not, then delete and reenter the RELLINK attribute for all the related link information.

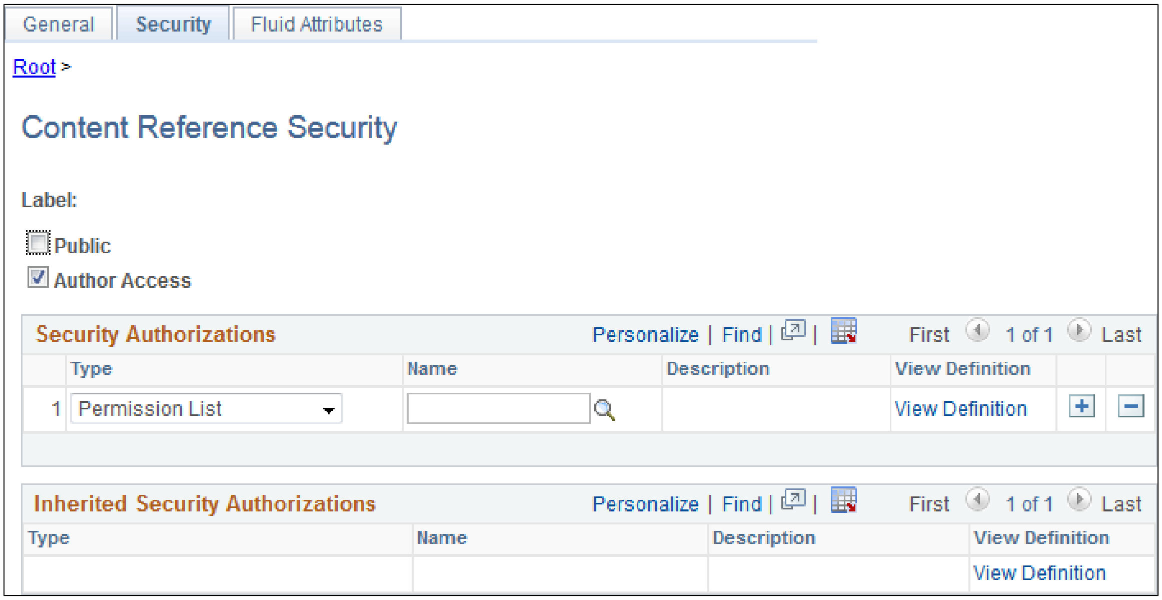
- You assigned security in the registry to the user for the content references listed as related link URLs.

  If not, assign security using the Content Reference Security page for registered related link URLs.

Setting Content Reference Permissions

Access the Content Reference Security page (Select **PeopleTools** > **Portal** > **Structure and Content.** Click the **Edit** link for a content reference; then select the Security page.)

This example illustrates the fields and controls on the Content Reference Security page. You can find definitions for the fields and controls later on this page.



This page is similar to the Folder Security page. You can specify the permission lists and roles that grant security access to the content reference.

Certain URL types inherit the security permission list and role authorizations from the referenced object. For these URL types, you can view, but not modify the security by using the Content Reference Security page. All fields except **Public** and **Author Access** are display-only. This is the list of URL types that inherit security authorization:

- *PeopleSoft Component.*
- *Worklist.*
- *PeopleSoft Script.*

When you save a content reference that points to a PeopleSoft component or script on the default local node, the security settings for the content reference are inherited automatically from the menu or script security settings. Any permission lists or roles that you entered on the Content Reference Security page before saving the content reference are deleted and replaced by those of the component or script. These new security settings are also propagated up the folder hierarchy. This ensures that the content reference is visible and all folders are viewable, enabling users to navigate to the component.

Note: If you copy, move, or add a content reference to a folder, the security settings of the content reference are propagated up the folder hierarchy. This ensures that the content reference is visible and all folders are viewable, and it enables users to navigate to the content reference. If you remove the content reference from that folder, the security settings that were applied when it was added to the folder are also removed from the folder hierarchy, without disturbing any settings still required for other content references remaining in that folder.

Note: While a content reference link does inherit cascaded security from it's own parent folder, it does not inherit cascaded security from the linked-to content reference's parent folder.

Field or Control	*Description*
Public	Select to make the content reference available to all users. When this check box is selected, all other fields except **Label** are hidden.
Author Access	Select to allow content reference access to the creator of the definition, regardless of how other permissions are set. This setting is useful for troubleshooting and testing purposes.

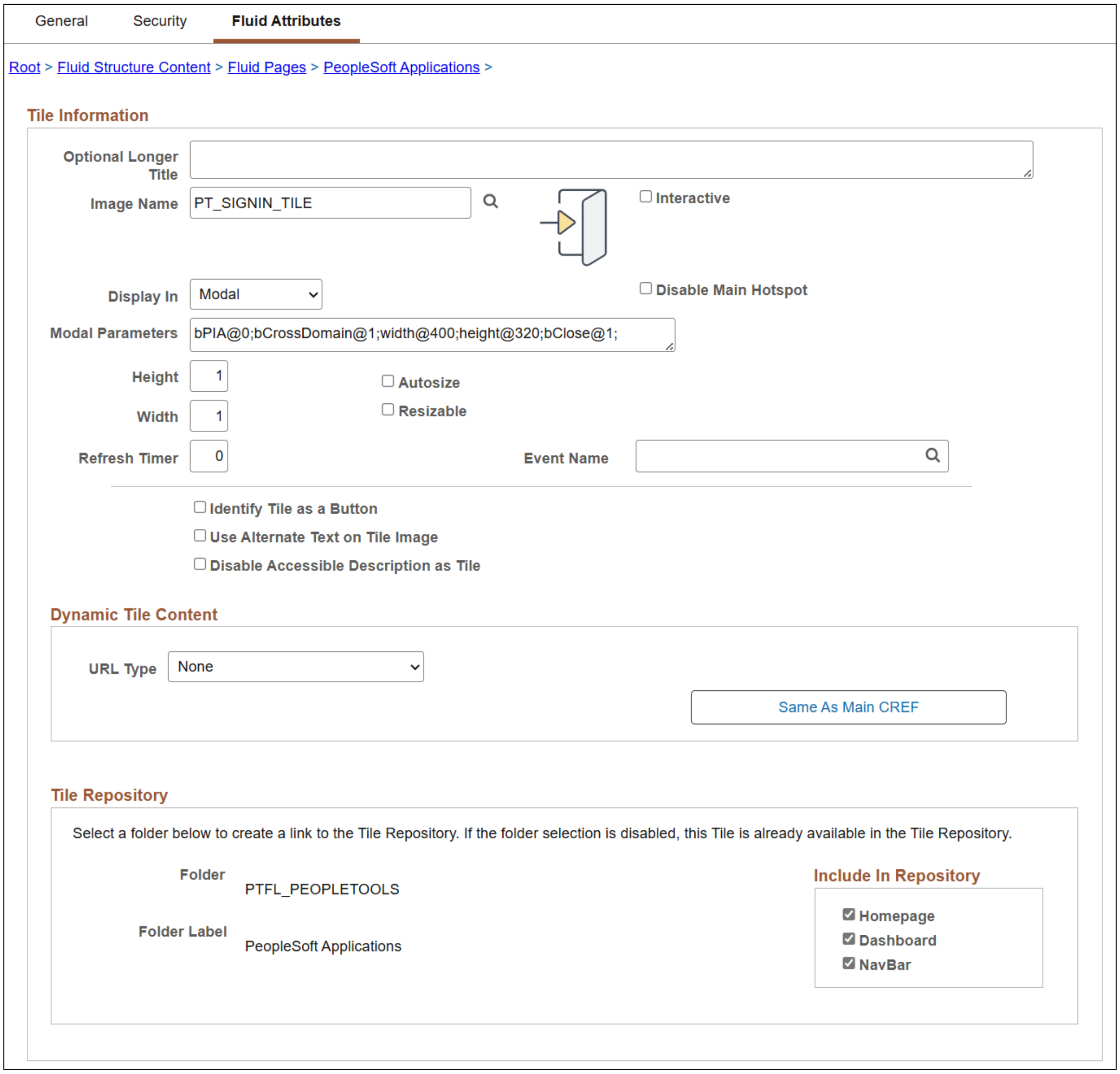
Related Links

Setting Folder Permissions
"Understanding the Registration Wizard" (Application Designer Developer's Guide)
"Understanding Permission Lists" (Security Administration)
"Understanding Roles" (Security Administration)

Setting Fluid Attributes for Content References

Access the Fluid Attributes page (Select **PeopleTools** > **Portal** > **Structure and Content.** Click the **Edit** link for a content reference; then select the Fluid Attributes tab.)

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



Tile Information

Use the Fluid Attributes page to configure the behavior and appearance of fluid content such as tiles, embedded related content, and related information. Certain fields pertain to tile definitions only while others pertain to all types of fluid content.

Field or Control	*Description*
Optional Longer Title	Use this field to enter a title that is more than 30 characters. Currently, the tile label has a limit of 30 characters. You should note when using longer titles on standard tiles (tiles that existed prior to PeopleTools 8.62 release and do not use the 8.62 styling) that the width of the tiles may require to be increased to display the longer label because the standard tiles do not wrap the title. For users of the administrator pages, the long title is available only on the tile search user interface. All other administrator pages that show a list of tiles use the standard tile label. **Note:** The longer title is not applicable to NavBar tiles.
Image Name	Click the Image lookup button to display the Browse Image Catalog page, which allows you to select an image (in SVG format only) from the database to display a custom static image as the fluid content. Otherwise the default image is displayed. See the following section, "Using the Browse Image Catalog Page" for more information. **Note:** When dynamic content is also configured for a tile, this static image is displayed only until the dynamic content is loaded.
Interactive	Select this option to display the fluid content as interactive, which allows users to enter data or click buttons within the fluid content. Since interactive fluid content require more resources than non-interactive ones, select this option only as needed. **Important!** You must also set dynamic tile content for the fluid content to be interactive; otherwise, a static content is displayed. See the following sections for more information. **Important!** While embedded related content can be defined as interactive, you should be aware of the limitations. See the section, “Limitations with Interactive Grouplets,” in Fluid Related Content for more information.

Field or Control	*Description*
Display In	Control how the target transaction or content appears once a user taps the tile. Options are: • *Cur Window*: Target content appears in the current browser window. • *Modal*: Target content appears in a modal window. • *NavBar*: Target content appears in the NavBar. **Note:** This behavior applies only when the tile is added to the NavBar. If this is selected and the tile is added to a homepage or dashboard, the target content displays in the current browser window when tapped. • *New Window*: Target content appears in a new tab in the browser.
Disable Main Hotspot	Select this option to allow links that are displayed dynamically within the tile to be available and usable. Selecting this option disables clicking the tile to display the target content defined for the tile.
Modal Parameters	Enter custom modal options. See “Modal Options” (Fluid User Interface Developer’s Guide) for more information.

Field or Control	*Description*
Height	Set the height of the tile. The default dimensions of a tile are 1 tile unit by 1 tile unit. Tiles up to 8x8 are supported. Note the following conditions and limitations with respect to the height and width of a tile: • A tile can never be larger than the width of the device. For example, if the width of a device is 2 in portrait mode, and 3 in landscape; the tile can only be maximum of width 2 on this device. • The valid size is checked only when the tile is initially loaded. If the device is rotated, or if the browser is resized, the tile is not resized. • Any tile sized larger than 2x2 is not guaranteed to display at the specified size depending on the device used to access the PeopleSoft system. For example, a 4x4 tile on a small form factor device in portrait mode would only show as a 2x3 tile. • Therefore, tiles larger than 2x2 need to implement variable sizing to accommodate different device sizes. Use large size tiles sparingly. See the following section titled “Tile Height and Width” for a table of tile widths and heights from 1 to 8.
Width	Set the width of the tile.

Field or Control	*Description*
Autosize	Select this option to allow the fluid content to dynamically resize itself based on the size of the content. Note the following conditions and limitations with respect to the autosize option: • The autosize option pertains to tiles and related information, but not to embedded related content. • The autosize option is best used on dashboards with few tiles. Be aware that layout changes can occur from the autosized content. • Autosize automatically accounts for resizing of the browser and rotation of the device. • The autosize and resizable options are mutually exclusive; both cannot be selected simultaneously. • The tile will not resize itself to larger than the dimensions of the device. • If the content is larger than the size of the tile, the tile will display scroll bars. • Due to the same-origin policy implemented by web browsers, for the autosize option to work with external content, the external content and the PeopleSoft application must either reside within the same domain or they must both set the document.domain property to the same value. Otherwise, the autosize option is ignored. For more information on how the domain and same-origin is determined, see Same-origin policy.

Field or Control	*Description*
Resizable	Select this option to allow the tile to be redisplayed at the specified alternate dimensions. Note the following conditions and limitations with respect to the resizable option: • The resizable option is best used for tiles that can provide both a summary view and detailed view. When a tile is resized by a user, the content is refreshed to the show content that is optimized for the alternate size (and vice versa when the size is toggled back to the original size). • The autosize and resizable options are mutually exclusive; both cannot be selected simultaneously. • The state of the resize is not saved; therefore, whenever the user returns to the dashboard, the tile is displayed at its original size. • If the alternate size of the tile is the same as the original size (either by definition or by computation of the maximum size allowed by the device), no resize icon is displayed.
Alternate Height	Set the alternate height for a resizable tile.
Alternate Width	Set the alternate width for a resizable tile.
Refresh Timer	Enter the time in seconds to set an automatic refresh period for dynamic content on a tile. When the timer limit has been reached, the system re-draws the tile so that it displays the current data, such as in the case with chart. The default value of 0 disables any automatic refresh. **Note:** The system enforces a 10 second minimum limit. Any value entered less than 10 seconds is ignored and is treated as 10 seconds. When setting this value, be sure to monitor performance of the page refreshes and adjust accordingly.
Event Name	Select the event name to subscribe to a push notification event defined on the server. Subscribing to an event allows the tile content to be updated by the event. See “Understanding the Push Notification Framework” (Fluid User Interface Developer’s Guide) and “Defining Events” (Fluid User Interface Developer’s Guide)for more information on defining push notification events.

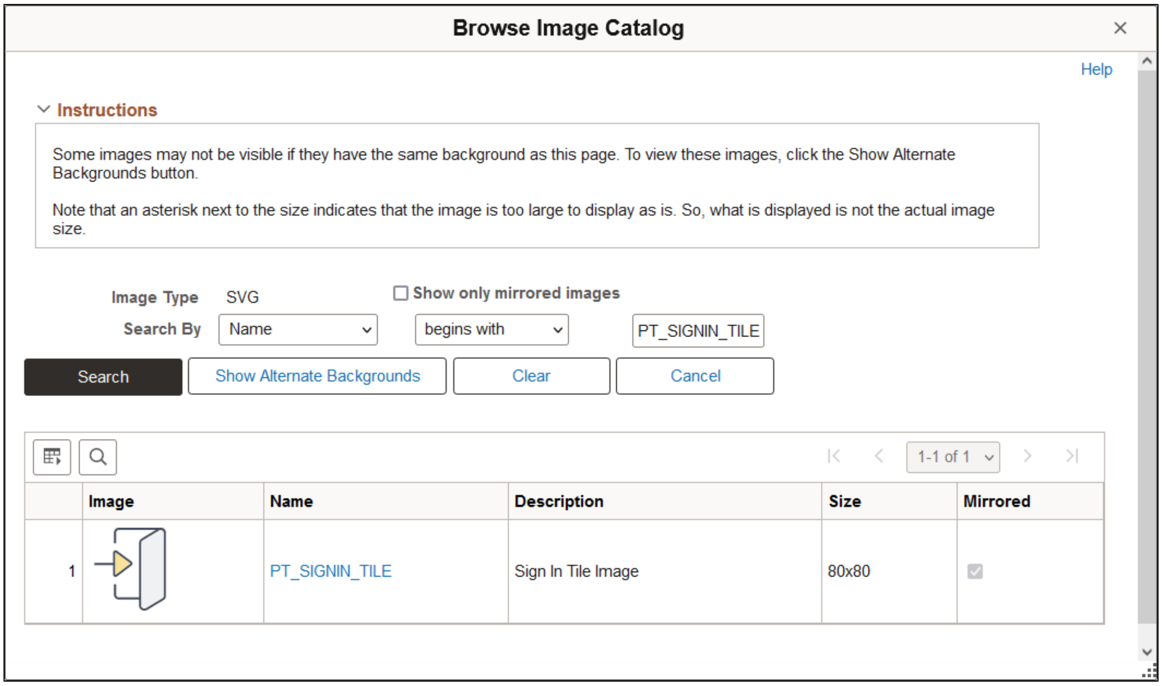
Field or Control	Description
Identify Tile as a Button	Use this setting to change a tile from being identified as a link to a button. This setting is typically used for a tile that you want to set as a section-action button. **Note:** This setting is not applicable to NavBar tiles. For more information on the use of a tile as a button, see Creating Different Presentations of Tiles.
Use Alternate Text on Tile Image	When you identify a tile as a button with no text, the hover text and alternate text indicates the function of the button. Use this setting to specify that the title of a tile (standard or long) is used as the Title and Alt Text if a tile is rendered simply as a button with no text. Typically used when you are identifying a section-action button with only an image so that the hover over and alternate text describes the function of the button. **Note:** This setting is not applicable to NavBar tiles. For more information, see Creating Different Presentations of Tiles.
Disable Accessible Description as Tile	Select to disable the tile accessible description. This setting is typically used in conjunction with the Identify Tile as Button setting so that the tile is no longer identified as a tile. Also, you can use this setting when you are configuring a tile as a link. **Note:** This setting is not applicable to NavBar tiles. For more information, see Creating Different Presentations of Tiles.

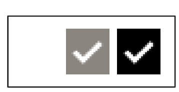
Field or Control	***Description***
Dynamic Tile Content**URL Type**	If you want the tile, embedded related content,or related information to display dynamic content, select one of these options: • Non-PeopleSoft URL • PeopleSoft Component • PeopleSoft Generic URL • PeopleSoft Script See the following sections for information on each option. Selecting None results in the system displaying the selected static image or a default static image if no image is defined on the content reference.
Dynamic Tile Content**Same as Main CREF**	Click the Same as Main CREF button to reset the dynamic content URL type and parameters to be the same as those already defined on the Content Ref Administration (General) page. **Important!** For PeopleSoft components (and for PeopleSoft generic URLs starting with c/ which represent a component URL), the target content URL and the dynamic content URL are required to identify *different components* to prevent data integrity errors in the component buffer. When the target content (on the General page) is one of these two types, the Same as Main CREF button is not displayed. For the dynamic content, you must manually define a component different from the target component before you can save the content reference definition. See "Limitations on Accessing a Component as a Grouplet and Then in a Standalone Manner" (Fluid User Interface Developer's Guide) for more information.

Using the Browse Image Catalog Page

Use the Browse Image Catalog page to select an SVG from the database to display a custom static image as the fluid content.

This example illustrates the fields and controls on the Browse Image Catalog page.



Field or Control	*Description*
Image Type	*SVG* is selected by default and cannot be changed.
Show only mirrored images	Select *Yes* to only retrieve images that are mirrored. Otherwise, both mirrored and unmirrored images are retrieved.
Search By	Select *Name* or *Description* and additional search criteria to limit the search results.
Search	Click the Search button to retrieve the images meeting the search criteria from the database.
Show Alternate Backgrounds	Click the Show Alternate Backgrounds button to redisplay each image in the current grid on three background colors: white, gray, and black. This option allows you to view white images, which are not visible on the default white background. In this example, the white check mark is visible on the gray and black backgrounds: 

Field or Control	*Description*
Clear	Click the Clear button to clear both the search criteria and the grid.
Cancel	Click the Cancel button to close the secondary window.
Name	Select an image by name to use it as the custom static image.

Tile Height and Width

PeopleTools supports tile sizes from 1 tile unit by 1 tile unit up to 8x8. Tile units are device dependent, which means that tiles are not actually square. The actual pixel width and height of a tile depends on the device type—either small form factor (SFF) or non-SFF as illustrated in the following table:

Width	*SFF*	*Non-SFF*	*Height*	*SFF*	*Non-SFF*
1	152	250	1	120	197
2	309	510	2	245	404
3	466	770	3	370	611
4	623	1030	4	495	818
5	780	1290	5	620	1025
6	937	1550	6	745	1232
7	1094	1810	7	870	1439
8	1251	2070	8	995	1646

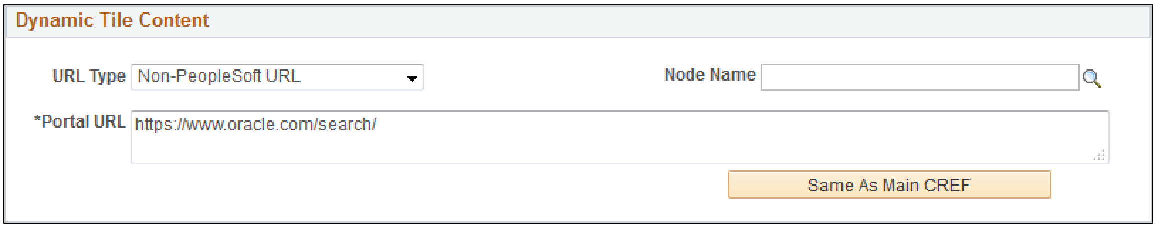
Note: Tiles up to 8x8 are supported.

Dynamic Tile Content: Non-PeopleSoft URL

Use the Non-PeopleSoft URL type to display external content as dynamic fluid content.

Important! For the autosize option to work with external content, the external content and the PeopleSoft application must reside within the same domain. Otherwise, the autosize option is ignored.

This example illustrates the fields and controls from the Dynamic Tile Content group box for a non-PeopleSoft URL. You can find definitions for the fields and controls later on this page.

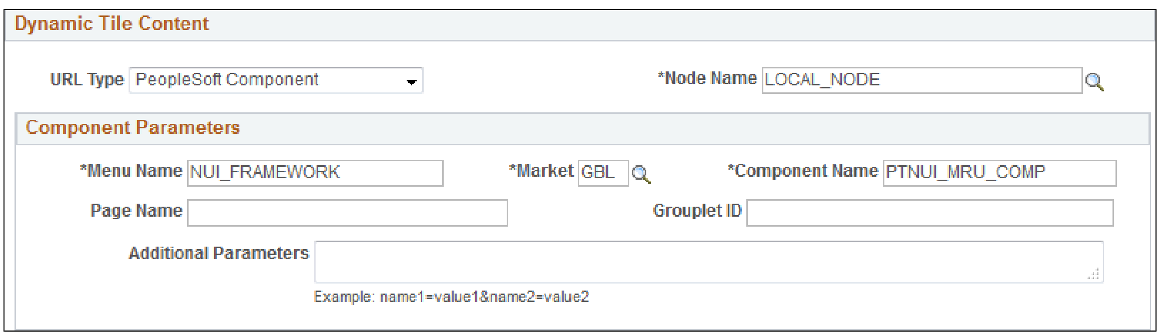


Field or Control	Description
Node Name	Optionally, enter a defined node name.
Portal URL	Enter the complete or partial URL in this field. If you selected a defined node, then enter the partial URL here. The result is the concatenation of the selected node's URI text and the portal URL that you entered.

Dynamic Tile Content: PeopleSoft Component

Use the URL type of PeopleSoft Component to display a *fluid* component as dynamic fluid content.

This example illustrates the fields and controls from the Dynamic Tile Content group box for a PeopleSoft component. You can find definitions for the fields and controls later on this page.



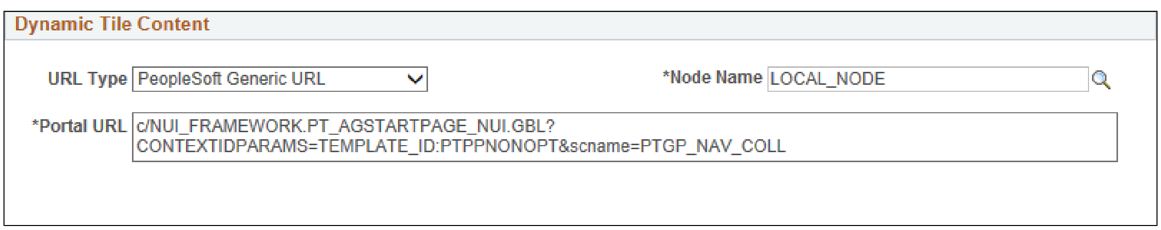
Field or Control	Description
Node Name	Enter the node for the target component.
Menu Name	Enter the menu for the target component.
Market	Select the market for the target component.
Component Name	Enter the name for the *fluid* component.

Field or Control	Description
Page Name	Optionally, enter the *fluid* page name. **Note:** Because multi-page components are not recommended as dynamic content, the page name is not required.
Grouplet ID	Enter the page field name as defined on the General tab of the group box definition. See "Working with Group Boxes" (Fluid User Interface Developer's Guide) for more information. **Note:** The grouplet ID is ignored when the definition is defined to be interactive; the entire component is rendered instead. For non-interactive definitions, if the grouplet ID is not specified, the entire page is rendered instead.
Additional Parameters	Enter any additional query string parameters to be appended onto the URL for the PeopleSoft component. For example, enter *emplid=8001*.

Dynamic Tile Content: PeopleSoft Generic URL

Use the PeopleSoft Generic URL type to display any PeopleSoft URL type as dynamic fluid content. The generic URL type can be used to define iScripts, worklists, and even component-based URLs.

This example illustrates the fields and controls from the Dynamic Tile Content group box for a PeopleSoft generic URL. You can find definitions for the fields and controls later on this page.

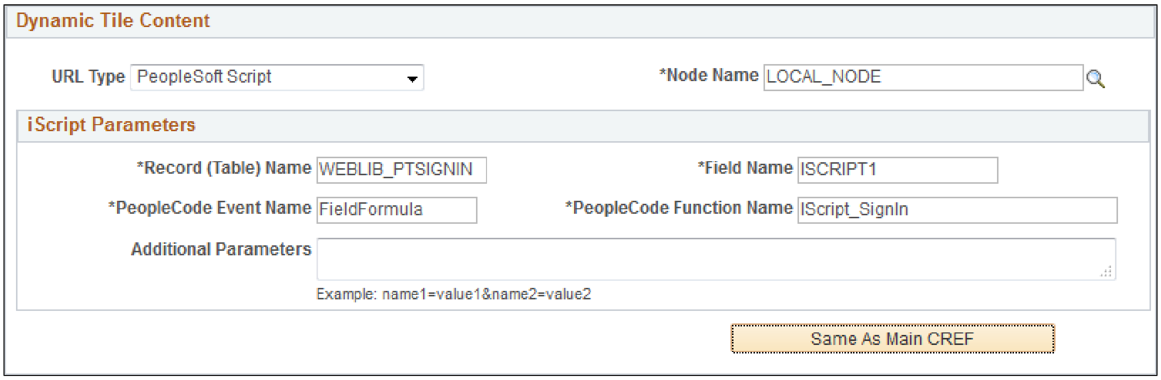


Field or Control	Description
Node Name	Optionally, enter a defined node name.
Portal URL	Enter the complete or partial URL in this field. If you selected a defined node, then enter the partial URL here. The result is the concatenation of the selected node's URI text and the portal URL that you entered.

Dynamic Tile Content: PeopleSoft Script

Use the PeopleSoft Script type to display a PeopleSoft iScript as dynamic fluid content.

This example illustrates the fields and controls from the Dynamic Tile Content group box for a PeopleSoft iScript. You can find definitions for the fields and controls later on this page.

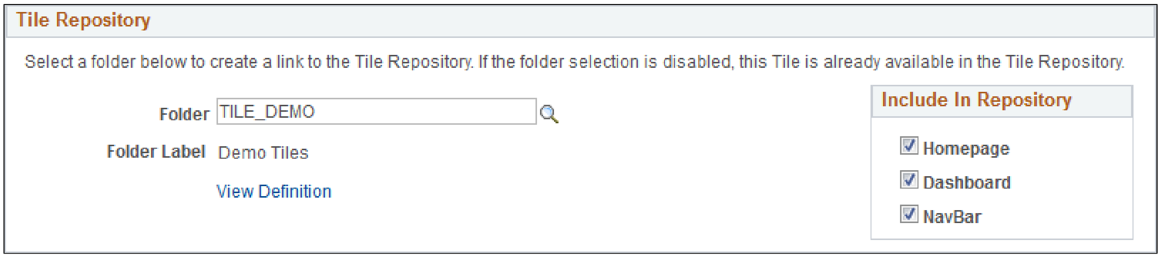


Field or Control	*Description*
Node Name	Enter the node for the target iScript.
Record (Table) Name	Enter the WEBLIB derived/work record name that contains the target iScript.
Field Name	Enter the field name that contains the target iScript.
PeopleCode Event Name	Enter the PeopleCode event name.
PeopleCode Function Name	Enter the user-defined PeopleCode function to be executed.
Additional Parameters	Enter any additional query string parameters to be appended onto the URL for the PeopleSoft script. For example, enter *emplid=8001*.

Tile Repository

Use the Tile Repository group box to specify the category (folder) where content reference definition (or its content reference link definition) resides within the *tile repository.* In this regard, the tile repository refers to the portal registry subfolders under Fluid Structure Content > Fluid Pages.

This example illustrates the fields and controls in the Tile Repository group box. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Folder	For an existing content reference outside the tile repository, select a subfolder within the tile repository. **Note:** Specifying a subfolder within the tile repository creates a content reference link there. The existing content reference definition remains wherever it is defined.
Folder Label	Displays the label for the selected folder.
View Definition	Click the View Definition link to view the definition for the content reference link.
Homepage	Select this option to allow the definition to be listed when the Add Tile dialog box is accessed from fluid homepages.
Dashboard	Select this option to allow the definition to be listed when the Add Tile dialog box is accessed from fluid dashboards.
NavBar	Select this option to allow the definition to be listed when the Add Tile dialog box is accessed from the NavBar.

Related Links

Understanding Tile Definitions and the Tile Repository

PeopleSoft Fluid UX Standards: Tiles

Reviewing Content Reference Permissions

You can use automated queries to discover the content references to which a given user, permission list, or role has access. Each query provides a different degree of detail about the properties of the content references.

Reviewing Content Reference Security by User

Access the User ID Queries page (**PeopleTools** > **Security** > **User Profiles** > **User Profiles** > **User ID Queries**).

This page contains several general-purpose security queries, along with the following links to content reference access queries for the selected user, which are described on the page:

- **User ID's Content Reference Access.**
- **User ID's Content Reference (includes Portal) Access.**
- **User ID's Content Reference (includes Menu, Component and Market) Access.**
- **User ID's Content Reference (includes Portal, Menu, Component and Market) Access.**

When you click one of the query links, a page appears in a new window, containing a grid with the information described for each content reference.

Reviewing Content Reference Security by Permission List

Access the Permission List Queries page (**PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** > **Permission List Queries**).

This page is similar to the User ID Queries page, with the following equivalent links to content reference access queries for the selected permission list:

- **Permission List's Content Reference Access.**
- **Permission List's Content Reference (includes Portal) Access.**
- **Permission List's Content Reference (includes Menu, Component and Market) Access.**
- **Permission List's Content Reference (includes Portal, Menu, Component and Market) Access.**

When you click one of the query links, a page appears in a new window, containing a grid with the information described for each content reference.

Reviewing Content Reference Security by Role

Access the Role Queries page (**PeopleTools** > **Security** > **Permissions and Roles** > **Roles** > **Role Queries**).

This page is similar to the User ID Queries page, with the following equivalent links to content reference access queries for the selected role:

- **Role's Content Reference Access.**
- **Role's Content Reference (includes Portal) Access.**
- **Role's Content Reference (includes Menu, Component and Market) Access.**
- **Role's Content Reference (includes Portal, Menu, Component and Market) Access.**

When you click one of the query links, a page appears in a new window containing a grid with the information described for each content reference.

Testing Content References

After creating or modifying a content reference and setting up permissions, test the content reference.

1. Click **Home** in the portal navigation header
2. Use the Menu pagelet or the Main Menu to navigate to the content reference.
3. Click the content reference.

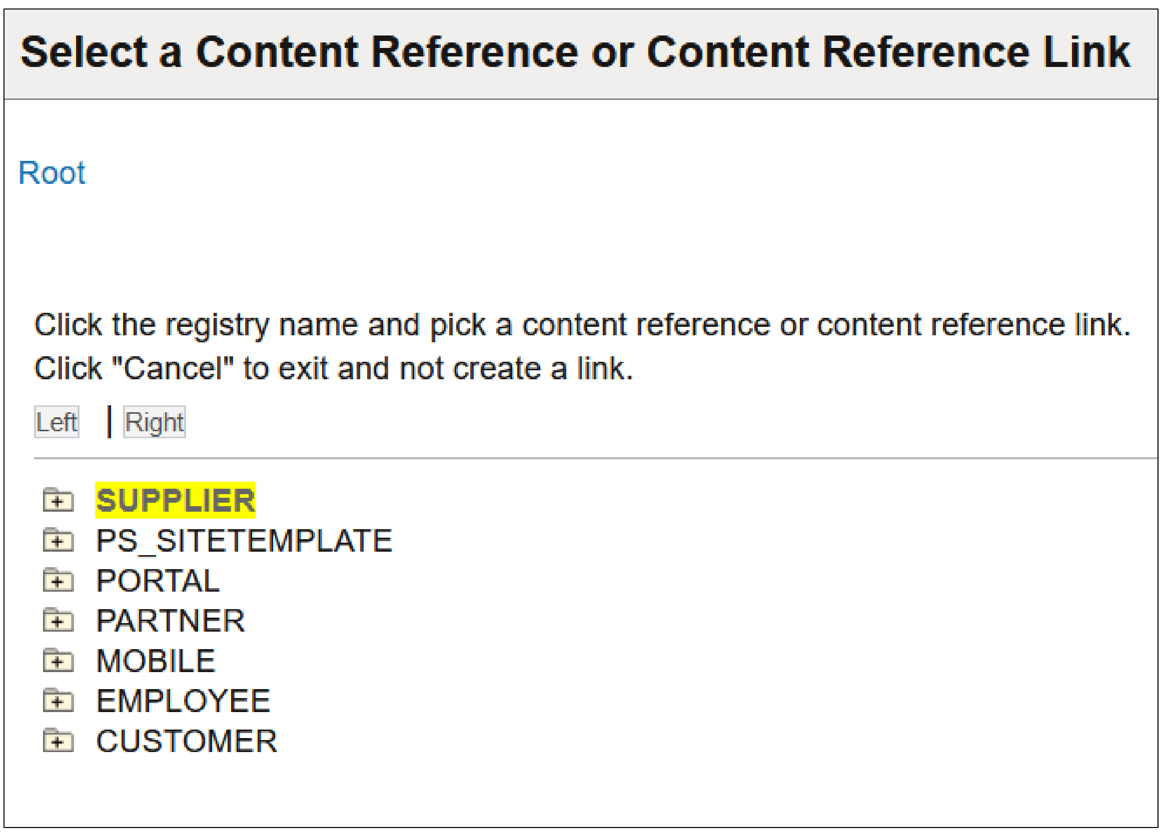
You can also click the **Test Content Reference** on the Content Ref Administration page.

Note: The **Test Content Reference** link appears on the Content Ref Administration page when the usage type of the content reference type is set to *Target* only.

Defining Content Reference Links

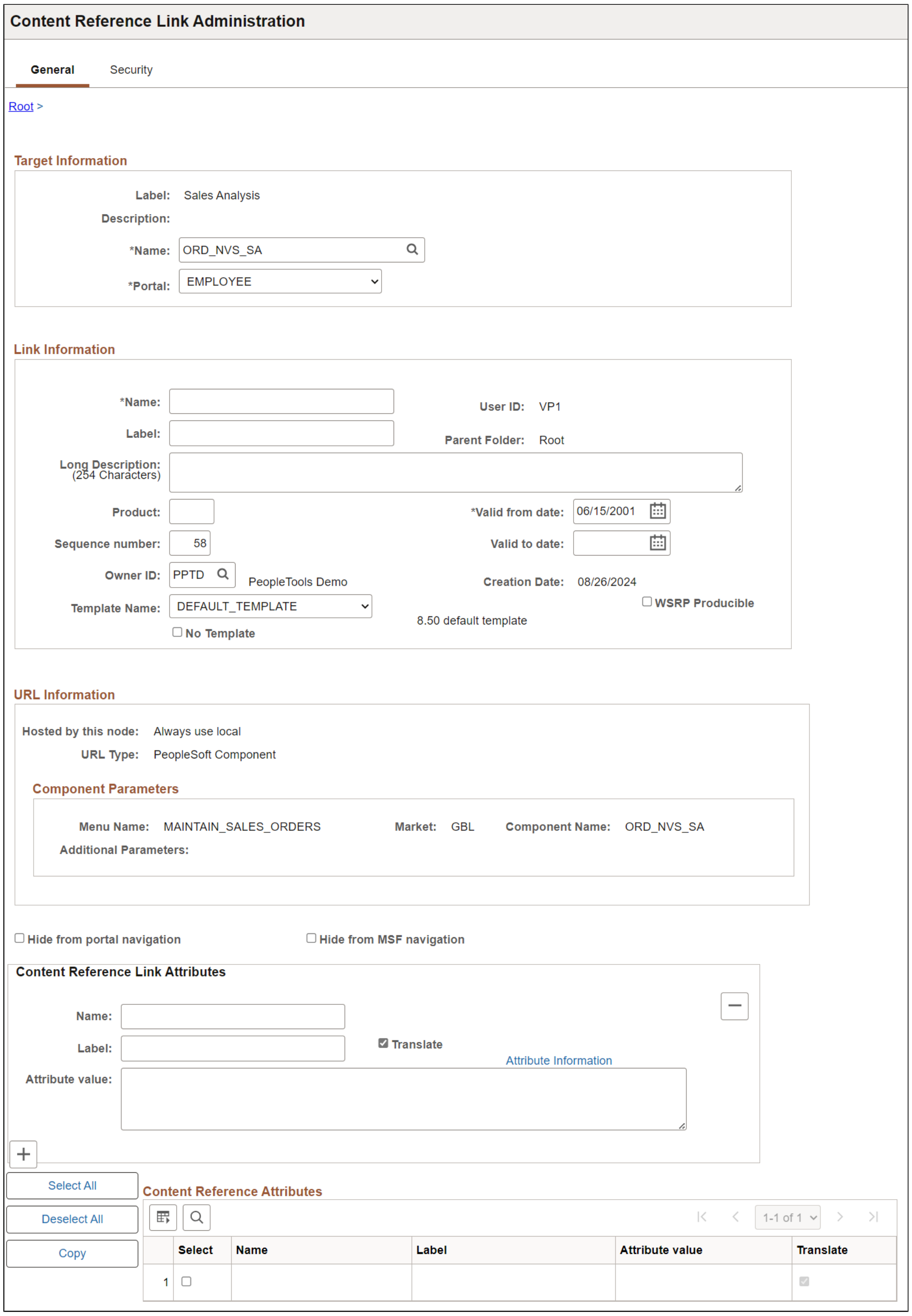
Access the Select a Content Reference or Content Reference Link page (Select **PeopleTools** > **Portal** > **Structure and Content.** Navigate the portal registry hierarchy into the parent folder in which you want the link to appear. Click the **Add Content Reference Link** link that appears after the Content References group box.)

The following example illustrates the Select a Content Reference or Content Reference Link page.



Click the folder icons to navigate the tree structure until you find the target of the link. Click the link and the Content Reference Link Administration page appears.

The following example illustrates the Content Reference Link Administration page.



PeopleSoft portals store content references based on their URLs, so you can't register two different content references that specify the same URL. Content reference links overcome this limitation by enabling you to define more than one menu item that invokes a given URL. You define a content reference link like any other content reference, but instead of specifying a URL, you specify an existing content reference as its destination. At runtime, clicking the content reference link on the portal menu connects to the destination content reference by URL.

Note: To the user, a content reference link looks and works the same as a content reference on the portal menu.

A content reference link can point to:

- A content reference of usage type *Target.*
- Another content reference link.

You can define a content reference link in the same locations that you can define a content reference of usage type *Target.* Multiple content reference links can connect to a single destination content reference.

By connecting one content reference link to another content reference link, you can create a chain of redirection, which always ends in a content reference of usage type *Target.* You don't need to know what the ultimate destination is when you define your new content reference link, and you can duplicate the property settings of the content reference link to which it connects rather than those of the ultimate destination. However, Oracle recommends that you limit your use of chains of content reference links because they impose processing overhead on your portal when selected.

Warning! If you delete a destination content reference or content reference link, all content reference links that connect to it are also deleted. This behavior propagates back through any content reference link chain.

Target Information

Field or Control	***Description***
Name	Select the destination content reference to which you want this content reference link to connect.
Portal	Select the portal that contains the destination content reference to which you want this content reference link to connect. The *Always use local* value refers to the portal where the content reference link is defined, so moving or copying it to a different portal changes the resolution of this value. **Important!** If you change this value, you must also select a new link content reference.

Link Information

Field or Control	***Description***
Name	Enter a unique object name for this content reference link. This field is required.
Test Content Reference Link	Click to test this content reference link. Clicking this button should produce the same result as clicking the content reference link from the portal menu.

If you omit values in following fields, their values are inherited from the destination content reference:

- **Label.**
- **Long Description.**
- **Product.**
- **Sequence number.**
- **Object owner identifier.**
- **No Template.**
- **Template Name.**
- **Valid from date.**
- **Valid to date.**

URL Information

These fields are display-only and reflect the URL information of the content reference that you are linking to.

See “URL Information” in Defining Content References.

Content Reference Attributes

If the destination content reference has defined content reference attributes, they appear in this grid. You can copy them from the grid to the **Content Reference Link Attributes** group box using the following page elements:

Field or Control	***Description***
Select	Select to designate a content reference attribute for copying to the **Content Reference Link Attributes** group box.
Select All	Click to designate all content reference attributes for copying to the **Content Reference Link Attributes** group box.
Copy	Click to copy the selected content reference attributes to the **Content Reference Link Attributes** group box.

Content Reference Link Attributes

Content reference link attributes and content reference attributes share the same page fields.

You can define content reference link attributes directly in this group box, or you can copy an attribute from the destination content reference using the **Content Reference Attributes** grid, and modify it to suit your purposes.

At runtime, the content reference link attributes defined here are combined with the content reference attributes defined at the link destination, and they are applied together.

Important! If a content reference link attribute defined here has the same name as a content reference attribute defined at the link destination, but a different attribute value, the value defined here takes precedence and is applied at runtime.
Content reference link attributes are independent and separate from the content reference attributes at the link destination. Modifying one has no effect on the other.

Setting Content Reference Link Permissions

Access the Content Reference Link Security page (Click the **Security** tab on the Content Reference Link Administration page.)

The fields on this page are disabled because content reference links inherit the security settings of the destination, linked-to object.

Note: While a content reference link does inherit cascaded security from it's own parent folder, it does not inherit cascaded security from the linked-to content reference's parent folder.

Related Links

Setting Content Reference Permissions

Managing Folders and Content References

This section discusses how to:

- Copy folders and content references.
- Move folders and content references.
- Delete folders and content references.
- Synchronize portal object security.
- Register components in multiple portals.
- Register components multiple times.

Copying Folders and Content References

Access the Copy Object to a Different Portal Registry page (Select **PeopleTools** > **Portal** > **Structure and Content** > **Portal Objects** > **Content Ref Administration.** Click the **Copy Object** button).

To copy an object:

1. Navigate to the portal registry to which you want to copy the object.

   Click the portal name to copy the object to root level. To expand the folders and navigate further into the registry structure, click the folder icon (not the folder name).

Important! Clicking the folder name starts the copying process.

2. (Optional) If you are copying a folder, the **Recurse subfolders?** check box appears. Select this check box to copy the folder and all the subfolders and content references within it.

3. Click the folder name to copy the object into that folder.

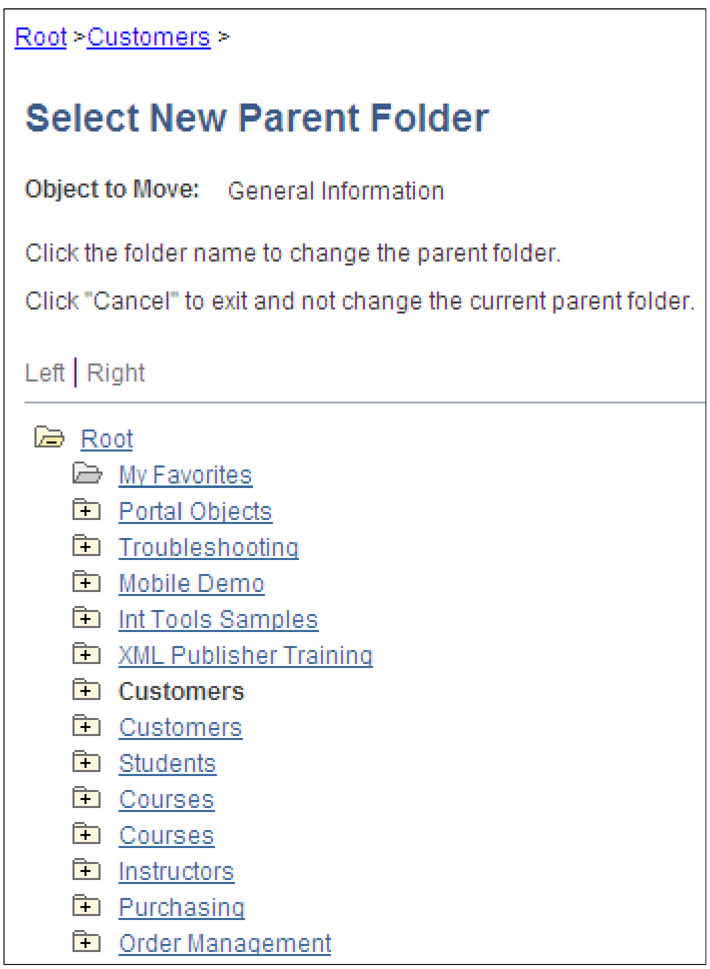
4. Click the **Yes - Copy** button to proceed.

Warning! If you attempt to copy an object to a location that already contains an object of the same name and type, the newly copied object replaces the existing object. For example, if you copy a folder and subfolders to another portal where a subfolder by that name already exists, the existing subfolder is deleted and replaced with the subfolder from the source portal. No warning message appears.

Moving Folders and Content References

Access the Select New Parent Folder page (**PeopleTools** > **Portal** > **Structure and Content** > **Portal Objects** > **Content Ref Administration** Click the **New Parent Folder** button).

The following example illustrates moving folders and content references on the Select New Parent Folder page.



The current parent folder is shown in bold type, and the folder tree is expanded just enough to show the location of the current parent folder. All other folders are shown collapsed. You can expand the tree to show content references, which are displayed as tree leaves and appear in normal text.

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

1. Navigate to the new parent folder by clicking the appropriate folder icons with plus signs.

   Note: When navigating, do not click the folder name unless it's the name of the new parent folder that you want to select. Once you click a folder name, as opposed to the icon, the folder name that you selected becomes the new parent folder.

2. Select the new parent folder by clicking the folder name.
3. Click **Save.**

Note: When you move a content reference that points to a component or script at the default local node, its security settings are cascaded up to its new parents all the way up to the root folder. In addition, when you move a folder, its security settings (not including security inherited from a subordinate object) are cascaded up to its new parent, all the way up to the root folder. This ensures that the folder or content reference is accessible in its new location.

Deleting Folders and Content References

To delete a folder or content reference:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**
2. In the portal registry, navigate to the folder or content reference that you want to delete.
3. Click the **Delete** button next to the object that you want to delete.

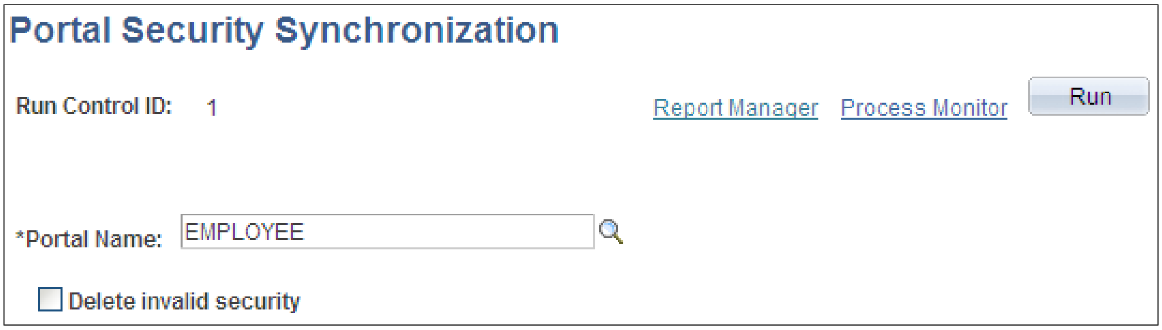
   All child folders and child content references are deleted when you delete the selected folder. Make sure that no critical dependencies exist for these objects before deleting the parent folder.

Warning! Do not delete folders or content references that are delivered with PeopleTools. These default objects have special significance to your portal. Changing them could make your portal unusable.
When you delete a content reference or a content reference link, all content reference links that connect to it are also deleted. This behavior propagates back through any content reference link chain.

Synchronizing Portal Object Security

Access the Portal Security Synchronization page (**PeopleTools** > **Portal** > **Sync Portal Security**).

The following example illustrates the Portal Security Synchronization page.



The hierarchical relationships and dependencies between objects in the portal registry determine what security settings each object must have. The portal won't work correctly if these security relationships aren't maintained. Here are some examples of these relationships:

- A folder that is not public or hidden must have at least the level of access that its immediate child objects (folders, content references, and content reference links) have.
- A content reference link must have exactly the same level of access as the object (content reference or content reference link) to which it links.
- A content reference that represents a PeopleSoft component or iScript must have exactly the same level of access as the object that it represents.

Portal object security settings can become unsynchronized when you move portal objects from one database to another using the Project Copy feature in PeopleSoft Application Designer. When you merge projects this way, if the projects contain any portal objects with identical names, the security settings of the portal objects in the last project copied overwrite the security settings of portal objects copied earlier. Also, when a copied portal object doesn't overwrite an existing object, it changes the structure of the resulting portal registry hierarchy.

Use the Portal Security Synchronization page to reinstate the correct security relationships between objects in the portal registry after you copy a project that contains portal objects.

This page enables you to run the Application Engine program PORTAL_CSS from the standard menu navigation. You can use this feature on local portals only, not on portals on remote databases. In addition, from any local portal, you can run this program against another local portal.

To synchronize portal object security, specify a local portal in the **Portal Name** field and click **Run.** The portal objects are synchronized in the following way:

1. The security settings of each content reference are compared to the component or iScript that it represents, and updated to match.
2. The security settings of each content reference link are compared to the content reference or content reference link to which it connects, and updated to match.
3. The security settings of each content reference and content reference link are propagated to its parent folder, in addition to the parent folder's existing settings.

   None of the parent folder's existing security access is reduced.

   Note: The settings aren't propagated if the parent folder is public or hidden.

   ---

4. The security settings of each folder are propagated to its parent folder, in addition to the parent folder's existing settings.

   None of the parent folder's existing security access is reduced.

   Note: The settings aren't propagated if the parent folder is public or hidden.

The following warning messages may appear when security synchronization cannot occur:

- No permissions in PSAUTHITEM for object %1
- Cref %1 points to Menu: %2, Component: %3 which doesn't exist.

  Note: When this message appears, you need to delete the invalid content reference .

The security administrator and portal administrator are responsible for determining why an object has no security or why an object does not exist.

Delete Invalid Security Option

When you move portal objects from one database to another, roles and permission lists assigned to folders and content references on the source database may not exist on the target database and therefore become invalid. Select the **Delete Invalid Security** check box on the Portal Security Synchronization page to remove nonexisting roles and permission lists from folders and content references.

Note: When the **Delete Invalid Security** option is selected, the PORTAL_CSS process runs more slowly because it checks every role and permission list on every portal registry structure. When a nonexisting role or permission list is found on the portal registry structure, it is removed from that portal registry structure.

Registering Components in Multiple Portals

After you use Registration Wizard to register a new component in a portal, you may need to register the component in other portals. You can register components in multiple portals in two ways:

- Use Registration Wizard again.

  Select the second option: **Add this component to a portal registry.** Then reenter the content reference label, long description, and sequence number (if not 1) to match the other portal's entry.

- Copy the component to other portals.

  When you have copied your component to another portal, open that portal to verify that it was copied as anticipated.

See Copying Folders and Content References.

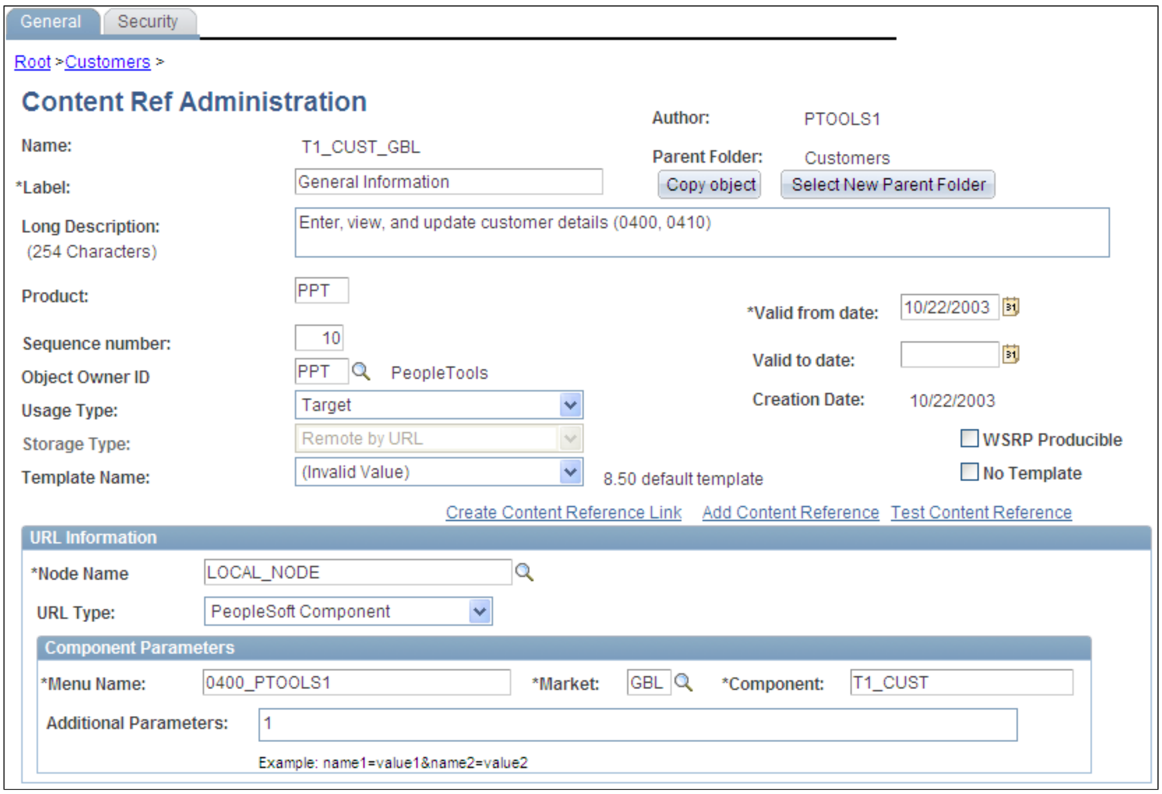
Related Links

Changing Default Portals

Registering Components Multiple Times Within a Portal

The portal registry does not allow a URL to be registered more than once. When you define a content reference, the component name is part of the URL. If you must register the same component URL in more than one location, add a unique character to the parameter field. (Be sure that you really must register a URL twice because this creates upgrade and maintenance concerns).

The following example illustrates that *1* has been added to the **Additional Parameter** field to allow a component to be registered a second time:



Related Links

Registering Components in Multiple Portals

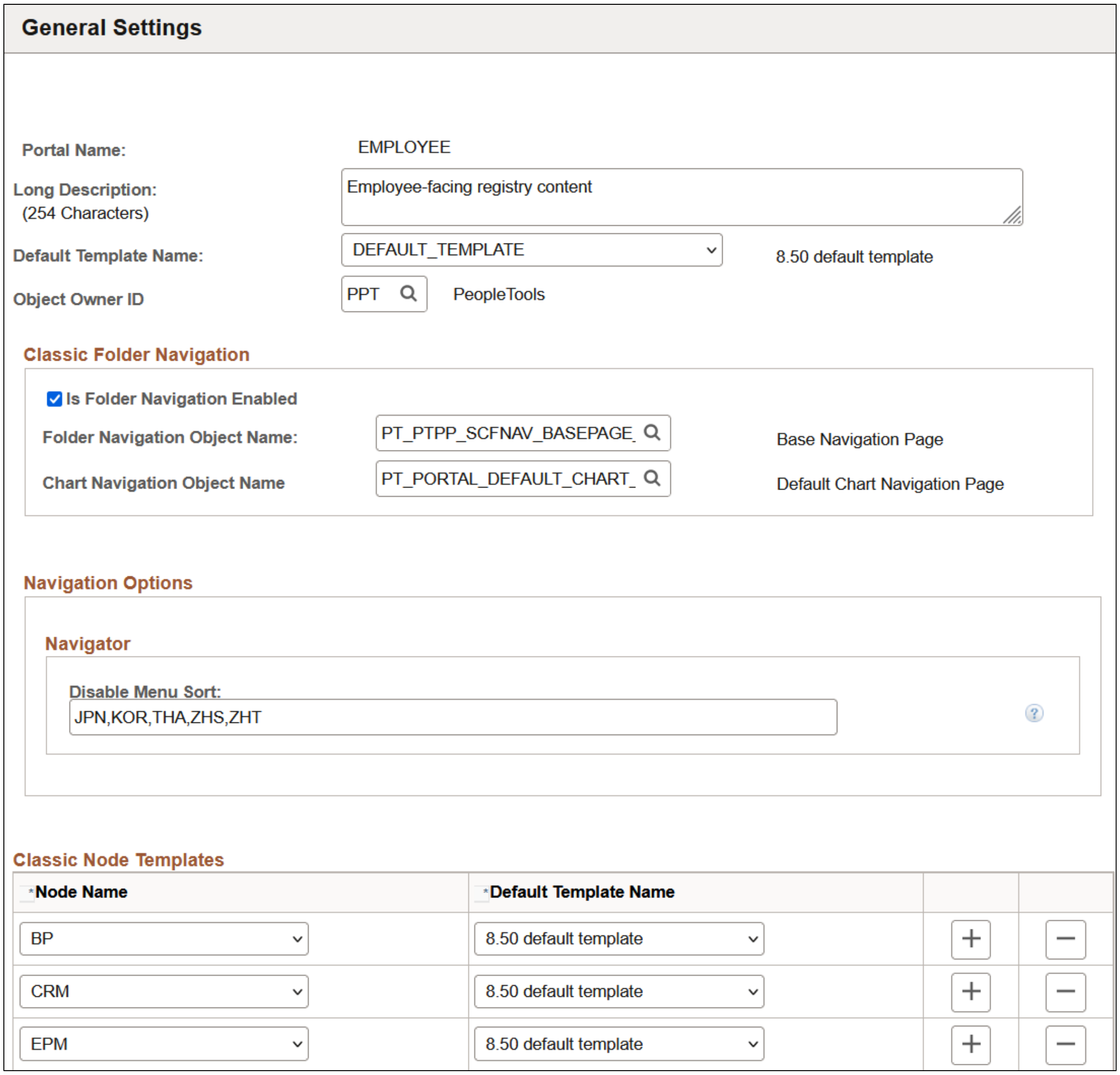
Managing General Portal Settings

This section discusses how to manage general portal settings.

Managing General Settings for Portals

Access the General Settings page (**PeopleTools** > **Portal** > **General Portal Settings**).

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



Use this page to manage the settings for the portal that you're signed in to. To modify other portals, use the Portal Definitions page.

Note: You must sign out of the portal and sign in again before your changes take effect.

Field or Control	***Description***
Default Template Name	Select the template to use to wrap all registered content that has not been associated with another template in this portal, as well as all unregistered content. ***Warning!*** If you change the default template, make sure that the new default is able to handle any arbitrary page.

See Administering Portal Definitions.

Classic Folder Navigation

You enter the default folder navigation values for the current portal on this page. Enabling folder navigation at the portal level enables folder navigation for all standard folders and SmartNavigation in the portal. You can subsequently disable folder navigation at the individual folder level.

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

See Adding and Editing Portal Definitions.

Note: If you disable and then re-enable folder navigation at this level, folder navigation is re-enabled for *all* folders. All overrides at the individual folder level are cleared.

Navigation Options

Field or Control	*Description*
Disable Menu Sort	Enter the PeopleSoft language codes of the languages for which you want to disable the menu sort functionality at the system level. When this feature is disabled, the menu in the NavBar is displayed in standard order only—that is, in the sequence number order that was available prior to PeopleTools 8.59. The menu sort is disabled for these languages by default: Japanese, Korean, Thai, Simplified Chinese, and Traditional Chinese. Enter PeopleSoft language codes in a comma-separated list. Select **PeopleTools** > **Utilities** > **International** > **Languages** to see all PeopleSoft language codes. **Note:** While the drop-down menu has been desupported, this setting applies to the menu in the NavBar

Classic Node Templates

Configure node-specific template information.

Field or Control	*Description*
Node Name	Select from the list of nodes defined for this portal. If you add a new node name, you must also specify the default template for that node.

Field or Control	*Description*
Default Template Name	Select the default template for all content references using this node. The selected template is applied only to content references for which the **No Template** check box is deselected but no template is selected.

Related Links

"Using Menu Pagelets" (Applications User's Guide)

Managing Portal Objects

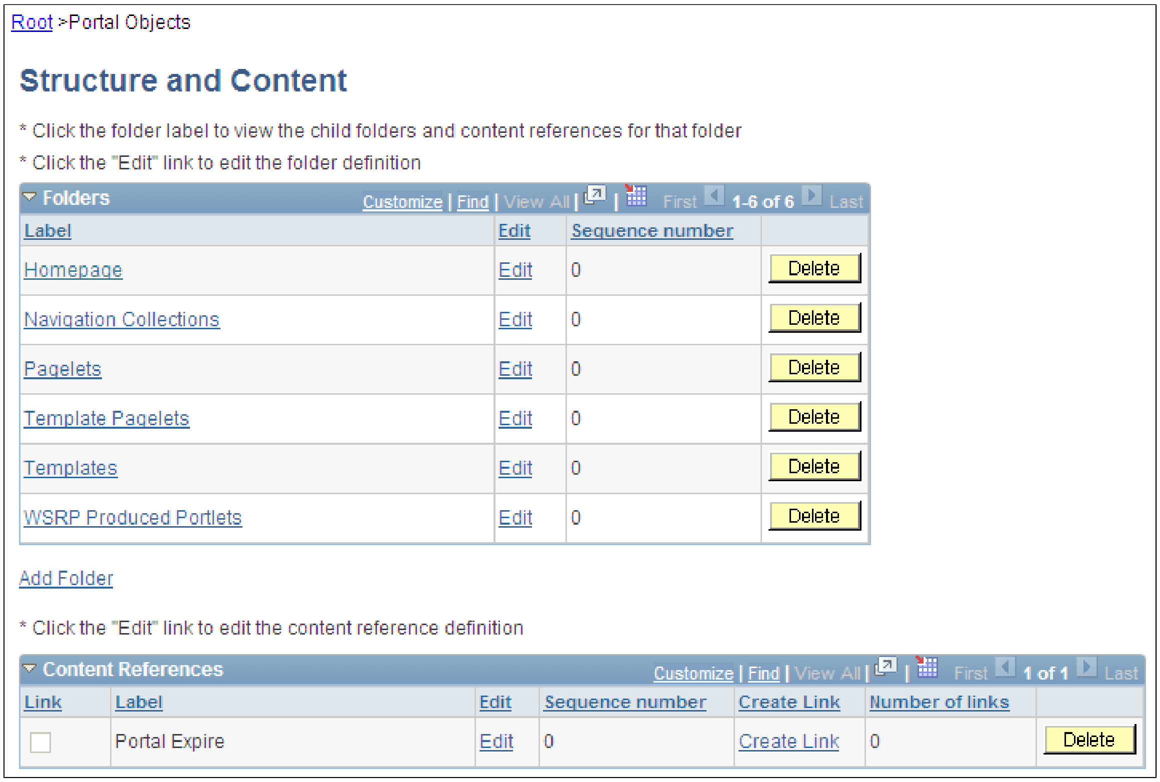
The Portal Objects folder is of special significance to the portal. This section discusses how to:

- View Portal Objects folder settings.
- Manage templates.
- Add and edit templates.
- Delete templates.

Viewing Portal Objects Folder Settings

Access the Structure and Content page (Select **PeopleTools** > **Portal** > **Structure and Content.** In the **Folders** group box, click the **Edit** link for Portal Objects).

This example illustrates the contents of the Portal Objects folder.

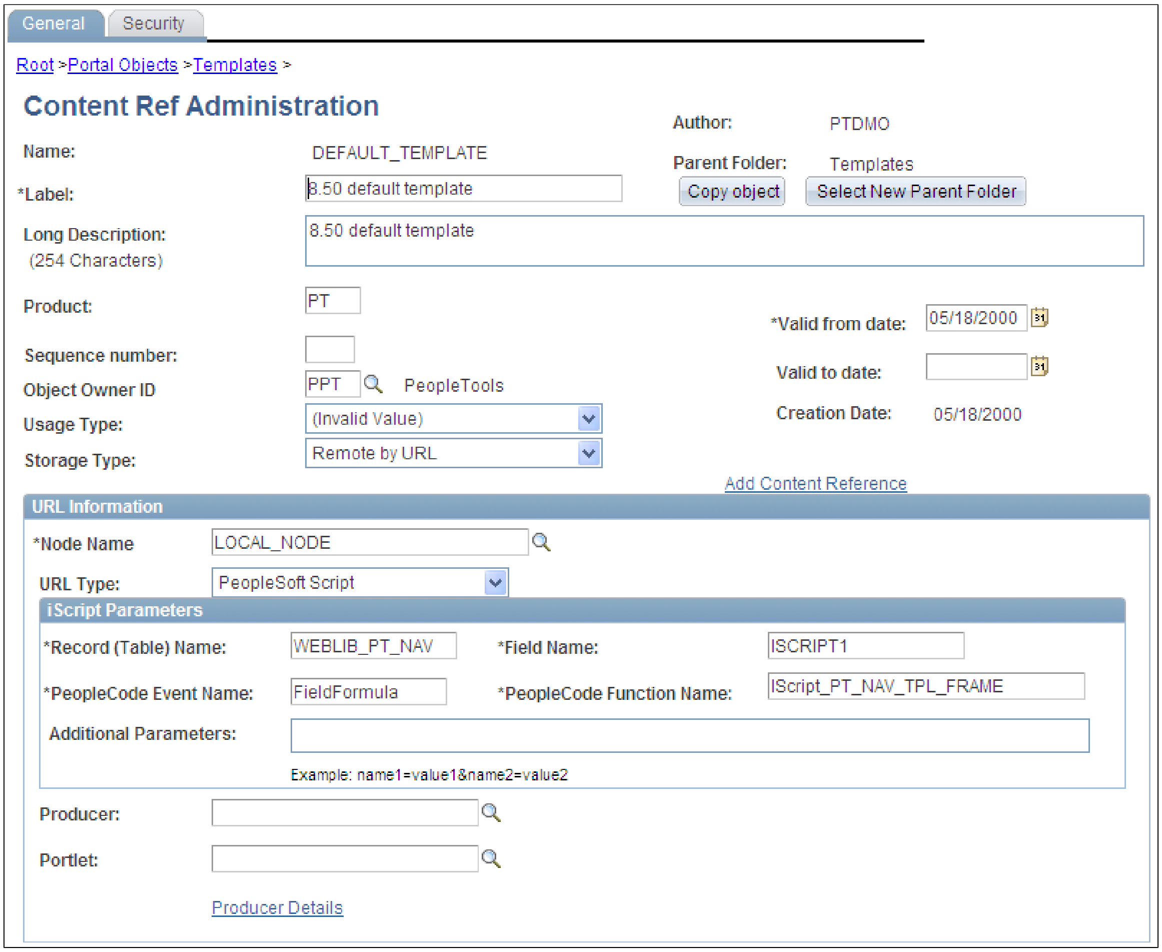


Understanding Templates

Templates are stored in the HTML catalog and are edited through the portal administration pages. For the portal to process templates properly, they must be referenced in the Templates folder within the Portal Objects folder. Storing templates in the HTML catalog enables them to be upgraded like other objects.

Templates are stored in the HTML catalog according to the following naming convention: Template names always begin with the literal *PR_*. Here's an example name: PR_*PORTALNAME_CONTENT REFERENCE ID*

This example illustrates a content reference definition for a portal template:



Templates can be larger than the maximum size permitted by the HTML catalog. Such templates are automatically divided into several files in the HTML catalog. In this case, a letter is appended to each HTML object that makes up the template, as shown here:

PR_EMPLOYEE_123456789_A

PR_EMPLOYEE_123456789_B

Related Links

Understanding Portal Templates

Adding and Editing Templates

Access the Content Ref Administration page (Select **PeopleTools** > **Portal** > **Structure and Content.** Then click these links in succession: Portal Objects, Templates).

To add a new template, click **Add Content Reference** in the **Content References** region at the bottom of the Content Ref Administration page.

To edit an existing template, click the **Edit** link next to the template that you want to edit in the **Content References** region at the bottom of the Content Ref Administration page.

See Defining Content References.

The fields for adding a dynamic template are the same as for any content reference, except for additional iScript parameter fields. These fields are enabled only if the storage type is set to *Remote by URL.*

If the **URL Type** field is set to *PeopleSoft Script,* the **iScript Parameters** group box appears. Select the record name, PeopleCode event name, field name, and PeopleCode function name that stores the iScript that the content reference accesses to build the template.

Related Links

Defining Folders

Defining Content References

Deleting Templates

You delete templates the same way that you delete folders and content references.

See Deleting Folders and Content References.

Note: Do not delete the DEFAULT_TEMPLATE portal template. This template has special significance to your portal and must always be available. Deleting this template could make your portal inoperative. Oracle recommends that you not delete any template unless you are fully aware of how it's used within the portal.

Reviewing Menu Item Information

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

- View menu item details.
- View menu folder structure information.
- View menu security details.

Common Elements Used in This Section

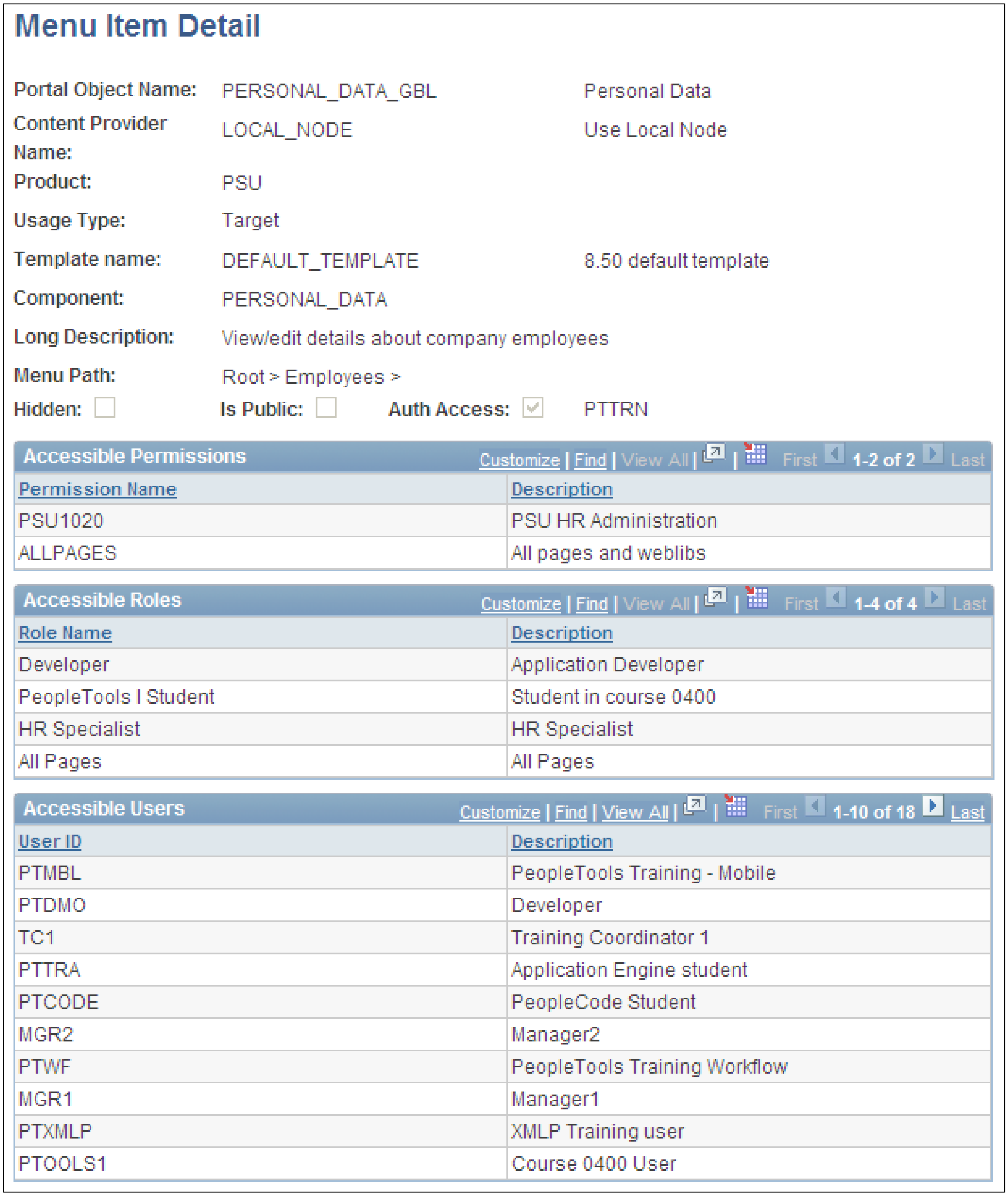
Field or Control	***Description***
Menu Item	Displays the menu item ID. The menu item label is displayed to the right.
Content Provider Name	Displays the database node that contains the PeopleSoft page.

Field or Control	*Description*
Product	(Optional) Displays text that is used to tag menu items for inquiry. **Note:** Each PeopleSoft product line has a naming standard. For example, PAPP is always used for PeopleSoft Interaction Hub.
Menu Path	Displays the navigation path for this menu item.
Hidden	If this check box is selected, then the menu item is hidden from portal menu navigation. However, you can still access the content by using a link that is incorporated on a target page or pagelet.
Is Public	If this check box is selected, the menu item is available for all users. No data appears in the **Accessible Permissions, Accessible Roles,** and **Accessible Users** scroll areas.
Auth Access (author access)	If this check box is selected, the author can access the menu item regardless of how other permissions are set.

Viewing Menu Item Details

Access the Menu Item Detail page (**PeopleTools** > **Portal** > **View Menu Item Detail**).

This example illustrates the fields and controls on the Menu Item Detail page. You can find definitions for the fields and controls later on this page.



Menu Item Detail

Portal Object Name: PERSONAL_DATA_GBL Personal Data

Content Provider Name: LOCAL_NODE Use Local Node

Product: PSU

Usage Type: Target

Template name: DEFAULT_TEMPLATE 8.50 default template

Component: PERSONAL_DATA

Long Description: View/edit details about company employees

Menu Path: Root > Employees >

Hidden: Is Public: Auth Access: PTTRN

Accessible Permissions — Customize | Find | View All | | First 1-2 of 2 Last

Permission Name	Description
PSU1020	PSU HR Administration
ALLPAGES	All pages and weblibs

Accessible Roles — Customize | Find | View All | | First 1-4 of 4 Last

Role Name	Description
Developer	Application Developer
PeopleTools I Student	Student in course 0400
HR Specialist	HR Specialist
All Pages	All Pages

Accessible Users — Customize | Find | View All | | First 1-10 of 18 Last

User ID	Description
PTMBL	PeopleTools Training - Mobile
PTDMO	Developer
TC1	Training Coordinator 1
PTTRA	Application Engine student
PTCODE	PeopleCode Student
MGR2	Manager2
PTWF	PeopleTools Training Workflow
MGR1	Manager1
PTXMLP	XMLP Training user
PTOOLS1	Course 0400 User

Use this page to review the portal registry information for a menu item. The details that appear include the exact users and the roles and permission lists that allow access to a given menu item.

Field or Control	***Description***
Usage Type	Displays the type of object that is registered in the portal as a content reference. Possible values are *Target, Pagelet, Frame Template, HTML Template,* and *Homepage Tab.*
Template Name	Displays the template that controls how content appears on the page. The default template is used except for special circumstances.
Component, iScript, and **URL Type**	Information that appears depends on the URL type that is entered on the Content Ref Administration page. Possible URL type values are *Non-PeopleSoft URL, PeopleSoft Component, PeopleSoft Generic URL, PeopleSoft Script, PeopleSoft Worklist URL,* and *Remote Portlet.* The availability of these values depends on the selected usage type.
Long Description	Displays the text that appears when you move the cursor over the menu navigation link.

Note: The **Accessible Permissions, Accessible Roles,** and **Accessible Users** scroll areas are populated only if the access type is set to *permission list.*

See Defining Content References.

Accessible Permissions

This section displays the permission list name and description for all permission lists that can access this menu item.

Note: If the component or iScript is using a local node with no extra parameters, the permission lists are automatically populated based on the menu object component security.

Accessible Roles

Displays the role name and description for all roles that can access this menu item.

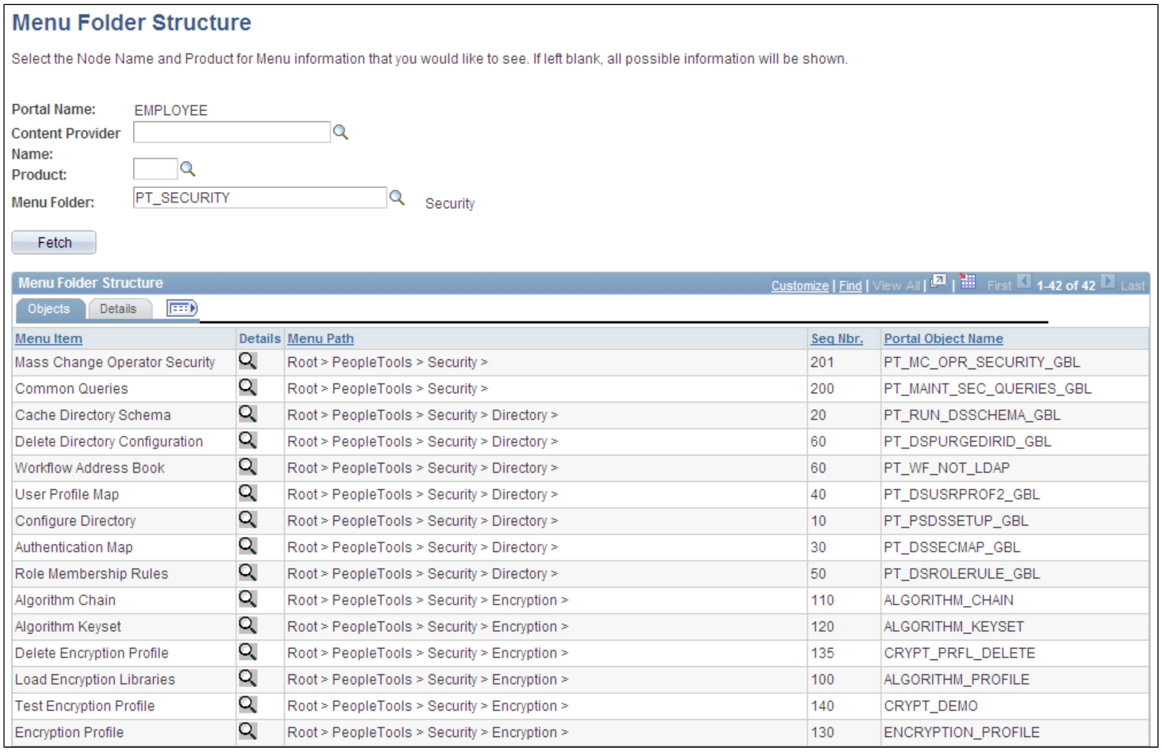
Accessible Users

Displays the user ID and description for all users that can access this menu item.

Viewing Menu Folder Structure Information

Access the Menu Folder Structure page (**PeopleTools** > **Portal** > **View Menu Folder Structure**).

This example illustrates the fields and controls on the Objects tab of the Menu Folder Structure page. You can find definitions for the fields and controls later on this page.



Use this page to view a list of content references for a selected portal registry folder. You see the contents of one or several menu folders.

Field or Control	***Description***
Portal Name	Displays the name of the portal that you are currently using.
Content Provider, Name, Product, and **Menu Folder**	Select a combination of criteria for filtering the results. If any of the fields are left blank, all possible information for those fields are shown.

Objects Tab

Select the Objects tab.

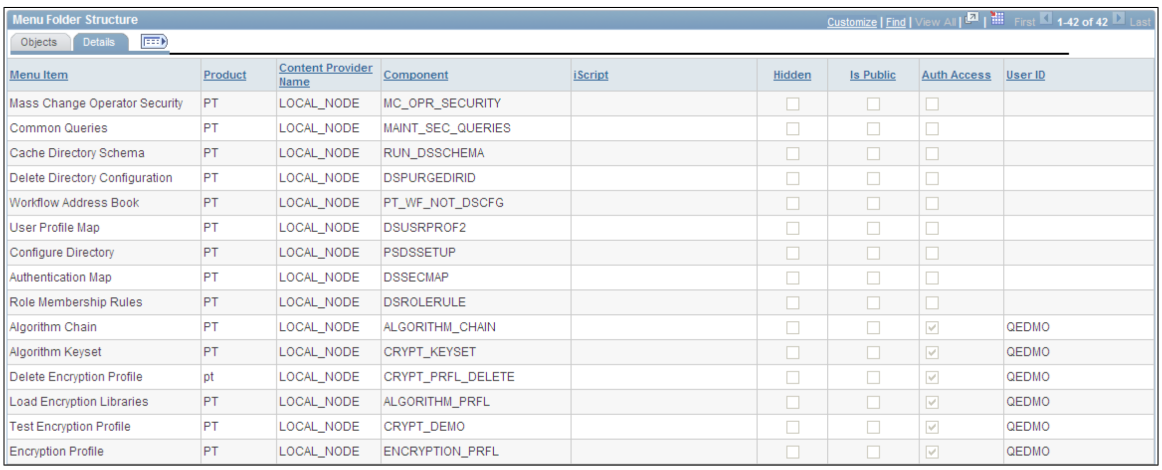
Field or Control	***Description***
Menu Item	Displays the menu item description.
	Click to view more detail information on the Menu Item Detail page.

Field or Control	*Description*
Seq Nbr (sequence number)	Displays where the menu item is located in its menu folder. If this field is blank, the menu item is added to the top of the list of content references within its folder and displayed in alphabetical order.
Portal Object Name	Displays the menu item ID.

Details Tab

Select the Details tab.

This example illustrates the fields and controls on the Details tab of the Menu Folder Structure page. You can find definitions for the fields and controls later on this page.



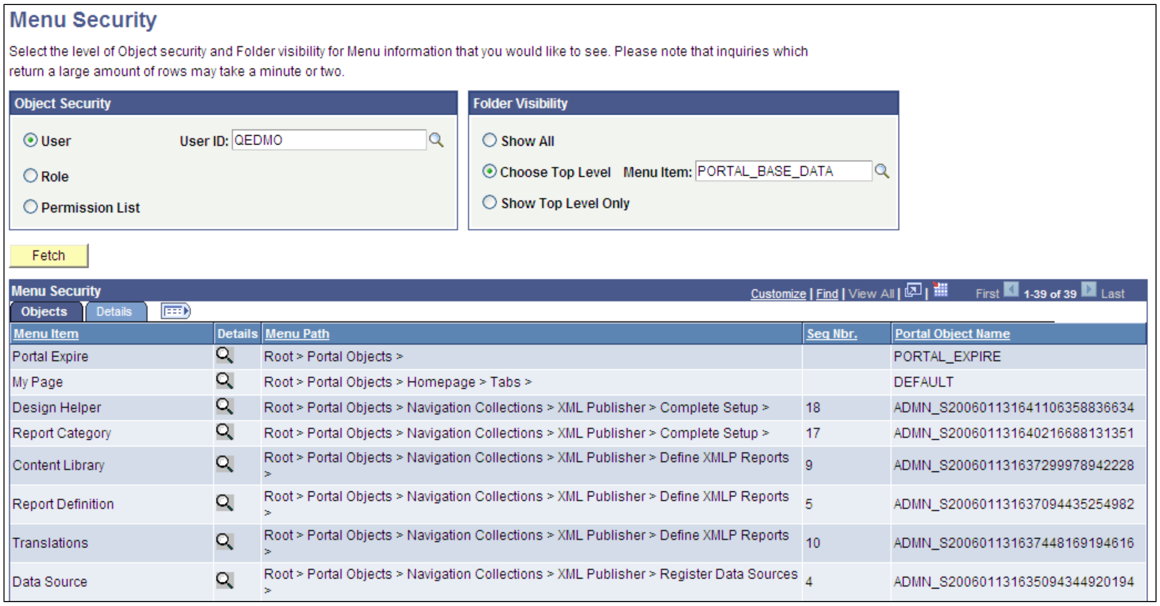
Menu Item	Product	Content Provider Name	Component	iScript	Hidden	Is Public	Auth Access	User ID
Mass Change Operator Security	PT	LOCAL_NODE	MC_OPR_SECURITY		☐	☐	☐	
Common Queries	PT	LOCAL_NODE	MAINT_SEC_QUERIES		☐	☐	☐	
Cache Directory Schema	PT	LOCAL_NODE	RUN_DSSCHEMA		☐	☐	☐	
Delete Directory Configuration	PT	LOCAL_NODE	DSPURGEDIRID		☐	☐	☐	
Workflow Address Book	PT	LOCAL_NODE	PT_WF_NOT_DSCFG		☐	☐	☐	
User Profile Map	PT	LOCAL_NODE	DSUSRPROF2		☐	☐	☐	
Configure Directory	PT	LOCAL_NODE	PSDSSETUP		☐	☐	☐	
Authentication Map	PT	LOCAL_NODE	DSSECMAP		☐	☐	☐	
Role Membership Rules	PT	LOCAL_NODE	DSROLERULE		☐	☐	☐	
Algorithm Chain	PT	LOCAL_NODE	ALGORITHM_CHAIN		☐	☐	☑	QEDMO
Algorithm Keyset	PT	LOCAL_NODE	CRYPT_KEYSET		☐	☐	☑	QEDMO
Delete Encryption Profile	pt	LOCAL_NODE	CRYPT_PRFL_DELETE		☐	☐	☑	QEDMO
Load Encryption Libraries	PT	LOCAL_NODE	ALGORITHM_PRFL		☐	☐	☑	QEDMO
Test Encryption Profile	PT	LOCAL_NODE	CRYPT_DEMO		☐	☐	☑	QEDMO
Encryption Profile	PT	LOCAL_NODE	ENCRYPTION_PRFL		☐	☐	☑	QEDMO

Field or Control	*Description*
User ID	If the **Auth Access** check box is selected, this field displays the author's user ID.

Viewing Menu Security Details

Access the Menu Security page (**PeopleTools** > **Portal** > **View Menu Security**).

This example illustrates the fields and controls on the Objects tab of the Menu Security page. You can find definitions for the fields and controls later on this page.



Use this page to view a list of folders or content references that a user ID, role, or permission list can access.

Note: The **Seq Nbr** (sequence number) column is hidden in this illustration, and because of the large number of objects, not all are shown.

Object Security

Select the level of object security for the menu information that you want to see.

Field or Control	***Description***
User, **Role**, and **Permission List**	Select a type of object security, and then select an object. You can view the complete menu folder structure that is available in the navigation (or hidden, but accessible from another page) for the selected security value.

Warning! To prevent unexpected errors and inordinately long retrieval times, do *not* select the *ALLPAGES* permission list in conjunction with the *Show All* folder option. Always narrow your folder options if you select the *ALLPAGES* permission list.

Folder Visibility

Select the level of visibility for the menu information that you want to see.

Field or Control	*Description*
Show All	Select to view all objects. **Note:** To avoid unexpected errors and inordinately long retrieval times, Oracle recommends that you not select this option if you select the *ALLPAGES* permission list in the **Object Security** section.
Choose Top Level	Select a level-one folder. The **Menu Security** grid displays all content references and their complete paths below this level that are accessible by the selected object security level.
Show Top Level Only	Select to display only the level-one folders that are accessible by the object security type.

Permission Lists

If either the **Role** object security option or the **Show Top Level Only** folder visibility option is selected, then this scroll area appears, displaying the permission lists that are associated with each menu item.

Publishing Pagelets to Other Portals

This section discusses how to:

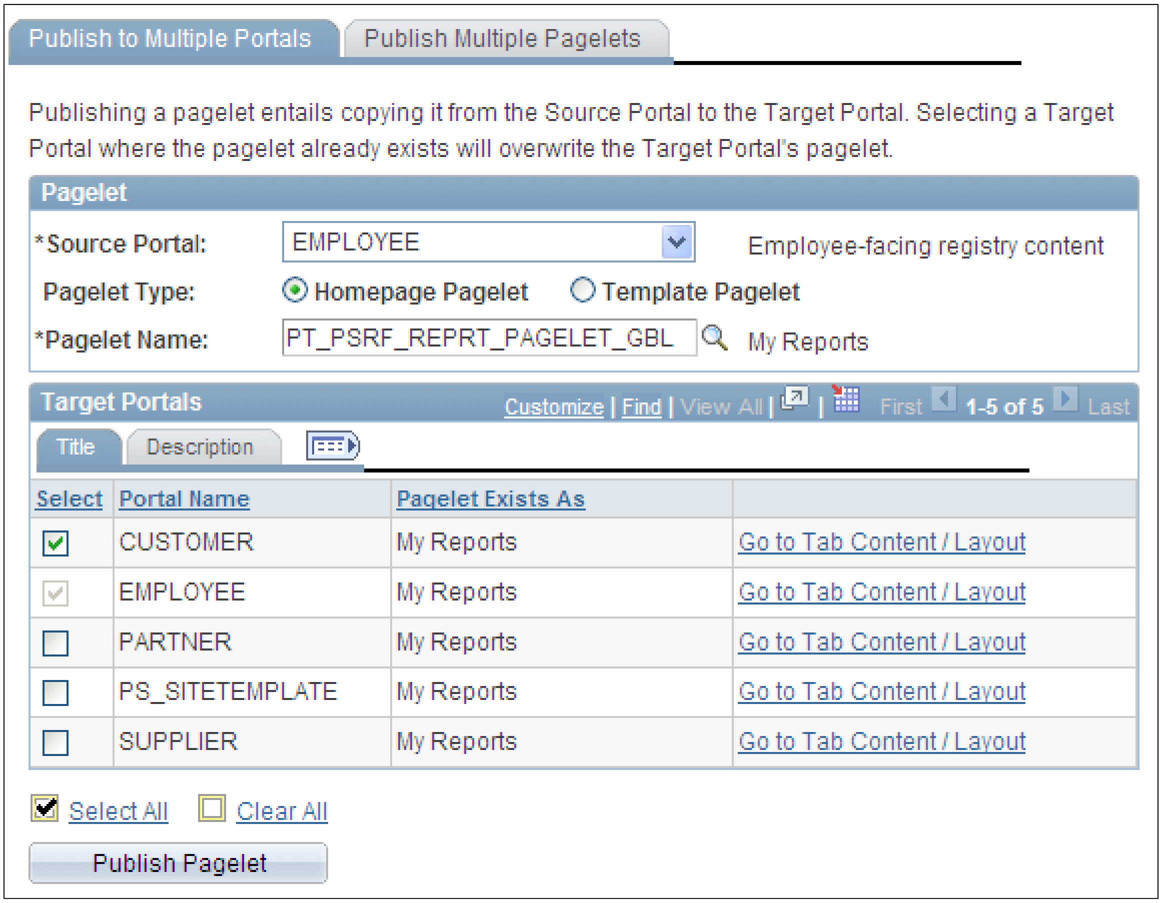
- Publish a pagelet.
- Publish pagelets to multiple portals.

Note: The Publish Pagelets component can be used with any pagelets including Pagelet Wizard pagelets.

Publishing Pagelets to Multiple Portals

Access the Publish to Multiple Portals page. (Select **PeopleTools** > **Portal** > **Portal Utilities** > **Publish Pagelets** > **Publish to Multiple Portals**.)

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



Pagelet

Field or Control	***Description***
Source Portal	Select the portal from which to copy a pagelet. You can select from portals that reside in the same database where you are currently signed in.
Pagelet Type	Select the type of pagelet that you want to copy. Options are: *Homepage Pagelet:* A homepage pagelet is a version of a pagelet that is available for use on the portal homepage and can be configured by the end user. *Template Pagelet:* A template pagelet is a version of a pagelet that can be added to a PeopleSoft template. These pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful when Context Manager or WorkCenter pages are used to display the template pagelet.

Field or Control	*Description*
Pagelet Name	Select a pagelet that exists in the selected Source Portal registry. This is the pagelet content reference that you want to publish to one or more other portal registries that reside in the database where you are currently signed in.

Target Portals

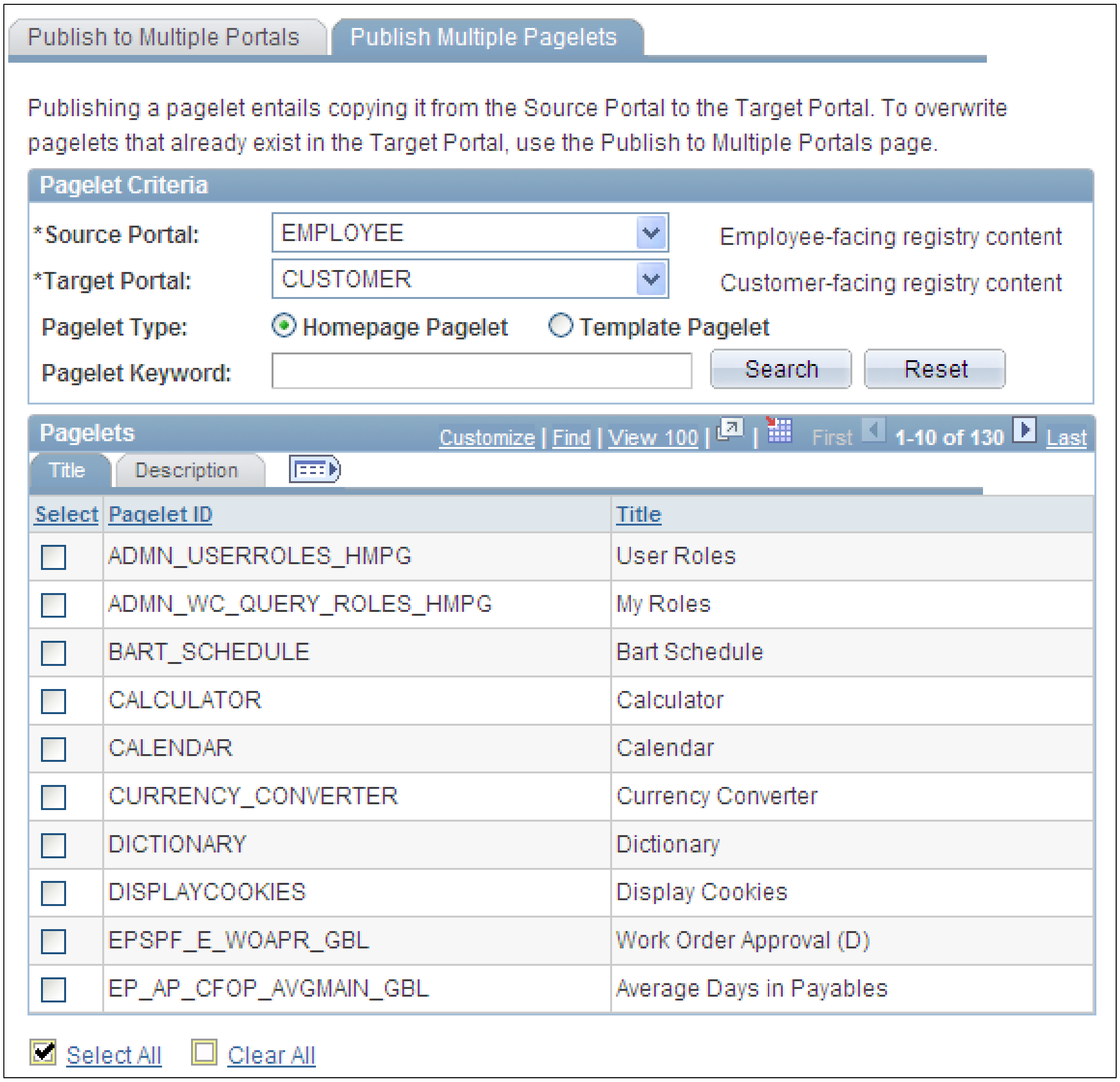
Warning! Selecting a target portal in which the selected pagelet already exists overwrites the pagelet content reference in the target portal.

Field or Control	*Description*
Select	Select to publish the selected pagelet to the selected target portal. Use the **Portal Name** and **Title** fields to identify the available target portals.
Pagelet Exists As	If the pagelet that you selected to copy and publish to a target portal already exists in the target portal, this field is populated with the pagelet title as it exists in the portal registry. For instance, in the example shown here the Technology News pagelet is selected for copying from the EMPLOYEE source portal to the CUSTOMER target portal. The **Pagelet Exists As** field shows that this pagelet also already exists in the CUSTOMER target portal. If you copy the pagelet to the CUSTOMER portal, the pagelet definition in the CUSTOMER portal is overwritten by the pagelet definition in the EMPLOYEE source portal.
Go to Tab Content/Layout	Click to display the Tab Content and Tab Layout pages where you can specify display settings for the pagelet.
Publish Pagelet	Select to copy the pagelet definition from the source portal and publish it in the selected target portals. Content reference are created in the registries of all selected target portals by means of the same values that exist in the original pagelet registration in the source portal. If the folder in which the original pagelet is registered does not exist for a selected target portal registry, it is generated dynamically.

Publishing Multiple Pagelets to a Portal

Access the Publish Multiple Pagelets page. (Select **PeopleTools** > **Portal** > **Portal Utilities** > **Publish Pagelets** > **Publish Multiple Pagelets**.)

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



Pagelet Criteria

Field or Control	*Description*
Source Portal	Select the portal from which to copy a pagelet. You can select from portals that reside in the same database where you are currently signed in.

Field or Control	***Description***
Target Portal	Select the portal to which you want to copy pagelets. You can select from portals that reside in the same database where you are currently signed in. After you select the source and target portals, the **Pagelet** group box displays a list of pagelets that are registered in the source portal but not the target portal.
Pagelet Type	Select the type of pagelet that you want to copy. Options are: *Homepage Pagelet:* A homepage pagelet is a version of a pagelet that is available for use on the portal homepage and can be configured by the end user. *Template Pagelet:* A template pagelet is a version of a pagelet that can be added to a PeopleSoft template. These pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful when Context Manager or WorkCenter pages are used to display the template pagelet.
Pagelet Keyword	Enter keywords to narrow the scope of pagelets displayed in the **Pagelet** group box.

Pagelets

Field or Control	***Description***
Select	Select to publish selected pagelets to the selected target portal. Use the **Pagelet ID, Title,** and **Description** fields on the Title and Description tabs to identify the pagelet.
Publish Selected Pagelets	Select to copy the pagelet definition from the source portal and publish it to any selected target portals. The portal registry entries for the content references are created in the target portal registry by means of the same values that exist for the original pagelet in the source portal registry. If a folder in which an original pagelet is registered does not exist for a selected target portal registry, it is generated dynamically.

Running Portal Utility Processes

This section provides an overview of portal utility processes and discusses how to:

- Run the Synchronize Collection Security process.
- Run the Delete Empty Folders process.
- Run the Clean Portal Project process.

Understanding Portal Utility Processes

This section discusses:

- The Synchronize Collection Security process.
- The Delete Empty Folders process.
- The Clean Portal Project process.

The Synchronize Collection Security Process

The Synchronize Collection Security (PTPP_SCSYNC) Application Engine process updates security for published custom navigation pages and navigation pagelets by adding permission lists, roles, or both from the content references and folders in the navigation collection to the content references for the published navigation page or pagelet.

The Synchronize Collection Security process runs for navigation collection publications according to the **Security Access** options that are set for the publication on the Publish Collection page. The process runs for navigation collection publications that are published with the *Allow Collection Sync* option selected. The process is not run for navigation collection publications that are published with the *Do Not Allow Collection Sync* option selected. The process also removes extraneous security settings for navigation collections that were published with the *Public Access* option selected.

Run this process when you initially publish or change a navigation page or pagelet with the *Allow Collection Sync* option selected or after security updates to the items within the collection.

See Publishing Navigation Collections.

The Delete Empty Folders Process

The Delete Empty Folders Application Engine process (PTPP_FD_DEL) deletes portal registry folder references that do not contain child folders or content. The process does not delete empty navigation collection folder placeholders or the My Favorites folder. The user ID running the process must have been assigned the security role of Portal Administrator.

Run this process when you are deleting content references during an upgrade or implementation, and you are removing out-of-date navigation and replacing it with the current delivered navigation. You should also run this process after copying a *deleted* portal project.

The Clean Portal Project Process

The Clean Portal Project Application Engine process (PTPP_CPPROJ) removes unnecessary and potentially harmful data from a portal project definition before you copy it to the PeopleSoft Interaction Hub. Run this process as a part of an upgrade or implementation when you want to load navigation from a content provider into the PeopleSoft Interaction Hub.

The Clean Portal Project process removes common objects from the content provider portal project which, because they are common, also exist in the PeopleSoft Interaction Hub. Run this process on the portal project before copying it to the PeopleSoft Interaction Hub so that you don't override these common objects configured for use with the PeopleSoft Interaction Hub with the common objects that were configured in the content provider.

The Clean Portal Project process ensures that you copy only the navigation objects that you think are necessary, such as registry structures, templates, and permissions.

Warning! Navigation collections cannot be rendered in the PeopleSoft Interaction Hub if the content provider registry is hosted by a remote node.
Because navigation collections actively access the registry, a published navigation collection page or pagelet cannot be rendered for a registry that is hosted by a remote node. If you want to render a content provider navigation collection page or pagelet in the PeopleSoft Interaction Hub, the content provider's registry must be hosted by a local node in the content provider database.

See Administering Portal DefinitionsUnderstanding Portal Administration.

Running the Synchronize Collection Security Process

Access the Synchronize Collection Security page. (Select **PeopleTools** > **Portal** > **Portal Utilities** > **Sync Nav Collection Security**.)

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

Request Parameters

Field or Control	***Description***
Portal Name	Select the portal for which you want to synchronize security for published navigation collections.
Collection Name	Select the navigation collection for which you want to synchronize security. Only navigation collections created for the selected portal are available for selection. Leave this field blank if you want to synchronize security for all navigation collection publications that exist in the selected portal. **Note:** Navigation collection pages or pagelets that were published with the *Do Not Allow Security Sync* option selected on the Publish Collection page are not eligible for processing.

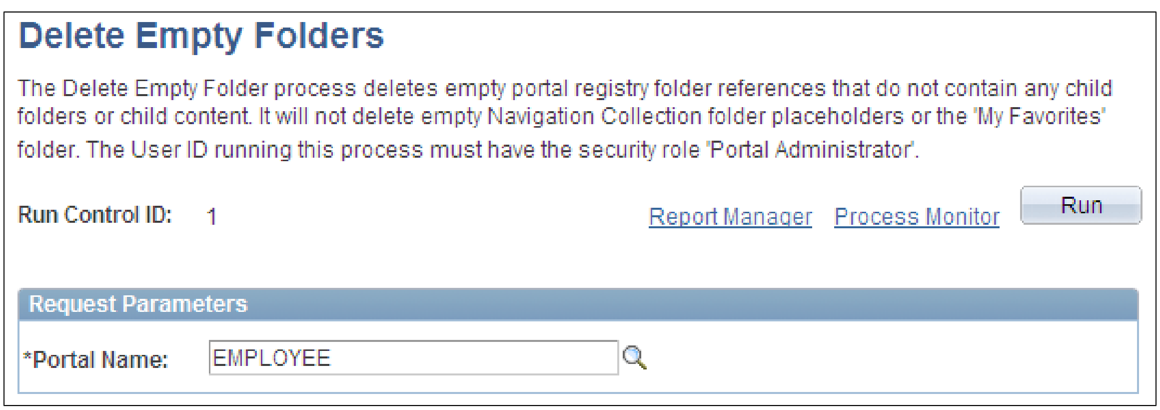
Related Links

Publishing Navigation Collections

Running the Delete Empty Folders Process

Access the Delete Empty Folders page. (Select **PeopleTools** > **Portal** > **Portal Utilities** > **Delete Empty Portal Folders**.)

This example illustrates the fields and controls on the Delete Empty Folders page. You can find definitions for the fields and controls later on this page.



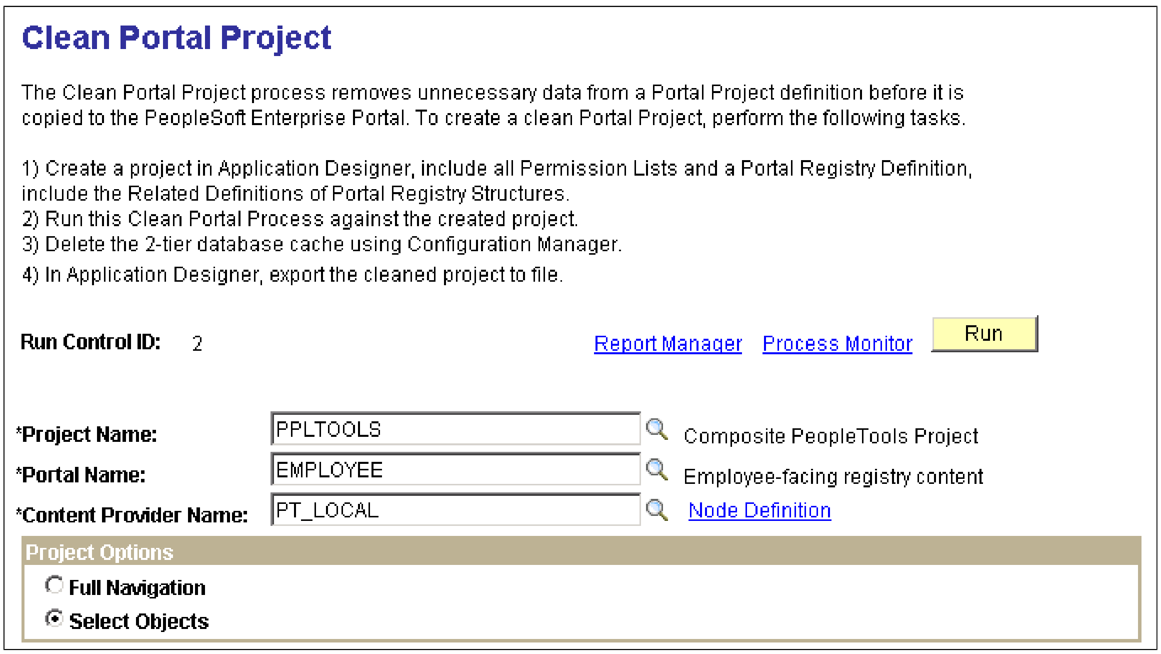
Request Parameters

Field or Control	*Description*
Portal Name	Select the portal for which you want to delete portal registry folder references that do not contain child folders or content.

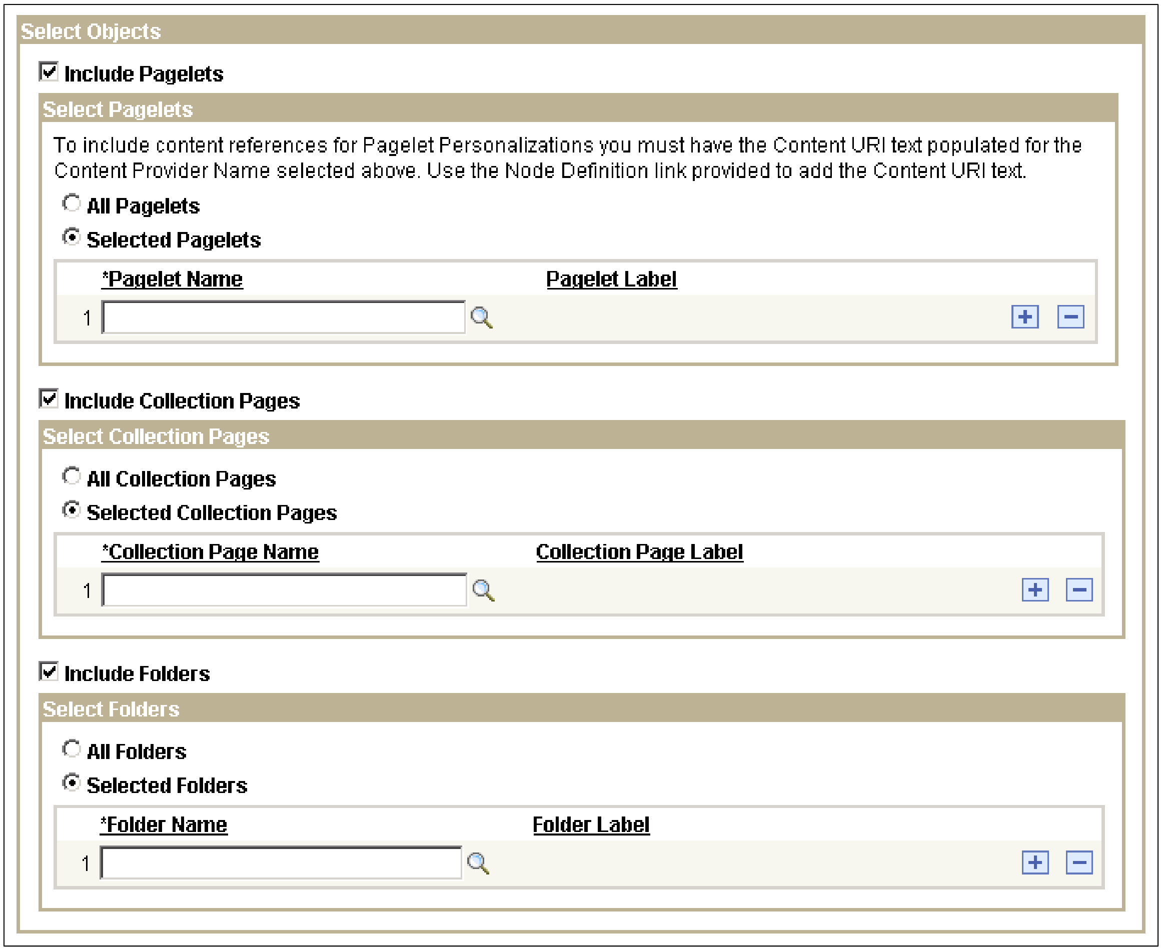
Running the Clean Portal Project Process

Access the Clean Portal Project page. (Select **PeopleTools** > **Portal** > **Portal Utilities** > **Clean Portal Project**.)

This example illustrates the fields and controls on the Clean Portal Project page. You can find definitions for the fields and controls later on this page.



This example illustrates the fields and controls on the Select Objects group box of the Clean Portal Project page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Project Name	Select the portal project that you want to clean for export into the PeopleSoft Interaction Hub. Projects available for selection are those in the database that contain at least one registry structure.
Portal Name	Select the portal that contains the registry structures that you want to copy. Available portal names are derived from the portal project that you selected, as well as from the project data that contains the portal name of the registry structures.
Content Provider Name	Select the content provider node name of the registry structure that you want to include in your portal project. Available node names are derived from the portal project that you selected.

Field or Control	*Description*
Node Definition	Select to access the Node Definitions - Portal page. If you select the Include Pagelets option, you must have a URL value in the node definition to enable the Clean Portal Project process to correctly include personalization pages for the pagelets that you copy over to the PeopleSoft Interaction Hub. Access the Node Definitions - Portal page to enter the content URI text for the content provider node name selected on this page. When you click this link, you are prompted to save or cancel any changes that you made in the component, and you are then transferred to the Node Definitions - Portal page. See Defining Portal Nodes.

Project Options

Field or Control	*Description*
Full Navigation	Select to run the Clean Portal Project process on all objects in the portal project.
Select Objects	Select to run the Clean Portal Project process on select objects in the portal project. When you select this option, the **Select Objects** group box appears, enabling you to select the type of navigation that you want to include in your portal project. Use this option if you don't want to copy and maintain the full content provider navigation structure.

Select Objects

Field or Control	*Description*
Include Pagelets	Select to include pagelets (including navigation collection pagelets) in the portal project in the run of the Clean Portal Project process. When you select this option, the **Select Pagelets** group box appears, enabling you to specify which pagelets you want to be processed. The actual published navigation collection pagelet content references are included for copying to the PeopleSoft Interaction Hub. The navigation collection definitions are not included. Pagelet personalization pages are included in the portal project if the content provider node name contains the correct URL. Click the **Node Definition** link to access the Node Definitions - Portal page, where you can enter this URL.
Include Collection Pages	Select to include navigation collection pages in the portal project in the run of the Clean Portal Project process. When you select this option, the **Select Collection Pages** group box appears, enabling you to specify which navigation collection pages you want to be processed. The actual published navigation collection pages are included for copying to the PeopleSoft Interaction Hub. The navigation collection definitions are not included.
Include Folders	Select to include folders in the portal project in the run of the Clean Portal Project process. When you select this option, the **Select Folders** group box appears, enabling you to specify which folders you want to be processed. Selecting a folder for inclusion in the process also includes all of its children and parents for copying into the PeopleSoft Interaction Hub.

Select Pagelets

Field or Control	*Description*
All Pagelets	Select to include all pagelets in the portal project in the run of the Clean Portal Project process.

Field or Control	Description
Selected Pagelets	Select to include only selected pagelets in the portal project in the run of the Clean Portal Project process. Selecting this option displays the **Pagelet Name** field, enabling you to select specific pagelets that you want to be processed.

Select Collection Pages

Field or Control	Description
All Collection Pages	Select to include all published navigation collection pages in the portal project in the run of the Clean Portal Project process.
Selected Collection Pages	Select to include only selected published navigation collection pages in the portal project in the run of the Clean Portal Project process. Selecting this option displays the **Collection Page Name** field, enabling you to select specific published navigation collection pages that you want to be processed.

Select Folders

Field or Control	Description
All Folders	Select to include all folders in the portal project in the run of the Clean Portal Project process.
Selected Folders	Select to include only selected folders in the portal project in the run of the Clean Portal Project process. Selecting this option displays the **Folder Name** field, enabling you to select specific folders that you want to be processed.

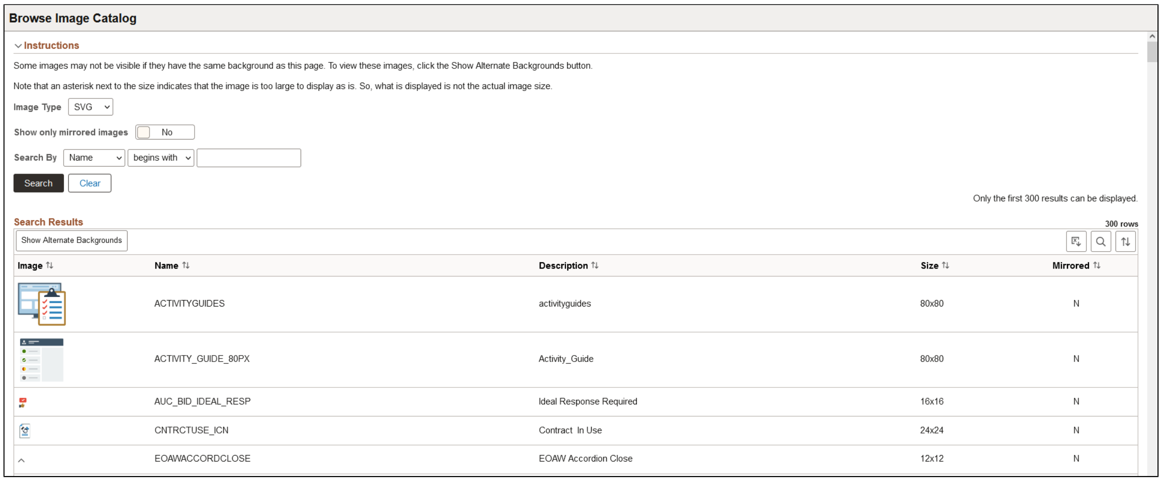
Browsing the Image Catalog

Use the Browse Image Catalog page to review images stored in the database.

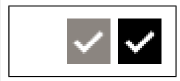
Navigation:

PeopleTools > **Portal** > **Browse Image Catalog**

This example illustrates the fields and controls on the Browse Image Catalog page.



Field or Control	***Description***
Image Type	Select from these image types: • *Bitmap* • *DIB* • *GIF* • *JPEG* • *PNG* • *SVG* – The default value.
Show only mirrored images	Select *Yes* to only retrieve images that are mirrored. Otherwise, both mirrored and unmirrored images are retrieved.
Search By	Select *Name* or *Description* and additional search criteria to limit the search results.
Search	Click the Search button to retrieve the images meeting the search criteria from the database.
Clear	Click the Clear button to clear both the search criteria and the grid.

Field or Control	*Description*
Show Alternate Backgrounds	Click the Show Alternate Backgrounds button to redisplay each image in the current grid on three background colors: white, gray, and black. This option allows you to view white images, which are not visible on the default white background. In this example, the white check mark is visible on the gray and black backgrounds: 

Related Links

“Using Images” (Application Designer Developer’s Guide)

Chapter 5

Using PeopleTools Branding Features

Understanding PeopleTools Branding

This topic provides an overview of PeopleTools branding features.

Overview of PeopleTools Branding

PeopleTools provides branding features that are powerful and flexible, allowing you to manage the look and feel of any PeopleSoft application. With PeopleTools, many common branding tasks, including managing definitions and objects associated with the overall site style are performed online. In addition, other tasks such as maintaining style sheets, images, and other objects can be performed either online or by using Application Designer.

With PeopleTools branding, you can apply branding definitions to:

- The entire system.
- Individual portals.
- Users by role or permission list.
- Individual classic components.
- Fluid components by node.

The key branding definition is the theme, which can be applied system wide, to specific portals, and by specific user attribute. A theme definition includes:

- Homepage header, footer (optional), and target page header (optional).

  Note: These header and footer definitions apply to classic homepages and classic pages only.

- Theme macro set (optional).
- Default theme style sheet for classic components (optional) if necessary to override the style sheet of the theme family.
- List of additional skins for classic components (optional).
- Mapped theme style sheets for fluid components by node (optional).
- Global theme style sheet for fluid components if necessary to override the style sheet of the theme family (optional).

At the system level, branding settings include:

- Selection of the default theme family (which includes the default theme and compatible default style sheet for classic components)
- Selection of a navigation collection for custom homepage tabs (optional).
- Additional style sheets and JavaScript objects for all classic components (optional).
- A default style sheet for all fluid components (optional).
- The branding framework application package and class.

At the portal registry level, additional settings can be specified, some of which override the system settings:

- Default theme.
- Default skin style sheet (optional).
- Default custom tabs collection (optional).
- Default custom homepage tab (optional).
- Default fluid homepage (optional).

At the user attribute level, additional settings can be specified, some of which override the system settings and the portal registry settings:

- Theme.
- Skin style sheet (optional).
- Custom tabs collection (optional).
- Default custom homepage tab (optional).
- Default fluid homepage (optional).

In addition to these system, portal, and user attribute settings, different branding can be applied to individual classic components. In Application Designer, the component's developer can specify style sheets and JavaScript objects as component properties. Through the browser, the portal administrator can specify additional custom style sheets and JavaScript objects as component properties. (These capabilities are in addition to existing features in Application Designer that allow the component's developer to apply a style sheet to each page definition, and to apply style classes to fields and field labels.)

Branding Definitions

PeopleTools allows you to maintain the following definitions used with branding:

- Branding system options for system-wide settings.
- Branding objects: HTML, JavaScript, style sheets, and images. (Definitions for these branding object types can also be created and maintained in Application Designer.)
- Pagelet branding, which allows you to modify display attributes of individual pagelets.

- Component branding, which allows you to add custom style sheets and JavaScript objects to individual classic components. These custom objects are in addition to any objects specified by the component's developer in Application Designer.
- Macro sets and macros that define the values for customizable branding elements such as colors, images, and so on.
- Header and footer definitions for classic pages, which are incorporated in theme definitions.
- Theme definitions, which include headers, a footer, and theme style sheets for classic and fluid components.
- Theme assignments, which allow you to specify themes and other options at the portal level and by user attribute.
- Branding system element types, which allow you to define the elements that can be used to construct header and footer definitions for classic pages.
- Branding user attribute types, which allow you to define the user attributes that can be used for theme assignments.
- Navigation collections designed for use as custom homepage tabs.

Style Sheets and JavaScript Objects

Levels at Which Style Sheets Can Be Applied

Style sheets are associated with different definitions throughout the system. Certain definitions require a style sheet, whereas one or more style sheets are optional for others. Moreover, different style sheet types are allowed with different definitions.

The following table summarizes the locations where style sheets can be attached for *classic components*:

Level	***#***	***Comments***
System default	1	Standard or free form. Practically speaking, the system default style sheet should be a standard style sheet that is compatible with the selected theme family. The system style sheet should be comprehensive and define both layout and style aspects of all user interface elements. Typically, you would select one of the three delivered layout style sheets: • PSSTYLEDEF • PSSTYLEDEF_SWAN • PSSTYLEDEF_TANGERINE
Additional system style sheets	0 – *n*	Standard or free form.

Level	**#**	**Comments**
Component	0 – *n*	Standard or free form. Component-level style sheets can be added by the component's developer, by the portal administrator, or both.
Page	0 or 1	Standard or free form. Page-level style sheets can be specified by the component's developer only.
Theme	0 or 1	Free form only.
Skin	0 – *n*	Free form only. While multiple skins can be associated with a theme definition, only one skin (or no skins) can be in effect per assignment (portal or user attribute).

The following table summarizes the locations where style sheets can be attached for *fluid components*:

Level	**#**	**Comments**
System default*	0 or 1	Standard or free form. If no style sheet is defined at the system level, PSSTYLEDEF_FMODE is used by default.
Component* or page*	0 – *n*	Standard or free form. Component-level and page-level style sheets can be added by the component's developer through PeopleCode.
Theme (global override)	0 or 1	Standard or free form.

* Theme definitions include the optional capability to associate mapped theme style sheets to base style sheets on a per node basis. You can create 0 – *n* mappings. If a base style sheet with a mapped theme style sheet is specified, then the mapped theme style sheet will be loaded directly beneath the base style sheet where the base style sheet is used.

Loading Sequence

At run time, the generated HTML includes style sheets and JavaScript objects in a specified sequence. Style sheets (or more specifically, the specific style classes defined in those style sheets) override any style class definitions of the same name loaded earlier in the sequence. Therefore, you can use theme and skin style sheets to overlay your branding look on top of styles defined at the system or component level.

Classic Components

For classic components, style sheets and JavaScript objects are loaded in the following order:

1. PSSTYLEDEF_REQ (automatically loaded by default for all classic components).*
2. System default style sheet.*
3. Additional system-level style sheets in the order defined.
4. Additional system-level JavaScript objects in the order defined.
5. Component-level style sheets in the component developer list in the order defined.
6. Component-level style sheets in the portal administrator list in the order defined.
7. Component-level JavaScript objects in the developer list in the order defined.
8. Component-level JavaScript objects in the portal administrator list in the order defined.
9. Page-level style sheet.
10. Theme style sheet.
11. Skin style sheet.

* A required style sheet that is always loaded. All other style sheets are optional.

Fluid Components

For fluid components, style sheets and JavaScript objects are loaded in the following order:

1. System default style sheet* (and its mapped theme style sheet if specified).
2. Component-level or page-level style sheets (and their mapped theme style sheets if specified) added by the component developer through PeopleCode.
3. Component-level JavaScript objects added by the component developer through PeopleCode.
4. Theme-level global override style sheet.

* A required style sheet that is always loaded. All other style sheets are optional.

Related Links

“AddStyleSheet” (PeopleCode Language Reference)
“AddJavaScript” (PeopleCode Language Reference)

Roles Required for PeopleTools Branding

Branding tasks require certain PeopleTools roles. The following list indicates which role provides access to which branding pages or additional branding features:

- Portal Administrator – Provides access to all of the menu items (pages) under the **PeopleTools** > **Portal** > **Branding** menu folder.

- Portal Manager – Provides access to all of the menu items (pages) under the**PeopleTools** > **Portal** > **Branding** menu folder except for those under the Theme Macro Set menu item.
- Secure Branding Administrator – Provides access to the Branding Objects pages and the Define Headers and Footers page. In order to use these pages, this role is required in addition to either Portal Administrator or Portal Manager.
- Company Info Administrator – Provides access to add and configure the CompanyInfo element, which can be added to a header definition on the Define Headers and Footers page. In order to add and configure the CompanyInfo element, this role is required in addition to Secure Branding Administrator and either Portal Administrator or Portal Manager. See Configuring a Custom Banner for more information.
- PeopleTools SVG Administrator – Grants permission to upload SVG images via the rich text editor.

Branding Process Overview

Use the following process when developing and applying your own branding:

1. Determine your branding requirements in collaboration with your marketing organization and your portal branding administrator.
2. Create new objects (images, style sheets, HTML objects, and JavaScript) as required to fulfill the branding requirements.

   See Understanding Branding Objects for more information.
3. (Optional) Create any new branding element type definitions to support your requirements.

   See for Creating a New Branding Element more information.
4. (Optional) Create and customize a macro set.

   See Defining Macro Sets and Macros.
5. Create new header and footer definitions (for classic homepages and classic pages only) combining delivered and new branding elements. Typically, it is easier to clone an existing definition and modify it to suit your needs.

   See Defining Headers and Footers for more information.
6. Create a branding theme definition using your new macro, header, footer, and theme style sheet definitions.

   See Assembling Branding Themes for more information.
7. (Optional) Generate mapped theme style sheets for fluid components.

   See Generating Theme Style Sheets for Fluid Components for more information.
8. (Optional) Create a navigation collection for custom homepage tabs.

   See Creating and Maintaining Navigation Collections for Custom Homepage Tabs for more information.

9. Specify branding system-wide settings.

   See Configuring Branding System Options for more information.

10. (Optional) Create any new user attribute type definitions to support your requirements.

    See Creating a New User Attribute Type for more information.

11. (Optional) Make any portal- or user attribute-specific theme and branding assignments.

    See Assigning Branding Themes for more information.

12. (Optional) Apply any component-specific branding to classic components.

    See Branding Classic Components for more information.

13. (Optional) Apply an pagelet-specific branding to pagelets.

    See Maintaining Pagelet Branding Attributes for more information.

Creating System Definitions and Objects for Branding

This topic provides an overview of branding objects and discusses how to:

- Create and maintain HTML objects.
- Create and maintain JavaScript objects.
- Create and maintain style sheet objects.
- Add and edit branding objects.
- Upload and maintain images.
- Add and edit images.
- Define element types.
- Define user attribute types.

Understanding Branding Objects

PeopleTools branding allows you to create and maintain branding objects such as HTML, JavaScript, free form style sheets, and images without requiring access to Application Designer. These objects can then be used in places such as branding system definitions, headers or footers, custom layout objects for pagelets and classic homepage tabs, and in other definitions.

While you can create and maintain HTML, JavaScript, free form style sheet objects, and images in Application Designer, these Application Designer objects *are not* accessible to be viewed or maintained through portal branding. But, if you enter the same name as an existing Application Designer object and if you have edit permissions on the managed object, then you will be prompted as to whether you want to overwrite the existing object through portal branding. However, objects that you create through portal branding can be viewed and updated in Application Designer.

Note: You can use Application Designer to migrate these portal branding objects from a test environment to a production environment. However, these objects will be treated as regular managed objects in the new system, and will not be accessible from these portal branding pages in the new system.

Related Links

"Understanding Projects" (Application Designer Developer's Guide)

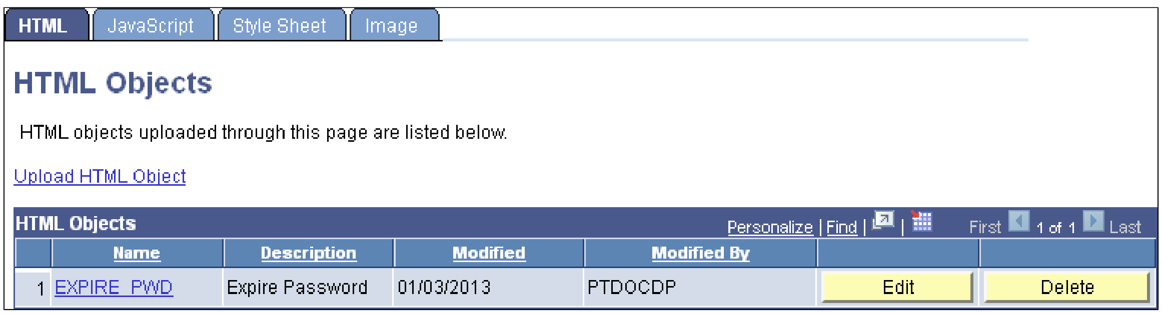
Creating and Maintaining HTML Objects

Use the HTML Objects page to upload and maintain HTML objects used for branding.

Navigation:

PeopleTools > **Portal** > **Branding** > **Branding Objects**

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



Field or Control	***Description***
Upload HTML Object	Click the link to create a new HTML object. See Adding and Editing Branding Objects.
Name	Click the object name link for an existing HTML object to view the object details including the HTML code.
Description	Displays the object's description.
Edit	Click the Edit button to modify the description or stored HTML code. See Adding and Editing Branding Objects.
Delete	Click the Delete button to delete the object from the database.

Note: The link to view the HTML object is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

HTML objects created in Application Designer cannot be viewed or edited in portal branding. However, HTML objects created in portal branding can be viewed and modified in Application Designer.

Related Links

"Understanding HTML Definitions" (Application Designer Developer's Guide)

Creating and Maintaining JavaScript Objects

Use the JavaScript objects page to upload and maintain JavaScript objects used for branding.

Navigation:

1. **PeopleTools** > **Portal** > **Branding** > **Branding Objects**.
2. Select the JavaScript tab.

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

Field or Control	***Description***
Upload JavaScript Object	Click the link to create a new JavaScript object. **Note:** JavaScript is stored as an HTML object in the database. See Adding and Editing Branding Objects.
Name	Click the object name link for an existing JavaScript object to view the object details including the JavaScript code.
Description	Displays the object's description.
Edit	Click the Edit button to modify the description or stored JavaScript code. See Adding and Editing Branding Objects.
Delete	Click the Delete button to delete the object from the database.

Note: The link to view the JavaScript object is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

JavaScript objects created in Application Designer cannot be viewed or edited in portal branding. However, JavaScript objects created in portal branding (and stored as HTML objects) can be viewed and modified in Application Designer.

Related Links

"Understanding HTML Definitions" (Application Designer Developer's Guide)

"Working with JavaScript in HTML Definitions" (Application Designer Developer's Guide)

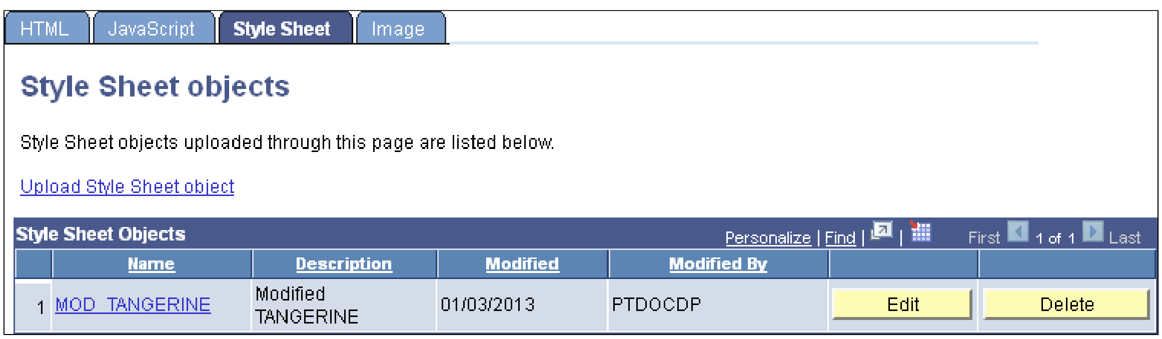
Creating and Maintaining Style Sheet Objects

Use the Style Sheet objects page to upload and maintain free form style sheet objects used for branding.

Navigation:

1. **PeopleTools** > **Portal** > **Branding** > **Branding Objects**.
2. Select the Style Sheet tab.

This example illustrates the fields and controls on the Style Sheet objects page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Upload Style Sheet Object	Click the link to create a new free form style sheet object. See Adding and Editing Branding Objects.
Name	Click the object name link for an existing free form style sheet object to view the object details including style definitions.
Description	Displays the object's description.
Edit	Click the Edit button to modify the description or stored style definitions. See Adding and Editing Branding Objects.

Field or Control	Description
Delete	Click the Delete button to delete the object from the database.

Note: The link to view the style sheet is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

Style sheet objects created in Application Designer cannot be viewed or edited in portal branding. However, free form style sheet objects created in portal branding can be viewed and modified in Application Designer.

Related Links

“Style Sheet Types” (Application Designer Developer’s Guide)

“Working With Free Form Sub Style Sheets” (Application Designer Developer’s Guide)

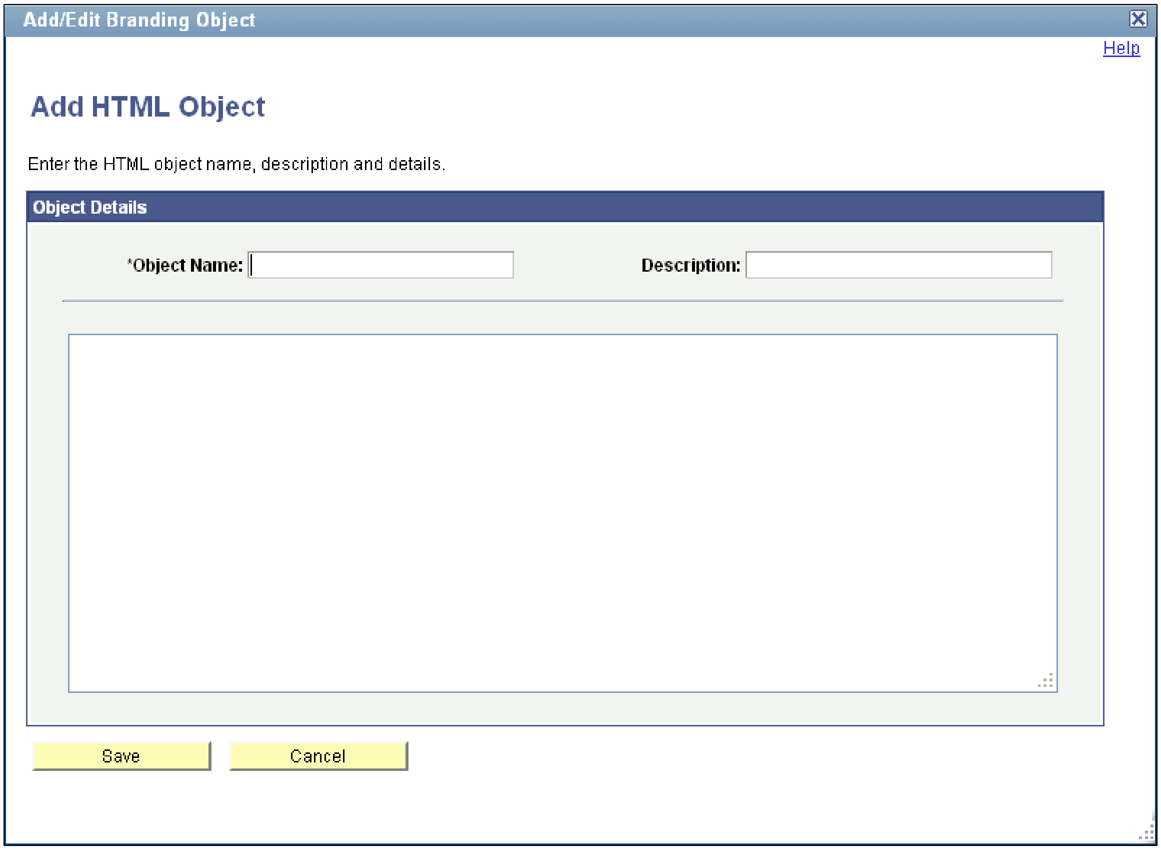
Adding and Editing Branding Objects

Use the Add/Edit Branding Object page to add or edit a branding object.

Navigation:

On the HTML Objects page, JavaScript objects page, or Style Sheet objects page, either click the Upload Object link or click the Edit button for an existing object.

This example illustrates the fields and controls on the Add/Edit Branding Object page (for an HTML object). You can find definitions for the fields and controls later on this page.



Field or Control	Description
Object Name	Enter an ID for the new object. **Important!** If you specify the name of an existing object of this type whether it was created in portal branding or through Application Designer, you will be prompted to overwrite the existing object after you click the Save button. However, if you have no permissions or read-only permissions on an Application Designer object, a duplicate ID error message will be displayed, and you will need to enter a different object ID. ***Warning!*** Just as you must not update PeopleTools-delivered objects in Application Designer, do not use portal branding to update these PeopleTools-delivered objects.
Description	Enter a descriptive name for the object.
Unlabeled text area	Enter the HTML, JavaScript, or free form style sheet code.

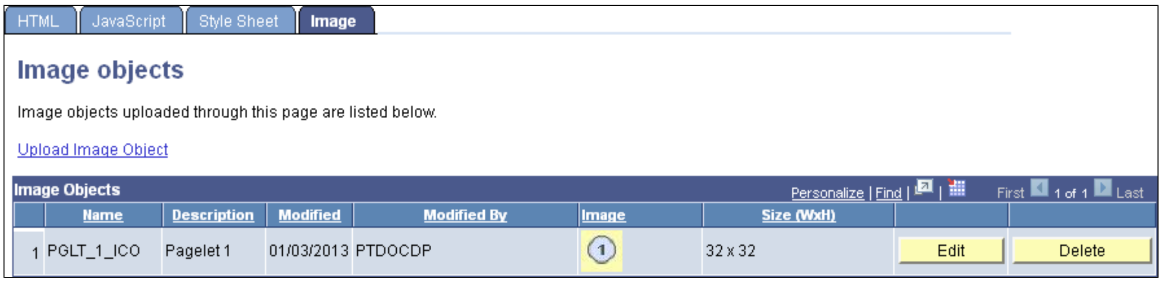
Uploading and Maintaining Images

Use the Image objects page to upload and maintain image objects used for branding.

Navigation:

1. **PeopleTools** > **Portal** > **Branding** > **Branding Objects**.
2. Select the Image tab.

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



Field or Control	***Description***
Upload Image Object	Click the link to upload a new image file. See Adding and Editing Images.
Name	Displays the image object's name.
Description	Displays the object's description.
Edit	Click the Edit button to modify the image including uploading a new image file. See Adding and Editing Images.
Delete	Click the Delete button to delete the object from the database.

Note: The link to view the image is displayed only if the user has read permissions on the object. The Edit and Delete buttons are displayed only if the user has read/write permissions on the object.

Image objects created in Application Designer cannot be viewed or edited in portal branding. However, image objects created in portal branding can be viewed and modified in Application Designer.

Related Links

"Understanding Image Definitions" (Application Designer Developer's Guide)

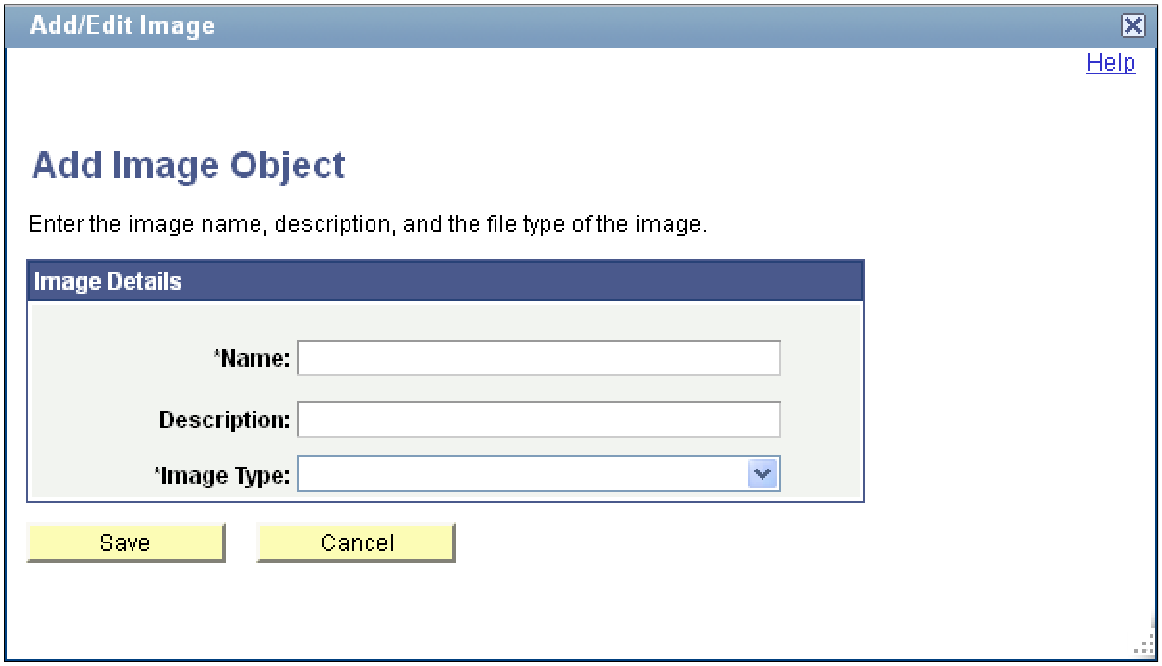
Adding and Editing Images

Use the Add/Edit Image page to add or edit an image.

Navigation:

On the Image objects page, either click the Upload Image Object link or click the Edit button for an existing image. Alternatively, on the Pagelet Branding page, click the **Upload new image** link.

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



Field or Control	*Description*
Name	Enter an ID for the new image object. **Important!** If you specify the name of an existing image object whether it was created in portal branding or through Application Designer, you will be prompted to overwrite the existing object after you click the Save button.
Description	Enter a descriptive name for the object.

Field or Control	***Description***
Image Type	**Important!** Some database platforms, such as Db2 for z/OS, limit the file size to 32 KB. Select one of the following image formats to access the File Attachment dialog box to browse and select the file to upload: • BMP Files [*.bmp] • DIB Files [*.dib] • GIF Files [*.gif] • JPG Files [*.jpg] • PNG Files [*.png] • SVG [*.svg] **Note:** The File Attachment dialog box does not filter your browsing by the selected file type. However, if you attempt to upload a file that does not match the specified format, then an error message will be generated and the upload will not occur.

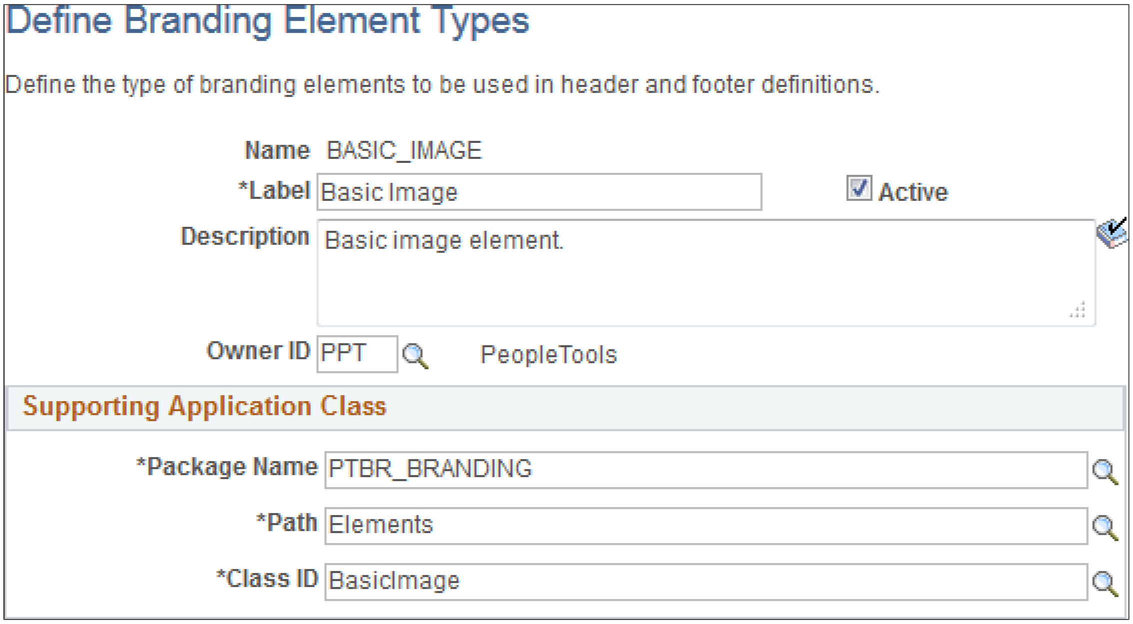
Defining Branding Element Types

Use the Define Branding Element Types page to create and maintain the types of branding elements to be used in header and footer definitions. Each branding element type is mapped to a supporting PeopleCode application class definition.

Navigation:

PeopleTools > **Portal** > **Branding** > **System Data** > **Define Branding Element Types**

This example illustrates the fields and controls on the Define Branding Element Types page.



Field or Control	***Description***
Name	Displays the unique identifier for this element type.
Label	Enter a short name for the element type, which will be displayed when an element of this type is selected to be used in a header or footer definition.
Active	This check box is selected by default. Deselect this check box to make the element type inactive. No new elements of inactive types can be added to header or footer definitions. Moreover, existing elements of inactive element types in existing header or footer definitions are ignored.
Description	Enter a description for the element type.
Owner ID	Select which product, feature, or application owns this element type definition.
Package Name	Select the name of the application package that contains the application class that defines this element type.
Path	Select the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Field or Control	*Description*
Class ID	Select the name of the application class that defines this element type.
Add to Project	Click the Add to Project link to access the Add To Project page to insert this branding element type definition into an ADS project. See "Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide) and Migrating Branding Data Using ADSs for more information.

Related Links

Creating a New Branding Element

Defining Headers and Footers

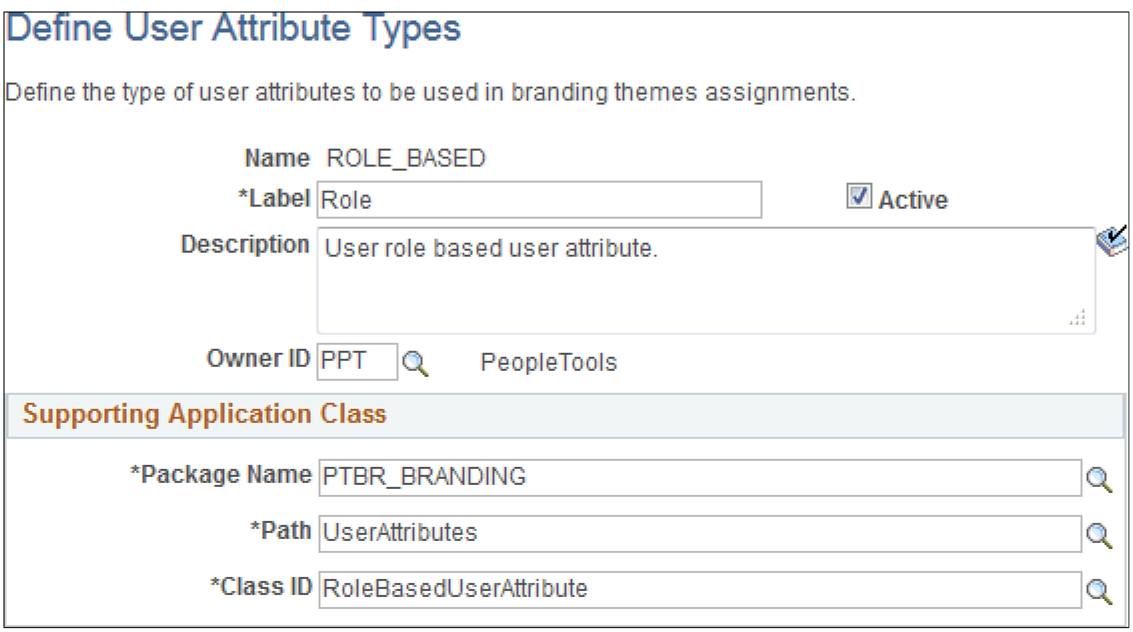
Defining User Attribute Types

Use the Define User Attribute Types page to create and maintain the types of user attributes to be used in branding theme assignments. Each user attribute type is mapped to a supporting PeopleCode application class definition.

Navigation:

PeopleTools > **Portal** > **Branding** > **System Data** > **Define User Attribute Types**

This example illustrates the fields and controls on the Define User Attribute Types page.



Field or Control	*Description*
Name	Displays the unique identifier for this user attribute type.
Label	Enter a short name for the user attribute type, which will be displayed when this type is selected to be used in a theme assignment.
Active	This check box is selected by default. Deselect this check box to make the user attribute type inactive. An inactive user attribute type cannot be selected for use in a branding theme assignment. Moreover, existing theme assignments with inactive user attribute types are ignored.
Description	Enter a description for the user attribute type.
Owner ID	Select which product, feature, or application owns this user attribute type definition.
Package Name	Select the name of the application package that contains the application class that defines this user attribute type.
Path	Select the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class ID	Select the name of the application class that defines this user attribute type.
Add to Project	Click the Add to Project link to access the Add To Project page to insert this branding element type definition into an ADS project. See "Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide) and Migrating Branding Data Using ADSs for more information.

Related Links

Creating a New User Attribute Type

Assigning Branding Themes

Administering System Branding

This topic discusses how to administer branding of your system.

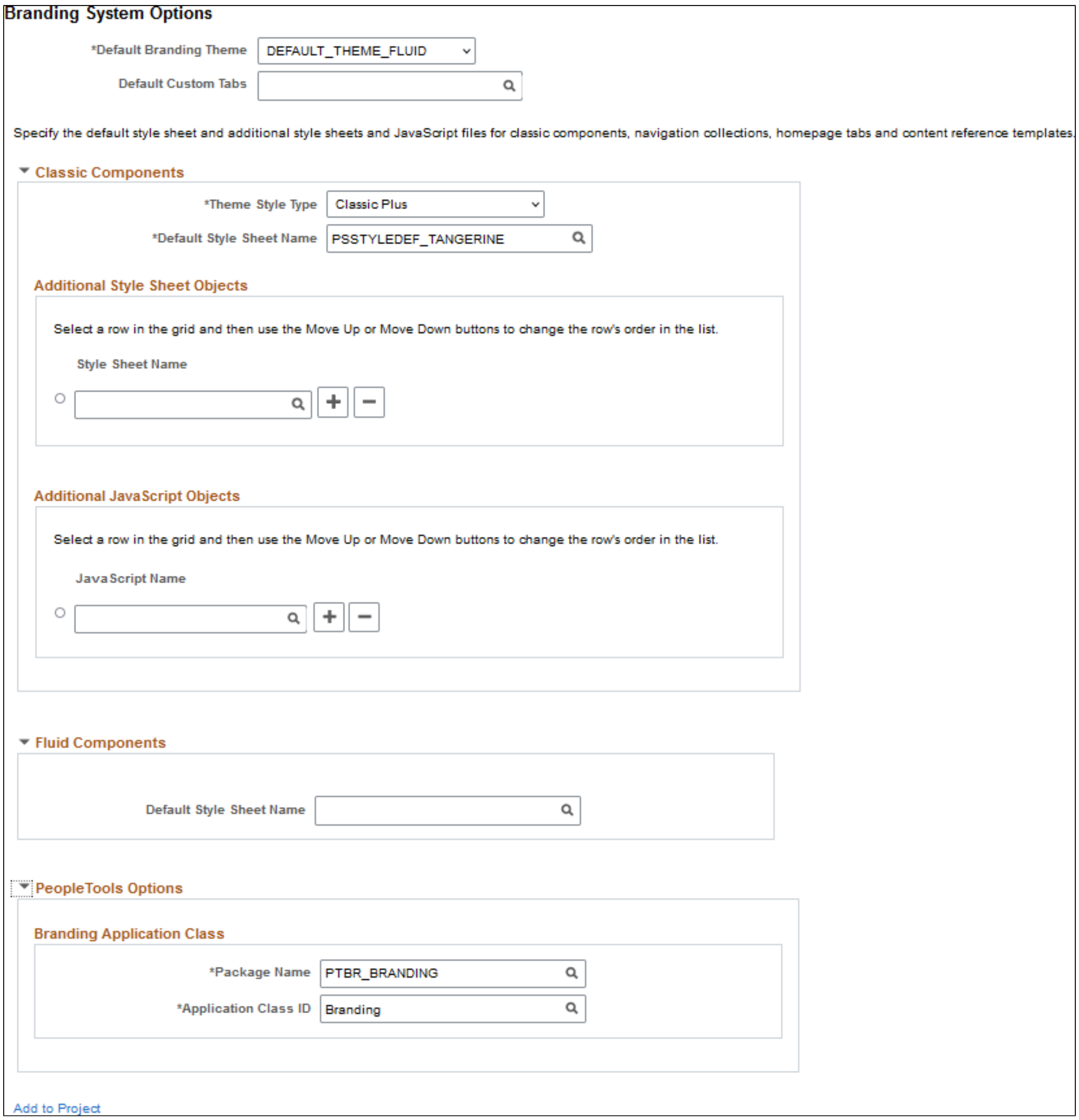
Configuring Branding System Options

Use the Branding System Options page to specify system-wide branding settings including the branding theme type (family), the branding theme, the default style sheet for classic components, and the default style sheet for fluid components.

Navigation:

PeopleTools > **Portal** > **Branding** > **Branding System Options**

This example illustrates the fields and controls on the Branding System Options page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Default Branding Theme	Select a default, system-wide branding theme. **Important!** In a cluster environment, ensure that all systems use the same setting for default branding theme. ***Warning!*** The branding theme selected must be compatible with the theme family selected in the Theme Style Type field. Otherwise, unexpected and undesirable results can occur.
Default Custom Tabs	Select a default, system-wide navigation collection that will be used to render custom homepage tabs. **Note:** Custom tabs are visible from classic homepages only.

Compatible Branding Themes

Use the following table to determine which branding theme style type (family) is compatible with a specified theme.

Branding Theme	*Description*	*Compatible Theme Style Types*	*Classic Navigation*	*Fluid Navigation*
DEFAULT_THEME_FLUID	The default branding theme for PeopleTools 8.55 (and above) that uses the fluid header on all classic and fluid pages.	• Classic Plus • Tangerine	NavBar: Menu	NavBar: Menu
PS_REVERT_TO_PRE858 **Note:** Do *not* use this theme.	For PeopleTools 8.58, an alternative theme that reverts to the original fluid and classic plus styling delivered prior to PeopleTools 8.58. **Note:** Icons and images such as those found in the fluid banner and classic portal header are not reverted.	Classic Plus Pre 858	NavBar: Menu	NavBar: Menu
DEFAULT_THEME_ TANGERINE*	The Tangerine theme for PeopleSoft 9.2 applications on PeopleTools 8.53 and PeopleTools 8.54.	Tangerine	Drop-down menu*	NavBar: Menu

Branding Theme	***Description***	***Compatible Theme Style Types***	***Classic Navigation***	***Fluid Navigation***
DEFAULT_THEME_TANGERINE_ALT*	For PeopleTools 8.54, an alternate Tangerine theme that includes a portal header that is similar to the look of PeopleSoft fluid pages.	Tangerine	• NavBar: Menu • Drop-down menu*	NavBar: Menu
DEFAULT_THEME_SWAN*	The default theme for PeopleSoft 9.1 applications on PeopleTools 8.50 to 8.52.	Swan	Drop-down menu*	NavBar: Menu
DEFAULT_THEME_CLASSIC*	The default theme for PeopleSoft 9.0 applications (and below) on PeopleTools releases prior to 8.50.	Classic	Drop-down menu*	NavBar: Menu

Important! * Classic branding themes (DEFAULT_THEME_TANGERINE, DEFAULT_THEME_TANGERINE_ALT, DEFAULT_THEME_SWAN, and DEFAULT_THEME_CLASSIC), their corresponding portal header definitions, the drop-down menu, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner and the fluid-like portal header, both of which contain the NavBar and Menu tile for navigating from both classic and fluid pages. For information on working with the fluid user interface, see "Working with Fluid Pages and Controls" (Applications User's Guide).
In addition, the Swan and Classic theme style types are not supported.

Classic Components

Field or Control	*Description*
Theme Style Type	**Important!** Set the Theme Style Type field first. When the value of this field is changed, a compatible value is automatically specified for the Default Style Sheet Name field; and while other compatible options may be available, the Default Branding Theme field is automatically updated to DEFAULT_THEME_FLUID. **Important!** In a cluster environment, ensure that all systems use the same setting for the theme style type—that is, all systems must enable the same classic plus setting or disable classic plus using the Tangerine setting. Use this field to specify the theme style type (family): • Classic Plus – On PeopleTools 8.56 and above, select this theme style type to allow configured classic components to be displayed with fluid-like styling (which is also referred to as “classic plus”). For more information, see PeopleTools Components Delivered as Classic Plus. • Tangerine – Select this theme style type to disable classic plus. This theme style type presents the classic styling that was originally available for PeopleSoft 9.2 applications on PeopleTools 8.53 and above. • Classic Plus Pre 858 – Do not select this theme style type; it is not supported. On PeopleTools 8.58, this theme was used to revert to the original classic plus styling available prior to PeopleTools 8.58. • Swan – Do not select this theme style type; it is not supported. This setting is retained for backward compatibility only with styling similar to PeopleSoft 9.1 applications on PeopleTools 8.50 to 8.52. • Classic – Do not select this theme style type; it is not supported. This setting is retained for backward compatibility only with styling similar to PeopleSoft 9.0 applications (and below) on PeopleTools releases prior to 8.50.

Field or Control	***Description***
Default Style Sheet Name	Select a default, system-wide style sheet for classic components. The value selected here will also be visible on the PeopleTools Options page. Moreover, if the previous value of the default style sheet was also used on the System Options page or the Registry Options page for one or more portal registries, the value selected here is updated to each of those pages. ***Warning!*** The style sheet selected should be compatible with the theme family selected in the Theme Style Type field. Otherwise, unexpected and undesirable results can occur. Therefore, specify the Theme Style Type field first; then retain the automatically selected value of this field or select another style sheet that is compatible within the selected theme family.
Style Sheet Name (Additional Style Sheet Objects)	Specify zero, one, or more than one style sheet to be loaded with all classic components.
JavaScript Name(Additional JavaScript Objects)	Specify zero, one, or more than one JavaScript object to be loaded with all classic components. **Important!** JavaScript objects are stored as HTML objects in the database. The lookup does not differentiate between standard HTML objects and JavaScript objects. Therefore, it is your responsibility to ensure that you have selected valid JavaScript to be loaded with a component.
Move UpMove Down	Select the row designating a style sheet or a JavaScript object, and then use the Move Up and Move Down buttons to change the order in which the objects are loaded with all classic components.

Fluid Components

Field or Control	*Description*
Default Style Sheet Name	Select a default, system-wide style sheet for fluid components. **Note:** If no style sheet is specified, then PSSTYLEDEF_FMODE is used as the default style sheet for all fluid components. The value selected here will also be visible on the PeopleTools Options page.

PeopleTools Options

The branding application class generates the portal headers, footers, and other aspects of the branding theme. The default branding application class is PTBR_BRANDING:Branding. If this delivered application class does not fully support your requirements, you can create your own by extending the delivered application class.

Field or Control	*Description*
Package Name	Select the name of the application package that contains the application class that defines system-wide branding functions. The default value is: PTBR_BRANDING. The value selected here will also be visible on the PeopleTools Options page.
Application Class ID	Select the name of the application class that defines system-wide branding functions. The default value is: Branding. The value selected here will also be visible on the PeopleTools Options page.

Add to Project

Click the Add to Project link to access the Add To Project page to insert the branding system options definition into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) and Migrating Branding Data Using ADSs for more information.

Related Links

Assembling Branding Themes

Creating and Maintaining Navigation Collections for Custom Homepage Tabs

“PeopleTools Options” (System and Server Administration)

Defining Navigation Collection System Options

Defining Navigation Collection Registry Options

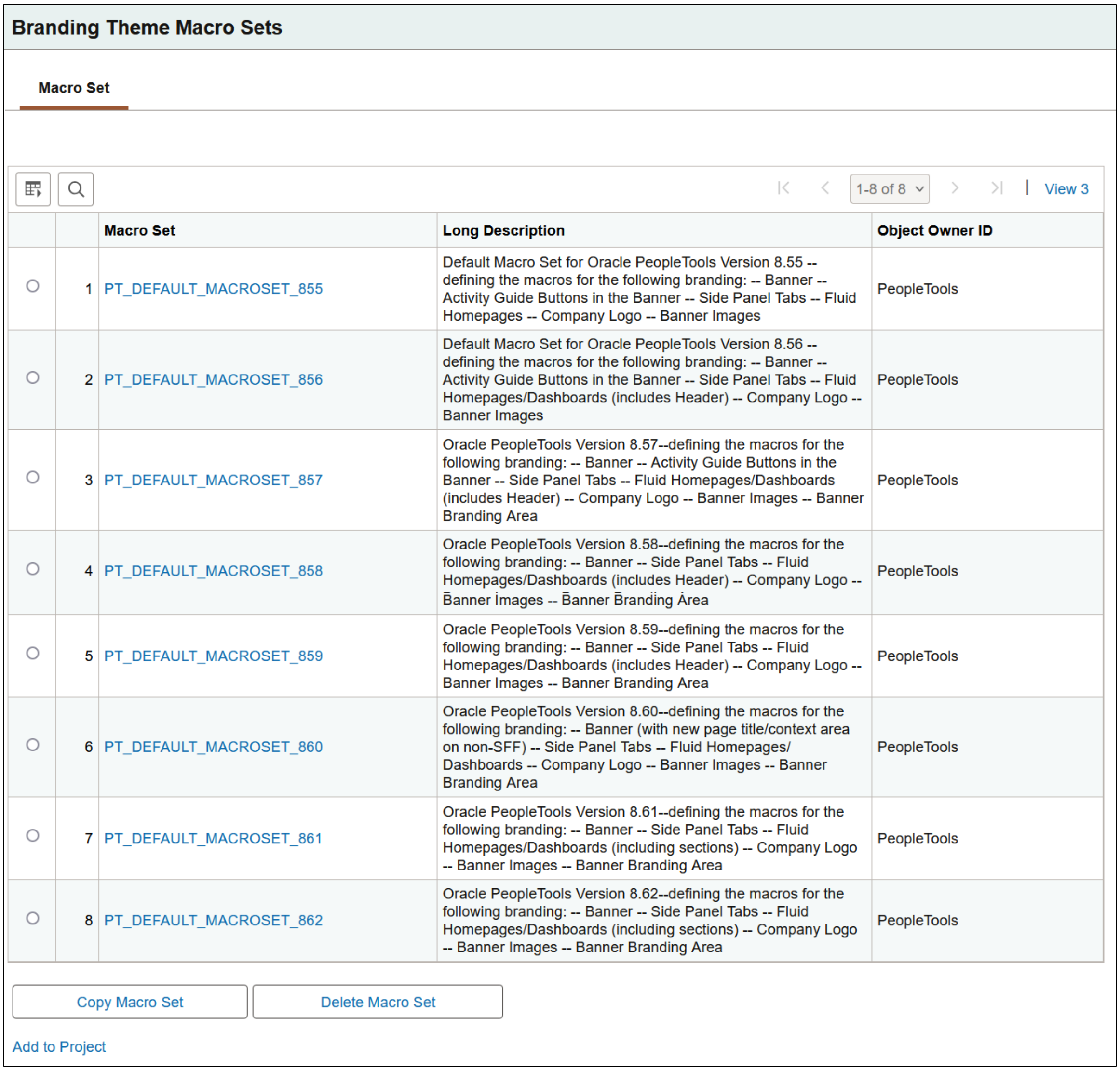
Defining Macro Sets and Macros

Use the Macro Set page to copy and manage macro sets.

Navigation:

PeopleTools > **Portal** > **Branding** > **Branding Theme Macro Set**

This example illustrates the fields and controls on the Macro Set page.



Branding Theme Macro Sets

Macro Set

1-8 of 8 View 3

		Macro Set	Long Description	Object Owner ID
○	1	PT_DEFAULT_MACROSET_855	Default Macro Set for Oracle PeopleTools Version 8.55 -- defining the macros for the following branding: -- Banner -- Activity Guide Buttons in the Banner -- Side Panel Tabs -- Fluid Homepages -- Company Logo -- Banner Images	PeopleTools
○	2	PT_DEFAULT_MACROSET_856	Default Macro Set for Oracle PeopleTools Version 8.56 -- defining the macros for the following branding: -- Banner -- Activity Guide Buttons in the Banner -- Side Panel Tabs -- Fluid Homepages/Dashboards (includes Header) -- Company Logo -- Banner Images	PeopleTools
○	3	PT_DEFAULT_MACROSET_857	Oracle PeopleTools Version 8.57--defining the macros for the following branding: -- Banner -- Activity Guide Buttons in the Banner -- Side Panel Tabs -- Fluid Homepages/Dashboards (includes Header) -- Company Logo -- Banner Images -- Banner Branding Area	PeopleTools
○	4	PT_DEFAULT_MACROSET_858	Oracle PeopleTools Version 8.58--defining the macros for the following branding: -- Banner -- Side Panel Tabs -- Fluid Homepages/Dashboards (includes Header) -- Company Logo -- Banner Images -- Banner Branding Area	PeopleTools
○	5	PT_DEFAULT_MACROSET_859	Oracle PeopleTools Version 8.59--defining the macros for the following branding: -- Banner -- Side Panel Tabs -- Fluid Homepages/Dashboards (includes Header) -- Company Logo -- Banner Images -- Banner Branding Area	PeopleTools
○	6	PT_DEFAULT_MACROSET_860	Oracle PeopleTools Version 8.60--defining the macros for the following branding: -- Banner (with new page title/context area on non-SFF) -- Side Panel Tabs -- Fluid Homepages/ Dashboards -- Company Logo -- Banner Images -- Banner Branding Area	PeopleTools
○	7	PT_DEFAULT_MACROSET_861	Oracle PeopleTools Version 8.61--defining the macros for the following branding: -- Banner -- Side Panel Tabs -- Fluid Homepages/Dashboards (including sections) -- Company Logo -- Banner Images -- Banner Branding Area	PeopleTools
○	8	PT_DEFAULT_MACROSET_862	Oracle PeopleTools Version 8.62--defining the macros for the following branding: -- Banner -- Side Panel Tabs -- Fluid Homepages/Dashboards (including sections) -- Company Logo -- Banner Images -- Banner Branding Area	PeopleTools

Copy Macro Set Delete Macro Set

Add to Project

To create a new macro set that you can customize, select and copy a sample macro set—for example, PT_DEFAULT_MACROSET_858.

Warning! Do not modify or delete any PeopleTools-delivered macro sets.

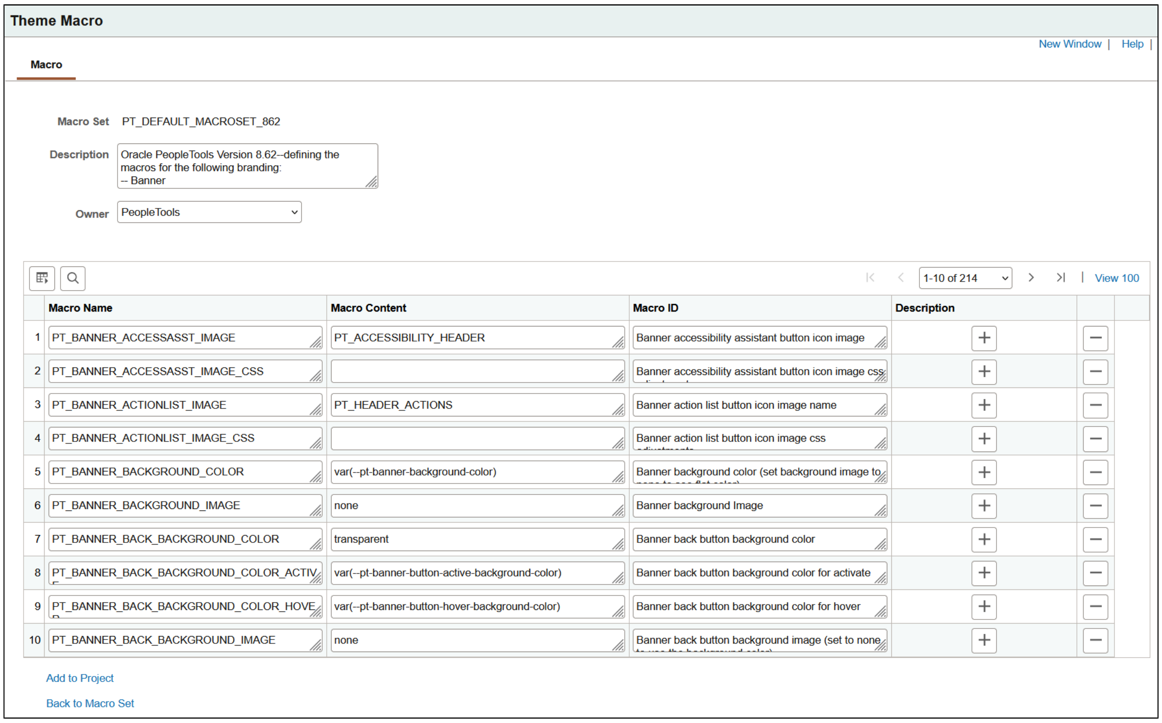
Defining Macros

Use the Macro page to specify custom values for delivered macros.

Navigation:

On the Macro Set page, click the link for a macro set.

This example illustrates the fields and controls on the Macro page.



Some macros are used in the fluid banner only (displayed on fluid pages only) while other macros are used in both the fluid banner and in the header for classic pages. Therefore, when you make an update to a macro that is shared both by the fluid banner and by the header for classic pages, you will make updates to the look of all of your pages simultaneously. Prior to updating any macros, you can use the download to Excel button on the grid to export the grid contents to an Excel spreadsheet. You can examine and analyze the macros to determine which ones to modify.

Note: Because activity guide buttons were moved from the fluid banner into an activity guide sub-banner in PeopleTools 8.57, you can no longer brand these buttons. Ignore macros beginning with PT_BANNER_AG. If you previously branded these items, your customized macros will have no effect.

To customize the value for a delivered macro, change the value of the Macro Content field only.

Add to Project

Click the Add to Project link to access the Add Project page to insert one or more macro set definitions into an ADS project. See "Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide) and <u>Migrating Branding Data Using ADSs</u> for more information.

Configuring the System to Use a Custom Macro Set

To configure the system to use a custom macro set, you must assemble and apply a custom branding theme. Moreover, your custom branding theme must incorporate:

- A custom theme style sheet for classic components that has been cloned from PT_BRAND_CLASSIC_TEMPLATE_862.

- A custom override style sheet for fluid components that has been cloned from either global override style sheet: PT_BRAND_FLUID_NOIMG_TEMPL_862 or PT_BRAND_FLUID_TEMPLATE_862.
- (Optional) A custom header definition cloned from DEFAULT_HEADER_FLUID.

To enable your custom macro set:

1. In Application Designer, open the PT_BRAND_CLASSIC_TEMPLATE_862 theme style sheet for classic components.
2. Save the classic theme style sheet as a cloned copy with a new name—for example, MY_BRAND_CLASSIC_TEMPLATE_862.
3. In Application Designer, open one of the global override style sheets:
    - If you do not need to brand images, select PT_BRAND_FLUID_NOIMG_TEMPL_862 because it is smaller, and therefore consumes less overhead.
    - If you do need to brand images, select PT_BRAND_FLUID_TEMPLATE_862.
4. Save the global override style sheet as a cloned copy with a new name—for example, MY_BRAND_FLUID_NOIMG_TEMPL_862 (or MY_BRAND_FLUID_TEMPLATE_862).
5. In the browser, go to the Theme Macro Set page:
    a. Select the PT_DEFAULT_MACROSET_862 macro set.
    b. Copy and save the macro set with a new name—for example, MY_MACROSET_862.
6. (Optional) In the browser, go to the Define Headers and Footers page:
    a. Open the DEFAULT_HEADER_FLUID definition.
    b. Save the header as a cloned copy with a new name—for example, MY_HEADER_FLUID.
7. In the browser, go to the Assemble Themes page:
    a. Open the DEFAULT_THEME_FLUID theme definition.
    b. Save the branding theme as a cloned copy with a new name—for example, MY_THEME_FLUID.
8. Update this custom branding theme to include the following:
    - (Optional) Homepage Header: Your custom header definition.
    - Macro Set: Your custom macro set.
    - Classic Components – Theme Style Sheet: Your custom style sheet for classic components.
    - Fluid Components – Global Override Style Sheet: Your custom style sheet for fluid components.

This example illustrates the fields and controls on the Assemble Themes page for a custom branding theme.

9. Apply your custom branding theme either system-wide on the Branding System Options page or selectively by portal or by user attribute on the Assign Branding Themes page.

Note: PeopleSoft Interaction Hub provides a theme builder feature that simplifies the customization of macros. See *PeopleSoft Interaction Hub: Branding*, "PeopleSoft Fluid User Interface for Branding in PeopleSoft Interaction Hub" for more information.

Updating Custom Macro Sets for the Current Release

If you have created a custom macro set, then you need to update that macro set with both the new macros introduced with each subsequent PeopleTools release as well as the existing macros that were updated in each subsequent PeopleTools release. Refer to each of the tables to determine the changes that you need to apply to your custom macro set. Alternatively, you could re-apply your branding to a cloned copy of the current release's PT_DEFAULT_MACROSET macro set as described in the preceding section, "Configuring the System to Use a Custom Macro Set."

Note the following regarding macro names and specifications:

- Any macro with a name similar to *_FULL_CSS requires a full CSS specification in the form of:

  ```
  CSSselector { cssProperty: cssValue; ...}
  ```

- Typically, any macro with a name similar to *_CSS (not *_FULL_CSS) requires a specification in the form of:

  ```
  CSSProperty: CSSValue;[CSSProperty:CSSValue;...]
  ```

New Macros in PeopleTools 8.57

The following macros were added in PeopleTools 8.57:

Macro Name	***Macro Content***	***Description***
PT_COMPINFO_ BACKGROUND	linear-gradient(rgb(23,54,81),rgb (57,106,144))	Company Information RTE background
PT_COMPINFO_ BOTTOMBORDER	1px solid rgb(81,113,150)	Company Information RTE bottom border
PT_COMPINFO_CSS[2]		Company Information RTE generic CSS
PT_COMPINFO_ HEIGHTTHRESHOLD	564px	Company Information window height threshold value where a height below this value applies new macros
PT_COMPINFO_ HEIGHTTHRESH_CSS[2]		Company Information RTE generic CSS for window height threshold

Macro Name	Macro Content	Description
PT_COMPINFO_ HEIGHTTHRESH_FULL_ CSS[1]		Company Information CSS for window height threshold (full CSS selector plus property/value)
PT_COMPINFO_ ORIENTLAND_CSS[2]		Company Information RTE generic CSS for landscape mode
PT_COMPINFO_ ORIENTLAND_FULL_CSS[1]		Company Information CSS for landscape mode (full CSS selector plus property/value)
PT_COMPINFO_ ORIENTLAND_SFF_CSS[2]		Company Information RTE generic CSS for landscape mode for small form factor only
PT_COMPINFO_SFF_CSS[2]		Company Information RTE generic CSS for small form factor
PT_COMPINFO_ TEXTCOLOR	#ffffff	Company Information RTE default font color

New Macros in PeopleTools 8.58

The following macros were added in PeopleTools 8.58:

Macro Name	Macro Content	Description
PT_BANNER_BACK_ IMAGE_NARROW_CSS	%AlignStart:4px;	Banner Back Button image adjustment css when text disappears from button (only image)
PT_BANNER_IMAGE_ BUTTON_RADIUS	1.2em	Banner image button border radius
PT_BANNER_SEARCH_ PRESSED_ACTIVE_CSS	background-color: rgba(0, 0,0,0.8);background-image: linear-gradient(to bottom, #000, #6c7784);	Banner button (search) depressed state general css on activate
PT_BANNER_SEARCH_ PRESSED_BOXSHADOW	inset 1px 1px 0 0 #5c6570, inset -1px 0 0 0 #5c6570, inset 0 2px 0 0 #eee	Banner button (search) depressed state box shadow

Macro Name	*Macro Content*	*Description*
PT_BANNER_SEARCH_ PRESSED_HOVER_CSS	background-color: rgba(255, 255,255,0.35);background-image: linear-gradient(to bottom, #505862, #6c7784);	Banner button (search) depressed state general css on hover
PT_BANNER_UPPER_ DECORATION	linear-gradient(to %AlignEnd, #b4ec51, #53a0fd)	Banner decorative border (actually a background)
PT_BANNER_UPPER_ DECORATION_HEIGHT	0.25em	Banner decorative border height
PT_HOMEPAGE_PAGEIND_ CONT_BACKGROUND	#8b9197	Homepage/Dashboard page indicator container background color
PT_HOMEPAGE_TILE_ BORDER_HOVER	#246FC7	Homepage/Dashboard Tile border color for hover
PT_HOMEPAGE_TILE_ BOXSHADOW_HOVER	0 8px 20px 0 rgba(29,39,51,0.10)	Homepage/Dashboard Title box shadow for hover
PT_HOMEPAGE_TILE_ TITLE_ALIGNMENT	center	Homepage/Dashboard Tile Title alignment (non-small form factor)
PT_HOMEPAGE_TILE_ TITLE_COLOR	#1e5ca4	Homepage/Dashboard Tile Title text color
PT_HOMEPAGE_TILE_ TITLE_CSS		Homepage/Dashboard Tile Title text general css adjustments
PT_NAVBAR_TOP_OFFSET	2.97em	NavBar Offset from top of window
PT_SEARCH_TRAY_ BORDER_COLOR	#5c6570	Banner Search Tray border color

Updated Macros in PeopleTools 8.58

The following macros were updated in PeopleTools 8.58:

Macro Name	*Macro Content*	*Description*
PT_BANNER_ BACKGROUND_COLOR	#1d2733	Banner background color (set background image to none to see flat color)
PT_BANNER_ BACKGROUND_IMAGE	none	Banner background Image
PT_BANNER_BACK_ BACKGROUND_COLOR_ ACTIVE	rgb(0,0,0,0.8)	Banner back button background color for activate
PT_BANNER_BACK_ BACKGROUND_COLOR_ HOVER	rgba(255,255,255,0.35)	Banner back button background color for hover
PT_BANNER_BACK_ BACKGROUND_IMAGE	none	Banner back button background image (set to none to use the background color)
PT_BANNER_BACK_ BACKGROUND_IMAGE_ ACTIVE	none	Banner back button background image for activate
PT_BANNER_BACK_ BACKGROUND_IMAGE_ HOVER	none	Banner back button background image for hover
PT_BANNER_BACK_ BORDER_BOTTOM_COLOR	#3d444e	Banner back button border color (bottom)
PT_BANNER_BACK_ BORDER_BOTTOM_COLOR _HOVER	#3d444e	Banner back button border color (bottom) for hover
PT_BANNER_BACK_ BORDER_COLOR	#3d444e	Banner back button border color
PT_BANNER_BACK_ BORDER_COLOR_HOVER	#3d444e	Banner back button border color for hover
PT_BANNER_BACK_ BOXSHADOW	none	Banner back button box shadow

Macro Name	***Macro Content***	***Description***
PT_BANNER_BACK_ BOXSHADOW_ACTIVE	none	Banner back button box shadow for activate
PT_BANNER_BACK_ BOXSHADOW_HOVER	none	Banner back button box shadow for hover
PT_BANNER_BORDER_ COLOR	#f0f0f0	Banner bottom border color
PT_BANNER_BUTTON_ BACKGROUND_COLOR_ ACTIVE	rgba(0,0,0,0.8)	Banner button default background activate color
PT_BANNER_BUTTON_ BACKGROUND_COLOR_ HOVER	rgba(255,255,255,0.35)	Banner button default background hover color
PT_BANNER_BUTTON_ BADGE_BG_COLOR	#39C5A6	Banner badge background color (primarily for notifications)
PT_BANNER_BUTTON_ BADGE_BORDER_COLOR	#39C5A6	Banner badge border color (primarily for notifications)
PT_BANNER_BUTTON_ BADGE_TEXT_COLOR	#000	Banner badge text color (primarily for notifications)
PT_BANNER_BUTTON_ BOXSHADOW	none	Banner button box shadow
PT_BANNER_BUTTON_ BOXSHADOW_ACTIVE	none	Banner button default activate boxshadow
PT_BANNER_BUTTON_ BOXSHADOW_HOVER	none	Banner button default hover boxshadow
PT_BANNER_BUTTON_ BOXSHADOW_HOVER_ RTL	none	Banner button default hover boxshadow (RTL)
PT_BANNER_BUTTON_ TEXT_COLOR	rgba(255,255,255,0.95)	Banner button default text color

Macro Name	***Macro Content***	***Description***
PT_BANNER_BUTTON_ TRUE_BG_COLOR_ACTIVE	rgba(0,0,0,0.8)	Banner button actual activate color
PT_BANNER_BUTTON_ TRUE_BG_COLOR_HOVER	rgba(255,255,255,0.35)	Banner button actual hover color
PT_BANNER_BUTTON _TRUE_BXSHADOW_ ACTIVE	none	Banner button actual activate box shadow
PT_BANNER_BUTTON _TRUE_BXSHADOW_ HOVER	none	Banner button actual hover box shadow
PT_BANNER_GUIDED _NOTIFICATION_ SEPARATOR	transparent	Banner border color separator for the notification button from the guided buttons
PT_BANNER_NAVBAR_ BACKGROUND_ACTIVE	transparent	Banner NavBar button background (general background property) activate
PT_BANNER_NAVBAR_ BACKGROUND_HOVER	rgba(255,255,255,0.35)	Banner NavBar button background (general background property) hover
PT_BANNER_NAVBAR_ BACKGROUND_IMAGE	none	Banner NavBar button background image (set to none to use the background color)
PT_BANNER_NAVBAR_ BOXSHADOW	none	Banner NavBar button box shadow
PT_BANNER_NAVBAR_ IMAGE_CSS	margin-top:-1px; margin-%AlignStart:0;	Banner navbar button icon image css adjustments
PT_BANNER_NAVBAR_ SIDE_BORDER_COLOR	transparent	Banner NavBar button Left or Right Border Color
PT_BANNER_SEARCH_ PRESSED_BG_COLOR	#6c7784	Banner button (search) depressed state background color
PT_BANNER_SUBMIT_ BACKGROUND_COLOR	none	Banner submit button background color

Macro Name	*Macro Content*	*Description*
PT_BANNER_SUBMIT_ BACKGROUND_COLOR_ ACTIVE	none	Banner submit button background color for activate
PT_BANNER_SUBMIT_ BACKGROUND_COLOR_ HOVER	none	Banner submit button background color for hover
PT_BANNER_SUBMIT_ BORDER_COLOR	none	Banner submit button border color
PT_BANNER_SUBMIT_ BORDER_COLOR_ACTIVE	none	Banner submit button border color for activate
PT_BANNER_SUBMIT_ BORDER_COLOR_HOVER	none	Banner submit button border color for hover
PT_BANNER_SUBMIT_ BOXSHADOW_HOVER	none	Banner submit button box shadow for hover
PT_BANNER_SUBMIT_ TEXTSHADOW_ACTIVE	none	Banner submit button text shadow for activate
PT_BANNER_SUBMIT_ TEXTSHADOW_HOVER	none	Banner submit button text shadow for hover
PT_BANNER_SUBMIT_ TEXTSHADOW_RTL	none	Banner submit button text shadow (RTL)
PT_BANNER_SUBMIT_ TEXT_COLOR	none	Banner submit button text color
PT_BANNER_SUBMIT_ TEXT_COLOR_ACTIVE	none	Banner submit button text color for activate
PT_BANNER_SUBMIT_ TEXT_COLOR_HOVER	none	Banner submit button text color for hover
PT_COMPINFO_ BACKGROUND	#1d2733	Company Information RTE background

Macro Name	***Macro Content***	***Description***
PT_COMPINFO_ BOTTOMBORDER	transparent	Company Information RTE bottom border
PT_COMPINFO_ TEXTCOLOR	rgba(255,255,255,0.95)	Company Information RTE default font color
PT_CONFIRMATION_ TEXTSHADOW	none	Confirmation Area text shadow
PT_CONFIRMATION_ TEXTSHADOW_RTL	none	Confirmation Area text shadow (RTL)
PT_HOMEPAGE_ BACKGROUND	#eff1f4	Homepage/Dashboard background
PT_HOMEPAGE_BANNER_ BORDER_COLOR	#f0f0f0	Homepage/Dashboard Banner border color different from default banner bottom border
PT_HOMEPAGE_DASHHDR _EMB_BORDER_COLOR	#ced4da	Dashboard Header Exterior border color when embedded in Activity Guide, Nav Collection, or MD
PT_HOMEPAGE_DASHHDR _EXTPIN_BACKGROUND	#f8f8f8	Dashboard Header Exterior pinned/fixed background (when interior has not yet rendered)
PT_HOMEPAGE_DASHHDR _EXTPIN_BORDERTOP_ CLR	#fff	Dashboard Header Exterior pinned/fixed top border color
PT_HOMEPAGE_DASHHDR _EXTPIN_BORDER_COLOR	#a8a8a8	Dashboard Header Exterior pinned/fixed border color
PT_HOMEPAGE_DASHHDR _EXTPIN_BOXSHADOW	0px 1px 16px rgba(0,0,0,0.35)	Dashboard Header Exterior pinned/fixed shadow
PT_HOMEPAGE_DASHHDR _EXT_BORDER_COLOR	#ced4da	Dashboard Header Exterior border color (default)
PT_HOMEPAGE_DASHHDR _INT_PIN_BCKGRD	#f8f8f8	Dashboard Header Interior pinned/fixed background

Macro Name	*Macro Content*	*Description*
PT_HOMEPAGE_DASHHDR_INT_UNPIN_LABEL_CLR	#5a5a5a	Dashboard Header Interior unpinned label text color
PT_HOMEPAGE_FOOTER_BACKGROUND	rgba(29,39,51,0.48)	Homepage/Dashboard Footer background
PT_HOMEPAGE_REFRESH_IMAGE_SCALE	1	Homepage/Dashboard Refresh button image scale factor
PT_HOMEPAGE_REFRESH_SIDE_ADJUST	-1px	Homepage/Dashboard Refresh button right (or left in RTL) positioning adjustment
PT_HOMEPAGE_REFRESH_TOP_ADJUST	1px	Homepage/Dashboard Refresh button top positioning adjustment
PT_HOMEPAGE_TILE_BORDER	1px solid #ced4da	Homepage/Dashboard Tile border
PT_HOMEPAGE_TILE_BOXSHADOW	none	Homepage/Dashboard Tile box shadow
PT_HOMEPAGE_TILE_BOXSHADOW_RTL	none	Homepage/Dashboard Tile box shadow (RTL)
PT_LOGO_CSS	transform: scale(0.75);	Company Logo general image adjustments for non-small form factor
PT_LOGO_CSS_SMALL	transform: scale(0.75);	Company Logo general image adjustments for small form factor
PT_LOGO_TOP_PADDING	0.5em	Company Logo top padding for non-small form factor
PT_LOGO_TOP_PADDING_SMALL	0.4em	Company Logo top padding for small form factor
PT_PANEL_TAB_RC_IMAGE_CSS	padding-%AlignStart:1px;	Panel Tab Related Content image specific CSS adjustments
PT_PANEL_TAB_RC_IMAGE_PADDINGTOP	0.3em	Panel Tab Related Content image top padding

Macro Name	***Macro Content***	***Description***
PT_PANEL_TAB_RC_IMAGE_TRANSFORM	scale(1)	Panel Tab Related Content image transformation (typically scalling)
PT_PANEL_TAB_SIDE1_BACKGROUND	#1d2733	Panel Tab (Side 1 or Navigation) background
PT_PANEL_TAB_SIDE1_BACKGROUND_ACTIVE	#1c579c	Panel Tab (Side 1 or Navigation) background for active
PT_PANEL_TAB_SIDE1_BACKGROUND_HOVER	#1d2733	Panel Tab (Side 1 or Navigation) background for hover
PT_PANEL_TAB_SIDE1_BORDER_COLOR	#1d2733	Panel Tab (Side 1 or Navigation) border color
PT_PANEL_TAB_SIDE1_BOXSHADOW_ACTIVE	none	Panel Tab (Side 1 or Navigation) box shadow for active
PT_PANEL_TAB_SIDE1_BOXSHADOW_ACTIVE_RTL	none	Panel Tab (Side 1 or Navigation) box shadow for active (RTL)
PT_PANEL_TAB_SIDE1_BOXSHADOW_HOVER	none	Panel Tab (Side 1 or Navigation) box shadow for hover
PT_PANEL_TAB_SIDE1_BOXSHADOW_HOVER_RTL	none	Panel Tab (Side 1 or Navigation) box shadow for hover (RTL)
PT_PANEL_TAB_SIDE1_IMAGE_CSS	%AlignStart:2px;	Panel Tab (Side 1 or Navigation) image specific CSS adjustments
PT_PANEL_TAB_SIDE1_IMAGE_PADDINGTOP	0.29em	Panel Tab (Side 1 or Navigation) image top padding
PT_PANEL_TAB_SIDE1_IMAGE_TRANSFORM	scale(1)	Panel Tab (Side 1 or Navigation) image transformation (typically scalling)
PT_PANEL_TAB_SIDE2_BACKGROUND	#b9ddfb	Panel Tab (Side 2 or Related Content) background

Macro Name	*Macro Content*	*Description*
PT_PANEL_TAB_SIDE2_ BACKGROUND_HOVER	#d4eafc	Panel Tab (Side 2 or Related Content) background for hover/active
PT_PANEL_TAB_SIDE2_ IMAGE_CSS	padding-%AlignStart:1px;	Panel Tab (Side 2) image specific CSS adjustments
PT_PANEL_TAB_SIDE2_ IMAGE_PADDINGTOP	0.3em	Panel Tab (Side 2) image top padding
PT_PANEL_TAB_SIDE2_ IMAGE_TRANSFORM	scale(1)	Panel Tab (Side 2) image transformation (typically scalling)
PT_SEARCH_TRAY_ BACKGROUND_COLOR	#6a7481	Banner Search Tray background color
PT_SEARCH_TRAY_ GOBUTTON_IMAGE	PT_NUI_EXECUTE_ARW	Banner search tray go button icon image name
PT_TITLE_SELECTOR_ IMAGE_CSS	transform:scale(1);	Banner Title Selector popup indicator icon image css adjustments

New Macros in PeopleTools 8.59

The following macros were added in PeopleTools 8.59:

Macro Name	*Macro Content*	*Description*
PT_BANNER_BACK_ BORDER_BOTTOM_COLOR _ACTIV	#657081	Banner back button border color (bottom) for activate
PT_BANNER_BACK_ BORDER_COLOR_ACTIVE	#657081	Banner back button border color for activate
PT_HOMEPAGE_NAVAREA _BACKGROUND	inherit	Homepage Navigation Area background
PT_HOMEPAGE_NAVAREA _BORDER_COLOR	#848c94	Homepage Navigation Area border color

Macro Name	***Macro Content***	***Description***
PT_HOMEPAGE_NAVAREA_CSS		Homepage Navigation Area generic CSS
PT_HOMEPAGE_NAVAREA_TEXT_COLOR	#1d2733	Homepage Navigation Area text color (default for area)
PT_HOMEPAGE_NAV_NEXT_IMAGE	PTNUI_NEXTPAGE_ICN	Homepage Navigation Next Page image
PT_HOMEPAGE_NAV_NEXT_IMAGE_CSS	text-align:center;margin-top:0.345em;	Homepage Navigation Next Page image position adjustments
PT_HOMEPAGE_NAV_PREV_IMAGE	PTNUI_PREVPAGE_ICN	Homepage Navigation Previous Page image
PT_HOMEPAGE_NAV_PREV_IMAGE_CSS	text-align:center;margin-top:0.345em;	Homepage Navigation Previous Page image position adjustments
PT_HOMEPAGE_PANEL_BORDER_COLOR	#848c94	Homepage SIDE PANEL border colors (QAB and Notification)
PT_HOMEPAGE_POSITION_TEXT_COLOR	inherit	Homepage Page Positioning Text Color
PT_HOMEPAGE_SEARCH_BORDER_COLOR	#000	Homepage Search Area in Banner border color
PT_HOMEPAGE_TITLE_SELECTOR_BACKGD_ACTIVE	#b9ddfb	Homepage Title Selector background for activate
PT_HOMEPAGE_TITLE_SELECTOR_BACKGD_HOVER	#f4f7fc	Homepage Title Selector background for hover
PT_HOMEPAGE_TITLE_SELECTOR_BORDER_ACTIVE	#1d2733	Homepage Title Selector border color for activate

Macro Name	Macro Content	Description
PT_HOMEPAGE_TITLE_SELECTOR_BORDER_HOVER	#868e96	Homepage Title Selector border color for hover
PT_HOMEPAGE_TITLE_SELECTOR_IMAGE	PTNUI_HPSEL_ARROW_ICN	Homepage Title Selector image
PT_HOMEPAGE_TITLE_SELECTOR_IMAGE_CSS		Homepage Title Selector image generic CSS
PT_HOMEPAGE_TITLE_SELECTOR_TEXT_ACTIVE	#1d2733	Homepage Title Selector text color for activate
PT_HOMEPAGE_TITLE_SELECTOR_TEXT_COLOR	inherit	Homepage/Dashboard title selector text color
PT_HOMEPAGE_TITLE_SELECTOR_TEXT_HOVER	#1c579c	Homepage Title Selector text color for hover

Updated Macros in PeopleTools 8.59

The following macros were updated in PeopleTools 8.59:

Macro Name	Macro Content	Description
PT_BANNER_BACK_BORDER_BOTTOM_COLOR	#657081	Banner back button border color (bottom)
PT_BANNER_BACK_BORDER_BOTTOM_COLOR_HOVER	transparent	Banner back button border color (bottom) for hover
PT_BANNER_BACK_BORDER_COLOR	#657081	Banner back button border color
PT_BANNER_BACK_BORDER_COLOR_HOVER	transparent	Banner back button border color for hover
PT_BANNER_BUTTON_BADGE_BG_COLOR	#53a0fd	Banner badge background color (primarily for notifications)

Macro Name	***Macro Content***	***Description***
PT_BANNER_BUTTON_BADGE_BORDER_COLOR	#53a0fd	Banner badge border color (primarily for notifications)
PT_BANNER_BUTTON_BADGE_TEXT_COLOR	transparent	Banner badge text color (primarily for notifications)
PT_HOMEPAGE_DASHHDR_EMB_BORDER_COLOR	#848c94	Dashboard Header Exterior border color when embedded in Activity Guide, Nav Collection, or MD
PT_HOMEPAGE_DASHHDR_EXT_BORDER_COLOR	#848c94	Dashboard Header Exterior border color (default)
PT_HOMEPAGE_TILE_BORDER	1px solid #848c94	Dashboard Tile border
PT_HOMEPAGE_TILE_BOXSHADOW_HOVER	0 8px 20px 0 rgba(29,39,51,0.30), inset 0 0 2px #246fc7)	Dashboard Title box shadow for hover
PT_HOMEPAGE_TITLE_SELECTOR_TEXT_COLOR	inherit	Dashboard title selector text color
PT_LOGO_CSS	transform: scale(0.8);	Company Logo general image adjustments for non-small form factor
PT_LOGO_CSS_SMALL	transform: scale(0.8);	Company Logo general image adjustments for small form factor
PT_LOGO_IMAGE_NAME	PTNUI_ORACLE_LOGO_SFF	Company Logo image name for non-small form factor
PT_LOGO_IMAGE_SIDE_MARGIN	-0.5em	Company Logo left (or right for RTL) side adjustment for non-small form factor
PT_LOGO_IMAGE_SIDE_MARGIN_SMALL	-0.5em	Company Logo left (or right for RTL) side adjustment for small form factor only
PT_LOGO_TOP_PADDING	0.2em	Company Logo top padding for non-small form factor
PT_LOGO_TOP_PADDING_SMALL	0.2em	Company Logo top padding for small form factor

New Macros in PeopleTools 8.60

The following macros were added in PeopleTools 8.60:

Macro Name	***Macro Content***	***Description***
PT_BANNER_CONTEXTLIST_IMAGE	PT_CONTEXT_MENU_ICN	Banner context list button icon image name
PT_BANNER_CONTEXTLIST_IMAGE_CSS	margin-top: 0.15em;transform: scale(1.1);	Banner context list button icon image css adjustments
PT_BANNER_FAVORITES_IMAGE	PT_FAV_HDR_ICN	Banner Favorites button icon image name
PT_BANNER_FAVORITES_IMAGE_CSS		Banner Favorites button icon image css adjustments
PT_BANNER_LOWER_DECORATION	var(--pt-strip-background)	Banner lower decorative separator
PT_BANNER_LOWER_DECORATION_CSS		Banner lower decorative separator css adjustments
PT_BANNER_LOWER_DECORATION_CSS_SSF		Banner lower decorative separator css adjustments for small form factor
PT_BANNER_LOWER_DECORATION_HEIGHT	var(--pt-strip-height)	Banner lower decorative separator height for non-small form factor
PT_BANNER_LOWER_DECORATION_HEIGHT_SFF	var(--pt-strip-height)	Banner lower decorative separator height for small form factor
PT_BANNER_PAGETITLE_BACKGROUND_COLOR	var(--pt-page-title-background-color)	Banner non-small form factor page title background color
PT_BANNER_PAGETITLE_BORDER_COLOR	var(--pt-page-title-border-color)	Banner non-small form factor page title bottom border color
PT_BANNER_PAGETITLE_TEXT_COLOR	var(--pt-page-title-text-color)	Banner non-small form factor page title text color
PT_BANNER_RECENTS_IMAGE	PT_MRU_HDR_ICN	Banner Recents button icon image name

Macro Name	***Macro Content***	***Description***
PT_BANNER_RECENTS_IMAGE_CSS		Banner Recents button icon image css adjustments
PT_BANNER_SEPARATOR_BORDER_COLOR	var(--pt-banner-separator-color)	Banner Back or Logo/QAB separator border color
PT_HOMEPAGE_TITLE_SELECTOR_OUTLINE_HOVER	1px dashed var(--pt-hover-outline-color)	Homepage title selector outline(really border) styling
PT_PANEL_TAB_SIDE2_BACKGROUND_ACTIVE	var(--pt-panel-tab-alt-active-background-color)	Panel Tab (Side 2 or Related Content) background for active
PT_SEARCH_SUBDUED_BACKGROUND_COLOR	transparent !important	Banner subdued search background color
PT_SEARCH_SUBDUED_BORDER_COLOR	var(--pt-banner-text-color)	Banner subdued search border color
PT_SEARCH_SUBDUED_IMAGE	PT_HEADER_SEARCH	Banner subdued search image within search input
PT_SEARCH_SUBDUED_IMAGE_CSS		Banner subdued search image within search input css adjustments
PT_SEARCH_SUBDUED_OPACITY	var(--pt-banner-search-subdued-opacity)	Banner subdued search opacity
PT_SEARCH_SUBDUED_OPACITY_HOVER	var(--pt-banner-search-subdued-hover-opacity)	Banner subdued search opacity for hover
PT_SEARCH_SUBDUED_TEXT_COLOR	var(--pt-banner-text-color)	Banner subdued search text/placeholder color

Updated Macros in PeopleTools 8.60

The following macros were updated in PeopleTools 8.60:

Macro Name	***Macro Content***	***Description***
PT_BANNER_BACKGROUND_COLOR	var(--pt-banner-background-color)	Banner background color (set background image to none to see flat color)
PT_BANNER_BACK_BACKGROUND_COLOR_ACTIVE	var(--pt-banner-button-active-background-color)	Banner back button background color for activate
PT_BANNER_BACK_BACKGROUND_COLOR_HOVER	var(--pt-banner-button-hover-background-color)	Banner back button background color for hover
PT_BANNER_BACK_BORDER_BOTTOM_COLOR	var(--pt-border-normal-color)	Banner back button border color (bottom)
PT_BANNER_BACK_BORDER_BOTTOM_COLOR_ACTIV	transparent	Banner back button border color (bottom) for activate
PT_BANNER_BACK_BORDER_COLOR	var(--pt-border-normal-color)	Banner back button border color
PT_BANNER_BACK_BORDER_COLOR_ACTIVE	transparent	Banner back button border color for activate
PT_BANNER_BACK_IMAGE_CSS	display:inline-block;margin-top:0.345em;transform:scale(1.5);	Banner back button icon image css adjustments
PT_BANNER_BACK_IMAGE_NARROW_CSS		Banner back button image adjustment css when text disappears from button (only image)
PT_BANNER_BORDER_COLOR	transparent	Banner bottom border color
PT_BANNER_BUTTON_BACKGROUND_COLOR_ACTIVE	var(--pt-banner-button-active-background-color)	Banner button default background activate color
PT_BANNER_BUTTON_BACKGROUND_COLOR_HOVER	var(--pt-banner-button-hover-background-color)	Banner button default background hover color
PT_BANNER_BUTTON_BADGE_BG_COLOR	var(--pt-badge-dot-alt-color)	Banner badge background color (primarily for notifications)

Macro Name	***Macro Content***	***Description***
PT_BANNER_BUTTON_BADGE_BORDER_COLOR	var(--pt-badge-dot-alt-color)	Banner badge border color (primarily for notifications)
PT_BANNER_BUTTON_TEXT_COLOR	var(--pt-banner-text-color)	Banner button default text color
PT_BANNER_BUTTON_TEXT_COLOR_ACTIVE	var(--pt-banner-text-color)	Banner button default text activate color
PT_BANNER_BUTTON_TEXT_COLOR_HOVER	var(--pt-banner-text-color)	Banner button default text hover color
PT_BANNER_BUTTON_TRUE_BG_COLOR_ACTIVE	var(--pt-banner-button-active-background-color)	Banner button actual activate color
PT_BANNER_BUTTON_TRUE_BG_COLOR_HOVER	var(--pt-banner-button-hover-background-color)	Banner button actual hover color
PT_BANNER_IMAGE_BUTTON_RADIUS	3px	Banner image button border radius
PT_BANNER_NAVBAR_BACKGROUND_ACTIVE	var(--pt-banner-button-active-background-color)	Banner NavBar button background (general background property) activate
PT_BANNER_NAVBAR_BACKGROUND_HOVER	var(--pt-banner-button-hover-background-color)	Banner NavBar button background (general background property) hover
PT_BANNER_TEXT_COLOR	var(--pt-banner-text-color)	Banner default font color
PT_BANNER_UPPER_DECORATION	var(--pt-strip-p860-background,none)	Banner decorative border for reversion to pre-8.60 (see PT_CUSTOM_PROPS_PRE860)
PT_BANNER_UPPER_DECORATION_HEIGHT	var(--pt-strip-p860-height,0px)	Banner decorative border height for reversion (use 0.25em to display) - (see PT_CUSTOM_PROPS_PRE860)
PT_COMPINFO_BACKGROUND	var(--pt-banner-background-color)	Company information RTE background

Macro Name	***Macro Content***	***Description***
PT_COMPINFO_TEXTCOLOR	var(--pt-banner-text-color)	Company information RTE default font color
PT_CONFIRMATION_BACKGROUND	var(--pt-confirmation-background-color)	Confirmation area background
PT_CONFIRMATION_BORDER_COLOR	var(--pt-confirmation-border-color)	Conformation area border color
PT_CONFIRMATION_TEXT_COLOR	var(--pt-confirmation-text-color)	Confirmation area text color
PT_HOMEPAGE_BACKGROUND	var(--pt-homepage-background)	Homepage/Dashboard background
PT_HOMEPAGE_BANNER_BORDER_COLOR	var(--pt-page-title-background-color)	Homepage/Dashboard banner border color different from default banner bottom border
PT_HOMEPAGE_DASHHDR_EMB_BORDER_COLOR	var(--pt-panel-border-color)	Dashboard Header exterior border color when embedded in Activity Guide, Nav Collection, or MD
PT_HOMEPAGE_DASHHDR_EXTPIN_BACKGROUND	var(--pt-panel-background-color)	Dashboard Header Exterior pinned/fixed background (when interior has not yet rendered)
PT_HOMEPAGE_DASHHDR_EXTPIN_BORDERTOP_CLR	var(--pt-panel-background-color)	Dashboard Header exterior pinned/fixed top border color
PT_HOMEPAGE_DASHHDR_EXTPIN_BORDER_COLOR	var(--pt-panel-border-color)	Dashboard Header exterior pinned/fixed border color
PT_HOMEPAGE_DASHHDR_EXTPIN_BOXSHADOW	0px 1px 16px var(--pt-modal-shadow-color)	Dashboard Header exterior pinned/fixed shadow
PT_HOMEPAGE_DASHHDR_EXT_BACKGROUND	var(--pt-page-background-color)	Dashboard Header exterior background (when interior has not yet rendered)
PT_HOMEPAGE_DASHHDR_EXT_BORDER_COLOR	var(--pt-panel-border-color)	Dashboard Header exterior border color (default)

Macro Name	***Macro Content***	***Description***
PT_HOMEPAGE_DASHHDR_EXT_BORDER_RADIUS	0	Dashboard Header exterior border radius (default)
PT_HOMEPAGE_DASHHDR_INT_PIN_BCKGRD	var(--pt-panel-background-color)	Dashboard Header interior pinned/fixed background
PT_HOMEPAGE_DASHHDR_INT_UNPIN_BCKGRD	var(--pt-page-background-color)	Dashboard Header interior unpinned background
PT_HOMEPAGE_DASHHDR_INT_UNPIN_HEADER_CLR	var(--pt-header-text-color)	Dashboard Header interior unpinned groupbox header text color
PT_HOMEPAGE_DASHHDR_INT_UNPIN_LABEL_CLR	var(--pt-label-text-color)	Dashboard Header interior unpinned label text color
PT_HOMEPAGE_FOOTER_BACKGROUND	var(--pt-modal-mask-background-color)	Homepage/Dashboard footer background
PT_HOMEPAGE_NAVAREA_BACKGROUND	var(--pt-page-title-background-color)	Homepage Navigation area background
PT_HOMEPAGE_NAVAREA_BORDER_COLOR	var(--pt-page-title-border-color)	Homepage Navigation area border color
PT_HOMEPAGE_NAVAREA_TEXT_COLOR	var(--pt-page-title-text-color)	Homepage Navigation area text color (default for area)
PT_HOMEPAGE_PANEL_BORDER_COLOR	var(--pt-panel-border-color)	Homepage side panel border colors (QAB and Notification)
PT_HOMEPAGE_REFRESH_SIDE_ADJUST	2px	Homepage/Dashboard refresh button right (or left in RTL) positioning adjustment
PT_HOMEPAGE_REFRESH_TOP_ADJUST	2px	Homepage/Dashboard refresh button top positioning adjustment
PT_HOMEPAGE_SEARCH_BORDER_COLOR	var(--pt-border-contrast-color)	Homepage Search area in banner border color

Macro Name	Macro Content	Description
PT_HOMEPAGE_TILE_BORDER	1px solid var(--pt-tile-border-color)	Homepage/Dashboard tile border
PT_HOMEPAGE_TILE_BORDER_HOVER	var(--pt-tile-hover-border-color)	Homepage/Dashboard tile border color for hover
PT_HOMEPAGE_TILE_BOXSHADOW_HOVER	0px 8px 24px -6px var(--pt-tile-hover-shadow-color)	Homepage/Dashboard title box shadow for hover
PT_HOMEPAGE_TILE_TITLE_COLOR	var(--pt-tile-header-text-color)	Homepage/Dashboard tile title text color
PT_HOMEPAGE_TITLE_SELECTOR_BACKGD_ACTIVE	var(--pt-button-active-background-color)	Homepage title selector background for activate
PT_HOMEPAGE_TITLE_SELECTOR_BACKGD_HOVER	var(--pt-button-hover-background-color)	Homepage Title selector background for hover
PT_HOMEPAGE_TITLE_SELECTOR_BORDER_ACTIVE	var(--pt-button-active-border-color)	Homepage Title selector border color for activate
PT_HOMEPAGE_TITLE_SELECTOR_BORDER_HOVER	var(--pt-button-hover-border-color)	Homepage Title selector border color for hover
PT_HOMEPAGE_TITLE_SELECTOR_TEXT_ACTIVE	inherit !important	Homepage Title selector text color for activate
PT_HOMEPAGE_TITLE_SELECTOR_TEXT_COLOR	inherit !important	Homepage/Dashboard title selector text color
PT_HOMEPAGE_TITLE_SELECTOR_TEXT_HOVER	inherit !important	Homepage Title selector text color for hover
PT_LOGO_CSS	margin-%AlignEnd:-0.4em;	Company Logo general image adjustments for non-small form factor
PT_LOGO_CSS_SMALL	margin-%AlignEnd:-0.4em;	Company Logo general image adjustments for small form factor

Macro Name	***Macro Content***	***Description***
PT_LOGO_IMAGE_SIDE_MARGIN	0	Company Logo left (or right for RTL) side adjustment for non-small form factor
PT_LOGO_IMAGE_SIDE_MARGIN_SMALL	0	Company Logo left (or right for RTL) side adjustment for small form factor only
PT_PANEL_TAB_SIDE1_BACKGROUND	var(--pt-panel-tab-background-color)	Panel Tab (Side 1 or Navigation) background
PT_PANEL_TAB_SIDE1_BACKGROUND_ACTIVE	var(--pt-panel-tab-active-background-color)	Panel Tab (Side 1 or Navigation) background for active
PT_PANEL_TAB_SIDE1_BACKGROUND_HOVER	var(--pt-panel-tab-hover-background-color)	Panel Tab (Side 1 or Navigation) background for hover
PT_PANEL_TAB_SIDE1_BORDER_COLOR	var(--pt-panel-tab-border-color)	Panel Tab (Side 1 or Navigation) border color
PT_PANEL_TAB_SIDE2_BACKGROUND	var(--pt-panel-tab-alt-background-color)	Panel Tab (Side 2 or Related Content) background
PT_PANEL_TAB_SIDE2_BACKGROUND_HOVER	var(--pt-panel-tab-alt-hover-background-color)	Panel Tab (Side 2 or Related Content) background for active
PT_PANEL_TAB_SIDE2_BORDER_COLOR	var(--pt-panel-tab-alt-border-color)	Panel Tab (Side 2 or Related Content) border color

Macros Removed in PeopleTools 8.60

The following macros of prior PeopleTools versions were removed in PeopleTools 8.60:

Macro Name	***Macro Content***	***Description***
PT_BANNER_AG_BTN_BACKGROUND_COLOR_ACTIVE	#fff	Banner Activity Guide all button background color for active
PT_BANNER_AG_BTN_BACKGROUND_IMAGE_ACTIVE	none	Banner Activity Guide all button background image for active

Macro Name	*Macro Content*	*Description*
PT_BANNER_AG_BTN_BORDER_COLOR_ACTIVE	#555	Banner Activity Guide all button border color for active
PT_BANNER_AG_BTN_TEXTSHADOW_ACTIVE	none	Banner Activity Guide all button text shadow for active
PT_BANNER_AG_BTN_TEXT_COLOR_ACTIVE	#000	Banner Activity Guide all button text color for active
PT_BANNER_AG_NEXT_BACKGROUND_COLOR	#fae6b4	Banner Activity Guide Next and Submit button background color
PT_BANNER_AG_NEXT_BACKGROUND_COLOR_HOVER	#ffd28c	Banner Activity Guide Next and Submit button background color for hover
PT_BANNER_AG_NEXT_BACKGROUND_IMAGE	linear-gradient(#fff,#ffd28c)	Banner Activity Guide Next and Submit button background image gradient
PT_BANNER_AG_NEXT_BACKGROUND_IMAGE_HOVER	linear-gradient(#fff,#e6aa4b)	Banner Activity Guide Next and Submit button background image gradient for hover
PT_BANNER_AG_NEXT_BORDER_COLOR	#e5be54	Banner Activity Guide Next and Submit button border color
PT_BANNER_AG_NEXT_BORDER_COLOR_HOVER	#cd8821	Banner Activity Guide Next and Submit button border color for hover
PT_BANNER_AG_NEXT_IMAGE	PT_GUIDED_NEXT_ICON_ALT	Banner Activity Guide Next directional image
PT_BANNER_AG_NEXT_IMAGE_CSS		Banner Activity Guide Next directional image specific CSS adjustments
PT_BANNER_AG_NEXT_IMAGE_DIMENSIONS	width:17px;height:17px;	Banner Activity Guide Next directional image dimensions (both width and height)
PT_BANNER_AG_NEXT_IMAGE_EDGEOFFSET	0.5em	Banner Activity Guide Next directional image offset from the edge of the button

Macro Name	Macro Content	Description
PT_BANNER_AG_NEXT_IMAGE_PADDING_TOP	1px	Banner Activity Guide Next directional image top padding
PT_BANNER_AG_NEXT_TEXTSHADOW	1px 1px 0px #feeec2	Banner Activity Guide Next and Submit button text shadow
PT_BANNER_AG_NEXT_TEXTSHADOW_HOVER	0 1px 0 #fef1de	Banner Activity Guide Next and Submit button text shadow for hover
PT_BANNER_AG_NEXT_TEXTSHADOW_RTL	-1px 1px 0px #feeec2	Banner Activity Guide Next and Submit button text shadow (RTL)
PT_BANNER_AG_NEXT_TEXT_COLOR	#000	Banner Activity Guide Next and Submit button text color
PT_BANNER_AG_PREV_BACKGROUND_COLOR	#ddd	Banner Activity Guide Previous button background color
PT_BANNER_AG_PREV_BACKGROUND_COLOR_HOVER	#bbb	Banner Activity Guide Previous button background color for hover
PT_BANNER_AG_PREV_BACKGROUND_IMAGE	linear-gradient(#fff,#ddd)	Banner Activity Guide Previous button background image
PT_BANNER_AG_PREV_BACKGROUND_IMAGE_HOVER	linear-gradient(#fff,#999)	Banner Activity Guide Previous button background image on hover
PT_BANNER_AG_PREV_BORDER_COLOR	#ccc	Banner Activity Guide Previous button border color
PT_BANNER_AG_PREV_BORDER_COLOR_HOVER	#888	Banner Activity Guide Previous button border color for hover
PT_BANNER_AG_PREV_IMAGE	PT_GUIDED_PREV_ICON_ALT	Banner Activity Guide Previous directional image
PT_BANNER_AG_PREV_IMAGE_CSS		Banner Activity Guide Previous directional image specific CSS adjustments

Macro Name	***Macro Content***	***Description***
PT_BANNER_AG_PREV_IMAGE_DIMENSIONS	width:17px;height:17px;	Banner Activity Guide Previous directional image dimensions (both width and height)
PT_BANNER_AG_PREV_IMAGE_PADDINGTOP	1px	Banner Activity Guide Previous directional image padding top
PT_BANNER_AG_PREV_TEXTSHADOW	1px 1px 0px #fff	Banner Activity Guide Previous button text shadow
PT_BANNER_AG_PREV_TEXTSHADOW_RTL	-1px 1px 0px #fff	Banner Activity Guide Previous button text shadow (RTL)
PT_BANNER_AG_PREV_TEXT_COLOR	#000	Banner Activity Guide Previous button border color
PT_BANNER_SEARCH_PRESSED_ACTIVE_CSS	background-color: rgba(0,0,0,0.8);background-image: linear-gradient(to bottom, #000, #6c7784);	Banner button (search) depressed state general css on activate
PT_BANNER_SEARCH_PRESSED_BG_COLOR	#6c7784	Banner button (search) depressed state background color
PT_BANNER_SEARCH_PRESSED_BOXSHADOW	inset 1px 1px 0 0 #5c6570, inset -1px 0 0 0 #5c6570, inset 0 2px 0 0 #eee	Banner button (search) depressed state box shadow
PT_BANNER_SEARCH_PRESSED_HOVER_CSS	background-color: rgba(255,255,255,0.35);background-image: linear-gradient (to bottom, #505862, #6c7784);	Banner button (search) depressed state general css on hover
PT_BANNER_SUBMIT_BACKGROUND_COLOR	none	Banner submit button background color
PT_BANNER_SUBMIT_BACKGROUND_COLOR_ACTIVE	none	Banner submit button background color for activate
PT_BANNER_SUBMIT_BACKGROUND_COLOR_HOVER	none	Banner submit button background color for hover
PT_BANNER_SUBMIT_BORDER_COLOR	none	Banner submit button border color

Macro Name	***Macro Content***	***Description***
PT_BANNER_SUBMIT_BORDER_COLOR_ACTIVE	none	Banner submit button border color for activate
PT_BANNER_SUBMIT_BORDER_COLOR_HOVER	none	Banner submit button border color for hover
PT_BANNER_SUBMIT_BOXSHADOW_ACTIVE	none	Banner submit button box shadow for activate
PT_BANNER_SUBMIT_BOXSHADOW_HOVER	none	Banner submit button box shadow for hover
PT_BANNER_SUBMIT_IMAGE	PT_GUIDED_SUBMIT_ICON	Banner submit button (guided activity guide) icon image name
PT_BANNER_SUBMIT_IMAGE_CSS		Banner submit button icon image css adjustments
PT_BANNER_SUBMIT_TEXTSHADOW	none	Banner submit button text shadow
PT_BANNER_SUBMIT_TEXTSHADOW_ACTIVE	none	Banner submit button text shadow for activate
PT_BANNER_SUBMIT_TEXTSHADOW_HOVER	none	Banner submit button text shadow for hover
PT_BANNER_SUBMIT_TEXTSHADOW_RTL	none	Banner submit button text shadow (RTL)
PT_BANNER_SUBMIT_TEXT_COLOR	none	Banner submit button text color
PT_BANNER_SUBMIT_TEXT_COLOR_ACTIVE	none	Banner submit button text color for activate
PT_BANNER_SUBMIT_TEXT_COLOR_HOVER	none	Banner submit button text color for hover

Macro Name	***Macro Content***	***Description***
PT_HOMEPAGE_PAGEIND_CONT_ BACKGROUND	#8b9197	Homepage/Dashboard page indicator container background color
PT_SEARCH_TRAY_BACKGROUND _COLOR	#6a7481	Banner search tray background color
PT_SEARCH_TRAY_BORDER_ COLOR	#5c6570	Banner search tray border color
PT_SEARCH_TRAY_CATEGORY_ CSS		Banner search tray category dropdown allowing multiple css specifications
PT_SEARCH_TRAY_GOBUTTON_ CSS		Banner search tray go/execute button allowing multiple css specifications
PT_SEARCH_TRAY_GOBUTTON_ IMAGE	PT_NUI_EXECUTE_ARW	Banner search tray go button icon image name
PT_SEARCH_TRAY_GOBUTTON_ IMAGE_CSS		Banner search tray go button icon image css adjustments
PT_SEARCH_TRAY_KEYWORD_ CSS		Banner search tray keyword search field allowing multiple css specifications

New Macros in PeopleTools 8.61

The following macros were added in PeopleTools 8.61:

Macro Name	***Macro Content***	***Description***
PT_BANNER_ACCESSASST_IMAGE	PT_ACCESSIBILITY_HEADER	Banner accessibility assistant button icon image name
PT_BANNER_ACCESSASST_IMAGE _CSS		Banner accessibility assistant button icon image css adjustments
PT_HOMEPAGE_SECTION_HDR_ COLPSE_IMAGE	PT_COLLAPSE_NUI	Homepage/Dashboard collapsible section collapsed icon

Macro Name	*Macro Content*	*Description*
PT_HOMEPAGE_SECTION_HDR_ COLPSE_IMAGE_CSS		Homepage/Dashboard collapsible section collapsed icon css adjustments
PT_HOMEPAGE_SECTION_HDR_ EXPAND_IMAGE	PT_EXPAND_NUI	Homepage/Dashboard collapsible section expanded icon
PT_HOMEPAGE_SECTION_HDR_ EXPAND_IMAGE_CSS		Homepage/Dashboard collapsible section expanded icon css adjustments
PT_HOMEPAGE_SECTION_HDR_ FOCUS_CSS		Homepage/Dashboard collapsible section focus css adjustments
PT_HOMEPAGE_SECTION_HDR_ TEXT_COLOR	var(--pt-page-text-color)	Homepage/Dashboard collapsible section text color
PT_HOMEPAGE_SECTION_HDR_ TEXT_COLOR_HOVER	var(--pt-header-hover-text-color)	Homepage/Dashboard collapsible section text hover color

Updated Macros in PeopleTools 8.62

The following macros were updated in PeopleTools 8.62:

Macro Name	*Macro Content*	*Description*
PT_BANNER_BACK_BORDER_ BOTTOM_COLOR	var(--pt-banner-button-border-color)	Banner back button border color (bottom)
PT_BANNER_BACK_BORDER_ COLOR	var(--pt-banner-button-border-color)	Banner back button border color

New Macros in PeopleTools 8.62

The following macros were added in PeopleTools 8.62:

Macro Name	*Macro Content*	*Description*
PT_BANNER_MIN_HEIGHT	var(--pt-banner-min-height)	Banner minimum height
PT_BANNER_MIN_HEIGHT_SFF	var(--pt-banner-min-height)	Banner minimum height for SFF

Defining Headers and Footers

Important! Classic header definitions (DEFAULT_HEADER_TANGERINE, DEFAULT_HEADER_TANGERINE_ALT, DEFAULT_HEADER_SWAN, and DEFAULT_HEADER_CLASSIC), their corresponding branding themes, the drop-down menu, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner and the fluid-like portal header, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see "Working with Fluid Pages and Controls" (Applications User's Guide).

Use the Define Headers and Footers page to create and maintain header and footer definitions by adding elements to a tree structure. The tree structure is represented as:

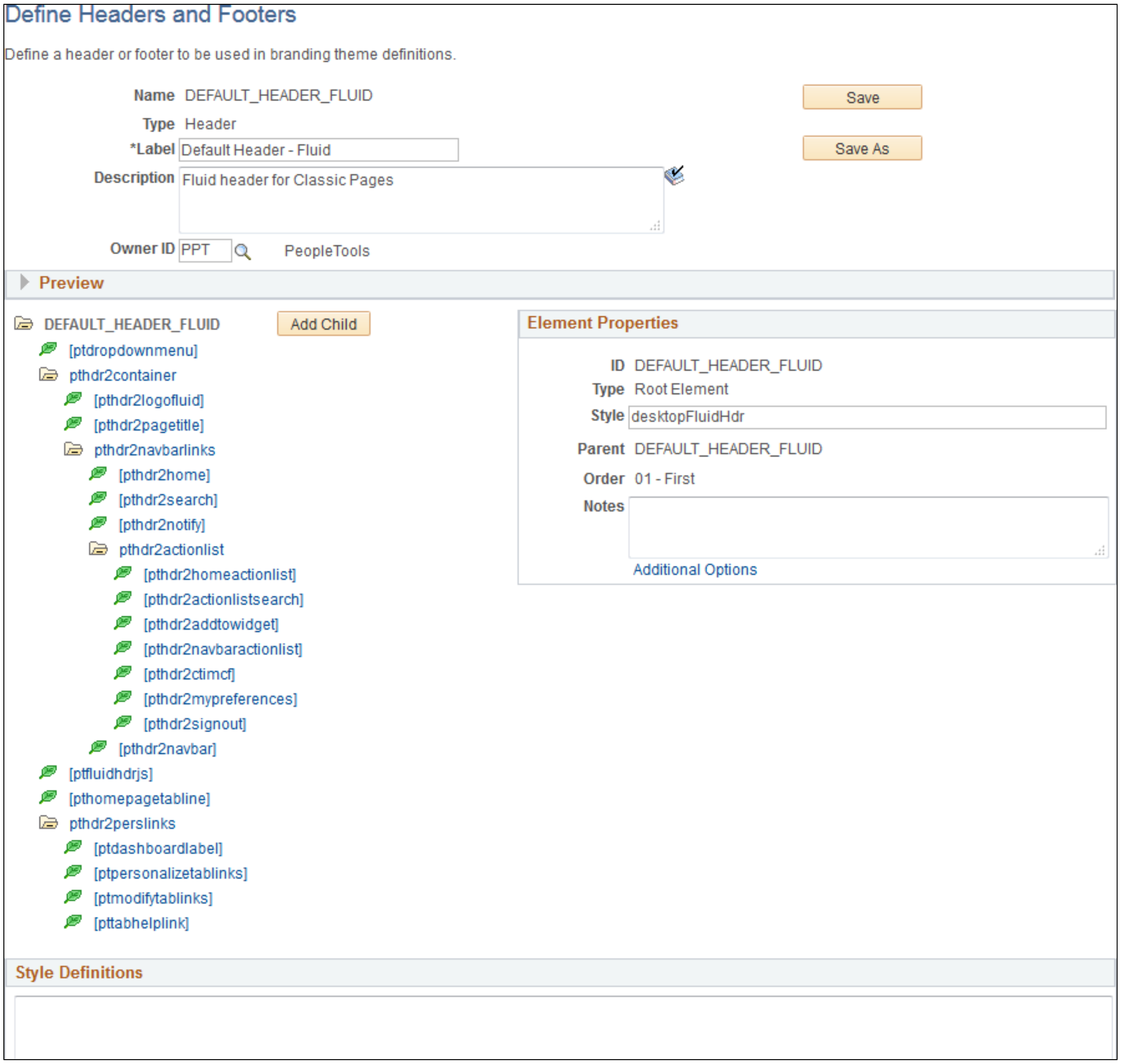
- Folders – These are container elements such as the root element (created automatically with the Root Element type) and other elements added as the Basic Container element type. Folders can contain children, which are leaves and other folders.
- Leaves – These are other elements added with other element types including basic types (Basic Link, Basic Text, Basic Image, and others) as well as specialized element types (Home Link, Sign Out Link, Dashboard Label, and others).

Note: Header and footer definitions are used on classic homepages and classic pages only. While you cannot directly modify the fluid banner that is displayed on fluid pages, you can modify definitions in a macro set that will customize the look of both the fluid banner and the header that is displayed on classic pages. In addition, you can use the CompanyInfo element to configure a custom banner for classic, fluid, or both; see Configuring a Custom Banner for more information.

Navigation:

PeopleTools > **Portal** > **Branding** > **Branding Headers and Footers**

This example illustrates the fields and controls on the Define Headers and Footers Page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Name	Displays the ID for the header or footer that was specified when the definition was created.
Type	Displays the type of the definition: • Footer • Header
Label	Enter a label for the header or footer definition.
Description	Enter a description for the header or footer definition.

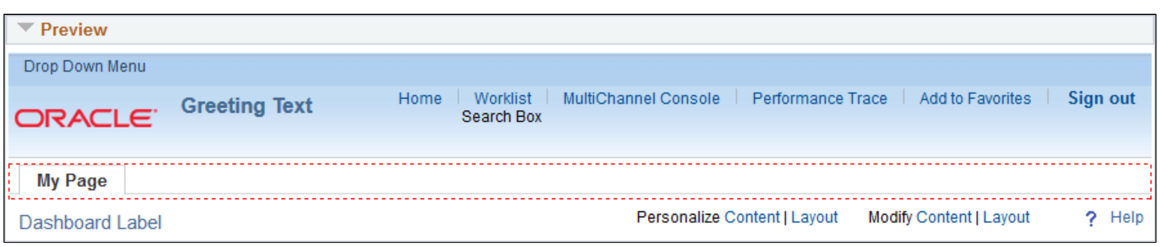
Field or Control	***Description***
Owner ID	Select which product, feature, or application owns this header or footer definition.
Save	Click the Save button to save changes to the current definition.
Save As	Click the Save As button to save the current definition with a new ID.
Delete	At the definition level, a Delete button will be displayed only for those definitions that are not included in a theme definition. Use this Delete button to delete the header or footer definition.
Add Child (in tree structure)	Select the root element or a container element and then click the Add Child button to add a child element.
Delete (in tree structure)	Select any element except the root element and then click Delete to delete the element (and any children).
Style Definitions	Enter any additional free form style class definitions in this long edit box.
Add to Project	Click the Add to Project link to access the Add To Project page to insert this branding header or footer definition into an ADS project. See "Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide) and Migrating Branding Data Using ADSs for more information.

Preview Section

Expand the Preview section to review the look of the current header or footer definition. The preview is updated after changes are made to any of the elements or style classes used in the definition. Tab out of any changed field or element to see the updated preview.

Important! The DEFAULT_HEADER_FLUID definition or clones of this definition cannot currently be previewed correctly.

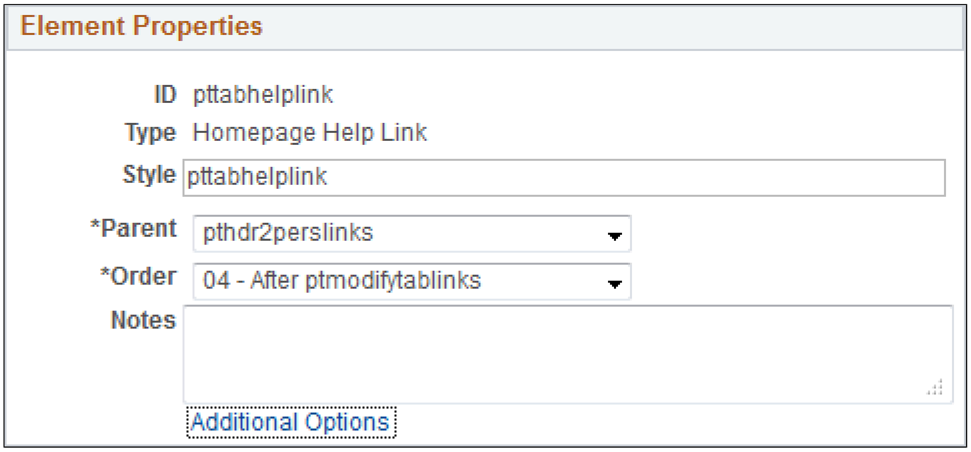
This example illustrates the preview of a header definition.



Element Properties

Use the Element Properties group box to review and set properties of the element selected in the tree.

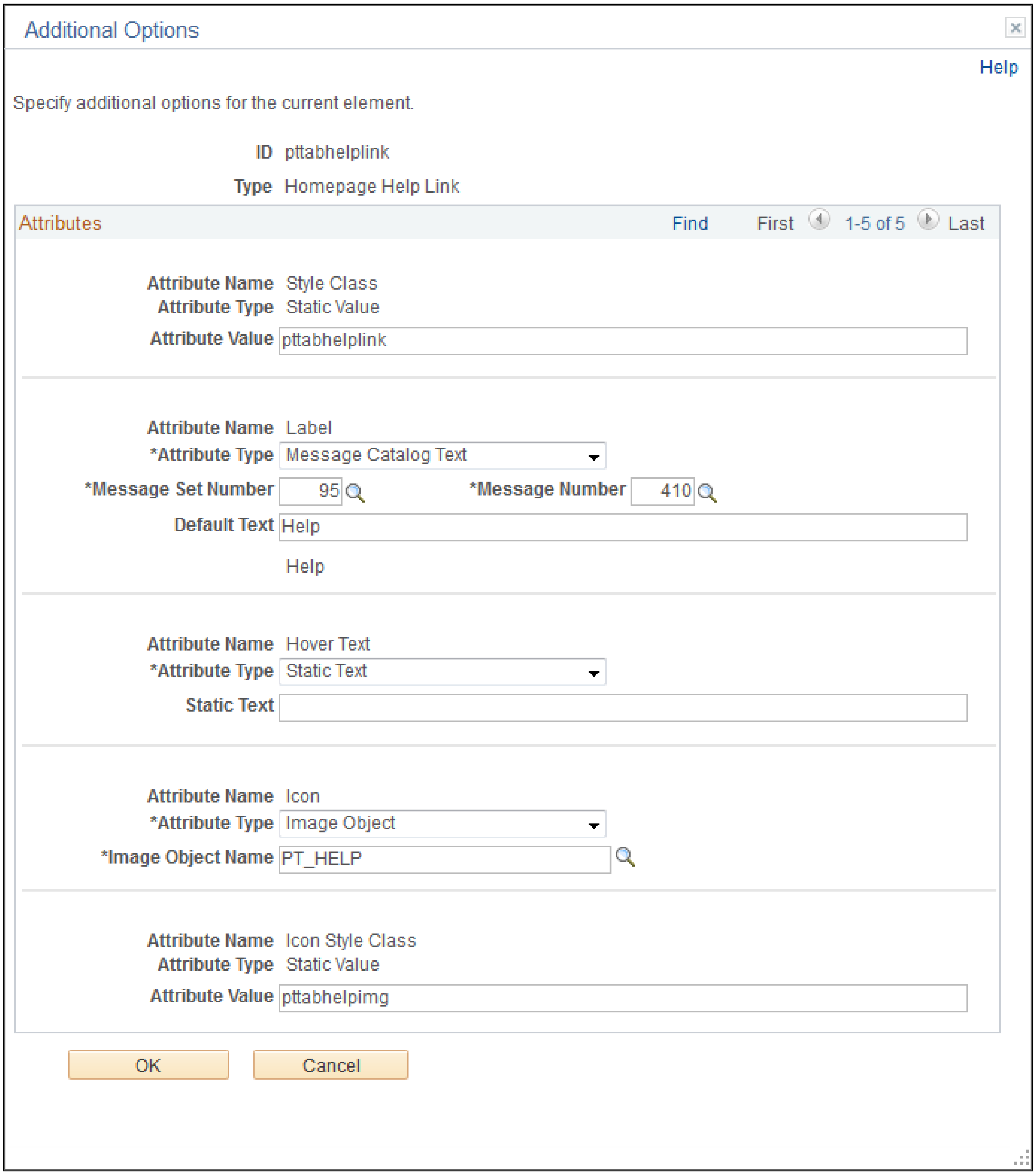
This example illustrates the fields and controls on the Element Properties group box.



Field or Control	***Description***
ID	Displays the ID of the element that was specified when the element was created.
Type	Displays the element type that was selected when the element was created. Valid element types are defined on the Define Element Types page.
Style	Enter one or more free form style class names separated by spaces to specify additional formatting for this element.
Parent	Select a new parent container for this element by container ID. **Note:** All container elements are listed except children of the current element.
Order	Select a new placement for this element among its sibling elements within the parent container.
Notes	Enter any optional notes about this element.
Additional Options	Click the Additional Options link to view the Additional Options page.

Additional Options Page

This example illustrates some of the fields and controls that can be found on the Additional Options page.



The fields displayed on the Additional Options page are controlled by the application class definition that supports this element type and will vary depending on the element type.

Field or Control	***Description***
ID	Displays the ID of the element.

Field or Control	Description
Type	Displays the element type.
Attribute Name	Displays the name of the attribute.
Attribute Type	Displays the attribute type. For some attributes, this value is not read-only and can be changed. Depending on the attribute type, additional fields are displayed showing the attribute value, static text, message set and number, default text, an object ID, and others.

Related Links

Defining Branding Element Types
Creating a New Branding Element

Configuring a Custom Banner

Important! The CompanyInfo element can only be configured in the DEFAULT_HEADER_FLUID header definition or a clone. While DEFAULT_HEADER_FLUID itself applies to classic components only, the CompanyInfo element within it can be configured for classic, fluid, or both.

You can configure a special CompanyInfo element in the DEFAULT_HEADER_FLUID header definition to create a custom banner that can include your own images, logos, and links. This custom banner is in addition to both the classic header and the fluid banner. The custom banner can be displayed above the system header on classic pages and homepages, above the fluid banner on fluid pages and homepages, or both.

When configuring the CompanyInfo element, be aware of these considerations:

- Review and adhere to the general limitations for using the rich text editor found in “Working With Rich Text Editor Fields” (Applications User’s Guide).
- Do not add new JavaScript functions–that is, do not use the <script> tag. However, you can reference PeopleSoft system JavaScript functions.
- Design your custom banner to account for the adaptive and responsive nature of both fluid and classic pages. Depending the simplicity of the design of your custom banner, you may be able to create a single combined definition for both fluid and classic pages. However, because fluid application requirements for being both adaptive and responsive differ from classic applications, you may need to select the check box to enable separate banner definitions.
- Avoid explicitly specifying font and other formatting using the rich text editor toolbar tools. Instead, create custom style classes for branding or use style classes delivered in your PeopleSoft system and apply those styles to the elements in your custom banner.

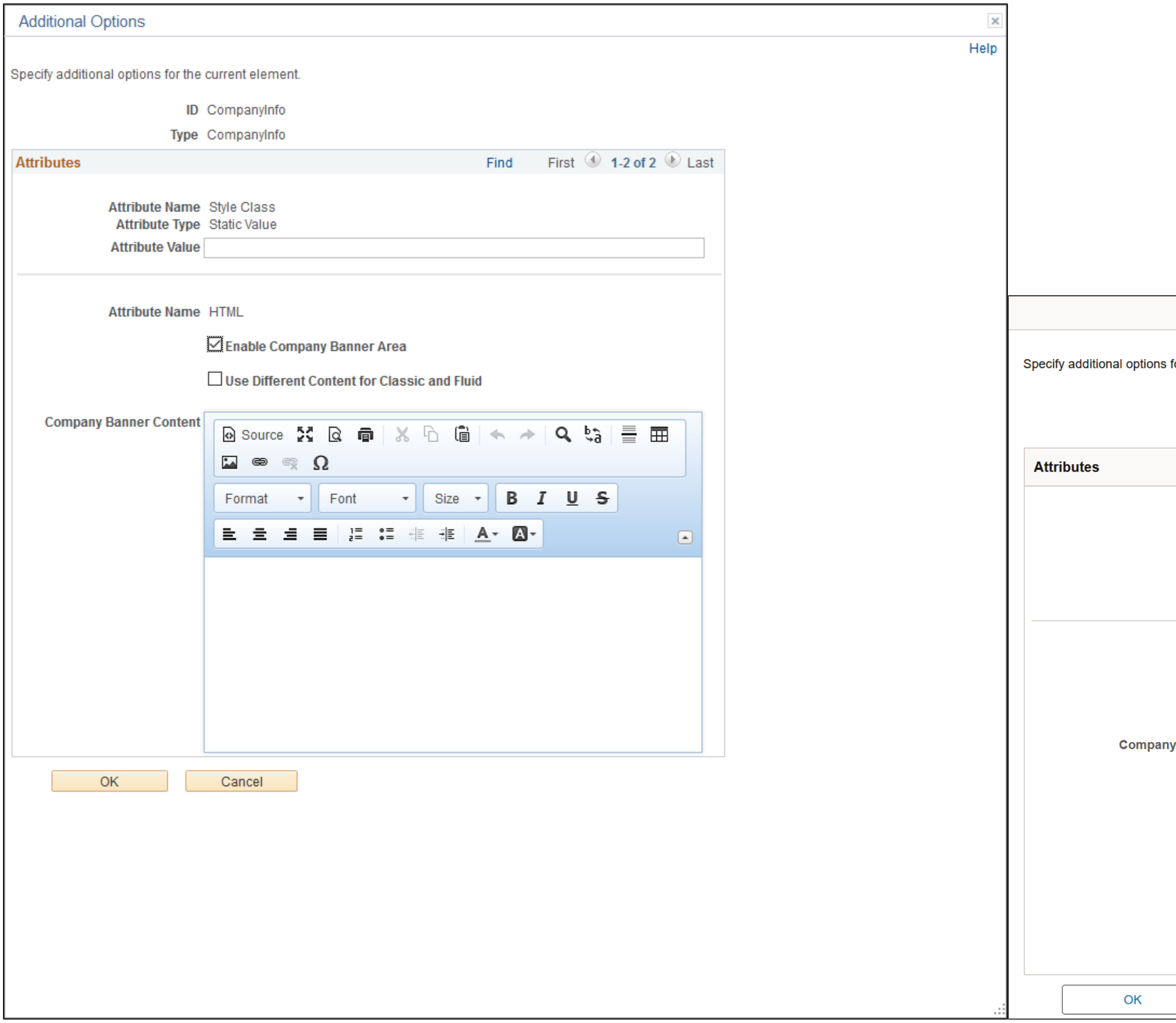
- Similarly, avoid specifying image attributes (such as height, width, border, and others) to images uploaded through the rich text editor. Remove any such attributes on the Image Properties page and use style classes instead to modify image rendering.
- Use the Source button to add <div> elements creating structure to which you can apply styling.
- In addition, use the Source button to review the HTML code to remove spurious formatting, image attributes, and other code that is not HTML5 compliant.

Note: In order to view and configure the CompanyInfo element, the branding administrator requires the Company Info Administrator role, in addition to the roles required to define headers and footers. See Roles Required for PeopleTools Branding for more information.

To configure a custom banner:

1. Select **PeopleTools** > **Portal** > **Branding** > **Define Headers and Footers**.
2. Select the DEFAULT_HEADER_FLUID header definition to customize it. (Otherwise, if you have cloned the version of DEFAULT_HEADER_FLUID delivered in PeopleTools 8.57 or later, then select your cloned header definition.)
3. Click the CompanyInfo element.
4. Click the Additional Options link.
5. Complete the additional options for the CompanyInfo element:

This example illustrates the fields and controls on the Additional Options page for the CompanyInfo element. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Style Class: Attribute Value	Enter one or more free form style class names separated by spaces to specify additional formatting for the CompanyInfo element.
Enable Company Banner Area	Select this check box to enable the custom banner. Selecting this option will reveal the rich text editor, which allows you to create the content for the custom banner.

Field or Control	*Description*
Use Different Content for Classic and Fluid	Select this check box to enable two separate rich text editor fields to create separate custom banners for classic and fluid. **Note:** When this check box is deselected, the same banner will be displayed for classic and fluid.
Banner Content	Use each rich text editor field to define the text, images, and links to be displayed in your custom banner. When the **Use Different Content for Classic and Fluid** check box is selected, two rich text editor fields are displayed: one for classic, one for fluid. Otherwise, only one rich text editor field is displayed. See "Working With Rich Text Editor Fields" (Applications User's Guide) for more information.

6. Click OK.
7. Click Save to save the header definition.
8. Test your custom banner by navigating to a classic page, a classic homepage, a fluid page, and a fluid homepage, depending on how you customized the content of the banner.
9. Optionally, use macros to define other aspects of the custom banner including its background color, bottom border, banner-specific style sheets and so on. See Updating Custom Macro Sets for the Current Release for the list of new macros added in PeopleTools 8.57 to support custom banners.

Related Links

Defining Macro Sets and Macros

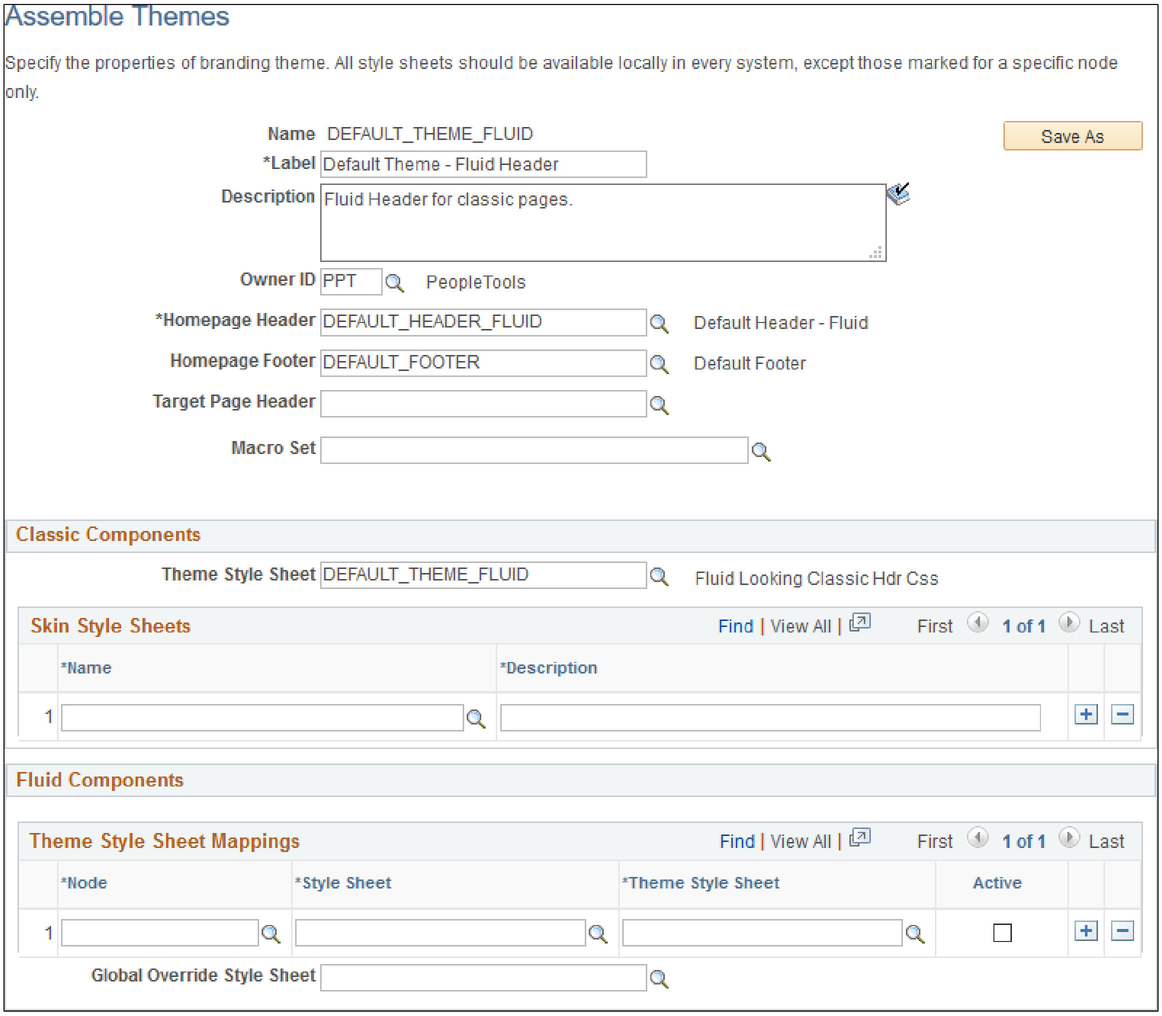
Assembling Branding Themes

Use the Assemble Themes page to create and maintain the definitions of branding themes.

Navigation:

PeopleTools > **Portal** > **Branding** > **Assemble Branding Themes**

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



Field or Control	***Description***
Name	Displays the ID of the theme that was specified when the theme was created.
Save As	Click the Save As button to save the current theme definition with a new ID.
Delete	Click the Delete button to delete the this theme definition. **Note:** The Delete button is displayed only for those theme definitions that are not assigned at the system, portal, or user attribute level.
Label	Enter a label for the theme definition.
Description	Enter a description for the theme definition.

Field or Control	*Description*
Owner ID	Select which product, feature, or application owns this theme definition.
Homepage Header	Select a header definition to be used for classic homepages only.
Homepage Footer	Select a footer definition to be used for classic homepages only.
Target Page Header	Select a header definition to be used for classic pages only. **Note:** If no header is specified, then the header defined for homepages will also be used on target pages.
Macro Set	Select a *custom* macro set for this branding theme.

Classic Components

Note: For a cluster of applications managed by PeopleSoft Interaction Hub, the theme style sheet and any skin style sheets must exist in all systems.

Field or Control	*Description*
Theme Style Sheet	Select an override style sheet for the theme for classic components. **Note:** Theme style sheets can only be free form style sheets.
Name(Skin Style Sheets)	Identify one or more free form style sheets that can be made available to use with this theme as skin style sheets. The skin style sheets identified here are made available for use on the Assign Branding Themes page.
Description	Enter a description of the skin.

Fluid Components

Use the Fluid Components group box to register a base style sheet and theme style sheet pair for a specific node. Alternatively, you can use the Register Fluid Theme Style Sheet Mappings page to perform this registration, which will ensure that the pair is mapped correctly. See Generating Theme Style Sheets for Fluid Components. The rows in the Theme Style Sheet Mappings grid specify a mapping of a theme style sheet to a base style sheet only; the grid does not specify that this base style sheet is in use. The base style sheet is assigned at the system level on the Branding System Options page, or at the component or page level through PeopleCode. Then, when the base style sheet is loaded by the system into the generated HTML, any mapped theme style sheet is loaded immediately after it.

Note: For a cluster of applications managed by PeopleSoft Interaction Hub, the global override style sheet and any mapped theme style sheets for a node specification of *ANY* must exist in all systems.

Field or Control	*Description*
Node	Select the node to which the style sheet and theme style sheet is to be applied. You can select a specific node name, or one of the following values: • *ANY* – Applies to all nodes in all systems. (This setting is applicable to a cluster of applications managed by PeopleSoft Interaction Hub.) • *LOCAL_NODE* – Applies to all nodes of the current system only. **Note:** The list of specific nodes includes non-local PeopleSoft nodes that have a defined content URI.
Style Sheet	Select the base style sheet for fluid components for the selected node. **Note:** The lookup lists style sheets on the current system. For remote nodes, you must ensure that the style sheet you select also exists in the remote system.
Theme Style Sheet	Select a mapped style sheet that has already been generated from the selected base style sheet. **Note:** The lookup lists style sheets on the current system. For remote nodes, you must ensure that the style sheet you select also exists in the remote system.
Active	Select this option to make the mapped pair active.
Global Override Style Sheet	Select an override style sheet for the theme for all fluid components on all nodes.

Add to Project

Click the Add to Project link to access the Add To Project page to insert this branding theme definition into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) and Migrating Branding Data Using ADSs for more information.

Reverting Branding to Pre-8.62 Visuals

After upgrading to PeopleTools 8.62, you can revert the system branding to pre-8.62 visuals. The reversion mostly changes the colors to the PeopleTools 8.61 colors. Consider the following two scenarios before you revert the system branding to pre-8.62.

Scenario A: You've upgraded to PeopleTools 8.62, and you are using the delivered system branding theme.

In this scenario, to revert the system branding to pre-8.62:

1. Create a custom branding theme by cloning the delivered DEFAULT_THEME_FLUID theme.

   Refer to Assembling Branding Themes for instructions.
2. Continue by completing the steps in Scenario B.

Scenario B: You've upgraded to PeopleTools 8.62, and you are using a custom system branding theme.

In this scenario, to revert the system branding to pre-8.62:

1. Select **PeopleTools** > **Portal** > **Branding** > **Assemble Branding Themes**.
2. Select the custom system branding theme.
3. In the Classic Components group box, in the Skin Style Sheets grid, select *PT_CUSTOM_PROPS_PRE862* and enter a description.
4. In the Fluid Components group box, in the Theme Style Sheet Mappings grid, select the following:
   a. In the Node field, select *ANY*.
   b. In the Style Sheet field, select *PSSTYLEDEF_FMODE*.
   c. In the Theme Style Sheet field, select *PT_CUSTOM_PROPS_PRE862*.
   d. In the Active field, select the check box.
5. Save your changes.
6. Sign out and sign back in.

Reverting Quick Access Bar (QAB) on Homepages to PeopleTools 8.59

In PeopleTools 8.60, by default, the quick access bar (QAB) is available on the banner. However, you can revert QAB to pre-8.60 on fluid homepages, for example, if you want to brand the logo area and require additional space for your logo in the banner. When you revert QAB to pre-8.60, QAB appears on the left on fluid homepages.

To revert QAB to pre-8.60:

1. Select **PeopleTools** > **Personalization** > **Define System Personalizations**.
2. Select Search.
3. From the search results, select PPTL.

4. Select the Format tab.
5. In the User Option column, locate HPQABPNL.
6. Select the Set Option Default Value link and change the value from "No" to "Yes."

   See “Understanding Internally Controlled Options” (Security Administration).
7. Save your changes.
8. Sign out and sign back in.

   On all fluid homepages, QAB appears on the left as a vertical bar. On classic pages, fluid pages, and classic homepages, QAB appears on the banner.

Related Links

Assigning Branding Themes
Generating Theme Style Sheets for Fluid Components
“Using the Quick Access Bar” (Applications User’s Guide)
Understanding Branding Objects

Assigning Branding Themes

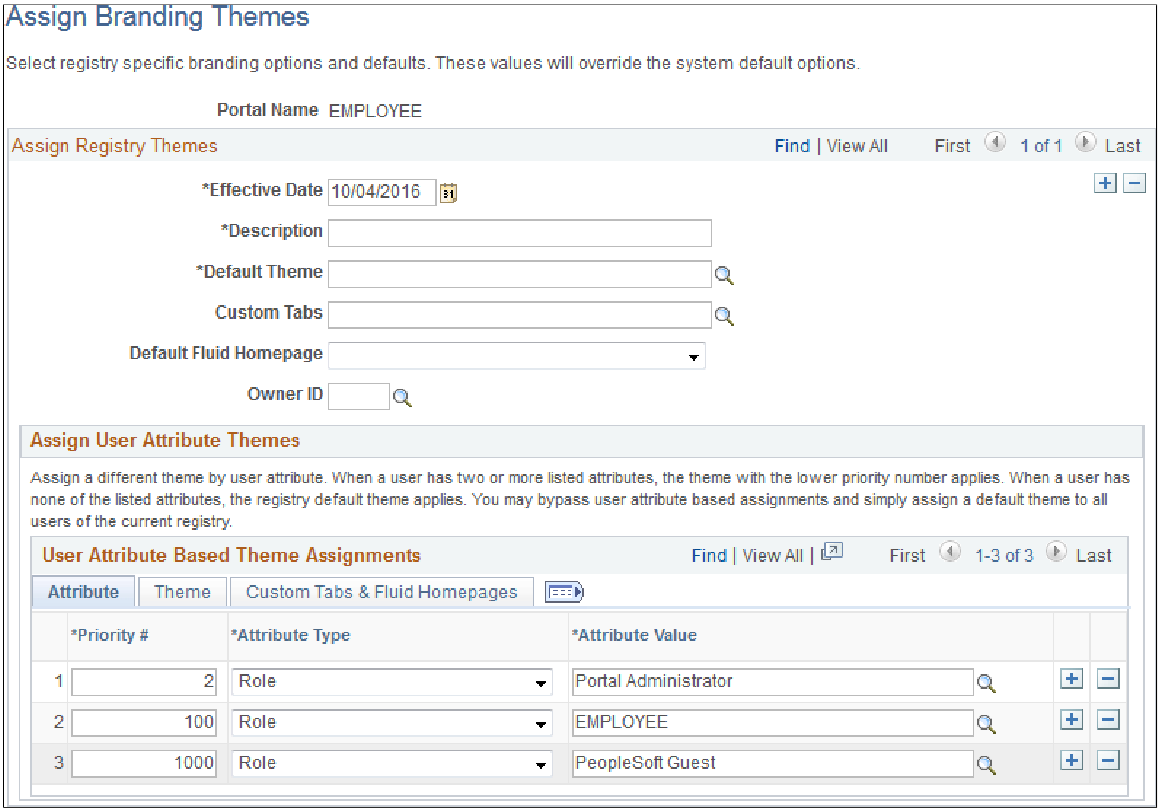
Use the Assign Branding Themes page to specify portal-specific branding options and defaults to override the system default settings. In addition, use this page to assign different themes, skins, custom tabs, and fluid homepages by user attribute.

Note: Once a portal-specific definition has been created, it cannot be entirely deleted. However, you can use effective dates to control whether a definition is currently in effect.

Navigation:

PeopleTools > **Portal** > **Branding** > **Assign Branding Themes**

This example illustrates the fields and controls on the Assign Branding Themes page.



Field or Control	Description
Portal Name	Displays the name of the current portal registry. **Note:** You can edit theme assignments for the current portal only. Therefore, sign into a different portal to modify its theme assignments.
Effective Date	Specify the effective date for this definition.
Label	Enter a label for the theme definition.
Default Theme	Select the default theme for this portal.
Default Skin	Select a default skin for all users who sign into this portal. **Note:** The list of available skins is specified on the Assemble Themes page.
Custom Tabs	Select the default custom tabs navigation collection for this portal. **Note:** Custom tabs are visible from classic homepages only.

Field or Control	*Description*
Default Custom Tab	Select a default classic homepage tab from the custom tabs navigation collection for all users who sign into this portal.
Default Fluid Homepage	Select a default fluid homepage for all users who sign into this portal. **Note:** If the Fluid Homepage field on the Web Profile Configuration - Look and Feel page specifies both a component for rendering fluid homepage tabs and a specific homepage, then any value set on this page will be overridden. See “Configuring Web Profile Settings for Fluid Applications” (Fluid User Interface Developer’s Guide) for more information.
Owner ID	Select which product, feature, or application owns this definition.

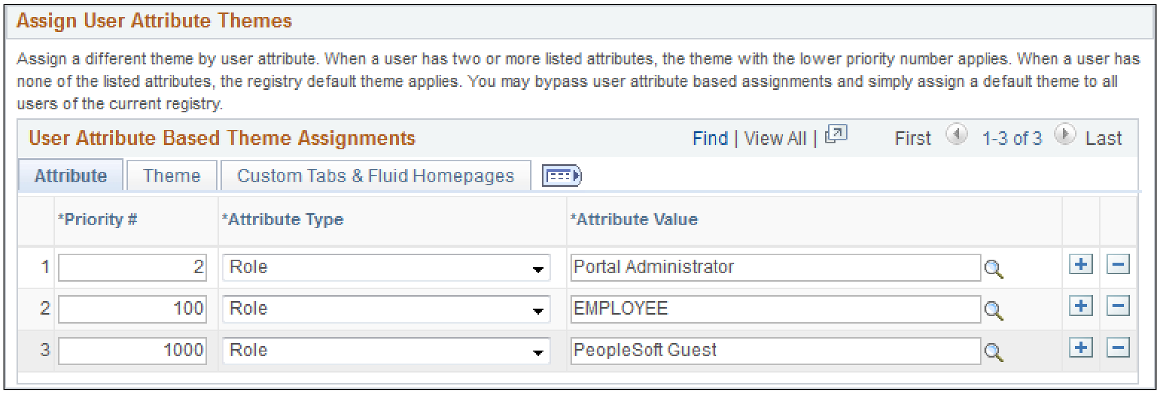
User Attribute Based Theme Assignments

Use the User Attribute Based Theme Assignments grid to override theme assignments, custom tabs, and default homepage tabs by user attribute. In PeopleTools, two user attribute types are delivered: by role or by permission list. You can use this grid to override the portal settings for the theme, skin, custom tabs collection, default custom tab, and default fluid homepage.

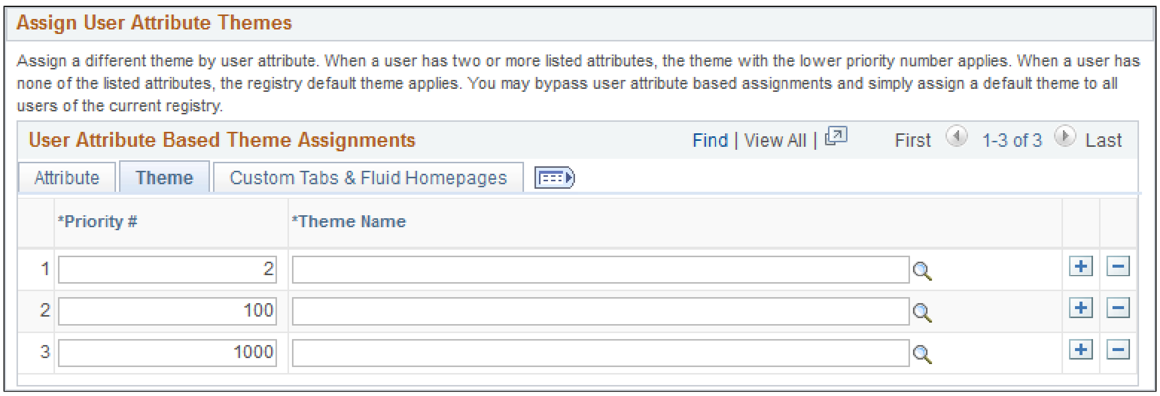
Important! Because user attribute based assignments override system-level and portal-level settings, you can create a custom user experience by user group. Moreover, you can set a classic custom tab as the default homepage for a group of users even when the system-wide personalization settings specify the fluid homepage as the default for all users, or vice versa. See the following section, “Setting the Default Homepage to be the Opposite of the System-Wide Setting,” for more information.

When you assign branding themes by user attribute, you also assign a priority sequence number to each attribute. The priority number dictates each attribute’s precedence in determining which theme will be displayed for each user. If a user has two or more attributes with a branding theme assignment, the attribute with the higher priority applies with *1* being the highest priority, *2* being the next highest priority, and so on. If a user has none of the attributes used in theme assignments, the portal-wide settings apply.

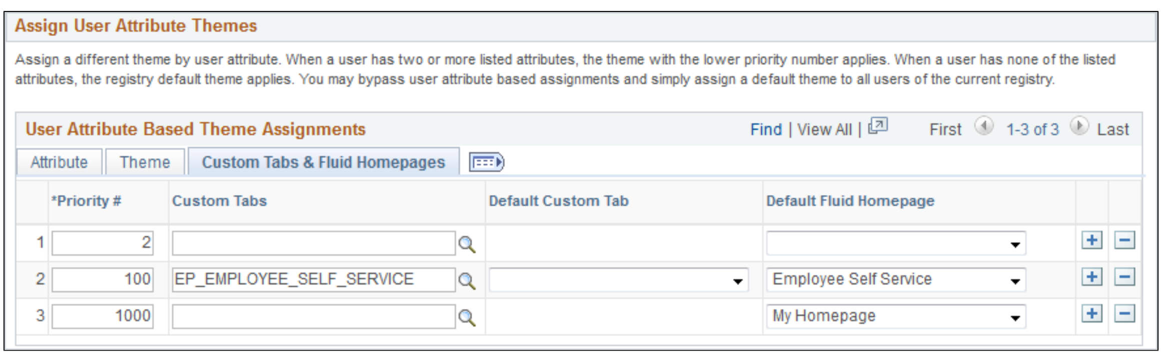
This example illustrates the fields and controls on the Attribute tab of the User Attribute Based Theme Assignments grid.



This example illustrates the fields and controls on the Theme tab of the User Attribute Based Theme Assignments grid.



This example illustrates the fields and controls on the Custom Tabs & Fluid Homepages tab of the User Attribute Based Theme Assignments grid.



Field or Control	*Description*
Priority #	Specify the priority for this attribute with *1* being the highest priority, *2* being the next highest priority, and so on.

Field or Control	Description
Attribute Type	Specify the attribute type. The PeopleTools-delivered values are: • *Role* • *Permission List*
Attribute Value	Select a value for the selected attribute type.
Theme Name	Select the theme for this attribute.
Custom Tabs	Select a custom tabs navigation collection for this attribute. **Note:** Custom tabs are visible from classic homepages only.
Default Custom Tab	Select a default classic homepage tab from the custom tabs navigation collection for this attribute.
Default Fluid Homepage	Select a default fluid homepage for this attribute. **Note:** If the Fluid Homepage field on the Web Profile Configuration - Look and Feel page specifies both a component for rendering fluid homepage tabs and a specific homepage, then any value set on this page will be overridden. See “Configuring Web Profile Settings for Fluid Applications” (Fluid User Interface Developer’s Guide) for more information.

Add to Project

Click the Add to Project link to access the Add To Project page to insert this portal-specific branding definition into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) and Migrating Branding Data Using ADSs for more information.

Setting the Default Homepage to be the Opposite of the System-Wide Setting

System-wide personalization settings specify the default homepage type (fluid or classic) for all users. Fluid is set as the default homepage type for desktop and laptop (HPPC), and for tablet users (HPTABLET). (See “Setting Device Homepage Defaults” (Fluid User Interface Developer’s Guide) for more information on these system-wide settings.)

Note: The default homepage type for smartphone users is always a fluid homepage, which is not affected by the system-wide homepage settings.

To set the default homepage to be the opposite of the system-wide setting for a specific group of users:

1. Create a navigation collection for custom homepage tabs.

   For example, when the system-wide personalization settings specify fluid as the default homepage type, include classic content references in the navigation collection.

2. Create a portal-specific branding theme assignment using the Assign Branding Themes page.

3. Create a user attribute based assignment in the User Attribute Based Theme Assignments grid:

   a. Specify the user group by role or permission list.

   b. Specify the custom homepage tabs navigation collection.

   c. Select a specific item from this navigation collection as the default custom tab.

      For example, select one of the classic content references.

      Important! Selecting a default custom tab is an essential step. Otherwise, users will still see the default homepage type as specified by the system-wide settings.

Note: When a classic content reference is selected as the default custom tab here or through user personalization, the user must add the Homepages tile to the NavBar in order to be able to navigate to any fluid homepages.

Related Links

Managing Homepages

Assembling Branding Themes

Creating and Maintaining Navigation Collections for Custom Homepage Tabs

Applying Branding to Other Objects

This topic discusses how to:

- Brand classic components.
- Maintain pagelet branding attributes.

Branding Classic Components

Important! The Component Branding page can be used with classic components only. If you attempt to add a value for a fluid component, an error message is displayed.

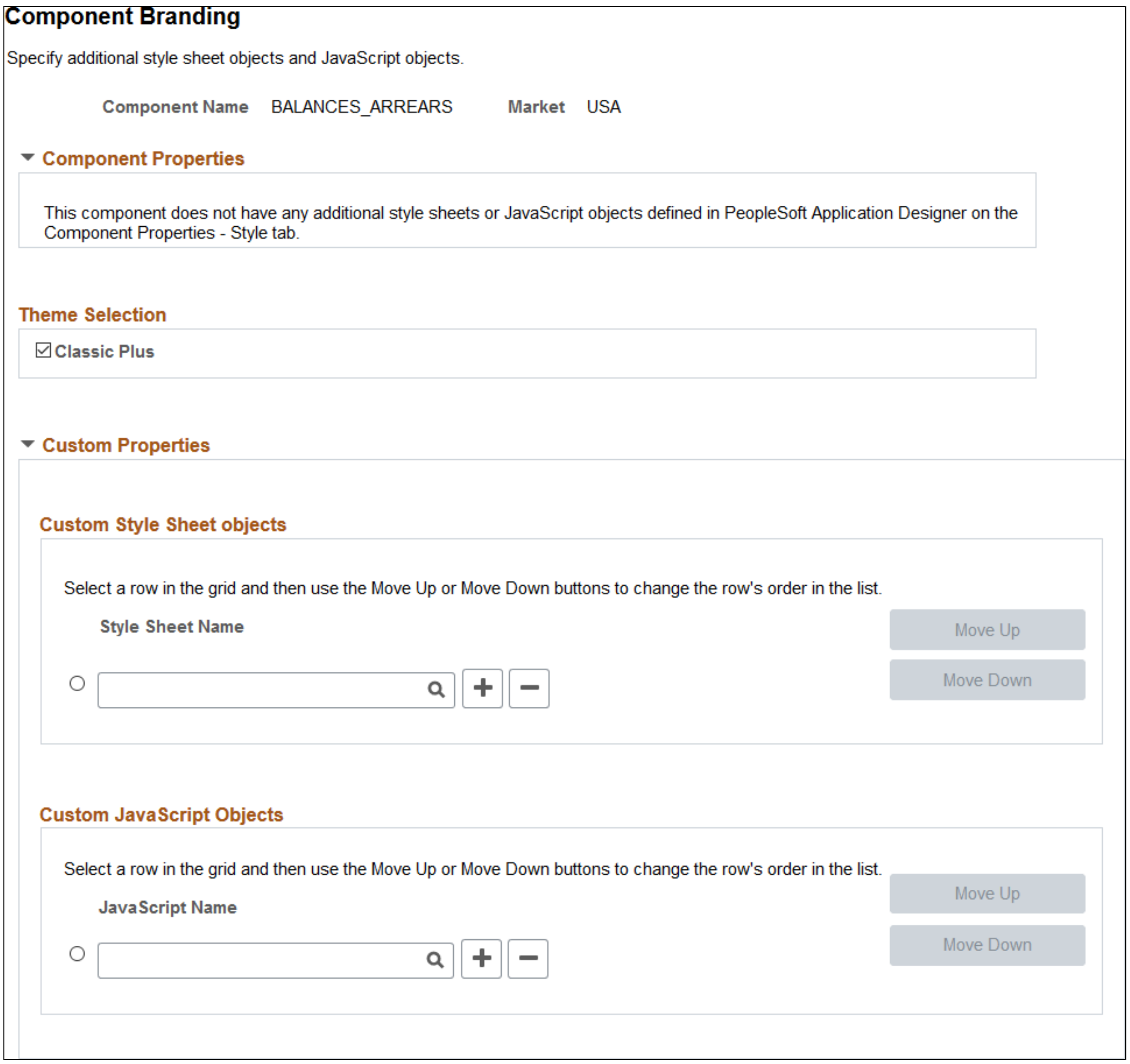
Use the Component Branding page to specify custom style sheet objects and JavaScript objects in addition to those specified by the component’s developer in Application Designer.

Important! If the current component has an associated component interface definition, then modifying the component definition by adding, deleting, or reordering style sheets or JavaScript objects will invalidate any associated component interface definition. A warning message will be issued when you save such a component on the Component Branding page. Therefore, you must also open and re-validate the associated component interface definitions in Application Designer.

Navigation:

PeopleTools > **Portal** > **Branding** > **Component Branding**

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



Field or Control	*Description*
Component Name	Displays the name of the selected component.
Market	Displays the market for the selected component.

Field or Control	***Description***
Component Properties	The Component Properties group box displays the additional style sheet objects and JavaScript objects specified by the component's developer in Application Designer. These values are display-only, and cannot be modified here.
Theme Selection - Classic Plus	This option is selected when a classic component has been configured to display with fluid-like styling (which is also referred to as "classic plus"). For more information, see PeopleTools Components Delivered as Classic Plus. **Note:** While this option can be selected or deselected on this page, the best practice is to set the option on the Style tab of the Component Properties dialog box in Application Designer only after reworking each page in the component for fluid-like styling. See "Reworking Components for Classic Plus Display" (Application Designer Developer's Guide).
Custom Properties	The Custom Properties group box displays the additional, custom style sheet objects and JavaScript objects specified by the portal administrator through the use of this Component Branding page.
Style Sheet Name	Specify zero, one, or more custom style sheets to be loaded for this component.
JavaScript Name	Specify zero, one, or more custom JavaScript objects to be loaded for this component. **Important!** JavaScript objects are stored as HTML objects in the database. The lookup does not differentiate between standard HTML objects and JavaScript objects. Therefore, it is your responsibility to ensure that you have selected valid JavaScript to be loaded with a component.
Move UpMove Down	Select the row designating a style sheet or a JavaScript object, and then use the Move Up and Move Down buttons to change the order in which the objects are loaded with the component.

Related Links

"Setting Style Properties" (Application Designer Developer's Guide)

"Validating Component Interfaces" (Component Interfaces)

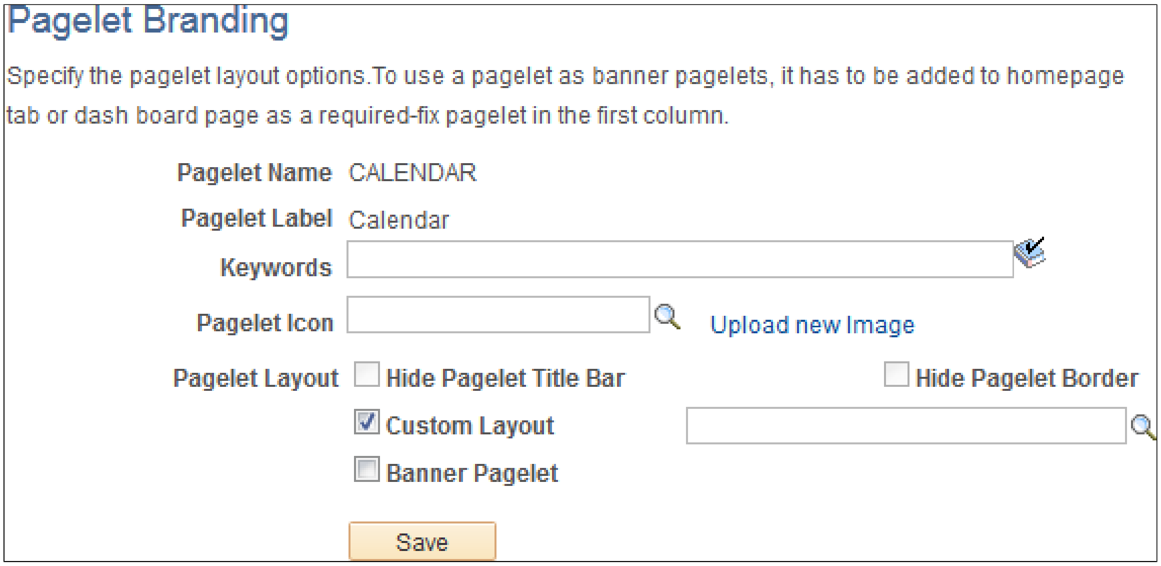
Maintaining Pagelet Branding Attributes

Use the Pagelet Branding page to specify pagelet title bar, border, and layout options.

Navigation:

PeopleTools > **Portal** > **Branding** > **Pagelet Branding**

This example illustrates the fields and controls on the Pagelet Branding page.



Note: If a pagelet is already in use on a homepage tab or dashboard page, clear the application server cache so that these branding changes become visible to all users. Alternatively, for testing purposes, you can remove the pagelet from the homepage or dashboard page and then add it back to view the changes on a per-user basis.

Field or Control	***Description***
Pagelet Name	Displays the ID of the selected pagelet.
Pagelet Label	Displays the label of the selected pagelet.
Keywords	Enter a keyword or keyword phrase to be stored as a content reference attribute. See Searching for and Importing Pagelets and Tiles for more information on how keywords are used to search for pagelet definitions. **Note:** Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language).

Field or Control	***Description***
Pagelet Icon	Displays the ID of the pagelet icon if one has been selected for this pagelet. If an icon has been selected, then the image itself displays below this field. **Note:** Oracle recommends image sizes from 16x16 pixels to 32x32 pixels for use as pagelet icons. Click the Look Up Pagelet Icon button to select an image from the database. Clear the pagelet icon field to display a pagelet without an icon. The PT_PORTAL_PGLT_HDR_IMG content reference attribute is created to store the image ID.
Upload new image	Click this link to access the Add Image Object page to upload a new image file. See Adding and Editing Images.
Hide Pagelet Title Bar	Select this option to hide the pagelet title bar. Hiding the title bar hides the pagelet's label and all user interaction buttons to personalize, minimize, and remove the pagelet, or to get pagelet help. **Note:** When this option is selected for a pagelet, users will not be able to use drag-and-drop to reposition the pagelet. The Personalize Layout page must be used instead. The PORTAL_HP_COMPONENT content reference attribute is created to store this setting.
Hide Pagelet Border	Select this option to hide the pagelet's borders. The PORTAL_HP_COMPONENT content reference attribute is created to store this setting.
Custom Layout	Select the Custom Layout check box to enable selection of a custom HTML layout object for this pagelet. Once this check box is selected, an additional unlabeled field and a lookup button are revealed allowing you to select an HTML object from the database. **Note:** When you select the Custom Layout check box, both the Hide Pagelet Title Bar and Hide Pagelet Border check boxes are cleared (if selected) and disabled. If this check box is itself cleared, then these check boxes are re-enabled; however, their former settings are not restored.

Field or Control	Description
Banner Pagelet	Select this option to identify this pagelet as one that can be used as a banner pagelet. **Note:** To use a pagelet as a banner pagelet, it has to be added to a homepage tab or dashboard page as a required-fixed pagelet in the first column. The PORTAL_BANNER_PGLT content reference attribute is created to store this setting.

Related Links

Defining Content References

Additional Branding Tasks

This topic provides an overview of navigation collection used to create custom homepage tabs and discusses how to:

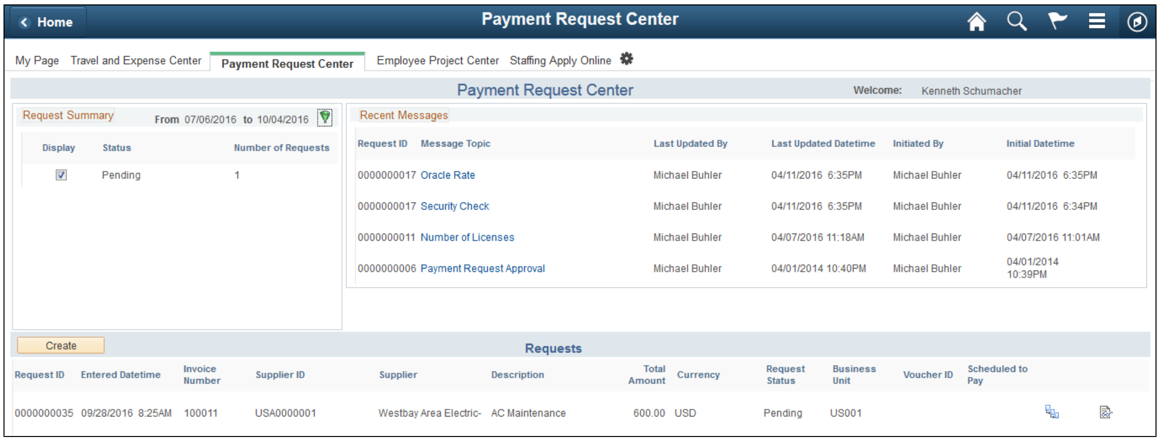
- Create and maintain navigation collections for custom homepage tabs.
- Generate theme style sheets for fluid components.
- Create new theme style sheets and skins.
- Use additional style sheets to inject your custom branding.
- Create a new branding element.
- Create a new user attribute type.
- Migrate branding data using application data sets (ADS).

Understanding Navigation Collections Used to Create Custom Homepage Tabs

PeopleTools allows you to use specially designed navigation collections as definitions for custom homepage tabs. These navigation collections allow you to create links to local or remote content references that are then rendered as homepage tabs, and which appear after standard homepage tabs beneath the portal header. Because simple content references can be added to these collections, you can now designate any content reference or WorkCenter as a homepage tab. The requirement for creating such a “custom tabs” navigation collection is that the links must reside in the root folder of the navigation collection; links in subfolders will not be displayed as homepage tabs.

Once a navigation collection has been created, it can be designated for use as homepage tabs in the Custom Tabs field on the Branding System Options page or on the Assign Branding Themes page.

The following example displays a homepage that includes four custom homepage tabs: Travel and Expense Center, Payment Request Center, Employee Project Center, and Staffing Apply Online. These tabs have been defined as custom homepage tabs through the use of a navigation collection. These tabs are displayed after the default homepage tab: My Page.



Note: To avoid forcing users to scroll horizontally on the homepage, Oracle recommends that you configure and display no more than seven standard or custom homepage tabs.

Using a navigation collection to designate homepage tabs also provides you with the following benefits with respect to remote content:

- Security is checked in real time against definitions in the content provider system.
- You do not need to manually import content references or security from the content provider system
- If the remote system is down or has not been configured, the homepage tab is not displayed.

Note: Remote items require additional processing whenever a user signs in or the cache expires to check security on the remote content provider system. In addition, the Navigation Collections component only supports adding remote component-based content references. To add other remote items to a navigation collection, you must manually create the content reference through the Structure and Content component.

Note: PeopleSoft Interaction Hub 9.1 Rev 2 (and above) is delivered with navigation collection named Custom Tabs, which includes 12 pre-defined links as sample data. See the documentation for PeopleSoft Interaction Hub for more information on this Custom Tabs navigation collection.

Related Links

Administering System Branding
Assigning Branding Themes

Creating and Maintaining Navigation Collections for Custom Homepage Tabs

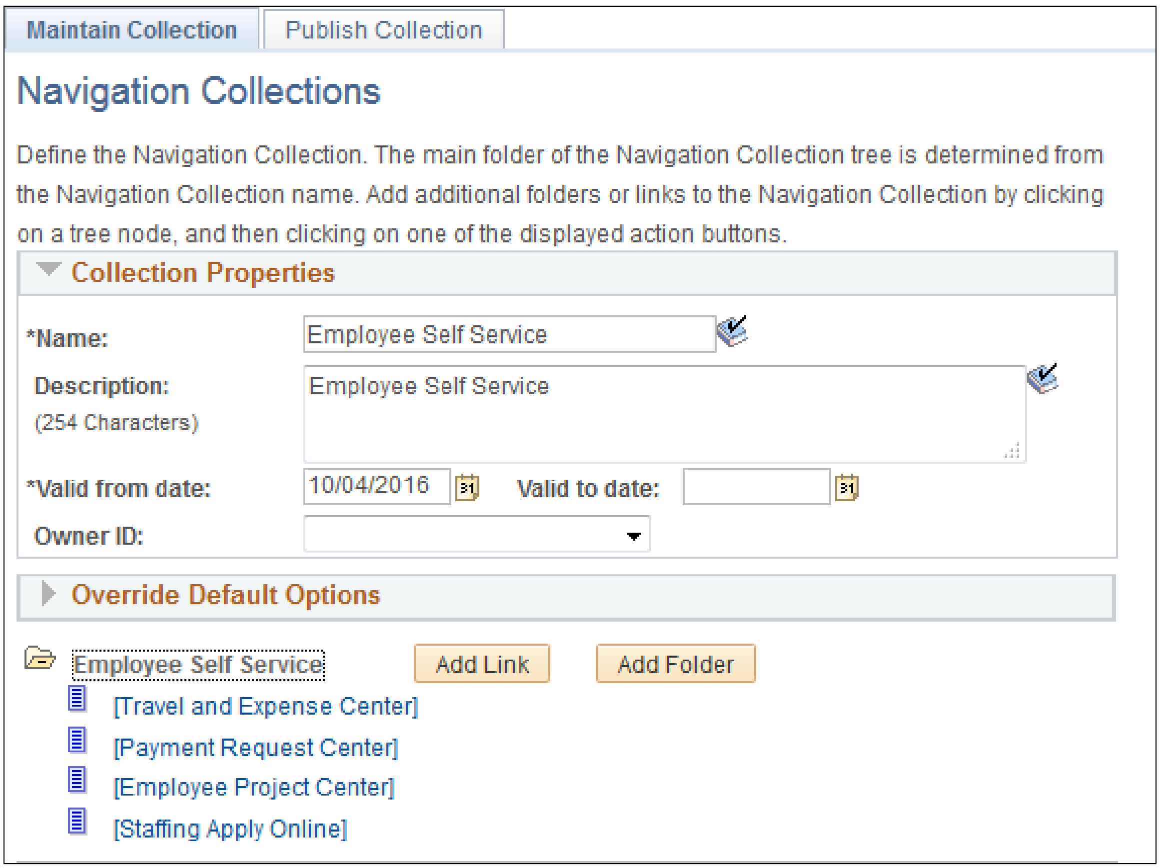
Use the Navigation Collections page to create and maintain links in any custom tabs navigation collection.

Navigation:

1. **PeopleTools** > **Portal** > **Portal Utilities** > **Manage Navigation Collections**
2. Do one of the following:
   - Click the Add Collection link to create a new navigation collection.
   - Click the Search button and select an existing navigation collection to maintain.

This example illustrates the fields and controls on the Navigation Collections page for a navigation collection that will be used to display custom homepage tabs.

Note: This Employee Self Service navigation collection was used to generate the custom tabs shown in the previous example.



To serve as a custom homepage tab, a link must reside in the root folder of the navigation collection, and not in a subfolder.

To maintain a custom tabs navigation collection:

- Click the Add Link button to define a new local or remote link.
- Select an existing link and click the Edit Link button to edit an existing link.
- Select an existing link and click the Delete Link button to delete an existing link.

Important! Do not edit the properties for the Custom Tabs navigation collection delivered with PeopleSoft Interaction Hub.

Field or Control	*Description*
Name	Enter a name for the navigation collection.
Description	Enter a description for this navigation collection.
Valid from date	Select the date on which you want this navigation collection to become active.
Valid to date	Select the date on which you want this navigation collection to become inactive.
Owner ID	Set the default owner ID for new definitions you create. Alternatively, leave the option blank, which is the default, to allow users to select the owner ID they want. The Owner ID helps identify which applications team last made a change to a definition.
Add Link	Click to access the Add Link page to define a link for a new custom tab. On the Add Link page, you can create a link to a local menu item or to a remote component based content reference. See Adding a Link, Selecting a Local Link, and Selecting a Remote Link for more information on creating local and remote links in navigation collections. **Note:** Do not select the **Open in a new window** check box. If you need to create a remote link to a non-component based content reference, then you will need to edit the custom tabs navigation collection through Structure and Content. See “Creating a Content Reference to a Remote, Non-Component-Based Item” for more information.
Add Folder	Click to add a subfolder to the navigation collection. **Important!** Links in subfolders are not supported for custom tabs navigation collection. Do not create a subfolder and links within the subfolder.
Edit Link	Click to access the Edit Link page for a link.
Delete Link	Click to access the Confirmation page to delete or cancel the deletion.

Maintaining Content References in a Custom Tabs Navigation Collection

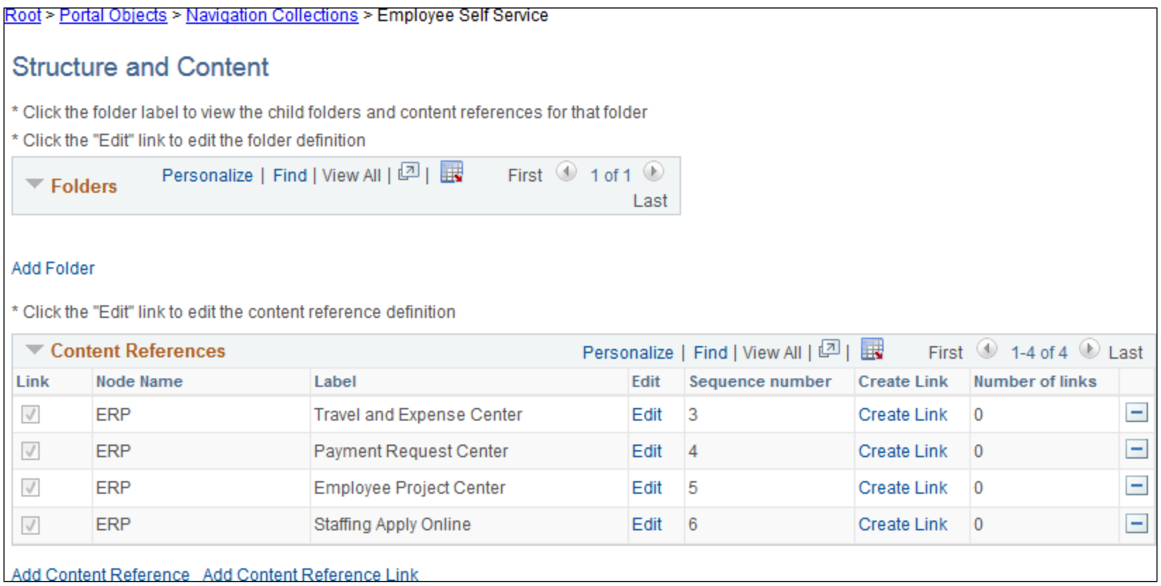
You can use the Structure and Content component to maintain content references in a custom tabs navigation collection. This will allow you to:

- Disable a homepage tab by selecting the hide from portal navigation attribute.
- Enable the display of a homepage tab (for example, one of the tabs delivered as sample data) by deselecting the hide from portal navigation attribute.
- Manually create a content reference to a remote non-component-based item.

To access a custom tabs navigation collection in the Structure and Content component:

1. Select **PeopleTools** > **Portal** > **Structure and Content**.
2. Click the Portal Objects link.
3. Click the Navigation Collections link.
4. Click the link for the custom tabs navigation collection.

This example illustrates the fields and controls on the Structure and Content page for a custom tabs navigation collection.



To disable a custom homepage tab, click the Edit link for that content reference. Then, select the Hide from portal navigation option and save the definition.

To enable a custom homepage tab, click the Edit link for that content reference. Then, deselect the Hide from portal navigation option and save the definition.

Creating a Content Reference to a Remote, Non-Component-Based Item

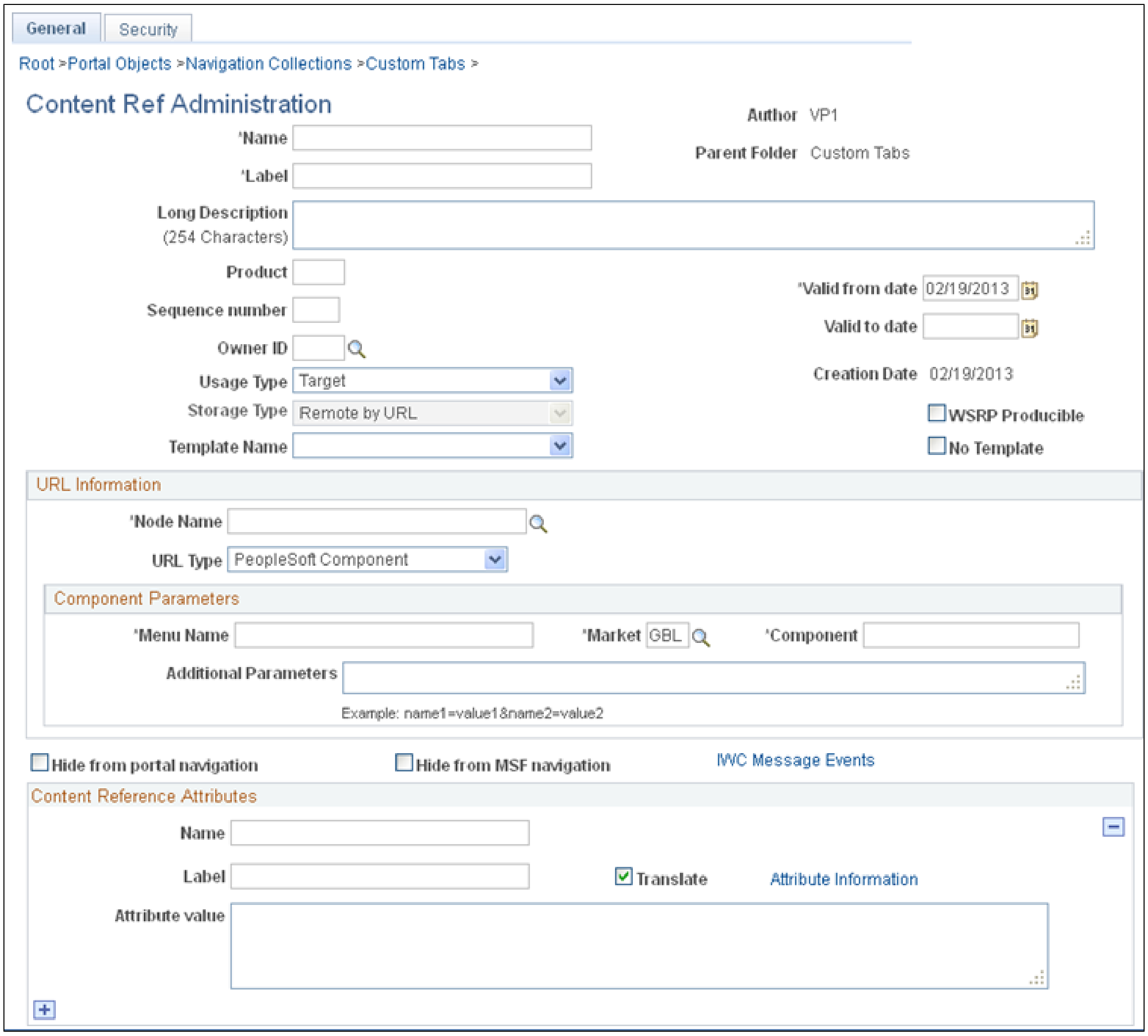
---

Note: Remote, non-component based content (such as a remote dashboard) must be accessible through unified navigation from the PeopleSoft Interaction Hub system before it can be designated as a homepage tab.

To manually create a content reference to a remote, non-component-based item such as a remote dashboard to be used as a custom homepage tab:

1. On the Structure and Content page click the link labeled Add Content Reference.

This example illustrates the fields and controls on the Content Ref Administration page.



2. Specify the node name of the portal host node, such as HRMS, ERP, or LOCAL_NODE.

Note: For a remote dashboard, the Node Name field can specify LOCAL_NODE or can be left blank.

3. In the URL Type field, select one of the following:

   - PeopleSoft Generic URL – For remote content from a PeopleSoft application.
   - Non-PeopleSoft URL – For remote content from a non-PeopleSoft system, such as an external web site.

4. In the Portal URL field, enter a partial or full URL depending on the URL type:

   - For a PeopleSoft Generic URL, navigate to the remote item through the menu from the local system. Copy the full URL. Then, remove the server, port, servlet, site, portal, and node from the URL, leaving you with a partial URL string.

For example, when navigating using unified navigation from a local PeopleSoft Interaction Hub system, a partial URL string would be similar to the following:

```
h/?tab=REMOTEUNIFIEDDASHBOARD&unifieddashboard=y&remotedburl=http%3a%2f%2⇒
fserver%3aport%2fpsp%2fps%2fEMPLOYEE%2fHRMS%2fh%2f%3ftab%3dHC_HRS_RECRUIT⇒
ING_HOMEPAGE_TAB%26pslnkid%3dHC_HRS_RECRUITING_HOMEPAGE_LNK&uninavpath=Re⇒
cruiting%20Dashboard{HC_HRS_RECRUITING_HOMEPAGE_TAB}
```

For example, a partial URL string for a remote SmartNavigation folder would be similar to the following:

```
c/ROLE_EMPLOYEE.HRCD_CO_DIRECTORY.GBL?NAVSTACK=Clear&pt_fname=HC_COMPANY_⇒
DIRECTORY&isleaf=false&abnpcode=WEBLIB_HRCD.ISCRIPT1.FieldFormula.IScript⇒
_setFolderFromTree&abnds=t&TREE_NAME=COMPANY_DIRECTORY&TREE_SETID=&TREE_U⇒
SERKEY=&TREE_EFFDT=&TREE_BRANCH=&abnnode=&cmd=uninav&unabnfldr=false&unin⇒
avabnpcode=
```

- For a Non-PeopleSoft URL, copy the full URL from the browser and paste it into the Portal URL field. For example:

  ```
  http://server.example.com/content/my_info.html
  ```

  Note: Alternatively, you can use the Node Definitions page to create a node definition for the non-PeopleSoft system, which would include the definition of the Content URI Text field—for example, `http://server.example.com/`. If you identify that non-PeopleSoft node through the NODE attribute as indicated in step 5, then you can specify a partial URL—for example, `content/my_info.html`. See Defining Portal Nodes for more information.

5. Define the following four content reference attributes to link the content reference to a remote menu item and enable real-time security checks:

Name	***Attribute Value***
CREFID	The name of the remote CREF as defined in the content provider system.
LOCATION	REMOTE
NODE	The portal host node name of the remote node.
PORTAL	The name of remote portal. This must be the same name as the current portal in the local system—for example, EMPLOYEE.

6. Save the content reference definition.

Note: The Custom Tabs navigation collection delivered with PeopleSoft Interaction Hub 9.1 Rev 2 (and above) includes definitions for several remote, non-component based items that can be used as examples, including the Recruiting Dashboard and the Manager Dashboard, among others.

Related Links

Understanding Navigation Collections Used to Create Custom Homepage Tabs

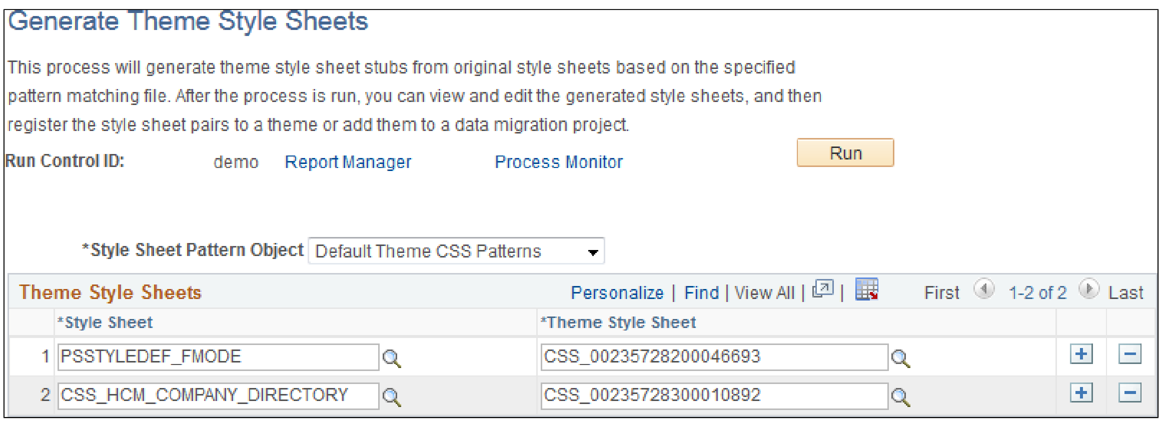
Generating Theme Style Sheets for Fluid Components

Use the Generate Theme Style Sheets page to create and maintain theme style sheet stubs for fluid components from original style sheets based on the specified pattern matching file.

Navigation:

PeopleTools > **Portal** > **Branding** > **Branding Theme Style Sheets**

This example illustrates the fields and controls on the Generate Theme Style Sheets page before process execution.

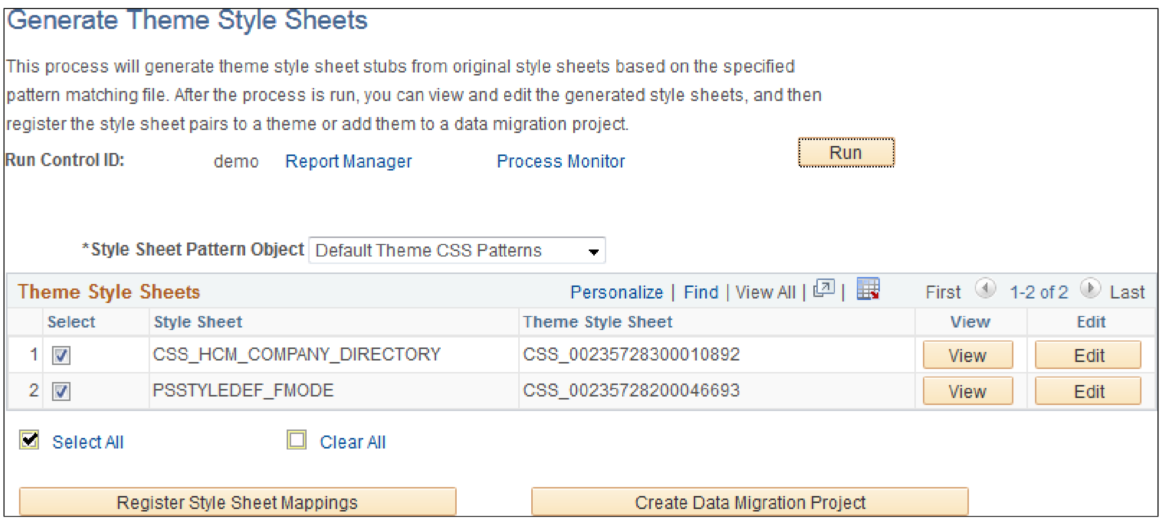


Field or Control	***Description***
Style Sheet Pattern Object	Select a style sheet pattern object. Currently, PeopleTools delivers one style sheet pattern object, Default Theme CSS Patterns, which will leave color attributes only in the generated style sheets. **Note:** By definition, a style sheet pattern object is an HTML definition that begins with the PTBR_CSSPTN_ prefix.
Style Sheet	Select an original style sheet from which the mapped theme style sheet will be generated. When the run control ID is first created, the system default style sheet for fluid components defined on the Branding System Options page will be added to the list. If none is defined, the PSSTYLEDEF_FMODE style sheet is added to the list.

Field or Control	*Description*
Theme Style Sheet	**Note:** A unique, random name is generated initially when a style sheet is selected. Enter a unique name for the mapped theme style sheet to be generated. **Note:** You can use the prompt to look up style sheet objects already defined in the current system to avoid using a non-unique name.

Click the Run button to execute the process. After process execution, additional options are displayed on this page.

This example illustrates the fields and controls on the Generate Theme Style Sheets page after process execution.



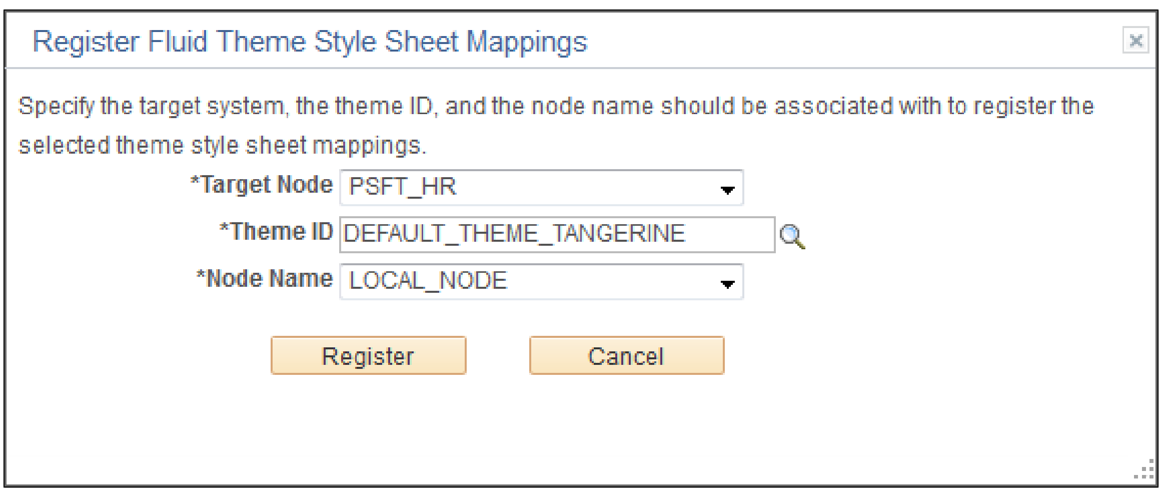
Field or Control	*Description*
SelectSelect AllClear All	Use the Select, Select All, and Clear All check boxes to select or deselect items for further processing.
View	Click the View button to view the content of the generated theme style sheet. **Note:** This button is displayed only for users who have read permissions on the style sheet object.

Field or Control	***Description***
Edit	Click the Edit button to edit the content of the generated theme style sheet. See Adding and Editing Branding Objects for more information. **Note:** This button is displayed only for users who have both read and write permissions on the style sheet object.
Register Style Sheet Mappings	Click this button to register the selected style sheet pairs to a theme definition. At least one base style sheet/theme style sheet pair must be selected.
Create Data Migration Project	Click this button to add the selected theme style sheets and all images they use to a Data Migration Workbench project for migration to a different database. At least one base style sheet/theme style sheet pair must be selected. **Note:** A Data Migration Project Workbench project is also referred to as an Application Data Set (ADS).

Register Fluid Theme Style Sheet Mappings Page

Use the Register Fluid Theme Style Sheet Mappings page to register a mapped theme style sheet to the specified theme definition. This page provides an alternative to manually selecting the style sheet pair on the Assemble Themes page.

This example illustrates the fields and controls on the Register Fluid Theme Style Sheet Mappings page.



Field or Control	***Description***
Target Node	Select the default local node to identify the *target system.*
Theme ID	Select the theme definition.

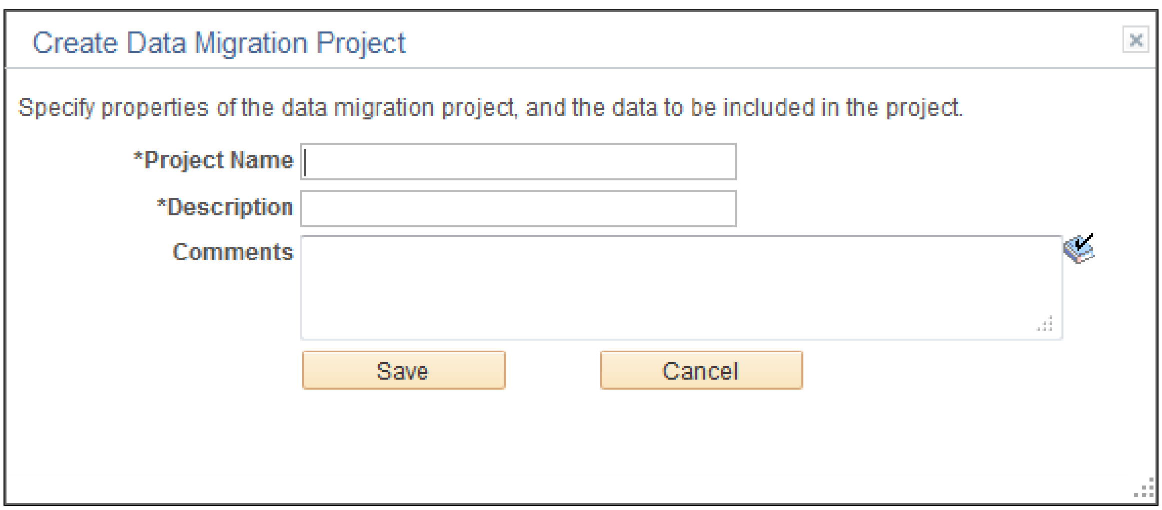
Field or Control	Description
Node Name	Select the node to which the style sheet and theme style sheet is to be applied. You can select a specific node name, or one of the following values: • *ANY* – Applies to all nodes in all systems. (This setting is applicable to a cluster of applications managed by PeopleSoft Interaction Hub.) • *LOCAL_NODE* – Applies to all nodes of the current system only. **Note:** The list of specific nodes includes non-local PeopleSoft nodes that have a defined content URI.

Create Data Migration Project Page

Use the Create Data Migration Project page to add the selected theme style sheets and all images they use to a Data Migration Workbench project. You can use this project to migrate all of the theme style sheets and images to a different system—for example, if you generate the theme style sheets on one system and then must distribute and register these style sheets on other systems in a cluster.

Once the project is created, you will be given the opportunity to open the project in Data Migration Workbench.

This example illustrates the fields and controls on the Create Data Migration Project page.



Field or Control	Description
Project Name	Enter a name for the data migration project.
Description	Enter a description for the data migration project.

Field or Control	Description
Comments	Enter any additional comments that will appear with the project in Data Migration Workbench.

Related Links

Assembling Branding Themes

"Defining ADS Project" (Lifecycle Management Guide)

Creating New Theme Style Sheets and Skins

For classic components, to create a new theme style sheet or skin:

1. In Application Designer, clone the DEFAULT_THEME_TANGERINE_ALT free form style sheet.

   The DEFAULT_THEME_TANGERINE_ALT contains style class definitions that override the style class definitions for the portal header found in PSSTYLEDEF_TANGERINE. The style class overrides in DEFAULT_THEME_TANGERINE_ALT are grouped as follows:

   - Header container
   - Logo
   - Header links
   - NavBar links
   - Global search widget
   - Hover menu
   - Menu icons

2. Modify one or more these override style classes in the new style sheet. Alternatively, you can delete style class overrides or an entire section to retain the settings from the PSSTYLEDEF_TANGERINE style sheet. See "Changing Font and Color Settings in Free Form Sub Style Sheets" (Application Designer Developer's Guide).

3. Add additional override style class definitions to modify the look of pagelets or other items on your site. The following example provides free form style sheet overrides for pagelets:

```
/********************
 * Pagelets
 ********************/
#ptpglts .pthpli {
  margin-bottom: 10px;
}

#ptcol0 .PTPAGELETHEADER,
#ptpglts .PTPAGELETHEADER {
  background: url(%Image(MY_THEME_PAGELET_HEADER_BG)) repeat-x scroll 0px 0px ⇒

#78A4CB;
  border-color: #78A4CB;
```



```
  border-radius: 4px 4px 0 0;
  padding: 4px;
}

#ptcol0 .PTPAGELETHEADER .ptpgltlabel,
#ptpglts .PTPAGELETHEADER .ptpgltlabel {
  color: #333333;
  font-size: 12pt;
  font-weight: bold;
  font-family: Tahoma,Arial,Helvetica,Geneva,sans-serif;
}

#ptcol0 .PTPAGELETBODY,
#ptpglts .PTPAGELETBODY {
  border-color: #78A4CB;
  border-radius: 0 0 4px 4px;
  border-style: none solid solid solid;
  border-width: 1px;
  background-color: #FFFFFF;
}

/* Banner pagelets */
#ptcol0 .PTPAGELET {
  padding-%AlignStart: 4px;
  padding-%AlignEnd: 4px;
}

#ptcol0 .ptbannerpglt2 .PTPAGELETBODY {
  border-radius: 4px;
  border-top-style: solid;
}

#ptcol0 .ptbannerpglt3 .PTPAGELETBODY {
  border-radius: 4px;
  border-style: none;
}
```

4. To use the new style sheet:

   - Create a new theme template and select the new style sheet as the theme template for classic components.
   - Modify an existing theme template and add the new style sheet as an available skin for classic components.

   See Assembling Branding Themes.

For fluid components, to create a new theme style sheet:

1. Generate a mapped theme style sheet. See Generating Theme Style Sheets for Fluid Components.
2. Edit the new style sheet to modify one or more style class definitions.
3. Map the new theme style sheet to its base style sheet for one or more nodes for use with fluid components.
4. Assign the base style sheet as the default system style sheet for fluid components, or assign the base style sheet at the component or page level through PeopleCode.

Using Additional Style Sheets to Inject Your Custom Branding

To use additional style sheets to inject your custom branding:

1. Develop custom style sheets, custom JavaScript objects, or both to implement your custom branding. See Creating and Maintaining Style Sheet Objects and Creating and Maintaining JavaScript Objects.
2. Use these custom definitions by:
   - Applying them to classic components system-wide on the Branding System Objects page. See Configuring Branding System Options.
   - Applying them to individual components on the Component Branding page. See Branding Classic Components.

Example

The following provides and example of how you can use a free form style sheet to modify the display certain page elements and to invoke custom JavaScript to provide additional formatting:

1. Upload a new free form style sheet object. For example, upload the following as MY_BRANDING_DEMO:

```
/* Page Titles */
.PAPAGETITLE, .PSSRCHTITLE, .PSQRYTITLE {
  color: #FF0000;
}

/* Grid Border */
.PSLEVEL1GRIDWBO, .PSLEVEL2GRIDWBO, .PSLEVEL3GRIDWBO {
  border-color: #FF0000;
}

/* Group Box Border */
.PSGROUPBOXWBO {
  border-color: #FF0000;
}

/* Highlighter Active Field */
.FIELD_HIGHLIGHTER {
  border-color: #FF0000;
  background-color: #D2D2D2;
}
```

2. Upload a new JavaScript object. For example, upload the following as FIELD_HIGHLIGHTER:

```
(function($) {
  $(document).ready(function(){
    $(document).delegate("input[type='text'], input[type='password'], " +
        "select, textarea", "focus", function(e) {
      // Add a style class when activated
      $(this).addClass("FIELD_HIGHLIGHTER");
    }).delegate("input[type='text'], input[type='password'], " +
        "select, textarea", "blur", function(e) {
      // Remove a style class when not active
      $(this).removeClass("FIELD_HIGHLIGHTER");
    });
  });
})(jQuery);
```

3. To inject this custom branding system wide, select **PeopleTools** > **Portal** > **Branding** > **Branding System Options**.
4. In the Additional Style Sheet Objects group box, add the MY_BRANDING_DEMO style sheet.

5. In the Additional Javascript Objects group box, add this JavaScript object first: PT_JQUERY_1_6_2_JS.
6. Add a new row and add your new JavaScript as a second JavaScript object: FIELD_HIGHLIGHTER.
7. Save your changes.
8. Navigate to any page that includes a page title, group box, and grid. For example, select the User Profile - Roles page, or the Assemble Themes page. Search and select value.

   Notice the style changes to the page title, group box borders, and grid borders introduced by the additional style sheet.
9. Place the cursor in any edit box or long edit box.

   Notice the highlighting changes to the field introduced by the additional JavaScript object, which calls jQuery to perform the dynamic highlighting.

Creating a New Branding Element

To create a new branding element:

1. Create an application class definition that extends the PTBR_BRANDING:Elements:BaseElement class.

   Use the PTBR_BRANDING:Elements:Basic* class definitions as examples.
2. Implement the following methods in the new class:
   - The class constructor to initialize the object. If the class includes additional attributes, ensure that the constructor also sets an initial value for these attributes.

     ```
     MyElement(ID as string)
     ```
   - clone to create an exact copy of the current object.

     ```
     clone() Returns
     PTBR_BRANDING:Elements:BaseElement
     ```
   - getHTML to return an HTML code fragment representing the object.

     ```
     getHTML(preview as boolean)
     Returns string
     ```
3. Optionally, implement the getStyleDefinitions method to return style definitions for the object as an in-line style definition block or an external style sheet link:

   ```
   getStyleDefinitions() Returns string
   ```
4. If the element includes additional attributes, implement the following:
   - A get property of type `PTBR_BRANDING:Attributes:`*`attr_type`*. The value of *attr_type* will depend on the type of attribute (text, image, HTML, and so on)—for example, `PTBR_BRANDING:Attributes:TextAttribute`.

- getAttributeByID to return the additional attribute by ID.

  getAttributeByID(*ID* as string, *create* as boolean) Returns PTBR_BRANDING:A⇒

  ttributes:BaseAttribute

5. Create a new element type definition.

   See Defining Branding Element Types.

Example

The following example provides the implementation of the PTBR_BRANDING:Elements:BasicText class.

```
import PTPP_PORTAL:UTILITY:Collection;
import PTBR_BRANDING:Attributes:BaseAttribute;
import PTBR_BRANDING:Attributes:TextAttribute;
import PTBR_BRANDING:Elements:BaseElement;



/**
  * BasicText Class
  */
class BasicText extends PTBR_BRANDING:Elements:BaseElement
   /* --- Properties --- */
   property PTBR_BRANDING:Attributes:TextAttribute Text get;


   /* --- Methods --- */
   method BasicText(&pId As string);
   method clone() Returns PTBR_BRANDING:Elements:BaseElement;

   method getAttributeByID(&pId As string, &pCreate As boolean) Returns PTBR_BRANDI⇒

NG:Attributes:BaseAttribute;

   method getHTML(&pPreview As boolean) Returns string;


private
   /* --- Private Properties --- */
   Constant &cstText = ".text";


   /* --- Private Methods --- */


end-class;



/**
  * Constructor
  *
  * @param pId ID of the object.
  *
  */
method BasicText
   /+ &pId as String +/

   %Super = create PTBR_BRANDING:Elements:BaseElement(&pId);

   %This.setElementType("BasicText");
   rem %this.setAdvancedOptionsPage (Page.);
```



```
   /* Initialize attributes */
   %This.StyleClass = "PSTEXT";
   %This.Text.setText("");

end-method;



/**
  * Make an exact copy of this object.
  * NOTE: Only create the object here, do not copy any class properties.
  *       If there are new class properties, override the copyProperties
  *       method and do it there.  Should invoke the %Super.copyProperties
  *       in that method though.
  *
  * @return BaseElement New object exactly matching this object.
  *
  */
method clone
   /+ Returns PTBR_BRANDING:Elements:BaseElement +/
   /+ Extends/implements PTBR_BRANDING:Elements:BaseElement.clone +/

   Local PTBR_BRANDING:Elements:BasicText &newObj = create PTBR_BRANDING:Elements:B⇒

asicText(%This.ID);

   /* Call the copy properties function */
   %This.copyProperties(&newObj);

   Return &newObj;

end-method;



/**
  * Get Attribute By ID
  *
  * @param pId     ID of the attribute.
  * @param pCreate True - create the attribute when not exist.
  *
  * @return BaseAttribute - Attribute object
  *
  */
method getAttributeByID
   /+ &pId as String, +/
   /+ &pCreate as Boolean +/
   /+ Returns PTBR_BRANDING:Attributes:BaseAttribute +/
   /+ Extends/implements PTBR_BRANDING:Elements:BaseElement.getAttributeByID +/

   Local PTBR_BRANDING:Attributes:BaseAttribute &getAttributeByID = %Super.getAttri⇒

buteByID(&pId, &pCreate);
   Local PTBR_BRANDING:Attributes:TextAttribute &TextAttribute;

   If ((&getAttributeByID = Null) And
         &pCreate) Then

      Evaluate &pId
      When &cstText
         &TextAttribute = create PTBR_BRANDING:Attributes:TextAttribute(&pId);
         &TextAttribute.Label = MsgGetText(219, 7509, "Text");
         &TextAttribute.setText("");
         &TextAttribute.IsVisible = True;
         &getAttributeByID = &TextAttribute;
         Break;

      End-Evaluate;

      If (&getAttributeByID <> Null) Then
```



```
            %This.Attributes.insert(&getAttributeByID);
         End-If;

      End-If;

      Return &getAttributeByID;

   end-method;


   /**
     * Get HTML of the element
     *
     * @param pPreview  True - preview mode.
     *
     * @return String - HTML
     *
     */
   method getHTML
      /+ &pPreview as Boolean +/
      /+ Returns String +/
      /+ Extends/implements PTBR_BRANDING:Elements:BaseElement.getHTML +/

      Local string &getHTML = "";

      &getHTML = GetHTMLText(HTML.PTBR_ELM_CONTAINER_SPAN, EscapeHTML(%This.ID), Escap⇒

   eHTML(%This.StyleClass), EscapeHTML(%This.Text.getRuntimeValue()));

      Return &getHTML;

   end-method;


   /*****************************
   *   Get/Set method pairs
   *****************************/

   /**
     * Text Property
     *
     */
   get Text
      /+ Returns PTBR_BRANDING:Attributes:TextAttribute +/

      Return (%This.getAttributeByID(&cstText, True) As PTBR_BRANDING:Attributes:TextA⇒

   ttribute);

   end-get;
```

Creating a New User Attribute Type

To create a new user attribute type:

1. Create an application class definition that extends the PTBR_BRANDING:UserAttributes:BaseUserAttribute class.

   Use the PTBR_BRANDING:UserAttributes:* class definitions as examples.

2. Implement the following methods in the new class:

- The class constructor to initialize the object.

  ```
  MyUserAttribute(ID as string)
  ```

- getPromptViewName to return the record definition to be used as the search prompt view. If no search prompt view exists, return an empty string.

  ```
  getPromptViewName() Returns string
  ```

- validateValue to return a Boolean value indicating whether the user has the specified attribute.

  ```
  validateValue(value as string) Returns boolean
  ```

3. Create a new user attribute type definition.

   See Defining User Attribute Types.

Example

The following example provides the implementation of the PTBR_BRANDING:UserAttributes:RoleBasedUserAttribute class.

```
import PTBR_BRANDING:UserAttributes:BaseUserAttribute;


/**
  * RoleBasedUserAttribute Class
  */
class RoleBasedUserAttribute extends PTBR_BRANDING:UserAttributes:BaseUserAttribute
   /* --- Properties --- */


   /* --- Methods --- */
   method RoleBasedUserAttribute(&pId As string);

   method getPromptViewName() Returns string;
   method validateValue(&pValue As string) Returns boolean;


end-class;


/**
  * Constructor
  *
  * @param pId ID of the object.
  *
  */
method RoleBasedUserAttribute
   /+ &pId as String +/

   %Super = create PTBR_BRANDING:UserAttributes:BaseUserAttribute(&pId);

   %This.setUserAttributeType("RoleBasedUserAttribute");

end-method;


/**
  * Return the role prompt view name
  *
```



```
  * @return string - the prompt view name
  *
  */
method getPromptViewName
   /+ Returns String +/
   /+ Extends/implements PTBR_BRANDING:UserAttributes:BaseUserAttribute.getPromptVi⇒

ewName +/

   Return Record.PSROLEDEFN_SRCH;

end-method;



/**
  * Validate whether the current has this role
  *
  * @return boolean - the result
  *
  */
method validateValue
   /+ &pValue as String +/
   /+ Returns Boolean +/
   /+ Extends/implements PTBR_BRANDING:UserAttributes:BaseUserAttribute.validateVal⇒

ue +/

   Return IsUserInRole(&pValue);

end-method;
```

Migrating Branding Data Using ADSs

PeopleTools delivers the following application data sets that can be used to migrate branding definitions and configurations:

- PTBR_ELEMENT_TYPE - Branding Element Type Definitions.
- PTBR_LAYOUT - Branding Header and Footer Definitions.
- PTBR_MACRO - Branding Macro Sets.
- PTBR_STYLE_SHEET_DEFN - Branding System Options.
- PTBR_THEME - Branding Theme Definitions.
- PTBR_THEME_ASSIGNMENT - Branding Theme Assignments.
- PTBR_USER_ATTRIBUTE_TYPE - Branding User Attribute Type Definitions.

The following sections describe each application data set for branding definitions, the record definitions and related language records (RLRs) included, the fields from the base record, along with a additional information how the fields in the data set correspond to pages and fields in the user interface.

PTBR_ELEMENT_TYPE - Branding Element Type Definitions

Data set: PTBR_ELEMENT_TYPE

Description: Branding element type definitions from the Define Branding Element Types page.

Records (and RLRs): PTBR_ETYPE (PTBR_ETYPE_LN) — Branding element type definition.

Navigation: **PeopleTools** > **Portal** > **Branding** > **System Data** > **Define Branding Element Types**.

See also: Defining Branding Element Types.

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_ETYPE	PTBR_ETYPE_ID	Name	N	Y
PTBR_ETYPE	DESCR	Description	N	Y
PTBR_ETYPE	DESCRLONG	Description (long)	N	Y
PTBR_ETYPE	OBJECTOWNERID	Owner ID	N	N
PTBR_ETYPE	ACTIVE_FLAG	Active	N	N
PTBR_ETYPE	PACKAGEID	Package Name	N	N
PTBR_ETYPE	QUALIFYPATH	Path	N	N
PTBR_ETYPE	APPCLASSID	Class ID	N	N
PTBR_ETYPE	CREATEDTTM	*Not applicable*	N	N
PTBR_ETYPE	CREATEOPRID	*Not applicable*	N	N
PTBR_ETYPE	LASTUPDDTTM	*Not applicable*	N	N
PTBR_ETYPE	LASTUPDOPRID	*Not applicable*	N	N

PTBR_LAYOUT - Branding Header and Footer Definitions

Data set: PTBR_LAYOUT

Description: Branding header and footer definitions from the Define Headers and Footers page.

Records (and RLRs): PTBR_LAYOUT (PTBR_LAYOUT_LN) — descriptors for the definition, PTBR_LTELT (PTBR_LTELT_LN) — layout element properties, PTBR_LTELTAT (PTBR_LTELTAT_LN) — additional layout element attributes, PTBR_LTATTR (PTBR_LTATTR_LN) — additional style classes, PTBR_LTATTR_CLA — Company Info content for classic, and PTBR_LTATTR_FLU — Company Info content for fluid.

Navigation: **PeopleTools** > **Portal** > **Branding** > **Branding Headers and Footers**.

See also: Defining Headers and Footers.

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_LAYOUT	PTBR_LAYOUT_ID	Name	N	Y
PTBR_LAYOUT	PTBR_LAYOUT_TYPE	Type	N	N
PTBR_LAYOUT	DESCR254	Description (long)	N	Y
PTBR_LAYOUT	OBJECTOWNERID	Owner ID	N	N
PTBR_LAYOUT	CREATEDTTM	*Not applicable*	N	N
PTBR_LAYOUT	CREATEOPRID	*Not applicable*	N	N
PTBR_LAYOUT	LASTUPDDTTM	*Not applicable*	N	N
PTBR_LAYOUT	LASTUPDOPRID	*Not applicable*	N	N
PTBR_LTELT	PTBR_LAYOUT_ID	Name	N	Y
PTBR_LTELT	PTBR_ELEMENT_ID	ID	N	Y
PTBR_LTELT	PTBR_ETYPE_ID	Type	N	N
PTBR_LTELT	DESCR254	Notes	N	Y
PTBR_LTELT	PTBR_PELEMENT_ID	Parent	N	N
PTBR_LTELT	PTBR_SEQ_NUM	Order	N	N
PTBR_LTELTAT	PTBR_LAYOUT_ID	Name	N	Y
PTBR_LTELTAT	PTBR_ELEMENT_ID	ID	N	Y
PTBR_LTELTAT	PTBR_ATTR_ID	Attribute Name	N	Y
PTBR_LTELTAT	PTBR_ATTR_TYPEVAL	Attribute Type	N	N

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_LTELTAT	PTBR_ATTR_TXTVALUE	Default Text	N	Y
PTBR_LTELTAT	PTBR_ATTR_SVALUE	Attribute Value	N	N
PTBR_LTELTAT	PTBR_ATTR_EB_CLA	Enable Company Banner Area	N	N
PTBR_LTELTAT	PTBR_ATTR_EB_FLU	Use Different Content for Classic and Fluid	N	N
PTBR_LTELTAT	PTBR_ATTR_LVALUE	Static HTML	N	N
PTBR_LTATTR	PTBR_LAYOUT_ID	Name	N	Y
PTBR_LTATTR	PTBR_ATTR_ID	*Not applicable*	N	Y
PTBR_LTATTR	PTBR_ATTR_TYPEVAL	*Not applicable*	N	N
PTBR_LTATTR	PTBR_ATTR_TXTVALUE	*Not applicable*	N	Y
PTBR_LTATTR	PTBR_ATTR_SVALUE	*Not applicable*	N	N
PTBR_LTATTR	PTBR_ATTR_EB_CLA	*Not applicable*	N	N
PTBR_LTATTR	PTBR_ATTR_EB_FLU	*Not applicable*	N	N
PTBR_LTATTR	PTBR_ATTR_LVALUE	Style Definitions	N	N
PTBR_LTATTR_CLA	PTBR_LAYOUT_ID	Name	N	N
PTBR_LTATTR_CLA	PTBR_ATTR_EB_CLA	Enable Company Banner Area	N	N
PTBR_LTATTR_CLA	PTBR_ATTR_LVAL_CLA	Company Banner Content	N	N
PTBR_LTATTR_FLU	PTBR_LAYOUT_ID	Name	N	N
PTBR_LTATTR_FLU	PTBR_ATTR_EB_FLU	Use Different Content for Classic and Fluid	N	N

Record	Field	UI Field	Mergeable	In RLR
PTBR_LTATTR_FLU	PTBR_ATTR_LVAL_FLU	Fluid Banner Content	N	N

PTBR_MACRO - Branding Macro Sets

Data set: PTBR_MACRO

Description: Branding macro set definitions from the Macro Set page.

Records (and RLRs): PSMACROSET (PSMACROSETLANG) — a branding macro set, and PSMACRO (PSMACROLANG) — individual branding macros.

Navigation: **PeopleTools** > **Portal** > **Branding** > **Branding Theme Macro Sets**.

See also: Defining Macro Sets and Macros.

Record	Field	UI Field	Mergeable	In RLR
PSMACROSET	PTMACROSETID	Macro Set	Y	Y
PSMACROSET	DESCR254	Long Description	Y	Y
PSMACROSET	OBJECTOWNERID	Owner	Y	N
PSMACROSET	VERSION	*Not applicable*	Y	N
PSMACROSET	LASTUPDDTTM	*Not applicable*	Y	N
PSMACROSET	LASTUPDOPRID	*Not applicable*	Y	N
PSMACRO	PTMACROSETID	Macro Set	Y	Y
PSMACRO	PTMACROID	*Not applicable*	Y	N
PSMACRO	PTMACRONAME	Macro Name	Y	Y
PSMACRO	PTMACROCONTENT	Macro Content	Y	N
PSMACRO	DESCR100	Description	Y	Y

PTBR_STYLE_SHEET_DEFN - Branding System Options

Data set: PTBR_STYLE_SHEET_DEFN

Description: Additional style sheets and JavaScript objects from the Branding System Options page.

Records (and RLRs): PSOPTIONSADDL (none) — Additional objects.

Navigation: **PeopleTools** > **Portal** > **Branding** > **Branding System Options**.

See also: Configuring Branding System Options.

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PSOPTIONSADDL	PTCOMPONENTTYPE	*Not applicable* **Note:** Always set to *C* for classic components.	N	*Not applicable*
PSOPTIONSADDL	PTSCRIPTTYPE	• Additional Style Sheet Objects • Additional JavaScript Objects	N	*Not applicable*
PSOPTIONSADDL	PTSCRIPTNAME	• Style Sheet Name • JavaScript Name	N	*Not applicable*
PSOPTIONSADDL	SEQNO	*Not applicable* **Note:** Row order from the grid.	N	*Not applicable*

PTBR_THEME - Branding Theme Definitions

Data set: PTBR_THEME

Description: Branding theme definitions from the Assemble Themes page.

Records (and RLRs): PTBR_THEME (PTBR_THEME_LN) — theme identifiers, PTBR_THMATTR (PTBR_THMATTR_LN) — theme building blocks, and PTBR_THEME_CSS — additional theme style sheets.

Navigation: **PeopleTools** > **Portal** > **Branding** > **Assemble Branding Themes**.

See also: Assembling Branding Themes.

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_THEME	PTBR_THEME_ID	Name	N	Y
PTBR_THEME	DESCR254	Description	N	Y
PTBR_THEME	OBJECTOWNERID	Owner ID	N	N
PTBR_THEME	CREATEDTTM	*Not applicable*	N	N
PTBR_THEME	CREATEOPRID	*Not applicable*	N	N
PTBR_THEME	LASTUPDDTTM	*Not applicable*	N	N
PTBR_THEME	LASTUPDOPRID	*Not applicable*	N	N
PTBR_THMATTR	PTBR_THEME_ID	Name	N	Y
PTBR_THMATTR	PTBR_ATTR_ID	Skin Style Sheets: Name	N	Y
PTBR_THMATTR	PTBR_ATTR_TYPEVAL	*Not applicable*	N	N
PTBR_THMATTR	PTBR_ATTR_TXTVALUE	Skin Style Sheets: Description	N	Y
PTBR_THMATTR	PTBR_ATTR_SVALUE	• Homepage Header • Homepage Footer • Target Page Header • Macro Set • Classic Components: Theme Style Sheet • Skin Style Sheets: Name • Fluid Components: Global Override Style Sheet	N	N
PTBR_THMATTR	PTBR_ATTR_LVALUE	Theme Style Sheet Mappings: Node, Style Sheet, Theme Style Sheet	N	N

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_THEME_CSS	PTBR_THEME_ID	Name	N	*Not applicable*
PTBR_THEME_CSS	MSGNODENAME	Theme Style Sheet Mappings: Node	N	*Not applicable*
PTBR_THEME_CSS	STYLESHEETNAME	Theme Style Sheet Mappings: Style Sheet	N	*Not applicable*
PTBR_THEME_CSS	PTBR_THEME_CSS	• Classic Components: Theme Style Sheet • Fluid Components: Global Override Style Sheet • Theme Style Sheet Mappings: Theme Style Sheet	N	*Not applicable*
PTBR_THEME_CSS	LASTUPDDTTM	*Not applicable*	N	*Not applicable*
PTBR_THEME_CSS	LASTUPDOPRID	*Not applicable*	N	*Not applicable*
PTBR_THEME_CSS	OBJECTOWNERID	*Not applicable*	N	*Not applicable*
PTBR_THEME_CSS	VERSION	*Not applicable*	N	*Not applicable*

PTBR_THEME_ASSIGNMENT - Branding Theme Assignments

Data set: PTBR_THEME_ASSIGNMENT

Description: Portal-specific branding theme overrides and user attributed-based overrides from the Assign Branding Themes page.

Records (and RLRs): PTBR_THM_REG (PTBR_THM_REG_LN) — the portal for the theme override, PTBR_THMRGAT (PTBR_THMRGAT_LN) — portal-wide settings, PTBR_THM_UA — user attributes, and PTBR_THMUAAT (PTBR_THMUAAT_LN) — user attribute override settings.

Navigation: **PeopleTools** > **Portal** > **Branding** > **Assign Branding Themes**.

See also: Assigning Branding Themes.

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_THM_REG	PORTAL_NAME	Portal Name	N	Y
PTBR_THM_REG	EFFDT	Effective Date	N	Y
PTBR_THM_REG	OBJECTOWNERID	Owner ID	N	N
PTBR_THM_REG	CREATEDTTM	*Not applicable*	N	N
PTBR_THM_REG	CREATEOPRID	*Not applicable*	N	N
PTBR_THM_REG	LASTUPDDTTM	*Not applicable*	N	N
PTBR_THM_REG	LASTUPDOPRID	*Not applicable*	N	N
PTBR_THMRGAT	PORTAL_NAME	Portal Name	N	Y
PTBR_THMRGAT	EFFDT	Effective Date	N	Y
PTBR_THMRGAT	PTBR_ATTR_ID	*Not applicable*	N	Y
PTBR_THMRGAT	PTBR_ATTR_TYPEVAL	*Not applicable*	N	N
PTBR_THMRGAT	PTBR_ATTR_TXTVALUE	*Not applicable*	N	Y
PTBR_THMRGAT	PTBR_ATTR_SVALUE	• Default Theme • Default Skin • Custom Tabs • Default Custom Tab • Default Fluid Homepage	N	N
PTBR_THMRGAT	PTBR_ATTR_LVALUE	*Not applicable*	N	N
PTBR_THM_UA	PORTAL_NAME	Portal Name	N	*Not applicable*

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_THM_UA	EFFDT	Effective Date	N	*Not applicable*
PTBR_THM_UA	PTBR_UATA_ID	*Not applicable*	N	*Not applicable*
PTBR_THM_UA	PTBR_SEQ_NUM	Priority #	N	*Not applicable*
PTBR_THM_UA	PTBR_UATYPE_ID	Attribute Type	N	*Not applicable*
PTBR_THM_UA	PTBR_UA_VALUE	Attribute Value	N	*Not applicable*
PTBR_THMUAAT	PORTAL_NAME	Portal Name	N	Y
PTBR_THMUAAT	EFFDT	Effective Date	N	Y
PTBR_THMUAAT	PTBR_UATA_ID	*Not applicable*	N	Y
PTBR_THMUAAT	PTBR_ATTR_ID	*Not applicable*	N	Y
PTBR_THMUAAT	PTBR_ATTR_TYPEVAL	*Not applicable*	N	N
PTBR_THMUAAT	PTBR_ATTR_TXTVALUE	*Not applicable*	N	Y
PTBR_THMUAAT	PTBR_ATTR_SVALUE	• Theme Name • Skin • Custom Tabs • Default Custom Tab • Default Fluid Homepage	N	N
PTBR_THMUAAT	PTBR_ATTR_LVALUE	*Not applicable*	N	N

PTBR_USER_ATTRIBUTE_TYPE - Branding User Attribute Type Definitions

Data set: PTBR_USER_ATTRIBUTE_TYPE

Description: User attribute type definitions from the Define User Attribute Types page.

Records (and RLRs): PTBR_UATYPE (PTBR_UATYPE_LN) — user attribute type definition.

Navigation: **PeopleTools** > **Portal** > **Branding** > **System Data** > **Define User Attribute Types**.

See also: Defining User Attribute Types.

Record	***Field***	***UI Field***	***Mergeable***	***In RLR***
PTBR_UATYPE	PTBR_UATYPE_ID	Name	N	Y
PTBR_UATYPE	DESCR	Description	N	Y
PTBR_UATYPE	DESCRLONG	Description (long)	N	Y
PTBR_UATYPE	OBJECTOWNERID	Owner ID	N	N
PTBR_UATYPE	ACTIVE_FLAG	Active	N	N
PTBR_UATYPE	PACKAGEID	Package Name	N	N
PTBR_UATYPE	QUALIFYPATH	Path	N	N
PTBR_UATYPE	APPCLASSID	Class ID	N	N
PTBR_UATYPE	CREATEDTTM	*Not applicable*	N	N
PTBR_UATYPE	CREATEOPRID	*Not applicable*	N	N
PTBR_UATYPE	LASTUPDDTTM	*Not applicable*	N	N
PTBR_UATYPE	LASTUPDOPRID	*Not applicable*	N	N

Working with CSS Variables

Oracle introduced the Redwood color scheme for PeopleSoft applications in PeopleTools 8.60 and implements the color specifications through cascading style sheets (CSS) variables.

The advantages of re-use and consistency that CSS variables provide enable application developers and administrators to maintain consistent user experience across applications.

This topic does not attempt to describe the delivered CSS variables. The intent is to provide an overview of the CSS variables in PeopleSoft systems and their usage.

Understanding the PT_CUSTOM_PROPS Style Sheet

PT_CUSTOM_PROPS is a free form style sheet, which is included in the fluid and classic plus composite style sheets. It contains the CSS variables or CSS custom properties delivered in PeopleTools 8.60.

The style sheet is divided into two parts:

- Raw palette definitions.
- Functional properties.

The palette definitions and functional properties are declared at the root level in the PT_CUSTOM_PROPS style sheet.

Raw Palette Definitions

PT_CUSTOM_PROPS includes 46 color definitions. The variable names begin with `--pt-palette-` followed by a sequential number. For example, `--pt-palette-0`.

For example: `--pt-palette-0: #000000;`

Note: Oracle recommends that you do not directly use a color palette definition; instead you should use a functional variable because the color palette is used in multiple places. If PeopleTools changes the color palette definition, it will affect the functional variables wherever the color palette is used and this may cause unexpected results for you.

Functional Properties

PT_CUSTOM_PROPS includes properties such as color, height, images, background, offsets and so on. You should use these functional properties to create consistent constructs.

A functional property generally follows this naming structure: `--pt-[<HighLevelUsage>-[SubUsagelevels...]-[State(s)]-][PropertyUsage]-<property>`.

Elements of Functional Properties	***Description***
pt	Refers to PeopleTools, the owner of the functional property.
HighLevelUsage	Specifies the overall usage of a property, that is, a page, control, button, switch, and so on.
SubUsagelevels	It's a qualification of the high level usage. For example, if the high level usage is a button, then the usage level is simple, primary, toggle, and so on. This element is optional.

Elements of Functional Properties	***Description***
State	Specifies the state of a control, such as on or off state of a switch, error condition, disabled or active state of a button, hover, selected, and so on. This element is optional.
PropertyUsage	Property usage qualifies the property to be set. For example, if you are setting the color property, the property usage element indicates the type of color that is being set, that is, you are specifically setting the background color or you are setting the text color. This element is optional.
Property	Specifies the CSS property that you want to set by using the functional variable, that is, the background value, the color value, the opacity value, the border value, and so on that you want to set. Property is the last element in the structure to make it easy to understand what is being set or specified by the functional variable.

For example: `--pt-button-primary-active-background-color`.

In the following example, you use functional variables to specify the background color and text color in your construct instead of using hard-coded colors.

```
html, body, form {
  height:100%;
  background:var(--pt-page-background-color);
  color:var(--pt-page-text-color);
}
```

Using the Delivered CSS Variables

The usage of the delivered CSS variables depends on your branding requirements.

This topic discusses:

- Use of CSS variables in your custom style sheet.
- Use of CSS variables in content branding.

Using CSS Variables in Custom Style Sheet

You can use the delivered CSS variables to automatically update any colors from the PeopleSoft application that are used by you for your custom constructs in the custom style sheet.

Consider the following example, which illustrates the use of the delivered CSS variables. You are using your custom style sheet, but you're planning to copy some of the colors from the PeopleSoft applications for your constructs, such as, the background color of the left panel, the color of the selected item in the panel, and the text color in the panel. Additionally, you plan to synchronize the colors that PeopleSoft applications use for these constructs with the colors for these constructs in your system. In this case, you could use the delivered functional CSS variables that are closest to the functionality you are trying to replicate within your custom style sheet. Whenever PeopleSoft applications update the colors in future releases, the colors used in your constructs will be automatically updated without any rework on your part.

Note: In your custom style sheet, if you plan to use only your colors or color scheme for your custom constructs and do not plan to use any of the colors from PeopleSoft applications, you do *not* need to use the delivered CSS variables.

Using CSS Variables in Content Branding

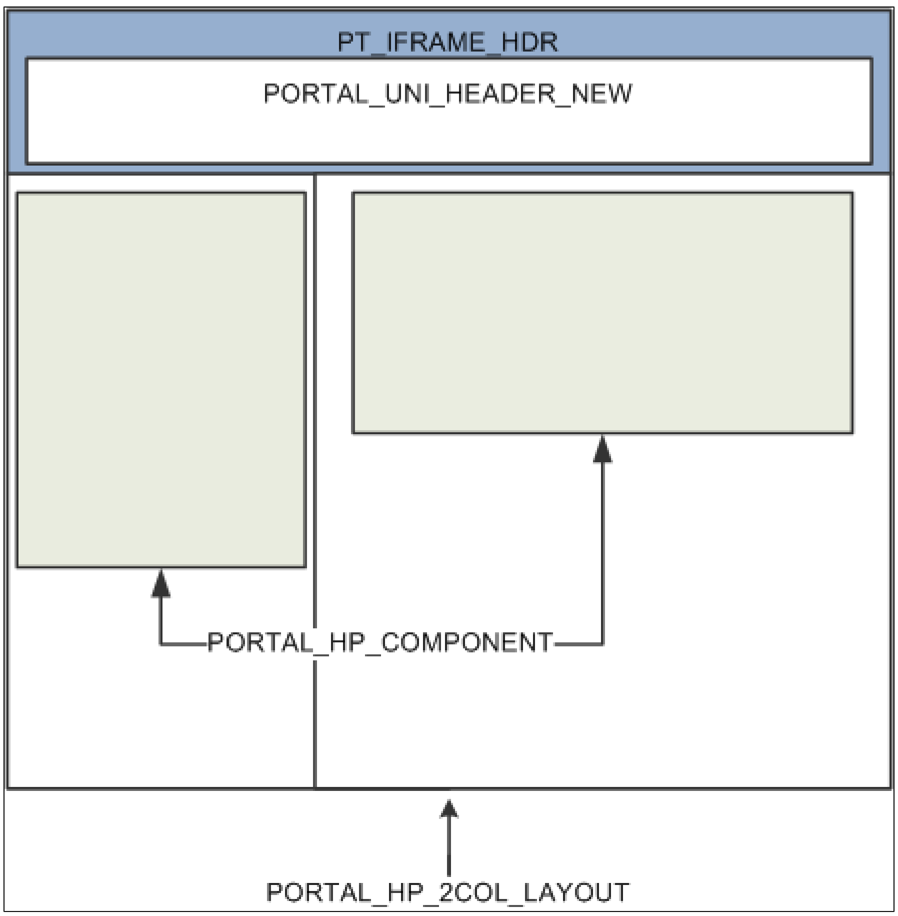
Another advantage of CSS variables is that it gives you greater control over global changes to controls internally. But you should be careful about making global changes because you should consider all contexts of usage. An example of the global changes can be seen in the delivered style sheet – PT_CUSTOM_PROPS_PRE860 – in which the palette is redefined with the pre-8.60 color scheme. This reversion style sheet allows you to retain the 8.59 color scheme. For more information on the delivered style sheet for reversion to the 8.59 color scheme, see Assembling Branding Themes, *Reverting Branding to Pre-8.60 Visuals*.

Chapter 6

Modifying the Portal Interface

Using Predefined Template HTML Definitions

The following example shows the arrangement of some of the HTML definitions that make up a template. The PORTAL_HP_USER_TEMPLATE HTML definition controls the overall page. The PORTAL_UNI_HEADER_NEW HTML definition controls the universal navigation header. The PORTAL_HP_COMPONENT HTML definition controls the appearance of each pagelet and appears twice in this two-column template.



This table describes the HTML definitions:

HTML Definition	***Description***
PORTAL_HP_2COL_LAYOUT	This is the HTML definition for a two-column user homepage. The default column widths are set to 33 percent for column one and 66 percent for column two.

HTML Definition	***Description***
PORTAL_HP_3COL_LAYOUT	This is the HTML definition for a three-column user homepage. The default column width is set to 33 percent for each column.
PORTAL_HP_ACTIONBAR	Use this HTML definition to create the pagelet action bar. The default action bar buttons are **Remove** and **Minimize/Restore.**
PORTAL_HP_ACTIONBARIMAGE	Use this HTML definition to create the **Customize** image button for the pagelet action bar. The image button is added only if you've defined an advanced personalization page.

PORTAL_HP_COMPONENT	Use this HTML definition to create all homepage pagelets. Modify it to change a pagelet's appearance.
	
	Do not move, delete, or change the first or last line of this HTML definition:
	
	`<!-- Begin Pagelet=%BIND(:6) -->.`
	
	`<!-- End Pagelet=%BIND(:6) -->`
	
	These special lines must exist exactly as shown. You must also ensure that all bind variables exactly match the bind variables that are used in the associated PeopleCode.

PORTAL_HP_USER_TEMPLATE	Each user homepage is generated from this HTML definition. Modify this definition to add default components that are displayed to all users.
PORTAL_UNI_HEADER_NEW	This HTML definition controls the universal header for the portal.

With the exception of the PORTAL_UNI_HEADER_NEW HTML definition, all homepage HTML definitions are built in the BuildStaticHPTemplate function.

This function is located in FUNCLIB_PORTAL.TEMPLATE_FUNC.FieldFormula.

The homepage modification functionality (minimize, maximize, and remove) is located in WEBLIB_PORTAL.PORTAL_HOMEPAGE.FieldFormula.

The iScripts are IScript_HPCompRemove, IScript_HPCompMinimize, and IScript_HPCompExpand.

Using Predefined Homepage HTML Definitions

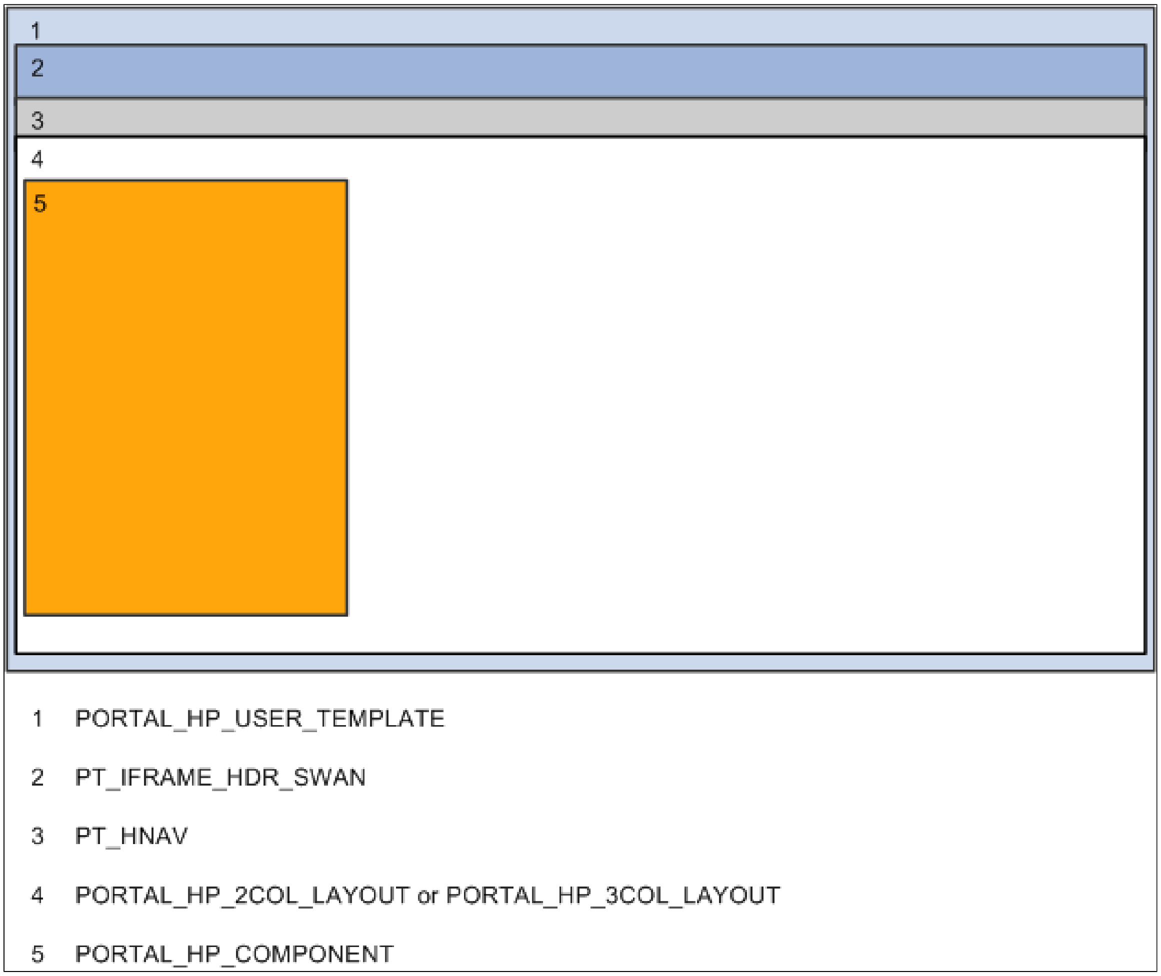
This section discusses how to:

- Construct homepages.
- Construct headers.

- Construct pagelets.
- Adapt homepage and pagelet definitions.

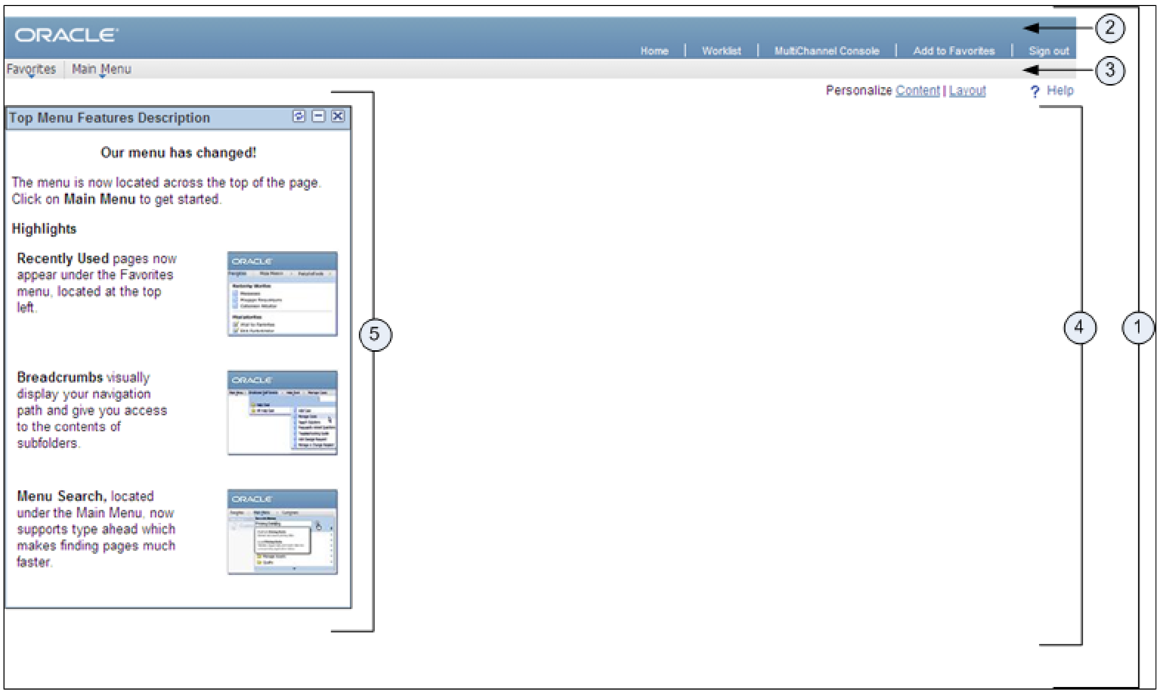
Constructing Homepages

This example shows the layout that is used for displaying a homepage with a header and pagelets. The PORTAL_HP_USER_TEMPLATE object serves as a container for the other HTML definitions found on the homepage.



This example shows an actual PeopleSoft homepage with numbers specifying these HTML definitions:

1. PORTAL_HP_USER_TEMPLATE
2. PORTAL_UNI_HEADER_NEW
3. PT_HNAV
4. PORTAL_HP_2COL_LAYOUT or PORTAL_HP_3COL_LAYOUT
5. PORTAL_HP_COMPONENT



This table describes the HTML definitions that the system uses to construct homepages:

HTML Definition	***Description***
PORTAL_HP_USER_TEMPLATE	This is the HTML definition for the homepage tab. This definition contains the iScript reference to the universal header, a bind variable for the two-column or three-column layout HTML, and binds for the Oracle image.
PORTAL_UNI_HEADER_NEW	This HTML definition contains all the HTML for the header portion of a homepage tab. The HTML definition is set up in WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula. GetUniHeaderHTML.
PT_IFRAME_HDR	This HTML definition contains all of the HTML for the iframe header.
PT_HNAV	This HTML definition contains the HTML for the drop-down navigation header.

HTML Definition	Description
PORTAL_HP_2COL_LAYOUT	This HTML definition contains all the pagelet HTML for the two-column layout (columns 1 and 2).
PORTAL_HP_2COL_LAYOUT_ACCESS	This HTML definition contains all the pagelet HTML for the two-column layout (columns 1 and 2) when the homepage is viewed in screen reader mode.
PORTAL_HP_3COL_LAYOUT	This HTML definition contains all the pagelet HTML for the three-column layout (columns 1, 2 , and 3).
PORTAL_HP_3COL_LAYOUT_ACCESS	This HTML definition contains all the pagelet HTML for the three-column layout (columns 1, 2 , and 3 when the homepage is viewed in screen reader mode.
PORTAL_HP_COMPONENT	This is the HTML definition for pagelets.

Constructing Headers

This table describes the HTML definitions that the system uses to construct headers:

HTML Definition	Description
PORTAL_HP_TABS	This is the HTML definition for all tabs. Use this HTML definition when more than one system tab exists for a user. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HOMEPAGE.FieldFormula.GetHPTabHTML().
PORTAL_HP_INACTIVE_TAB	This is the HTML definition for the inactive tab. The inactive tab name is linked. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HOMEPAGE.FieldFormula.GetHPTabHTML().
PORTAL_HP_ACTIVE_TAB	This is the HTML definition for the active tab. Only one active tab can exist. The tab name is static text. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HOMEPAGE.FieldFormula.GetHPTabHTML().
PORTAL_HP_PERSONALIZE_LINKS_TD	This is the HTML definition for the **Personalize Content** and **Personalize Layout** links. This HTML definition is used only when more than one system tab exists for a user. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HOMEPAGE.FieldFormula.GetHPTabHTML().

HTML Definition	***Description***
PORTAL_HP_PERSONALIZE_LINKS	This is the HTML definition for the **Personalize Content** and **Personalize Layout** links. This HTML definition is used only when one system tab exists for a user. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HOMEPAGE. FieldFormula.GetHPTabHTML().
PORTAL_HEADER_ADDTOFAVORITES	This is the HTML definition for the **Add To Favorites** link. This HTML definition is set up in WEBLIB_PORTAL. PORTAL_HEADER.FieldFormula.GetUniHeaderHTML().
PORTAL_HEADER__ADDTOFAVSFORM	This is the HTML definition for the Add to Favorites form. This form contains the PortalActualURL and PortalContentURL hidden fields. The Add to Favorites page uses these hidden fields to construct the content reference to the page to which you are adding a favorite (PortalContentURL) and a link back to the actual page (PortalActualURL). This HTML definition is set up in WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula. GetUniHeaderHTML().
PORTAL_HEADER_WL	This is the HTML definition for the **View Worklist** link. This HTML definition is set up in WEBLIB_PORTAL.PORTAL_ HEADER.FieldFormula.GetUniHeaderHTML().

Constructing Pagelets

This table describes the HTML definitions that the system uses to construct pagelets:

HTML Definition	***Description***
PORTAL_HP_ACTIONBAR	This is the HTML definition for the pagelet action bar. For a personalized tab, the pagelet action bar always includes a Maximize image button. It may also contain up to three more image buttons (Remove, Edit, and Help). This HTML definition is not populated for a pagelet on a system tab.
PORTAL_HP_ACTIONBAR_EDIT	This is the HTML definition for the Edit image button on the pagelet action bar.
PORTAL_HP_ACTIONBAR_HELP	This is the HTML definition for the Help image button on the pagelet action bar.
PORTAL_HP_ACTIONBAR_REFRESH	This is the HTML definition for the Refresh image button on the pagelet action bar.
PORTAL_HP_ACTIONBAR_REMOVE	This is the HTML definition for the Remove image button on the pagelet action bar.

HTML Definition	***Description***
PORTAL_HP_COMPONENT	This is the HTML definition for pagelets.
PT_HP2_BANNER	This is the HTML definition for the banner pagelet.
PT_HP2_PGLTHNDL	This is the HTML definition for the pagelet label when the pagelet cannot be dragged.
PT_HP2_PGLTHNDL_DRAG	This is the HTML definition for the pagelet label when the pagelet can be dragged.
PT_HP2_PGLT_HDR_IMG	This is the HTML definition for the pagelet header image.

Adapting Homepage and Pagelet Definitions

This section discusses how to:

- Build portal homepage tabs by using fixed-name HTML and image definitions.
- Add images to homepage tabs.

Building Portal Homepage Tabs by Using Fixed-Name HTML and Image Definitions

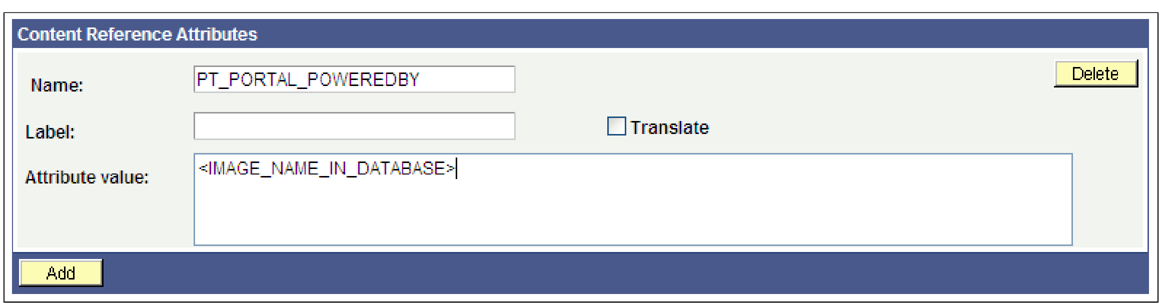
PeopleSoft applications use fixed-name HTML and image definitions to build portal homepage tabs. You can override default HTML definitions and images that are used for homepage elements, such as tabs and pagelets, by adding attributes on the Content Ref Administration page.

Access the Content Ref Administration page (Select **PeopleTools** > **Portal** > **Structure and Content.** Click the **Portal Objects** link.)

To work with homepages, click the **Homepages** link, then the **Tabs** link. Click a homepage tab label.

To work with pagelets, click the **Pagelets** link, then a pagelet group name link. Click a pagelet label.

This example illustrates building portal homepage tabs using fixed-name HTML and image definitions.



To override a default fixed-name HTML or image definition:

1. In the **Content Reference** group box at the bottom of the page, click the **Edit** link for a content reference.
2. Scroll to the **Content Reference Attributes** group box for a content reference.
3. In the **Name** field, enter the name of the default definition that you want to change.
4. Deselect the **Translate** check box.

   Note: You must deselect the **Translate** check box for the override to be successful.

5. In the **Attribute Value** field, specify the new definition value.

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

6. Click **Save.**

You can omit the **Label** field value.

As shown in the example, to replace the Powered by PeopleSoft image on the homepage tab, add a untranslated attribute named PT_PORTAL_POWEREDBY with a new image definition name in the database.

This table shows a list of default definitions that you can change for a tab:

HTML or Image Definition	***Value***
PT_PORTAL_POWEREDBY	*Image name in database.*
PT_PORTAL_IC_REFRESH	*Image name in database.*
PT_PORTAL_IC_EDIT	*Image name in database.*
PT_PORTAL_IC_CLOSE	*Image name in database.*
PT_PORTAL_IC_COLLAPSE	*Image name in database.*
PT_PORTAL_IC_EXPAND	*Image name in database.*
PT_PGLT_HELP	*Image name in database.*
PORTAL_HP_COMPONENT	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR	*HTML definition name in database.*

HTML or Image Definition	***Value***
PORTAL_HP_ACTIONBAR_REFRESH	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR_REMOVE	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR_EDIT	*HTML definition name in database.*
PORTAL_HP_ACTIONBAR_HELP	*HTML definition name in database.*
PORTAL_HP_2COL_LAYOUT	*HTML definition name in database.*
PORTAL_HP_3COL_LAYOUT	*HTML definition name in database.*
PORTAL_HP_USER_TEMPLATE	*HTML definition name in database.*

Adding Images to Homepage Tabs

You can specify up to 15 extra images for a homepage by using %BIND variables on PORTAL_HP_COMPONENT and PORTAL_HP_ACTIONBAR_xxx HTML definitions. Homepage generation logic replaces %BIND12 to %BIND26 on the homepage tab by the images that are specified as attributes.

Note: Bind variables for images and all HTML components (tab and pagelet) should be not be translated. To make an item not translatable, deselect the **Translate** check box.

The following %BIND variables can be replaced by the Homepage Generation service (specify bind variables in PeopleSoft Application Designer):

Bind Variable	***Definition Replaced***
%BIND(:1)	The PT_PORTAL_POWEREDBY image (Powered By PeopleSoft image).
%BIND(:2)	The result of a function call to %Request.Full URI.
%BIND(:3)	The result of a function call to %Request.GetHelpURL ('PERSHOEPAGE').
%BIND(:4)	The result of a function call to refreshonExpire.
%BIND(:5)	The PT_PGLT_HELP image.

Bind Variable	***Definition Replaced***
%BIND(:6)	The result of a function call to %Request.GetHelpURL(&Pglt. HelpID).
%BIND(:7)	The PT_PORTAL_IC_EDIT image (edit image for a pagelet).
%BIND(:8)	The result of a function call to SetRefreshPage.
%BIND(:9)	The PT_PORTAL_IC_CLOSE image.
%BIND(:10)	The PT_PORTAL_IC_COLLAPSE image.
%BIND(:11)	The PT_PORTAL_IC_EXPAND image.
%BIND(:12) to %BIND(:26)	The value of the tab attribute IMAGE_BIND_*X,* where *X* ranges from 12 to 26.
%BIND(:28)	The PT_PORTAL_IC_REFRESH image.

This code sample shows how to specify custom %BIND images to display custom images for the PORTAL_HP_COMPONENT HTML definition:

```
<!-- Begin Pagelet=%BIND(:5) -->
<!-- PageletState=MAX -->
<TABLE id="" CLASS="PTPAGELET" WIDTH="100%" CELLPADDING="0" CELLSPACING="0" BORDER=⇒

"1">
<TR>
     <TD>
          <!--Image 12 will be displayed here.-->
          <IMG SRC="%BIND(:12)" NAME="anything you like">
          <!--Image 13 will be displayed here.-->
          <IMG SRC="%BIND(:13)" NAME="anything you like">
          <!--Image 14 will be displayed here.-->
          <IMG SRC="%BIND(:14)" NAME="anything you like">
          <!--Image 26 will be displayed here.-->
          <IMG SRC="%BIND(:26)" NAME="anything you like">
     </TD>
</TR>
   <TR>
      <TD CLASS="PTPAGELETHEADER">
         %BIND(:1)
         %BIND(:2)</nobr>
      </TD>
   </TR>
   <TR>
      <TD CLASS="PTPAGELETBODY" WIDTH="100%">
         <Pagelet Name="%BIND(:2)">
            <Source Pagelet="%BIND(:4)" href="%BIND(:3)" />
         </Pagelet>
      </TD>
   </TR>
</TABLE>
<BR>
```



```
<!-- End Pagelet=%BIND(:5) -->
```

The HTML definitions PORTAL_HP_2COL_LAYOUT and PORTAL_HP_3COL_LAYOUT HTML can be modified similarly to display extra images.

The Homepage Generation service replaces the image that is specified by the attribute IMAGE_BIND_12 in place of bind variable %BIND(:12), and so on.

Warning! Pagelet HTML is parsed by the portal servlet. Be sure that the HTML is properly formatted and that none of the matching tags are missing. Improperly formatted HTML and missing tags may result in unexpected web server behavior.

Using Custom Portal Headers

The following list presents custom headers that you can use with the portal:

- PortalServletURI
- PortalURI
- PortalActualURL
- PortalContentURL
- PortalContentProvider
- PortalRegistryName
- PortalTargetFrame
- PortalPagelet

These are valid parameters that you can pass with the PeopleCode GetParameter method. PortalContentURL retrieves the registered URL, while PortalActualURL retrieves the entire URL, including any additional query string parameters.

Changing Link Colors

Use the PSSTYLEDEF style sheet to change link colors.

To change the link color:

1. Open the PSSTYLEDEF style sheet in PeopleSoft Application Designer.
2. Locate and open the PSHYPERLINK class.
   a. Modify the attributes as appropriate.
   b. On the Properties page, modify the pseudo classes.
   c. Specify default font attributes for <A> and <TD> in your own style sheet or in an embedded style sheet at the top of PORTAL_UNI_HEADER_NEW.

This applies font attributes to text that has no class associated with it. For example:

```
<STYLE TYPE="text/css">
   td { font-family:arial; font-size:9pt; }
      a:visited { font-family:arial; font-size:9pt; color: #5c93ae; text-⇒

decoration:underline }
      a:link { font-family:arial; font-size:9pt; color: #5c93ae; text-dec⇒

oration: underline }
      a:hover { font-family:arial; font-size:9pt; color: #5c93ae; text-de⇒

coration: underline }
      a:active { font-family:arial; font-size:9pt; color: #5c93ae; text-d⇒

ecoration: underline }
</STYLE>
```

Note: Oracle recommends that you do not alter the delivered PSSTYLESHEET style sheet definition. Make a copy of PSSTYLESHEET and modify the copy.

Related Links

“Understanding Style Sheets and Style Classes” (Application Designer Developer’s Guide)

Applying Changes

To immediately view changes that you make to a page, click the browser **Refresh** button. This reloads the updated page, instead of waiting for the old page to expire and the new page to be loaded.

These functions, which are found in FUNCLIB_PORTAL.TEMPLATE_FUNC FieldFormula, are also available to force a refresh:

Function	***Description***
`ForceRefreshPage(&url)`	Refreshes the default tab for the current user.
`&myPage = URLEncode("?tab=DEFAULT&user=" | %userid);`	Forces the page that is identified by the parameter to refresh. The parameter is the escaped query string of the page that is being refreshed, including the question mark.
`ForceRefreshPage(&myPage);`	Refreshes the default tab for the current user.
`ForceRefreshHomePageTab(&tab)`	Refreshes the homepage tab for the current user with the tab name that is passed in.
`ForceRefreshHomePage()`	Refreshes the default homepage tab of the current user.

Chapter 7

Working with Portal Templates

Understanding Portal Templates

The PeopleSoft portal integrates content from numerous sources and presents the merged content on a single web page in a coherent, consistent fashion that keeps users within the portal framework. A portal template tells the portal servlet what content to place on the assembled page and where each piece of content should be placed.

A portal template is HTML code. However, in addition to standard HTML tags, the portal template can include PeopleSoft tags that are used for giving instructions to the portal servlet during page assembly. A normal browser cannot interpret these special tags. The portal servlet follows instructions in the PeopleSoft tags and then removes them before passing the final page back to the user's browser.

The content in a template falls into four categories:

- HTML for the template itself.
- HTML for the main target content.
- HTML for the related target content.
- Additional HTML, such as for the navigation header.

Applying Template Types

This section provides an overview of template types and discusses how to:

- Use page-based static templates.
- Use frame-based static templates.
- Use frame-based dynamic templates.
- Use iframe-based dynamic templates.

Understanding Template Types

Each portal template is either page-based, frame-based, or inline frame-based (iframe-based), and each template is either static or dynamic. Template types come in five possible combinations:

- Page-based static templates.

  These are content references with a usage type of *HTML template* and a storage type of *Local (in HTML Catalog)*.

- Page-based dynamic templates.

  These are content references with a usage type of *HTML template* and a storage type of *Remote by URL*.

- Frame-based static templates.

  These are content references with a usage type of *Frame template* and a storage type of *Local (in HTML Catalog)*.

- Frame-based dynamic templates.

  These are content references with a usage type of *Frame template* and a storage type of *Remote by URL*.

- iFrame-based dynamic templates.

  These are content references with a usage type of *Inline frame template* and a storage type of *Remote by URL*.

A page-based template uses HTML tables to generate a page. Content is placed in each table cell that is defined by the template. The portal servlet uses the page-assembly process to retrieve documents for each cell. The assembled page is then sent as a single unit to the user's browser.

A frame-based template uses frames, rather than HTML tables, to divide the page into parts. The portal servlet constructs the appropriate URL for each frame (<src> tag in the frameset) and sends the entire frameset to the browser. The browser then retrieves the content for each frame.

An iframe-based template uses HTML with embedded inline frames to divide the page into parts. The portal servlet constructs the appropriate URL for each section of the page and sends the HTML to the browser. The browser then retrieves the content for each section and each inline frame.

A static template is based on HTML code that is entered into the HTML area of a template content reference (a content reference that exists in the Portal Objects\Templates folder of the portal registry). Bind variables cannot be used within the HTML area. The HTML includes any HTML that is required for the template itself, plus tags that specify the URLs for template pagelets. You can view and update the contents of a static template on the portal administration pages.

A dynamic template is retrieved from the web in real time, as part of the page-assembly process, rather than stored in the database with the content reference. A dynamic template can use bind variables. You cannot view the HTML contents of the dynamic template directly on the portal administration pages. Instead, the portal administration pages enable you to identify the URL that points to the resource that provides the template content. If the dynamic template is implemented by an iScript, you can use PeopleSoft Application Designer to navigate to the record and field where the iScript exists. The content provided by the URL is the HTML that is used for the dynamic template at runtime.

See Managing Portal Objects.

Using Page-Based Static Templates

This static template example combines a universal navigation header with target content. It is based on HTML tables, not frames.

```
<html>
<head>
```



```
</head>
<body>
<table>
<tr><td>
<Pagelet Name="UniversalNavigation">
    <SOURCE Node="LOCAL_NODE" href="s/WEBLIB_PORTAL.PORTAL_HEADER.FieldFormula.
   IScript_UniHeader" />
</Pagelet>
</td></tr>
<tr><td>
<Target Name="TargetContent"/>
</td></tr>
</table>
</body>
</html>
```

This template is composed of HTML for three items:

- The template itself.

  Some HTML is required for the overall template. The code for this template is represented in the preceding example by all of the text that is not highlighted. This HTML code remains on the assembled page that is sent to the user's browser after the page-assembly process.

- A template tag for the universal navigation header pagelet.

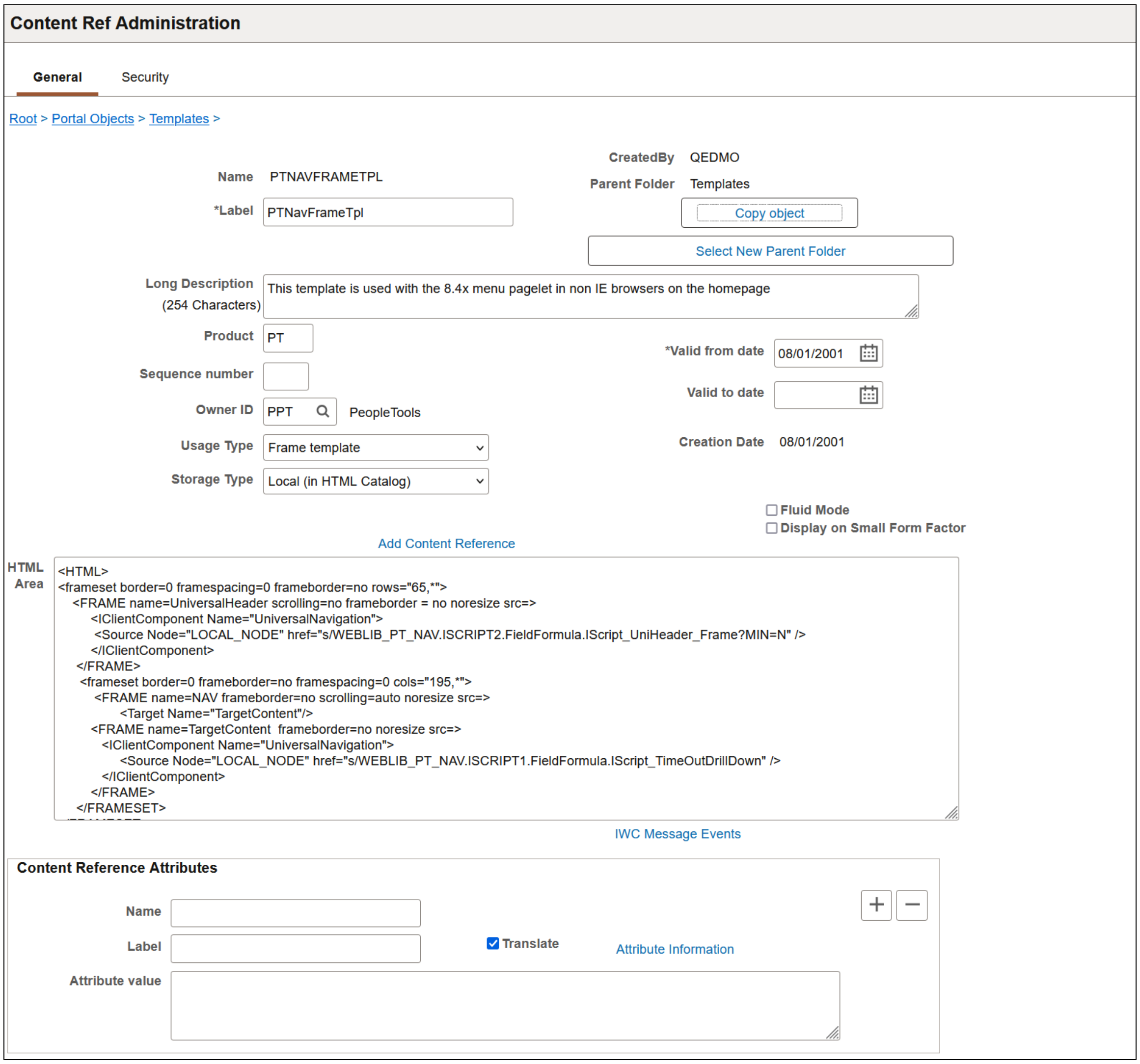
  This is the first block of HTML code that is emphasized in the preceding example. At runtime, the portal servlet replaces the Pagelet tag with the navigation header in the template for the final assembled page.

- A template tag for target content.

  At runtime, the portal servlet replaces the Target tag—the second block of emphasized HTML code in the previous example—with whatever content the user requested by clicking a link or button.

Using Frame-Based Static Templates

This example illustrates a content reference definition for a frame-based static template.

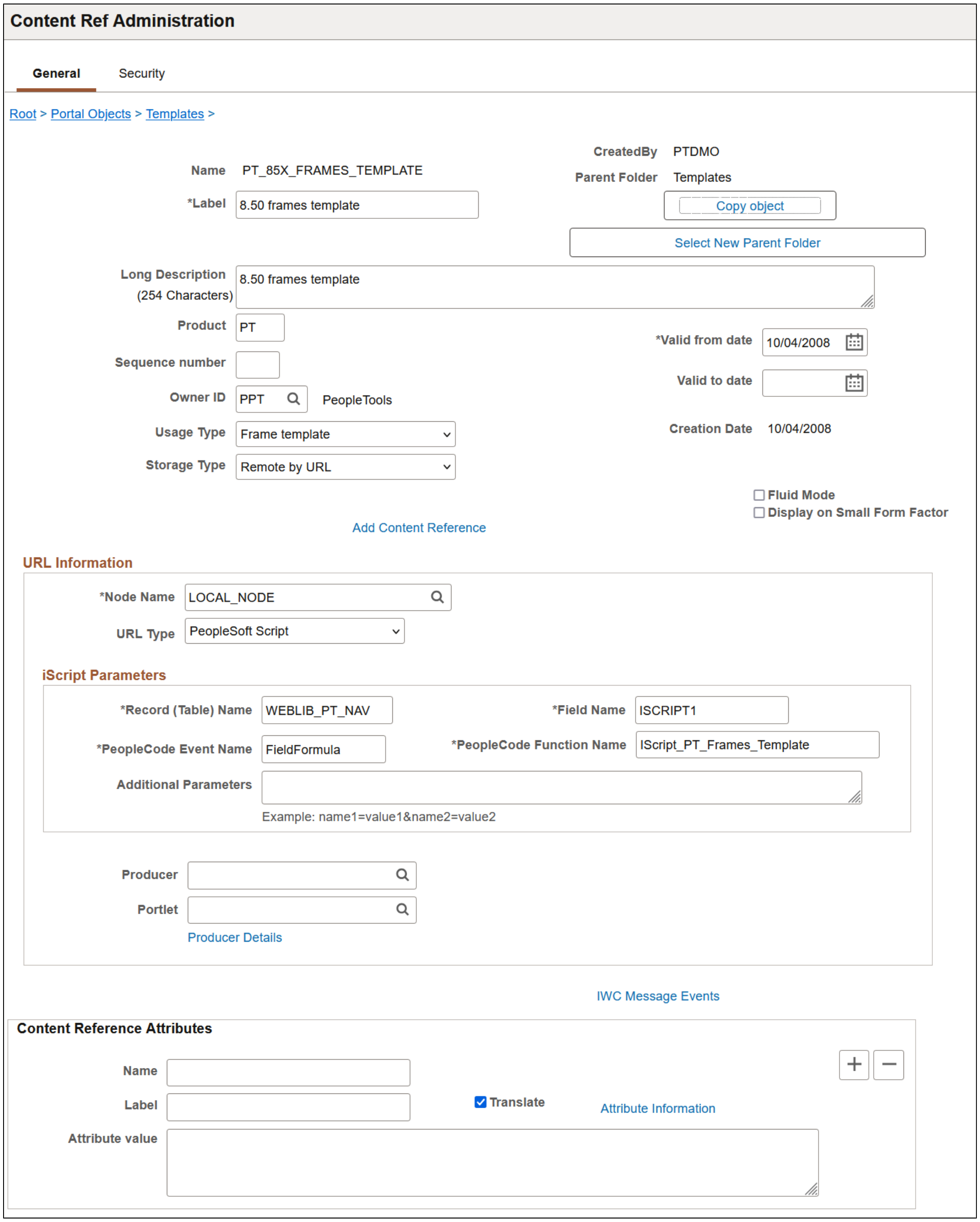


The Content Ref Administration page shows how the HTML exists within the context of the whole content reference. Note that the usage type is set to *Frame template.* In the previous example of a page-based template, which uses HTML tables, the usage type would have been set to *HTML template.*

Using Frame-Based Dynamic Templates

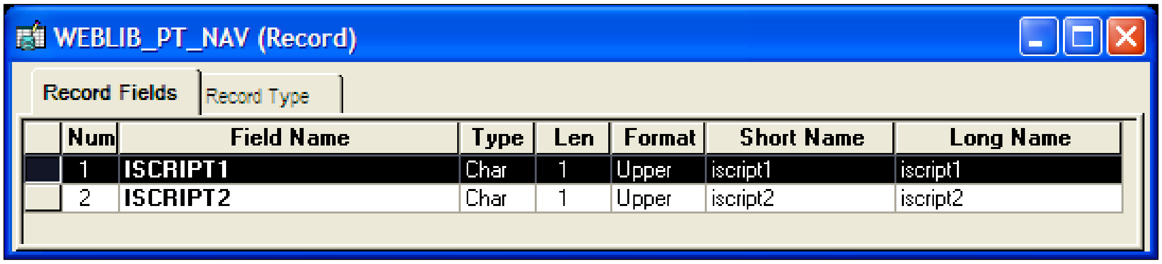
All dynamic frame-based templates are retrieved from a URL rather than from an HTML document. Most commonly, the URL for dynamic templates is an iScript URL. As with static templates, the template content reference must be defined as a template on the portal administration pages. However, instead of including specific HTML content, the dynamic template references an iScript. The iScript is associated with a specified field in a specified record.

This example illustrates the content reference definition for a frame-based dynamic template named PT_85X_FRAMES_TEMPLATE:



The content reference storage type is set to *Remote by URL.* This setting is required for dynamic templates. Additionally, no HTML area displays the associated HTML code for the template, as you would expect with a static template. Instead, when the URL type for this example is set to *PeopleSoft Script,* an iScript parameters area appears on the page, in which you can define the record name, field name, PeopleCode event name, and PeopleCode function name that specify the iScript to use for the template.

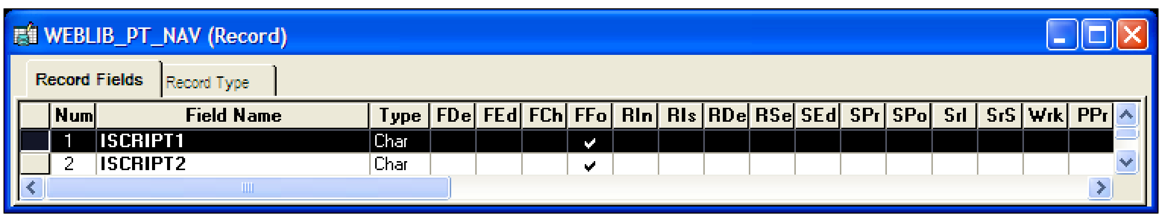
In this example, an iScript that dynamically generates the template is located in the WEBLIB_PT_NAV record in the **ISCRIPT1** field, as shown in PeopleSoft Application Designer in this example:



Additionally, the iScript function is associated with the FieldFormula PeopleCode event. The PeopleCode function name is IScript_PT_Frames_Template.

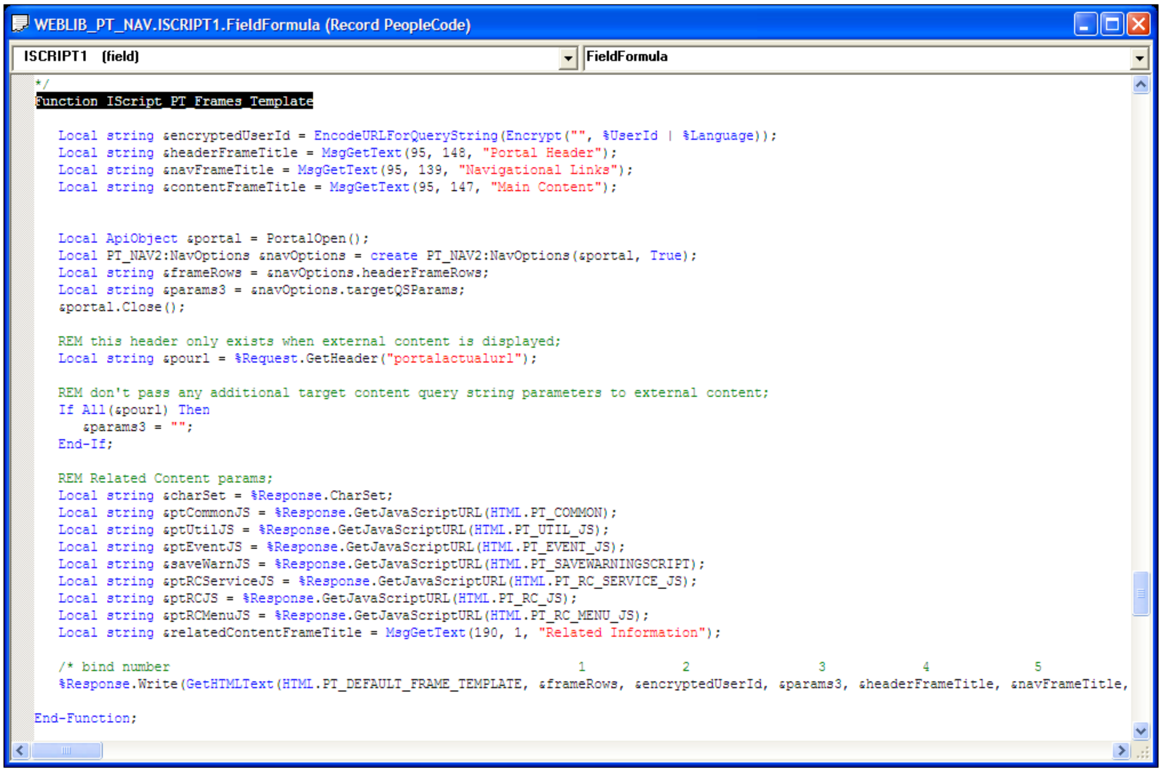
You can select the PeopleCode Display view to see which fields have PeopleCode programs defined for which PeopleCode events. By convention, function libraries have PeopleCode defined for the FieldFormula event.

This example illustrates which fields have PeopleCode programs defined for which PeopleCode events.



Once you open the appropriate record and field in PeopleSoft Application Designer, you can view the FieldFormula PeopleCode. The iScript is referenced by the PeopleCode function name in the template's definition. An easy way to find the exact reference is to copy the PeopleCode function name from the template definition and paste it into the Find dialog box of the PeopleCode editor once you open the corresponding record and field.

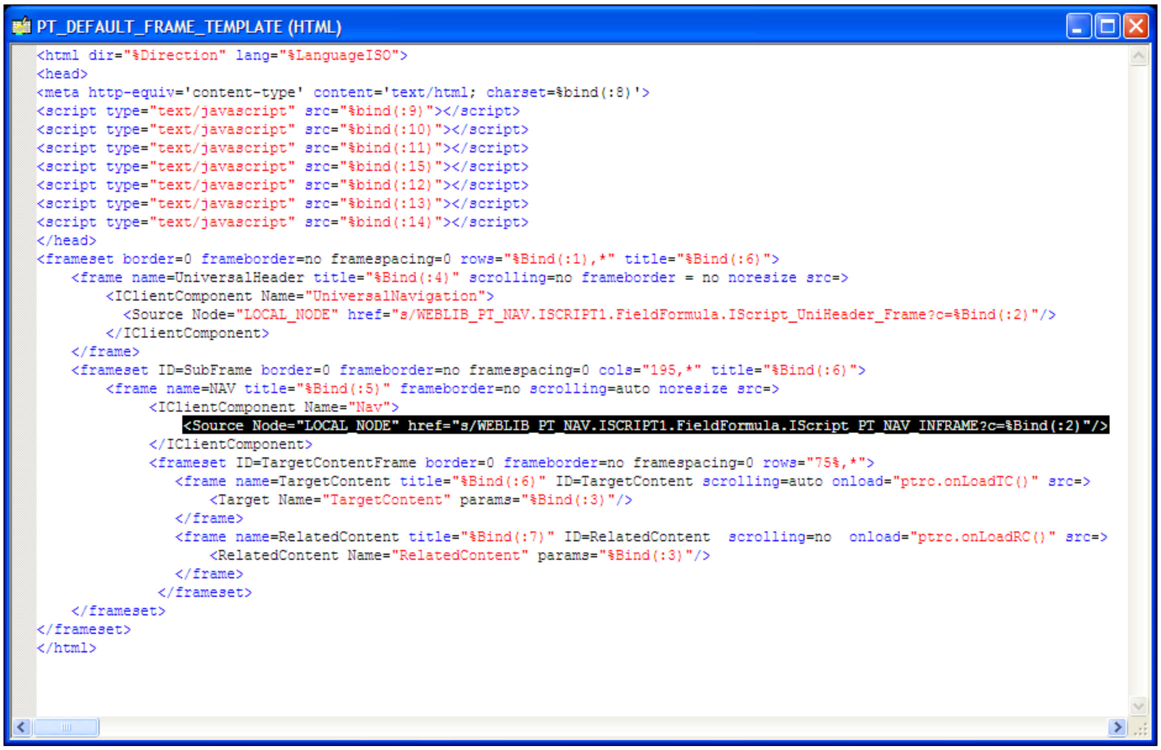
This example illustrates the PeopleCode for the iScript that is referenced by the portal template.



The following code calls an HTML definition called PT_DEFAULT_FRAME_TEMPLATE from the HTML catalog and passes the definition bind variables:

```
%Response.Write(GetHTMLText(HTML.PT_DEFAULT_FRAME_TEMPLATE, &frameRows, &encryptedU⇒

serId, &params3, &headerFrameTitle,
     &navFrameTitle, &contentFrameTitle, &relatedContentFrameTitle, &charSet, &ptCo⇒

mmonJS, &ptUtilJS, &ptEventJS, &ptRCMenuJS,
     &ptRCServiceJS, &ptRCJS, &saveWarnJS));
```

The resulting HTML code forms the basis for the dynamic portal template at runtime. You can open the HTML definition PT_DEFAULT_FRAME_TEMPLATE in PeopleSoft Application Designer, as shown in this example, and see where the bind variable is passed in (note that this is a frame-based template, as frame tags are present):



```
PT_DEFAULT_FRAME_TEMPLATE (HTML)
<html dir="%Direction" lang="%LanguageISO">
<head>
<meta http-equiv='content-type' content='text/html; charset=%bind(:8)'>
<script type="text/javascript" src="%bind(:9)"></script>
<script type="text/javascript" src="%bind(:10)"></script>
<script type="text/javascript" src="%bind(:11)"></script>
<script type="text/javascript" src="%bind(:15)"></script>
<script type="text/javascript" src="%bind(:12)"></script>
<script type="text/javascript" src="%bind(:13)"></script>
<script type="text/javascript" src="%bind(:14)"></script>
</head>
<frameset border=0 frameborder=no framespacing=0 rows="%Bind(:1),*" title="%Bind(:6)">
    <frame name=UniversalHeader title="%Bind(:4)" scrolling=no frameborder = no noresize src=>
        <IClientComponent Name="UniversalNavigation">
          <Source Node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_UniHeader_Frame?c=%Bind(:2)"/>
        </IClientComponent>
    </frame>
    <frameset ID=SubFrame border=0 frameborder=no framespacing=0 cols="195,*" title="%Bind(:6)">
        <frame name=NAV title="%Bind(:5)" frameborder=no scrolling=auto noresize src=>
             <IClientComponent Name="Nav">
                 <Source Node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_PT_NAV_INFRAME?c=%Bind(:2)"/>
             </IClientComponent>
             <frameset ID=TargetContentFrame border=0 frameborder=no framespacing=0 rows="75%,*">
                <frame name=TargetContent title="%Bind(:6)" ID=TargetContent scrolling=auto onload="ptrc.onLoadTC()" src=>
                    <Target Name="TargetContent" params="%Bind(:3)"/>
                </frame>
                <frame name=RelatedContent title="%Bind(:7)" ID=RelatedContent  scrolling=no  onload="ptrc.onLoadRC()" src=>
                    <RelatedContent Name="RelatedContent" params="%Bind(:3)"/>
                </frame>
              </frameset>
    </frameset>
</frameset>
</html>
```

Related Links

Administering Content References

Using iFrame-Based Dynamic Templates

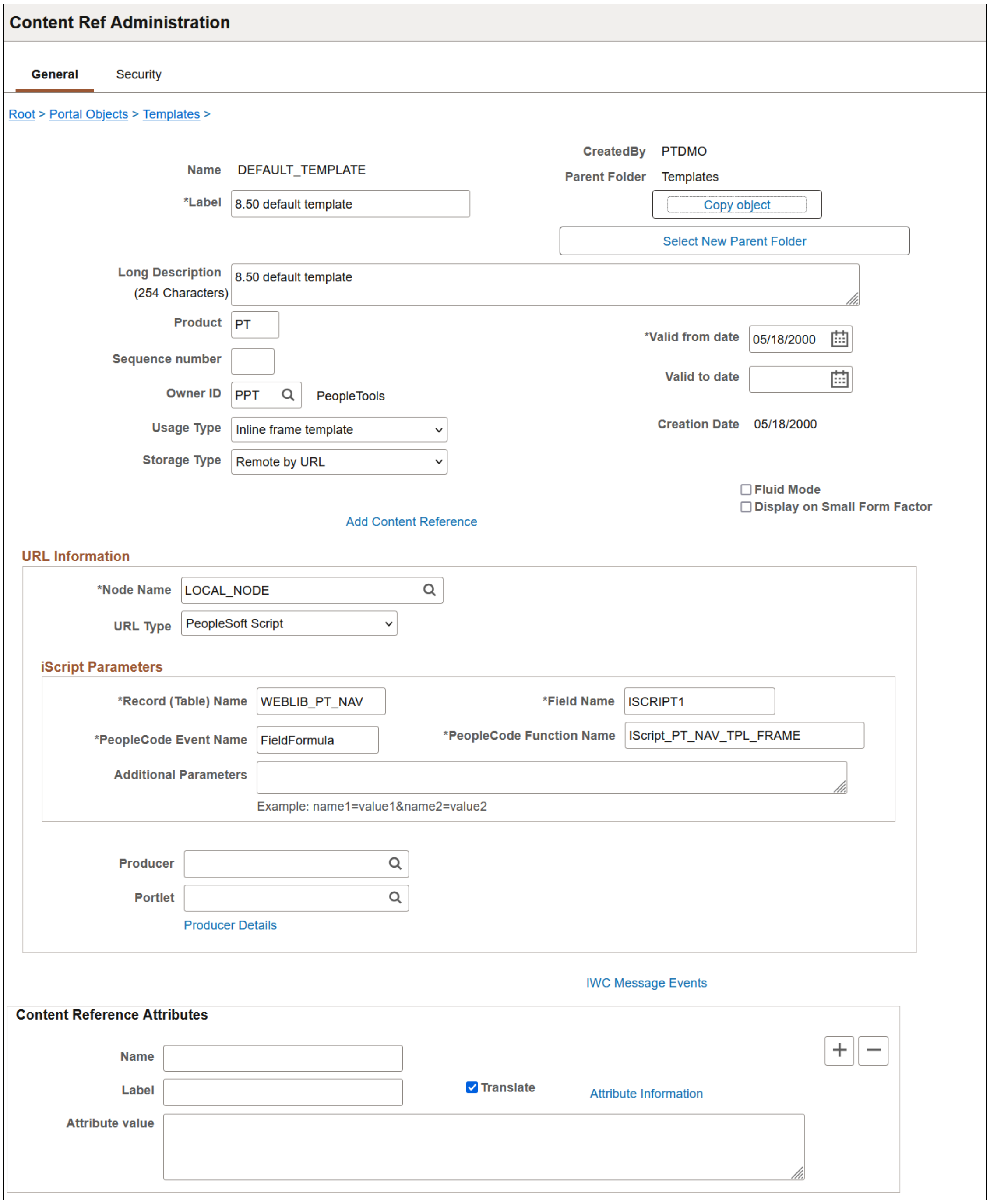
All dynamic iframe-based templates are retrieved from a URL rather than from an HTML document. Most commonly, the URL for dynamic templates is an iScript URL. As with other templates, the template content reference must be defined as a template on the portal administration pages. However, instead of including specific HTML content, the dynamic template references an iScript. The iScript is associated with a specified field in a specified record.

Important! HTML templates are desupported as of PeopleTools 8.50. PeopleSoft applications require iframe templates to function as designed. Features that are unsupported in HTML templates experience significant or total loss of functionality; such features include:

- Modal windows used for prompts, secondary pages, and grid zoom.
- Main Menu drop-down navigation.
- Related content.
- WorkCenter pages.
- Global search header.

Important! The drop-down menu, menu pagelets, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see "Working with Fluid Pages and Controls" (Applications User's Guide).

This example illustrates the content reference definition for an iframe-based dynamic template named DEFAULT_TEMPLATE:

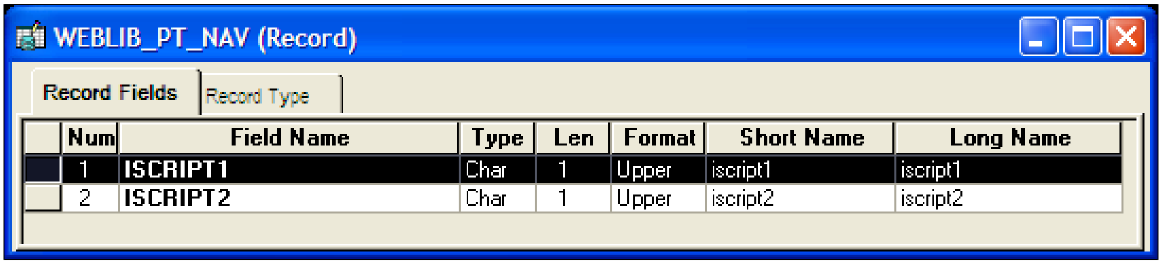


The content reference storage type is set to *Remote by URL.* This setting is required for dynamic templates. Additionally, no HTML area displays the associated HTML code for the template, as you would expect with a static template. Instead, when the URL type for this example is set to *PeopleSoft*

Script, an iScript parameters area appears on the page, in which you can define the record name, field name, PeopleCode event name, and PeopleCode function name that specify the iScript to use for the template.

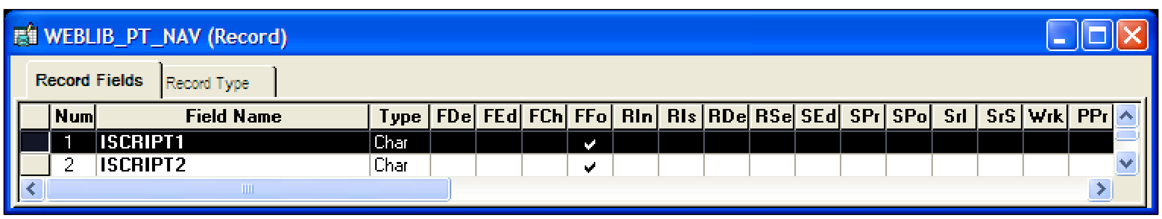
The iScript that dynamically generates the template is located in the WEBLIB_PT_NAV record in the **ISCRIPT1** field, as shown in this example:

This example illustrates the WEBLIB_PT_NAV record associated with a dynamic template.



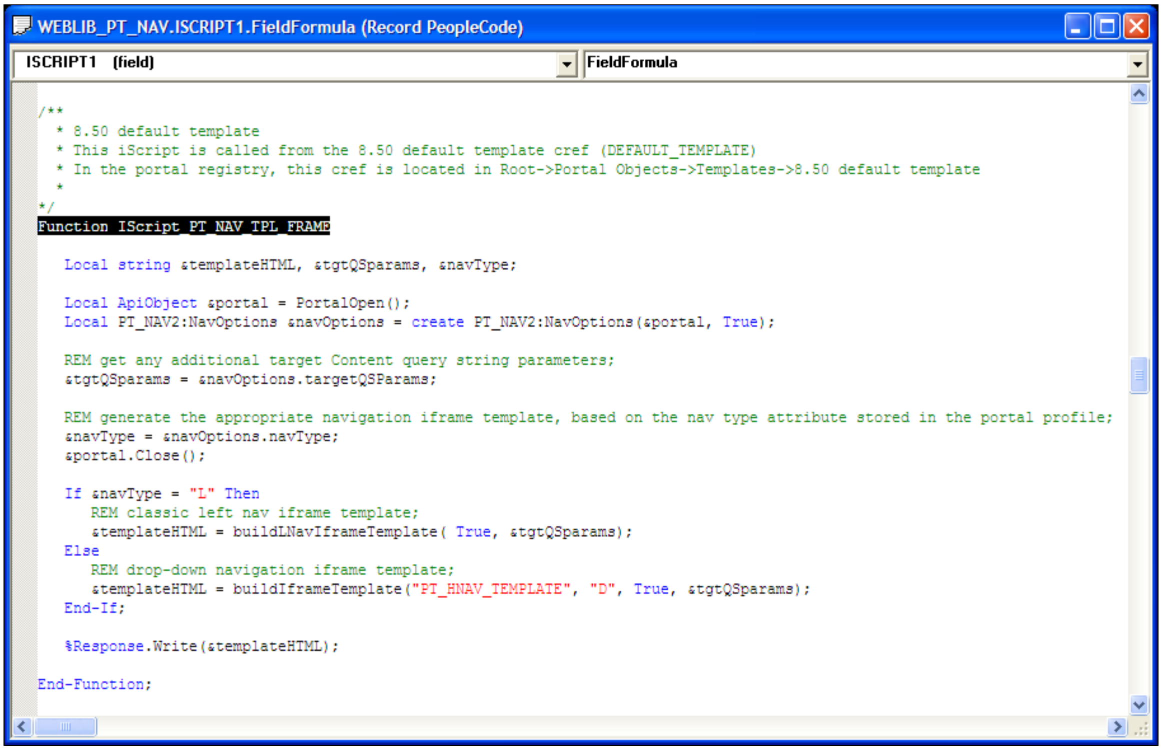
Additionally, the iScript function is associated with the FieldFormula PeopleCode event. The PeopleCode function name is IScript_PT_Frames_Template.

You can select the PeopleCode Display view to see which fields have PeopleCode programs defined for which PeopleCode events. By convention, function libraries have PeopleCode defined for the FieldFormula event.



Once you open the appropriate record and field in PeopleSoft Application Designer, you can view the FieldFormula PeopleCode. The iScript is referenced by the PeopleCode function name in the template's definition. An easy way to find the exact reference is to copy the PeopleCode function name from the template definition and paste it into the Find dialog box of the PeopleCode editor once you open the corresponding record and field.

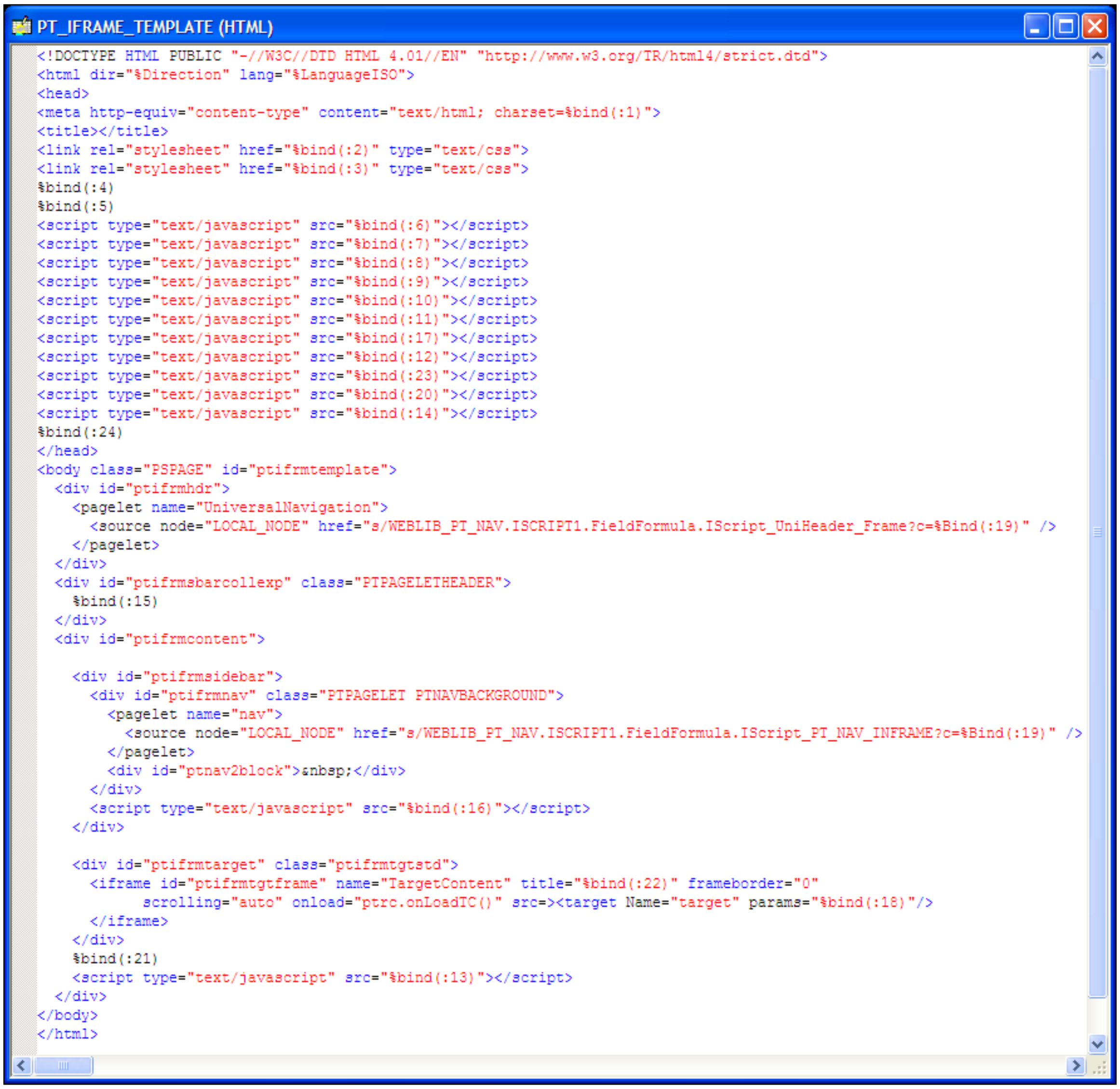
This example illustrates the PeopleCode of the iScript that is referenced by the portal template:



The following code calls the PT_IFRAME_TEMPLATE HTML definition from the HTML catalog and passes the definition bind variables:

```
   Return GetHTMLText(HTML.PT_IFRAME_TEMPLATE, &charSet, &requiredToolsSS, &sysDefa⇒
ultSS, &domainJS, &expireJS, &ptCommonJS, &ptEventJS, &ptUtilJS, &ptAjaxJS, &crossD⇒
omainJS, &saveWarnJS, &iframeHdrJS, &iframeJS, &rcJS, &navHeader.HTML, &nav2JS, &pt⇒
PopupJS, &params1, &encryptedUserId, &rcMenuJS, &rcHTML1, &tgtIframeTitle, &rcServi⇒
ceJS, &cssOverride);
```

The resulting HTML code forms the basis for the dynamic portal template at runtime. You can open the PT_IFRAME_TEMPLATE HTML definition in PeopleSoft Application Designer, as shown in this example, and see where the bind variable is passed in (note that this is an iframe-based template, as `<iframe>` tags are present):



PT_IFRAME_TEMPLATE (HTML)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html dir="%Direction" lang="%LanguageISO">
<head>
<meta http-equiv="content-type" content="text/html; charset=%bind(:1)">
<title></title>
<link rel="stylesheet" href="%bind(:2)" type="text/css">
<link rel="stylesheet" href="%bind(:3)" type="text/css">
%bind(:4)
%bind(:5)
<script type="text/javascript" src="%bind(:6)"></script>
<script type="text/javascript" src="%bind(:7)"></script>
<script type="text/javascript" src="%bind(:8)"></script>
<script type="text/javascript" src="%bind(:9)"></script>
<script type="text/javascript" src="%bind(:10)"></script>
<script type="text/javascript" src="%bind(:11)"></script>
<script type="text/javascript" src="%bind(:17)"></script>
<script type="text/javascript" src="%bind(:12)"></script>
<script type="text/javascript" src="%bind(:23)"></script>
<script type="text/javascript" src="%bind(:20)"></script>
<script type="text/javascript" src="%bind(:14)"></script>
%bind(:24)
</head>
<body class="PSPAGE" id="ptifrmtemplate">
  <div id="ptifrmhdr">
    <pagelet name="UniversalNavigation">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_UniHeader_Frame?c=%Bind(:19)" />
    </pagelet>
  </div>
  <div id="ptifrmsbarcollexp" class="PTPAGELETHEADER">
    %bind(:15)
  </div>
  <div id="ptifrmcontent">

    <div id="ptifrmsidebar">
      <div id="ptifrmnav" class="PTPAGELET PTNAVBACKGROUND">
        <pagelet name="nav">
          <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_PT_NAV_INFRAME?c=%Bind(:19)" />
        </pagelet>
        <div id="ptnav2block"> </div>
      </div>
      <script type="text/javascript" src="%bind(:16)"></script>
    </div>

    <div id="ptifrmtarget" class="ptifrmtgtstd">
      <iframe id="ptifrmtgtframe" name="TargetContent" title="%bind(:22)" frameborder="0"
            scrolling="auto" onload="ptrc.onLoadTC()" src=><target Name="target" params="%bind(:18)"/>
      </iframe>
    </div>
    %bind(:21)
    <script type="text/javascript" src="%bind(:13)"></script>
  </div>
</body>
</html>
```

Inline frame-based, or iframe-based, templates enable you to embed a frame within an HTML document rather than as a frameset. You can use the iframe `<source>` attribute to load content into the template. PeopleSoft applications use this attribute for target content only. The header and navigation content are in the HTML document itself as you see in the preceding PT_IFRAME_TEMPLATE HTML definition:

Note the following points about the standard iframe template:

- The Usage Type is set to *Inline frame template;* this is a requirement for *all* iframe-based templates.
- The iframe HTML definition supports only the <pagelet> and <target> meta-tags.

Important! The IClientComponent meta-tag is not supported.

- Using inline frames enables you to click the navigation and see only the target content change. The entire page does not refresh. This approach allows for much more flexibility and facilitates the header and navigation working together.
- When you navigate to different content within the iframe template, the URL in the browser address bar may not be up to date.
- If you create your own iframe-based template, include the !DOCTYPE at the top of the code.

Developing Portal Templates

This section discusses:

- Template pagelets based on pages.
- Inheritance of style classes.
- Template pagelet tags.
- Considerations for non-PeopleSoft content providers.
- Considerations for frame-based templates.
- Partial path nodes.
- Pagelet names in templates.
- Considerations for forms and templates.
- Error message display.

Template Pagelets Based on Pages

Template pagelets can be provided by any URL, but usually they are based on either a page or an iScript. If you are using template pagelets based on pages, add the query string parameter *&target=PanelGroupName* at the end of the query string parameter list for the template pagelet in the template HTML. This prevents PeopleSoft Pure Internet Architecture from assuming the same default name, Main, for each page. Failure to specify the target parameter results in a JavaScript error when the portal servlet tries to load the template. If more than one form exists with the name Main on the same page, the JavaScript on that page cannot determine which Main component to use. In the template HTML, add the target parameter to the source specification of each page-based template pagelet that is not meant to be the target content.

Here is an example:

```
<tr>
<td><Pagelet Name="Related Links">
     <Source Node="LOCAL_NODE" href="c/PORTAL_ADMIN.PORTAL_RLNK_1COL.GBL?target=
        relatedlinks"/>
    </Pagelet>
</td>
</tr>
```

Inheritance of Style Classes

When the portal servlet assembles portal pages based on PeopleSoft Application Designer pages, the assembled pages inherit the style classes that are defined for them in PeopleSoft Application Designer. Because you can develop a template that includes pagelets built with different style sheets in PeopleSoft Application Designer, you may have conflicting style classes in your template.

For example, if you have a style class named Big on page X and you also have a style class named Big on page Y, conflicting styles may be on the resulting page. When conflicting style classes exist, the style sheet that is associated with the page that is used for the target content always takes precedence over competing style sheets in the template.

Template Pagelet Tags

Template HTML can contain three PeopleSoft-specific tags that are used as portal servlet directives to assemble content on the page: Pagelet tags, Source tags, and Target tags.

Pagelet

Note the following attribute, element, and content information for the Pagelet tag:

Field or Control	***Description***
Attributes	Name. Used to identify the component in a comment in the final assembled page.
Elements	Source tag.
Contents	None.

Source

Note the following attribute, element, and content information for the Source tag:

Field or Control	***Description***
Attributes	Product tag. Used to identify which PeopleSoft Pure Internet Architecture web server to route to. The Product tag value should be the name of the registered content provider for the content.
Elements	None.
Contents	Query string to access the page or iScript implementing the component.

Contents must be escaped. Ampersands must be written as *&*. Use the IClientComponent tag to specify content that is not specific to the PeopleSoft Pure Internet Architecture. If a pagelet is implemented by a web server that is not a registered content provider, the entire URL of the pagelet can be specified in the Source tag contents, and the product attributes can be omitted.

Target

Note the following attribute, element, and content information for the Target tag.

Field or Control	***Description***
Attributes	Name. Used to identify the component in a comment on the final assembled page.
Elements	None.
Contents	None.

Considerations for Non-PeopleSoft Content Providers

To determine which template to apply to requested content, the portal servlet invokes the portal registry API to look up the content reference that is associated with the target content URL.

Warning! The information in this section applies to any content that is generated by a system for which the PeopleTools release is earlier than 8.4, or any time the portal cannot determine from the submitted URL what content reference is being requested.

The same URL can supply different content pages depending on the context. For example, with PeopleSoft application pages, the URL for submit buttons before PeopleTools 8.4 always looks like this: http://server/servlets/iclientservlet/peoplesoft8.

Because of this, the target content can specify its registered URL by using a custom response header, PortalRegisteredURL. When the target content supplies this response header, the portal servlet uses it to identify the URL that is registered in the portal, rather than using the actual URL that is used to get the content. All PeopleSoft Pure Internet Architecture technologies, such as pages, iScripts, and queries, supply this URL.

The value of this header must be the string with which the page should be registered. You must register the content with the identical string that it will return as this header.

If content does not return the header information, the portal servlet identifies the content by the URL in the request that is submitted to the portal.

You can override the value of the PortalRegisteredURL response header in a PeopleSoft Pure Internet Architecture script or page by adding the header to the response yourself, like this:

```
%Response.SetHeader("PortalRegisteredURL", &myURL);
```

You can do this to register the content with additional parameters.

Considerations for Frame-Based Templates

In a frame-based template, all relative URL references refer back to the psc servlet. To construct a URL reference to a page that is assembled by the portal servlet, use either the GenerateXXXXPortalURL or GenerateXXXXPortalRelURL PeopleCode functions.

Partial Path Nodes

URLs that are not registered but partially match a registered node have the template of the associated node applied. This helps you to manage unregistered URLs and to ensure that they achieve a consistent appearance within the portal. Most pages have the template formatting that you and portal users expect, rather than resorting to the default template when an exact node cannot be found.

Note: When the content reference isn't registered, it doesn't use the template defined for the node. Instead, it uses the default template. Therefore, unregistered URLs use the default template, not the template assigned to the node.

This table shows examples of nodes, the URI, and sample URLs that would use the template that is specified by the node:

Node	***URI***	***Sample Match***
Yahoo	http://www.yahoo.com	http://www.yahoo.com/business/webleaders/peoplesoft.html
Yahoo Sports	http://www.yahoo.com/sports	http://www.yahoo.com/sports/nba/heat/teamreport.html
Yahoo NFL	http://www.yahoo.com/sports/nfl	http://www.yahoo.com/sports/nfl/dolphins/teamreport.html

Pagelet Names in Templates

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

```
<Pagelet Name="My Pagelet">
<Source Product="Portal"  Pagelet="MY_PAGELET" />
</Pagelet>
```

If the portal servlet sees the Pagelet attribute in the Source tag, it looks up that pagelet name in the registry and uses the associated URL there.

Considerations for Forms and Templates

When the application posts data from a form in an HTML template, post data is lost if the action of the form is in a frame template. To solve this problem, make the frame source for the target a special request to the portal to repost the form data. The HTML response to that request includes a form containing the posted field and value pairs in hidden input fields and a Script tag that submits that form. When the browser receives this request, the JavaScript immediately submits the form to the original URL, causing the original post request to occur.

To make the frame source for the target a special portal request:

1. Construct the frameset.

   The source of the target frame is the portal, with the query string cmd=framerepost&key=xxx (or something similar).

2. Store the post data on the session object as a PropertySet (name and value pairs) for later retrieval, with a key.

   The key must uniquely identify the post data. The URL to post to can be kept in the PropertySet or put on the query string.

3. When the portal receives a frame repost request, it uses the incoming key to look up the stored PropertySet, then uses PropertySet to construct the HTML response with the name and value pairs in hidden input fields.

   The form also has a **Submit** button. The HTML response contains an inline JavaScript (after the form) that submits the form.

See *PeopleSoft 9.2 Application Installation* for your database platform.

Combining Forms and Frames

If you include a form on a page that's been assembled by the portal servlet, and that page's target is displayed in a frame (either in a frame template or otherwise), then the form action cannot be Post. The portal servlet retrieves the page by using Get—not Post—regardless of the form action. Therefore, form data is lost if the form action was originally Post. Making the form action Get resolves this problem.

Error Message Display

When a pagelet cannot be retrieved, an error message appears.

You can disable this error message. If the error is not displayed, then the pagelet disappears from the template without a message to the user.

To control the display of this error message, use the DisplayError attribute of the IClientComponent tag:

```
<IClientComponent Name="Example" DisplayError=false>
   <Source Product="">http://www.InvalidServer/InvalidPage.html</Source>
</IClientComponent>
```

If the portal servlet finds DisplayError equal to False, it does not display the error but adds the empty pagelet comment tags. If DisplayError is equal to True, or if the attribute is not found, the error message is displayed.

This feature works best when the HTML for the template can flow around the empty content properly.

Chapter 8

Using Portal Caching Features

Understanding Portal Caching

Information on your PeopleSoft portal that changes frequently must be regularly updated, but for best efficiency, information that remains static should be cached for quick retrieval. Information that changes less frequently should be updated on a more deliberate schedule. The composition of changing and static information varies depending on the user, role, application, portal, web server, information structure, and information content involved.

Caching Configuration Options

Important! Caching applies to classic homepages and pagelets only, and not to fluid homepages.

When properly configured, caching significantly boosts portal performance. PeopleSoft portal technology provides a wide range of caching options. These options target different elements of the portal environment, use a variety of criteria for applying caching, and have varying degrees of scope. Not all caching options apply to all circumstances.

You can target portal caching to a specific pagelet, template, portal, or web server. You can also apply caching based on the content reference type or browser type. In some cases, you can specify the scope of caching to be by user, by role, or by application.

Some types of caching take place on the portal web server, and others are applied at the browser without requiring individual browser configuration. You can make various caching settings in the Web Profile Configuration component (WEB_PROFILE), on a personalization page, or in a template HTML element.

With server caching, the cached data is stored on the portal web server. When a user's browser requests a piece of information from the portal, the server transmits the currently available data. The currency of the data provided depends on whether the server has refreshed the data since the last request. This is controlled by the server-based caching criteria that you defined for that data, such as a time interval or a specified number of requests.

Note: When browser caching is enabled, you can manually refresh the browser, which reloads the entire browser HTML content from the portal server. This doesn't update the information that's cached on the server, however.

Default Homepage Caching

The default homepage is automatically cached on the server only when you select the **Allow Public Access** check box in the on the Web Profile Configuration - Security page. Caching the default homepage according to these rules protects the server from having to regenerate the same homepage repeatedly. This is useful for guest sign-in scenarios.

Related Links

"Cache Settings" (System and Server Administration)

Configuring Portal Security

Cookie Rules

Administering Server-Based Caching

This section provides an overview of caching scope and discusses how to:

- Implement target content caching.
- Implement pagelet caching.
- Implement metadata caching.
- Implement proxied JavaScript caching.

Understanding Caching Scope

You can define caching for server-based portal content with one of three levels of scope:

- Private scope.

  This is also known as the *user* scope. The portal maintains the cached content independently for each user. This scope is useful when the content that is displayed, such as a 401K balance, is specific to the user who's signed in.

- Role scope.

  Each user is assigned one or more roles. The portal maintains a single cache of the content for all users with a given combination of roles. This scope is useful when the content that is displayed, such as the menu structure, is specific to the role set of the user who's signed in. If your user population has many unique role combinations, your caching memory requirements may be nearly the same as private scope caching.

- Public scope.

  This is also known as the *application* scope. The portal maintains a single cache of the content for all users who are using a given application. This scope is useful when all users who have access to this content, such as company news articles, bulletins, or messages, see the same thing. Because only one cache object is managed in memory for your entire user population, this is the least memory-intensive scope option.

Implementing Target Content Caching

You can cache the content of an individual target by specifying several caching parameters in the HTML of each locally defined template that invokes the target. You must also enable target content caching for the portal.

Important! Target content is cached in memory on the web server. If you use this feature excessively, the cache could consume all available memory. Use target content caching only to cache content that is static or doesn't require constant updating and that is accessed frequently by many users.

Specifying Caching Parameters

You apply target content caching by inserting a Cache element inside a TargetContent element in the HTML of a locally defined template.

The Cache element has these attributes:

Attribute	***Description***
Scope	Use one of these values: • *user:* The portal maintains the cached target content independently for each user. • *role:* The portal maintains a single cache of the target content for all users in a role. • *application:* The portal maintains a single cache of the target content for all users who are using an application.
Interval	Specify the number of seconds that should pass before the portal refreshes the target content cache.

This example shows caching that is defined with the application scope and an interval of 1200 seconds:

```
<TargetContent Name="MyContent">
    <Cache Scope="application" Interval="1200">ANY</Cache>
</TargetContent>
```

Note: The Cache element must not be empty, but its content can be anything.

Enabling Target Content Caching

Any Cache elements that you insert in template HTML are ignored by the portal unless you enable target content caching globally for the portal. Select the **Cache Target Content** check box in the **On the Web Server** group box of the Web Profile Configuration - Caching page.

Related Links

Understanding Portal Templates

Applying Template Types

Developing Portal Templates

Configuring Caching

Implementing Pagelet Caching

If you're developing a pagelet and you know that its content will remain static for all users, you can specify the application or public scope. If you know that the pagelet's content will remain static for all users with the same role, you can specify the role scope. However, if the pagelet content is static only for individual users, you must specify the user or private scope. If you're not the pagelet developer, you probably won't know when the pagelet content will change, so to be sure you must specify the user scope.

Important! Pagelet content is cached in memory on the web server. If you use this feature excessively, the cache could consume all available memory. Use pagelet caching only to cache content that is static or doesn't require constant updating, and is accessed frequently by many users.

You can cache the content of an individual pagelet by specifying caching parameters in one of these locations:

- The HTML of any locally defined template that invokes the pagelet.
- A content reference attribute in the pagelet definition.

Note: If you specify caching parameters for a pagelet in both locations, the caching parameters in the template take precedence, and the caching parameters in the content reference attribute are ignored.

The presence of the caching parameters is sufficient to enable caching for the associated pagelet; no additional setting is required. To disable pagelet caching, delete the caching parameters.

Specifying Pagelet Caching in the Template

You apply pagelet caching in a template by inserting a Cache element inside a Pagelet element or IClientComponent element in a locally defined template.

The Cache element has these attributes:

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Scope</td><td>Use one of these values:
user: The portal maintains the cached pagelet independently for each user.

Note: In a high-volume environment, caching pagelets with the user scope requires a separate web server cache of each pagelet for every user, which can result in significant memory problems. Because of this, caching pagelets with the role or application scope whenever possible is strongly recommended.
role: The portal maintains a single cache of the pagelet for all users in a role.
application: The portal maintains a single cached version of the pagelet for each user. The cache manager clones a copy of the meta pagelet and stores a copy for each user. For example, if 1000 users have signed in and they all have this pagelet on their homepage, 1000 copies of the pagelet will be stored in cache. Each of these pagelet copies has the associated time to live (TTL) variable. Each time the system builds the user's homepage, it reviews the TTL to determine whether the time has come to refresh the pagelet. When a TTL time expires, that copy of the meta pagelet is invalidated and refreshed from the application server. When each user's copy expires, his or her copy will be updated with the latest version.</td></tr>
<tr><td>Interval</td><td>Specify the number of seconds that should pass before the cache expires and is subject to an automatic refresh. This value depends on the frequency with which the data changes, and the need for data to be 100 percent accurate.

For example, a company news article rarely changes once it’s been published. You might want to set the refresh period for the article pagelet to be 86400 seconds so that it's cached for a 24-hour period.</td></tr>
</table>

This example shows caching that is defined in an IClientComponent element with the role scope and an interval of 1800 seconds:

```
<IClientComponent Name="Header">
    <Cache Scope="role" Interval="1800">ANY</Cache>
    <Source Product="Portal">ICType=Script...</Source>
</IClientComponent
```

Note: The Cache element must not be empty, but its content can be anything.

Specifying Pagelet Caching in the Content Reference

You apply pagelet caching in a template by defining a special content reference attribute on the pagelet definition.

To implement caching for a pagelet:

1. Open the content reference administration page of the pagelet that you want to cache.

   Select the folder containing the pagelet, and edit its content reference.

   a. Select **PeopleTools** > **Portal** > **Structure and Content**.

   b. Click the Portal Object link.

   c. Click the Pagelets link.

   d. Click the link for the pagelet folder.

   e. Click the Edit link for the pagelet.

2. Add a new content reference attribute.
3. Specify the attribute name and label.
   - In the **Name** field, enter *PSCACHECONTROL.*
   - In the **Label** field, enter *PSCache-Control.*
4. Specify the attribute value with this syntax:

   `cache_scope,max-age=`*`interval`*

Parameter	***Description***
cache_scope	Use one of these values: • *private:* The portal maintains the cached pagelet independently for each user. **Note:** In a high-volume environment, caching pagelets with the private scope requires a separate web server cache of each pagelet for each user, which can result in significant memory usage. Because of this, caching pagelets with the role or public scope wherever possible is strongly recommended. • *role:* The portal maintains a single cache of the pagelet for all users in a role. • *public:* The portal maintains a single cache of the pagelet for all users who are using an application.

Parameter	Description
interval	Specify the number of minutes that should pass before the cache expires and is subject to an automatic refresh.

For example, consider a PeopleSoft Interaction Hub environment in which users can personalize their company news pagelet. The articles don't change frequently, and users don't need them to be current; therefore, you might specify private caching with a three-hour (180 minute) expiration:

```
private,max-age=180
```

5. Save the content reference.

   The next time a user signs in and displays the pagelet, the caching that you specified takes effect.

Managing the Caching of Pagelets

You can enable users to personalize their homepage pagelets. When a user personalizes a pagelet, the result should be visible only to that user, and therefore should be cached only for that user. To achieve this goal, as soon as the user attempts to personalize the pagelet, regardless of the caching scope that you originally specified, the portal automatically changes the caching scope to apply on a private basis and maintains the cached pagelet independently for that user.

Important! This automatic scope change applies only to the user who is personalizing the pagelet and to the portal on which the pagelet is personalized. On other portals, and for other users, the originally defined caching scope for that pagelet still applies. If the personalizations aren't stored as portal-specific data, then the personalizations for a pagelet that was originally defined with a wider caching scope can become visible to others who are accessing the same site through another portal.
Therefore, when developing personalizable pagelets, you need to ensure that stored pagelet personalization metadata is keyed by both user ID and portal name so that you can populate the pagelet independently for each portal. This prevents user personalizations on one portal from being cached and visible to a larger audience on another portal.

Disabling Manual Pagelet Refresh

When you implement pagelet caching, a refresh button appears by default in the pagelet header. You can prevent users from refreshing the pagelet manually by hiding the refresh button, which you do by selecting a check box on the pagelet's content reference administration page.

Related Links

Administering Content References
Managing Pagelet Attributes
Understanding Caching Scope

Implementing Metadata Caching

Metadata caching is governed by portal-wide properties that control caching for these portal metadata objects as a group:

- The portal registry.

- Node definitions.
- Content references.
- Locally defined templates.

Portal metadata is likely to change far less frequently than pagelets or target content, so caching it on the server is an easy way to improve portal performance. Caching improves system performance by decreasing service requests from the web server to the application server.

Note: Metadata caching always applies with the application scope—the portal maintains a single metadata cache for all users who are using an application.

You implement metadata caching for a portal by setting these properties in the **On the Web Server** group box of the Web Profile Configuration - Caching page:

Field or Control	***Description***
Cache Portal Objects	Select to enable metadata caching. Metadata caching is enabled by default. You must restart the web server or wait for the stale interval to expire when you change: • The default portal for a database. • The default local node for a database. • The remote node to local or a local node to remote. • Templates that are associated with a content reference, node, or portal. • HTML within a template. **Note:** When you add or delete a content reference or folder, this change shows up in the menu structure immediately because it's rendered within the application server, not the portal servlet.
Cache Stale Interval	Specify the number of seconds that should pass before the portal refreshes the metadata cache. The default value of this property is *86400* (24 hours). With this setting, if you change the default local node in the database, the change doesn't take effect until the following day.

Field or Control	Description
Cache Purge All Hit Count	Set to the maximum total number of HTTP requests that the web server should receive for objects in the metadata caches of all portals. When the web server receives this number of hits, it purges all metadata caches, forcing the portals to refresh them. This setting enables you to manage application server memory consumption. If the web server is consuming too much memory, you can lower this value. This causes the web server to reclaim the memory that is used for cached portal objects more often. The cost of this is increased CPU usage on both the web and the application server when the web server deletes the cached objects and then retrieves them again from the application server. The default value of this property is *1000.* Set to *–1* to disable the feature.

Note: The timing properties—the cache stale interval and the cache purge all hit count—compete for effect. The first timing property to reach the specified interval or number of hits takes precedence. When the metadata cache on a portal is refreshed, the cache stale interval and cache purge all hit count counters for all portals on the same web server are reset.

Related Links

Configuring Caching

Implementing Proxied JavaScript Caching

In addition to target content, pagelets, and metadata, you can cache proxied JavaScript objects on the server. This is recommended for production systems; you might want to disable this caching during development.

You enable caching of proxied JavaScript for a portal by selecting the **Cache Proxied JavaScripts** check box in the **On the Web Server** group box of the Web Profile Configuration - Caching page.

Related Links

Configuring Caching

Administering Browser-Based Caching

This section provides an overview of browser-based caching and discusses how to:

- Implement homepage caching.
- Control navigation pagelet caching.
- Implement PeopleSoft page caching.

Understanding Browser-Based Caching

A modern web browser can cache the information that it receives from a web server in several ways. Its preference settings determine how much storage space to use for caching, as well as the criteria by which it should refresh its cache.

How the Portal Controls Browser-Based Caching

If the browser is accessing a straight HTML page, it caches the page as a whole. If it's accessing a frameset, it caches the individual frames in the frameset so that each frame's content can be refreshed independently.

Rather than depend on users to enable and set the timing criteria for caching, PeopleSoft portal technology takes advantage of a feature of HTTP. The portal web server can specify a caching interval in the HTTP response header of the HTML page or frame content that it delivers. The browser applies this value to determine when to stop using its cached copy of the data and request the most recent version from the server.

Note: Even when the browser requests a fresh copy of an HTML page or frame content, the portal web server controls whether the content it sends is cached data or fresh data based on its own caching settings.

When to Disable Browser-Based Caching

The caching system increases system response performance in typical production environments in which users access the portal web site by using computers that have been assigned specifically to them or computers that they own. However, some environments don't conform to this scenario.

You should consider disabling browser-based caching in these environments:

- Any environment in which multiple users access the same computer.

  These include kiosk-type environments—such as libraries, shared work spaces, and so on—in which multiple users access the same machine. Users might access the same machine within minutes of each other before the browser cache for the previous user times out. Security is enforced in this situation, but the new user may be confused by an inability to access certain items.

- A development or testing environment.

  When content references are created or changed for a portal during development or testing, the new data might not appear to be immediately available due to caching.

- Any environment in which you are using web server-based navigation caching.

Related Links

Configuring Caching

Implementing Homepage Caching

In a PeopleSoft portal, a homepage is a combination of portal content elements that are assembled by the portal and delivered as a single HTML page, which is cached on each user’s browser.

Based on the configuration that you specify, the portal enables or disables homepage caching by using the homepage's HTTP header. If caching is enabled, the header also conveys the time interval before the web server is accessed again to get a fresh homepage. The browser does not access the web server after it initially receives the homepage until the specified interval elapses. You can also specify whether a particular browser model and version should use or ignore the caching information in the header.

In any case, if a user clicks the browser's **Refresh** button, the homepage is accessed from the web server again, overwriting the homepage that is cached on the browser and restarting the interval counter. If any of the content is cached on the web server, the browser is refreshed from the server cache.

Enabling Homepage Caching

Important! Caching applies to classic homepages and pagelets only, and not to fluid homepages.

You enable homepage caching for a site by selecting the **Cache Homepage** check box in the **On the Browser** group box of the Web Profile Configuration - Caching page. When this check box is selected (the default value), two related page elements become editable:

- **Homepage Stale Interval.**
- **Browsers.**

When the Cache Homepage check box is selected, the following occurs:

- All non-component-based pagelets will be cached on the browser by default unless the pagelet has a PORTAL_NOBROWSERCACHE content reference attribute set to *Y.*
- Component-based pagelets will not be cached on the browser by default unless the pagelet has a PORTAL_NOBROWSERCACHE content reference attribute set to *N.*

  Important! Do not enable homepage caching for any component-based pagelet that uses interactive processing mode.

When this check box is deselected, any PORTAL_NOBROWSERCACHE content reference attributes are ignored.

Specifying the Homepage Stale Interval

Specify the homepage caching interval for a site by entering a value in seconds in the **Homepage Stale Interval** field in the **On the Browser** group box of the Web Profile Configuration - Caching page. When the specified interval elapses, the browser sends a new request for the homepage to the portal web server the next time a user accesses the homepage. The default value for this field is *1200.*

Disabling Caching for Selected Browsers

Because browsers don't all process HTML in exactly the same way, you might need to disable homepage caching for selected browser versions. This can be useful if you have one or two supported browsers and want to disable caching for nonstandard browsers that could pose an administration problem.

To disable homepage caching for a browser version, add a row to the **Browsers** grid of the Web Profile Configuration - Caching page, enter the browser's user agent ID, and select the **Cache Home Page** check box for that row. Homepage caching is enabled for all browser versions except those with this check box deselected, regardless of whether they're listed in the grid.

Note: As a convenience, several well-known browser types are included as sample rows in the configuration. All of the sample rows have homepage caching enabled by default.

Related Links

Configuring Caching

Controlling Navigation Pagelet Caching

When you use the portal menu to access target content other than your homepage, the portal delivers that content within a frameset. The portal header, menu, and target content regions are independent and can be managed separately by the browser. The menu region contains the portal's navigation pagelet, which can be cached on the browser with the user scope.

Without caching, every time the user clicks a menu folder, the browser requests a new copy of the navigation pagelet from the portal web server, which can significantly affect performance. With caching, the browser saves all of the menu states that are generated by user activity and can immediately restore them on demand.

Users manage navigation pagelet caching by using the **Time page held in cache** personalization option to specify a caching interval.

To specify a caching interval:

1. In PeopleSoft Pure Internet Architecture, select **My Personalizations.**
2. On the Personalizations page, click the **Personalize Option** button for general options.
3. Enter an override value in minutes for the **Time page held in cache** option and click **OK.**

   The default value of this option is *900* minutes. To disable navigation pagelet caching, specify *0* minutes. The maximum value can be *525600* minutes, which is one year.

Note: For the new setting to take effect, you must delete the browser cache.
With caching in effect, any changes in the menu structure while the user is signed in to the portal aren't available until the user signs out and signs in again, or until the caching interval elapses.

Implementing PeopleSoft Page Caching

PeopleSoft pages are application pages that appear in the target content region of the portal frameset.

Being able to control PeopleSoft page caching on the browser is useful for situations in which PeopleSoft applications are deployed to kiosk workstations where multiple users access the applications. Disabling caching means that users can't click the browser's **Back** button to view another person's transaction or to view any other sensitive data.

Enabling PeopleSoft Page Caching

You enable PeopleSoft page caching for a site by selecting the **Cache Generated HTML** check box in the **On the Browser** group box of the Web Profile Configuration - Caching page. PeopleSoft page caching is enabled by default.

With PeopleSoft page caching disabled, when users click the **Back** button they receive a message indicating that data is missing or that the page has expired.

Note: The side effect of turning off caching completely is degraded performance. For each new page, the system calls the database. However, PeopleTools offers a compromise related to browser caching in the form of navigation pagelet caching, which doesn't expose any sensitive information.

See Controlling Navigation Pagelet Caching.

Specifying Supported States

When the browser caches a PeopleSoft page, it stores the state of that page, including any unsaved changes. With the page state in the cache, users clicking the browser's **Back** button see the cached page in the state that it was in when they last viewed it. The browser caches as many page states as its own settings for storing temporary data allow.

The portal also maintains its own cache of page states for each browser window. You can specify how many of these states the portal should consider valid and support for further interaction. This setting, combined with the number of states that the browser maintains, determines whether users can continue to work with previously accessed page states.

For example, if you specify that the portal should support two states, and the user clicks the **Back** button three times, any activity that requires refreshing the displayed page from the web server (such as field validation) fails, producing a page unavailable message in the browser.

You specify how many states the portal should support by entering a number in the **Number of States Supported** field in the **On the Browser** group box of the Web Profile Configuration - Caching page. The default value of this field is *5* states.

Related Links

Configuring Portal Security
Configuring Caching

Administering Web Server-Based Navigation Caching

This section provides an overview and discusses how to implement web server-based navigation caching.

Understanding Web Server-Based Navigation Caching

Web server-based navigation caching considerably boosts performance by automatically storing cached data on the web server when users select items from the navigation menu. This form of caching saves unnecessary calls to the application server.

The portal maintains a single cache for all users with a given combination of roles. If your user population has limited role combinations, your caching memory requirement is significantly reduced.

See Understanding Caching Scope.

Implementing Web Server-Based Navigation Caching

Implement web server-based navigation caching by selecting the **Cache Menu** check box in the **On the Web Server** group box of the Web Profile Configuration - Caching page.

The cached data is stored in the following folder and cannot be moved:

PS_CFG_HOME\webserv\peoplesoft\applications\peoplesoft\PORTAL\psftcache

You must delete the cache periodically to keep the folder at a manageable size.

To prevent conflicts, disable browser-based caching and homepage caching.

Disabling Browser-Based Caching

To disable browser-based caching:

1. Select **PeopleTools** > **Personalization** > **Personalization Options.**
2. Select *PeopleTools (PPTL)*.
3. Remove the *METAXP* personalization definition.
4. Click **Save.**

Disabling Homepage Caching

To disable homepage caching:

1. Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.**
2. Select the appropriate web profile.
3. Access the **Browsing** grid on the Caching page.
4. Deselect the **Cache Homepage** check box for the appropriate browser.
5. Click **Save.**

Related Links

Configuring Caching

Chapter 9

Configuring the Portal Environment

Understanding the Authentication Domain

The portal *authentication domain* is the domain in which the portal is running and across which the single signon authentication token is valid. It's specified as a web server property and is used extensively throughout the PeopleSoft Pure Internet Architecture and portal runtime systems. An authentication domain is expressed as a string that completes the domain portion of a URL, for example, .example.com.

Note: The leading period is required. The correct string is, for example, .example.com, and not example.com.

The authentication domain supports the following functionality:

- Cross-frame JavaScript updates between the PeopleSoft Pure Internet Architecture and the portal.

  Failure to set the authentication domain correctly for the portal and PeopleSoft Pure Internet Architecture applications causes JavaScript security errors to appear in the browser when PeopleSoft Pure Internet Architecture pages are accessed through a portal frame-based template. (The default template through which all PeopleSoft Pure Internet Architecture pages are displayed is frame-based.) The authentication domain must be set for both the portal web server and other PeopleSoft content web servers.

- PeopleCode global variable sharing between components on the homepage and components within a frame.

  Failure to set the authentication domain correctly for the portal and PeopleSoft Pure Internet Architecture applications that use different web servers causes a new, incompatible session to be created on the PeopleSoft Pure Internet Architecture web server when the user accesses a PeopleSoft Pure Internet Architecture component through a frame-based template.

- Single signon among PeopleSoft applications.

  Failure to specify the authentication domain correctly prevents the PeopleSoft authentication cookie from being passed to the target PeopleSoft application and forces the target system to reauthenticate the user.

- Cookie sharing between the portal and third-party web applications.

  If cookies need to be shared between web applications, then each web application must be accessed over a common domain name.

  To share cookies, specify the authentication domain as the Cookies Passed to Server (forwarding domain) property in the portal's web profile. You specify this property on the web Profile Configuration - Cookie Rules page.

See Configuring Cookie Rules.

Base-Level and Extended Authentication Domains

You can define the portal authentication domain as a *base-level* authentication domain and as an *extended* authentication domain.

You define the base-level authentication domain during the PeopleSoft Pure Internet Architecture setup. This domain is stored as part of your web server configuration. It enables PeopleCode global variable sharing, which is required for initial access to the portal. The portal uses the base-level domain if you don't define an extended authentication domain.

Important! You must specify a base-level authentication domain during the setup of every PeopleSoft application with which your portal interacts. The authentication domain must be identical and be stored on the web server of each application.

See *PeopleSoft 9.2 Application Installation* for your database platform.

You can define an optional extended authentication domain in your portal web profile. An extended authentication domain overrides, but must be compatible with, the base-level authentication domain. For example, if you entered *.example.com* during the PeopleSoft Pure Internet Architecture setup, only values such as *.us.example.com* and *.fr.example.com* would be valid.

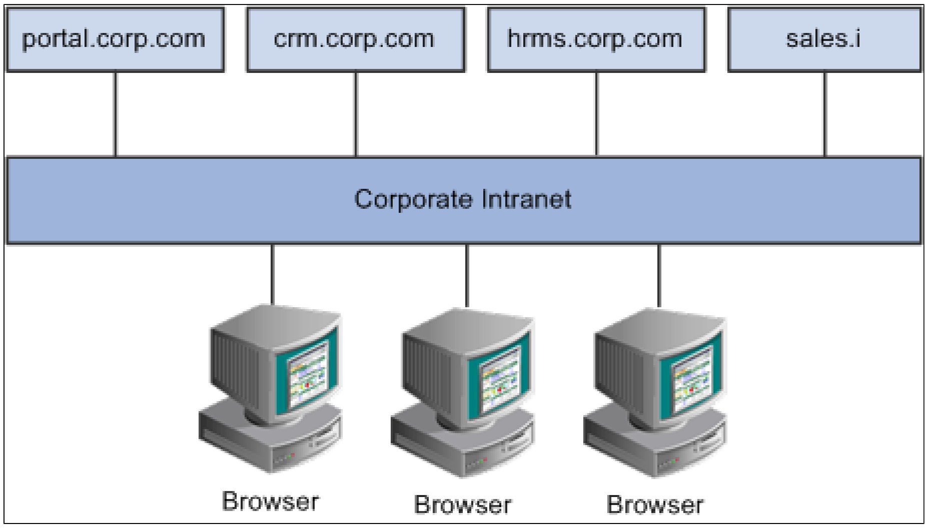
Note: If you defined a base-level or extended authentication domain, you must use it in all URLs that you specify in your portal. For example, if your authentication domain is .example.com, then instead of using the URL http://mymachine:8080/pshome/signon.html, you must use the URL http://mymachine.example.com:8080/pshome/signon.html.

You specify the extended authentication domain on General tab of the Web Profile Configuration page.

See Configuring General Portal Properties.

An Example of Multiple Applications on a Portal

In the following example, the CRM and HRMS web profiles need to be defined with Domain Name Server (DNS) names that include the same authentication domain as the DNS name of the portal web server. They also each need the Authentication Domain property in their web profiles set to this value.



Web servers that don't have the same server domain as the portal (such as sales.i) can still be used to serve content to the portal. However, cookies set by the portal are not forwarded to these servers. The sales.i server in the example can provide pages and applications to the portal, but it cannot host a PeopleSoft application that supports single signon functionality with the portal.

Related Links

"Understanding WebLogic" (System and Server Administration)

Configuring Web Profiles

This section provides an overview of web profiles and discusses how to configure web profiles.

Related Links

Understanding Changes in Portal Configuration Settings

Understanding Web Profiles

A web profile is a specific configuration of portal-related properties that control or affect the functionality of your PeopleTools portal. These predefined web profiles provide several configurations for use with your PeopleTools portal:

- *DEV*

  This web profile provides basic portal functionality for development, including trace and debug settings that are appropriate for development.

Important! Never run your production system with the DEV web profile. This profile is for development environments and is not tuned for performance.

- *KIOSK*

  This web profile uses the same settings as the PROD web profile, except that public user access is enabled for the Guest user, and all options for storing caching or persistent cookies on the browser are disabled.

- *PROD*

  This web profile uses the settings that are most commonly needed in a production environment that authenticates users.

- *TEST*

  This web profile uses the same settings as the DEV web profile, except that fewer trace and debug properties are enabled.

Additional web profiles might be included as part of PeopleSoft Interaction Hub or an application portal pack.

You can modify the property settings of any web profile via the Web Profile Configuration component, which includes pages for these types of properties:

- General
- Portal security
- Virtual addressing
- Cookie rules
- Caching
- Trace and debug options
- Appearance and character
- Custom properties.

Specifying an Initial Web Profile

As you perform the PeopleSoft Pure Internet Architecture setup procedure for your system, you're prompted to specify which web profile to apply to your portal. You're also prompted for an existing user ID and password. This information is saved in encrypted form which the portal servlet can use to gain secure access to the web profile.

The default web profile is PROD. The default web profile user ID is PTWEBSERVER; the password is set during product installation.

The PTWEBSERVER account provides the portal servlet with minimal security access, sufficient only to launch the portal environment, but without access to any pages or other PeopleSoft Pure Internet Architecture objects. This account uses the PTPT1500 permission list, which is set to never time out, and provides necessary access 24 hours a day, seven days a week.

Note: Your PeopleSoft application might be delivered with the account that you specify here locked. You must unlock it before you can access and configure the web profile, or you will receive an error message about incorrect site configuration on the sign-in page.
You can unlock the web profile account by deselecting the **Account Locked Out** check box on the User Profile - General page. You can alternatively issue the following SQL command against your database (this example uses the PTWEBSERVER account):

```
update PSOPRDEFN set ACCTLOCK=0 where OPRID='PTWEBSERVER'
```

See “Setting General User Profile Attributes” (Security Administration).

You can specify any of the delivered profiles, or you can enter a different profile name. In that case, the portal is set up to use a profile by that name. If a profile by the name that you specify doesn't exist in the database, internal default settings are used until you sign in to the portal and create a profile with that name. The portal then automatically configures itself according to those settings.

As long as no profile exists by the name that you specified during the PeopleSoft Pure Internet Architecture setup, the internal default settings remain in effect, which is indicated on the site sign-in page. When the site is in this mode, every browser request triggers an attempt to load the named profile. Therefore, you shouldn't use the site for extended periods like this.

Once you complete the PeopleSoft Pure Internet Architecture setup, you can then use the Web Profile Configuration component to modify the properties to reflect your settings.

Note: The default internal settings are not the same as the DEV profile. Don't modify any of the delivered profiles so that you always have access to unmodified reference versions. You can make a copy of any profile and modify it, or you can define a new profile instead.

Copying and Deleting Web Profiles

To copy a web profile:

1. Select **PeopleTools** > **Web Profile** > **Copy Web Profile.**
2. Enter or select a web profile name.
3. On the Web Profile Save As page, enter a new name for the web profile in the **To** edit box.

To delete a web profile:

1. Select **PeopleTools** > **Web Profile** > **Delete Web Profile.**
2. Enter or select a web profile name.
3. On the Delete Web Profile page, click **Delete Web Profile.**

Changing the Web Profile After the PeopleSoft Pure Internet Architecture Setup

After the PeopleSoft Pure Internet Architecture setup, to select a different web profile, you must edit the text file that stores this information, called configuration.properties. Each PeopleSoft portal that you set up has its own copy of configuration.properties, located in this directory:

PS_CFG_HOME\webserv*web_server*\applications\peoplesoft\PORTAL\WEB-INF\psftdocs*sitename*

The configuration.properties file includes these properties:

- WebProfile

  Change the value of this property to the name of the web profile that you want to apply to the portal, for example, WebProfile=MYPROFILE.

- WebUserId and WebPassword

  Change these values to specify a different user ID and password for the portal servlet to use to access the web profile. You must encrypt the password by using PSCipher.

  To encrypt the user ID or password:

  1. Run the PSCipher.bat file:

     See "Using the PSCipher Utility" (Security Administration).

     Note: On a UNIX machine, change the PSCipher.sh script file permissions so that you can run the program.

     ---

  2. Copy the encrypted string and paste it into the configuration.properties file replacing the existing value, for example:

     WebPassword=*encrypted_password*

     Important! Make sure that the entire encrypted string, (including all symbol characters), is copied.

After you save the configuration.properties file, restart your web server and the new profile takes effect.

Alternatively, you can use psadmin to specify the web profile that is used. You must restart your web server for the new profile to take effect. For more information about using psadmin, see "Administering a PIA Site" (System and Server Administration)

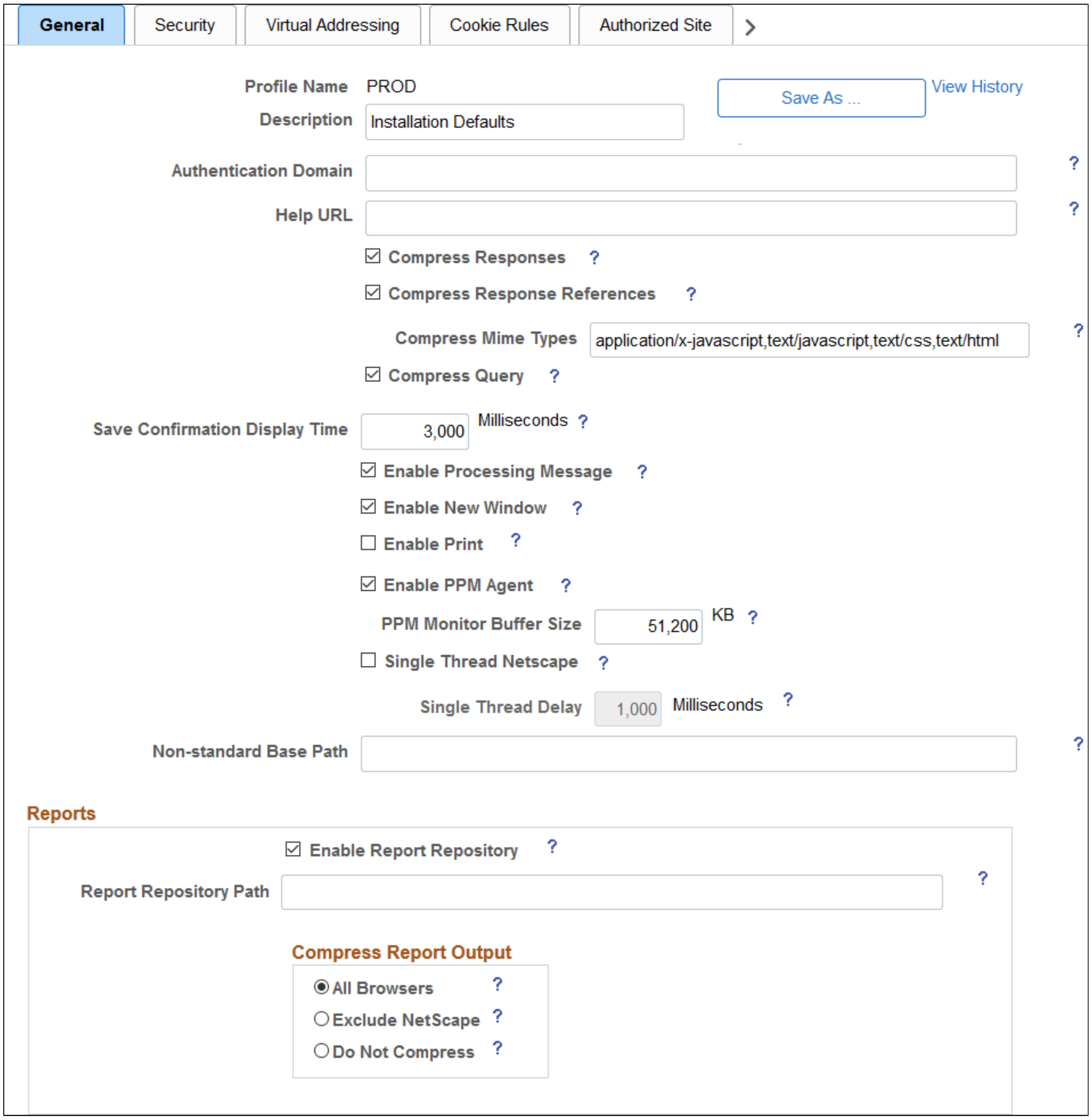
Related Links

"Using the integrationGateway.properties File" (Integration Broker Administration)

Configuring General Portal Properties

Access the Web Profile Configuration page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration**).

This example illustrates the fields and controls on the Web Profile Configuration - General page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Save As	Click to save a copy of the current web profile under a new name. On the Save Web Profile As page, enter a new profile name and click **OK.** Use this button to modify one of the profiles that is delivered with your PeopleSoft application.
View History	Click to access the Web Profile History search page in a new browser window.

Field or Control	***Description***
Authentication Domain	Enter the name of the extended authentication domain in which the portal is running, starting with a leading period. This value overrides, but must be compatible with, the base-level authentication domain. For example, if you entered *.example.com* during the PeopleSoft Pure Internet Architecture setup, only values such as *.us.example.com* and *.fr.example.com* are valid. An authentication domain is required for a variety of portal functions. For example, if a cookie is shared on web servers srv1.example.com and srv2.example.com, you must specify an authentication domain of *.example.com.* This field requires a value if you specified a default addressing server name on the Web Profile Configuration - Virtual Addressing page. You must qualify that server name with this domain name. The default value of the authentication domain is the one that you specified during the PeopleSoft Pure Internet Architecture setup. That value is stored as the web server session cookie domain and kept in an internal web server file. For Oracle WebLogic, that file is weblogic.xml. The preferred method for changing that value is to rerun the website setup. **Note:** The value that you enter in this field is automatically converted to lowercase. **Note:** You must enter your authentication domain to enable all functionality associated with PeopleSoft Related Content Framework.

Field or Control	*Description*

Help URL	Specify the URL that points to an installation of PeopleSoft online help. When users click the Help link, the appropriate context-sensitive PeopleSoft documentation is displayed. To remove the Help link from all application pages, omit this value.
	
	You can point to either an installation on Oracle's hosted documentation website or to a local installation of PeopleSoft online help:
	
	• To point to an installation on Oracle's hosted documentation website, construct a URL similar to the following:
	
	`https://docs.oracle.com/pls/topic/lookup?id=%CONTEXT_ID%&ctx=`*`library_ID`*
	
	For example:
	
	`https://docs.oracle.com/pls/topic/lookup?id=%CONTEXT_ID%&ctx=pt858latest`
	
	For more information on available library IDs and configuring the help URL to point to an installation on Oracle's hosted documentation website, see Configuring Context-Sensitive Help Using the Hosted Online Help Website.
	
	• To point to a local installation of PeopleSoft online help, construct the URL similar to the following:
	
	`http://`*`webserver`*`:`*`port`*`/help_folder/help.html?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%`
	
	For example:
	
	`http://myserver.example.com:8080/help-PB/help.html?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%`
	
	For more information on configuring the help URL to point to a local installation of PeopleSoft online help, see Configuring Context-Sensitive Help Using a Locally Installed Online Help Website.

Field or Control	*Description*
	Note: This setting applies only to the Help link displayed on PeopleSoft pages accessed through a browser. Use the PeopleTools Options page to configure F1 help for Application Designer and other Windows client applications. See “PeopleTools Options” (System and Server Administration) for more information.
Compress Responses	Select to enable compression in the communication between the web server and the browser. Gzip and Compress protocols are supported. This check box is selected by default.
Compress Response References	Select to enable compression of cache files that are delivered from the web server to the user's browser. Only cache files with the Multipurpose Internet Mail Extensions (MIME) types that are specified in the **Compress MIME Types** text box are compressed. Gzip and compress protocols are supported. This check box is deselected by default.
Compress MIME Types	Specify the MIME types of the cache files to be compressed as a comma-separated string. This field is available only if you selected the **Compress Response References** check box. The default value is *application/x-javascript,text/javascript, text/css,text/html.*
Compress Query	This property applies to browser requests in which the content type section of the URL is /q/, indicating a query. Select to enable compression of query responses to the browser. This check box is selected by default.
Save Confirmation Display Time	Specify in milliseconds how long the save confirmation image should appear for the user if the save confirm personalization option is enabled. The default value is *3000* milliseconds (3 seconds).
Enable Processing Message	Select to enable processing notification while the system processes a request. This check box is selected by default.

Field or Control	*Description*
Enable New Window	Enables or disables the New Window link in the pagebar site wide: • Select to enable the New Window link in the pagebar for components that have the pagebar enabled and that have the New Window option selection. • Deselect to disable the New Window link site wide. This check box is selected by default.
Enable Print	Enables or disables the Print link in the pagebar site wide: • Select to enable the Print link in the pagebar for components that have the pagebar enabled. • Deselect to disable the Print link site wide. This check box is deselected by default.
Enable PPM Agent (enable PeopleSoft Performance Monitor agent)	Select to activate the PeopleSoft Performance Monitor agent for this portal. The performance monitoring system has *agents* and *monitors.* An agent captures units of data (PMUs), and a monitor views and analyzes agent flows. This check box controls whether the agent is active on sites that use this profile. This check box is selected by default.
PPM Monitor Buffer Size (PeopleSoft Performance Monitor buffer size)	Specify the maximum size, in kilobytes (KB), of the Monitor Server buffer. Data is discarded once this limit is reached. This field is available only when **Enable PPM Agent** is selected. The default value is *51200* KB (50 megabytes).
Single Thread Netscape	Select to indicate that requests from a Netscape browser should be single-threaded to prevent crashes. Older versions of Netscape browsers had problems with multithreading responses and would crash. Select this check box if your users are using older versions of Netscape browsers and are experiencing browser crashes. This check box is deselected by default. When selected, the **Single Thread Delay** option appears.
Single Thread Delay	Specify a delay, in milliseconds, for single-threaded Netscape requests. This field is available only when **Single Thread Netscape** is selected. The default value is *1000* milliseconds (one second).

Field or Control	*Description*
Non-standard Base Path	In the rare circumstance that you receive an error message that the base physical path has not been set, you must enter the location of the directory that contains the signon.html file for your application. This field is blank by default.

See Viewing Web Profile History, Understanding the Authentication Domain, Configuring Caching, Configuring Virtual Addressing, “Working with Performance Monitor Web Profile Properties” (Performance Monitor), “Setting Internet Properties” (Application Designer Developer’s Guide), and “Using the Pagebar” (Applications User’s Guide).

See *PeopleSoft 9.2 Application Installation* for your database platform.

Reports

Field or Control	*Description*
Enable Report Repository	Select to enable users to view the report repository. When this check box is deselected, no report files can be viewed by users. This check box is selected by default.
Report Repository Path	(Optional) Specify a drive and directory path to indicate where the site that is served by this web profile should look for generated reports. This value overrides the location that is specified during the PeopleSoft Pure Internet Architecture setup; you can leave this field blank to use the original value. **Note:** Changing the report repository path prevents you from viewing existing reports unless the old subdirectory structure is moved to the new location. Be careful when editing this field. **Note:** (For Microsoft Windows only) If the report repository resides on a different machine from the web server, specify the report repository path in UNC format instead of mapped drive format. For example, if the report repository is on MachineX in folderY, the Report Repository Path should be specified as: //MachineX/folderY/psreports or \\\\MachineX\\folderY\\psreports If MachineX/folderY is mapped to the Y: drive, do not specify the Report Repository Path as Y:\psreports.

Field or Control	***Description***
Compress Report Output	Specify how reports should be compressed: *All Browsers:* Reports are always compressed. This is the default value. *Exclude Netscape:* Reports are compressed, except for users who are using Netscape Navigator. *Do Not Compress:* Reports are never compressed.

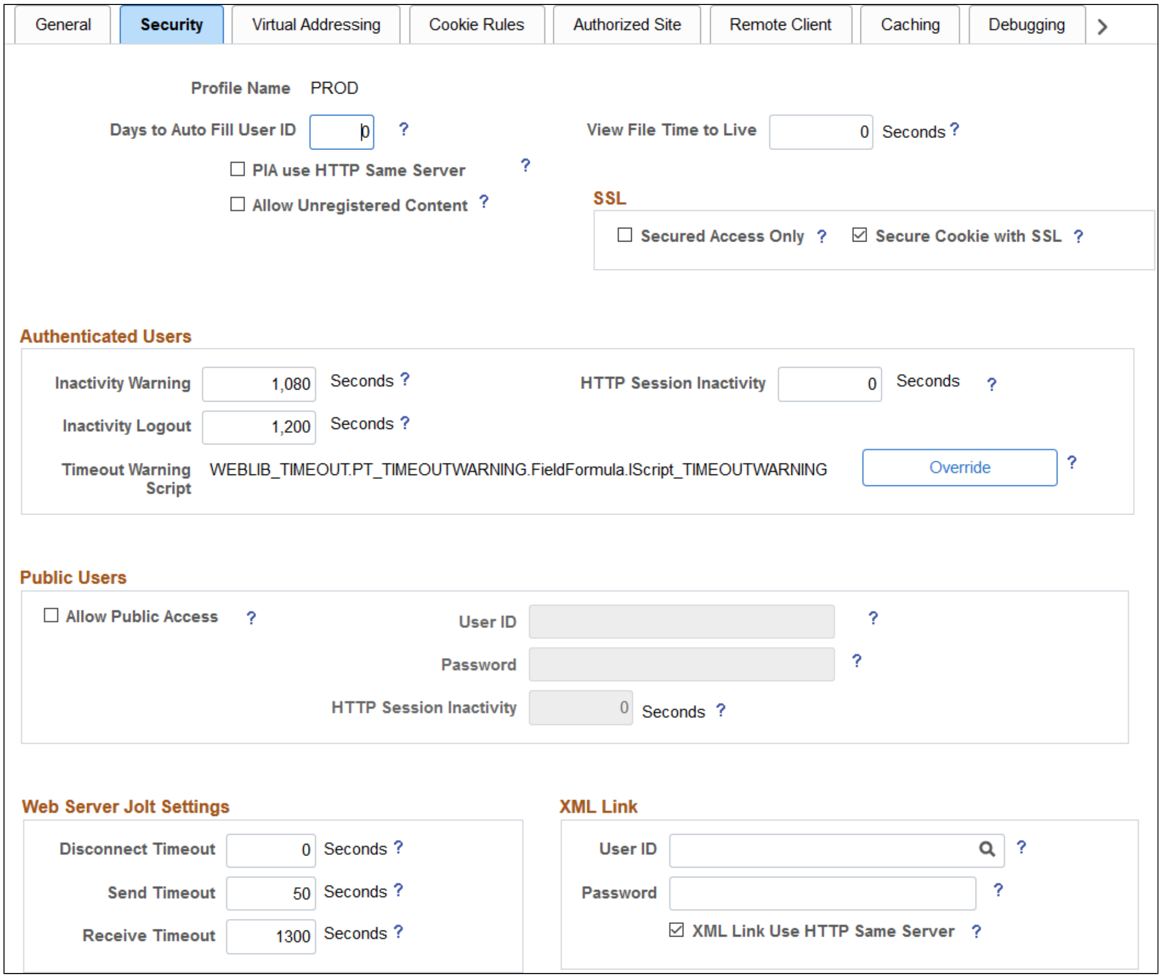
Related Links

“AuthTokenDomain” (PeopleCode API Reference)
“Understanding WebLogic” (System and Server Administration)

Configuring Portal Security

Access the Web Profile Configuration page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Security tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Security page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Days to Autofill User ID	If this field is populated, the system caches the user ID and automatically inserts it on the sign-in page. This a convenience for users. This feature is implemented through the use of a stored cookie on the browser. The default value is *7* days. Specify a value of *0* to disable this feature. Use *0* in a public area or kiosk situation, or if your security policy doesn't allow cookies stored on the browser.

Field or Control	***Description***
View File Time to Live	Specify, in seconds, how long the portal should wait after sending a file attachment to a user's browser before removing that file from the web server storage. If this value is *0,* then the amount of time that the file remains on the web server is always greater than or equal to one minute. The specific amount of time beyond one minute depends on the size of the file. If the value is *greater than 0,* then the amount of time that the file remains on the web server depends, approximately, only on the value of that property in seconds. The default value is *0.*
PIA use HTTP Same Server (PeopleSoft Pure Internet Architecture use HTTP same server)	Select to specify that the portal should use the HTTP protocol instead of HTTPS for requests that are issued by the portal for content that is hosted on the same server as the portal servlet. Setting this property is necessary when the portal web server is behind an SSL accelerator or when SSL is terminated on a device in front of the portal web server, such as a reverse proxy server, or between different sites that are on the same web server. You can also use this property to improve the performance of homepage pagelets that are provided by the PeopleSoft Pure Internet Architecture servlet that run on the same web server as the portal, and from which the web server receives SSL requests. That is, SSL has not been terminated by a device in front of the web server. You must also specify the default addressing protocol and port on the Web Profile Configuration - Virtual Addressing page. See Using SSL Accelerators.

Field or Control	*Description*
Allow Unregistered Content	Select to instruct the portal to serve both registered and unregistered content. The portal generally allows any external content links to be wrapped with the portal header and navigation frames. Deselecting this option prevents content references from being displayed in the portal unless they are explicitly registered. This check box is not selected by default. **Note:** If **Allow Unregistered Content** is not selected, the portal won't wrap a content link if it is not registered in the portal registry or if it isn't coming from a registered node. The portal checks whether the content link is from a registered node by checking whether the content link starts with the URI specified in the **Content URI Text** or the **Portal URI Text** in the Node Definition page, on the Portal tab. For security purposes, the portal checks only against the **Content URI Text** and **Portal URI Text** if it contains text that is longer than 12 characters.

SSL (Secure Sockets Layer)

Field or Control	*Description*
Secured Access Only	Select to enforce SSL if the entire website requires the SSL protocol. This prevents users from using non-SSL protocols to access any link within this website or application. If only some pages require SSL access, deselect this check box (the default setting).

Field or Control	*Description*
Secure Cookie with SSL	Select to prevent the single sign on token from traveling over an insecure network. If you select this check box and the scheme of the current request is HTTPS (an SSL server), the system sets the secure attribute of the single sign on cookie (PS_TOKEN) to True. PeopleSoft single sign on functionality also applies at the web server level. For example, suppose that you have two web servers: server X and server Y. Web server X is an SSL site, and web server Y is not. In these situations, many sites want server Y to trust the authentication token, PS_TOKEN, issued by server X. This requires that you set the PS_TOKEN to be secure. If the PS_TOKEN is not secure, when a user signs in through server Y, the browser sends PS_TOKEN to server Y over the unencrypted, non-SSL link. This is typical behavior for browsers when dealing with cookies that aren't secure. Potentially, in this situation, a hacker could obtain this token from the clear network and use it to sign in to the SSL-secure server X. Another important use of this property relates specifically to the PeopleSoft portal. When the portal uses a proxy server to relay content with an HTML template, it should forward only PS_TOKEN cookies that are marked secure over SSL connections. **Note:** By selecting this check box, you effectively disable single sign on functionality with any non-SSL servers. If, at your site, you want users to sign in to an HTTPS server, and then want to use single signon functionality with HTTP servers, deselect this check box to enable single signon functionality between HTTPS and HTTP servers. **Note:** Before you deselect this check box, make sure that you are aware of all the security implications, such as the security of the HTTPS server being compromised. This check box is selected by default.

Authenticated Users

When single signon has been set up, the portal web server and any content provider web servers should have the same values for the following fields:

- Inactivity Warning

- Inactivity Logout
- HTTP Session Inactivity

Field or Control	***Description***
Inactivity Warning	Specify the number of seconds that the portal should wait before warning users that the browser session will expire. Users can continue with the current session by clicking the **OK** button in the warning message, which resets the inactivity timer. If a user doesn't respond within the inactivity logout interval, the session ends and the expired connection page appears. The default value is *1080* seconds (18 minutes).
HTTP Session Inactivity	Specify the number of seconds of inactivity after which the HTTP session times out for authenticated users. When the interval passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session. From the user experience, the user may be put in a search page or home page because the previous session state is lost at this point. This property is unaffected by any permission list setting. When this property isn't specified, the HTTP session time-out interval is the same value as the authenticated user inactivity logout property that is specified on this page. This property is similar to the public users **HTTP Session Inactivity** property that is specified on this page. Use this property with caution. Don't use this property if you don't understand its usage.

Field or Control	*Description*
Inactivity Logout	Specify the value in seconds of the inactivity time-out interval that applies to PeopleSoft applications to which a user is signed in. When the interval passes with no user activity, the user's browser displays the page that is specified by the **Expire Page - Page** field on the Web Profile Configuration - Look and Feel page. The web server applies this value in JavaScript. **Note:** The inactivity logout can be overridden by a permission list inactivity time-out setting. The default value is *1200* seconds. **Note:** Depending on the application implementation, authenticated users might also experience an HTTP session inactivity time-out, which by default is the same as the inactivity logout value that you specify here. However, you can override this behavior by using the **HTTP Session Inactivity** property on this page. This property independently controls the HTTP session inactivity time-out. The **HTTP Session Inactivity** property is unaffected by any permission list setting.
Timeout Warning Script	Displays the name of the time-out warning script that is currently in effect. The default value is *WEBLIB_TIMEOUT.PT_TIMEOUTWARNING.FieldFormula.IScript_TIMEOUTWARNING.*
Override	Click to change the time-out warning script. The Override Timeout Warning Script page appears, with these fields: • **Record (Table) Name:** The default value is *WEBLIB_TIMEOUT.* • **Field Name:** The default value is *PT_TIMEOUTWARNING.* • **PeopleCode Event Name:** The default value is *FieldFormula.* • **PeopleCode Function Name:** The default value is *IScript_TIMEOUTWARNING.* The values of these fields are concatenated to produce the value in the **Timeout Warning Script** field.

See “Setting General Permissions” (Security Administration).

Public Users

Public users are PeopleSoft application users who are not required to go through a PeopleSoft sign in page. As a result, public users are neither identified nor authenticated by their own user ID and password. Public users also are not forwarded to the sign in page after any period of inactivity.

Because they access sites without credentials, public users should be limited to informational websites where sensitive data is not accessible.

You should enable public user access under these conditions:

- When you provide users with direct links to pages that are not secured.

  Note: If a user clicks a direct link that accesses a page for which the public user ID is not authorized, the system displays the sign-in page prompting the user to supply her PeopleSoft credentials.

- When you use an external authentication method.

  Important! If you are using an external authentication method in conjunction with the PeopleCode *SwitchUser* function, then the **Inactivity Warning** and **Inactivity Logout** values specified for authenticated users are restored.

  See “SwitchUser” (PeopleCode Language Reference).

- When you want crawlers to search your PeopleSoft site and need to provide access to the portal.

Field or Control	***Description***
Allow Public Access	Select so that the system bypasses the user sign-in page when users click a direct link to a page. The system uses the values that you specify in the **User ID** and **Password** fields for authentication. This check box is deselected by default. When you select this check box, the system enables the **User ID, Password,** and **HTTP Session Inactivity** fields.
User ID	Enter the account that the system uses to authenticate public users. This field is enabled when you select the **Allow Public Access** check box. **Note:** This is *not* the same as the WebUserId property in the configuration.properties file, which you specify during the PeopleSoft Pure Internet Architecture installation for accessing the web profile.

Field or Control	*Description*
Password	Enter the password for the account that you entered in the **User ID** field. This field is enabled when you select the **Allow Public Access** check box Passwords can include up to 64 characters. **Note:** This is *not* the same as the WebPassword property in the configuration.properties file, which you specified during the PeopleSoft Pure Internet Architecture installation for accessing the web profile.
HTTP Session Inactivity	Specify the value in seconds of the inactivity period that applies to public users. When the period passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session. This field is enabled when you select the **Allow Public Access** check box. **Note:** If you are using an external authentication method in conjunction with the SetAuthenticationResult function in Signon PeopleCode or the SwitchUser() function in Non-Signon PeopleCode, you are logged out if you are inactive. The default value is *1200* seconds. See "SwitchUser" (PeopleCode Language Reference).

Web Server Jolt Settings

Field or Control	*Description*
Disconnect Timeout	Specify the amount of time to wait while disconnecting the Oracle Jolt connection. A value of *0* means no limit. The default value is *0* seconds.

Field or Control	***Description***
Send Timeout	Specify the maximum number of seconds that the servlet allows for a request to be sent to the application server. This setting does not indicate a maximum amount of time for the service to finish; it indicates only the maximum amount of time to send the request to the application server. The default value is *50* seconds.
Receive Timeout	Specify the maximum number of seconds that the servlet should wait for a response from the application server. Make sure that the value of this field is greater than your application server service time-outs, such as the Service Timeout setting for PSAPPSRV that appears in the psappsrv.cfg configuration file on the application server. The default value is *600* seconds.

XML Link

Field or Control	***Description***
User ID	Enter the user account that the system uses to authenticate users for XML Link technology.
Password	Enter the password for the account name that you entered in the **User ID** field.
XML Link Use HTTP Same Server	Select to indicate that the portal should use the HTTP protocol instead of HTTPS for requests that are issued by the xmllink servlet for content that is hosted on the same server as the xmllink servlet. This check box is selected by default.

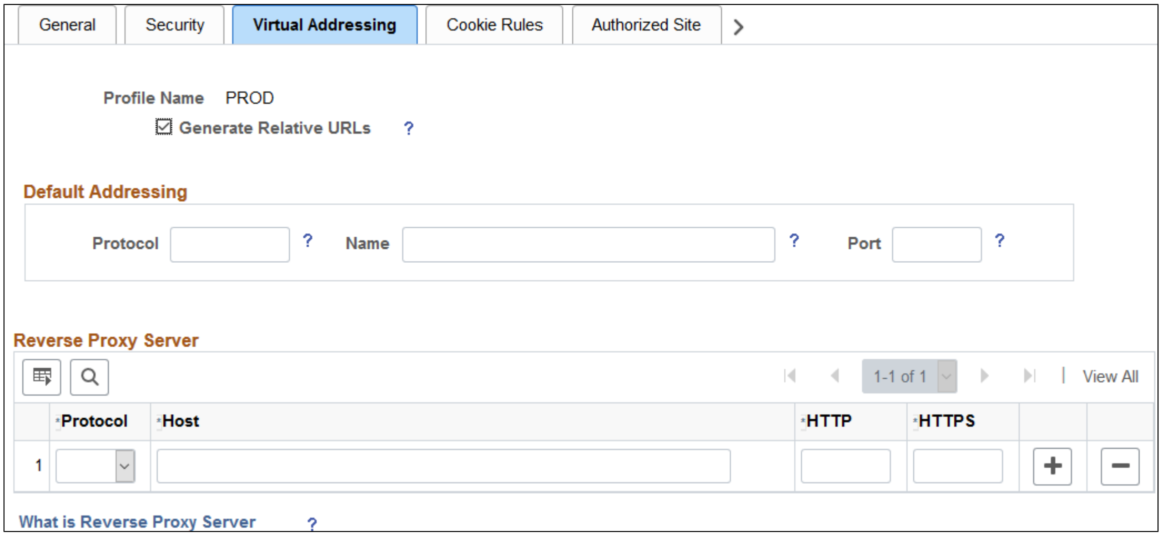
Related Links

Improving Same-Server Performance Under SSL
"Understanding WebLogic" (System and Server Administration)

Configuring Virtual Addressing

Access the Web Profile Configuration page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Virtual Addressing tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Virtual Addressing page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Generate Relative URLs	Select to generate relative URLs. This setting is for use with proxy server implementations. This check box is selected by default.

Default Addressing

Field or Control	***Description***
Protocol	Specify the default protocol—*HTTP* or *HTTPS*—to override the protocol that is used by the PeopleSoft Pure Internet Architecture and the portal to construct URL references. When this property is omitted, the protocol of the incoming request to the PeopleSoft Pure Internet Architecture servlet is used. You must set this property when the browser makes an SSL connection and SSL is terminated before the PeopleSoft Pure Internet Architecture servlet, using a device such as an SSL accelerator or a reverse proxy server, or when you select the **PIA use HTTP Same Server** property on the Web Profile Configuration - Security page. In this situation, the protocol of the request to the PeopleSoft Pure Internet Architecture servlet is HTTP, but the PeopleSoft Pure Internet Architecture must generate URL references with an HTTPS protocol.

Field or Control	*Description*
Name	Enter a default server name to override the server name that is used by the PeopleSoft Pure Internet Architecture and the portal to construct URL references. When this property is omitted, the server name of the incoming request to the PeopleSoft Pure Internet Architecture servlet is used to construct URL references. You must set this property when the server on which the PeopleSoft Pure Internet Architecture servlet resides is different from the server to which the browser is connected. This can occur when a reverse proxy server or load balancer is configured in front of the web server. In these situations, the PeopleSoft Pure Internet Architecture must generate URL references with the server that is used by the browser for its initial connection. **Note:** If you specified an authentication domain on the Web Profile Configuration - General page, it must be part of the server name that you specify here. The value that you enter in this field is automatically converted to lowercase.
Port	Enter a default port to override the port that is used by the PeopleSoft Pure Internet Architecture and the portal to construct URL references. When this property is omitted, the port of the incoming request to the PeopleSoft Pure Internet Architecture servlet is used to construct links on a PeopleSoft Pure Internet Architecture or portal page. You must set this property when the port through which the PeopleSoft Pure Internet Architecture servlet is accessed is different from the port to which the browser is connected. This can occur when an SSL accelerator is used or when a reverse proxy server is configured in front of the web server, and the reverse proxy server or load balancer listens on a different port from the web server, or when you select the **PeopleSoft Pure Internet Architecture use HTTP Same Server** property on the Web Profile Configuration - Security page. In these situations, the PeopleSoft Pure Internet Architecture must generate URL references with the port that is used by the browser for its initial connection.

Reverse Proxy Server List

Use this grid to specify the reverse proxy servers through which the portal can expect to retrieve content. External content that is retrieved from these sources and contains relative references is rewritten by the portal to contain relative references instead of absolute references to preserve reverse proxy server requirements.

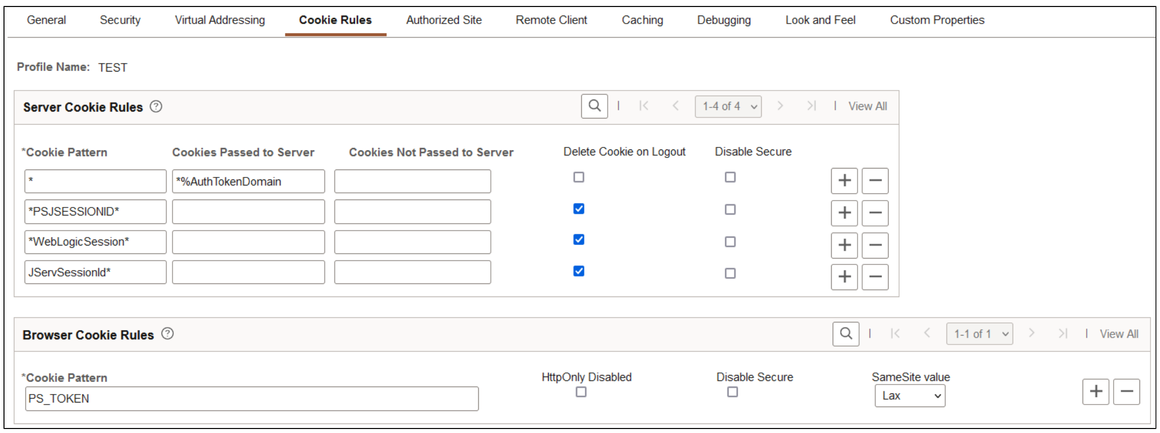
For each reverse proxy server on the list, you specify:

- The protocol to use.
- The server's host or machine name.
- The server's HTTP port number.
- The server's HTTPS port number.

Configuring Cookie Rules

Access the Web Profile Configuration page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Cookie Rules tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Cookie Rules page. You can find definitions for the fields and controls later on this page.



This page defines rules that determine how the portal passes cookies to other servers and browsers. Use the respective grids to define which cookies are passed, and how. Both server and browser cookie names are case sensitive.

Oracle WebLogic web servers use browser cookies (containing the server path and domain) to establish session identity. These cookies have a default name that's used to retrieve the cookie on each request to the web server. In an environment in which multiple web servers are in use with the portal, you need to define unique session cookie names between web servers to prevent one cookie from overwriting another cookie of the same name set by a different web server. The path or domain of the cookie must be different to prevent overwriting.

Server Cookie Rules

Note: Use the first three fields to specify patterns to match, where the asterisk (*) is treated as a wildcard, matching zero or more characters.
The portal knows its own domain (from the authentication domain), and the first default rule instructs it to pass portal servlet cookies to any server in the same domain, over both secure and insecure connections. All other cookies can continue to be passed to any server on any domain.

Field or Control	*Description*
Cookie Pattern	Specify the name pattern for cookies to which this rule applies. For example, to specify all cookies whose names start with *HR_,* enter the value *HR_**.
Cookies Passed to Server	Specify the name pattern for the servers to which the specified cookies should be passed. For example, to specify all servers in the domain .hr.example.com, enter the value **.hr.example.com.* You can use a substitution parameter %AuthTokenDomain to represent the authentication domain currently in effect.
Cookies Not Passed to Server	Specify the name pattern for the servers to which the specified cookies should not be passed. You can use a substitution parameter %AuthTokenDomain to represent the authentication domain currently in effect. **Note:** This field takes precedence over the **Cookies Passed to Server** field.
Delete Cookie on Logout	Select to indicate that the cookie should be deleted when the user signs out of the portal. **Note:** This check box is deselected for the first default rule (matching all cookies), but any cookies that are generated by PeopleSoft with, for example, the PSJSESSIONID prefix, are deleted when the user signs out. If you're running multiple PeopleSoft portals and want cookies to be maintained after the user signs out, update this page to specify which cookies to preserve or delete. ***Warning!*** Oracle recommends that you use extreme caution when setting all cookies to be deleted on sign-out. Cookie pattern "*" means all cookies. Some cookies must be kept. For example, the Arrow Point cookie (ARPT) is used by some load balancers to ensure sticky http(s) sessions.
Disable Secure	Cookies are sent secure by default over secure (HTTPS) connections. Select this setting to disable the secure attribute for this cookie, which allows it to be sent in requests over insecure (HTTP) connections. This setting applies only when HTTPS is being used.

Browser Cookie Rules

Use the Browser Cookie Rules grid to define exceptions to the default settings for browser cookies.

Unless an exception rule is defined in this Browser Cookie Rules grid, PeopleSoft cookies are delivered as HTTP-only by default. For browsers that support this option, this prevents client side scripts such as JavaScript from accessing a PeopleSoft cookie. Also, unless an exception rule is defined in this Browser Cookie Rules grid, PeopleSoft cookies are delivered as Secure by default in HTTPS mode This setting applies only when HTTPS is being used. The secure attribute is always disabled for all cookies when HTTP is being used. FInally, unless an exception rule is defined in this Browser Cookie Rules grid, PeopleSoft cookies are delivered with a SameSite value of Strict.

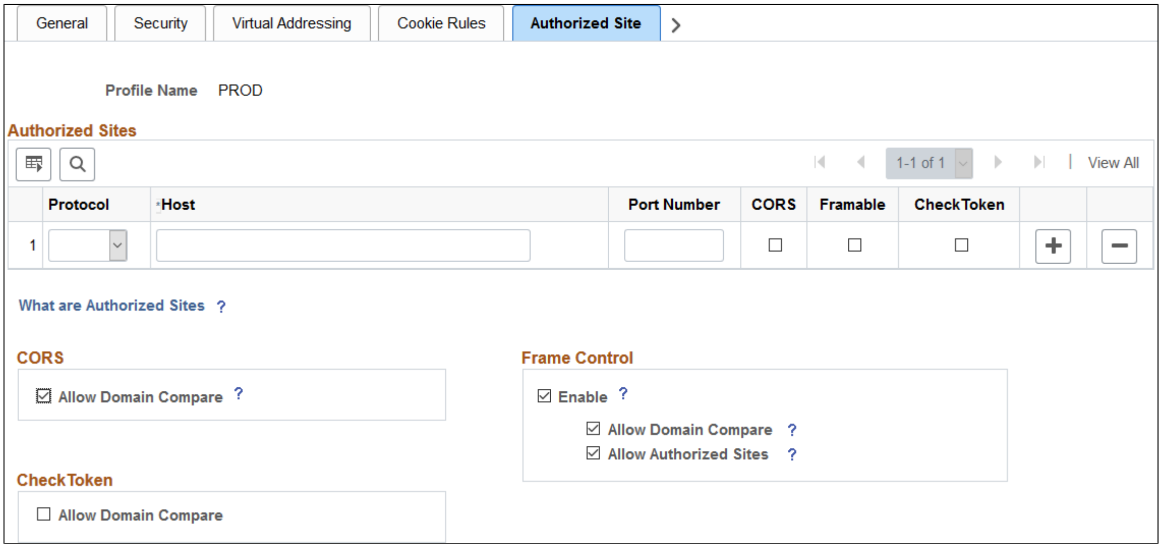
Note: The following cookies cannot be controlled using the web profile configuration: PS_TOKENEXPIRE, PS_DEVICEFEATURES, PS_LOGINLIST, http*refresh, IOS_FULLSCREEN, ps_theme, and psback.

Field or Control	***Description***
Cookie Pattern	Specify the name pattern for cookies to which this rule applies. For example, to specify all cookies whose names start with *HR_,* enter the value *HR_**. For browser cookies only, if a cookie is identified with both an absolute name and name via a wildcard, the cookie rule using an absolute name is given precedence. For example, a rule for PS_LASTSITE is given precedence over a rule for PS_*SITE.
HttpOnly Disabled	Select to indicate that the cookie can be accessed by JavaScript from the browser side.
Disable Secure	Cookies are sent secure by default over secure (HTTPS) connections. Select this setting to disable the secure attribute for this cookie, which allows it to be sent in requests over insecure (HTTP) connections. This setting applies only when HTTPS is being used.
SameSite value	Select the SameSite value • *Strict* – Send the cookie in first-party contexts only; the cookie is not sent in requests initiated by third-party websites. **Note:** Strict is the default for all PeopleSoft cookies. • *Lax* – Do not send the cookie on normal cross-site sub-requests (for example to load images or frames into a third party site); however, do send the cookie when a user is navigating to the originating third-party site (for example, when following a link). • *None* – Send the cookie in all contexts: in response to both first-party and third-party requests. The Secure attribute must also be set when the SameSite attribute is set to None; otherwise, the browser will reject the cookie.

Configuring Authorized Sites

Access the Authorized Site page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Authorized Site page tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Authorized Site page. You can find definitions for the fields and controls later on this page.



Use the Authorized Site page to maintain sites that are authorized to request resources from this web server for three mutually independent purposes:

- Allowing other sites to make requests to this site using the Cross-Origin Resource Sharing (CORS) standard.
- Allowing other sites to embed content from this site—that is, allow sites to request “framable” content.
- Allowing other sites to participate in a single signon configuration with this site (using a check token ID).

The Authorized Sites grid serves as an “allowlist” identifying only those hosts, domains, or subdomains that are allowed to request resources from this web server for only the specific purpose or purposes specified. To facilitate configuration of these features within your own intranet, you can use the **Allow Domain Compare** check box on a feature-by-feature basis:

Field or Control	***Description***
Allow Domain Compare	Select this check box to authorize all other hosts within the same authentication token domain as this web server. **Note:** When this check box is deselected, you must create individual host, domain, or subdomain entries to authorize other hosts within the same authentication token domain.

Authorized Sites Grid

Note: A site or domain can be added to the grid one or more times.

Field or Control	***Description***
Protocol	Select a protocol (*http* or *https*) to restrict requests from this host to that specific protocol.
Host	Enter a fully qualified host name (for example, **host.example.com**), a domain name (for example, **.example.com**), or a subdomain name (for example, **.us.example.com**). **Note:** Domain and subdomain names begin with a **.** (a period); host names do not.
Port Number	Specify a port number to restrict requests from this host to that specific port number.
CORS	Select this check box to add this entry to the allowlist for CORS processing.
Framable	Select this check box to add this entry to the allowlist to allow the site to embed content from this web server.
CheckToken	Select this check box to add this entry to the allowlist for participating in single signon.

Cross-Origin Resource Sharing (CORS)

You can maintain which sites are authorized to request resources from this web server using the CORS standard. For example, one resource (a web page or a script, for example) makes a cross-origin HTTP request when it requests a resource (an image, a CSS style sheet, a script, and so on) from a different domain than the one which served itself. For security reasons, browsers restrict cross-origin HTTP requests initiated from within scripts. The CORS standard gives web servers cross-domain access controls, which enable secure cross-domain data transfers.

To be allowed to make requests to this site using the CORS standard, another site must either be explicitly identified on this page, have its domain or subdomain explicitly listed on this page, or be implicitly included because it is a member of the same domain as this web server when Allow Domain Compare is selected.

For more information on CORS, see HTTP access control (CORS).

Note: If CORS is not involved in the request—that is, a simple request that does not include an Origin request header, then the request is processed as a non-CORS request and the settings on this page are ignored.

Framable

Whether this web server will allow its content to be framed by the requesting site depends on the combination of check boxes selected.

Field or Control	*Description*
Enable Frame Control	Select this check box to turn on control of which sites can embed content from this web server. **Important!** When this option is deselected, frame control is disabled entirely, which means that any site can embed content from this web server. Any other check boxes related to framing are ignored, even if they are already selected.
Allow Domain Compare	Select this check box to authorize all hosts within the same authentication token domain as this web server.
Allow Authorized Sites	Select this check box to authorize any hosts in the Authorized Sites Grid with the Framable check box also selected to embed content from this web server.

For more information on X-Frame-Options, see HTTP Header Field X-Frame-Options.

Embedding PeopleSoft Content

Note the following limitations on embedding and accessing PeopleSoft content on a non-PeopleSoft system:

- The PeopleSoft web server and the non-PeopleSoft web server must reside within the same subdomain.
- The user accessing the PeopleSoft content must already be signed into the PeopleSoft system in the same browser session.
- PeopleSoft content is limited what is available via the psc servlet with these additional restrictions:
  - Only classic components are supported. No fluid components or features are supported.
  - The features on classic pages that are supported include standard component search, autocomplete, file attachments, and modal windows.
  - The features on classic pages that are not supported include items in the portal header and the page bar (notifications, global search, breadcrumbs, new window, and so on).
  - In addition, classic homepages, interwindow communication, dashboards, WorkCenters, and activity guides are not supported.

Use the following procedure to embed PeopleSoft content on a non-PeopleSoft system:

1. On the PeopleSoft web profile configuration, on the Virtual Addressing page, deselect the Generate Relative URLs check box.

2. On the Authorized Site page, authorize the non-PeopleSoft system to embed content from this site.

3. Then stop and restart the PeopleSoft web server.

4. On the non-PeopleSoft system, modify the HTML for a page that embeds PeopleSoft content to add a reference to a required iScript in the <head> element:

```
<head>
<script type="text/JavaScript" src="http://ps_webserver.example.com:8080/psc/S⇒
ITE/PORTAL/NODE/s/WEBLIB_PTHDR.ISCRIPT1.FieldFormula.IScript_GET_PORTAL_JS"></⇒
script>
</head>
```

5. Add a frame that embeds the PeopleSoft classic component using the following HTML attributes:

Attribute	***Value***
id	`'ptifrmtgtframe'`
name	`'TargetContent'`
src	The full URL to the target component.

For example:

```
<iframe id="ptifrmtgtframe" name="TargetContent" title="Main Content" framebor⇒
der="2" scrolling="auto" src="http://ps_webserver.example.com:8080/psc/SITE/PO⇒
RTAL/NODE/c/MENU_NAME.COMPONENT_NAME.GBL?PAGE=PAGE_NAME" width="100%" height="⇒
100%" style="border-color:red;" ></iframe>
```

CheckToken

Each PeopleSoft database participating in single signon must define a check token ID on the default local node definition on the local database, as well as on the remote node definitions for each participating default local node on all participating systems. When a request is sent from a site where PS_TOKEN originated to another site where it is consumed, an additional validation request is sent by that consumer to verify that the token is still valid at the originator site using this check token ID.

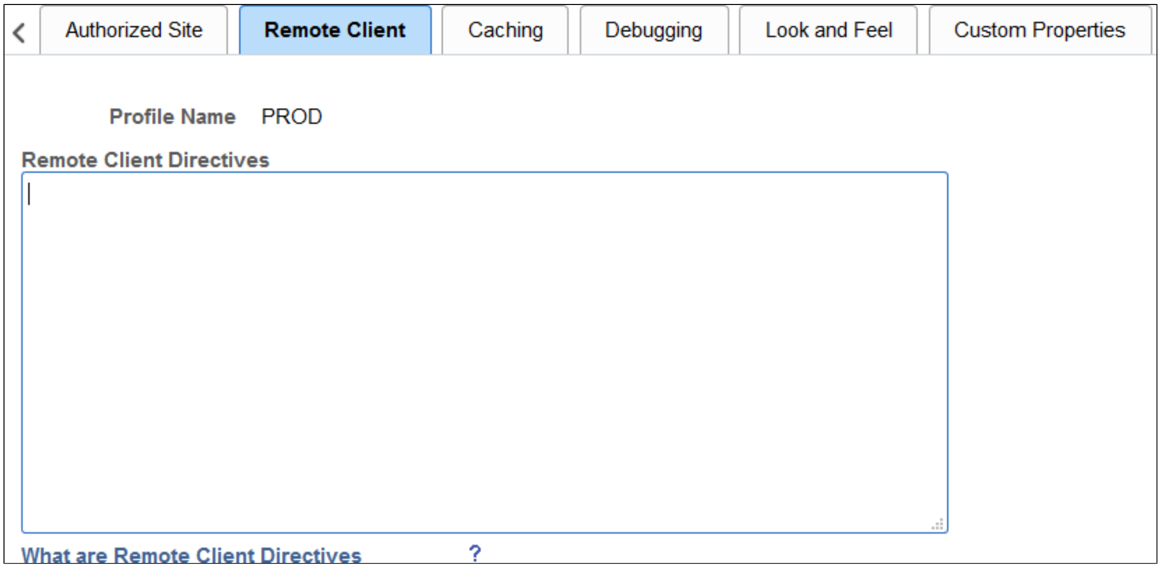
To participate in a single signon configuration, another site must either be explicitly identified on this page, have its domain or subdomain explicitly listed on this page, or be implicitly included because it is a member of the same domain as this web server when Allow Domain Compare is selected.

For more information, see “Understanding PeopleSoft-Only Single Signon” (Security Administration) and “Defining Nodes for PeopleSoft-Only Single Signon” (Security Administration)

Configuring Remote Client Directives

Access the Remote Client page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Remote Client page tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Remote Client page. You can find definitions for the fields and controls later on this page.



Use the Remote Client page to define directives used by the PeopleSoft system to identify and log remote client addresses.

When a remote client makes a request to a PeopleSoft system, it is rare that the remote client connects directly to the PeopleSoft system. Instead, multiple intermediate machines such as firewalls, proxy servers, load balancers, routers, and so on handle the request on behalf of the actual remote client. Any of these machines may forward the request after putting their own address in place of the actual remote client's address. When this address substitution takes places, then that network machine must put the actual remote client address elsewhere in the packet usually in an HTTP request header. If it does not do this, then the actual remote client address is lost to downstream hosts. When this occurs, the address recorded by the PeopleSoft system is the HTTP request's last hop, which is frequently a load balancer, reverse proxy server, or other box within your intranet.

The Remote Client page allows you to configure your PeopleSoft web server with directives that will allow it to more accurately identify and log the actual remote client address from the requests that it receives. These directives include:

- **RemoteIPHeader** – Identifies which HTTP request header is used by the systems in your internal networks.
- **RemoteIPSeparator** – Identifies the character used to separate addresses appended to the HTTP request header.
- **TrustedProxies** – Identifies the list of machines on your internal, controlled networks through which inbound requests can pass.

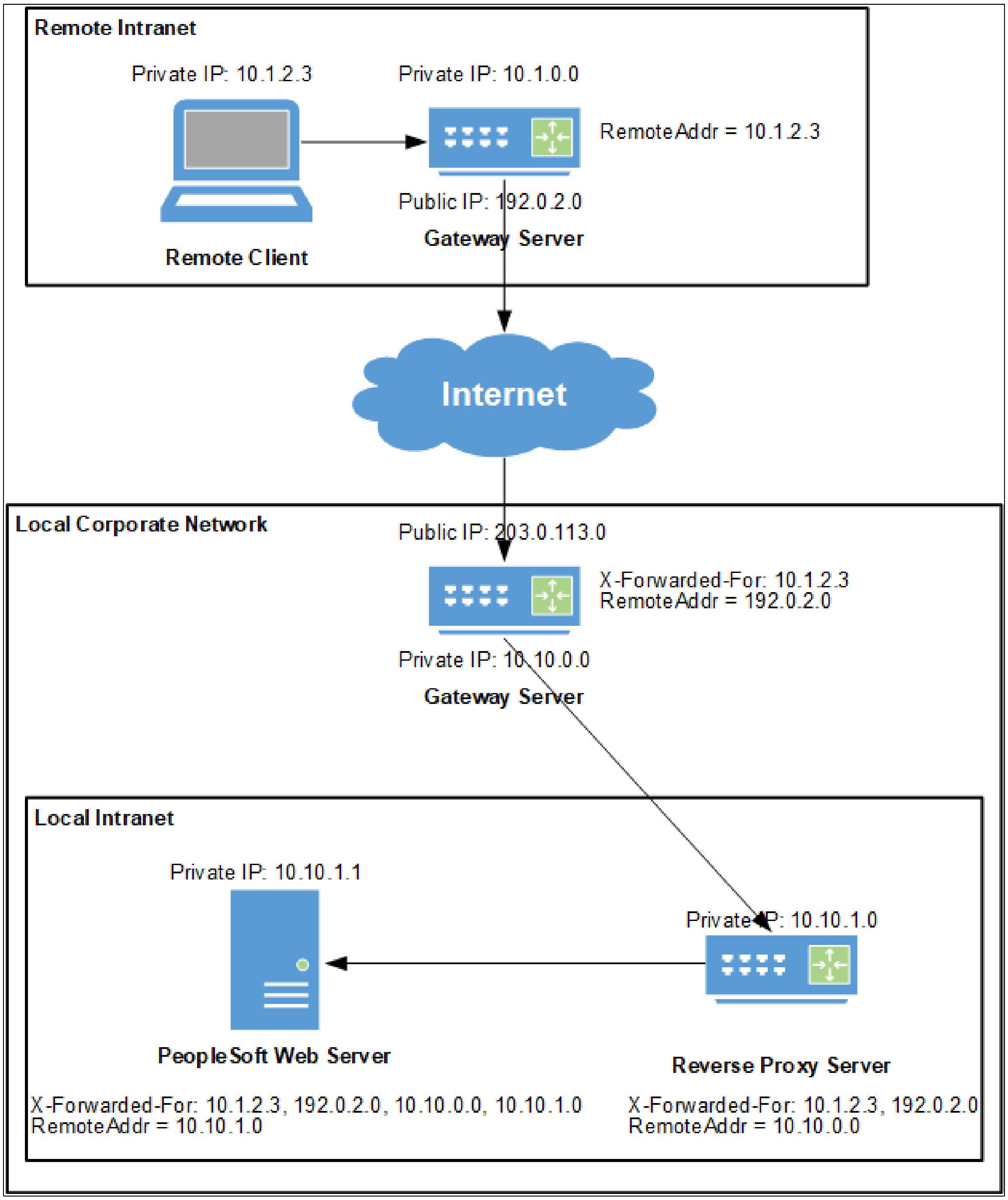
- **RemoteClientFormat** – Identifies the format (IP address or DNS name) to be recorded and reported by the PeopleSoft system.

Each of these directives is discussed in more detail in the following sections.

While these directives can help the PeopleSoft system more accurately identify and consistently report remote client access, much depends on the network outside of the PeopleSoft system. For example, consistency in the configuration of your internal network and how each machine forwards remote client information is key. But other aspects are out of your control. For example, if the actual remote client is on its own intranet that has a firewall or gateway server that connects to the internet, then the address of the remote gateway is effectively the remote client address. Addresses and forwarding headers included by other upstream systems outside of your control are deemed untrusted by the PeopleSoft system, *even if those addresses and headers are forwarded by your network intact.*

The following example illustrates how the effective remote client address of the remote gateway is recorded by the PeopleSoft system rather than the actual remote client address. Each system that forwards a request appends the address of the machine that contacted it to the HTTP request header—in this example, the X-Forwarded-For header.

The following example illustrates how the effective remote client address of the remote gateway is recorded by the PeopleSoft system rather than the actual remote client address. Each system that forwards a request appends the address of the machine that contacted it to the HTTP request header—in this example, the X-Forwarded-For header.



For the purposes of this example, the remote client directives have been configured as follows:

```
# An example of remote client directives
RemoteIPHeader = X-Forwarded-For
RemoteIPSeparator = ,
TrustedProxies = 10.10.0.0 10.10.1.0
RemoteClientFormat = c-ip
```


After the PeopleSoft web server appends the remote address of the previous hop, the HTTP request header that the PeopleSoft system processes includes:

```
X-Forwarded-For: 10.1.2.3, 192.0.2.0, 10.10.0.0, 10.10.1.0
```

When the PeopleSoft web server processes this request header, it reads it from right to left stripping off known trusted hosts. In this case, 10.10.1.0 and 10.10.0.0 are the addresses of trusted hosts, so they are stripped off. The next address, 192.0.2.0 is unknown to the PeopleSoft web server because it is not listed as a trusted host. Processing of the request header stops, and this is the remote client address that is logged and reported by the PeopleSoft system. In reality, it is not the actual remote client address (10.1.2.3), but the effective remote client address for the remote gateway instead. Because the c-ip directive is in effect, the address is reported and logged as an IP address, and no reverse DNS lookup is attempted.

General Considerations

Note the following considerations when specifying directives:

- All directives are optional.
- Directive names are not case sensitive.
- Each directive fits on one line.
- Except for the TrustedProxies directive, only one value for a directive is allowed. If a directive is specified more than once, the last value specified is in effect.
- The general format is Directive Name = value, in which = is used to separate the directive's name and its value.
- A # marks the beginning of a comment. All text from the # through the end of line is ignored.

RemoteIPHeader

This directive specifies the HTTP request header to be used by your controlled corporate network. If this directive is omitted, then evaluation of the remote client as described in this section is disabled.

For example:

```
RemoteIPHeader = X-Forwarded-For
```

Default value: none

RemoteIPSeparator

This directive specifies the single character that separates the IP addresses in the HTTP request header that are appended by the trusted hosts in your controlled corporate network.

For example:

```
RemoteIPSeparator = ;
```

Default value: space character

TrustedProxies

Each address in this list identifies a trusted host in your controlled corporate network through which inbound HTTP requests may have passed. Each address in the list is an IPv4 or IPv6 formatted address or a host name that can be resolved to an IP address (or addresses). Internally, PeopleSoft resolves all these addresses into a single list of IP addresses. Multiple directives with this name are allowed. Therefore, subsequent TrustedProxies directives append additional values to those from prior directives.

When using this directive, separate trusted hosts with a space. For example:

```
TrustedProxies = 10.1.1.1 10.2.1.1 proxy1.example.com 192.168.1.1
```

Default value: none

RemoteClientFormat

This directive specifies whether the PeopleSoft system logs and reports the remote client using its IP address or its host name. Legal values are c-ip or c-dns and are case insensitive. All other values cause the directive to be ignored. This directive, or its default value when omitted, is enabled regardless of the presence or absence of the RemoteIPHeader directive.

Note: If the DNS resolver contacted by the PeopleSoft system cannot resolve the DNS name of the remote client, then even when this directive is set to c-dns, the address logged and reported by the PeopleSoft system will be an IP address.

For example:

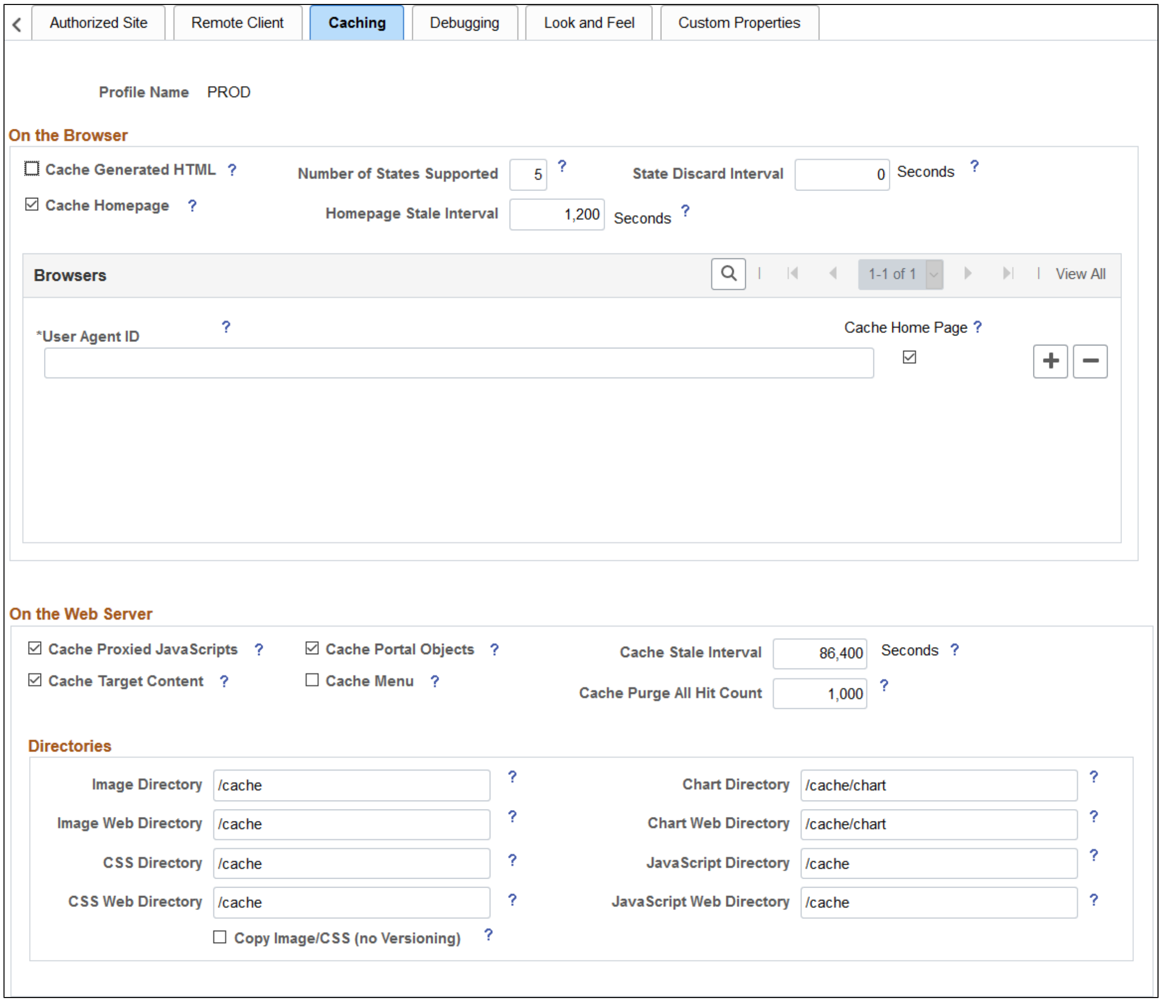
```
RemoteClientFormat = c-dns
```

Default value: c-ip

Configuring Caching

Access the Caching page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Caching page tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Caching page. You can find definitions for the fields and controls later on this page.



On the Browser

Field or Control	*Description*
Cache Generated HTML	Select to indicate that PeopleSoft application pages should be cached by the user's browser. For security reasons, deselect this check box in kiosk environments. **Note:** This setting does not affect caching of classic homepages and pagelets, which is determined by the Cache Homepage field. This check box is selected by default. **Note:** When you use menu navigation, the back button works even if this option is deselected because menu navigation does not generate the No-Cache directive.

Field or Control	***Description***
Number of States Supported	Specify how many browser states the portal should support when the user accesses PeopleSoft application pages. **Note:** If you have applications that make numerous server requests, you may want to increase this value. This increases the virtual machine's memory requirements, so be prepared to allocate more memory accordingly. The default value is *5* states.

Field or Control	***Description***
State Discard Interval	When a user signs out (either explicitly or because of a session timeout), the system normally clears all application states for the user's session. This property enables clearing of application states for individual windows instead. If your users have a long session time-out and many browser windows open, use this property to force the portal web server to release cached states for windows that a user has closed. Every time a user clicks the **New Window** link on a portal page, or when transfer PeopleCode opens a new window, the portal web server creates and maintains a state cache for that new browser window. Even if the user closes the window, the web server continues to maintain the cache during the interval that is specified by the **Inactivity Logout** field on the Web Profile Configuration - Security page. If users open many application browser windows simultaneously and then close most of them while still continuing to work actively in one or two remaining windows, you may be using more web server memory than is required and unnecessarily degrading its performance. You can improve performance by using this property to specify an independent time-out interval that applies to individual browser windows, which forces the portal web server to release cached states for windows that users have closed. When a user clicks the **New Window** link, and each time user activity in a window produces a request for data from the web server, the portal applies a current timestamp to its corresponding state cache. The next time the user clicks the **New Window** link, the portal examines the timestamp for each state cache. For any timestamp that's older than the status block time-out, the portal web server discards the associated window state cache on the assumption that the window must have been closed. **Note:** The portal performs this comparison only when the user clicks the **New Window** link. Enter a value that represents the number of seconds for the inactivity time-out per window. This time-out should be as short as possible, but still long enough to avoid discarding the state cache of any window that's still in use. A good starting value is the same value as the inactivity logout. If you don't specify this property, or if you enter *0,*status blocks are tracked per session—all status blocks are deleted when the user signs out or when the inactivity logout interval expires.

Field or Control	Description
	Note: A potential inconvenience with this property is that users might be inactive in a window without closing it, then return to it after the interval that you specify here has expired. In that case, they're presented with the application's search page.
Cache Homepage	**Important!** This setting applies to classic homepages and pagelets only, and not to fluid homepages. Select to apply homepage caching on the browser. When this check box is selected, the **Homepage Stale Interval** field and the **Browsers** grid become available. This check box is selected by default. When the Cache Homepage check box is selected, the following occurs: • All non-component-based pagelets will be cached on the browser by default unless the pagelet has a PORTAL_NOBROWSERCACHE content reference attribute set to *Y.* • Component-based pagelets will not be cached on the browser by default unless the pagelet has a PORTAL_NOBROWSERCACHE content reference attribute set to *N.* **Important!** Do not enable homepage caching for any component-based pagelet that uses interactive processing mode. When this check box is deselected, any PORTAL_NOBROWSERCACHE content reference attributes are ignored.
Homepage Stale Interval	**Important!** This setting applies to classic homepages only, and not to fluid homepages. Enter the number of seconds that the browser should wait before requesting an updated homepage from the portal server. The default value is *1200* seconds (20 minutes).

See Implementing Homepage Caching, Implementing PeopleSoft Page Caching.

Browsers

Use this grid to identify the browsers that you don't want to cache the homepage. For each browser make, model, and version, supply the identifying user agent ID and deselect the **Cache Home Page** check box. To re-enable caching for a listed browser, select the browser's **Cache Home Page** check box.

Any browser that is not listed caches the homepage if the global **Cache Homepage** check box is selected.

Note: If the global **Cache Homepage** check box is deselected, you can't enable homepage caching for individual browsers.

See Implementing Homepage Caching.

On the Web Server

Field or Control	*Description*
Cache Proxied JavaScripts	Select to enable caching of proxied JavaScript objects on the portal server. This check box is selected by default.
Cache Portal Objects	Select to enable metadata caching. This check box is selected by default.
Cache Stale Interval	Enter the number of seconds that should pass before the portal refreshes the metadata cache. The default value of this property is *86400* seconds (24 hours). **Note:** This property competes for effect with **Cache Purge All Hit Count.**
Cache Target Content	Select to cache all target content HTML that has an appropriately defined Cache element. This check box is selected by default.
Cache Menu	**Important!** This setting applies to the classic drop-down menu, and not to the Navigator menu in the NavBar. Select to enable web server-based navigation caching. For all profiles, including those that are newly created, the default value is *False.*

Field or Control	*Description*
Cache Purge All Hit Count	Specify the maximum total number of HTTP requests that the web server should receive for metadata objects before it purges the metadata cache, forcing the portal to refresh the cache. The default value is *1000* requests. Set this property to *0* to disable the feature. **Note:** This property competes for effect with the cache stale interval.

See Implementing Target Content Caching, Implementing Metadata Caching, Implementing Proxied JavaScript Caching, Administering Web Server-Based Navigation Caching.

Directories

Field or Control	*Description*
Image Directory	Enter the image file cache directory. The default value is */cache.*
Image Web Directory	Enter the image file web cache directory. The default value is */cache.*
CSS Directory (cascading style sheet directory)	Enter the cascading style sheet (CSS) cache directory. The default value is */cache.*
CSS Web Directory	Enter the CSS web cache directory. The default value of this field is */cache.*
Copy Image/CSS (No Versioning)	Select to have the system write a copy of the image and CSS cache with no version number. This check box is provided in case an external reference to the PeopleSoft stylesheet is needed. ***Warning!*** Enable this setting with care. In a production environment, it can double the number of files that are stored in the site's cache directory. Unless a clear need exists to use a custom solution to access style sheets and the like, this check box should be disabled. This check box is deselected by default.

Field or Control	***Description***
Chart Directory	Enter the directory of the cache for the chart image file. The default value is */cache/chart.*
Chart Web Directory	Enter the directory of the web cache for the chart image file. The default value is */cache/chart.*
JavaScript Directory	Enter the Java Script cache directory. The default value is */cache.*
JavaScript Web Directory	Enter the JavaScript web cache directory. The default value is */cache.*

Recent Search Results

Important! Persistent search, the drop-down menu, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications.

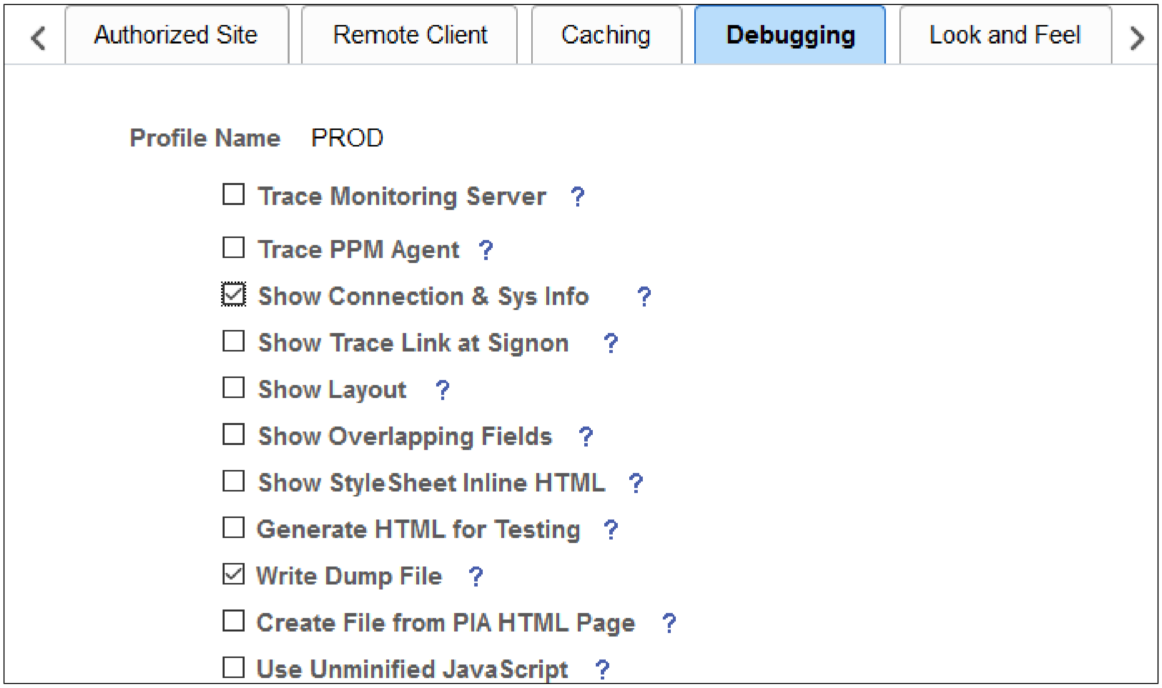
Field or Control	***Description***
Enable Caching	Select to temporarily store recent (persistent) search results on the browser and web server. Deselecting this setting suppresses all search persistence property settings that you configure in Application Designer definitions. The default value is *unchecked.*

See “Setting Record Field Use Properties” (Application Designer Developer’s Guide).

Configuring Trace and Debug Options

Access the Web Profile Configuration page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Debugging tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Debugging page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Trace Monitoring Server	Select to have the PeopleSoft Performance Monitor and PPMI servlets write debug information to the web server log. This check box is deselected by default. **Note:** This feature is not an agent trace. For an agent trace, use **Trace PPM Agent.**
Trace PPM Agent	Select to enable PeopleSoft Performance Monitor tracing on performance agents. This check box is deselected by default.
Show Connection Information	Select to have the application generate a system information page, which includes the browser, OS, PeopleTools release, application release, service pack, page definition name, component definition name, menu definition name, user ID, database name, database type, and application server address when you press **Ctrl+J.** The default is deselected. **Note:** This information is useful for orientation and troubleshooting purposes, but might not be suitable for end users.
Show Trace Link at Signon	Select to display a URL link at sign-in. The link opens a page for setting trace parameters.

Field or Control	***Description***
Show Layout	Select to apply border and color attributes in a table layout for pages. This enables developers to see the position of PeopleSoft Application Designer objects in HTML.
Show Overlapping Fields	Select to include comments in generated HTML pages that may help in diagnosing page layout problems, such as fields overlapping other fields.
Show StyleSheet Inline HTML	Select to insert the page's style sheet into its generated HTML. **Note:** Only classic style sheets are generated inline; fluid style sheets are not generated inline.
Show JavaScript Inline HTML	Select to display all the JavaScript functions that are used for processing in the generated HTML page.
Generate HTML for Testing	Select to alter the generated HTML to assist with testing and troubleshooting. For example, this option provides additional white space and comments on the page to aid readability. Also, it includes additional name attributes for reference from SQA robot scripts. **Note:** Selecting this option may cause some pages or pagelets not to appear correctly. You can also set this in the application server configuration file with the TracePIA option. To ensure that all pages appear correctly, verify that TracePIA isn't enabled in the configuration file.
Write Dump File	Select to have the system write a log file to the web server if an Oracle Jolt exception error occurs.

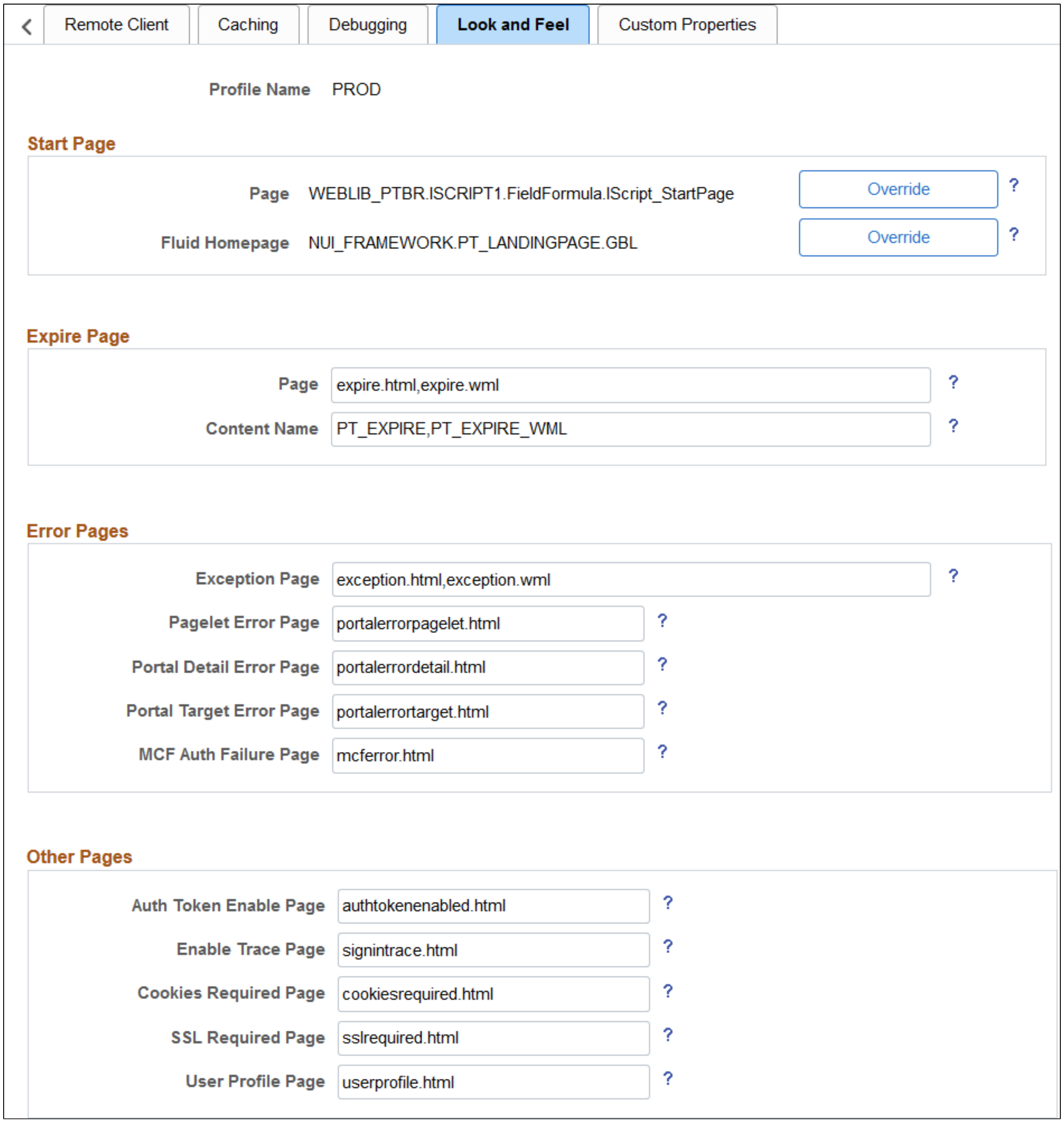
Field or Control	*Description*
Create File from PIA HTML Page (create file from PeopleSoft Pure Internet Architecture HTML page)	Select to view and debug the source HTML that the application server generates. The system saves each generated page as *PS_CFG_HOME*\appserv*domain*\LOGS *client**element**N*.html. The variables in the name are: • *Domain:* The name of the application server domain. • *Client:* The name of the machine or Internet Protocol (IP) address where the browser is running. • *Element:* The name of the process or object that generates the HTML (query name for query, program name for iScripts, and so on). • *N:* The state number for the generated page. ***Warning!*** Use this tracing feature only for troubleshooting and testing. Enabling this feature generates numerous directories and files on the application server, which significantly affects performance. Use it for short periods, preferably only for a single-user test scenario. Never enable it on a production website.
Use Unminified JavaScript	Select to disable the execution of minified JavaScript for the PeopleSoft Pure Internet Architecture domain. During runtime, the PeopleSoft system compresses, or minifies, all JavaScript stored in HTML definitions created in Application Designer. This is the default behavior. JavaScript that has been minified has been stripped down to only the essential strings required for runtime execution to save bandwidth. For more information on minified JavaScript in HTML definitions, see “Working with JavaScript in HTML Definitions” (Application Designer Developer’s Guide).

See Configuring Custom Properties and “Working with Performance Monitor Web Profile Properties” (Performance Monitor).

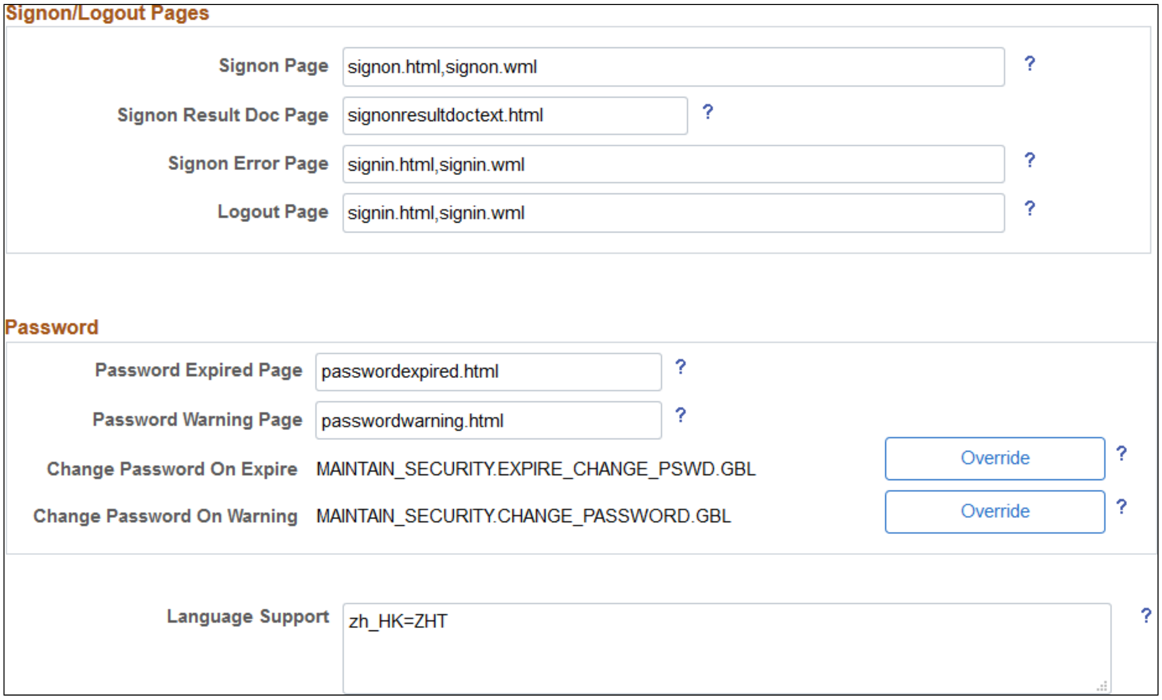
Configuring Look and Feel

Access the Web Profile Configuration page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Look and Feel tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Look and Feel page. You can find definitions for the fields and controls later on this page.



This example illustrates the fields and controls on the Web Profile Configuration - Look and Feel page. You can find definitions for the fields and controls later on this page.



Start Page

Field or Control	*Description*
Page	Displays the script to which the system redirects users after a successful sign-in. This value references the iScripts that build the PeopleSoft navigation. The default value is *WEBLIB_PTBR.ISCRIPT1.FieldFormula.IScript_StartPage.*
Override	Click to change the start page script. The Override Start Page Script page is displayed with these fields: • **Record (Table) Name:** The default value is *WEBLIB_PTBR.* • **Field Name:** The default value is *ISCRIPT1.* • **PeopleCode Event Name:** The default value is *FieldFormula.* • **PeopleCode Function Name:** The default value is *IScript_StartPage.* The values of these four fields are concatenated to produce the value of the **Page** field on the Look and Feel page.

Field or Control	***Description***
Fluid Homepage	Displays the name of the component used to render fluid homepage tabs. The default value is *NUI_FRAMEWORK.PT_LANDINGPAGE.GBL.* See "Web Profile Configuration - General Page" (Fluid User Interface Developer's Guide) for more information.

Expire Page

Field or Control	***Description***
Page	Displays the HTML page containing text variables that are defined in text.properties. This page appears when user inactivity exceeds the limit that is specified by the **Inactivity Logout** field in the **Authenticated Users** region of the Web Profile Configuration - Security page. Do not change this setting. The default value is *expire.html,expire.wml.*
Content Name	Enter the content name that is stored in the HTML catalog. It appears when a page has expired due to reaching the limit that is specified by the **Number of States Supported** field in the **On the Browser** region of the Web Profile Configuration - Caching page. Change this value with caution. The default value is *PT_EXPIRE,PT_EXPIRE_WML.*

See Configuring Portal Security, Configuring Caching.

Error Pages

Field or Control	***Description***
Exception Page	Enter the name of the page that is used in Java to handle exceptions. Change this value with caution. The default value is *exception.html,exception.wml.*

Field or Control	***Description***
MCF Auth Failure Page (multichannel framework authorization failure page)	Enter the name of the multichannel framework authorization failure page. The default value is *mcferror.html.*
Pagelet Error Page	Enter the name of an initial error message page to be displayed in the user's browser when a portal pagelet is unavailable. The default value is *portalerrorpagelet.html.*
Portal Target Error Page	Enter the name of an initial error message page to be displayed in the user's browser when target content is unavailable. The default value is *portalerrortarget.html.*
Portal Detail Error Page	Enter the name of an error message page that can be displayed in the user's browser to provide more detail about an error when retrieving portal content. The default value is *portalerrordetail.html.*

Because your PeopleSoft portal can aggregate its content from many different sources, errors can occur for a variety of reasons. An error may occur if:

- A content server is down.
- An invalid URL is specified.
- A portal node from which content is being requested is inactive.
- A portal node fails to deliver requested content within the time-out interval that you specify by using the PSTIMEOUT content reference attribute.

See Configuring Pagelet Time-out Settings.

For practical purposes, all of these reasons result in the requested content being unavailable.

The values that you specify for the pagelet error page, portal target error page, and portal detail error page are the names of HTML pages that present customizable, translatable error messages. The default pages are delivered with the PeopleSoft system and provide useful error messages without any modification.

You can use the delivered error pages, modify them, or create your own. You can choose additional error message information to be displayed by using bind-type variables of two types in your error pages:

- Numeric message variables

  These map to strings that are predefined in the following file:

  PS_CFG_HOME\webserv\peoplesoft\applications\peoplesoft\PORTAL\WEB-INF\psftdocs *sitename*\text.properties

For example, the numeric variable 5006 maps to the phrase “The portal was unable to retrieve the page you are looking for.”

- Session variables

  These map to system information that depends on the current page and activity.

  Session variables have values that are dynamically resolved by the portal engine at runtime. Available session variables include:

  - ErrorTitle
  - ErrorDescription
  - ErrorURL
  - AccessedURL
  - OriginalURL
  - TargetPage
  - StackTrace
  - DetailError

    DetailError is the name of the page that is specified by the **Portal Detail Error Page** field. This can be used to specify a link target.

To invoke a numeric or session variable, you insert it between the strings `<%=` and `%>` in your HTML. For example: `<%=5009%>` or `<%=ErrorDescription%>`. Examine the delivered error pages for examples of how to use the numeric and session variables within your HTML.

Other Pages

Field or Control	***Description***
Auth Token Enable Page (authorization token enable page)	Enter the name of the page to be displayed when the site being accessed is configured with an authentication domain (or web server session cookie domain), and the URL from the browser doesn't include the domain. This page contains a link to a sign-in page that produces the correct URL for the site. You can modify the content interface and design, but do not change the internal meta-tags that generate the correct URL. The default value is *authtokenenabled.html.*

Field or Control	***Description***
Enable Trace Page	Enter the name of the page to be used to set trace parameters. If you enable tracing, this page appears before the sign-in page so that you can set the trace parameters and then sign in to the system. The default value is *signintrace.html.*
Cookies Required Page	Enter the name of a page to be displayed when the browser does not accept cookies. You should configure browsers to accept cookies. The default value is *cookiesrequired.html.*
SSL Required Page	Displays the name of the page that should appear if the **Secured Access Only** check box is selected in the **SSL** region of the Web Profile Configuration - Security page, and the user is unable to proceed without SSL. Do not change this setting. The default value is *sslrequired.html.*
User Profile Page	Displays the name of the page that appears when the user clicks the link from a password-expired page. Do not change this setting. The default value is *userprofile.html.*

See Configuring General Portal Properties.

Signon/Logout Pages

Field or Control	***Description***
Signon Page	Displays the name of the page that redirects to the servlet for the sign-in process. Do not change this setting. The default value is *signon.html,signon.wml.*

Field or Control	***Description***
Signon Result Doc Page (signon result document page)	If your portal site is configured with sign-in PeopleCode to additionally validate the user, that code can call a result document if the user authentication failed. You can present the result document to the user in several ways by specifying one of these HTML documents in this field: *signonresultdoctext.html:* The result text is wrapped in a standard portal error page. This is the default value. *signonresultdocpage.html:* The result text itself is formatted as the error page. *signonresultdocredirect.html:* This page redirects the user to your own result document, using the *<%=resultDoc%>* session variable element.
Signon Error Page	Enter the name of the page that should appear when the user makes an error signing in. To customize your sign-in page, clone signin.html as a starting point. Change this value with caution. The default value is *signin.html,signin.wml.*
Logout Page	Enter the name of the page that should appear when the user signs out. You may define a custom logout page. Change this value with caution. The default value is *signin.html,signin.wml.*

Password

Field or Control	***Description***
Password Expired Page	Displays the name of the page that should appear when the user password is expired. Do not change this setting. The default value is *passwordexpired.html.*

Field or Control	Description
Password Warning Page	Displays the name of the page that should appear when the user's password is about to expire in the number of days specified in PeopleSoft security. Do not change this setting. The default value is *passwordwarning.html.*
Change Password on Expire	Displays the Change Password page content ID. The system uses the value in passwordexpired.html to take the user to the Change Password page when a password is expired. The default value is *MAINTAIN_SECURITY.EXPIRE_ CHANGE_PSWD.GBL.*
Override	Click to override the expired password change page. The Override Change Password on Expire Page page appears. • **Menu Name:** The field is blank by default. • **Component:** The field is blank by default. • **Market:** The default value is *Global.*
Change Password On Warning	Displays the content ID for the Change Password page. The system uses the value that you enter in passwordwarning.html to take the user to the Change Password page when a password warning is required. The default value is *MAINTAIN_SECURITY.CHANGE_ PASSWORD.GBL.*
Override	Click to override the Password Change Warning page. The Override Change Password on Warning Page page appears. • **Menu Name:** The field is blank by default. • **Component:** The field is blank by default. • **Market:** The default value is *Global.*

(Optional) Language Support

Specify additional mappings from International Organization for Standardization (ISO) locale codes to PeopleSoft language codes, using one line per entry.

The PeopleSoft system uses proprietary codes to represent user languages, and the PeopleSoft Pure Internet Architecture sign-in page appears in the PeopleSoft language corresponding to the user's browser language setting (an ISO locale code). This field enables you to provide mappings from additional ISO locale codes to PeopleSoft language codes to support a wider range of browser-specific language settings.

For example, the entry *zh_HK=ZHT* maps the ISO locale code *zh_HK* (Hong Kong Chinese) to the PeopleSoft language code *ZHT* (traditional Chinese) so that traditional Chinese is the default sign-in language for browsers with the preferred language set to Hong Kong Chinese.

Your PeopleSoft application is delivered with default mappings, shown on the Manage Installed Languages page. The default mappings don't require entries in this field.

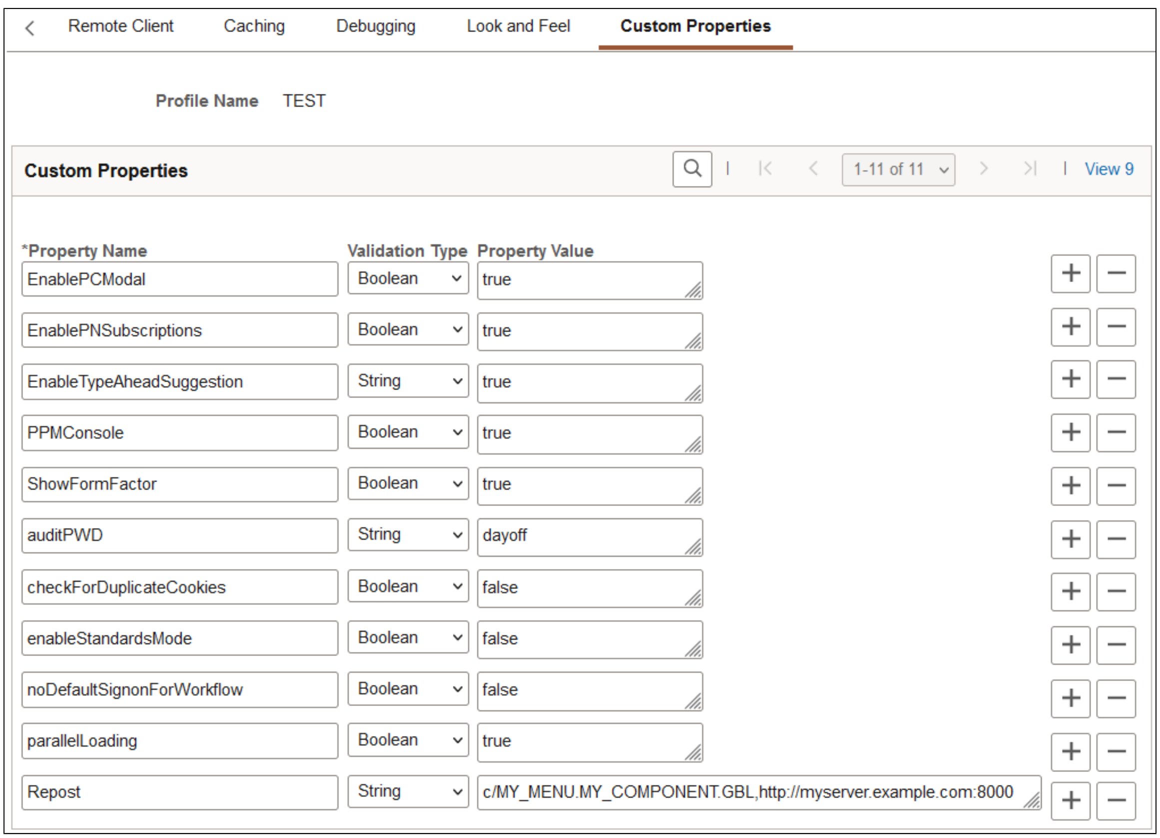
Related Links

"Understanding the Addition of New Language Definitions" (Global Technology)

Configuring Custom Properties

Access the Web Profile Configuration page. (Select **PeopleTools** > **Web Profile** > **Web Profile Configuration.** Select the Custom Properties tab.)

This example illustrates the fields and controls on the Web Profile Configuration - Custom Properties page. You can find definitions for the fields and controls later on this page.



This page enables the use of custom web profile properties that are needed only for specific additional purposes. For each property, you enter the property name, select its validation type, and provide the appropriate property value. The validation type ensures that the property value that you enter is the correct format. The currently available custom properties are:

Field or Control	***Description***
AJAXConsole	This property, although not displayed, enables you to launch an interface for viewing diagnostic information related to AJAX parameters that are required by PeopleSoft support personnel, who will provide you with the correct values when necessary.
attachmentExtraWindow	This property enables you to prevent the appearance of a message window when you attempt to view certain types of attachments (such as *.doc files) that are not directly displayed by the browser but are instead displayed by the application associated with files of that type. The message varies based on the browser you use. This message window requires explicit user action to dismiss it. The property takes a value of 2 to prevent the appearance of the window. All other values used for this property will allow the appearance of this window.
auditPWD	This property enables certain debug and control settings that are required by PeopleSoft support personnel, who will provide you with the correct values when necessary. Never set this property in a production environment unless directed to do so.

Field or Control	***Description***
checkForDuplicateCookies	Duplicate cookies occur when all of the following are true: • You have two PeopleSoft applications installed for which you have not implemented single signon functionality. • The authentication domain that you specify for one application's web server is a subset of the authentication domain that you specify for the other, such as *.example.com* and *.us.example.com.* • A user attempts to sign in to both applications simultaneously from the same browser. When these conditions are met, the browser presents multiple cookies to each application, which produces unpredictable browser behavior, such as displaying the sign-in page or a page expiration message. For this property, specify a validation type of *Boolean* and enter one of these property values: *True:* The portal checks for sets of duplicate cookies. When a duplicate is found, the user is taken back to the sign-in page with this message: “Your browser sent multiple sets of cookies.” You can change the text of this message by editing number 107 in the errors.properties file of the portal site. *False:* The portal doesn't check for sets of duplicate cookies. This is the default setting, which applies when this property isn't specified. To avoid this issue altogether with applications that don't use single signon functionality, make sure that you specify authentication domains that aren't subsets of each other, such as *.user.example.com* and *.corp.example.com.*
EnableAutocomplete	This property, although not displayed, enables AJAX features associated with auto-complete. For this property, specify a validation type of *Boolean* and enter *false* as the property value to disable auto-complete at the system level. **Note:** If this property is not set to false, you can control autocomplete through personalizations.
EnablePCModal	***Warning!*** The Oracle-delivered setting is *true*. Do not change this setting.
EnablePNSubscriptions	As of PeopleTools 8.58, you do not need to explicitly configure the web server for push notifications. The EnablePNSubscriptions custom web profile property is no longer used, and is ignored regardless of its setting.

Field or Control	*Description*
enableWorklistNotify	Use the enableWorklistNotify property to change the behavior of the notify feature on classic pages to the classic workflow notify behavior. On classic pages, the default is to use the fluid notify feature (the equivalent of setting this property to false or not adding this property to the web profile configuration page). Add this property and set it to true to display the classic workflow notify feature.
FocusOnList	Use the FocusOnList property to change the behavior of the cursor after retrieving search results on classic component search pages only. The default behavior is to set the cursor focus on the first search criteria field after search results are returned (the equivalent of setting this property to false). Add this property and set it to true to set the cursor focus on the first search result within the list of results returned.
IDDA	Use the IDDA property to enable the Instrumented Development Diagnostic Aid (IDDA) logger and specify the what is logged as an *integer* value. See “Enabling IDDA Logging” (System and Server Administration) for more information.
InPortalContext	Use the InPortalContext property to determine whether to maintain the context of the portal system when navigating to remote content provider system links provided under PeopleSoft Interaction Hub’s PeopleSoft menu (that is, links such as Financials Supply Chain PT8.4x, Human Resources PT8.4x, and so on). “Maintaining the portal context” means that the menu of the portal system is displayed when navigating to these links rather than the menu of the content provider system. Specify one of these values: • *true* (default) – Maintain the portal context. • *false* – Do not maintain the portal context; display the menu of the content provider system instead. You must set this parameter in the active web profile of both the portal system and the content provider system for this to take effect. **Note:** This setting pertains to using the links under PeopleSoft Interaction Hub’s PeopleSoft menu only. For menu items configured via unified navigation, the portal context is always maintained. In addition, this property has no effect on fluid homepages and components in the content provider system. The fluid header always maintains portal context.

Field or Control	***Description***
mutualAuth	This property, although not displayed, controls whether your environment is set up for mutual authentication. The default value is *True.* **Note:** If you are using Lightweight Directory Access Protocol (LDAP) authentication, you must use the default setting. If you are not using LDAP authentication, a Java exception error caused by a failure to retrieve a certificate property may appear in your webserver log. If this occurs, you must change this value to *False* to disable the authentication from trying to retrieve the certificate property.
noDefaultSignonForWorkflow	Use the noDefaultSignonForWorkflow property to control whether a user sees the default signon page after clicking a workflow link. When noDefaultSignonForWorkflow is set to true, the system disables Public Access for workflow notification URLs. By default, noDefaultSignonForWorkflow is set to *False.* See the following table for information related to the behavior of this setting. See the following section, “noDefaultSignonForWorkflow Property, for more information. **Note:** This property applies to worklist URLs, not all workflow URLs. Only the URLs that contain a /w/ use this property.
parallelLoading	The parallelLoading property enables parallel pagelet loading on classic homepages, dashboards, and WorkCenters, which improves the performance of these pages. **Note:** Beginning with PeopleSoft 9.2 applications, this property is set to true in the configuration.properties file. However, pagelets with global variables should not be loaded in parallel; such pagelets must be loaded sequentially. When the parallelLoading property is set to true, you must also add the BGLOBALVALUES content reference attribute to the content reference definition for each pagelet that includes global variables; set the BGLOBALVALUES attribute to true. Then, these pagelets will be loaded sequentially while all other pagelets will be loaded in parallel.
PPMConsole	This property, although not displayed, enables you to launch an interface for viewing diagnostic information related to agents and PPMI clients. The default value is *False.*

Field or Control	***Description***
psback_unencoded_d	By default, the psback cookie has encoded delimiters; this is the web server default when psback_unencoded_d is undefined. However, in a clustered environment that includes mixed PeopleTools versions, you may need to allow unencoded delimiters in the psback cookie. If your mixed clustered environment includes any PeopleTools 8.53 or earlier releases, PeopleTools 8.54.20 or earlier, or PeopleTools 8.55.02 or earlier, you will need to allow unencoded delimiters by adding the psback_unencoded_d custom property to the active web profile on each system in the clustered environment: • *false* (default) – The psback cookie is generated with encoded delimiters. • *true* – The psback cookie is generated with unencoded delimiters

ps_theme	Add the ps_theme custom property (type = *String*) to identify the default local node for the designated portal (or hub) system in a clustered environment. The ps_theme property is defined on content provider systems only for routing push notifications in a clustered environment or to enforce branding assignments configured on the PeopleSoft Interaction Hub gateway system. Use the following format to set the value of this cookie:
	
	`DEFAULT_LOCAL_NODE:http://my_server.my_d⇒`
	
	`omain:port_no/psp/site_name/`
	
	For example:
	
	`PS_NODE:http://server01.example.com:8000⇒`
	
	`/psp/MYSITE01/`
	
	Important! The ps_theme property is generated automatically when Automated Configuration Management (ACM) is used to configure a clustered environment. In addition, the property can be manually configured in the Configure Unified Navigation Center on the designated portal system.

Field or Control	*Description*

Repost	Use the Repost custom property to identify a specific “punchback” (or callback) URL that a third-party site can use to bypass SameSite requirements for passing cookies. You can define up to 11 repost URLs: Repost, Repost1, Repost2, ... Repost10.
	
	Specify the punchback URL as a string value in the following format:
	
	`portal_URL,http://host_name:port_no`
	
	For example, a full PeopleSoft URL includes the site, portal, and node names as follows:
	
	`http://myserver.example.com:8000/psp/ps_⇒`
	`site/EMPLOYEE/HRMS/c/MY_MENU.MY_COMPONEN⇒`
	`T.GBL`
	
	The value of this property would be set as follows:
	
	`c/MY_MENU.MY_COMPONENT.GBL,http://myserv⇒`
	`er.example.com:8000`
	
	See the following section, “Repost URLs for SameSite Exceptions,” for more information on implementing a repost URL.

ShowFormFactor	Use the ShowFormFactor property to enable the Form Factor drop-down list to be displayed on the sign-in page. The default value is false.
unauthSessionInterval	Use the unauthSessionInterval property to set session time-out (in seconds) during the authentication process. The default value is *120* seconds. The purpose is to limit the effectiveness of denial-of-service attacks on failed authentications. To determine the proper time for this property, check the time in seconds that it takes to send an HTTP(S) request from the browser to the web server and multiply the result by 2.

Repost URLs for SameSite Exceptions

Use the Repost custom property to identify a specific “punchback” (or callback) URL that a third-party site can use to bypass SameSite requirements for passing cookies. You can define up to 11 repost URLs: Repost, Repost1, Repost2, ... Repost10. See the Repost entry in the previous section on how to set the string value of the Repost custom property. Typically, punchout/punchback integration is implemented for B2B e-commerce scenarios, such as procurement from a third-party website and so on.

For more information on SameSite cookies, see Configuring Cookie Rules.

In addition to defining the Repost custom property, you must also update the PeopleCode that generates the punchback URL by appending `?&cmd=Repost`. For example:

```
&rec.PB_URL.Value = GenerateScriptContentURL(%Portal, %Node, Record.WEBLIB_PB_DC, F⇒

ield.PB_ISCRIPT_LIB, "FieldFormula", "IScript_DCCatcher") | "?&cmd=Repost";
```

Finally, you must decode the input parameters received in the punchback operation. For example:

```
Declare Function URL_Decode PeopleCode FUNCLIB_PORTAL.PORTAL_GEN_FUNC FieldFormula;

&CXML_URLENCODED = %Request.GetParameter("cxml-urlencoded");
&CXML_URLENCODED = URL_Decode(&CXML_URLENCODED);


&CXML_BASE64 = %Request.GetParameter("cxml-base64");
&CXML_BASE64 = URL_Decode(&CXML_BASE64);
```

noDefaultSignonForWorkflow Property

This table describes the behavior of the system based on the noDefaultSignonForWorkflow setting.

Configuration	***User Session***	***User Action***	***System Behavior***
Public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to true.	The user has no session with the system.	The user clicks a workflow notification URL in the email.	The system displays the sign-in page to the user. Once the user signs in to the system with a valid user ID and password, the system takes the user to the workflow notification based on the user profile privileges.
Public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to true.	The user already holds a valid session with the system through the public access.	The user clicks a workflow notification URL in the email.	Because the user already has a valid session, the system takes the user to the workflow notification. If the public access user profile does not have workflow privileges, the system displays an authorization failure page to the user.
Public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to true.	The user is already signed in to the system with a valid user ID and password (other than default sign-in)	The user clicks a workflow notification URL in the email.	The system takes the user to the workflow notification based on the user profile privileges.
When the public user access (default signon) is enabled and the custom web property “noDefaultSignonForWorkflow” is set to false.	Any	Any	The system behaves as configured.

Configuration	User Session	User Action	System Behavior
When the public user access (default signon) is disabled and the custom web property "noDefaultSignonForWorkflow" is set to false/true.	Any	Any	The system behaves as configured.

Note: If the public user does not have workflow privileges, then close the existing public user session (browser) before clicking the workflow link.

See "PeopleSoft Sign In" (Security Administration) and "Viewing Monitor Servlet Diagnostics" (Performance Monitor).

Viewing Web Profile History

Access the Web Profile History page. (Select **PeopleTools** > **Web Profile** > **Web Profile History**.)

This example illustrates the fields and controls on the Web Profile History page.

Use this page to review the current portal attributes of a web server, website, and web profile, including the web server's HTTP and HTTPS listen ports, the web profile that was last loaded, and the current HTTP session's cookie name and authentication domain.

In addition, the **Properties** field displays the relevant settings of the web server instance at the time it was loaded by the portal.

Defining Portal Nodes

This section provides an overview of portal nodes and discusses how to:

- Specify the node type.
- Set portal node characteristics.

Understanding Portal Nodes

You can create and maintain a node definition in the Node Definitions component (IB_NODE). A node is defined once but can be used for multiple purposes. Several pages in the Node Definitions component are used only for integration purposes by PeopleSoft Integration Broker. For portal purposes, define a portal node by using the Node Definitions and Portal pages.

Nodes must be blank for external content references.

Note: You cannot delete a node that is in use by a content reference when the **Save** button is clicked. If the node is in use, the node name is not deleted.

Related Links

"Understanding Nodes" (Integration Broker Administration)

Specifying the Node Type

Access the Node Definitions page. (Select **PeopleTools** > **Portal** > **Portal Node Definitions.**.)

Most of the options on the Node Definitions page apply only to Integration Broker nodes, except for the Node Type. A node type designates the URL format for the node.

Field or Control	***Description***
Node Type	*External:* Select if this node represents a non-PeopleSoft system, such as an external website, like www.example.com. *ICType:* Select if this node represents a PeopleTools database prior to PeopleTools 8.4. This allows compatibility between pre-PeopleTools 8.4 databases and portal databases for PeopleTools 8.4 and later. *PIA* (PeopleSoft Pure Internet Architecture): Select if this node represents a PeopleSoft database that is using PeopleTools 8.4 or later. This is the default setting.

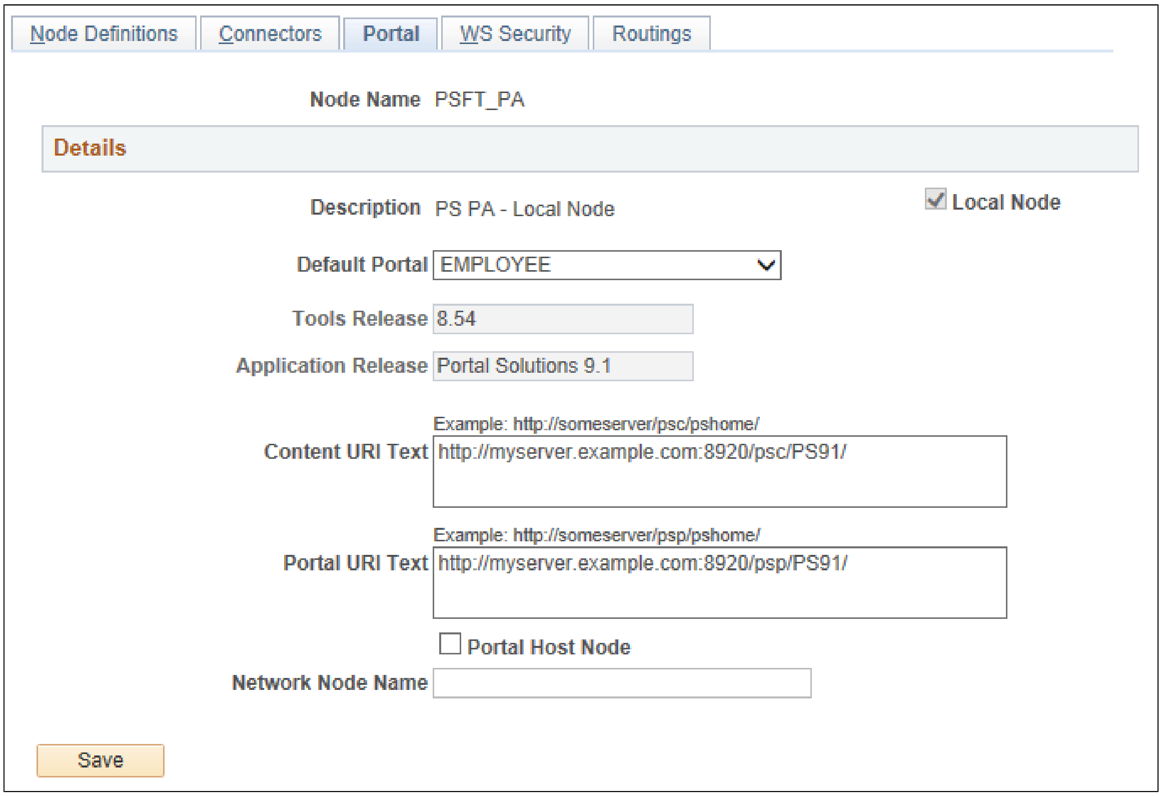
Related Links

"Defining Node Parameters" (Integration Broker Administration)

Setting Portal Node Characteristics

Access the Portal page. (From the Node Definitions page, select the Portal page.)

This example illustrates the fields and controls on the Node Definitions - Portal page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Node Name	Displays the node name, description, and local node check box status from the Node Definitions page.
Default Portal	For local nodes only, you can specify the portal definition to which this node is associated. **Note:** The portal host node for a portal is specified on the Portal Definitions page.
Tools Release	Enter the version of PeopleTools that is running. This is a free-form field. You can enter any information that you want; however, Oracle recommends that you use the same PeopleTools release that is available when you press **Ctrl + J.** **Important!** The Tools Release value is required to support the check token ID required for single signon configurations and for the remote pagelet search feature.

Field or Control	*Description*
Application Release	Enter the version of the PeopleSoft application that you are running. This is a free-form field. You can enter any information that you want; however, Oracle recommends that you use the same application name release that is available when you press **Ctrl + J.**

Content URI Text Portal URI Text	Enter the uniform resource identifier (URI) of the content servlet (psc) and the URI text for the portal servlet (psp). For example:
	`http://myserver.example.com:8010/psc/MY_⇒`
	`SITE/`
	`http://myserver.example.com:8010/psp/MY_⇒`
	`SITE/`
	Important! The combination of the server name, colon, and port number in these URIs must not exceed 30 characters in length.

Portal Host Node	**Note:** The Portal Host Node and the Network Node Name fields are required when configuring unified navigation, which consists of a central portal system and one or more content provider systems. These fields must be set only for *remote node definitions* on the portal system and on each content provider system. These fields are set so that unified fluid homepages are able to obtain the corresponding Integration Broker node for a given portal host node. *Do not set these fields on local node definitions.* Select the Portal Host Node check box *only* for a node definition for a remote portal host node.
Network Node Name	Enter the name of the default local node of the remote system *only* for a node definition for any remote node.

Related Links

Understanding Unified Navigation

Defining All Participating Nodes

"Implementing PeopleSoft-Only Single Signon" (Security Administration)

Implementing Single Signon Functionality

This section provides an overview of single signon functionality and discusses how to sign in to a non-default portal.

Understanding Single Signon Functionality

PeopleSoft PeopleTools supports single signon functionality for use with the PeopleSoft Pure Internet Architecture. Within the context of your PeopleSoft system, *single signon* means that after a user has been authenticated by one PeopleSoft application server, that user can access additional PeopleSoft application servers without entering a user ID or password. Although the user is actually accessing a different application server and database, the user navigates seamlessly through the system.

The single signon authentication service is shared by all PeopleTools web services and involves these steps:

1. The first application server (node) checks the HTTP request for a PeopleSoft authentication cookie.
2. If an authentication cookie is not found, then the server returns a language-specific sign-in page.
3. The user completes the sign-in page, entering a user ID and password.
4. The server authenticates the user based on the user ID and password.
5. A web browser cookie stores a unique access token for the user after the initial authentication.
6. An httpSession object is created for the newly authenticated session.
7. When the user connects to another PeopleSoft application server (node), the second application server uses the token in the browser cookie to reauthenticate the user without an additional sign-in process.

Because the portal integrates content from various data sources and application servers into a unified interface, you need to configure single signon functionality before deploying the portal. Users sign in once and then can navigate freely without encountering numerous sign-in pages.

However, when frame-based content is accessed across different machines, single signon functionality does not work when you access content on another web server. If the web servers are on the same internet domain, such as *.corp.example.com,* single signon functionality remains effective and a sign-in page does not appear.

Related Links

“PeopleSoft Sign In” (Security Administration)
“Understanding the Delivered External Authentication Solutions” (Security Administration)
“Understanding Single Signon” (Security Administration)
“AuthTokenDomain” (PeopleCode API Reference)

Signing In to a Non-Default Portal

To sign in to a portal other than the default, use a URL such as: http://portalserver/psp/ps/MY_PORTAL?cmd=login.

To create a sign-in page that signs in to a non-default portal, change signon.html like this:

```
<meta HTTP-EQUIV='Refresh' CONTENT='1; URL=../psp/ps/MY_PORTAL/?cmd=login'>
```

If you are a portal administrator and want to sign in to a database to work with PeopleTools-specific menus, replace the existing portal name with the site template name in the URL to make sure that you are administering your local PeopleTools menu and not inadvertently altering the database. Replace this:

http://../psp/ps/EMPLOYEE/

Instead, use this:

http://../psp/ps/PS_SITETEMPLATE

Related Links

“AuthTokenDomain” (PeopleCode API Reference)
Using Custom Portal Headers

Redirecting Pages with Sensitive Data

If you have a page that contains sensitive data, and you want to redirect the user to it, you should use the provided template HOMEPAGE_DESIGNER_TEMPLATE. For the content reference for the component, specify this as the template name. You should also use PeopleCode to redirect the user instead of using an HTML area on the page.

The following example shows the type of PeopleCode that you could use:

```
&Res = SwitchUser(QE_SWITCHUSER.QE_SWITCH_USER, QE_SWITCHUSER.QE_SWITCH_USER_PWD, Q⇒

E_SWITCHUSER.QE_SWITCHUSERAUTHT, QE_SWITCHUSER.QE_SWITCHUSERINFO);

If (&Res = True) Then
   Transfer( False, MenuName.PROCESSMONITOR, BarName."INQUIRE", ItemName."PROCESSMO⇒

NITOR", Panel."PMN_PRCSLIST", "U");
Else
   QE_SWITCHUSER.QE_SWITCHRES = "N";
   QE_SWITCHUSER.QE_PPCAUTHTOKEN = %AuthenticationToken;
   QE_SWITCHUSER.QE_RESULTDOC = %ResultDocument;

   Record.QE_SWITCH_SRCH.QE_CUR_USER.Value = %UserId;
End-If;
```

Overriding Page Encoding

You can override the character set encoding for any page that is registered in the portal. You define a content reference attribute called PORTAL_ENCODING_OVERRIDE with a value that specifies the character set to use when processing a page. The portal servlet then overrides all other indicators for encoding, including headers and meta-tags.

The standard encoding for PeopleSoft pages is UTF-8. The portal can usually determine the encoding of a page; however, if an external, non-UTF-8 page is not displayed correctly, overriding the page encoding may help.

Related Links

“Understanding Character Sets” (Global Technology)

Importing Menu Groups into the Portal Registry

This section provides an overview of menu group importing and discusses how to:

- Import menu groups.
- Select components to include in portal navigation.

Understanding Menu Group Importing

The Menu Import Application Engine process (PORTAL_MENU) imports menu groups from the menu definition into a portal registry.

Important! Only system administrators should perform this process. If the user does not have security to the menus and pages, just a folder titled *Label* gets created.

The process creates folders and content references that reflect the menu definition hierarchy. This table shows the folder hierarchy that is imported:

Level	***Menu Item***
Folder level one	Menu group
Folder level two	Menu name
Folder level three	Bar name
Content reference	Component name

Note: Use the Menu Import process to migrate custom menus only. The navigational menus that are delivered with your PeopleSoft application have already been added to the appropriate portal registry. The Menu Import process has no effect on components that are already registered. However, you can use the process to pick up additional components that are not delivered with the PeopleSoft system.

Before you use the Menu Import process, note these points:

- Object names that are assigned by the Menu Import process are autogenerated.

  If a folder name or content reference name already exists, the system appends a number to the generated object name. Consequently, you may have folder object names such as USE, USE1, USE2, and USE3. As a result, use the registration wizard if you plan to copy the application database registry structures into PeopleSoft Interaction Hub. When you use the registration wizard, you can control the object naming, and you can ensure the uniqueness of object names across databases.

- If a component resides in multiple menus, the process creates a separate content reference for each menu and component combination.

- The navigation hierarchy created by the Menu Import process does not match the delivered PeopleSoft hierarchy.

  To keep the navigation consistent, consider putting all custom menus into a common menu group and generating the registry structures for that menu group. Then, once the registry structures are generated, manually move the content references to an appropriate folder in the existing navigation hierarchy.

  To create a common menu group, open the custom menu in PeopleSoft Application Designer. On the Use tab of the Menu properties dialog box, enter a unique name in the **Menu Group** field. After you save the menu, it is available for other menus to use.

Using the Menu Import Process with PeopleSoft Interaction Hub

If you run the Menu Import process against multiple application databases and merge multiple registries into PeopleSoft Interaction Hub, bar names (level-three folders) can be overwritten because they frequently have duplicate names. The last folder that is copied takes precedence over all previous folders.

The autogenerated object names are based upon the current database portal registry. The Menu Import process cannot check external databases for existing object names, but it does check the current database.

If you run this process on different databases, the different databases may likely contain identical folder or content reference object names that refer to different objects. This can become problematic if you decide to copy these portal registry structures into PeopleSoft Interaction Hub. The same-named objects overwrite each other. You might need to delete the generated folders and content references to make them unique in the future.

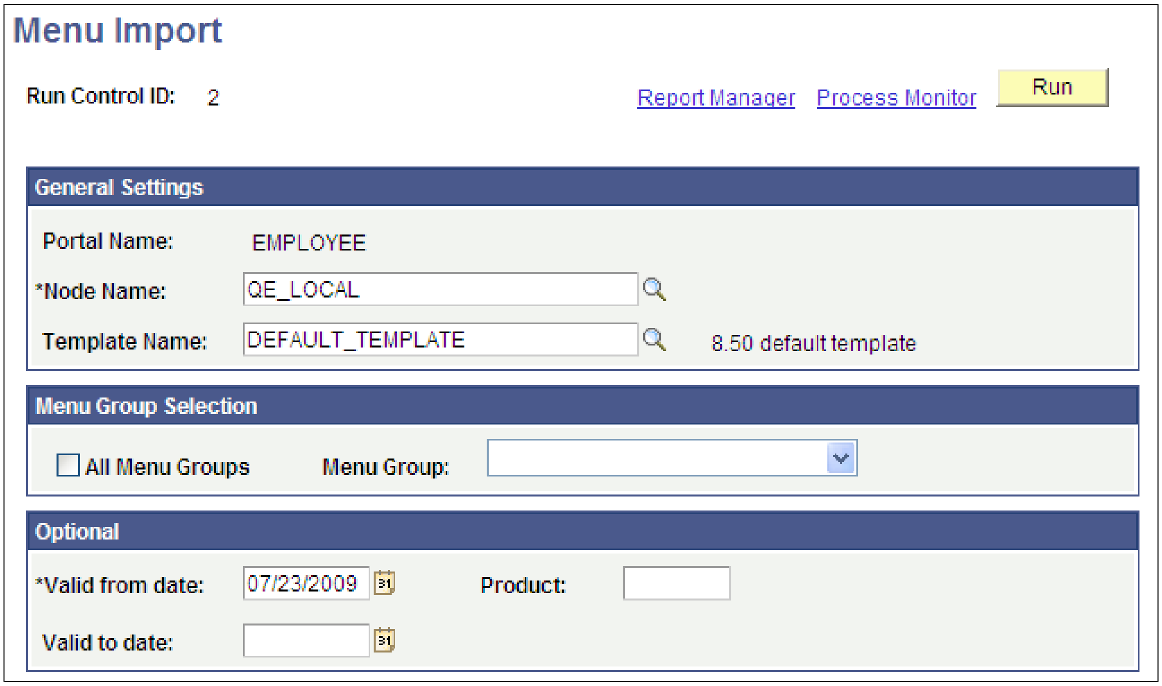
Use the registration wizard to register menus and components into the portal registry. Although with the registration wizard you must register items one by one, you can control folder and object names at the time of the import.

See “Understanding the Registration Wizard” (Application Designer Developer’s Guide).

Importing Menu Groups

Access the Menu Import page. (Select **PeopleTools** > **Portal** > **Import Menu Groups**.)

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



The Menu Import process creates content references for all components that belong to the menu group. This process should be used only to migrate custom menus.

Field or Control	***Description***
Node Name	Select the node of the content provider where the registry structure resides. The default value is the currently running node. **Note:** Never select the local node for an application database. If you copy the generated content references to a PeopleSoft Interaction Hub database, the content reference assumes that the component resides in the PeopleSoft Interaction Hub database, when it does not.
Template Name	(Optional) Specify the name of a template for the portal servlet to use. If you leave this field blank, the servlet uses the template that is associated with the selected node. If a template is not associated with the node, the default template is used for all content references that are created by the import process. The default value is the portal's default template. **Note:** Best practice is to not specify a template so that you can perform a mass template change at the content provider node level instead of having to change each content reference template individually.
All Menu Groups	Select to import all menu groups.

Field or Control	***Description***
Menu Group	Select a menu group to import.
Valid from date and **Valid to date**	Specify dates to make visible content references and folders that are associated with the menu that you are importing. In general, use January 1, 1900 for the valid from date and leave the valid to date blank unless you need to limit the time frame.
Product	Enter a unique, 4-character identifier for the menu. If you perform queries against the database, you can use this identifier to identify items that are associated with the menu.
Run	Click to begin the import process. PeopleSoft Process Scheduler launches an Application Engine program that imports the selected menu group definitions into the portal registry. This process typically takes several minutes, but can require more time depending on the number of items to be processed. Permission list settings are initially set to conform with the corresponding permissions set for the menu groups being imported. Once the menu group definitions are imported into the portal registry, no dynamic connection exists between the menu group definition in PeopleSoft Application Designer and the portal registry. If you change the security for the menu or component, the registry security is synchronized as long as the menu or component is in the same database as the registry structure. If you change the menu or component name, you must manually update the registry structure.
Process Monitor	Click to view the status of the menu import.

After you run the process, click the **Home** button in the universal navigation header and verify that the menu group has been imported into the portal registry. The new menu group should now appear along with the other folders at the top level.

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

Selecting Components to Include in Portal Navigation

During the menu import process, you can control which components are included in your portal navigation system.

To select components to include in portal navigation:

1. From PeopleSoft Application Designer, open the appropriate component.

2. Select **File** > **Definition Properties** and select the Use tab.
3. Select the **Include in Navigation** check box.

   After a menu import, the component is visible to users with appropriate permissions. (You can still change the permissions for any content reference after the menu import as necessary.)

   If you deselect the **Include in Navigation** check box, the component is not imported into the portal registry during the Menu Import process. You can also set this option at the menu level.
4. Click **OK.**

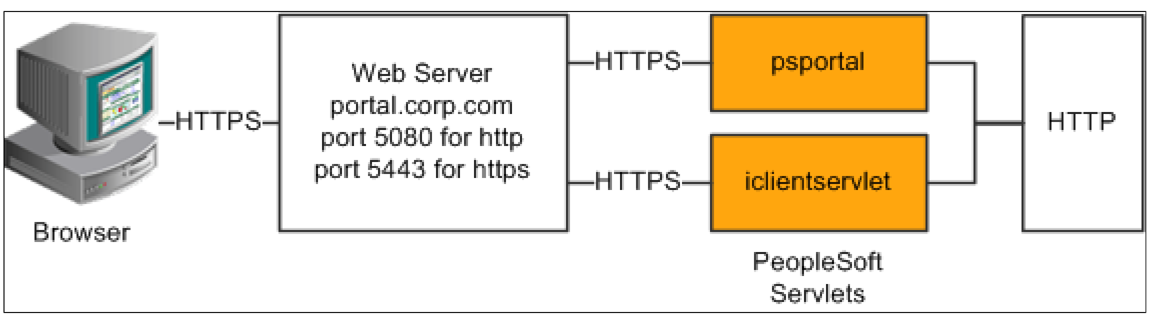
   Changes that you make to the component in PeopleSoft Application Designer are not reflected in the portal until you do another menu import.

Related Links

"Using PeopleSoft Application Designer" (Application Designer Developer's Guide)

Improving Same-Server Performance Under SSL

You might run portal and PeopleSoft applications over SSL connections to secure the transmissions between the browser and the web server. However, defining each homepage pagelet as an HTTPS request can cause portal performance to degrade significantly. To achieve secure transmissions between the browser and the server and to achieve acceptable homepage performance, configure the portal to use HTTP connections whenever it needs to talk to a PeopleSoft application that is hosted on the same server as the portal. This diagram shows this setup:



This configuration requires changes to the PeopleTools web configuration properties file. It also requires that a web server filter be set up to block non-SSL connections, except those from the portal. The portal does not need to use HTTP or HTTPS to connect to the PeopleSoft Pure Internet Architecture servlet hosting the same website. This configuration is needed only when multiple sites are used on the same web server.

Web Profile Settings

You must tell the portal to use HTTP for all connections that it makes back to the same server on which it is hosted. To do so, modify the web profile configuration for the portal. Change these properties:

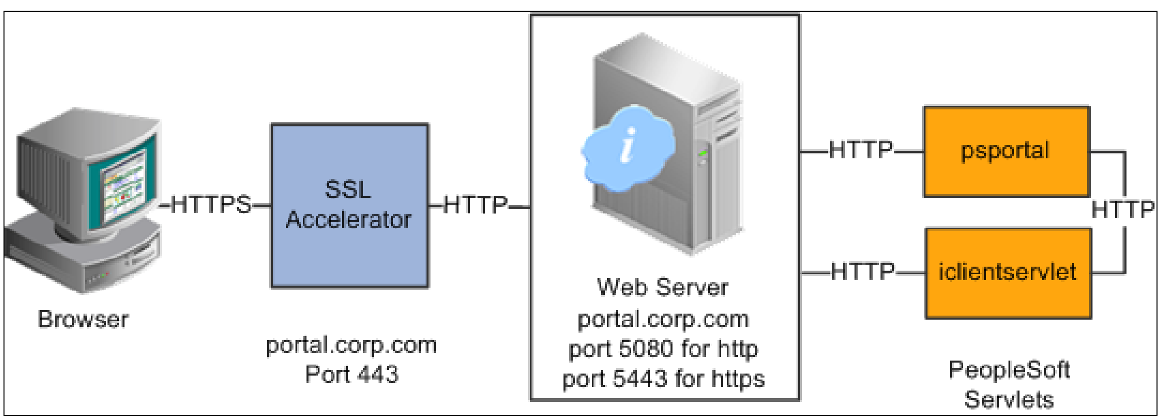
Web Profile Page	*Field Name*	*Field Value*
Security	**PIA use HTTP Same Server**	Selected.
Virtual Addressing	**Protocol**(in the **Default Addressing** region of the page)	*HTTPS.*
Virtual Addressing	**Port**(in the **Default Addressing** region of the page)	The port on which the web server is listening for HTTPS requests, for example, *5443.* You can leave this field blank if you use the default SSL portal of 443.

Related Links

Configuring Web Profiles
Proxy Architecture and Relative URLs

Using SSL Accelerators

You can improve SSL performance by adding an SSL accelerator in front of the web server. You need to have the portal use HTTP for requests to the PeopleSoft Pure Internet Architecture to prevent the portal connection to the PeopleSoft Pure Internet Architecture from going back through the accelerator. This diagram shows this setup:



The configuration is similar to that for configuring a portal running under SSL to use HTTP connections to the PeopleSoft Pure Internet Architecture on the same server; however, you must change the default port to point to the port on which the SSL accelerator is configured to listen.

Important! You must continue to configure Web Server to listen for HTTPS requests, even though the accelerator intercepts all SSL connections. Failure to do so causes the portal to function incorrectly.

Web Profile Settings

Change these properties:

Web Profile Page	***Field Name***	***Field Value***
Security	**PIA use HTTP Same Server**	Selected. (This might reduce the amount of time it takes to make an HTTP request to the portal.)
Virtual Addressing	**Protocol**(in the **Default Addressing** region of the page)	*HTTPS.*
Virtual Addressing	**Port**(in the **Default Addressing** region of the page)	The port on which the SSL accelerator is listening for HTTPS requests, for example, *443.*

Related Links

Configuring Web Profiles
“Understanding WebLogic” (System and Server Administration)

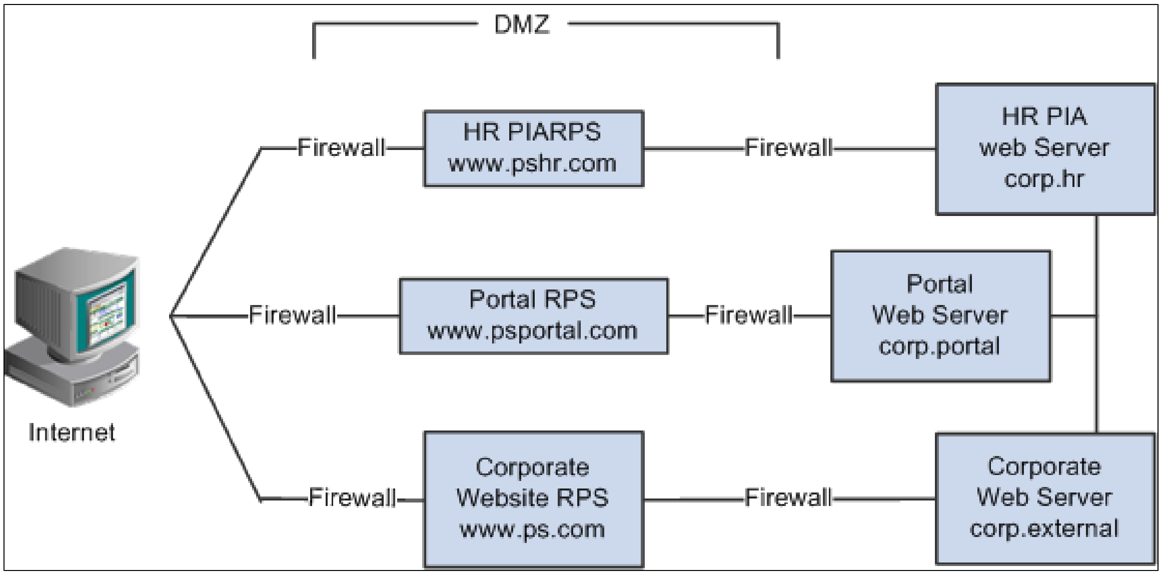
Using Reverse Proxy Servers

This section provides an overview of reverse proxy servers and discusses how to:

- Make reverse proxy settings.
- Issue requests by using a proxy server.
- Specify hosts that should not be proxied.

Understanding Reverse Proxy Servers

A reverse proxy server accepts requests from client browsers on behalf of another web server and then relays those requests to the web server. The reverse proxy server checks web requests and responses to ensure that they adhere to network security policies. For example, a reverse proxy server validates that a command intended for the target web server is a valid command for that server. In this diagram, the HR web server, corp.hr, is behind a reverse proxy server, www.pshr.com:



All internet requests for the HR web server are relayed by www.pshr.com to corp.hr. The site URL routes the client to the proxy as if it were a web server. Replicated content is delivered from the proxy cache to the external client without exposing the origin server or the network residing safely behind the firewall. You can balance the load of an overtaxed web server by installing multiple instances of the web server.

Any external content must come from a node that's registered in the portal database so that the portal servlet can look up the node name based on the URL of the content. The external content HTML is rewritten to use that proxied format.

Related Links

Using Firewalls with Reverse Proxy Servers

Using Reverse Proxy Servers with SSL Accelerators

Making Reverse Proxy Settings

On the Web Profile - Virtual Addressing page, specify the reverse proxy server to use by entering it in the **Reverse Proxy Server List** grid. Include the protocol, the host name, the HTTP port, and the HTTPS port.

Related Links

Configuring Virtual Addressing

Issuing Requests by Using a Proxy Server

For you to make a portal issue requests by using a proxy server, the proxy server settings are controlled by these Java system properties:

- http.proxyHost
- http.proxyPort
- https.proxyHost
- https.proxyPort

To use a proxy server for the HTTP protocol, set these properties:

```
http.proxyHost=http_proxy_server_host_name
http.proxyPort=http_proxy_server_port
```

To use a proxy server for the HTTPS protocol, set these properties:

```
https.proxyHost=https_proxy_server_host_name
https.proxyPort=https_proxy_server_port
```

The properties must be passed to the Java interpreter as command-line directives.

Alter the setEnv.cmd or setEnv.sh file located in WebLogic domain's bin folder. Define the properties on the command line that starts the Java Virtual Machine like this:

```
set PROXY=-Dhttp.proxyHost=proxyhostname -Dhttp.proxyPort=proxy_port -Dhttps.proxyH⇒

ost=proxyhostname -Dhttps.proxyPort=proxy_port
```

Later add the preceding property in startPIA.cmd or startPIA.sh in the Java command line that starts the WebLogic server domain.

```
%JAVA_HOME%\bin\java -ms64m -mx64m
-classpath %JAVA_CLASSPATH% %PROXY%
-Dweblogic.class.path=%WEBLOGIC_CLASSPATH%
-Dweblogic.home=. -Djava.security.manager -Djava.security.policy==.\weblogic.policy⇒

 weblogic.Server

goto finish
```

Specifying Hosts That Should Not Be Proxied

If your portal implementation retrieves content from both an intranet and the internet, you may have a problem retrieving both internal and external content if the proxy server is not aware of the address of the intranet content. You can prevent this situation by specifying that certain hosts (such as your intranet hosts) should not be accessed through the proxy server.

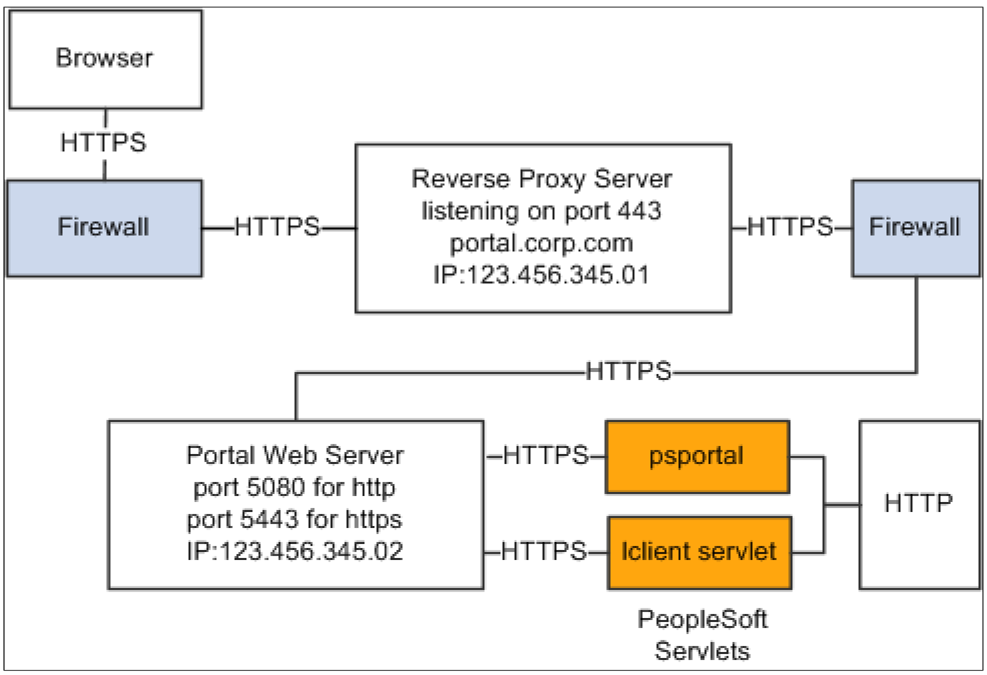
For Oracle WebLogic, specify the hosts not to be proxied in the HTTP_PROXY_NONPROXY_HTTPHOSTS and HTTP_PROXY_NONPROXY_HTTPSHOSTS environment variables with the setEnv.cmd (setEnv.sh in Unix) script. Refer to comments in setEnv for specific syntax.

You can list a set of patterns that should not be accessed through proxies, with each one separated by the pipe character (|). This example shows a setting that makes all HTTP connections to the eng domain bypass the proxy settings:

```
*.eng|*.eng.example.com
```

Using Firewalls with Reverse Proxy Servers

You can set up a buffer zone in front of the portal web server to prevent unauthorized access to the portal web server and create a more secure environment. A buffer zone is typically configured with a firewall that allows access to a reverse proxy server, which relays incoming requests through a second firewall to the portal web server. This diagram shows this setup:



The configuration is similar to that for configuring a portal running under SSL to use HTTP connections to the PeopleSoft Pure Internet Architecture on the same server; however, you must make these adjustments in the web profile:

Web Profile Page	***Field Name***	***Field Value***
Security	**PIA use HTTP Same Server**	Selected.
Virtual Addressing	**Protocol**(in the **Default Addressing** region of the page)	*HTTPS.*
Virtual Addressing	**Name**(in the **Default Addressing** region of the page)	The DNS name of the reverse proxy server, for example, *portal.corp.com.*

Web Profile Page	***Field Name***	***Field Value***
Virtual Addressing	**Port**(in the **Default Addressing** region of the page)	The port on which the reverse proxy server is listening for HTTPS requests if it's different from the port on which the web server is listening.

Hosts File Setup

The hosts file on the portal web server must have an entry that directs DNS requests for the content provider server name (the reverse proxy server in the previous example) to the portal web server, for example, 123.456.345.02. This enables the portal to make requests for content that is hosted on the same server directly without going back through the reverse proxy server.

For example:

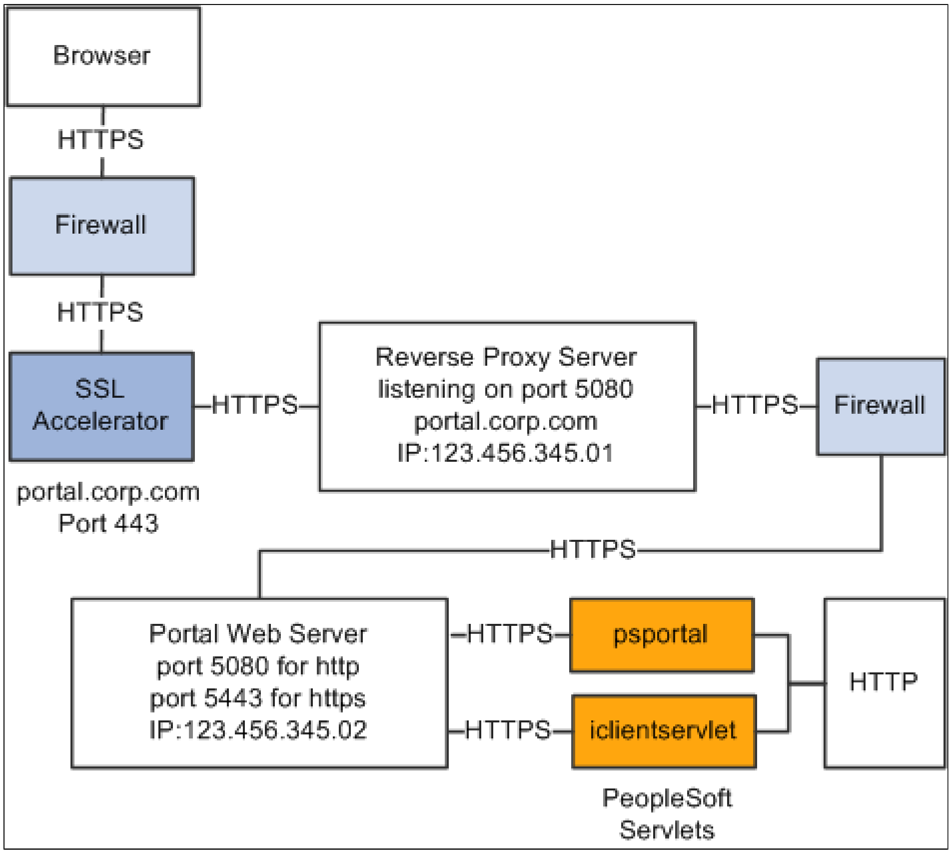
```
123.456.345.02   portal.corp.com
```

Related Links

Configuring Web Profiles

Using Reverse Proxy Servers with SSL Accelerators

You can combine an SSL accelerator with a reverse proxy server, as illustrated in this diagram:



The configuration is similar to that for configuring a portal running under SSL to use HTTP connections to the PeopleSoft Pure Internet Architecture on the same server; however, you must make these adjustments in the web profile:

Web Profile Page	***Field Name***	***Field Value***
Security	**PIA use HTTP Same Server**	Selected.
Virtual Addressing	**Protocol**(in the **Default Addressing** region of the page)	Set to *HTTPS.*
Virtual Addressing	**Name**(in the **Default Addressing** region of the page)	The DNS name of the reverse proxy server, for example, *portal.corp.com.*
Virtual Addressing	**Port**(in the **Default Addressing** region of the page)	The port on which the SSL accelerator is listening for HTTPS requests, for example, *443.*

Hosts File Setup

The hosts file on the portal web server must have an entry that directs DNS requests for the content provider server name (portal.corp.com) to the portal web server, for example, 123.456.345.02. This enables the portal to make requests for content that is hosted on the same server directly without going back through the reverse proxy server.

For example:

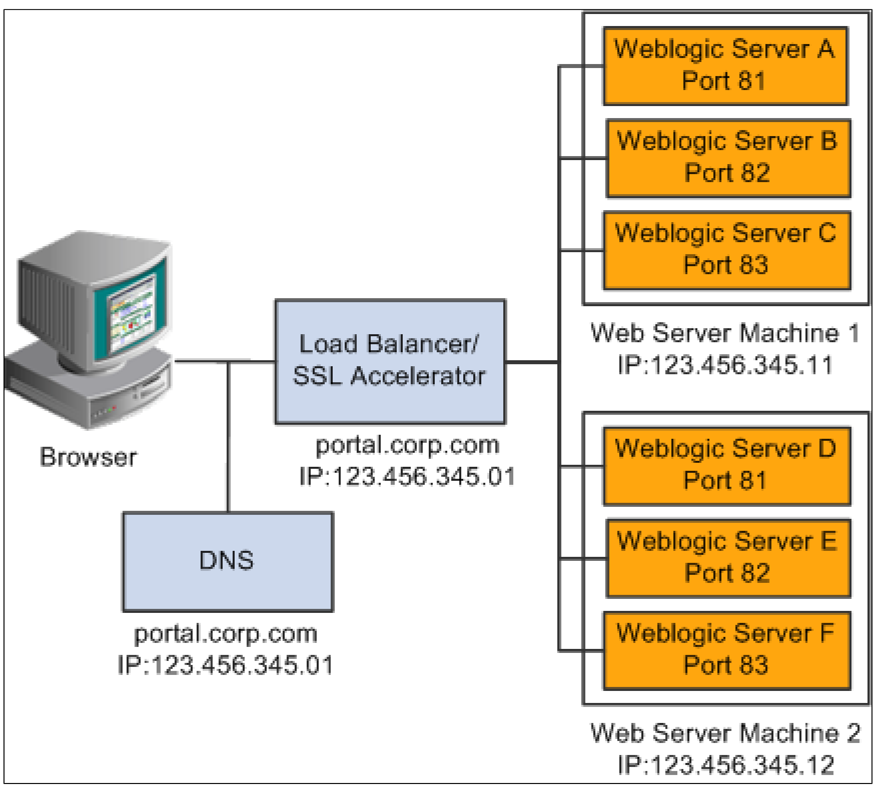
```
123.456.321.02   portal.corp.com
```

Related Links

Configuring Web Profiles
"Understanding WebLogic" (System and Server Administration)

Applying Load Balancing

You can apply load balancing to a portal by using a hardware load-balancing solution to distribute load across multiple instances of the web server on multiple web server machines. Many of today's hardware load balancers have SSL accelerating options that you should also consider. The number and size of web server machines and the number of web server instances per machine vary and determine load-balancing requirements and hardware selection. This diagram shows a possible load-balancing setup:



The hardware load balancing solution must be configured to enable sticky sessions, which ensures that all requests from a user's browser session are delivered to the same web server session. In the example previous diagram, if the initial request from the browser was directed to Oracle WebLogic Server F, then all subsequent requests from that browser session must be directed to the same instance of Oracle WebLogic.

Keep multiple web server sessions on a single machine to keep the heap size of the Java Virtual Machine manageable and to allow the Java garbage collection to work optimally.

Hosts File Configuration

The hosts file on the portal web server machines in the previous diagram—which we'll call portalws1 and portalws2—must have an entry that directs DNS requests for the portal content provider server name (portal.corp.com) to the portal web server (123.456.345.11 on portalws1). This allows the portal to request content hosted on the same server directly without going back through the load balancer.

This is an example hosts file on portalws1.corp.com:

```
123.456.345.11     portal.corp.com
123.456.345.11     portalws1.corp.com
```

Creating Access Control for Files on the Web Server

You can create dynamic access control to any parent folder and files under the system-defined **stogroup** storage root folder on your web server. In Application Designer, extend the PeopleTools-delivered PTPP_SG:ContentAuthorizorBase base class. Then, implement the Authorize method to determine whether a user is authorized to access a parent folder (or the parent folder and all subfolders) under the storage root. Then, on the Storage Group Management page, create a storage group, which serves as an alias to the storage group parent folder on the web server. This storage group associates your PeopleCode implementation to this parent folder. Finally, provide your users with the URLs to files on the web server.

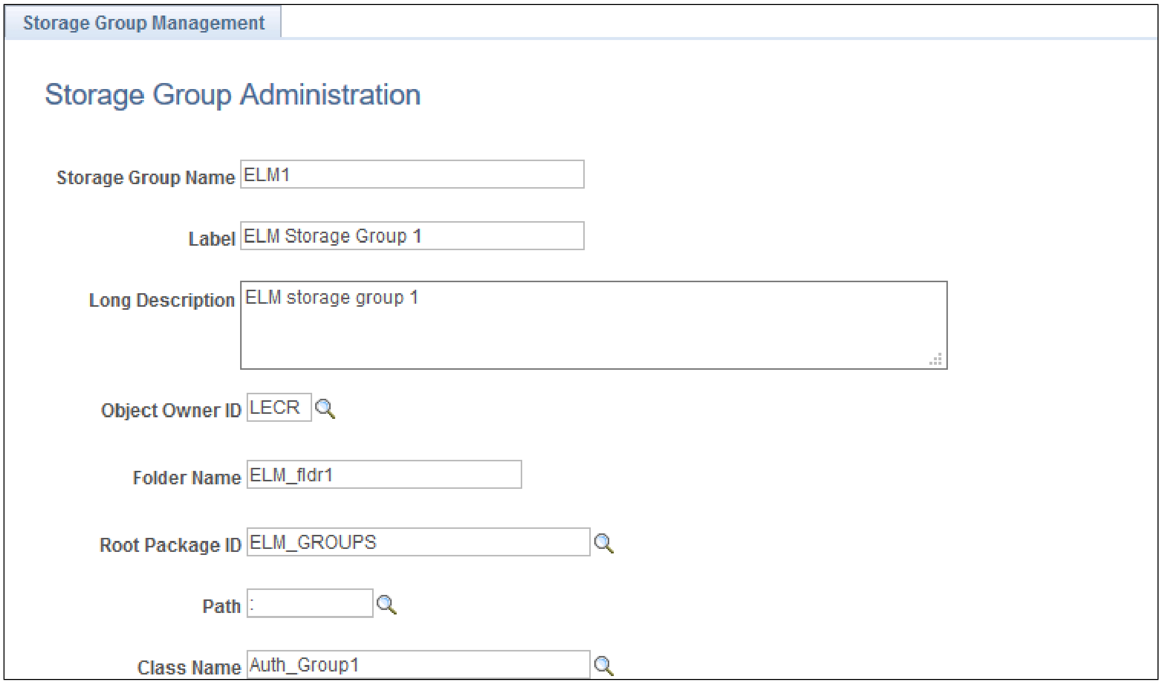
Storage Group Management Page

Use the Storage Group Management page to create an alias to the storage group parent folder on the web server and associate your PeopleCode implementation to the parent folder.

Navigation:

PeopleTools > **Portal** > **Storage Group Management**

This example illustrates the fields and controls on the Storage Group Management page. You can find definitions for the fields and controls later on this page.

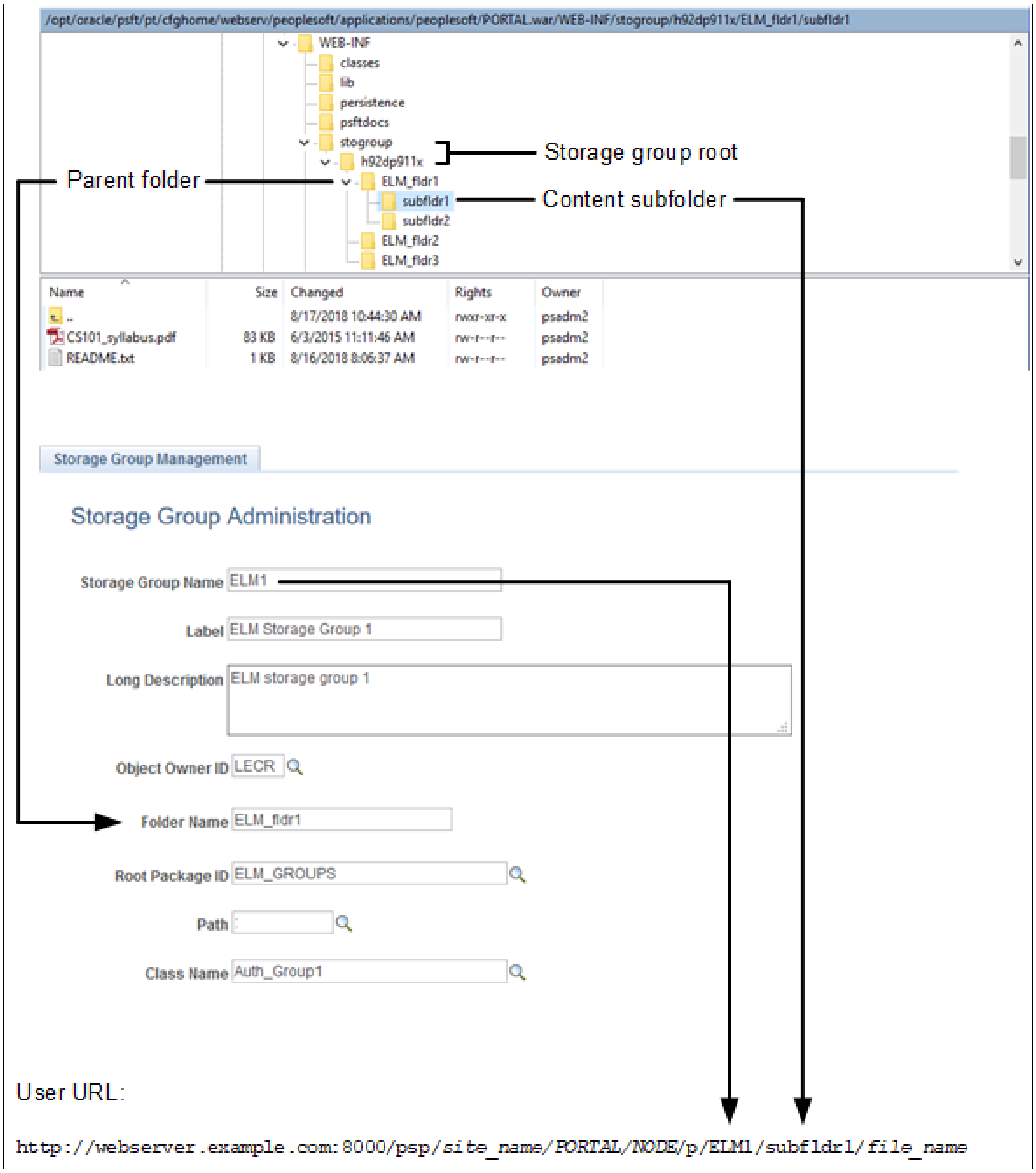


Field or Control	***Description***
Storage Group Name	Enter the name for the storage group, which serves as an alias for a parent folder on the web server. A parent folder must contain one or more subfolders.
Label	Enter a label for the storage group.
Long Description	Enter a description for the storage group.
Object Owner ID	Select the ID for the object owner.
Folder Name	Enter the name of the storage group parent folder on the web server.
Root Package ID	Select the name of the application package that contains your implementation of the PTPP_SG:ContentAuthorizorBase base class.
Path	Select the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class Name	Select the name of the application class that contains your implementation.

Setting Up Access Control

To set up access control, you create the storage group root folder and storage group parent folders on your web server. Then, you create a subfolder structure as needed and upload files to these subfolders on the web server.

The following diagram illustrates the relationship of the file system on the web server to the definition on the Storage Group Management page and to a URL provided to users.



To set up access control:

1. Create the *storage group root folder* under PS_CFG_HOME/peoplesoft/applications/peoplesoft/PORTAL.war/WEB-INF/. The required name for the storage group root folder is: **stogroup**/*site_name*.

Note: Use 755 as the permissions for the folders in the storage group hierarchy.

2. Create one or more storage group parent folders under this storage group root folder.
3. Under each parent folder, create one or more subfolders and upload files as needed.

Note: Files must be stored in and accessed from the subfolders, not from the storage group parent folder.

4. In Application Designer, create a custom implementation for the PTPP_SG:ContentAuthorizorBase base class.
5. Implement the "Authorize" (PeopleCode API Reference) method to return one of three authorization results for the current user: Authorized for the specified subfolder, authorized for the specified subfolder and any of its subfolders, or denied.
6. Using the Storage Group Management page, associate your PeopleCode implementation to a parent folder on the web server.
7. Provide users with URLs to the files that they can access on the web server using the **psp** servlet. For example, the URL to CS101_syllabus.pdf in the subfldr1 folder would be:

```
http://webserver.example.com:8000/psp/site_name/PORTAL/NODE/p/ELM1/subfldr1/CS⇒

101_syllabus.pdf
```

You can use a function provided in a delivered function library to facilitate generation of these URLs. For example, use the following PeopleCode program to generate the URL for this same CS101_syllabus.pdf file. The value returned can be assigned to a hyperlink or button, for example.

```
Declare Function GetStorageGroupContentURL PeopleCode FUNCLIB_PORTAL.PTPP_SG_F⇒

UNC FieldFormula;
ViewContentURL(GetStorageGroupContentURL(%Portal, %Node, "ELM1", "subfldr1/CS1⇒

01_syllabus.pdf"), True);
```

GetStorageGroupContentURL

Syntax

GetStorageGroupContentURL(*PortalName*, *NodeName*, *StorageGroupName*, *rel_path*)

Description

Use the GetStorageGroupContentURL function to generate a private content URL for a specific file.

Parameters

Parameter	*Description*
PortalName	Specifies the portal name as a string value.

Parameter	***Description***
NodeName	Specifies the node name as a string value.
StorageGroupName	Specifies the storage group name as a string value.
rel_path	Specifies the relative file path from the storage group parent folder as a string value.

Returns

A string value.

Example

```
Declare Function GetStorageGroupContentURL PeopleCode FUNCLIB_PORTAL.PTPP_SG_FUNC F⇒

ieldFormula;
ViewContentURL(GetStorageGroupContentURL(%Portal, %Node, "ELM", "course1/index.html⇒

"), True);
```

Related Links

"ViewContentURL" (PeopleCode Language Reference)

Chapter 10

Working with Navigation Pages and Navigation Collections

Understanding Navigation Pages

Important! Standard navigation pages (DFANs), custom navigation pages (CFANs), SmartNavigation, and other forms of classic navigation are no longer supported. The information in this topic pertains to a prior version of PeopleSoft classic applications only. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see “Working with Fluid Pages and Controls” (Applications User’s Guide).

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

- Standard navigation pages.

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

- Custom navigation pages.

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

  See Creating and Maintaining Navigation Collections

- SmartNavigation pages.

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

  Note: While existing SmartNavigation pages may be available in your PeopleSoft system, the SmartNavigation feature has been desupported.

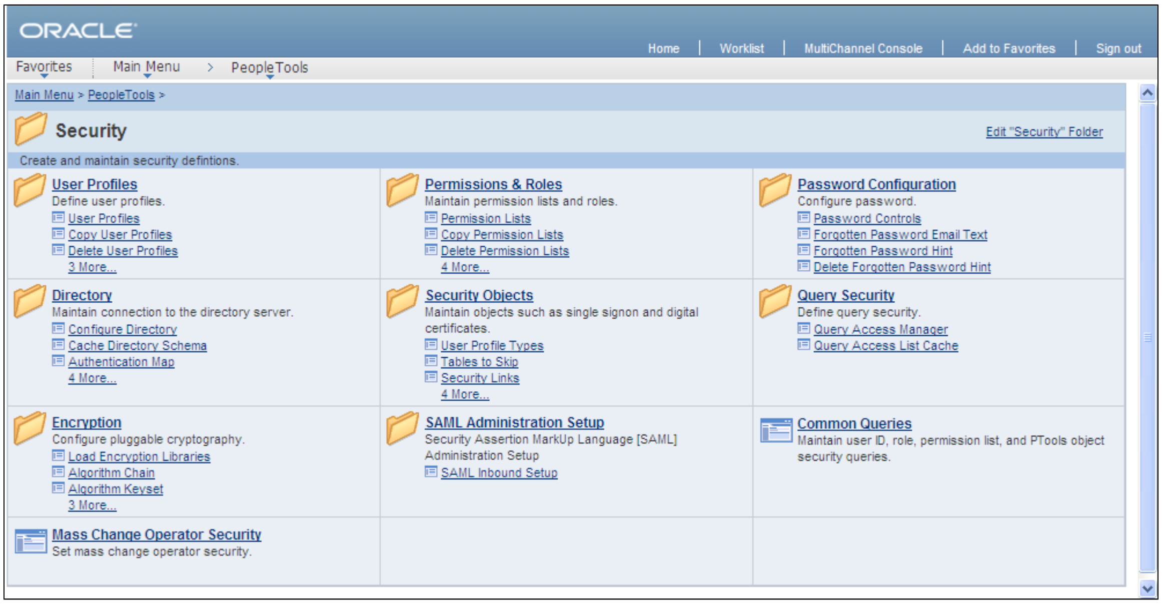
The following sections further describe standard navigation pages, custom navigation pages, and navigation page display caching.

Standard Navigation Pages

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

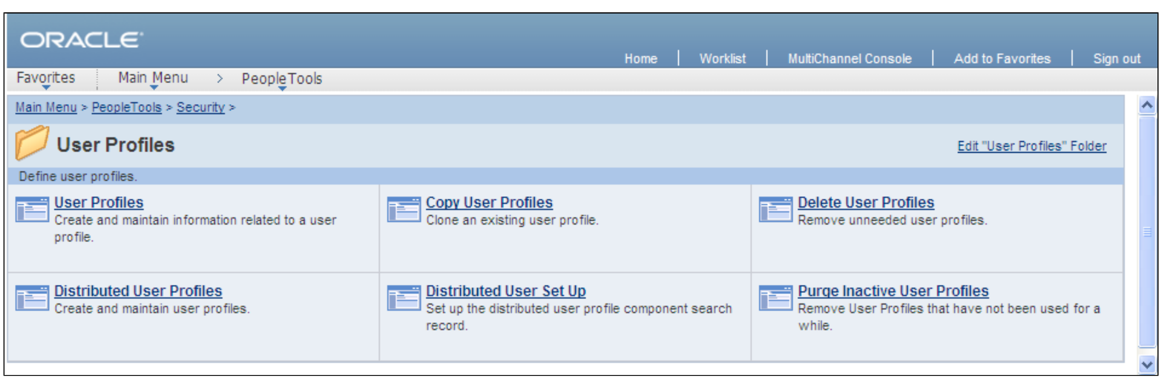
You access standard navigation pages by clicking a folder icon in the menu navigation. Items contained within the folder, including content references and other folders, are arranged in columns on the page.

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



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

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



Standard Navigation Page Display

To enable the display of standard navigation pages:

1. Select **PeopleTools** > **Portal** > **General Portal Settings.**
2. Select the **Is Folder Navigation Enabled** check box.
3. Enter *PT_PTPP_SCFNAV_BASEPAGE_SCR* (the delivered Base Navigation Page object) in the **Folder Navigation Object Name** field.

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

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

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

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

See Administering Folders, Administering Portal Definitions.

Folder and Link Sequence on Standard Navigation Pages

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

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

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

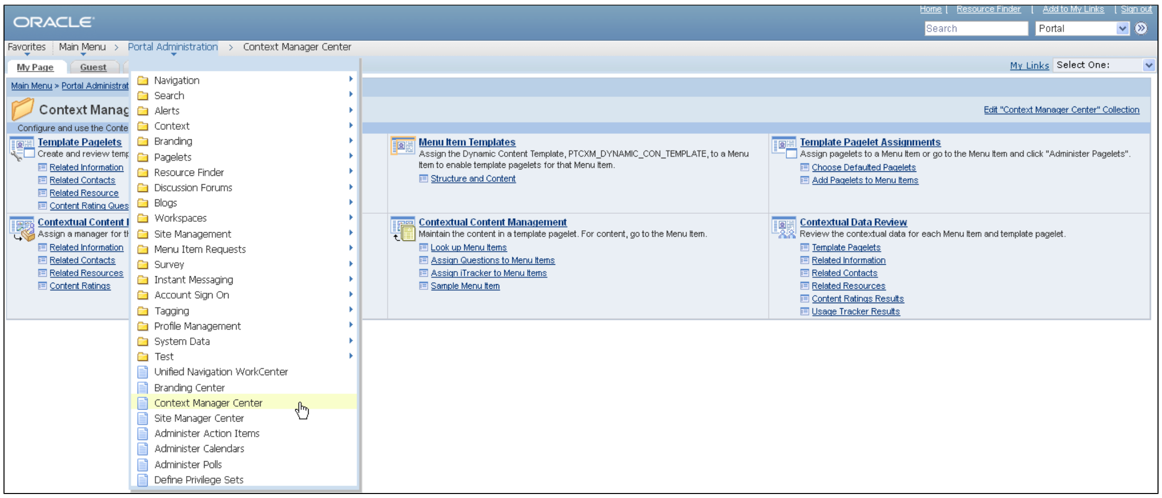
See Administering Folders, Administering Content References.

Custom Navigation Pages

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

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

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



Similar to standard navigation pages, when you click a folder link on a custom navigation page, you access another navigation page that reflects the contents of the selected folder. When you click a content reference link on a custom navigation page, you access the component to which the content reference points.

See Creating or Editing a Navigation Collection.

Folder and Link Sequence on Custom Navigation Pages

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

See Editing a Folder, Editing a Link.

Navigation Page Display Caching

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

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

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

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

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

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

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

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

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

Understanding Navigation Collections

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

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

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

After being defined, a navigation collection can be published as a custom navigation page or as a navigation pagelet for use on a homepage, dashboard page, or WorkCenter page. In addition, navigation collections can be accessed through a tile that has been defined to display the navigation collection and links in an activity guide.

When you publish navigation collections, you can choose to publish navigation collections as public content. If you do not choose to provide public access to a navigation collection, you can choose one of these two options for setting security access to published navigation collections:

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

Note: Public access is allowed by default.

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

Navigation collections can contain these elements:

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

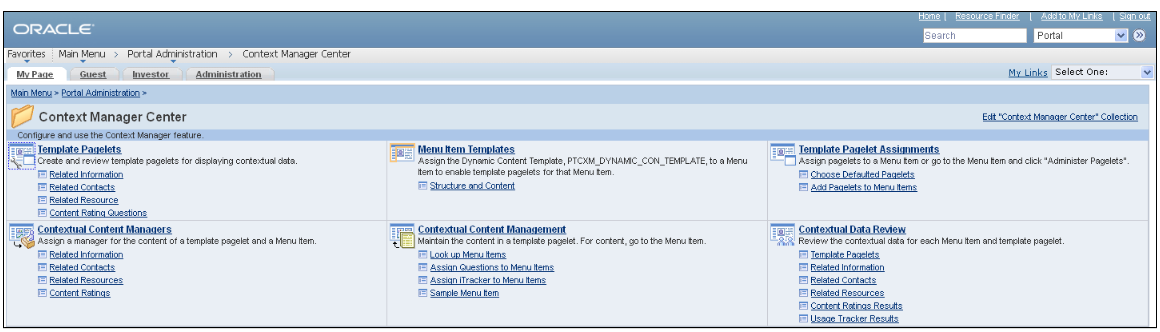
  The navigation collection will contain all contents of the menu folder and its subfolders.
- Links to remote, component-based content references.

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

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

Navigation Collection Published as a Navigation Page

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



Important! Standard navigation pages (DFANs), custom navigation pages (CFANs), SmartNavigation, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see “Working with Fluid Pages and Controls” (Applications User’s Guide). However, navigation collections published as pagelets and used on classic WorkCenters, but not on classic homepages, remain a supported navigation mechanism.

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

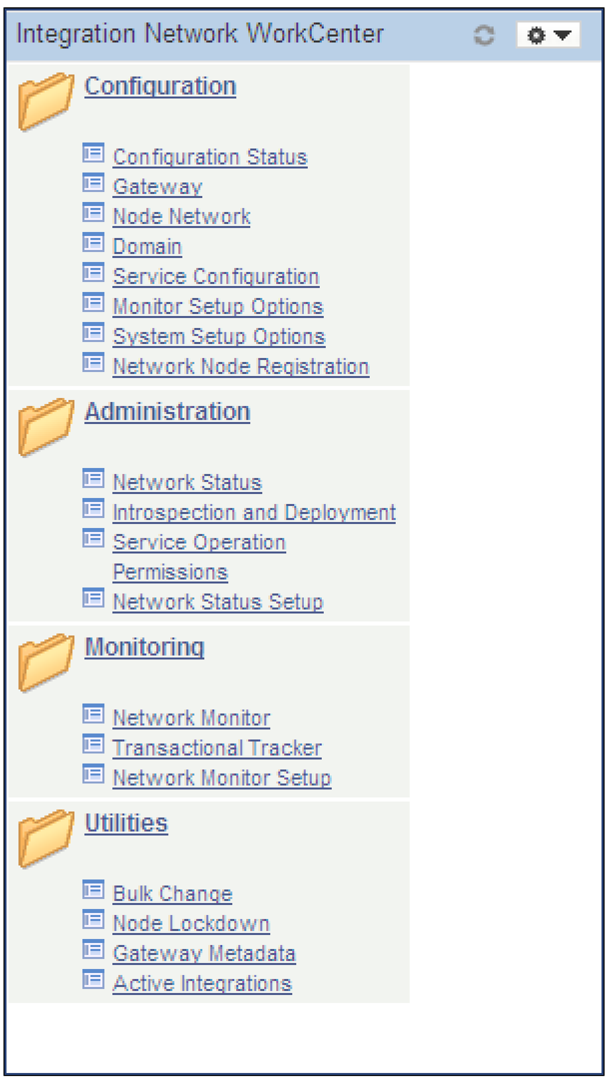
Click any folder link to access another navigation page that contains any folders and content references nested in the parent folder.

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

See Creating and Maintaining Navigation Collections.

Navigation Collection Published as a Navigation Pagelet

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



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

Related Links

Publishing Navigation Collections

Understanding Portal Utility Processes

Fluid Navigation Collections

PeopleTools provides several features that combined together allow you to display what are referred to as "fluid navigation collections." These features allow you to launch navigation collections in a fluid activity guide wrapper. You can choose to re-present existing navigation collections consisting entirely of classic content references as well as add fluid content references to new or existing navigation collections. Navigation collections are created and maintained exactly the same way that they always have been. After a navigation collection has been defined, you can determine whether to launch that navigation collection inside a fluid wrapper as a "fluid navigation collection." These fluid navigation collections are another way that you can create WorkCenter-like functionality in fluid applications.

The PeopleTools features that support fluid navigation collections include:

- The activity guide framework, which is the wrapper used to display the navigation collection links and target content.
- The PTPP_NAVCOLL activity guide data source and supporting templates.
- Tile Wizard, which can be used to create a tile definition to launch the fluid navigation collection. (See Step 2: Data Source Information - Navigation Collection for more information.)

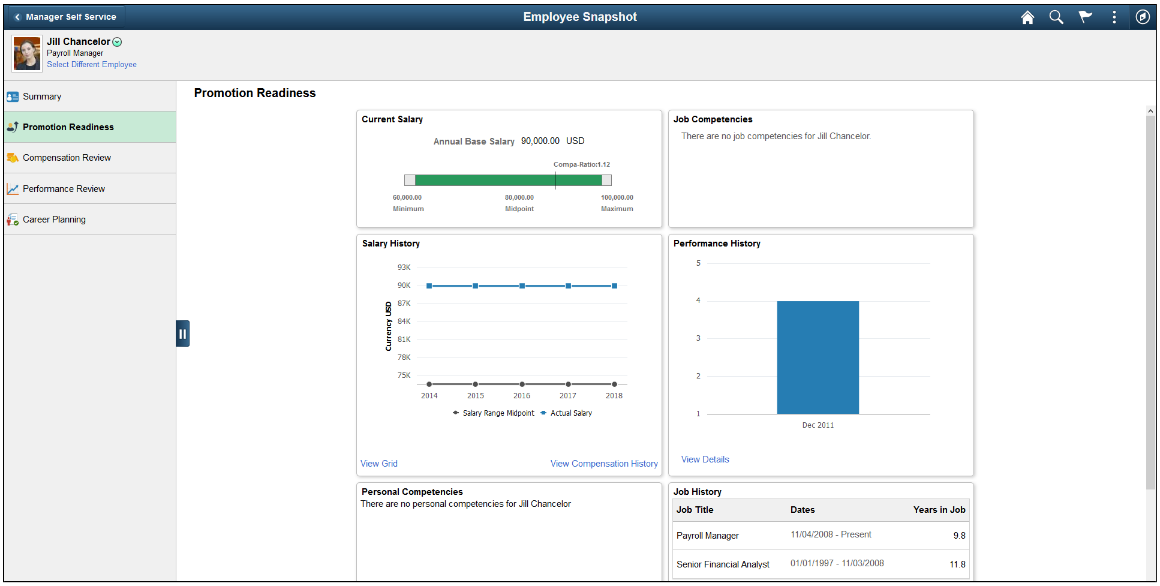
There are two ways to launch a fluid navigation collection from a tile:

- In an *optimized,* non-guided, non-sequential activity guide (using the PTPPNAVCOL activity guide template).
- In a *non-optimized,* non-guided, non-sequential activity guide (using the PTPPNONOPT activity guide template).

In addition, you can create a NavBar tile from the navigation collection to display the navigation collection links within the NavBar.

Note: By design, on a smart phone, a fluid navigation collection is displayed as the links only (that is, what is displayed in the left panel on other devices). Also by default, clicking a link transfers you to the target component and exits the fluid navigation collection. You can return to the navigation collection using the Back button. To override this behavior, you can add the JumpPage=N query string parameter to the URL that launches the navigation collection.

The following example shows the Employee Snapshot navigation collection displayed in an optimized activity guide. With an optimized activity guide, folders and navigation links are displayed in the left panel.



Guidelines for Creating Fluid Navigation Collections

Note the following guidelines and limitations for fluid navigation collections:

- While classic content references can be included in fluid navigation collections, there are differences when these classic pages are viewed on smartphones. The text and content of the classic page will be displayed at a readable size. However, these classic pages will often exhibit other usability issues on smartphones since the fixed page layout may still require horizontal scrolling to view additional fields on the page.
- If the navigation collection is to be rendered in the optimized format, then any fluid components must adhere to the guidelines for optimized activity guides. See Guidelines and Limitations for Using Fluid Components in Activity Guides for more information.
- If the link target is a fluid component that employs an internal two-panel construct (that is, it either includes a Side Page 1 page, was built from the PeopleTools 8.54 PSL_TWOPANEL layout page, or was designed with an internal left panel similar to PSL_TWOPANEL), you must employ one of these methodologies to accommodate this link target:
  - Specify either the Replace Window option or the Open in a new window option for the link.
  - Enable left panel collision handling with the slide-out left panel on the fluid navigation collection. See "Left Panel Collision Handling with Slide-out Left Panels" (Fluid User Interface Developer's Guide) for more information.

  Note: This accommodation is not required for a non-optimized rendering of the navigation collection.
- If the link target is a fluid component that opens the component search page (that is, the search type is not set to None), specify either the Replace Window option or the Open in a new window option for this link.

- If the link target is a fluid wrapper component—that is, a master/detail component, a fluid activity guide, or another fluid navigation collection—specify either the Replace Window option or the Open in a new window option for this link.

Important! A fluid wrapper component can never be displayed inside another fluid wrapper component.

- By default, the first component in the navigation collection is opened automatically when the navigation collection is launched. Therefore, the first component should be the most frequently used component and ideally should be quick to load. More importantly, the first component cannot be a fluid component with a two-panel layout.
- For classic dashboards, specify either the Replace Window option or the Open in a new window option for the dashboard item within the navigation collection definition.
- For classic WorkCenters, specify the Replace Window option (or the Open in a new window option).
- For a classic component that has been configured with the Related Content frame, specify the Open in a new window option.
- Fluid navigation collections and activity guides do not support conditional navigation.

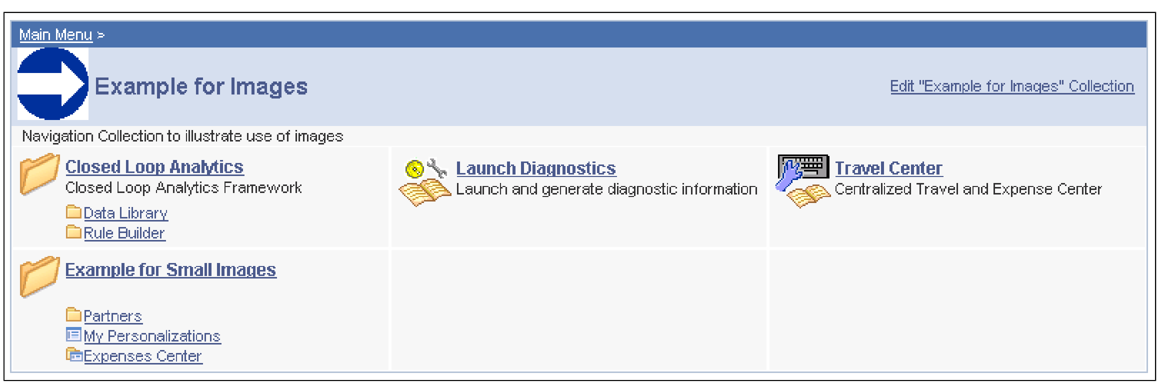
Related Links

Using Tile Wizard
Manually Creating Tile Definitions to Launch Fluid Navigation Collections
PeopleSoft Fluid UX Standards: Navigation Collections

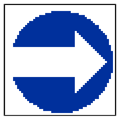
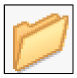
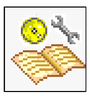
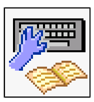
Navigation Collection Images

The images that you define for your navigation collections are published and displayed on navigation collection pages. You can associate images with navigation collections themselves, as well as with navigation collection folders and content references. You can define these images at the system, registry, source, and individual navigation collection levels.

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



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

Term	***Definition***
	This image was defined as the collection image for this specific navigation collection.
	This image was defined as the default large folder image on the System Options page, and was not overridden at the source reference, registry, or navigation collection level. Large folder images appear for folders at the top level of a navigation collection publication.
	This image was defined as the content image in the navigation collection definition. Large content images appear for content references at the top level of a navigation collection publication.
	This image represents a navigation collection that was added to this navigation collection. This image was selected as the collection image in the referenced navigation collection definition, and was not overridden in this navigation collection definition. Large collection images appear for navigation collections at the top level of a navigation collection publication.
	This image was defined as the small folder image. Small folder images appear for folders nested within top-level folders on a navigation collection publication.
	This image was defined as the small content image. Small content images appear for content references nested within top-level folders of a navigation collection publication.
	This image was defined as the small collection image. Small collection images appear for navigation collections nested within top-level folders of a navigation collection publication.

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

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

For example, while the Expenses Center is displayed with its default small image, when you access the Example for Small Images navigation page, the Expenses Center appears with its unique large image.

Common Elements Used on Navigation Pages

Important! Standard navigation pages (DFANs), custom navigation pages (CFANs), SmartNavigation, and other forms of classic navigation are no longer supported. The information in this topic pertains to a prior version of PeopleSoft classic applications only. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see “Working with Fluid Pages and Controls” (Applications User’s Guide).

Field or Control	***Description***
Breadcrumbs	Your portal administrator can configure your navigation pages to display the navigation path or breadcrumbs. These breadcrumbs appear across the top of a navigation page and are generated according to the navigation path used to access the current content displayed. You can click the breadcrumbs to navigate through the path that you used to access the current content. See Setting Up Navigation Collection Options.
Edit <folder name> Folder	The **Edit <folder name> Folder** link appears if you have accessed a standard navigation page and you have permissions defined for your user ID or role that enable you to access the Structure and Content component. Select the **Edit <folder name> Folder** link to access the Folder Administration page, where you can edit aspects of the folder that was used to generate the selected standard navigation page. See Defining Folders.
Edit <Navigation Collection name> Collection	The **Edit <Navigation Collection name> Collection** link appears if you have accessed a custom navigation page and you have permissions defined for your user ID or role that enable you to edit navigation collections. Select the **Edit <Navigation Collection name> Collection** link to access the navigation collections component, where you can edit the definition that was used to build the selected custom navigation page. See Creating and Maintaining Navigation Collections.
	Click the spell-check button to check the spelling of text in the associated field. See “Using Spell Check” (Applications User’s Guide).

Setting Up Navigation Collection Options

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

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

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

Understanding Navigation Collection Options

You define and override navigation collection options at different levels. Options defined at the system level provide default values, with options defined at the registry, source, and navigation collection levels being capable of overriding each previous level.

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

Options	***System (default)***	***Registry (overrides system default)***	***Source Reference (overrides system and registry values)***	***Collection Reference (overrides system, registry, and source reference values)***
Registry object prefix	X	None available	None available	None available
Show breadcrumbs	X	X	None available	None available
Show main menu breadcrumb	X	X	None available	None available
Registry node name	X	None available	None available	None available
Large (parent) image	X	X	X (with attribute name PTPP_IMAGE)	X
Small (child) image	X	X	None available	None available
Style sheet	X	X	None available	X
Show images	X	X	None available	X

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

You define and set navigation collection options on these pages:

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

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

Related Links

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

Common Elements Used in This Section

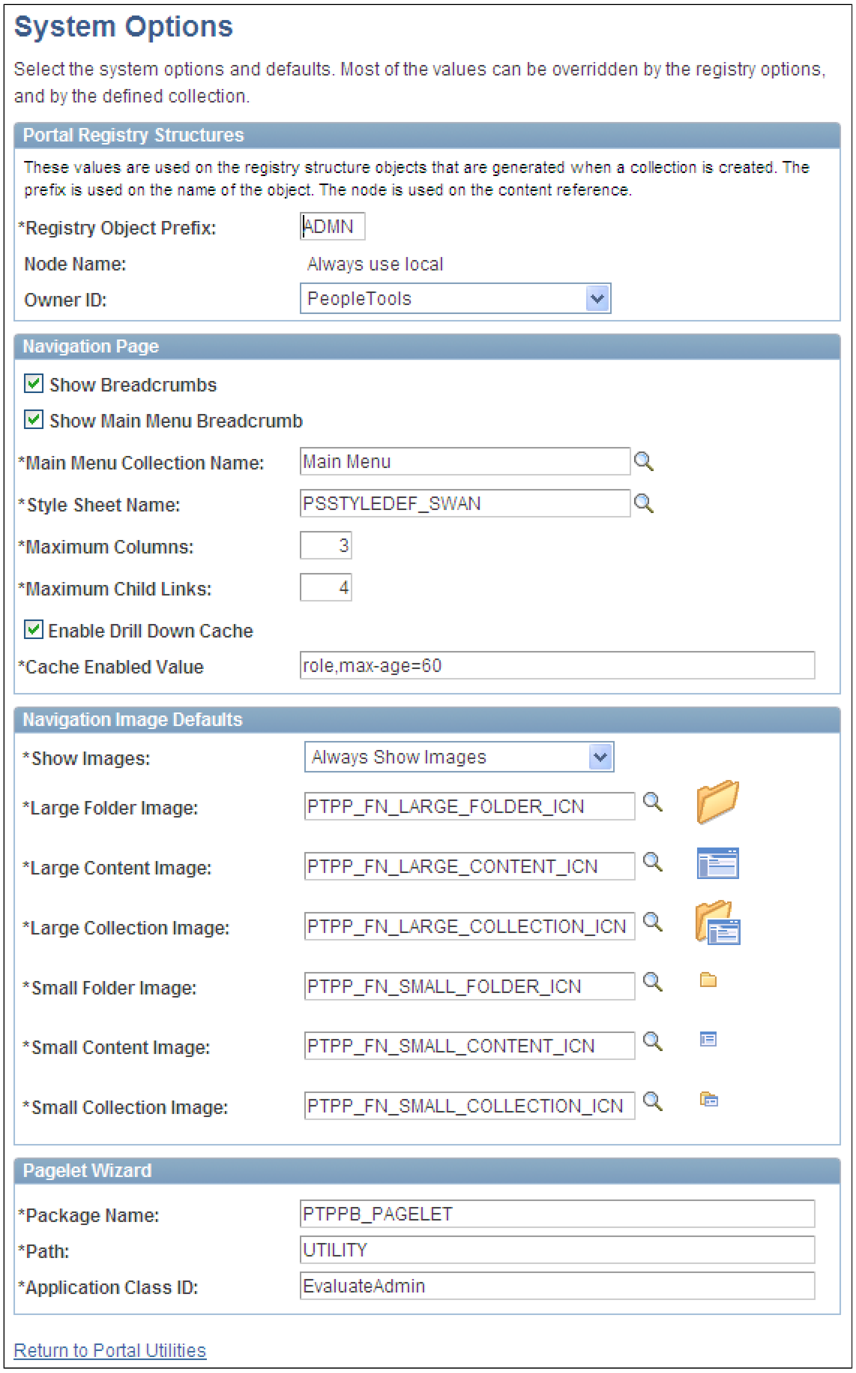
Field or Control	***Description***
Show Breadcrumbs	Select to indicate that you want navigation path to appear on published navigation pages. These breadcrumbs reflect the path taken by users as they navigate through left-navigation menu folders and navigation page folders. Displaying breadcrumbs facilitates user navigation because the left-navigation menu is not updated to reflect the user's navigation through navigation pages. The first breadcrumb in the Main Menu collection points to the delivered PT_PTPP_PORTAL_ROOT navigation collection, which is based on the PORTAL_ROOT_OBJECT folder. Alternatively, you can select a different navigation collection to be used as the first (main menu) breadcrumb using the **Main Menu Collection Name** field, or you can choose not to display a first breadcrumb. The next set of breadcrumbs displayed depend on the left-navigation menu folders that the user opened to access the navigation collection page. Subsequent breadcrumbs displayed depend on the folder path that the user accesses on the navigation collection page.
Show Main Menu Breadcrumb	If you selected the Show Breadcrumbs option, select this option to include the Main Menu collection name as the first entry in the breadcrumb path displayed on the navigation collection page publication. If you deselect this option, the first entry in the breadcrumb path depends on the level-1 left-navigation folder that the user opens to access the navigation collection page. The **Main Menu** link does not appear as the first link in the breadcrumbs.
Main Menu Collection Name	This field appears when the **Show Main Menu Breadcrumb** option is selected. The default value is *Main Menu,* which designates that the first breadcrumb displayed accesses the Main Menu page. You can choose to replace the first breadcrumb **Main Menu** link with a link that accesses an alternate navigation collection that you have created. The first breadcrumb Main Menu link to the Main Menu page provides access to the entire navigation for your database, filtered by security. You may want to provide a first breadcrumb link that provides access to a navigation collection that you have configured to contain only selected folders and content references.

Field or Control	***Description***
Style Sheet Name	Select the cascading style sheet (CSS) that you want to use to generate navigation collection pages. Cascading style sheets are defined in PeopleSoft Application Designer. This setting applies to both navigation pages and navigation pagelets. **Note:** You must use the same style class names that are found on the stylesheet PTPP_STYLEDEFN. See “Understanding Style Sheets and Style Classes” (Application Designer Developer’s Guide).“Creating Standard and Sub Style Sheets” (Application Designer Developer’s Guide)
Maximum Columns	Enter the maximum number of columns to display on a navigation page. Navigation pagelets display one or two columns depending on the user's settings on the Personalize Layout page. Narrow pagelets display contents in one column. Wide pagelets display contents in three columns. This setting applies to navigation pages only. See “Personalizing Classic Homepages and Dashboards” (Applications User’s Guide).
Maximum Child Links	Enter the maximum number of links to display beneath each folder on the navigation collection publication. If the number of links in the folder exceeds this number, an **X More...** link appears beneath the folder that jumps to a page from which you can access all links stored in the navigation collection folder. For example, if you set **Maximum Child Links** to *4* and the navigation collection folder contains five links, three links appear on the navigation collection page along with the **X More...** link. When you select the **X More...** link, a page containing all five links in the navigation collection folder appears. This setting applies to both navigation pages and navigation pagelets.

Defining Navigation Collection System Options

Access the System Options page. (Select **PeopleTools** > **Portal** > **Portal Utilities** > **Nav Collection System Options**.)

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



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

See Understanding Navigation Collection Options.

Portal Registry Structures

Field or Control	***Description***
Registry Object Prefix	Enter the value that the system prefixes to registry structure object names (content and folder references) that are created when you publish navigation collections, template pagelets, and feed definitions. For example, the object name of a published navigation pagelet has this format: *PREFIX*_SC_PGT_*OBJECT_LABEL*. The object name of a published navigation page has this format: *PREFIX*_SC_SP_*OBJECT_LABEL*.
Node Name	You should not change the default value delivered with your system. This default value corresponds to the content provider node for your database. For example, the following databases are delivered with the listed default node names: PeopleSoft FSCM: *ERP* PeopleSoft CRM: *CRM* PeopleSoft HCM: *HRMS* PeopleSoft Interaction Hub: *EMPL*
Owner ID	Use this value to easily identify navigation collections created by your organization when searching for navigation collections through PeopleSoft Application Designer or SQL queries of the portal registry tables. Do not select a delivered PeopleSoft owner ID. You can define unique owner IDs for your organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer.

Navigation Page

Field or Control	***Description***
Enable Drill Down Cache	Select to enable caching for navigation pages that you access by selecting a folder on a navigation page. You must also enter a value in the **Cache Enabled Value** field on this page. See Navigation Page Display Caching.

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

Navigation Image Defaults

Field or Control	***Description***
Show Images	Select one of these values: *Always Show Images:* Select if you always want images to appear on navigation pages and pagelets. *Never Show Images:* Select if you never want images to appear on navigation pages and pagelets. *Only Show Defined Images:* Select if you want only specifically defined images to appear on navigation pages and pagelets. If you select this option, only images defined as part of the navigation collection or defined on the source reference are displayed on the navigation collection's publications. System-level and registry-level default images are not displayed.
Large Folder Image	Select the default image that you want to display for top-level folders on a navigation page or pagelet.
Large Content Image	Select the default image that you want to display for top-level content references on a navigation page or pagelet.
Large Collection Image	Select the default image that you want to display for top-level navigation collection references on a navigation page or pagelet.
Small Folder Image	Select the default image that you want to display for a folder within a folder on a Navigation page or pagelet.
Small Content Image	Select the default image that you want to display for a content reference within a folder on a Navigation page or pagelet.

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

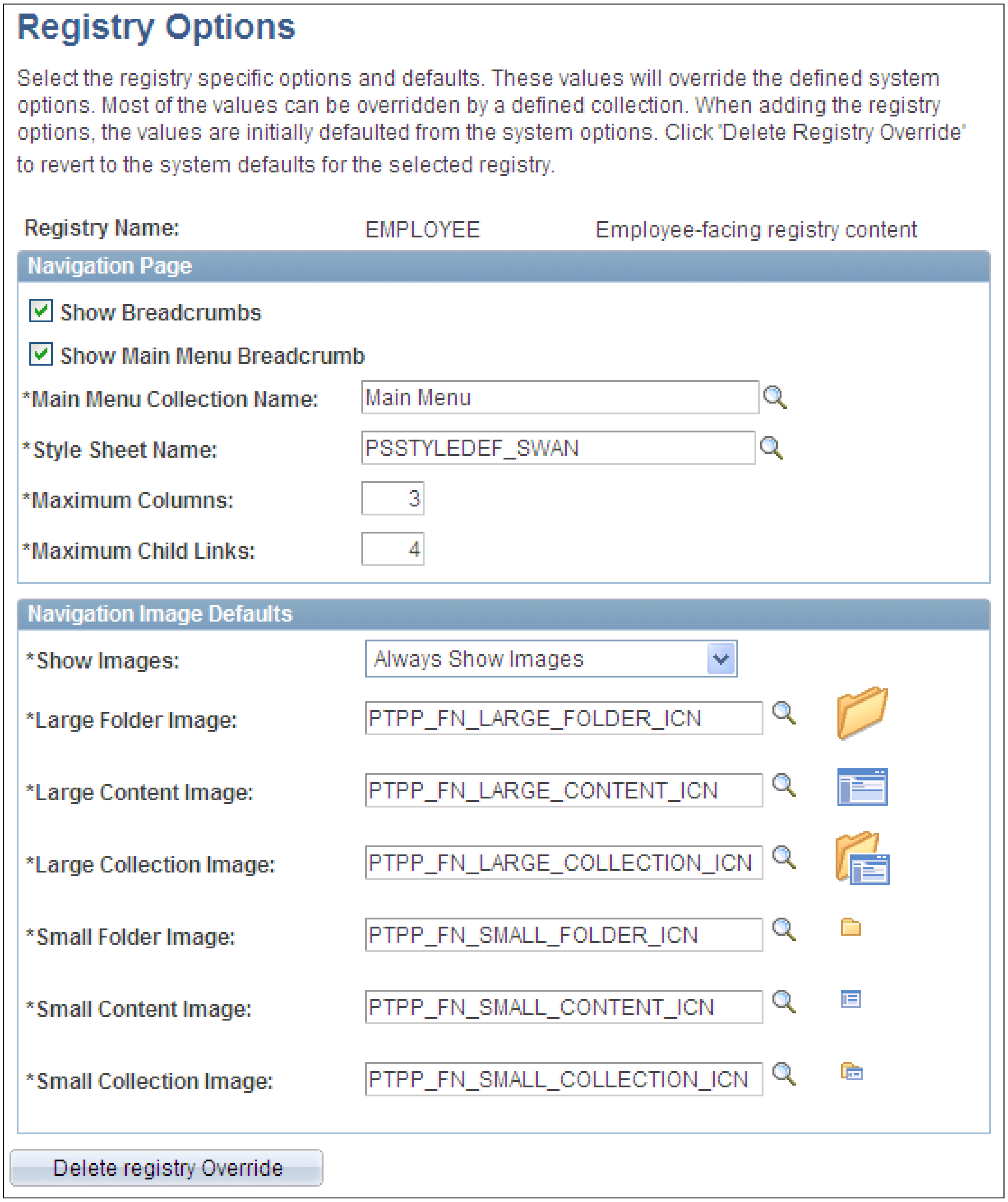
Pagelet Wizard

Field or Control	*Description*
Package Name	Specify the application package name to use for the display format.
Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method.
Application Class ID	Enter the application class ID to use for the display format. The class must exist in the application package name that you specify.

Defining Navigation Collection Registry Options

Access the Registry Options page. (Select **PeopleTools** > **Portal** > **Portal Utilities** > **Nav Collection Registry Options**.)

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



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

See Understanding Navigation Collection Options.

Field or Control	*Description*
Registry Name	Select the portal registry for which you want to define navigation collection options that will override system-level defaults.

Navigation Image Defaults

Field or Control	*Description*
Show Images	*Always Show Images:* Select if you always want images to appear on navigation pages and pagelets accessed by means of the selected portal registry. *Never Show Images:* Select if you never want images to appear on navigation pages and pagelets accessed by means of the selected portal registry. *Only Show Defined Images:* Select if you want only specifically defined images to appear on navigation pages and pagelets accessed by means of the selected portal registry. If you select this option, only images defined as a part of the navigation collection or defined on the source reference are displayed on the navigation pages and pagelets. System-level and registry-level default images are not displayed.
Large Folder Image	Select the image that you want to display for top-level folders on a Navigation page or pagelet accessed by means of the selected portal registry.
Large Content Image	Select the image that you want to display for top-level content references on a Navigation page or pagelet accessed by means of the selected portal registry.
Large Collection Image	Select the image that you want to display for top-level navigation collection references on a Navigation page or pagelet accessed by means of the selected portal registry.
Small Folder Image	Select the image that you want to display for a folder within a folder on a Navigation page or pagelet accessed by means of the selected portal registry.
Small Content Image	Select the image that you want to display for a content reference within a folder on a Navigation page or pagelet accessed by means of the selected portal registry.

Field or Control	***Description***
Small Collection Image	Select the image that you want to display for a navigation collection within a folder on a Navigation page or pagelet accessed by means of the selected portal registry.
Delete Registry Override	Click to revert the portal registry's navigation collection options to the system-level defaults defined on the System Options page.

Creating and Maintaining Navigation Collections

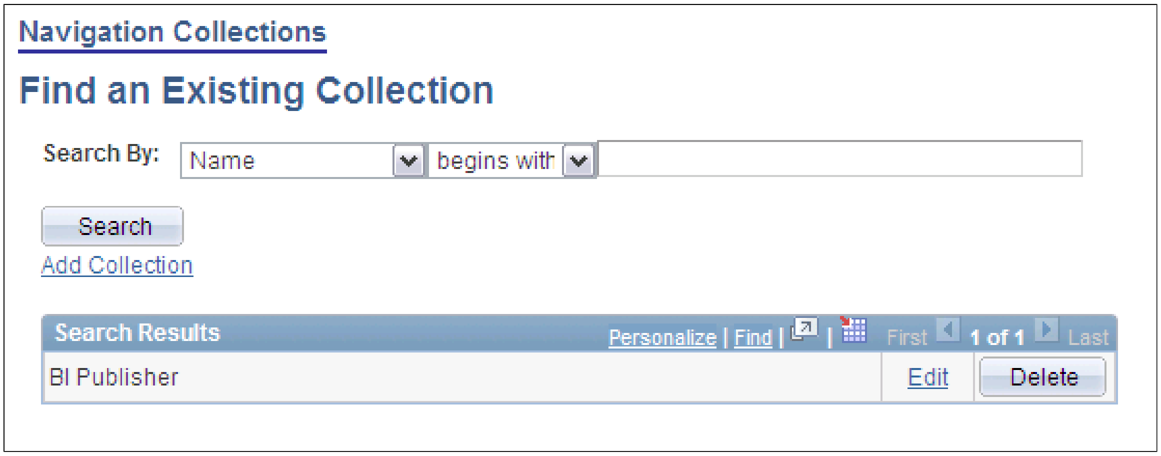
This section discusses how to:

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

Finding Existing Navigation Collections

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

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



Field or Control	***Description***
Search by	Select one of these values: *Description:* Select to conduct your search based on navigation collection description text. *Name:* Select to conduct your search based on navigation collection name text. Use the *begins with* and *contains* options to limit your search results.
Add Collection	Select to access the Maintain Collection page, where you can define a new navigation collection.

Search Results

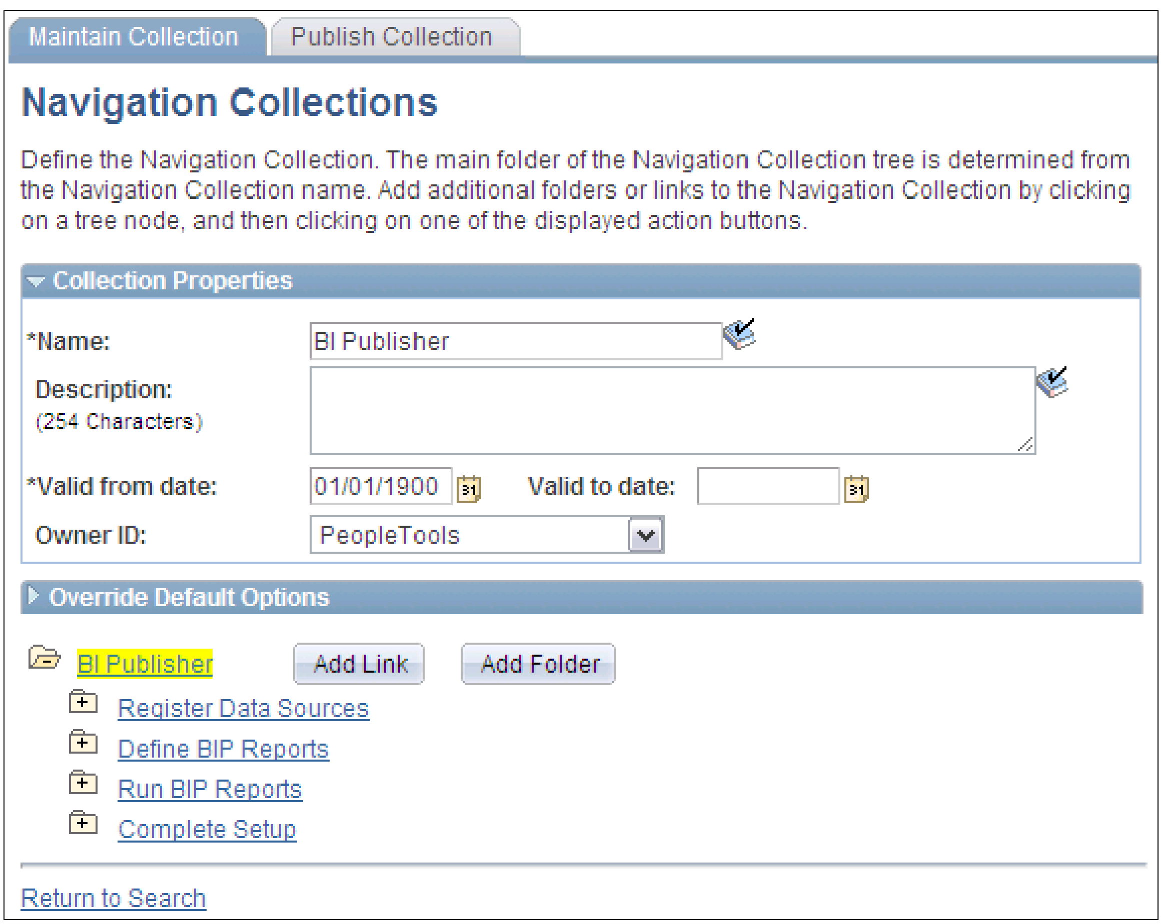
Field or Control	***Description***
Edit	Select to access the Maintain Collection page, where you can edit an existing navigation collection.

Field or Control	*Description*
Delete	Click to delete the existing navigation collection. You are prompted with the Navigation Collections - Delete Confirmation page, where you can confirm or cancel the deletion of the navigation collection. **Note:** The **Delete** button does not appear for the Main Menu navigation collection. While you can edit this navigation collection, you should not delete it because the Main Menu navigation collection is used by the system as a default page to display when it encounters certain errors. Displaying the Main Menu rather than an error page enables you to continue to navigate through your system.

Creating or Editing a Navigation Collection

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

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



Note: If you modify a navigation collection that has already been published, you do not need to republish the collection using the Publish Collection page. Saving your edits on the Maintain Collection page is sufficient because navigation collection publications are dynamically generated for display based on their saved navigation collection definitions.

Collection Properties

Field or Control	***Description***
Name	Enter a name for the navigation collection. This text appears as the label on the published navigation page or pagelet.
Description	Enter a description of the navigation collection. This text can appear in several locations. If the navigation collection is published as a navigation page, the description text appears when the mouse cursor is moved over the menu item link that is used to access the page. The description text also appears on the navigation page. If the navigation collection is published as a navigation pagelet, the description text appears on the navigation pagelet.
Valid from date/Valid to date	Enter a range of dates during which this navigation collection is available. This information is copied to the portal registry when you register the navigation collection using the options on the Publish Collection page. Oracle recommends that you leave the **Valid to date** field clear if you anticipate that the navigation collection should remain active indefinitely.
Owner ID	Use this value to identify navigation collections created by your organization when you are searching for navigation collections through PeopleSoft Application Designer or SQL queries on the portal registry table. Do not select a delivered PeopleSoft owner ID. You can define unique owner IDs for your organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer . If a default value has been set at the system or registry level, it appears, but can be overridden. See Setting Up Navigation Collection Options.

Override Default Options

The **Override Default Options** group box enables you to enter values that override any navigation collection defaults defined at the system, registry, and source reference levels and which apply to only the navigation collection that you are creating. Leave fields clear if you don't want to override any default values.

See Setting Up Navigation Collection Options.

Field or Control	***Description***
Style Sheet Name	Select the style sheet for this navigation collection.
Maximum Child Links	Enter the maximum number of child links to display. **Note:** If more links are available for this collection, a **More** link appears.
Show Images	Select one of these values: *Always Show Images:* Select to always display images on pages and pagelets for this navigation collection. *Never Show Images:* Select if you don't want images to appear on pages and pagelets for this navigation collection. *Only Show Defined Images:* Select to display only specifically defined images on pages and pagelets for this navigation collection. If you select this option, only images defined as a part of this navigation collection or the source reference are displayed on the pages and pagelets for this navigation collection. System-level and registry-level default images are not displayed. **Note:** If you select the *Only Show Defined Images* option, small images do not appear on your navigation collection pages or pagelets because small images can be defined only as system-level and registry-level defaults.
Collection Image	Select the large image that you want to display as the top-level image for this navigation collection.

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

The **Name** value that you entered for the navigation collection appears as a root node link at the top of the hierarchy. Each node below the navigation collection name is either a folder or content reference link that you defined using the following options and their associated pages.

A navigation collection may be composed of a series of individual content reference links, a series of folders containing content reference links, or a combination of individual content reference links and folders.

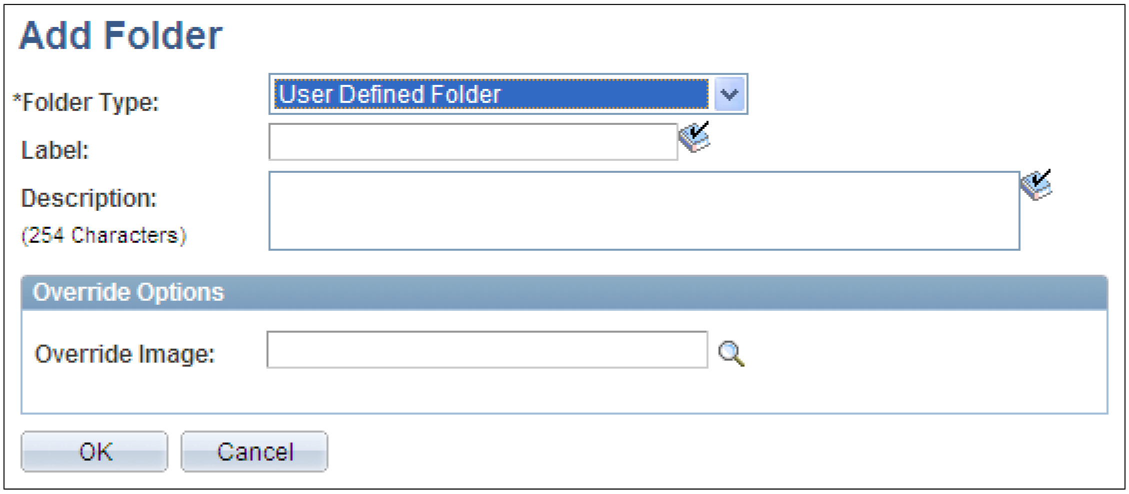
Field or Control	***Description***
Add Link	Click to access the Add Link page. This option appears when you select the navigation collection root node folder or user-defined folders in the navigation collection hierarchy. If you use this option at the root-node level of the hierarchy, the content reference link that you create appears directly off the root node, at the same level in the navigation collection structure as a folder. See Understanding Navigation Collections.
Edit Link	Click to access the Edit Link page. This option appears when you select a content reference link in the navigation collection hierarchy.
Delete Link	Click to access the navigation collections - Delete Confirmation page, where you are prompted to confirm or cancel the deletion of the selected content reference link. This option appears when you select a content reference link in the navigation collection hierarchy.
Add Folder	Click to access the Add Folder page. This option appears when you select the navigation collection root node folder or user-defined folders in the navigation collection hierarchy. If you use this option at the root-node level of the hierarchy, the folder that you create appears directly off the root node. Using this option while in any folder in the hierarchy creates a folder within the selected folder.
Edit Folder	Click to access the Edit Folder page. This option appears when you select any folder other than the root node folder.
Delete Folder	Click to access the Navigation Collections - Delete Confirmation page, where you are prompted to confirm or cancel the deletion of the selected folder. This option appears when you select any folder other than the root note folder in the navigation collection hierarchy. This deletes the folder only from the navigation collection.

Field or Control	***Description***
	Designates a folder for which content is user-defined. These folders are added on the Add Folder page by means of the *User Defined Folder* folder type. Click to expand the node and access options that enable you to edit or delete the folder. Alternatively, select the associated folder name link to display options for editing or deleting the folder, but not expand folder contents.
	Designates a folder for which content is menu-based. These folders are added on the Add Folder page by means of the *Menu Folder* folder type. You cannot view folder contents because a menu-based folder automatically inherits the content and folder references registered in the selected menu folder. This content cannot be edited from within navigation collections. Select the associated folder name link to access options for editing aspects of the folder or delete the folder from the navigation collection.
	Designates a content reference link in the hierarchy. Click to access options that enable you to edit or delete the content reference from the navigation collection.

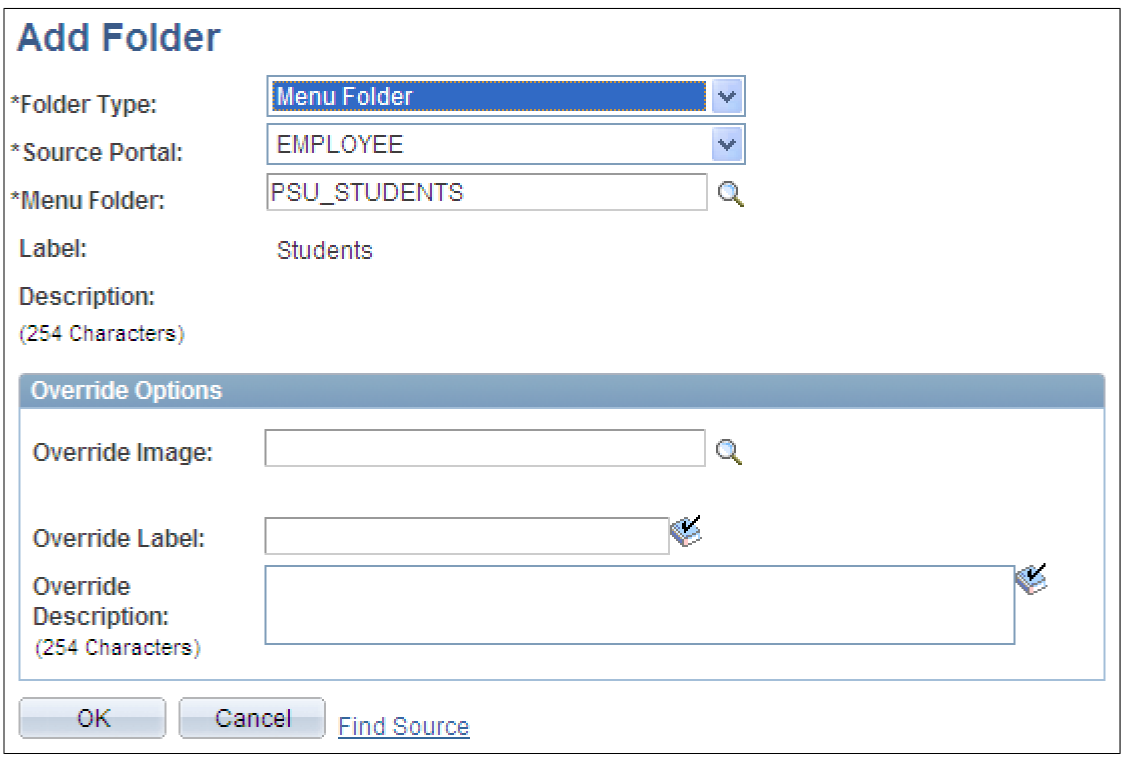
Adding a Folder

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

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



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



Field or Control	***Description***
Folder Type	Select one of these values: *Menu Folder:* Creates a reference to an existing folder in the portal registry. The reference automatically inherits all of the content and folder references registered in the selected menu folder. *User Defined Folder:* Creates a user-defined folder, enabling you to specify the folder label, description, and valid from date and to date for the folder.
Source Portal	This field appears when the **Folder Type** value is set to *Menu Folder.* Select the portal registry from which you want to select a folder. The current portal is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases in which a navigation collection will be copied between portals, and the folder reference that you are defining should automatically refer to the current portal. See Understanding Portal Administration.
Source Folder	This field appears when the **Folder Type** value is set to *Menu Folder.* Select a folder to reference. Click the **Browse Portal Registry** button to access the Select Source Folder page, where you can view a graphical display of the selected portal registry and select the folder that you want to reference.
Label	If you are adding a menu-based folder, this field is populated with the label on the source, which can be overridden in the **Override Label** field. This text appears as link text for the folder on the Maintain Collection page, as well as on the navigation collection page or pagelet.
Description	If you are adding a menu-based folder, this field is populated with the description on the source, which can be overridden in the **Override Description** field. This text appears when the mouse cursor is moved over a folder link and as descriptive text beneath the folder link on the navigation collection page or pagelet.

Override Options

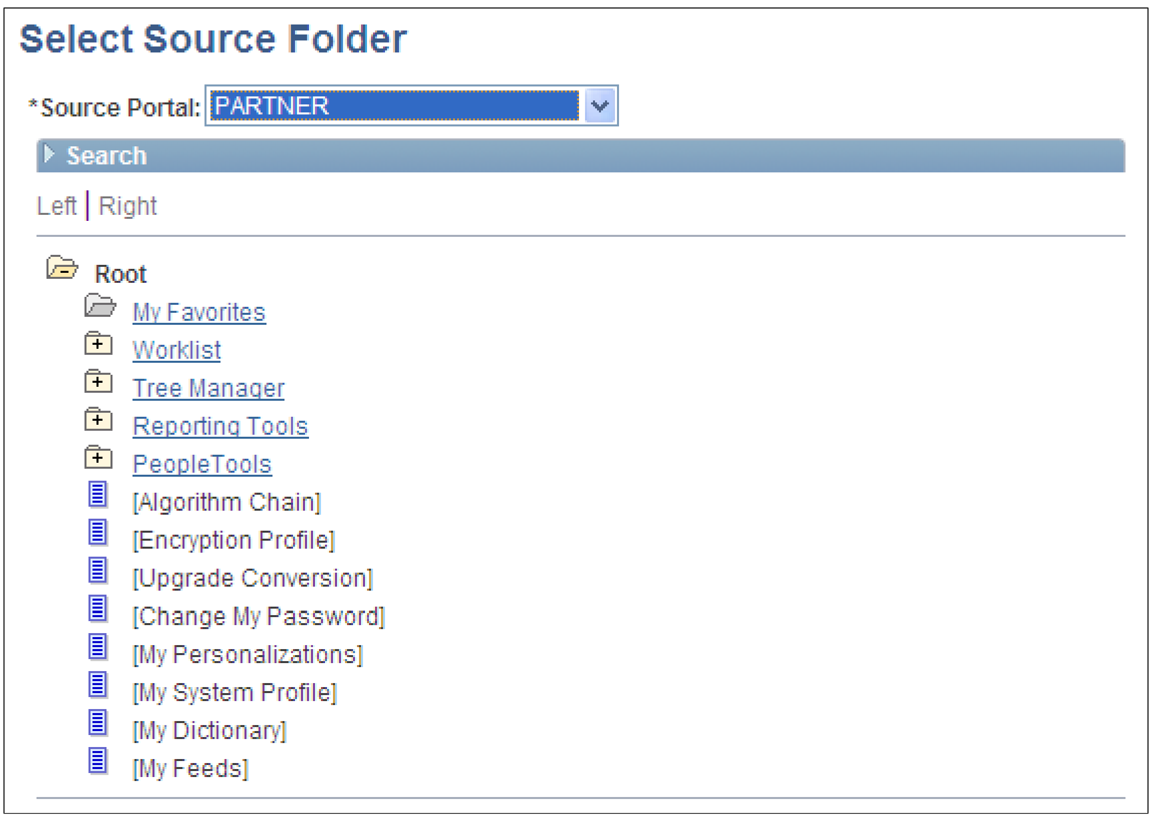
Field or Control	*Description*
Override Image	Select the image that you want to display as the icon for the folder reference. If a default folder image has been defined at the system or registry level, this image overrides the default. This image appears as the large image for the folder on the navigation collection publication. When defined for folders nested within folders, the image appears on the navigation collection page that is accessed to display the contents of the parent folder. If you override the image, changes to the image on the source reference are not reflected on the navigation collection publication. If you want source reference changes to be reflected on the navigation collection publication, do not override this image. See Setting Up Navigation Collection Options, Navigation Collection Images.
Override Label	If you are adding a menu folder, you can enter a label to override label text in the **Label** field. If you override label text, changes to label text on the source reference are not reflected on the navigation collection publication. If you want source reference changes to be reflected on the navigation collection publication, do not enter override text in this field.
Override Description	If you are adding a menu-based folder, you can enter a description to override description text in the **Description** field. If you override description text, changes to description text on the source reference are not reflected on the navigation collection publication. If you want source reference changes to be reflected on the navigation collection publication, do not enter override text in this field.

Field or Control	***Description***
Find Source	This link appears for users who can access the Structure and Content pages when the **Folder Type** value is set to *Menu Folder.* Select to access the Folder Administration page, where you can view details about the selected menu folder. You may want to access this page to add a source reference image, change a source reference label, or check source reference security. See Administering Folders.

Selecting a Source Folder

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

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



Field or Control	*Description*
Source Portal	Select the portal registry from which you want to select a folder to reference. The portal designated on your originating page is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases in which a navigation collection will be moved between portals, and the folder reference that you are defining should automatically refer to the current portal. See Administering Portal Definitions.

Search

Field or Control	*Description*
Search by	Select one of these values: *Description:* Select to conduct your search based on folder description text. This search is not case-sensitive. You can perform partial match searches. For example, doing a label search on *source* may not return any *source* results, but may return results with *resource.* *Label:* Select to conduct your search based on folder label text. Alternatively, search by drilling down into the tree structure displayed at the bottom of the page.

Search Results

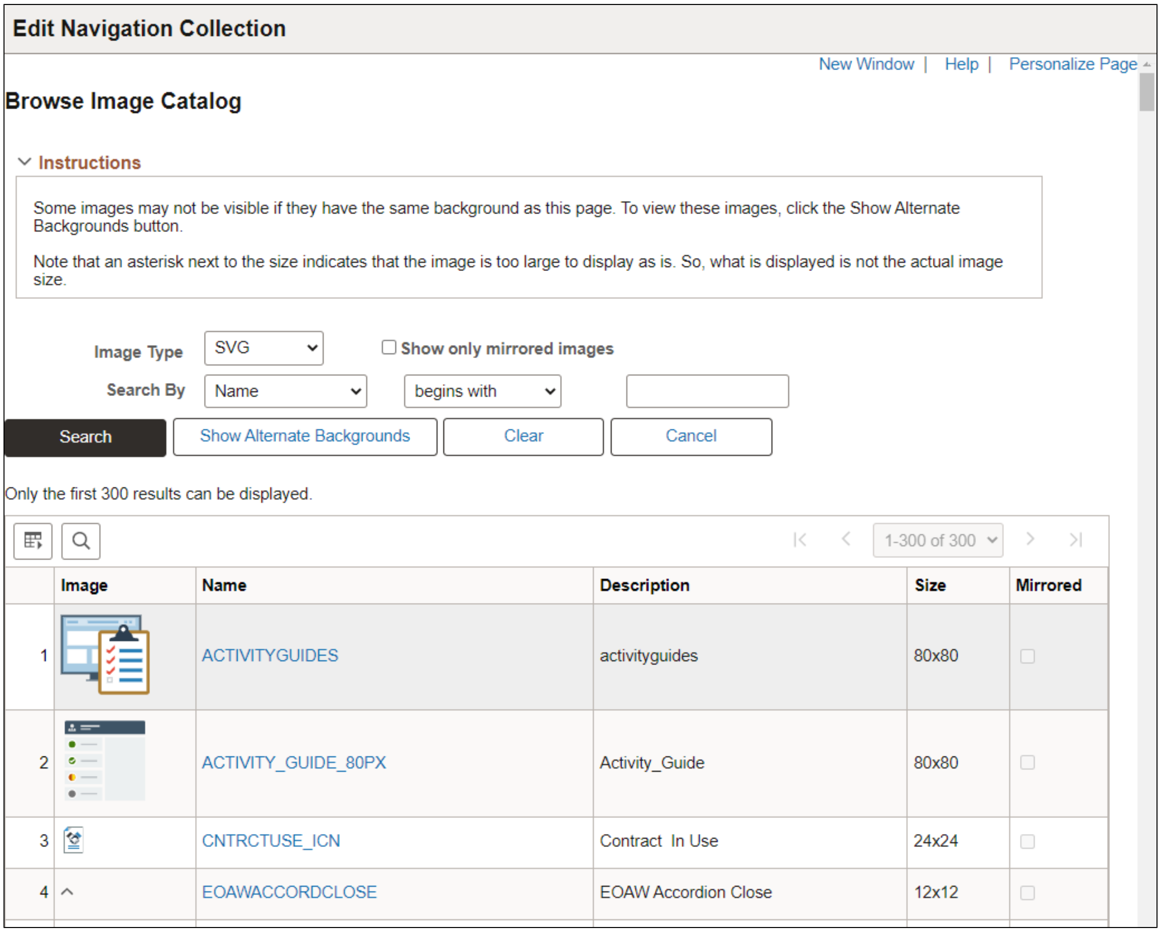
Field or Control	*Description*
Label	Select the linked label text to locate and select the referenced folder in the portal registry hierarchy displayed at the bottom of the page.
Description	Displays the text description of the folder.

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

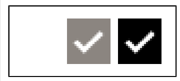
Selecting Navigation Collection Images

Access the Browse Image Catalog page. (Click the **Browse Image Catalog** button on the Add Folder, Edit Folder, Add Link, or Edit Link pages.)

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



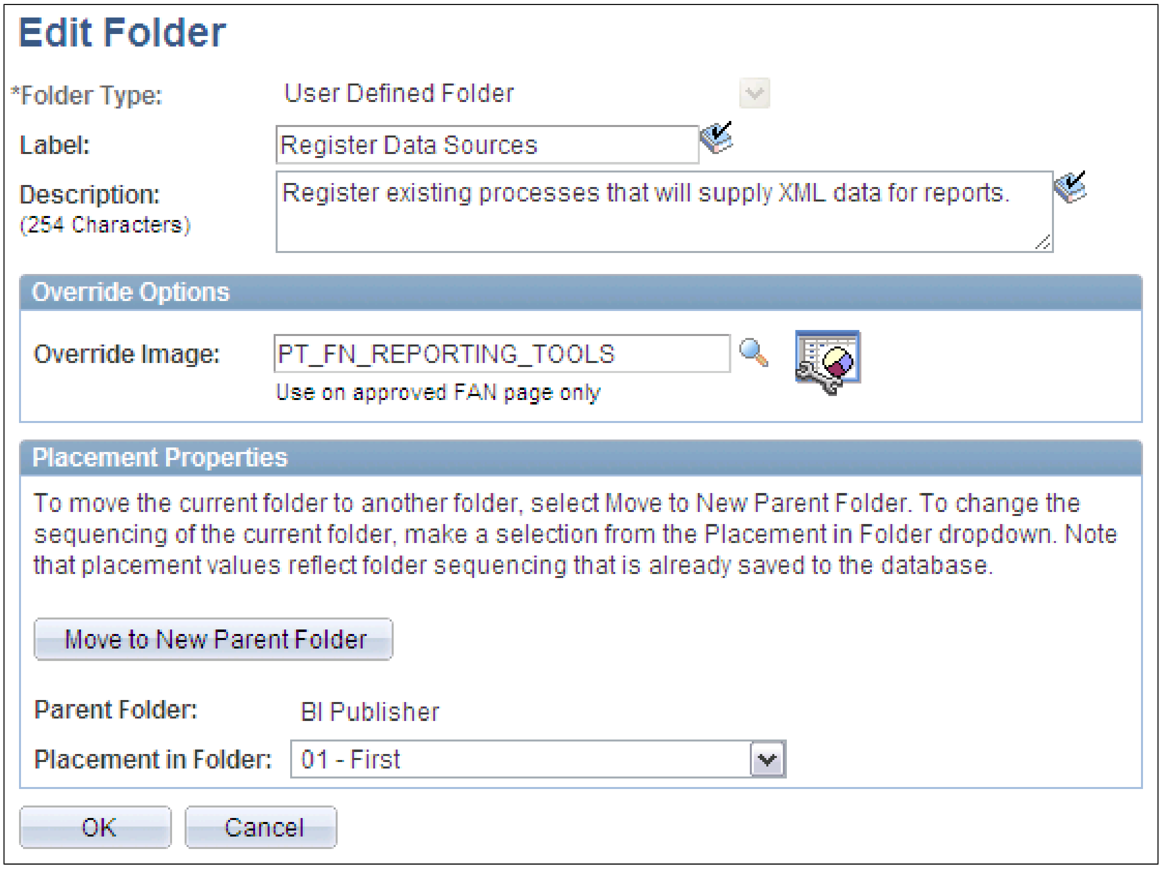
Field or Control	***Description***
Image Type	Select from these image types: • *Bitmap* • *DIB* • *GIF* • *JPEG* • *PNG* • *SVG* – The default value.
Show only mirrored images	Select *Yes* to only retrieve images that are mirrored. Otherwise, both mirrored and unmirrored images are retrieved.
Search By	Select *Name* or *Description* and additional search criteria to limit the search results.

Field or Control	***Description***
Search	Click the Search button to retrieve the images meeting the search criteria from the database.
Show Alternate Backgrounds	Click the Show Alternate Backgrounds button to redisplay each image in the current grid on three background colors: white, gray, and black. This option allows you to view white images, which are not visible on the default white background. In this example, the white check mark is visible on the gray and black backgrounds: 
Clear	Click the Clear button to clear both the search criteria and the grid.
Cancel	Click the Cancel button to close the secondary window.
Name	Select an image by name to use it as the custom static image.

Editing a Folder

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

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



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

See Adding a Folder.

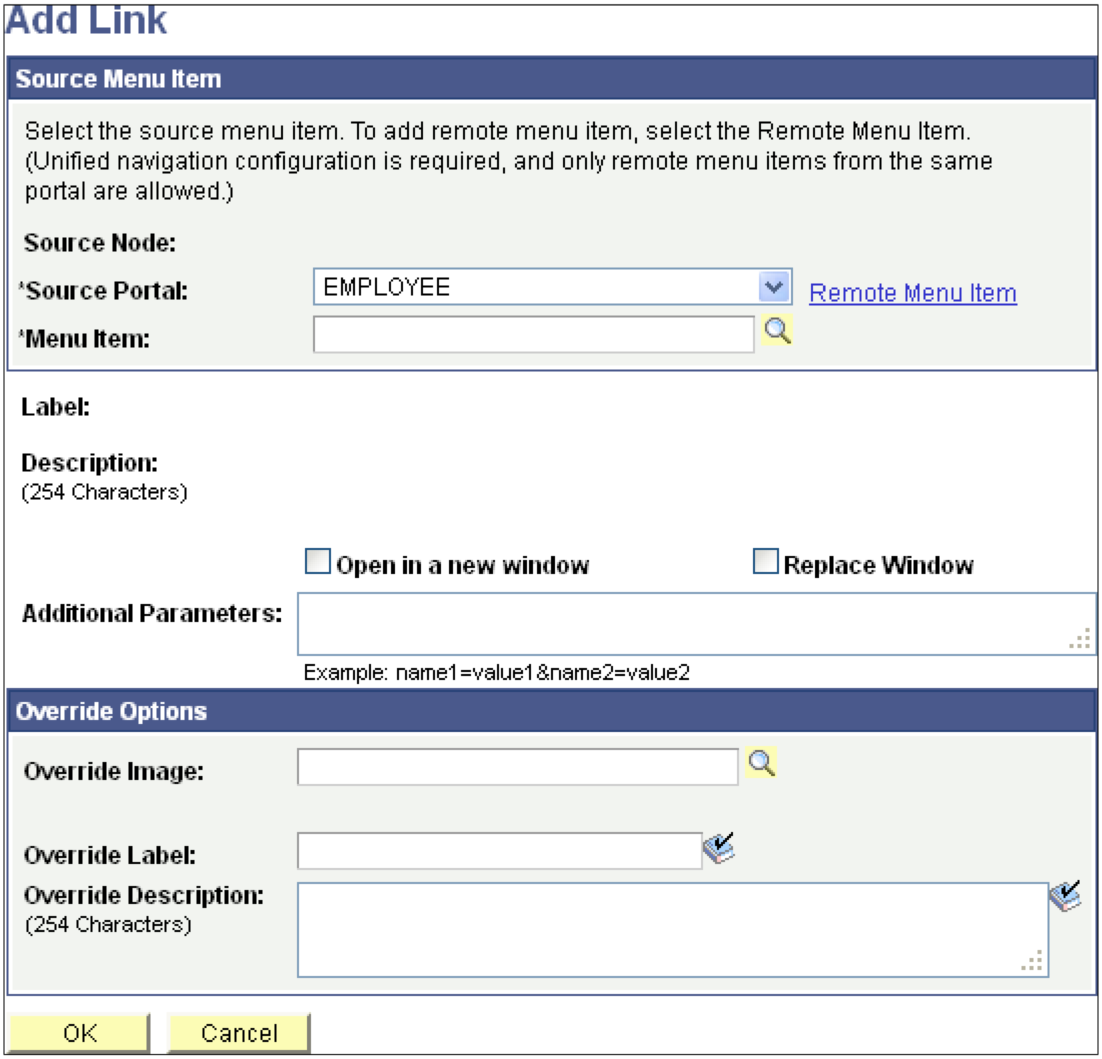
Placement Properties

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

Adding a Link

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

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



Field or Control	***Description***
Source Node	Displays the name of the remote source node if a remote menu item has been selected.

Field or Control	***Description***
Source Portal	Select the portal registry from which you want to select a content reference. The current portal is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases in which a navigation collection will be copied between portals, and the content reference that you are defining should automatically refer to the current portal. See Administering Portal Definitions. **Note:** For remote menu items, the **Source Portal** defaults to the current portal and is read only.
Menu Item	Select a local content reference to add this link to the navigation collection. Click the **Browse Portal Registry** button to access the Select Source Link page, where you can view a graphical display of the selected portal registry and select the content reference that you want to add. See Selecting a Local Link. **Note:** You can use this option to add a published navigation page to the navigation collection that you are creating.
Remote Menu Item	Select a remote component-based content reference to add to the navigation collection from the Select a Content Reference page. See Selecting a Remote Link.
Label	Displays the existing menu content reference label text, but you can override this default using the **Override Label** field. This text appears as link text for the content reference on the Maintain Collection page, as well as on the navigation collection publication.
Description	Displays the existing description text, but you can override the default using the **Override Description** field. This description text appears when the mouse cursor is moved over the content reference link.
Open in a new window	Select to open the page in a new browser window. This option and **Replace Window** are mutually exclusive.
Replace Window	Select to replace the current browser window, including navigation and target area, with the new content. **Replace window** updates the template and navigation to display the template and navigation of the target content reference. This option and **Open in new window** are mutually exclusive.

Field or Control	*Description*
Additional Parameters	Define additional query parameters that are appended to the uniform resource locator (URL) used to access the content reference. Defining these query parameters here eliminates the need to register the same content reference multiple times with different parameters. For example, you can use these additional parameters to point users to a specific row of data on an application business transaction page. The query string parameter can be the search record field name = field value. These query parameters are stored as attributes with the content reference and are applied to the URL at runtime.

Override Options

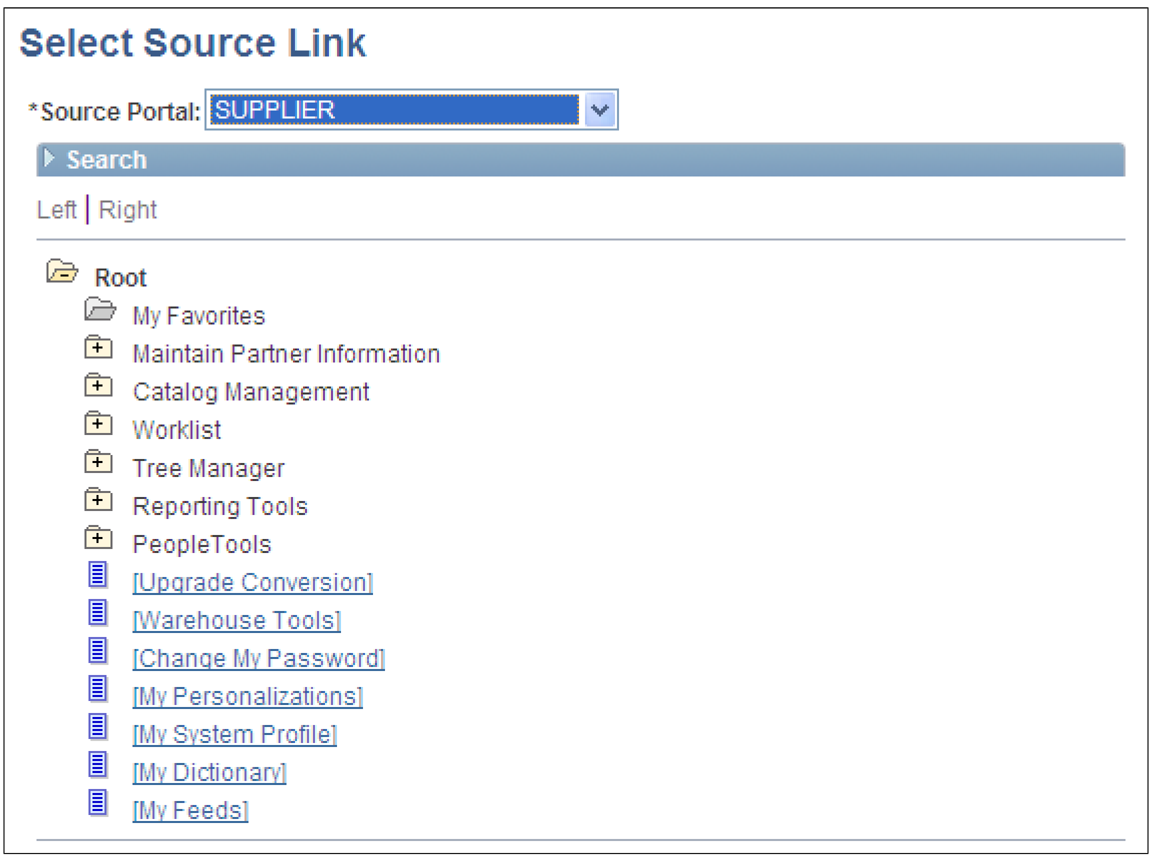
Field or Control	*Description*
Override Image	Select the image to display as the large image for the content reference. If a default content image has been defined at the system or registry level, the selected image overrides the default. This large image appears for the content reference on the navigation collection publication. See Understanding Navigation Collection Options.
Override Label	Enter a label to override label text in the **Label** field. If you override label text, changes to label text on the source reference are not reflected on the navigation collection publication. If you want source reference changes to be reflected on the navigation collection publication, do not enter override text in this field.
Override Description	Enter a description to override description text in the **Description** field. If you override description text, changes to description text on the source reference are not reflected on the navigation collection publication. If you want source reference changes to be reflected on the navigation collection publication, do not enter override text in this field.

Field or Control	***Description***
Find Source	This link appears for users who can access the Structure and Content pages once a **Source Link** value has been selected. Select to access the Content Ref Administration page, where you can view content reference details, such as security, image attributes, valid to date and from date, and content provider for the selected source content reference. See Administering Content References.

Selecting a Local Link

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

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



Field or Control	***Description***
Source Portal	Select the portal registry from which you want to select a content reference. The portal designated on your originating page is selected by default. Available source portal values include all portals defined in the PeopleSoft databases, as well as a special *Local Portal* value. The *Local Portal* option is useful for cases in which a navigation collection will be moved between portals, and the content reference that you are defining should automatically refer to the current portal. See Administering Portal Definitions.

Search

Field or Control	***Description***
Search by	Select one of these values: *Description:* Select to conduct your search based on content reference description text. This search is not case-sensitive. You can perform partial match searches. For example, doing a label search on *source* may not return any *source* results, but may return results with *resource.* *Label:* Select to conduct your search based on content reference label text. Alternatively, search by drilling down into the tree structure displayed at the bottom of the page.

Search Results

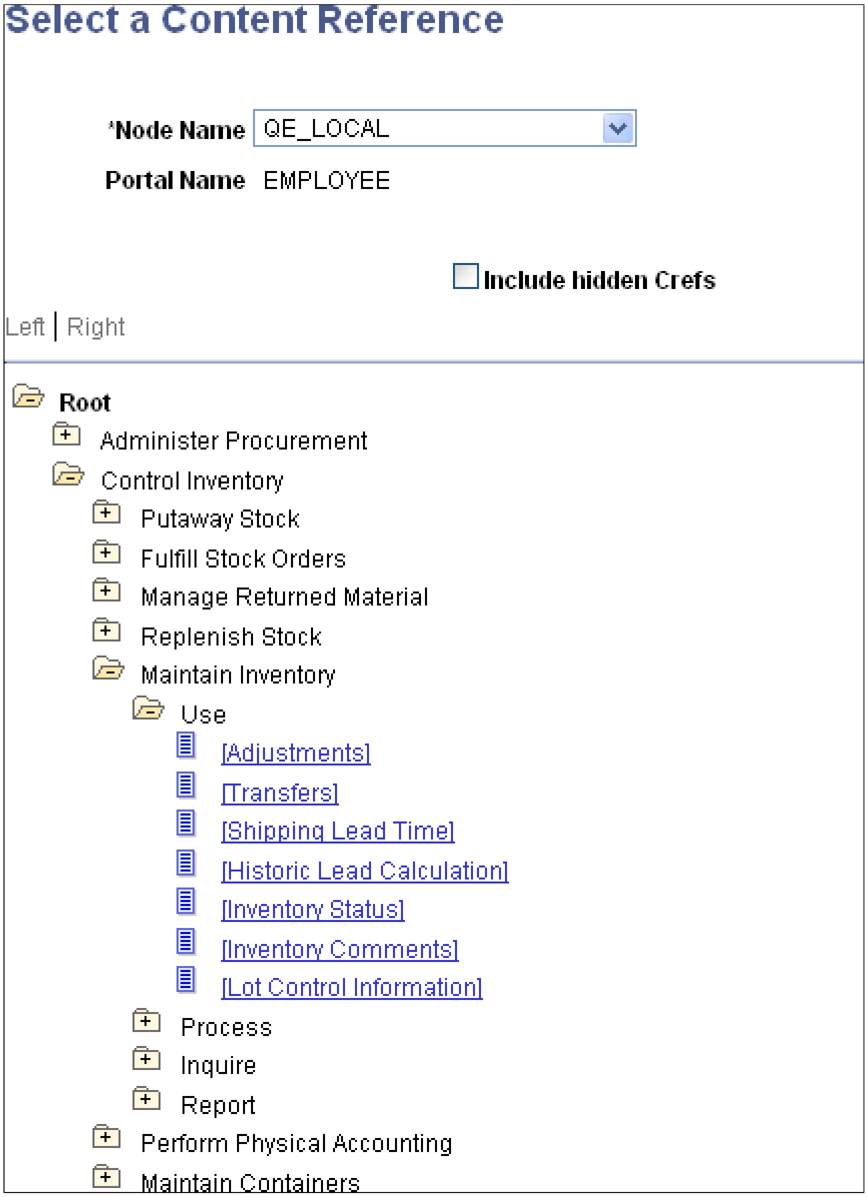
Field or Control	***Description***
Label	Select the linked label text to locate and select the content reference in the portal registry hierarchy displayed at the bottom of the page.
Description	Displays the text description of the content reference.

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

Selecting a Remote Link

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

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



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

To select a remote link:

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

Related Links

Understanding Unified Navigation

Editing a Link

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

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

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

See Adding a Link.

Placement Properties

Field or Control	***Description***
Move to New Parent Folder	Click to move the content reference that you are editing to a folder other than the folder designated in the **Parent Folder** field.
Parent Folder	Displays the parent folder of the content reference that you are editing.
Placement in Folder	Select a value to change the placement of the content reference within its selected navigation collection folder. Available placement values reflect saved content reference sequencing.

Publishing Navigation Collections

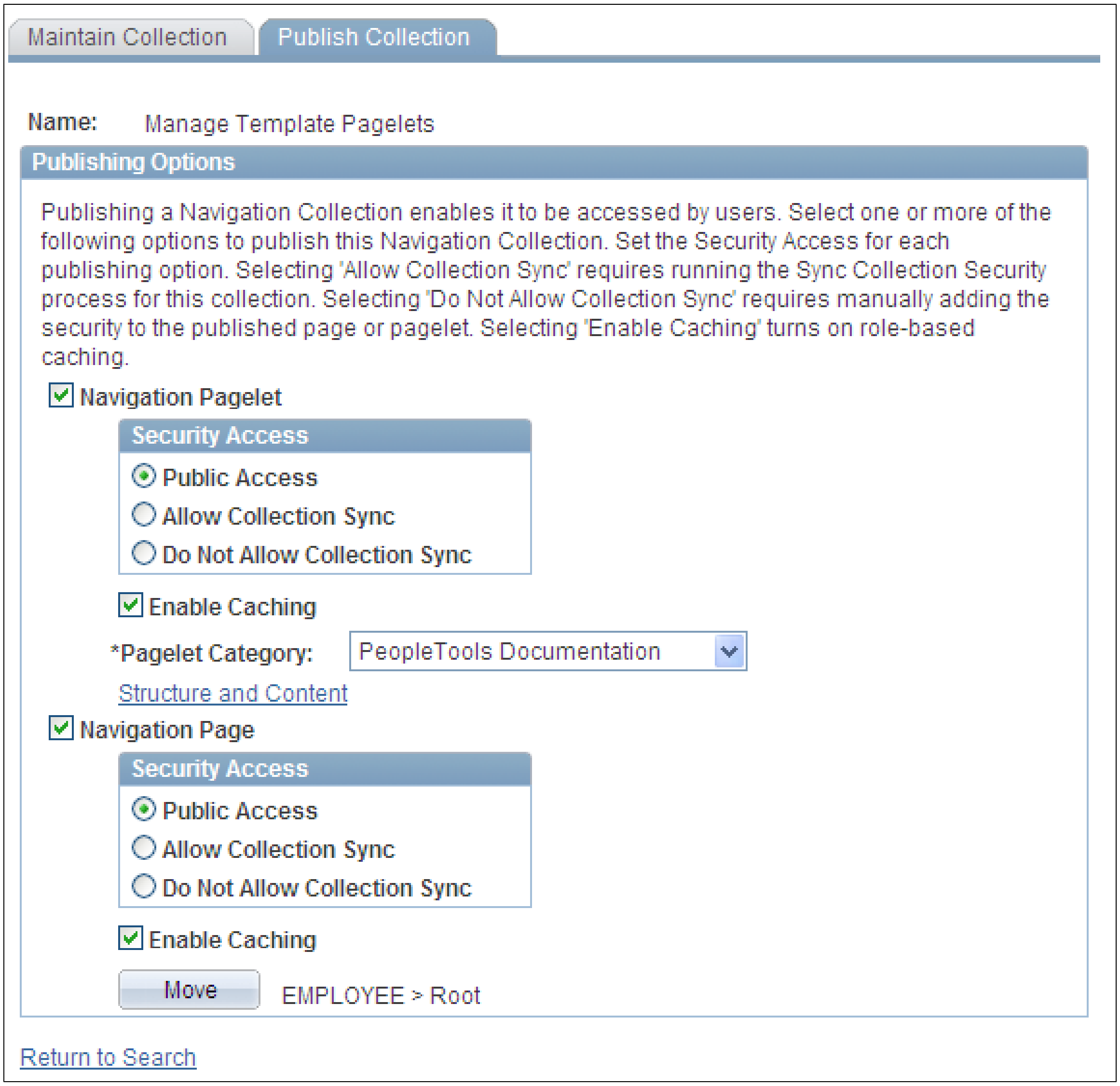
Important! Standard navigation pages (DFANs), custom navigation pages (CFANs), SmartNavigation, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see “Working with Fluid Pages and Controls” (Applications User’s Guide). However, navigation collections published as pagelets and used on classic WorkCenters, but not on classic homepages, remain a supported navigation mechanism.

This section discusses how to publish navigation collections.

Publishing a Navigation Collection

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

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



Field or Control	***Description***
Name	Displays the name of the navigation collection defined on the navigation collection page.

Publishing Options

Field or Control	***Description***
Navigation Pagelet	Select to publish the navigation collection as a navigation pagelet, which can be added to a user's homepage tab.

Field or Control	*Description*
Security Access	Select one of these values: *Public Access:* Select to give all users access to the pagelet. This option is selected by default. *Allow Collection Sync* (allow collection synchronization): Select to make the pagelet eligible for processing by the Sync Collection Security process. *Do Not Allow Collection Sync* (do not allow collection synchronization): Select if you do not want the pagelet security to be updated by the Sync Collection Security process. Select this option to ensure that security that you manually define for the pagelet cannot be overridden by the process. See Understanding Portal Utility Processes.
Enable Caching	Select to have role-based caching enabled for this published pagelet. Selecting this option adds portal caching attributes to the published navigation pagelet. See Understanding Portal Caching.
Pagelet Category	Select the category in which you want the navigation pagelet to appear for selection on the Personalize Content page. Available pagelet categories are defined in the Structure and Content component. Select the Portal Objects folder link, then select the Pagelets folder link. Select the Add Folder link to add a pagelet category folder. See Managing Portal Objects.
Structure and Content	Select to access the Content Ref Administration page, where you can access details about the content reference created for the navigation pagelet publication. For example, you can check the results of the Sync Collection Security (synchronize collection security) Application Engine process (PTPP_ SCSYNC) and change content reference label text or manually update the security. See Understanding Portal Utility Processes, Administering Content References.
Navigation Page	Select to publish the navigation collection as a navigation page, which can be accessed from the menu.

Field or Control	***Description***
Security Access	Select one of these values: *Public Access:* Select to give all users access to the page. This option is selected by default. *Allow Collection Sync* (allow collection synchronization:. Select to make the page eligible for processing by the Sync Collection Security process. *Do Not Allow Collection Sync* (do not allow collection synchronization): Select if you do not want the page security updated by the Sync Collection Security process. Select this option to ensure that any security that you manually define for the page is not overridden by the process. See Understanding Portal Utility Processes.
Enable Caching	Select to have role-based caching enabled for this published page. Selecting this option adds portal caching attributes to the published navigation page. See Understanding Portal Caching.
Move	Click to access the Select Source Folder page, where you can select the parent folder in which you want the navigation page to appear in the left navigation menu. See Selecting a Source Folder.
Structure and Content	Select to access the Content Ref Administration page, where you can access details about the content reference created for the navigation page publication. For example, you can check the results of the Sync Collection Security process and change navigation page label text. See Understanding Portal Utility Processes, Administering Content References.

Publishing Navigation Collections as Pagelet Wizard Pagelets

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

- At step 3, designating the pagelet as configurable (administrators), personalizable (users), or both.

- At step 4, indicating that you intend to use a custom XSL template for transforming the output format for the pagelet.
- At step 5:
  - Specifying a custom XSL template.
  - Displaying or hiding the top level title, description, or both.
  - Displaying or hiding folder descriptions.
  - Specifying the image type for folders.
  - Specifying the list type for links.
  - Adding pagelet headers and footers.
- At step 6:
  - Indicating whether the pagelet will be published as a homepage pagelet, template pagelet, embedded pagelet, or marked for WSRP production.
  - Displaying or hiding the pagelet title bar and borders.

Related Links

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

Chapter 11

Administering Unified Navigation

Understanding Unified Navigation

This section provides an overview of:

- Unified navigation.
- The Configure Unified Navigation navigation collection.
- Definition of terms.
- Considerations for remote content.
- Limitations and additional considerations.

About Unified Navigation

Important! Any PeopleSoft system can serve as the designated portal system in a cluster; PeopleSoft Interaction Hub is not required for unified navigation.

Unified navigation provides a framework to federate multiple PeopleSoft applications as a cluster under a designated, central portal system. Using the delivered Configure Unified Navigation navigation collection, you can configure and federate portal registries from content provider systems into a unified system. Your administrators and users will log into the central portal system to access local resources on the portal system as well as remote resources made available through unified navigation. Once unified navigation is configured, your users and administrators will be able to:

- Access remote content references and resources, which includes:
  - Accessing remote navigation pages through the main menu of the portal system.
  - Adding remote content references as Favorites on the portal system.
  - Accessing remote dashboards through the main menu of the portal system.
  - Accessing remote WorkCenter pages with remote pagelets through the main menu of the portal system.
  - Creating related content service definitions in the portal system using either local or remote content references.
  - Creating related content services in the content provider systems using content references local to that system.
  - Assigning local (to the portal system) or remote content references (from a content provider system) as related content services (or related actions) to content references on the portal system.

  - Assigning local (to the content provider system) or remote content references (from a different content provider system) as related content services (or related actions) to content references on the content provider system.

- Synchronize user system profiles, or user personalization data, or both from the portal system to the content provider systems.

In this example:

- The URL in the address field indicates that the portal system is the current system and that the content resides on a content provider node (HRMS).
- The page header (Home and Sign out links, search, and so on), the main menu, and homepage tabs are generated by the portal system. This ensures that portal functions and navigation remain local to the portal system.
- The content in the target area of the page comes from the content provider system. (If the content reference in the content provider system is defined with a specific template, then the content is also rendered using that template.)
- Related content services defined for the remote content reference are displayed. In this example, the related content has been defined as related actions.

Related Links

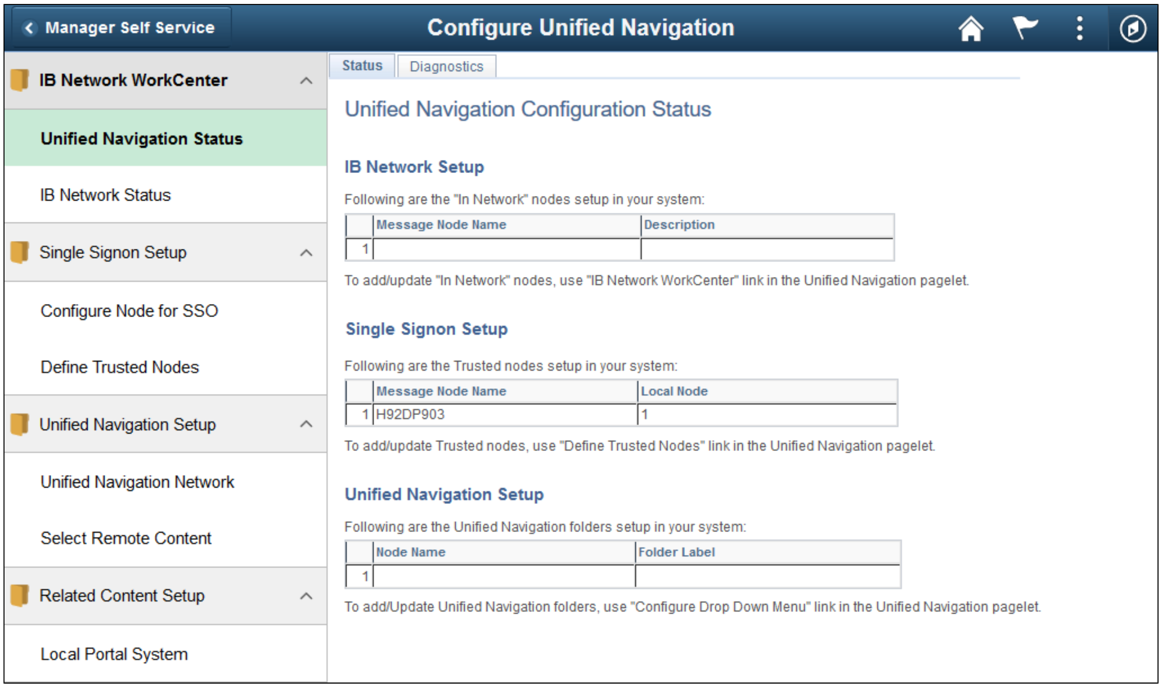
Considerations for Remote Content

Configure Unified Navigation Fluid Navigation Collection

The Configure Unified Navigation navigation collection provides links to the steps that need to be completed to configure the systems in your cluster for unified navigation.

To access the Configure Unified Navigation navigation collection, select **PeopleTools** > **Portal** > **Configure Unified Navigation**. Alternatively, add the Configure Unified Navigation tile to your homepage.

This example illustrates the Configure Unified Navigation navigation collection prior to completing configuration of unified navigation.



When you click links in the Configure Unified Navigation navigation collection, transaction pages are displayed from the system on which you are configuring unified navigation. The Configure Unified Navigation navigation collection presents the tasks that you need to complete. The following procedure rearranges the tasks into a more logical order than is presented in the Configure Unified Navigation navigation collection.

Configuration of unified navigation consists of seven tasks, which are documented in these topics:

1. Complete prerequisite configuration steps (all systems).

   See Completing Initial System Configuration.

2. Configure node definitions for all participating nodes (all systems).

   See Defining All Participating Nodes.

3. Configure a shared integration gateway:

   a. Create the shared gateway definition (portal system only).

      See Creating the Shared Gateway Definition.

   b. Point to the shared gateway (content provider systems).

      See Pointing to the Shared Gateway.

4. Add remote nodes to the integration network (portal system only).

   See Adding Remote Nodes to the Integration Network.

5. Activate each Integration Broker domain (all systems).

   See Creating the Shared Gateway Definition.

6. Set up and test single signon:

   a. Configure single signon (all systems).

      See Identifying Trusted Nodes.

   b. Test single signon and the unified navigation node network (portal system only).

      See Testing the Single Signon Configuration.

7. Complete unified navigation:

   a. Add remote folders to the main menu (portal system only).

      See Configuring Remote Content and Completing Configuration of Remote Content.

   b. (Optional) Import pagelets (portal system systems).

      Importing Pagelets from Content Provider Systems.

8. (Optional) configure related content to take advantage of remote content and remote services. Related content can be configured from the Configure Unified Navigation navigation collection

   You can create and assign related content for the portal system and for content provider systems.

   See Creating and Assigning Related Content Within the Unified Navigation Framework.

Definition of Terms

Unified navigation requires that two types of nodes be defined on both the local portal system and the remote content provider systems:

Term	***Definition***
Cluster	In the context of unified navigation, multiple *content provider systems* are federated by the *portal system* into a single cluster.
Content provider system	A content provider system is a PeopleSoft application, such as PeopleSoft HCM, PeopleSoft FSCM, and so on, that provides content to other systems. In the context of unified navigation, the *portal system* federates the content provider systems into a single *cluster*.

Term	***Definition***
Default local node	The default local node is the predefined *local node* that designates the system on which the database is installed. Default local nodes generate security artifacts, such as PS_TOKEN, and participate in Integration Broker integrations. For example, in the PeopleSoft FSCM database, its default local node is PSFT_EP. In the PeopleSoft HCM database, its default local node is PSFT_HR. In the context of unified navigation, each participating default local node must have an active definition in all participating systems, must be identified to the integration gateway, must be established as in-network for the integration node network (on the portal system only), and must be established as a trusted node for single signon.
Local node	A local node is any node that is local to the system on which the application content references are registered. Local nodes are designated as either the *default local node* or as a *portal host node*. For example, PSFT_HR and HRMS are defined as local nodes on a PeopleSoft HCM system. Conversely, PSFT_HR and HRMS are defined as *remote nodes* on a PeopleSoft FSCM system.
Portal host node	A portal host node is the node that hosts a specific *portal registry* and the portal's content. For example, ERP, HRMS, and CRM all host the EMPLOYEE portal on PeopleSoft FSCM, PeopleSoft HRMS, and PeopleSoft CRM systems, respectively. In the context of unified navigation, each participating portal host node must have an active definition in all participating systems and is used as the content provider node for remote folders and pagelets.
Portal registry	The portal registry is the collection of all registered content references for a specific portal. A portal registry contains folder definitions and content reference definitions in a tree-like structure. A portal registry is defined on a *portal host node*.

Term	**Definition**
Portal system	In the context of unified navigation, any PeopleSoft application can be designated as the *portal system* that federates all participating *content providers* into a single *cluster*. **Note:** In a unified navigation cluster, only one PeopleSoft application can serve as the portal system. All other systems in the cluster are referred to as content provider systems. In other contexts—for example, a standalone PeopleSoft application system—the PeopleSoft application itself serves as its own portal system.
Remote node	A remote node is any node that is remote to the system on which the configuration process or content access is being performed. For example, PSFT_HR and HRMS are defined as local nodes on a PeopleSoft HCM system. Conversely, PSFT_HR and HRMS are defined as remote nodes on a PeopleSoft FSCM system.

Considerations for Remote Content

When a menu item (content reference) from a content provider system is selected from the main menu, the information that is displayed comes from multiple sources as follows:

- The page header (Home, Sign out, and other links) plus the main menu come from the portal system.
- The target area content is fetched from the content provider system.
- The template for the target area content is also fetched from the remote system and is used to render the target content. However, if a content reference with the same name is registered in the local portal system, then the template for the local content reference is used to render the remote content reference. In addition, if the remote content reference is defined without a template, then the default template for the local portal system is used to render the remote content reference.
- When remote content is displayed, the New Window, Help, Customize Page, and other links invoke actions that are relative to the content provider system or the portal system, depending on which link is clicked:
  - The New Window link opens a new window redisplaying the current content—that is, the header and main menu come from the portal system and the content comes from the content provider system.
  - The Help link displays the help topic from the help system that is configured for the content provider system.
  - The Personalize Page link replaces the target area with the page personalization information from the content provider system.

- The Copy URL icon button copies the URL of the current page—that is, a URL on the portal system that includes the content provider node and content reference.

Note: Templates from the portal system or from the content provider system are not cached.

Remote Dashboards

Since dashboards are accessed through content references, remote dashboards defined in a content provider system can be accessed through a remote folder that is configured for unified navigation.

Note: Because homepages (unlike dashboards) are displayed only when a user directly signs into a system, homepages from content provider systems are not accessible through unified navigation.

Remote dashboards are subject to the following limitations:

- A pagelet on a remote dashboard will not function properly if the pagelet performs a JavaScript document.write or document.writeln operation.
- When the portal system's main menu is used to navigate to the remote dashboard, then the menu breadcrumbs are updated properly. However, if the PeopleCode RedirectURL method is used to load the remote dashboard, then the menu breadcrumbs are not updated.

Related Links

Managing Dashboard Pages

Limitations and Additional Considerations

Unified navigation can be used with these limitations and additional considerations in PeopleTools 8.54:

- Configuration of unified navigation *must* be completed on the portal system with associated configuration steps performed on the content provider system. Use of unified navigation features (for example, navigation to remote content references) must also occur on the portal system.
- Navigation to menu items on content provider systems is supported through the portal system's main menu only.

  Note: You cannot use homepage menu pagelets such as the Enterprise Menu pagelet or the Main Menu pagelet to navigate to remote menu items. The remote folders are hidden and will *not* appear in these pagelets.

- Pagelet icon images used in remote pagelets must be copied from the content provider system to the portal system. In addition, if both systems do not use the same style (for example, one uses TANGERINE, and the other uses a custom style), the cascading style sheets (CSS) must be copied from the content provider system to the portal system.
- Templates for remote content references must conform to the standards outlined in this document.

  See Considerations for Remote Content.

- Do not use the following special characters in the labels for a remote folder: { } #.

- Unified navigation supports same portal integration only—for example, EMPLOYEE portal to EMPLOYEE portal, CUSTOMER portal to CUSTOMER portal, or PARTNER portal to PARTNER portal, and so on.
- You should always use the content provider system's portal host node to create remote folders or import pagelets.
- Remote dashboards must conform to the standards outlined in this document.

  See Considerations for Remote Content.
- The minimum release required to implement unified navigation on a content provider system is PeopleTools 8.54.

Completing Initial System Configuration

You should complete the following steps prior to using the Configure Unified Navigation center to configure unified navigation on your systems:

1. On both the designated portal system and all content provider systems, set the authentication domain to the lowest level domain shared by all systems—for example, if all systems share the same subdomain, then .us.example.com would be appropriate; otherwise, use .example.com instead.
2. On both the designated portal system and all content provider systems, deselect the Generate Relative URLs option on the Virtual Addressing page of the web profile.
3. Stop and restart each web server after making these changes.

Related Links

Configuring General Portal Properties
Configuring Virtual Addressing

Configuring the Integration Gateway

To configure a shared integration gateway, you must:

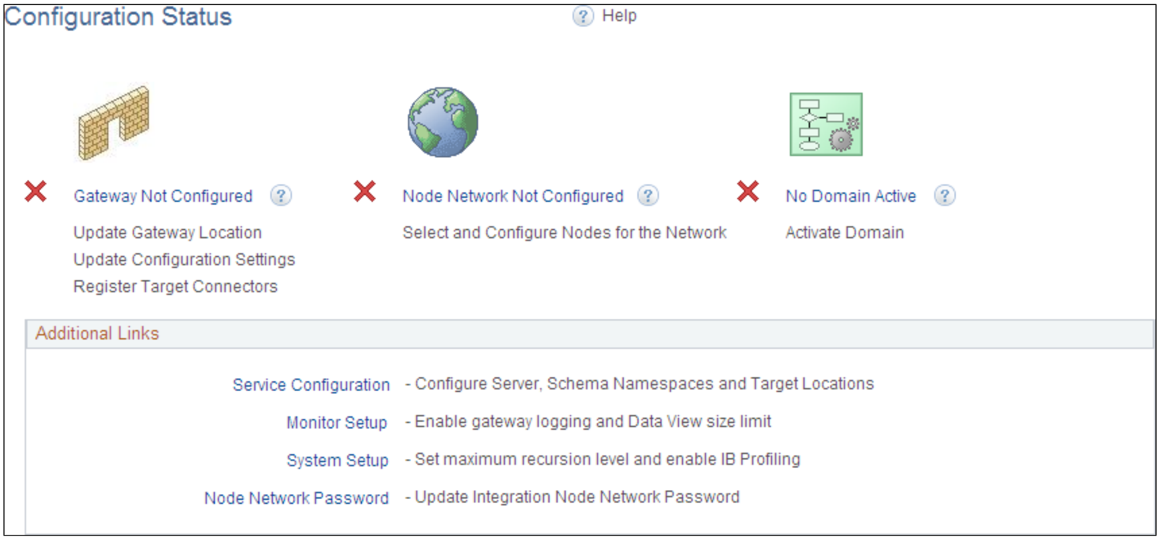
1. Create the shared gateway definition (designated portal system only).
2. Point to the shared gateway (content provider systems).

Creating the Shared Gateway Definition

To configure the shared Integration Broker integration gateway on your designated portal system:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
2. In the Configure Unified Navigation navigation collection, select the **IB Network Status** link to open the integration network's Configuration Status page.

This example illustrates the Configuration Status page showing that the Integration Network is not configured.



3. Click the Gateway Not Configured link.

   The Gateways page appears.

4. Configure the shared integration gateway:

   a. Enter the machine name and complete URL to the PeopleSoftListeningConnector in the **Gateway URL** field.

   Note: Copy or make note of this full gateway URL for use when configuring each of the content provider systems.

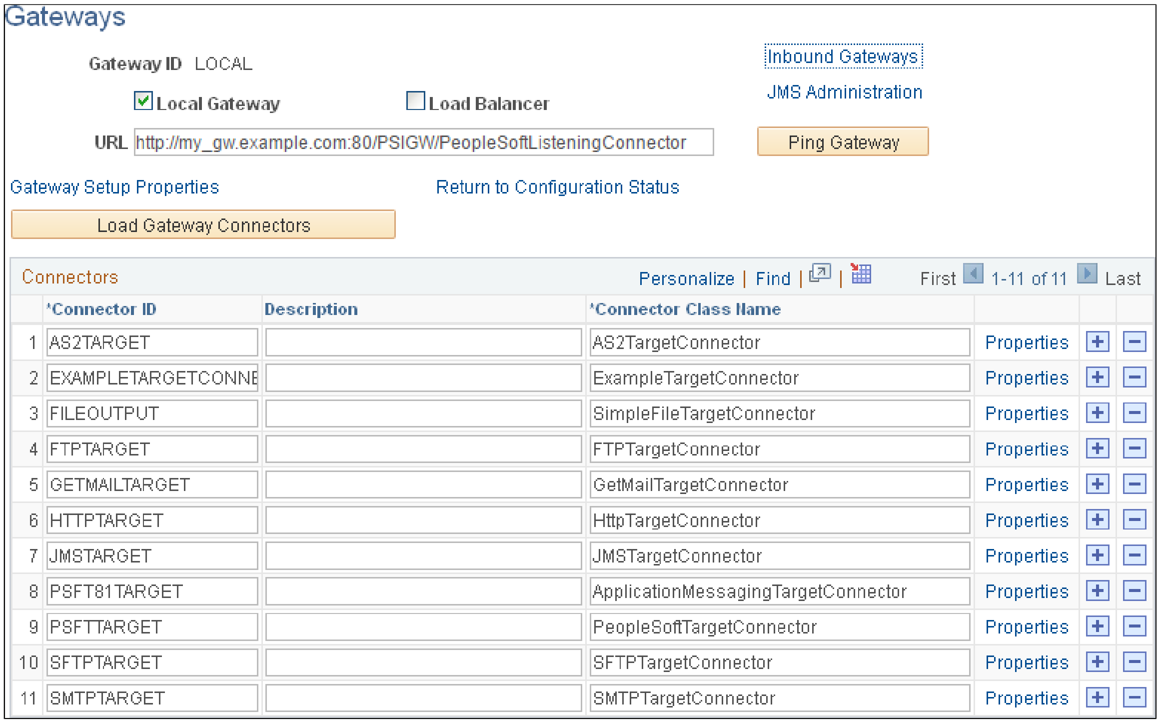
   ---

   b. Click the **Ping Gateway** button.

   The status should return as active.

   c. Click the **Load Gateway Connectors** button to load the gateway connectors.

This example illustrates the fields and controls on the Gateways page with connectors loaded.



d. Click the **Save** button to save your changes.

5. Click the **Gateway Setup Properties** link.

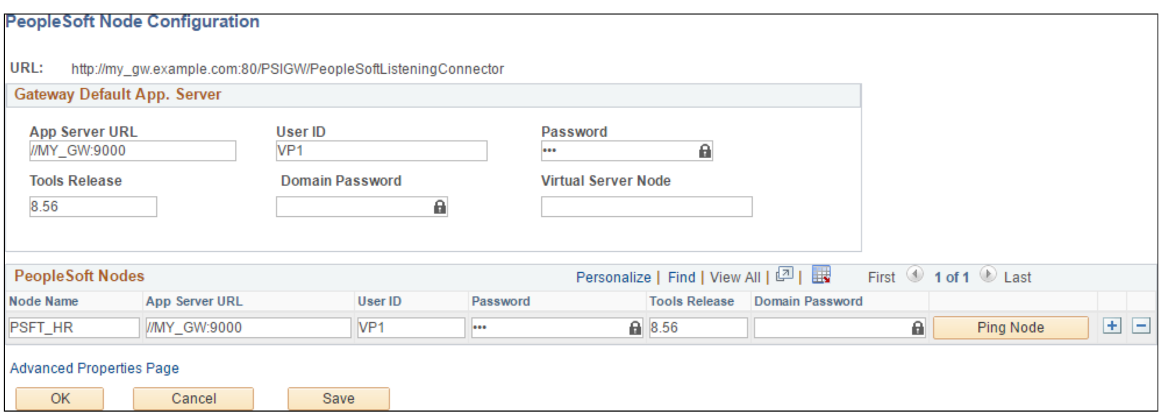
6. Log in on the Gateway Properties page.

   The PeopleSoft Node Configuration page appears.

   a. Enter the URL to the gateway and other values in the Gateway Default App. Server group box.

   b. At a minimum, enter the values for the default local node in the PeopleSoft Nodes group box.

   This example illustrates the fields and controls on the PeopleSoft Node Configuration page.



   c. Click the Save button.

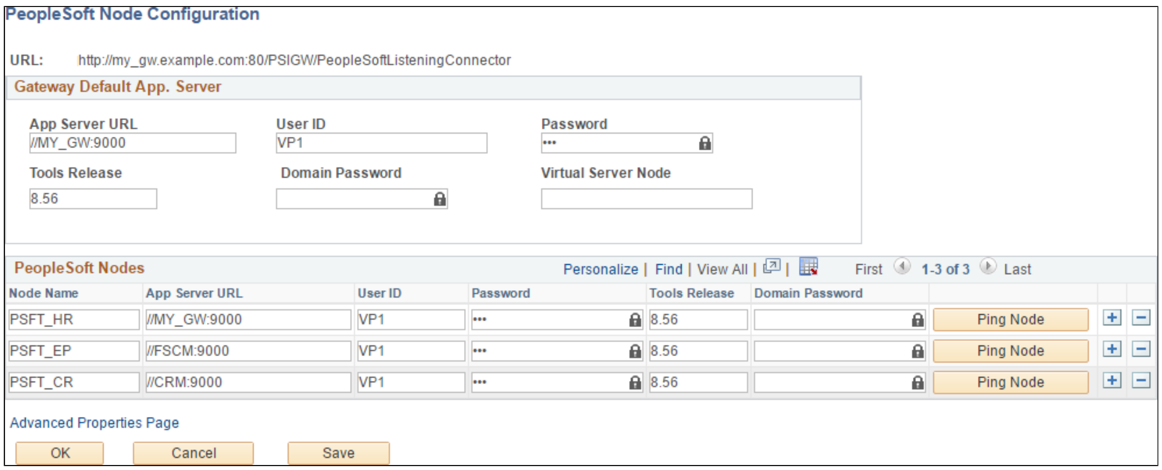
d. Click the Ping Node button.

The status should return as success.

e. On the Ping Node Results page, click the Return button.

f. If you have the information now, you can define other participating nodes in the PeopleSoft Nodes group box at this time. For each content provider system, create entries for the default local node.

This example illustrates the PeopleSoft Node Configuration page with a shared gateway configured.



Note: Alternatively, you can return to this PeopleSoft Node Configuration page once you have identified and defined these nodes.

See Understanding Participating Nodes.

g. Save your changes.

h. Then, for each default local node defined, click the Ping Node button.

The status might not return success if the node is not yet defined with the same parameters in both systems.

Note: If you click the Ping Node button for a any node that is a portal host node, the following error will result:

`Integration Broker Service: Destination node does not match the local node. (158,506)`

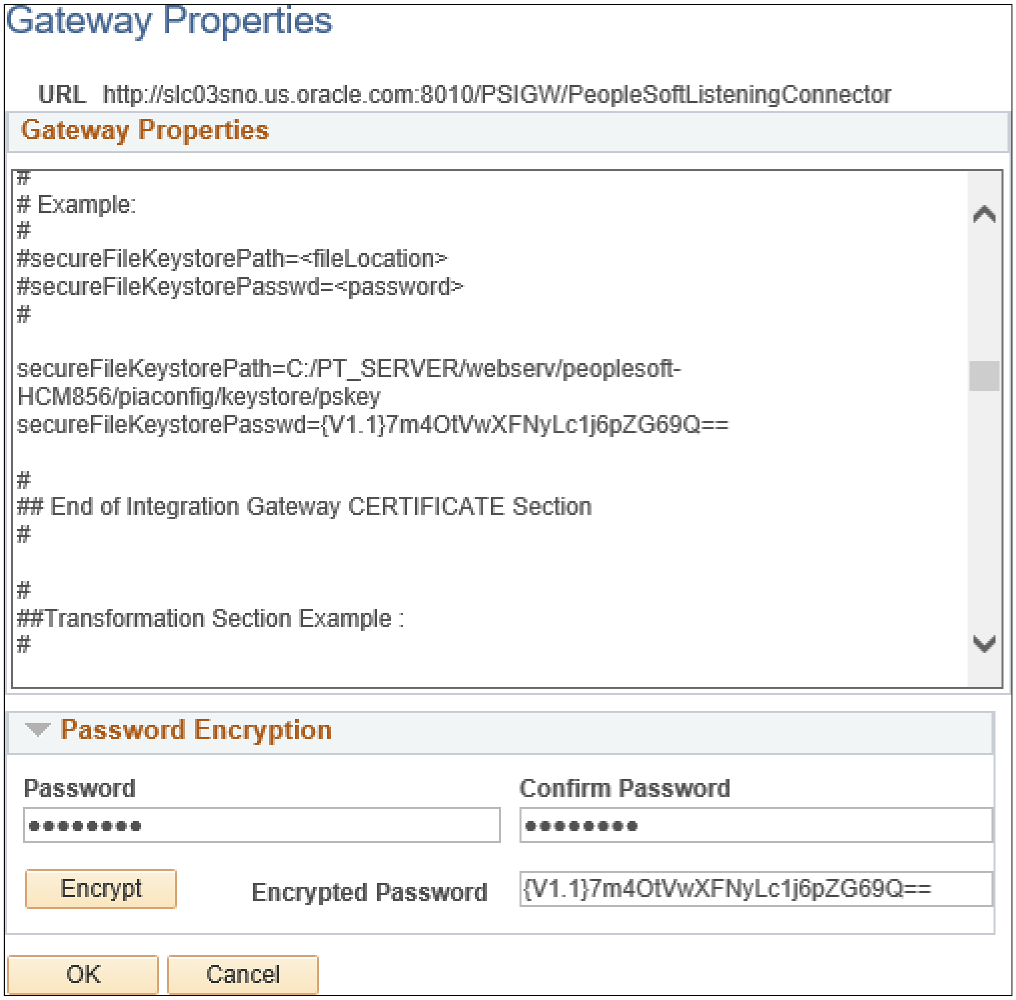
---

7. On the PeopleSoft Node Configuration page, click the **Advanced Properties Page** link.

The Gateway Properties page appears.

a. Enter the full path to the key store file and enter the encrypted key store password:

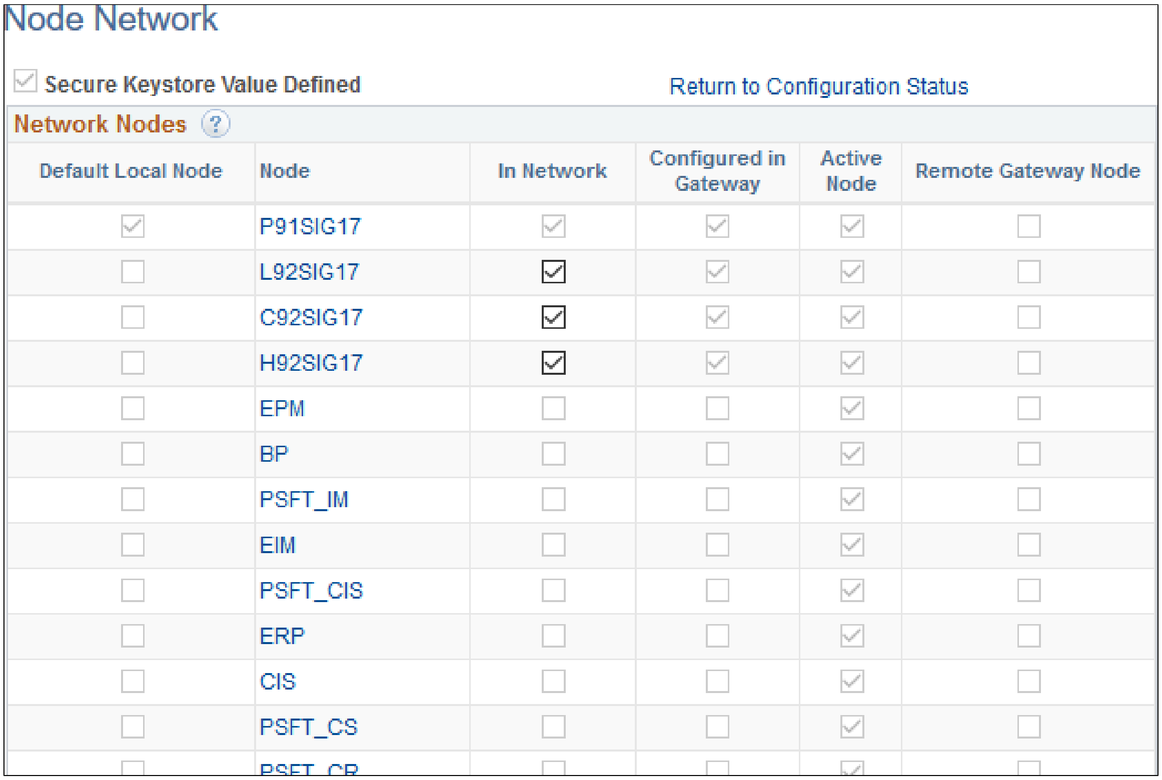
This example illustrates the fields and controls on the Gateway Properties page for setting the key store location and encrypted password.



b. Click the **OK** button.

8. On the PeopleSoft Node Configuration page, click the **Save** button again.
9. Click the **OK** button.
10. On the Gateways page, click the **Return to Configuration Status** link.
11. Click the Node Network Not Configured link (or the Node Network Configured link if that is displayed instead).
12. Click the heading for the Configured in Gateway column twice to sort the grid by nodes that have been configured in the shared gateway.
13. Ensure that the portal system's default local node is selected as in network (it is selected by default).

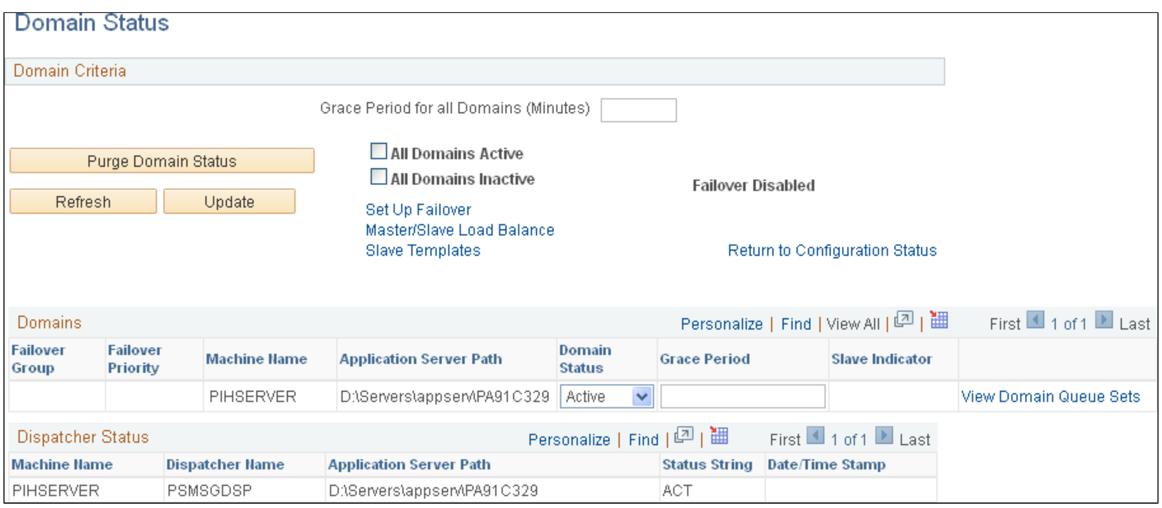
This example illustrates the Node Network page showing six default local nodes as in-network.



14. Click Save, then click the Return to Configuration Status link.
15. Click the No Domain Active link.

    The Domain Status page appears.

    This example illustrates the Domain Status page with an activated domain.

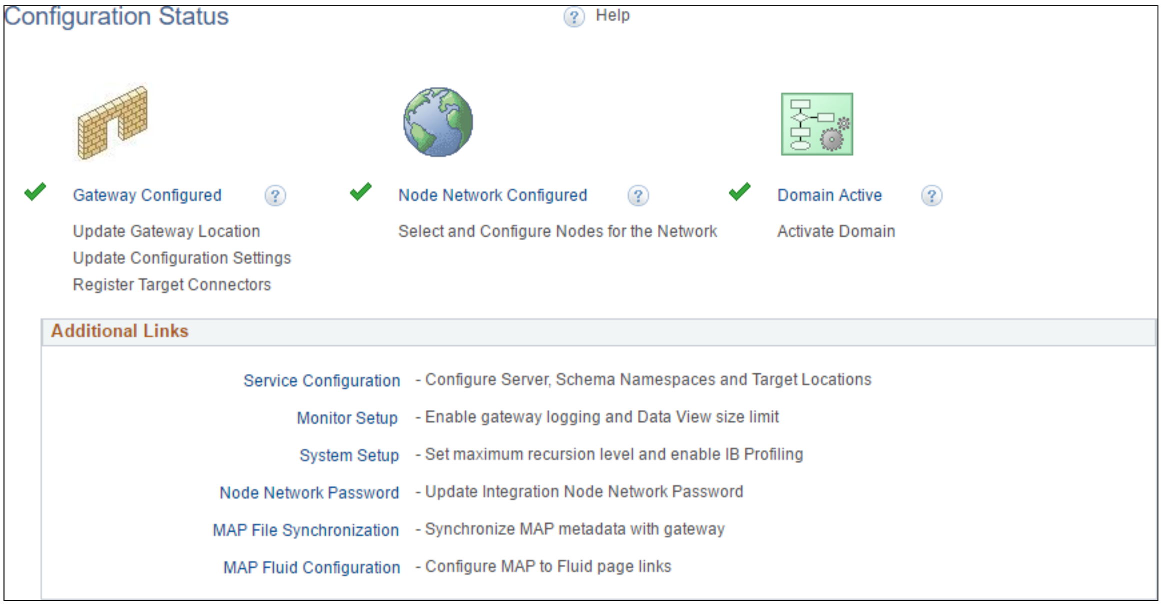


    a. In the Domains group box, set the status for this machine to *Active.*
    b. Click the Update button to update the domain status.

16. Return to the Configuration Status page (click the Return to Configuration Status link).

    The status should show that all three steps have been completed and that your integration gateway and integration network are configured.

    This example illustrates the Configuration Status page showing a configured integration network.



Important! While this integration network is active, its configuration might not yet be complete if you have not added all remote nodes to the integration gateway.

Related Links

"Using the Configuration Status Page" (Integration Broker Administration)
"Administering Integration Gateways" (Integration Broker Administration)
"Setting Oracle Jolt Connection Properties" (Integration Broker Administration)
"Using the integrationGateway.properties File" (Integration Broker Administration)
"Registering Nodes in the Network" (Integration Broker Administration)
"Working with the Domain Status Page" (Integration Broker Administration)

Pointing to the Shared Gateway

To point to the shared Integration Broker integration gateway on each content provider system:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)

2. In the Configure Unified Navigation navigation collection, select the **IB Network Status** link to open the integration network's Configuration Status page.

3. Click the Gateway Not Configured link.

   The Gateways page appears.

4. Point to the shared integration gateway:

a. Enter the complete URL to the PeopleSoftListeningConnector for the shared integration gateway the **Gateway URL** field.

Note: This is the full gateway URL that you copied or made note of previously when configuring the portal system.

---

b. Click the **Ping Gateway** button.

The status should return as active.

c. Click the **Save** button to save your changes.

5. On the Gateways page, click the **Return to Configuration Status** link.
6. Click the No Domain Active link.

   The Domain Status page appears.

   a. In the Domains group box, set the status for this machine to *Active.*

   b. Click the Update button to update the domain status.

7. Return to the Configuration Status page (click the Return to Configuration Status link).

   The status should show that all two steps have been completed.

Setting Up Single Signon

This section provides an overview of single signon and participating nodes, and discusses how to:

- Define all participating nodes.
- Identify trusted nodes.
- Adding remote nodes to the integration network.
- Testing single signon.

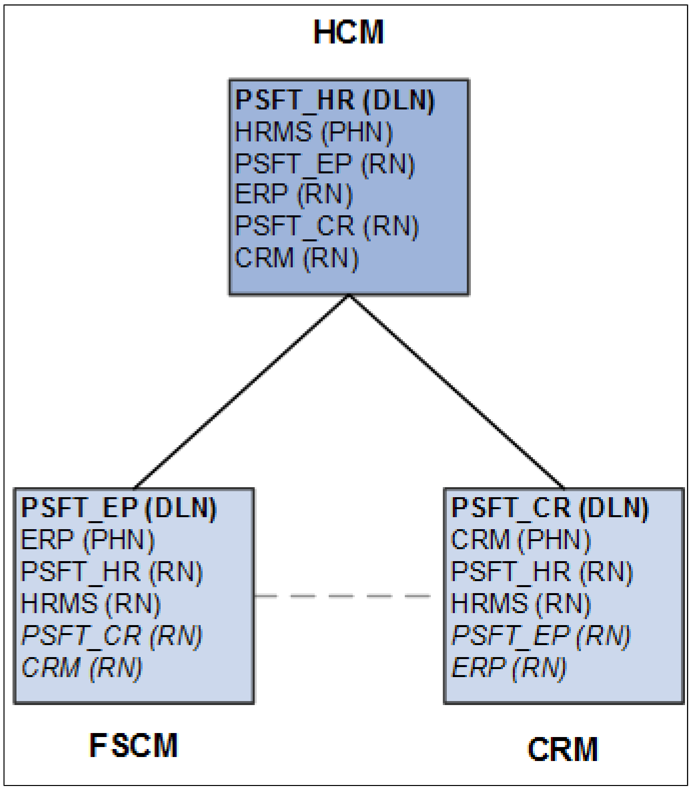
Understanding Single Signon for Unified Navigation

In a PeopleSoft environment, single signon is deployed for a number of reasons. For users, single signon provides the ability for them to navigate freely within a environment of multiple applications after being authenticated only once. For integration of PeopleSoft applications and systems, single signon identifies those systems that are trusted participants in the integration. With unified navigation, single signon is deployed for both purposes: to identify those trusted systems and to allow users to navigate freely to resources on those trusted systems.

Understanding Participating Nodes

The first step to configuring single signon is to define the participating nodes from each content provider system in the designated portal system, and conversely, to define the default local node of the portal system in each content provider system. Then, in each content provider system, define the participating nodes from the portal system *and* from all other content provider systems in the cluster.

The following diagram illustrates three systems participating in unified navigation: HCM (the designated portal system), FSCM, and CRM.



Under each system, the default local node is highlighted in bold and identified with the notation (DLN). For example, on the HCM system, the default local node is PSFT_HR. On the FSCM system, the default local node is named PSFT_EP. The portal host nodes are identified with the notation (PHN). On the FSCM system, the portal host node is named ERP. On the CRM system, the portal host node is named CRM.

Important! Unified navigation supports same portal integration only—for example, EMPLOYEE portal to EMPLOYEE portal, CUSTOMER portal to CUSTOMER portal, or PARTNER portal to PARTNER portal, and so on. Moreover, you should always use the content provider system's portal host node to create remote folders or import pagelets.

Finally, under each system, the remote nodes that need to be defined are identified with the notation (RN). For example, on the HCM system, four remote nodes would need to be defined: PSFT_EP, ERP, PSFT_CR, and CRM. On each of the content provider systems, the two remote nodes representing the portal system would need to be defined at a minimum: PSFT_HR and HRMS. Then, on each content provider system, configure each participating node from each of the other content provider systems. These additional remote nodes are also designated with (RN) and are highlighted in italics.

Note the following when creating node definitions:

- Using delivered (standard) node names simplifies the number of steps to complete the integration.
- Each node name must be unique across all of the participating systems.

  For example, multiple copies of a PeopleSoft application can be federated as content provider systems in a cluster—for example, two copies of PeopleSoft FSCM. You must rename both the default local node and portal host node on one of the copies—for example, PSFT_EP2 and ERP2.
- When using non-standard (non-delivered) node names, you must create node definitions for these nodes in each participating system.

Defining All Participating Nodes

This section discusses how to:

- Define default local nodes.
- Define portal host nodes.

Defining Default Local Nodes

To define default local nodes on all systems:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
2. In the Configure Unified Navigation navigation collection, expand the Single Signon Setup section and select the **Configure Node for SSO** link to open the Nodes page in the target area.
3. If the node name already exists in the system, then select that node definition. Otherwise, add a new value for the non-standard node name.
4. Select the Node Definitions page:
   a. Enter a description for the node.
   b. Verify that the node type is PIA.
   c. For default local nodes, the Default Local Node, Local Node, and Active Node check boxes must be selected.

   Note: When configuring a remote node definition for this node, only the Active Node check box must be selected.

   d. For default local nodes only, set the authentication option to *Password* or *Certificate.* If the authentication type is password, enter a node password.

   Note: When configuring a remote node definition for this node, you must enter the same node password.

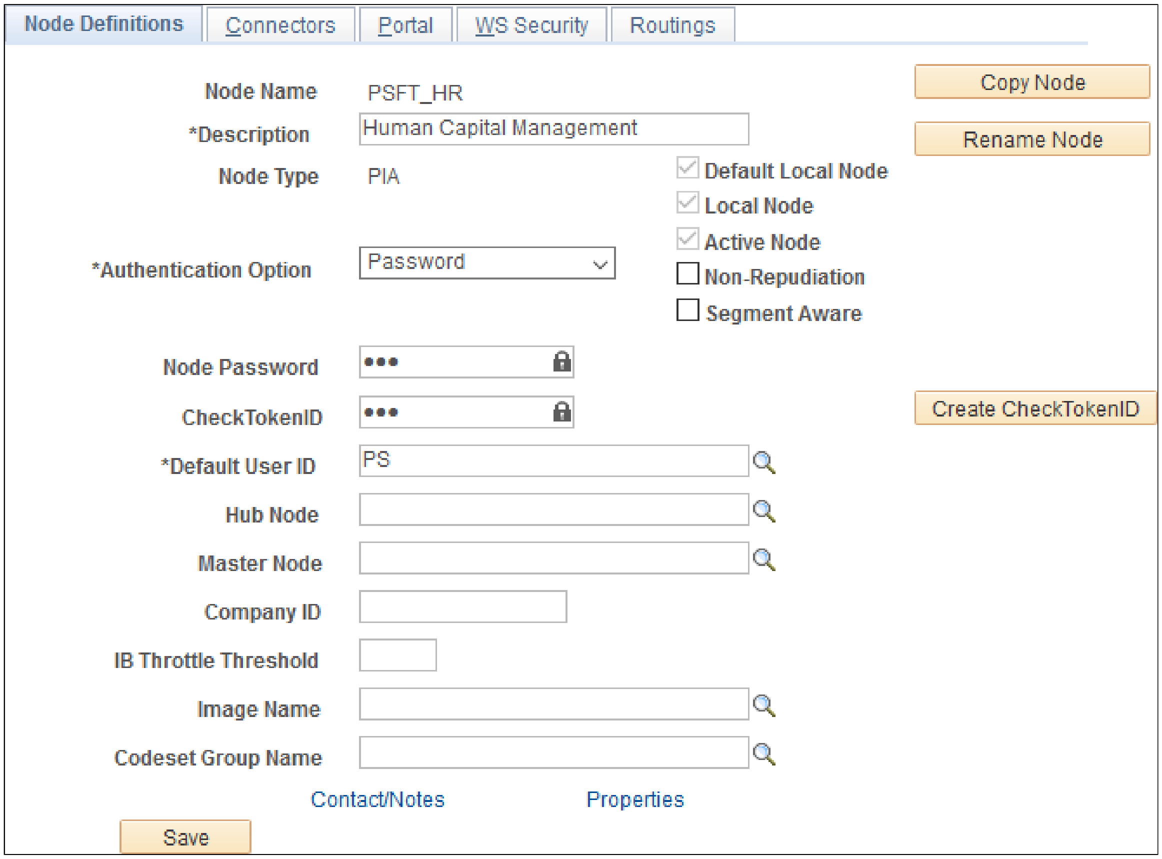
   e. Enter the default user ID for the node.

   Note: When configuring a remote node definition for this node, you must enter the same default user ID. If necessary, create a user profile for this ID.

f. Create a check token ID for the node, which is required for single signon configurations.

Note: When configuring a remote node definition for this node, you must enter the same check token ID.

This example illustrates the fields and controls on the Node Definitions page showing the default local node for the portal system (HCM).



5. Select the Connectors page:

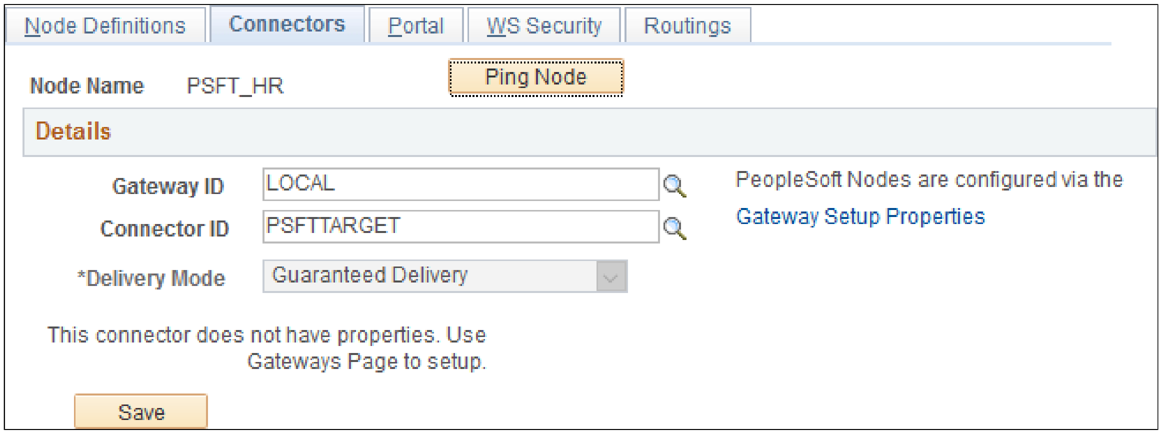
a. Enter the ID for the shared integration gateway in the Gateway ID field.

Note: When configuring a remote node definition for this node, use the same ID.

---

b. Enter PSFTTARGET as the connector ID.

This example illustrates the fields and controls on the Connectors page showing the default local node for the portal system (HCM).



c. If the remote node will use the shared integration gateway and you did not add this node to the gateway configuration previously, do so now. If you are creating all node definitions before configuring the gateway, then remember the node name for each default local node and enter into the gateway properties later.

6. Select the Portal page:

a. Ensure that the correct default portal is selected.

b. Ensure that a value is defined for the Tools Release field.

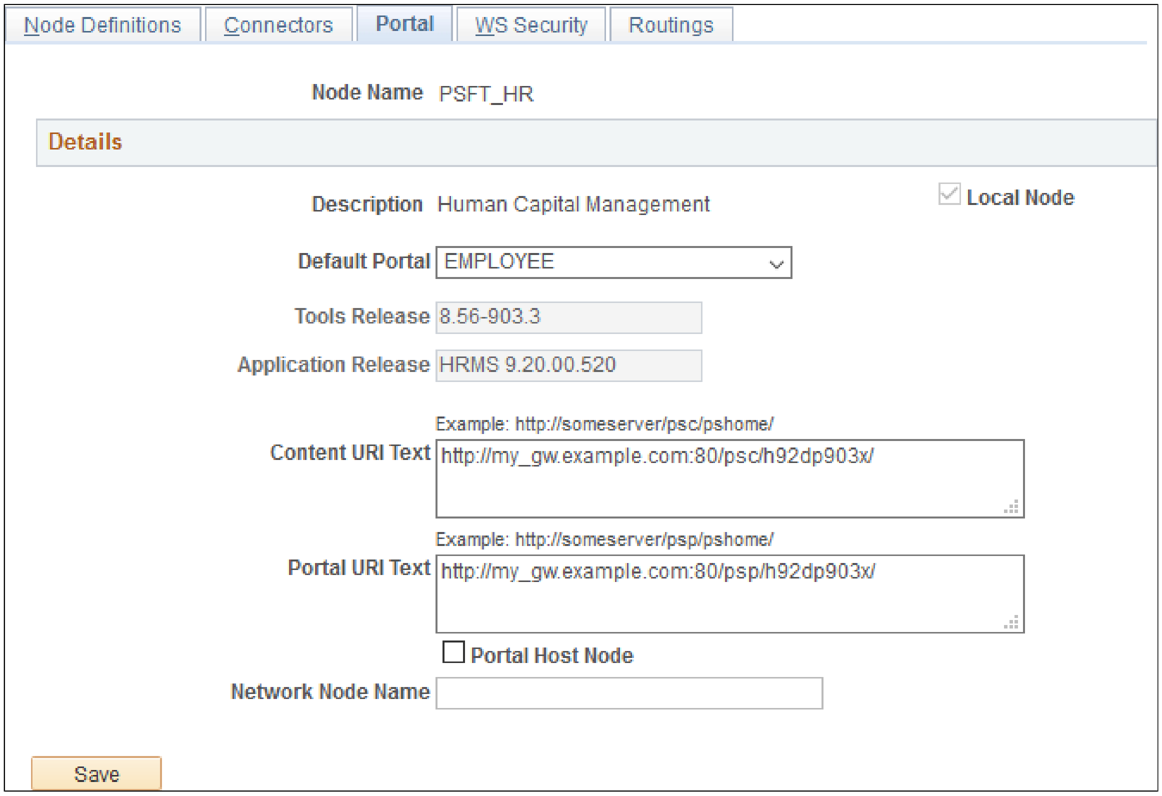
c. Enter values for the Content URI Text field and the Portal URI Text field as are appropriate for the web server port and site on this local system.

Note: When configuring a remote node definition for this node, use the same values as defined on the local system.

d. For each remote node definition, enter the name of the default local node of the remote system in the Network Node Name field.

Note: Do not set this field on local node definitions.

This example illustrates the Portal page showing default local node of the portal system (HCM).



e. Save the node definition.

7. Repeat steps 2 through 6 for each default local node on each content provider system.
8. Then, using the values specified in the local node definitions, repeat steps 2 through 6 for each remote definition of each default local node.

Defining Portal Host Nodes

To define portal host nodes on all systems:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
2. In the Configure Unified Navigation navigation collection, expand the Single Signon Setup section and select the **Configure Node for SSO** link to open the Nodes page in the target area.
3. If the node name already exists in the system, then select that node definition. Otherwise, add a new value for the non-standard node name.
4. Select the Node Definitions page:

    a. Enter a description for the portal host node.

    b. Verify that the node type is PIA.

    c. For portal host nodes, the Local Node and Active Node check boxes must be selected.

Note: When configuring a remote node definition for this node, only the Active Node check box must be selected.

---

d. For portal host nodes, the Authentication Option field value is *None*.

e. Enter the default user ID for the node.

Note: When configuring a remote node definition for this node, you must enter the same default user ID. If necessary, create a user profile for this ID.

---

5. Select the Connectors page:

a. Enter the ID for the shared integration gateway in the Gateway ID field.

Note: When configuring a remote node definition for this node, use the same ID.

---

b. Enter PSFTTARGET as the connector ID.

Note: If you click the Ping Node button for a remote node that is a portal host node, the following error will be displayed:

```
Integration Broker Service: Destination node does not match the
local node. (158,506)
```

---

6. Select the Portal page:

a. Ensure that the correct default portal is selected.

b. Ensure that a value is defined for the Tools Release field.

c. Enter values for the Content URI Text field and the Portal URI Text field as are appropriate for the web server port and site on this local system.

Note: When configuring a remote node definition for this node, use the same values as defined on the local system.

---

d. Select the Portal Host Node check box only for a node definition for a remote portal host node.

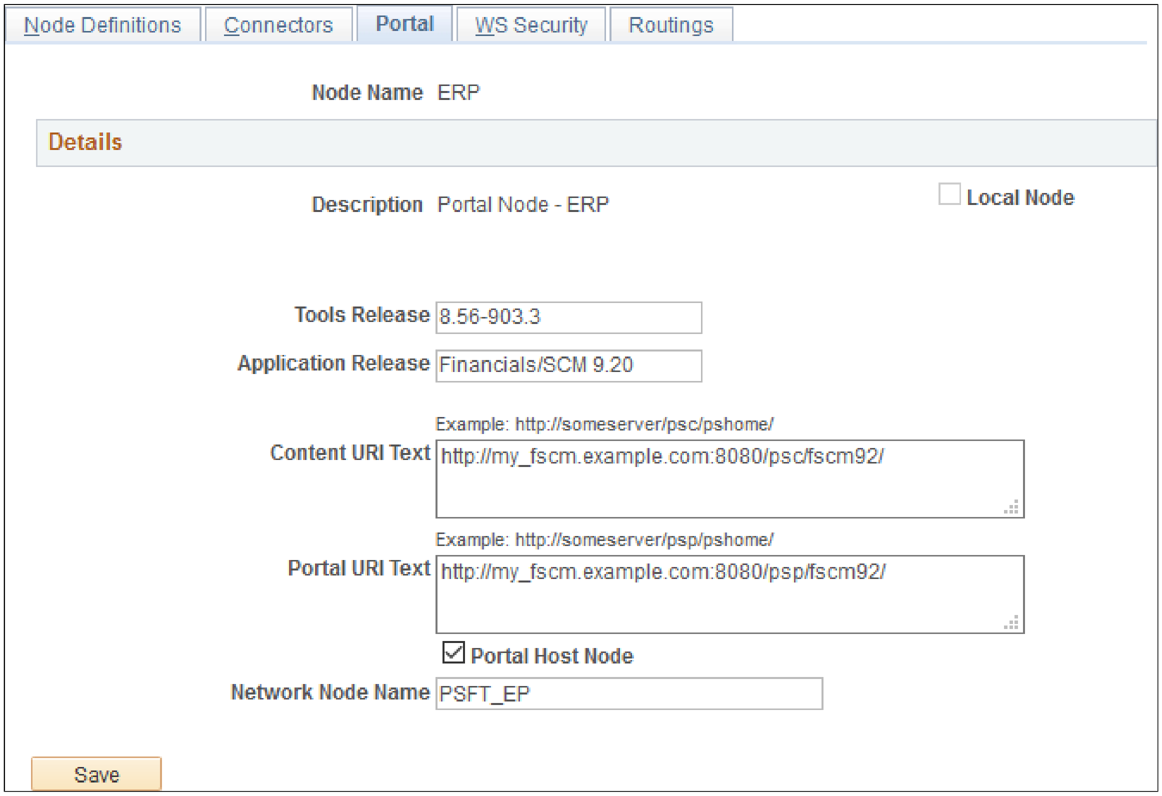
Note: Do not select this check box on local node definitions.

---

e. For each remote node definition, enter the name of the default local node of the remote system in the Network Node Name field.

Note: Do not set this field on local node definitions.

This example illustrates the Portal page showing the definition of a remote portal host node (ERP from the FSCM system).



f. Save the node definition.

7. Repeat steps 1 through 6 for each portal host node on each content provider system.
8. Then, using the values specified in the local node definitions, repeat steps 1 through 6 for each remote definition of each portal host node.

Related Links

"Defining Node Parameters" (Integration Broker Administration)

"Specifying Gateways and Connectors" (Integration Broker Administration)

Setting Portal Node Characteristics

"Implementing PeopleSoft-Only Single Signon" (Security Administration)

Identifying Trusted Nodes

After the remote nodes have been defined on each system, all the default local nodes participating in the single signon configuration need to be identified on *each* system.

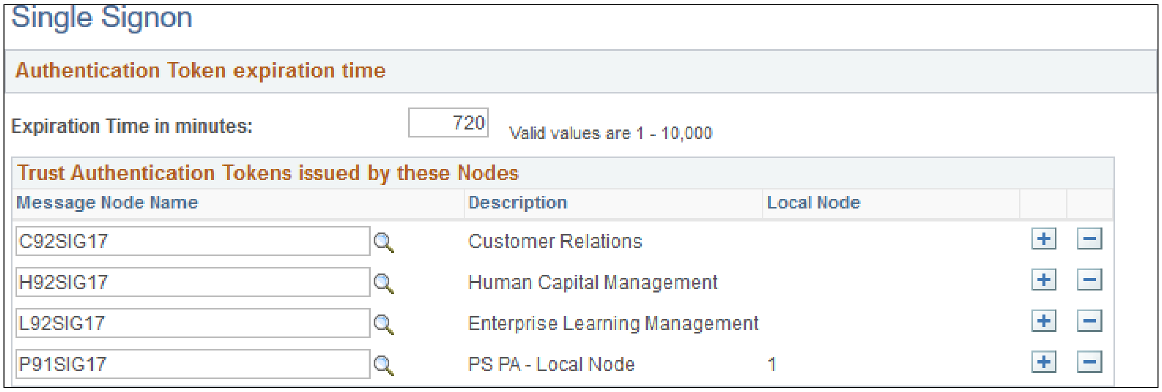
To identify the trusted nodes on each system:

1. Select **PeopleTools** > **Portal** > **Configure Unified Navigation**.

2. In the Configure Unified Navigation center, expand the Single Signon Setup section and select the **Define Trusted Nodes** link to open the Single Signon page in the target area.

3. Add the default local node from each content provider system as a trusted node.

   This example illustrates the Single Signon page listing all trusted default local nodes.

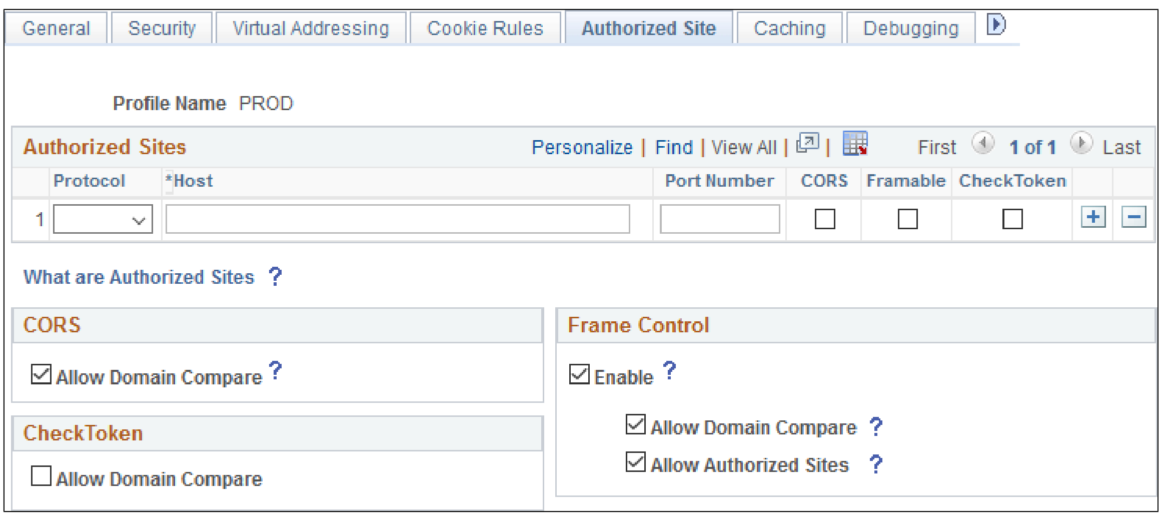


4. Save the list of trusted nodes.

5. Select **PeopleTools** > **Web Profile** > **Web Profile Configuration**.

6. Select the Authorized Site page.

   This example illustrates the fields and controls on the Web Profile Configuration - Authorized Site page.



7. Using the CheckToken settings, add each remote host participating in the single signon configuration, either explicitly, by specifying the host's domain or subdomain, or by selecting the Allow Domain Compare check box when appropriate.

8. Save the configuration. Then, stop and restart the web server.

Related Links

"Working with the Single Signon Page" (Security Administration)
Configuring Authorized Sites

Adding Remote Nodes to the Integration Network

Once content provider nodes have been defined in the portal system, only default local nodes need to be added to the integration network to complete the network configuration.

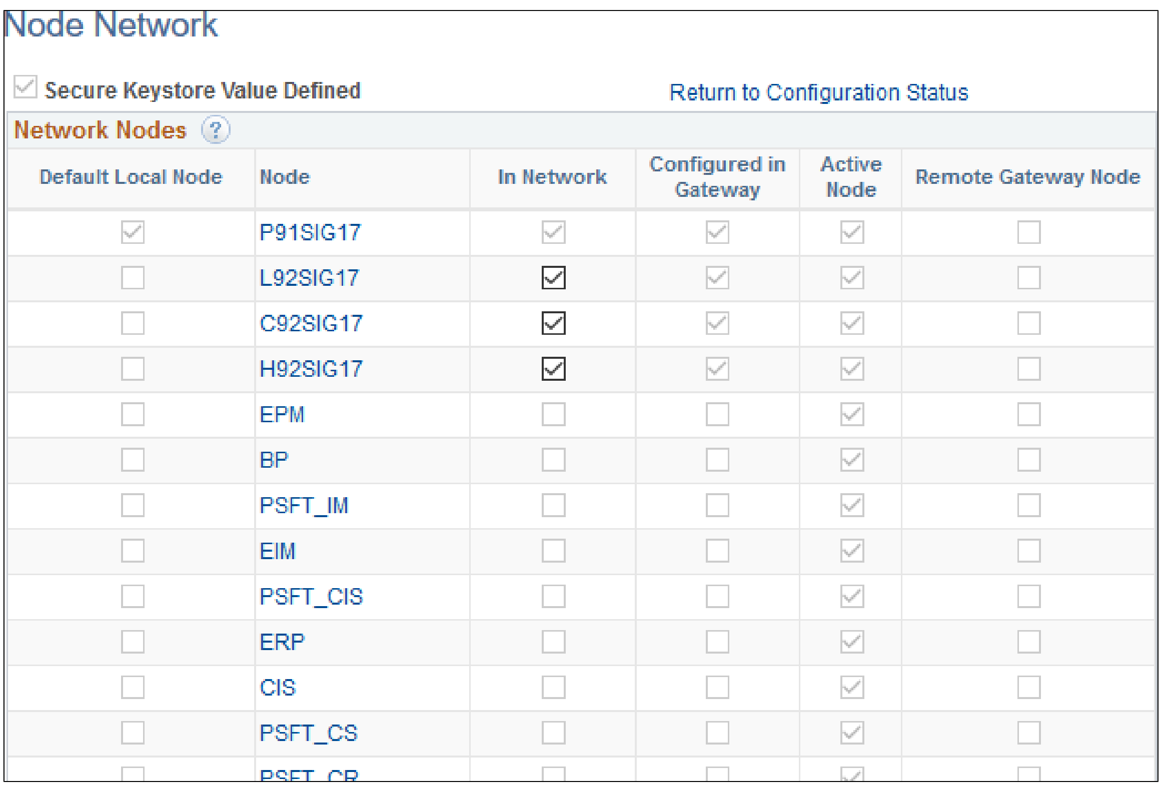
Note: If you added default local nodes from the content provider systems to the integration network previously, then you will not need to complete this procedure.

See Configuring the Integration Gateway.

To add default local nodes from the content provider systems to the integration network:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
2. In the Configure Unified Navigation navigation collection, select the **IB Network Status** link to open the integration network's Configuration Status page.
3. Click the Node Network Configured link.
4. For each remote node that is a default local node, select the check box to make the node in network.

   This example illustrates the Node Network page showing six default local nodes as in-network.



5. Save the changes to the integration network.

Related Links

“Registering Nodes in the Network” (Integration Broker Administration)

Completing the Unified Navigation Configuration

This topic discusses how to:

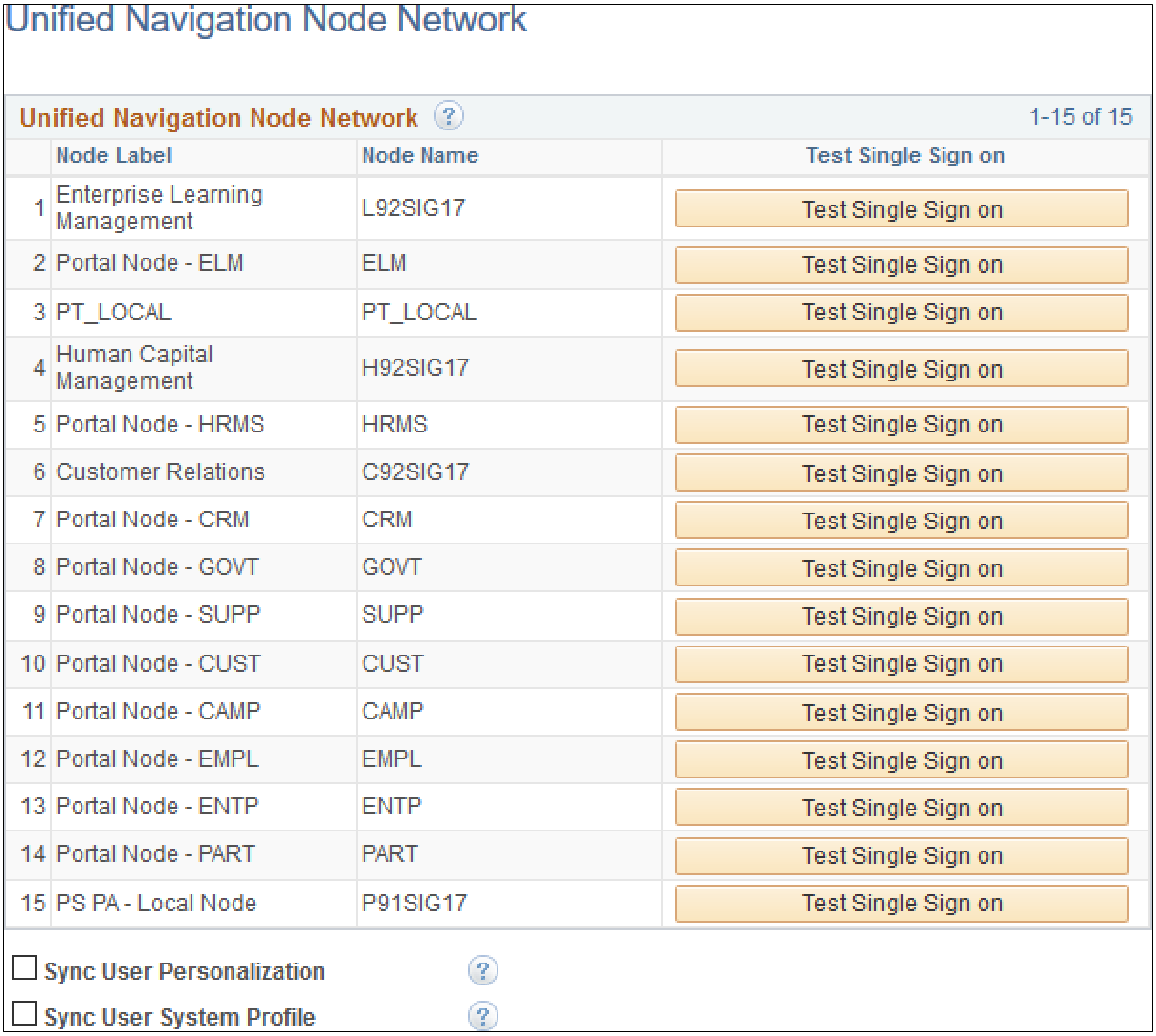
- Test the single signon configuration.
- Synchronize user data.
- Configure remote content.
- Complete configuration of remote content.

Testing the Single Signon Configuration

To test single signon in the unified navigation node network on the designated portal system:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
2. In the Configure Unified Navigation navigation collection, expand the Unified Navigation Setup section and click the **Unified Navigation Network** link to open the Unified Navigation Node Network page in the target area.

This example illustrates the fields and controls on the Unified Navigation Node Network page.



Unified Navigation Node Network

Unified Navigation Node Network 1-15 of 15

	Node Label	Node Name	Test Single Sign on
1	Enterprise Learning Management	L92SIG17	Test Single Sign on
2	Portal Node - ELM	ELM	Test Single Sign on
3	PT_LOCAL	PT_LOCAL	Test Single Sign on
4	Human Capital Management	H92SIG17	Test Single Sign on
5	Portal Node - HRMS	HRMS	Test Single Sign on
6	Customer Relations	C92SIG17	Test Single Sign on
7	Portal Node - CRM	CRM	Test Single Sign on
8	Portal Node - GOVT	GOVT	Test Single Sign on
9	Portal Node - SUPP	SUPP	Test Single Sign on
10	Portal Node - CUST	CUST	Test Single Sign on
11	Portal Node - CAMP	CAMP	Test Single Sign on
12	Portal Node - EMPL	EMPL	Test Single Sign on
13	Portal Node - ENTP	ENTP	Test Single Sign on
14	Portal Node - PART	PART	Test Single Sign on
15	PS PA - Local Node	P91SIG17	Test Single Sign on

Sync User Personalization

Sync User System Profile

3. Click the Test Single Signon button to validate the single signon configuration for that remote node.

Synchronizing User Data

To synchronize user data from the portal system to all other content provider systems defined in this node network:

1. Select **PeopleTools** > **Portal** > **Configure Unified Navigation**.
2. In the Configure Unified Navigation navigation collection, expand the Unified Navigation Setup section and click the **Unified Navigation Network** link to open the Unified Navigation Node Network page in the target area.
3. Select the Sync User Personalization check box to synchronize user personalization settings (My Personalizations settings such as time zone) from the portal system to all remote other content provider systems.

4. Select the Sync User System Profile check box to synchronize user profile settings (My System Profile settings such as email addresses) from the portal system to all other remote content provider systems.

5. Click the Save button to execute the synchronization.

Configuring Remote Content

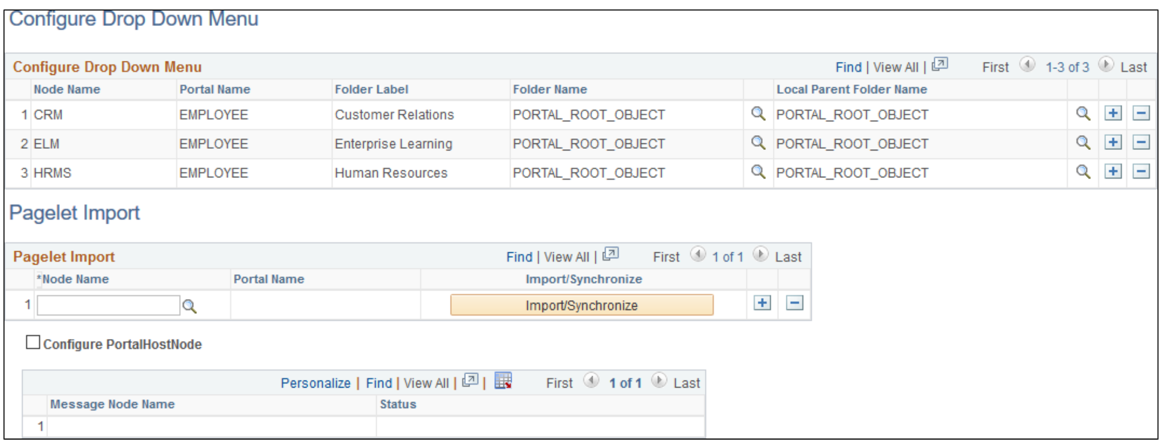
Configuring remote content creates access to the menu structure of the remote system within the main menu of the portal system. When a remote folder or content reference is added to the main menu of the portal system, it can be added under a unique local parent folder. Alternatively, multiple remote folders or content references can be added under a single local parent folder.

To add remote folders and content references to the main menu of the portal system:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)

2. In the Configure Unified Navigation navigation collection, expand the Unified Navigation Setup section and click the **Select Remote Content** link to open the Configure Drop Down Menu page in the target area.

This example illustrates the fields and controls on the Configure Drop Down Menu page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Node Name	Select a *portal host node* from the list of defined content provider nodes. **Note:** While default local nodes are also listed, use the portal host node for the content provider.
Portal Name	The current portal name is automatically filled in after you select the remote folder or local parent folder.

Field or Control	Description
Folder Label	Enter the label for the new folder as it will appear in the designated portal system's main menu. **Important!** Do not use the following special characters in the labels for a remote folder: { } #.
Folder Name	Click the lookup button, navigate the tree of folders, and select the remote folder from the content provider node. **Note:** The selected remote folder and all menu items and subfolders contained within it will be available in the main menu on the designated portal system.
Local Parent Folder Name	Click the lookup button, navigate the tree of folders, and select the parent folder on the designated portal system.
Pagelet Import	The Pagelet Import grid is discussed in a different topic. See Importing Pagelets from Content Provider Systems.
Configure PortalHostNode	**Important!** Whenever you add a remote folder to the unified navigation configuration, you must always select the Configure PortalHostNode check box before saving the configuration. In a unified navigation configuration, the Configure PortalHostNode field ensures that the Home, Notifications, Logout options on the portal system's header point to the portal system.

Configuring Multiple Remote Folders in the Same Local Parent Folder

Two or more remote folders can be configured under the same local parent folder. To configure multiple remote folders:

1. In the **Configure Drop Down Menu** grid, click the **Add row** button to add a new row.
2. In the **Node Name** field, enter or select a portal host node from the list of defined content provider nodes.
3. In the **Folder Label** field, enter the label for the new folder as it will appear in the portal system's main menu.

   Important! Do not use the following special characters in the labels for a remote folder: { } #.

4. In the **Folder Name** field, click the lookup button, navigate the tree of folders, and select the remote folder from remote content provider node.
5. If the next remote folder is on the same remote node, skip to the next step. Otherwise, in the **Node Name** field, enter or select a different portal host node from the list of defined content provider nodes.

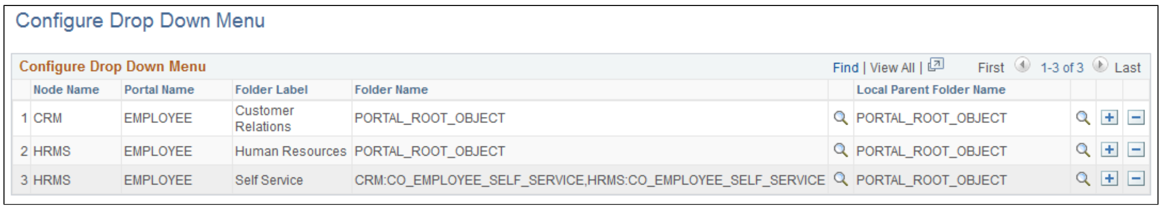
6. In the **Folder Name** field, click the lookup button, navigate the tree of folders, and select the remote folder from remote content provider node.

7. Repeat steps 5 and 6 for each remote folder to be added under the local parent folder.

8. In the **Local Parent Folder Name** field, click the lookup button, navigate the tree of folders, and select the local root folder on the portal system.

9. Click Save to save the configuration.

   In the following example, the local folder labelled Self Service includes two remote folders from different content providers: CRM:CO_EMPLOYEE_SELF_SERVICE and HRMS:CO_EMPLOYEE_SELF_SERVICE.



Configure Drop Down Menu

Configure Drop Down Menu — Find | View All | First 1-3 of 3 Last

	Node Name	Portal Name	Folder Label	Folder Name	Local Parent Folder Name
1	CRM	EMPLOYEE	Customer Relations	PORTAL_ROOT_OBJECT	PORTAL_ROOT_OBJECT
2	HRMS	EMPLOYEE	Human Resources	PORTAL_ROOT_OBJECT	PORTAL_ROOT_OBJECT
3	HRMS	EMPLOYEE	Self Service	CRM:CO_EMPLOYEE_SELF_SERVICE,HRMS:CO_EMPLOYEE_SELF_SERVICE	PORTAL_ROOT_OBJECT

Completing Configuration of Remote Content

To complete configuration of remote content, you must do the following:

- Update all content reference definitions that specify LOCAL_NODE on each content provider system.
- Synchronize security settings for remote folders in the portal system.

Updating All Content Reference Definitions That Specify LOCAL_NODE

On each content provider system, update all content reference definitions that specify LOCAL_NODE as the node name to use the portal host node name instead. For example, on a PeopleSoft FSCM content provider system, the following SQL would update LOCAL_NODE to the ERP node:

```
update PSPRSMDEFN set PORTAL_CNTPRV_NAM='ERP' where PORTAL_REFTYPE='C' and PORTAL_C⇒

NTPRV_NAM='LOCAL_NODE' and PORTAL_NAME='EMPLOYEE';
```

After updating the content reference definitions, you must delete the application server cache on both the portal system and the content provider system, and then restart both application servers.

Synchronizing Security Settings For Remote Folders

Remote folders are added with security set to public in the portal system. Therefore, to ensure that only authorized users are allowed to see the remote folders in the main menu, you need to synchronize the new folder's security with the settings from the content provider system. To synchronize the security settings between the systems, you must manually remove the public setting and apply the same security settings that exist in the content provider system to the folder’s content reference definition in the portal system.

Importing Pagelets

Once unified navigation configuration is complete, you can import pagelets from content provider systems for use in the portal system. Your designated portal system offers two methods for importing pagelets:

- You can import or update all pagelets in bulk from a remote node.
- You can import or update an individual pagelet for use within a specific PeopleTools feature, such as homepage tab content, a WorkCenter, or as a related content service.

See Importing Pagelets from Content Provider Systems for details on each of these methods.

Importing Pagelets from Content Provider Systems

Once unified navigation configuration is complete, you can import pagelets from content provider systems for use in the designated portal system. Your portal system offers two methods for importing pagelets:

- You can import all pagelets in bulk from a remote node. After importing pagelets, if changes are made to one or more pagelets on the content provider system, you can update the pagelets in bulk.
- You can import or update an individual pagelet for use within a specific PeopleTools feature, such as homepage tab content, a WorkCenter, or as a related content service.

Note: Templates that are used by the imported pagelets must be copied from the content provider systems to the designated portal database. See Considerations for Remote Content.

Importing Pagelets in Bulk

Use the Pagelet Import section of the Configure Drop Down Menu page to import or update all of the pagelets from a content provider system making them available for use on the portal system.

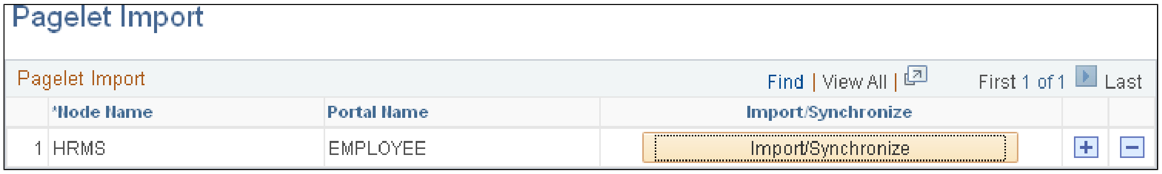
Note: Security is checked at runtime and does not have to be manually copied for imported pagelets.

To import or update all of the pagelets from a content provider system:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
2. In the Configure Unified Navigation navigation collection, expand the Unified Navigation Setup section.
3. Click the **Select Remote Content** link to open the Configure Drop Down Menu page in the target area.

   Note: The Pagelet Import section is displayed beneath the Configure Drop Down Menu section.

This example illustrates the fields and controls in the Pagelet Import section of the Configure Drop Down Menu page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Node Name	Select a portal host node from the list of defined content provider nodes.
Portal Name	The current portal name is automatically filled in after you select the remote node.
Import/Synchronize	Click the Import/Synchronize button to import all pagelets from the content provider into the portal system. After importing pagelets, if changes are made to one or more pagelets on the content provider system, click the Import/Synchronize button to update the pagelets in bulk.

Searching to Import Individual Pagelets

Use the Pagelet Search page to import or update a single pagelet to be used within that PeopleTools feature—for example, on a homepage tab or in a WorkCenter. See Pagelet Search Page for more information.

Setting Up Related Content

Understanding the Setup of Related Content Within Unified Navigation

Configuring related content is a common task. You use the same PeopleTools pages and methodologies as on any other PeopleSoft application system. However, with unified navigation, you have additional options for defining remote resources as related content as well as configuring related content for content references on content provider systems.

Once unified navigation has been configured to incorporate one or more content provider systems, you have expanded options for configuring related content including:

- Creating related content services in the designated portal system using the either local or remote content references.
- Creating related content services in the content provider system using content references local to that system.
- Assigning local (to the portal system) or remote content references (from a content provider system) as related content services (or related actions) to content references on the portal system.

- Assigning local (to the content provider system) or remote content references (from a different content provider system) as related content services (or related actions) to content references on the content provider system.

  For example, this would allow you to assign the content reference for employee expenses from PeopleSoft FSCM to the content reference for employee information from PeopleSoft HCM.

Creating and Assigning Related Content Within the Unified Navigation Framework

This section provides high-level procedures for how to:

- Create related content service definitions.
- Configure related content for application pages on the portal system.
- Configure related content for application pages on a content provider system.

Creating Related Content Service Definitions

Creating a related content service within any system uses standard Related Content Framework pages delivered in PeopleTools. However, you can begin from the Configure Unified Navigation navigation collection as documented here:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
2. In the Configure Unified Navigation navigation collection, expand the Related Content Setup section.
3. To create a service on the portal system, click the **Local Portal System** link.
4. Click the Create a New Related Content Service link.
5. Add a new value.

   The Define Related Content Service page is displayed.
6. Define the related content service:
   - For a service on the portal system, you can base the service on a local resource or on a content reference from a content provider system.
   - For a service on a content provider system, you can base the service on a resource local to that content provider system.

Configuring Related Content for Application Pages on the Portal System

Configuring related content within the portal system uses standard Related Content Framework pages delivered in PeopleTools. However, you can begin from the Configure Unified Navigation navigation collection as documented here:

1. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)

2. In the Configure Unified Navigation navigation collection, expand the Related Content Setup section.
3. To assign related content on the portal system, click the **Local Portal System** link.
4. Click the Assign Related Content to Application Pages link.

   The Select a Content Reference page is displayed.
5. Browse the menu structure and select a content reference on the portal system.

   The Assign Related Content page is displayed.
6. Assign one of the following service types from the following sources:
   - Content reference – From the portal system or from a content provider system.
   - Pagelet – From the portal system.
   - Service – From the portal system.

Configuring Related Content for Application Pages on a Content Provider System

Configuring related content on the content provider system also uses standard Related Content Framework pages delivered in PeopleTools. However, you can begin from the Configure Unified Navigation navigation collection as documented here:

1. Sign on to the content provider system directly.
2. Access the Configure Unified Navigation navigation collection. (See Configure Unified Navigation Fluid Navigation Collection.)
3. In the Configure Unified Navigation navigation collection, expand the Related Content Setup section.
4. To assign related content on the content provider system, click the **Local Portal System** link.
5. Click the Assign Related Content to Application Pages link.

   The Select a Content Reference page is displayed.
6. Browse the menu structure and select a content reference on the content provider system.

   The Assign Related Content page is displayed.
7. Assign one of the following service types from the following sources:
   - Content reference – From the content provider system or from a different remote content provider system.
   - Pagelet – From the content provider system or from a different remote content provider system.
   - Service – From the content provider system.

Related Links

Creating and Managing Related Content Service Definitions

Managing Related Content Configurations and Data

Troubleshooting Unified Navigation Setup

This topic discusses the following troubleshooting resources:

- Unified Navigation Configuration Status page, which displays read-only configuration information.
- Integration Broker Diagnostics page, which displays read-only information on node definitions across all systems.
- Other troubleshooting considerations.

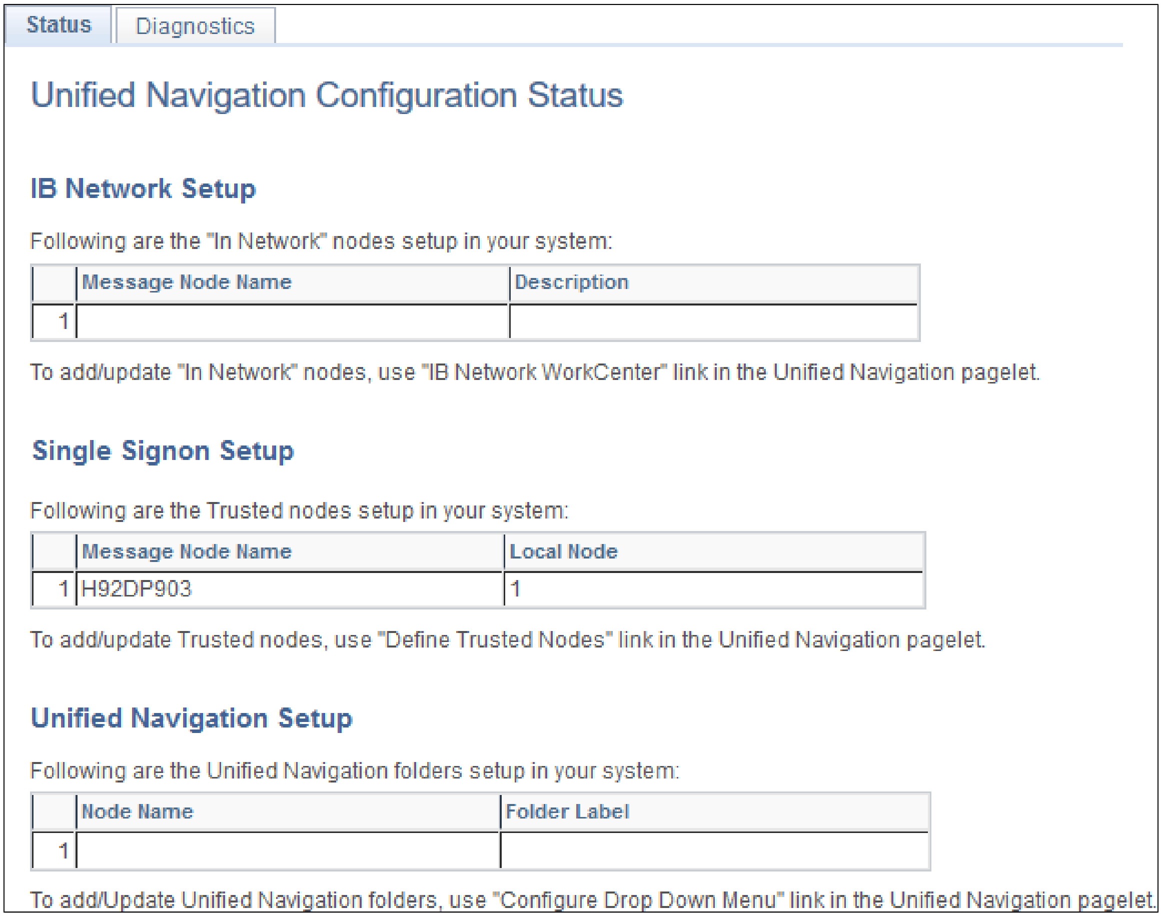
Unified Navigation Configuration Status Page

The Unified Navigation Configuration Status page is a read-only page that displays the nodes in the Integration Broker network, the trusted nodes in single signon, and the folders pointing to remote content that are defined in the designated portal system.

To view the Unified Navigation Configuration Status page:

1. Select **PeopleTools** > **Portal** > **Configure Unified Navigation**.
2. In the Configure Unified Navigation center, click the **Unified Navigation Status** link to open the Unified Navigation Configuration Status page in the target area.

This example illustrates the nodes in the Integration Broker network, the trusted nodes in single signon, and the folders for remote content created in the portal system.



Related Links

Creating the Shared Gateway Definition

Identifying Trusted Nodes

Configuring Remote Content

Integration Broker Diagnostics Page

The Integration Broker Diagnostics page displays details of the nodes in the Integration Broker network and the trusted nodes in single signon. The Integration Broker Diagnostics page enables you to review the details of each active node without having to open the Node Definitions page for each node to check the values entered in the node. However, if you need to modify any value of a node, you must open the Node Definitions page *on the system in question*.

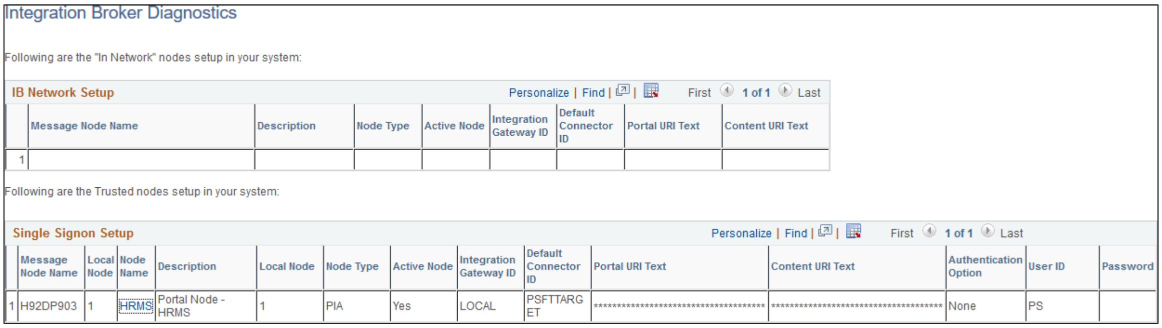
To view the Integration Broker Diagnostics page:

1. Select **PeopleTools** > **Portal** > **Configure Unified Navigation**.

2. In the Configure Unified Navigation center, click the **Unified Navigation Status** link to open the Unified Navigation Configuration Status page in the target area.

3. On the **Unified Navigation Configuration Status** page, select the **Diagnostics** page.

This example illustrates the nodes in the Integration Broker network and the trusted nodes in the single signon setup.



Field or Control	*Description*
Message Node Name	• In the IB Network Setup grid, the Message Node Name column displays all the active nodes in the unified navigation cluster. • In the Single Signon Setup grid, the Message Node Name column displays only the active default local nodes in the unified navigation cluster.
Node Name	In the Single Signon Setup grid, the Node Name column displays all the active nodes associated with each default local node in the unified navigation cluster. Click a node name to open the Node Definitions page where you can update to edit the *local* configuration of this node.
Portal URI Text	The Portal URI Text column displays the URI text for the portal servlet (psp) defined for the node. If you need to change this value, click the node name in the Single Signon Setup grid to edit the *local* configuration of this node.
Content URI Text	The Content URI Text column displays the URI text for the content servlet (psc) defined for the node. If you need to change this value, click the node name in the Single Signon Setup grid to edit the *local* configuration of this node.

Related Links

Defining All Participating Nodes

Other Troubleshooting Considerations

Custom Templates

Oracle delivers unified navigation with support for PeopleTools-delivered, iframe-based templates (for example, DEFAULT_TEMPLATE and PT_IFRAME_TEMPLATE) and WorkCenter templates.

Note: Unified navigation supports only iframe templates.

If your remote content references use custom templates or other PeopleSoft-delivered iframe templates, then those templates must conform with the following standards:

- The template HTML must include the PeopleSoft header and main menu pagelets:

  For example, to include the PeopleSoft header, use HTML code similar to the following:

  ```
  <div>
      <pagelet name="UniversalNavigation">
        <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IS⇒

  cript_UniHeader_Frame?c=%Bind(:5)" /> </pagelet>
    </div>
  ```

  In addition, to include the PeopleSoft main menu, use HTML code similar to the following:

  ```
   <div>
      <pagelet name="Nav">
        <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IS⇒

  cript_PT_NAV_INFRAME?navtype=dropdown&c=%Bind(:5)" />
      </pagelet>
    </div>
  ```

  Important! In both examples, the source element's node attribute *must* be set to LOCAL_NODE. Otherwise, unified navigation will fail to operate on the remote content reference.

- For all pagelets included in the template HTML, except for the PeopleSoft header and main menu pagelets, the source element's node must be set to the portal host node of the content provider system:

  For example, if the content provider system is PeopleSoft HRMS, set the source element's node to HRMS as shown in the following code:

  ```
  <div id="omtickerpagelet">
       <iframe name="NOTETICKER" id="NOTETICKER" frameborder="no" src="">
         <pagelet name="NoteTicker">
            <source node="HRMS" href="s/WEBLIB_OM.NOTE.FieldFormula.IScript_disp⇒

  layNoteTicker?OSEC=1&CSEC=1&PSEC=1&LSEC=1&ASEC=1"/>
         </pagelet>
       </iframe>
  </div>
  ```

The following HTML code represents a template that has been modified to adhere to the preceding standards. In this example, the relevant portions of the code are highlighted in bold:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/stric⇒

t.dtd">
<html dir="%Direction" lang="%LanguageISO">
<head>
<meta http-equiv="X-UA-Compatible" content="IE=8">
```

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">

</head>
<body class="PSPAGE" id="ptifrmtemplate">
  <div>
    <pagelet name="UniversalNavigation">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript⇒

_UniHeader_Frame?c=var" />
    </pagelet>
  </div>
  <div>
    <pagelet name="Nav">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript⇒

_PT_NAV_INFRAME?navtype=dropdown&c=var" />
     </pagelet>
    </div>
  <div id="ptifrmcontent">
    <div id="ptifrmtarget">
   <iframe id="ptifrmtgtframe" name="TargetContent" title="Target_Title" frameborde⇒

r="0" scrolling="auto" onload="ptrc.onLoadTC()" src="">
        <target Name="target" params="var"/>
      </iframe>
    </div>
  </div>
  <div id="omtickerpagelet">
     <iframe name="NOTETICKER" id="NOTETICKER" frameborder="no" src="">
       <pagelet name="NoteTicker">
          <source node="HRMS" href="s/WEBLIB_OM.NOTE.FieldFormula.IScript_displayNo⇒

teTicker?OSEC=1&CSEC=1&PSEC=1&LSEC=1&ASEC=1"/>
       </pagelet>
     </iframe>
  </div>
</body>
</html>
```

Errors When Accessing Remote Content Through Navigation Pages

When a remote content reference is accessed through a remote navigation page, a “not authorized” error can occur even though the user is authorized to access to the content reference on the content provider system. This error usually occurs because the portal registry definition for the remote content reference uses LOCAL_NODE, instead of a specific node. Under unified navigation, LOCAL_NODE will resolve to the portal host node on the portal system, and not a node on the content provider system. Because the content reference (component and page) does not exist on the portal system, the “not authorized” error results.

To resolve this issue, you must identify and update content references on the content provider system to use the portal host node of the content provider system. In the following example for a PeopleSoft FSCM application, a SQL statement updates the node value to ERP, the portal host node:

```
update PSPRSMDEFN set PORTAL_CNTPRV_NAM='ERP' where PORTAL_REFTYPE='C' and
PORTAL_CNTPRV_NAM='LOCAL_NODE' and PORTAL_NAME='EMPLOYEE';
```

See Completing Configuration of Remote Content.

Chapter 12

Working with Conditional Navigation

Understanding Conditional Navigation

With the introduction of PeopleSoft Fluid User Interface in PeopleTools, PeopleTools offers administrators the capability to render a page either in classic mode or fluid mode based on conditions and factors. Additionally, you can direct a user to a different content reference, instead of the default content reference, based on conditions and factors. Administrators can configure alternate content reference by specifying conditions. While generating the default content reference, the conditions for an alternate content reference, if specified, are evaluated at runtime.

For example, in an HCM system that is not configured for conditional navigation, when you click the classic Home and Mailing Address content reference (Self Service > Personal Information > Home and Mailing Address) the classic Home and Mailing Address page would be displayed to a user. With conditional navigation, you can configure the classic Home and Mailing Address content reference to point to a fluid equivalent of the page. So, you will see the fluid equivalent of the Home and Mailing Address page unless other conditions are met that force the display of the classic page. This example is a simple illustration of conditional navigation where a classic content reference is mapped to a fluid content reference.

PeopleSoft supports the following mapping of content references:

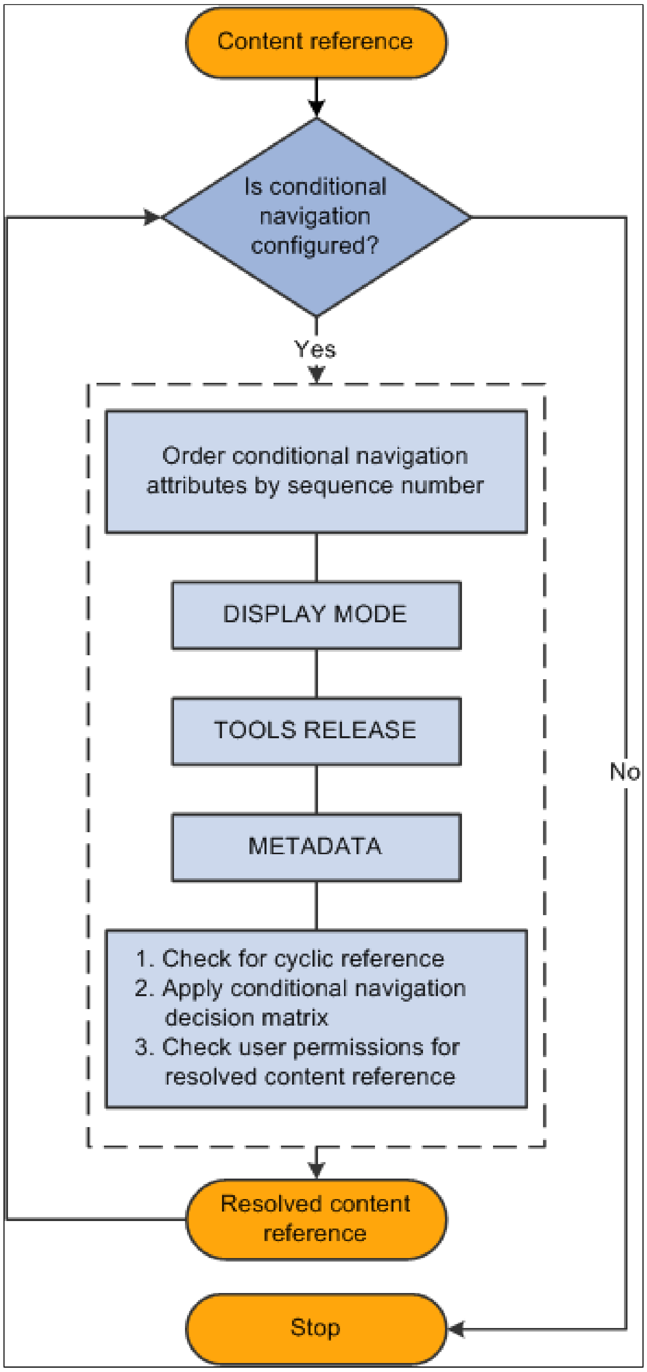
- One content reference mapped to one alternate content reference.
- One content reference mapped to multiple alternate content references.
- Multiple content references mapped to a single alternate content reference.

See Sample Scenarios of Conditional Navigation.

Conditional navigation is dependent on these factors:

- Content reference settings for classic or fluid mode.
- Conditional navigation configuration of the content reference.
- User access to the configured alternate content reference.

This example illustrates the process flow when conditional navigation is configured.



In a conditional navigation process flow, note the following:

- Multiple conditions or attributes can be configured for a content reference.
- Multiple conditions are evaluated based on their sequence number.
- Conditions with duplicate sequence numbers will not be evaluated.
- Upon successful evaluation of conditions, the system encounters a resolved content reference. The first successful condition will stop further evaluation of other conditions for this content reference.

- A resolved content reference may also be configured for conditional navigation, so in this case, the conditions configured in the resolved content reference are also evaluated.
- This process continues till the system does not encounter any other alternate content reference.

Configuring Conditional Navigation

Configuring conditional navigation involves configuring alternate content references. You use content reference system attributes to configure conditional navigation, and these system attributes are prefixed with CN_. You define the condition in the attribute name field, and you specify the alternate content reference in the attribute value field.

You can use the following content reference attributes:

- CN_DISPLAYMODE.
- CN_TOOLSREL.
- User defined content reference attribute.

Limitations

Note the following limitations when configuring conditional navigation:

- Activity guides and fluid navigation collections do not support conditional navigation.
- Do not create fluid alternate content references for WorkCenter based pagelets.

Configuring an Alternate Content Reference Using the CN_DISPLAYMODE Attribute

The CN_DISPLAYMODE_*N* attribute is used to navigate from a classic content reference to a fluid content reference and vice versa. Configuring this attribute ensures that when you access a content reference configured for display mode the appropriate content reference (classic or fluid) is automatically displayed based on the factors.

Note: When you use the CN_DISPLAYMODE_*N* attribute, ensure that you configure bidirectional alternate content references. That is, a fluid content reference must be configured to point to a classic content reference and a classic content reference must be configured to point to a fluid content reference.

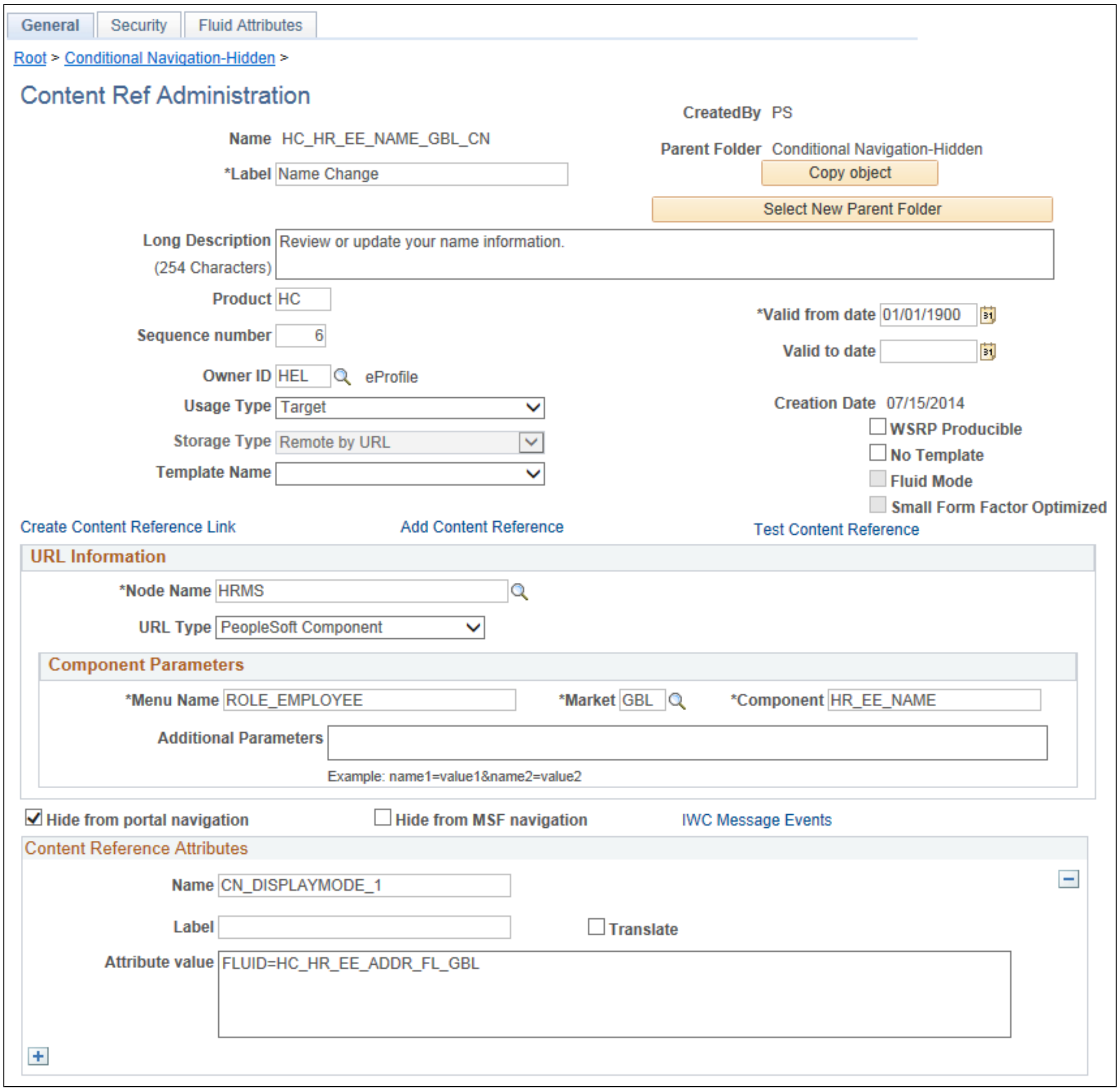
When you want to specify multiple conditions for a content reference, you can sequence the conditions by replacing *N* with a number.

The following table shows the attribute name and attribute-value syntax:

Attribute Name	***Attribute Value***
CN_DISPLAYMODE_*N*	KEY1=CREF

Use the Content Ref Administration page (**PeopleTools** > **Portal** > **Structure and Content** and click the Edit link of a content reference) to configure an alternate content reference by adding the required attributes in the Content Reference Attributes section.

This example illustrates the fields and controls on the Content Ref Administration - Content Reference Attributes section. You can find definitions for the fields and controls later on this page.



This table describes the fields that are associated with conditional navigation. For information on all the fields on the Content Ref Administration page, see Defining Content References.

Field or Control	***Description***
Additional Parameters	**Important!** If any additional parameters are defined, all of the additional parameters and only these additional parameters must be included in the request URL.
Name	Enter an attribute, for example, CN_DISPLAYMODE_1.

Field or Control	***Description***
Translate	Do *not* select the Translate check box. Conditional navigation does not support the Translate check box.
Attribute value	Enter the alternate content reference. For example, FLUID=HC_HR_EE_ADDR_FL_GBL.

Configuring an Alternate Content Reference

To configure an alternate content reference, do the following:

1. Open the content reference definition that you want to configure.
2. In the Content Reference Attributes section, enter CN_DISPLAYMODE_1 in the Name field.
3. Optionally, enter a label for the attribute in the Label field.
4. Do not select the Translate check box.
5. Enter an alternate content reference in the Attribute value field.

   For example: FLUID=HC_HR_EE_ADDR_FL_GBL.

   Note: You can add additional content reference attributes. When you add multiple content reference attributes, ensure that you enter sequence numbers for the attributes.

Sample Scenarios of Conditional Navigation

This topic discusses a few scenarios that involve configuring conditional navigation.

Case 1: Configuring Conditional Navigation in a Classic CREF that has an Alternate Fluid CREF

This sample scenario illustrates a case in which you are using the CN_DISPLAYMODE_N attribute to navigate from a classic content reference to the fluid equivalent content reference. You must configure the attribute in the classic and fluid content references as shown in the following example.

Example of CN_DISPLAYMODE_*N* attribute defined for a classic content reference:

Attribute Name	***Attribute Value***
CN_DISPLAYMODE_1	FLUID=HC_HR_EE_ADDR_FL_GBL

Example of CN_DISPLAYMODE_*N* attribute defined for a fluid content reference:

Attribute Name	*Attribute Value*
CN_DISPLAYMODE_1	CLASSIC=HC_HR_EE_ADDR_GBL

Case 2: Accessing Both Classic CREF as well as Fluid CREF from Menu Navigation

This sample scenario illustrates a case in which you want to navigate to both the classic content reference and the fluid content reference from the menu navigation. In conditional navigation, if you have a classic content reference and you configure it to point to a fluid equivalent, the classic CREF is converted to the fluid equivalent in all places including menu navigation. Therefore, menu navigation will no longer display the classic CREF. In this case, if you want to display both the classic and the fluid equivalent content references in menu navigation, PeopleSoft recommends that you create a copy of the classic content reference, but you must not configure conditional navigation on this new CREF.

Case 3: Configuring Conditional Navigation in a Classic CREF That has an Alternate Mobile Application Platform (MAP) Page

This sample scenario illustrates a case in which you have a MAP page for the approval process. For example, the MAP CREF name is HC_APPROVAL_PROMOTION_MAP. You want to configure this MAP CREF as an alternate content reference for a classic content reference. To implement this case, you should complete these steps:

Note: In the decision matrix, MAP pages are treated as fluid pages.

1. Open the classic approval content reference.
2. Configure the classic content reference with the MAP CREF as the alternate content reference.

Attribute Name	*Attribute Value*
CN_DISPLAYMODE_1	FLUID=HC_APPROVAL_PROMOTION_MAP

After configuring conditional navigation, when a user clicks the classic approval page, the MAP approval page will be displayed instead of the classic approval page.

Conditional Navigation Decision Matrix

The following decision matrix is for fluid mode operation only:

Note: As of PeopleTools 8.55, a PeopleSoft system operates in fluid mode only; classic mode is not available as a system setting.

Input CREF	*Alternate CREF Configuration*	*Resolved CREF*
Classic	Fluid	Fluid

Input CREF	***Alternate CREF Configuration***	***Resolved CREF***
Classic	Classic	Classic
Fluid	Fluid	Fluid
Fluid	Classic	Fluid
Fluid	Not configured	Fluid
Classic	Not configured	Classic

Consider the following regarding the decision matrix:

- In case multiple content references resolve to the same alternate content reference, duplicate links are generated, but the duplicate links within the context of a folder will be removed and a user will see only one alternate content reference.
- MAP pages are treated as fluid pages.

PeopleCode that Supports Conditional Navigation

The following PeopleCode items support conditional navigation:

Note: Application logic that uses these PeopleCode items for URL generation will automatically support conditional navigation and no change in the application logic is required. However, if the URL is created by other means, conditional navigation may not be supported.

- AbsolutePortalURL
- AbsoluteContentURL
- Transfer
- FindCREFByURL
- FindCREFByName
- GetAbsoluteContentURL
- GenerateComponentContentRelURL
- GenerateComponentContentURL
- GenerateComponentPortalRelURL
- GenerateComponentPortalURL
- GenerateComponentRelativeURL
- FindCRefLinkByName

- ViewURL
- GetURL
- GenerateHomepagePortalURL
- GenerateHomepageRelativeURL

Related Links

"OverrideConditionalNav" (PeopleCode Language Reference)

Additional Content Reference Attributes in Conditional Navigation

In addition to the CN_DISPLAYMODE_N attribute, PeopleSoft supports the following content reference attributes that can be used to configure alternate content references.

You can configure multiple attributes for a content reference, and you specify the order of evaluating attributes by adding a sequence number. For example, in a content reference, you could use CN_DISPLAYMODE_1, CNTOOLSREL_2 and so on.

CN_TOOLSREL_N

Applications may provide different content references for the same functionality based on the features supported by a PeopleTools version. For example, applications can provide a Change My Password content reference for 8.53 and another content reference for 8.54.

This table describes the attribute and the attribute value syntax:

Attribute Name	***Attribute Value***	***Recommendation***
CN_TOOLSREL_N	KEY1=CREF1[, KEY2=CREF2]*	KEY value should match the output of %ToolsRel meta-variable. However, if the %ToolsRel meta-variable returns, for example, 8.54.09, the KEY value of 8.54 will be valid because internally the system recognizes this condition.

Using the example of two implementations of Change My Password where CREF1 points to the 8.53 implementation and CREF2 points to the 8.54 implementation, the CN_TOOLSREL content reference attribute can be configured with the following values:

Attribute Name	***Attribute Value***
CN_TOOLSREL_1	8.53=CREF1, 8.54=CREF2

User-Defined Content Reference Attribute

In addition to the delivered content reference attributes, PeopleSoft allows you to create user-defined conditional navigation attributes using certain system meta-HTML meta-variables.

This table describes the attribute and the attribute value syntax:

Attribute Name	***Attribute Value***	***Recommendation***
CN_METAVAR1[_METAVAR2]*_*N*	KEY1=CREF1[, KEY2=CREF2]*	At runtime, the resolved attribute name should match one of the configured values. The attribute values should exactly match the output of the system variable resolved attribute name.

When you create a conditional navigation attribute using a meta-HTML meta-variable, you need to follow these guidelines for the attribute syntax:

- The attribute name must begin with CN_ as a prefix.
- Use a meta-variable as a condition that needs to be satisfied, for example, add a system meta-variable such as BROWSER.
- Enter the meta-variable name in all caps without the leading %. For example, the %Browser meta-variable is entered as BROWSER.
- Use “_” as a separator when you want to include multiple system variables. Internally, the system converts the “_” separator to “%” at runtime.
- You can repeat the meta-variable multiple times.
- Replace *N* with a number that is used for sequencing when multiple content reference attributes are used. This value is *required*.

This table shows sample values for a user-defined attribute. In this example, the alternate content reference will be resolved only when the browser is Firefox and the version is 32. For all other browsers and for all Firefox versions except 32, the original content reference will be displayed.

Attribute Name	***Attribute Value***
CN_BROWSER_BROWSERVERSION_1	FIREFOX32.0=PT_PSPMUMTESTINFO_GBL

Only the following meta-variables are supported to configure an alternate content reference:

- %AppsRel
- %Browser
- %BrowserVersion

- %DBName
- %DBType
- %ToolsRel

Related Links

“Meta-HTML Reference” (PeopleCode Language Reference)

Overriding Conditional Navigation

In certain cases, you may want to override conditional navigation that is configured for a content reference.

You can also override conditional navigation during pre processing and during post processing. The following table describes pre-processing and post-processing in conditional navigation:

Field or Control	Description
Pre-processing	Pre-processing refers to the execution of code at the time of page or URL generation at the application server layer. That is, the factors required for conditional navigation and the conditions specified for alternate content references are evaluated at the time of page or URL generation.
Post-processing	Post-processing refers to the evaluation of factors and conditions for a given content reference at the web server layer. For example, when a user receives an email notification and clicks on the link, the factors and conditions are evaluated at the web server layer to determine whether a classic page or a fluid page should be displayed. **Important!** Post-processing is performed on URLs that use the psp (portal) servlet only. No post-processing is performed on URLs that use the psc (content) servlet. Therefore, ensure that any URLs provided to users are constructed using psp and not psc.

Overriding Conditional Navigation at the Component Level

Conditional navigation must not be applicable for configuration pages that are used by administrators, for example, Structure and Content, Tree Manager, Query Drill Down. You can override conditional navigation at the component level by using the OverrideConditionalNav built-in function.

```
/* The default value is %CNAV_SkipAll. */
OverrideConditionalNav();
```

Note: This built-in function must be invoked as the first executable line in the component PreBuild event.

Overriding Conditional Navigation During Pre Processing

You can also use the OverrideConditionalNav built-in function to suppress conditional navigation processing by other PeopleCode operations. The following code excerpt illustrates how you can override conditional navigation for a particular code segment rather than for the life of the program.

In the following code snippet, conditional navigation is ignored by the GenerateComponentPortalURL function:

```
OverrideConditionalNav(%CNAV_SkipAll);
&URL = GenerateComponentPortalURL(%Portal,%Node,MenuName. "ROLE_MANAGER", "GBL", Co⇒

mponent."HR_PROMOTE_APPR","","");
OverrideConditionalNav(%CNAV_RestoreAll);
SendEmail(&URL);
```

Note: You must re-enable conditional navigation for the remainder of the program's execution.

Overriding Conditional Navigation During Pre Processing and Post Processing

Certain circumstances require that conditional navigation be overridden both during pre-processing and at some future point whenever a user accesses a URL. To ensure that the generated URL always skips the specified conditional navigation settings, you can post-process this URL to append the skipcnav query string parameter.

The following example temporarily disables conditional navigation to generate a URL to a classic component. Then, conditional navigation is re-enabled for the remainder of the program's execution. However, conditional navigation could be invoked at a later date whenever the generated link is clicked. Therefore, post-processing of the generated URL appends the skipcnav query string parameter to the URL to ensure that conditional navigation is always ignored for every user who clicks the link in every situation.

```
/* Disable conditional navigation before generating the URL. */
OverrideConditionalNav();
&URL = GenerateComponentPortalURL(%Portal, %Node, MenuName."ROLE_MANAGER", "GBL", C⇒

omponent."HR_PROMOTE_APPR", "", "");

/* IMPORTANT: Re-enable conditional navigation */
OverrideConditionalNav(%CNAV_RestoreAll);

&URL = &URL | "?Action=U&BUSINESS_UNIT=US100&REQ_ID=0000000067";
/* Post-processing ensures that CN is skipped when the user clicks the link.  */
&URL = &URL | "&skipcnav=1";

/* Invoke MCFOutboundEmail.Send to send the email including the generated URL */
...
```

The skipcnav query string parameter takes the same integer values that are valid for the *mode* parameter of the OverrideConditionalNav function. This includes the ability to combine two values by using their sum as the integer value. See the *mode* parameter of the OverrideConditionalNav function for more information.

Related Links

"OverrideConditionalNav" (PeopleCode Language Reference)

Working with Other PeopleTools Features in Conditional Navigation

This topic describes how search results and related content are handled in the conditional navigation feature.

Working with Search Results in Conditional Navigation

The conditional navigation feature takes into consideration the links that are displayed when a user uses global search. If conditional navigation is configured, the content references displayed on clicking the links on the Search Results page would be as per the conditional navigation guidelines on the search feature.

Consider the following scenario to understand how conditional navigation is implemented for classic display mode and fluid display mode.

- In Classic A content reference, Fluid A is configured as an alternate content reference.

  Classic A is indexed, but not Fluid A.
- Classic B content reference is not configured for any alternate content reference.

  Classic B is indexed.
- Fluid C content reference is not configured for any alternate content reference.

  Fluid C is indexed.

For example, a user searches on the word “Employee” and assume that the search hits are on Classic A, Classic B, and Fluid C.

User Searches in Fluid Display Mode on Any Device

User sees three search results: Fluid A, Classic B, and Fluid C.

Fluid A content reference is displayed rather than Classic A because the user is using fluid display mode.

User Searches in Classic Display Mode on Any Device

Users sees two search results: Classic A and Classic B.

Classic A content reference is displayed rather than Fluid A because the user is using classic display mode.

Working with Related Content in Conditional Navigation

The conditional navigation feature supports the related content features.

Authorization Service

In a system that is not configured for conditional navigation, the related content framework checks for authorization at the level of the input content reference, but in a system that is configured for conditional navigation, the authorization is checked at the level of resolved content reference.

Note: In a system configured for conditional navigation, you must invoke the authorization service for alternate content references.

See "Understanding Developing and Invoking the Security Authorization Service" (Security Administration) for invoking authorization service.

Suppressing Duplicate Related Actions

Conditional navigation supports the mapping of multiple content references to a single content reference. In such a scenario, related actions configured for these content references will appear multiple times because of the mapping to a single content reference. In conditional navigation, duplicate related actions are suppressed within the context of a folder for fluid.

For example, if related actions are configured on a fluid page for three classic content references — ClassicCREF1, ClassicCREF2, and ClassicCREF3 — and if these three content references are mapped to a single fluid content reference (NUICREF1), in fluid mode, the duplicate related actions are suppressed and the related actions menu will display NUICREF1 only once in the context of a folder. In classic mode, the duplicate related actions are not suppressed and are shown as ClassicCREF1, ClassicCREF2, and ClassicCREF3.

Passing Service URL Parameters to Alternate Content References

Note: The following information is applicable to fluid mode only.

When you map multiple classic content references with related actions configured to a single fluid content reference, the Service URL Parameters of the first content reference are passed to the mapped fluid content reference. For example, three classic content references — ClassicCREF1, ClassicCREF2, and ClassicCREF3 — with configured related actions are mapped to a single fluid content reference (NUICREF1), by default, the Service URL Parameters of ClassicCREF1 are passed to NUICREF1.

If you want to pass the Service URL Parameters of ClassicCREF2 or ClassicCREF3 to NUICREF1, you can use the PTCS_SUPPRESS service parameter. However, multiple related actions with the same label will be displayed. To avoid this and display the service label instead of the content reference label, you can configure an additional service parameter, PTCS_LABELTYPE.

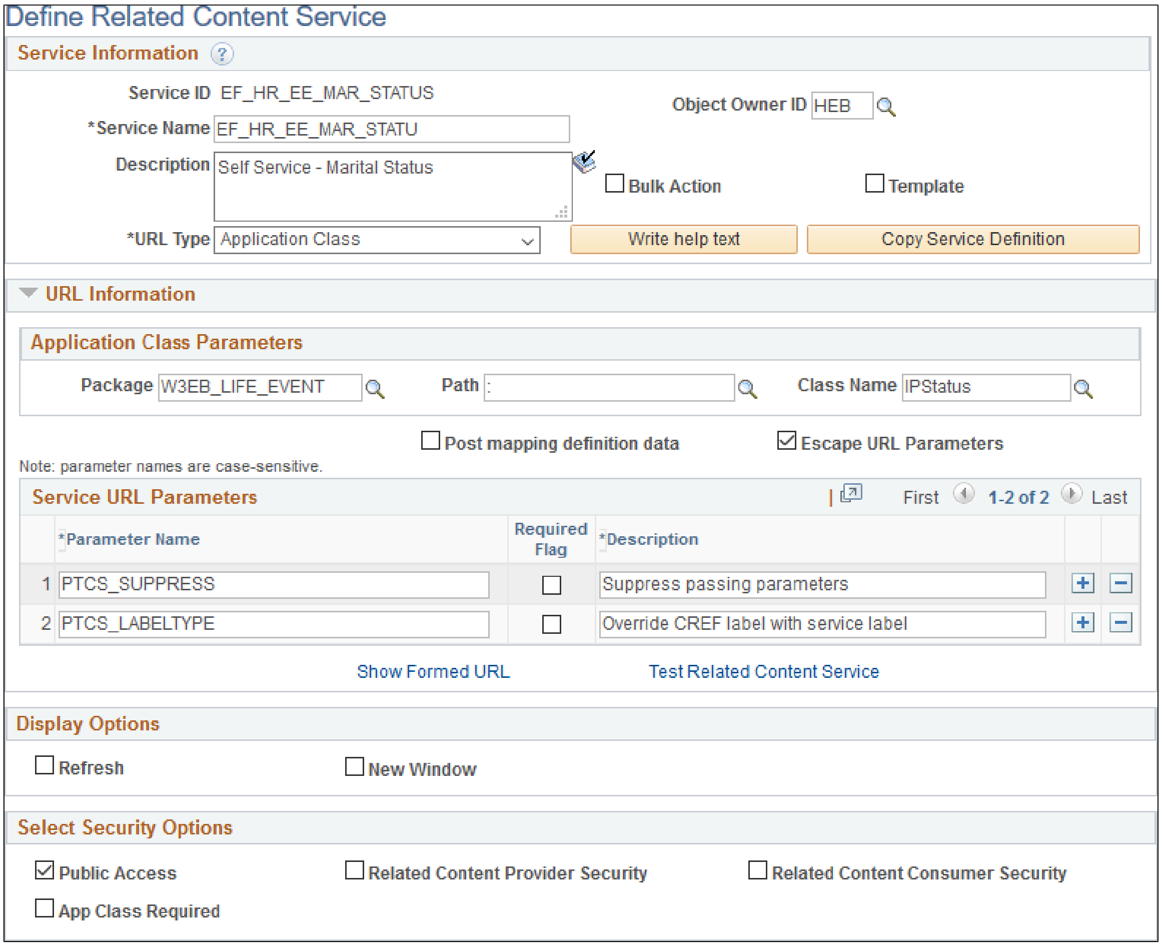
To define and configure the related content service:

1. Open the service definition that you want to modify, for example, EF_HR_EE_MAR_STATUS (**PeopleTools** > **Portal** > **Related Content Service** > **Define Related Content Service**).
2. In the **Service URL Parameters** group box:
   a. Enter *PTCS_SUPPRESS* in the **Parameter Name** field.
   b. Enter a description.
3. In the **Service URL Parameters** group box:

a. Enter *PTCS_LABELTYPE* in the **Parameter Name** field.

b. Enter a description.

This example illustrates adding two service parameters to the service definition.



4. Save the service definition.

5. Edit the service configuration of the modified service definition (**PeopleTools** > **Portal** > **Related Content Service** > **Manage Related Content Service**).

6. On the Configure Service page, for the PTCS_SUPPRESS parameter:

   a. Select *Fixed Value* in the **Mapping Type** column.

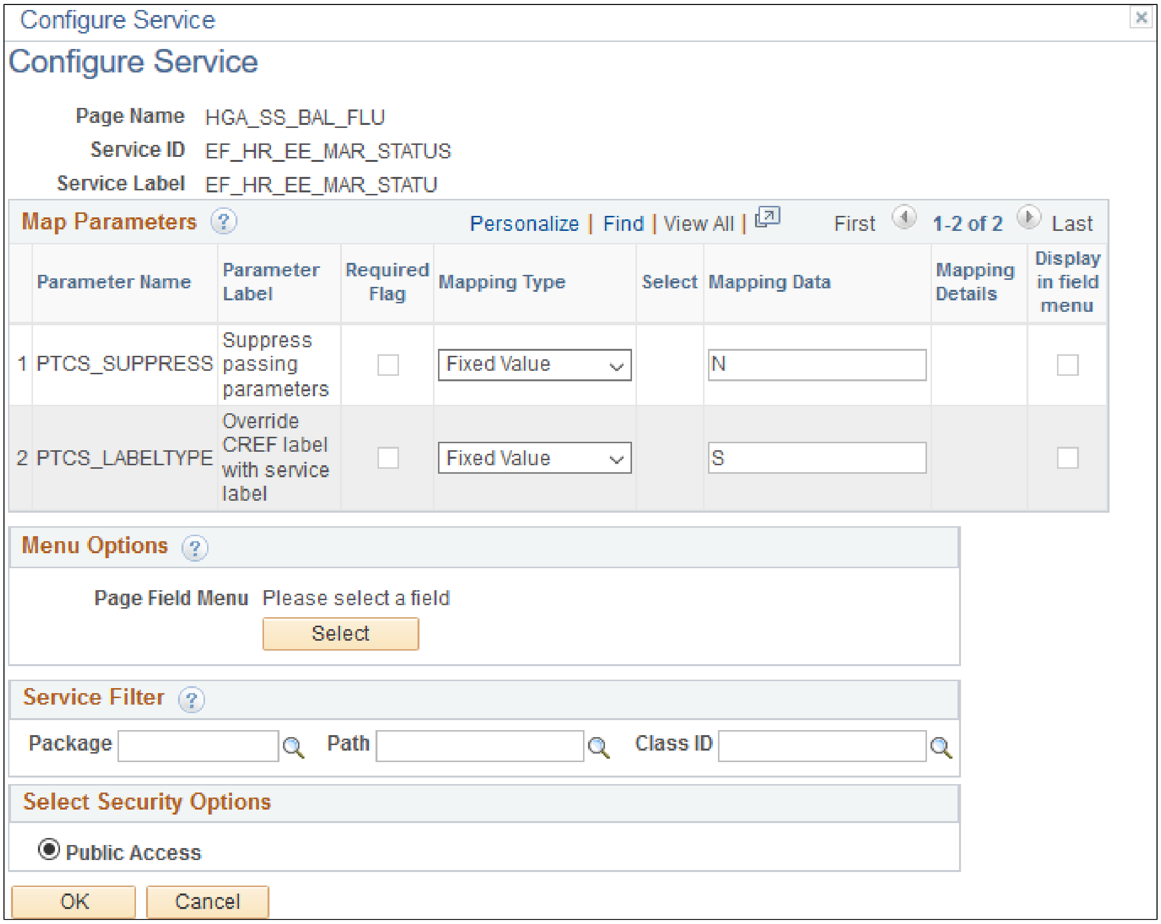
   b. Enter *N* in the **Mapping Data** column.

7. For the PTCS_LABELTYPE parameter:

   a. Select *Fixed Value* in the **Mapping Type** column.

   b. Enter *S* in the **Mapping Data** column.

This example illustrates configuring the PTCS_SUPPRESS service.



8. Save the configuration.

Related Links

Creating and Managing Related Content Service Definitions

Managing Related Content Configurations and Data

Configuring Related Content, Related Actions, and Menu and Frame Layouts

Troubleshooting Conditional Navigation

To ensure that you have configured conditional navigation correctly on your system, check the following:

- Verify whether content reference attributes for conditional navigation are correctly defined on the Content Ref Administration page.
- Ensure that the request URL has been constructed using the psp (portal) servlet.

Important! Post-processing is performed on URLs that use the psp (portal) servlet only. No post-processing is performed on URLs that use the psc (content) servlet. Therefore, avoid using URLs that use the psc servlet since they will be accessed as-is and will fail to reflect any conditional navigation that has been configured.

- Ensure that the request URL has been constructed with any and all of the additional parameters defined for the content reference on the Content Ref Administration page. If any additional parameters are missing or if there are extra additional parameters, the system will fail to identify the actual content reference and conditional navigation will fail.
- Verify whether authentication permissions have been configured for alternate content references. If permissions are not configured, conditional navigation will not resolve to the expected content reference and will evaluate the next condition.

  These logs are available in application server logs.
- Verify cyclic references.

  You can configure alternate content references in such a way that it forms a loop. In these cases, conditional navigation identifies cyclic references, breaks the loop, and returns the original content reference.
- Verify whether the OverrideCNavDisplayMode PeopleCode function is used.
- Verify whether the classic or fluid property for the content reference is selected.

  You may want to check this property because it is possible that the component is marked as fluid, but the content reference is not marked as fluid.
- Because the classic drop-down menu might be cached on the web server depending on your web profile configuration, you may need to clear the web server's cache to test conditional navigation.

  Note: The Navigator menu displayed in the NavBar is never cached on the web server.

- Use the application server logging to trace conditional navigation. Set LogFence = 4 to trace conditional navigation.

  Logs specific to conditional navigation are prefixed with COND_NAV.

  Sample conditional navigation log trace:

This example illustrates a sample conditional navigation trace log in application server logs.

```
COND_NAV:DisplayMode - Checking for CREF[HC_GP_ABS_MGRSS_BAL_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HGA_SS_BAL_FLU_MSS_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HGA_SS_BAL_FLU_MSS_GBL]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HGA_SS_BAL_FLU_MSS_GBL]

COND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.GP_ABS_MGRSS_BAL.GBL?
PAGE=GP_ABS_SS_BALANCES]
COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/HGA_MANAGER_FL.HGA_SS_BAL_FLU.GBL?
MGR=Y&PAGE=GP_ABS_SS_BALANCES]

COND_NAV:DisplayMode - Checking for CREF[HC_GP_ABS_MGRSS_HIST_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HGA_SS_REQHIST_FLU_MSS_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HGA_SS_REQHIST_FLU_MSS_GBL]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HGA_SS_REQHIST_FLU_MSS_GBL]

COND_NAV:DisplayMode - Checking for CREF[PTSF_GBLSRCH_FLUID_GBL]
COND_NAV:DisplayMode - Alternate CREF[PTSF_GLOBAL_SEARCH_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use Original CREF[PTSF_GBLSRCH_FLUID_GBL]


COND_NAV:DisplayMode - Alternate CREF[PTSF_GLOBAL_SEARCH_GBL] found for [CN_DISPLACOND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.GP_ABS_MGRSS_HIST.GBL?
PAGE=GP_ABS_SS_REQHIST]
COND_NAV:INFO Decision Matrix result - Use Original CREF[PTSF_GBLSRCH_FLUID_GBL]COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/HGA_MANAGER_FL.HGA_SS_REQHIST_FLU.GBL?
MGR=Y&PAGE=GP_ABS_SS_REQHIST]
COND_NAV:DisplayMode - Checking for CREF[HC_GP_ABS_MGRSS_REQ_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HGA_SS_REQ_FLU_MSS_GBL] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HGA_SS_REQ_FLU_MSS_GBL]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HGA_SS_REQ_FLU_MSS_GBL]
COND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.GP_ABS_MGRSS_REQ.GBL?
PAGE=GP_ABS_SS_REQUEST]
COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/HGA_MANAGER_FL.HGA_SS_REQ_FLU.GBL?
MGR=Y&PAGE=GP_ABS_SS_REQUEST]
COND_NAV:DisplayMode - Checking for CREF[HC_HR_MGR_FULL_PART_GBL_CN]
COND_NAV:DisplayMode - Alternate CREF[HC_HR_MSS_CT_FL_GBL_1] found for [CN_DISPLAYMODE_1]
COND_NAV:INFO Decision Matrix result - Use resolved CREF[HC_HR_MSS_CT_FL_GBL_1]
COND_NAV:INFO - Alternate CREF Resolved is [HC_HR_MSS_CT_FL_GBL_1]
COND_NAV:Entry. Input URL:
[http://myserver.example.com:8000/psp/h92jddocx_newwin/EMPLOYEE/HRMS/c/ROLE_MANAGER.HR_MGR_FULL_PART.GBL?
PAGE=HR_MGR_FULL_PART]
COND_NAV: Resolved:  Node Name - [HRMS] and URL -
[http://myserver.example.com:8000/psc/h92jddocx_newwin/EMPLOYEE/HRMS/c/EL_MANAGER_FL.HR_MSS_CT_FL.GBL?
TRANS_NAME=HR_FULL_PART_STATUS&PAGE=HR_MGR_FULL_PART]
```

Chapter 13

Administering Homepages and Dashboard Pages

Understanding Homepages and Dashboard Pages

Classic homepages and dashboards are PeopleSoft pages that aggregate and display pagelets that share some common or similar purpose. Tiles are the PeopleSoft fluid user interface equivalent of pagelets. Therefore, similar to classic homepages and dashboards, fluid homepages and dashboards aggregate tiles that share some common or similar purpose. Homepages and dashboards provide quick access to useful information by presenting concise but feature-rich pagelets or tiles to the user. These pagelets and tiles can vary in function and complexity, enabling users to see not only an overview, but to drill into the data to access more granular and detailed information.

In these topics, the terms *homepage* and *tabbed homepage* are used interchangeably. In addition, the terms *dashboard* and *dashboard page* are used interchangeably.

A fluid homepage or dashboard can contain sections. A section is a collection of tiles under a categorization label that generalizes the purpose of the tiles in that section.

The term *standard* homepage or dashboard is used to refer to fluid homepages or dashboards without sections. The term *standard* is also used to refer to the area at the top of a sectionalized homepage or dashboard where tiles that are not assigned to any sections are placed.

Note: The labels of system-defined homepages and dashboards are translatable; however, personalized labels of homepages and dashboards are not translatable—this applies to personal fluid homepages as well as the personalized labels of fluid homepages, fluid dashboards, classic homepages and classic dashboards.

Related Links

Understanding Pagelets
Understanding Pagelet Wizard
Creating and Maintaining Tile Definitions Manually
Using Tile Wizard
(Fluid only) Creating Sections on Fluid Homepages

Homepages

Note these points about homepages:

- Homepages appear immediately after users sign in to PeopleSoft applications or when the user clicks the **Home** link or button.
- PeopleSoft applications can include definitions for one or more homepages.

If multiple classic homepages are defined, each displays beneath the portal header as a tab. On a fluid homepage, additional homepages do not display as tabs, but are available from a drop-down selector at the top of the page.

- If you enable personalization, users can personalize homepages. The system stores each user's homepage personalizations in the PSPRUHDEFN (which is updated when a user personalizes the homepage), PSPRUHTAB, and PSPRUHTABPGLT tables.

You manage homepage definitions by using the Structure and Content component. Rarely should anyone other than the portal administrator have access to Structure and Content since access to this component gives any user unrestricted access to the entire portal registry. The location in the portal registry differs between classic and fluid homepage definitions:

- Classic homepage definitions reside in the Portal Objects > Homepage > Tabs folder.
- Fluid homepage definitions reside in the Fluid Structure Content > Fluid Homepages folder.

Dashboards

Note these points about dashboard pages:

- Dashboards are a type of homepage that have been designated as *operational dashboards* through a content reference attribute.
- Dashboards do not appear automatically after sign in; instead, you access a dashboard by clicking a content reference link, which you can configure to appear anywhere in the menu structure.
- PeopleSoft applications can have multiple dashboards.
- If you enable personalization, users can personalize dashboards. Similar to homepages, the system stores each user's dashboard personalizations in the PSPRUHDEFN (which is updated when a user personalizes the dashboard), PSPRUHTAB, and PSPRUHTABPGLT tables.
- WorkCenter dashboards are a special class of dashboards that can be deployed as target content in a WorkCenter.

  Note: WorkCenter dashboards are available on classic WorkCenters only.

Unlike homepages, to create and manage dashboards, you do *not* need access to the Structure and Content component. You can use the Manage Dashboard Pages component. No access to the portal registry is necessary. By eliminating the need to access the portal registry to manage dashboard pages, your PeopleSoft application offers increased flexibility of administration and decreased chance of down time due to portal registry mismanagement errors.

Note: When you use the Manage Dashboard Pages component to create new dashboard pages, the system automatically registers the dashboard as a homepage in the portal, places the content reference in the correct folder, and selects the **Operational Dashboard** check box. No functionality has been removed from the Structure and Content pages; you can also use them to manually create and manage definitions that you specify as dashboards.

Fluid Homepages and Dashboards

Fluid homepages and dashboards share many characteristics of their classic counterparts:

- They are equivalent in function to their classic counterparts.
- They are managed in the same manner as their classic counterparts.
- Their definitions stored in the portal registry accessed through the Structure and Content component.

Note: Fluid homepages can be configured as required or optional; classic homepages cannot be.

Summary of Features Available to Homepages and Dashboards

The following table summarizes which features are available on classic homepages, classic dashboards, fluid homepages, and fluid dashboards.

Feature	***Classic Homepages***	***Classic Dashboards***	***Fluid Homepages***	***Fluid Dashboards***
Display pagelets, including remote pagelets	Y	Y	N	N
Display tiles, including remote tiles	N	N	Y	Y
Organize tiles into named sections	N	N	Y	Y
Support administrator configuration	Y	Y	Y	Y
Allow user personalization	Y	Y	Y	Y
Allow administrator to disable user personalization	N	N	Y	Y
Support access to similar definitions (homepages or dashboards) directly from the current definitions	Y	N	Y	N
Support enhanced tile functionality (larger tile sizes, autosize, resizable, and interactive tiles)	N	N	Y	Y
Support merged definitions across a cluster	Y	N	Y	N

Managing Homepages

This section discusses how to:

- Create and configure homepage definitions.

- Set homepage permissions.
- Specify the pagelets or tiles that can appear on a homepage.
- (Fluid only) Create sections on fluid homepages.
- (Classic only) Arrange pagelets on classic homepages.
- (Fluid only) Arrange tiles on fluid homepages.
- (Fluid only) Publish additional tiles to already personalized homepages (using the Tile Publish page or using a PeopeCode program).
- Specify the homepage sequence.
- (Fluid only) Publish personalizations as a new fluid homepage.
- (Fluid only) Disable user personalization on fluid homepages.
- (Fluid only) Restrict fluid homepage and NavBar personalization.

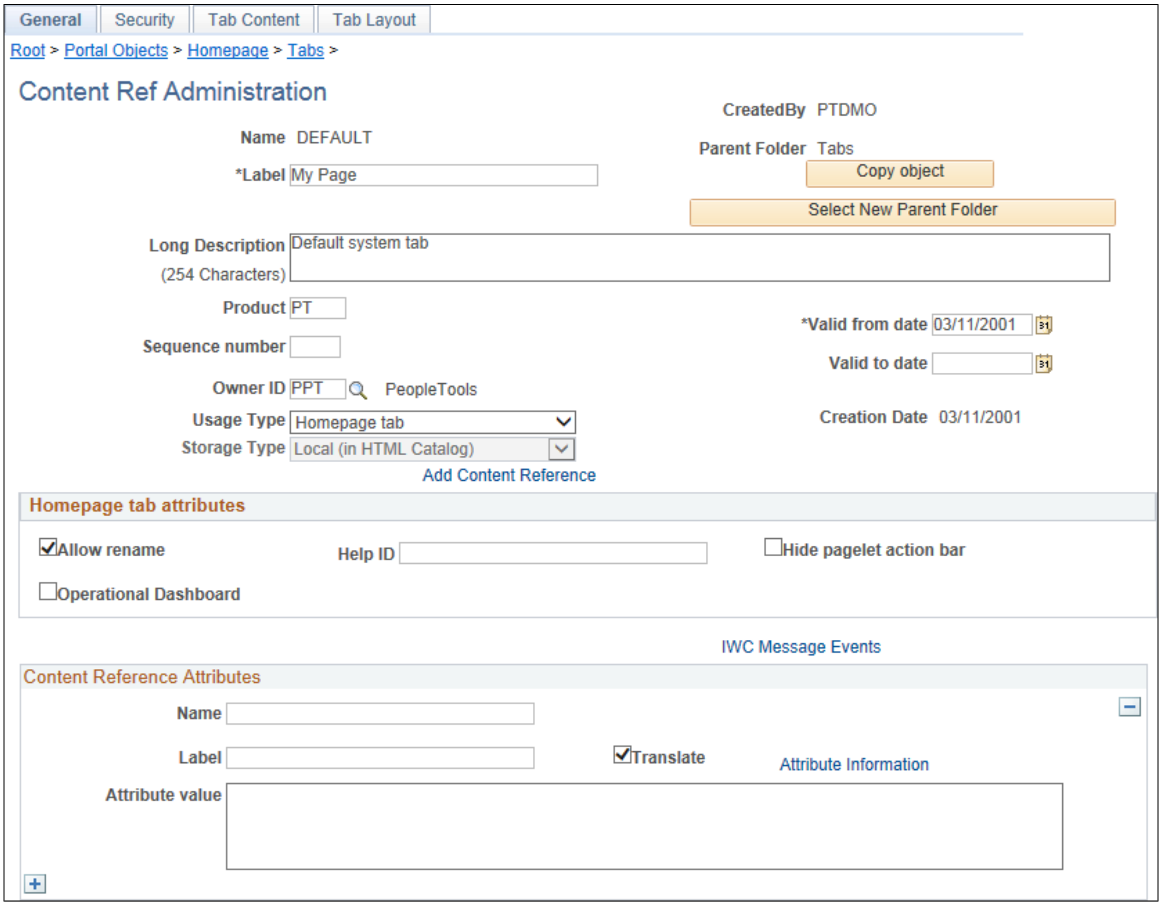
Creating and Configuring Homepage Definitions

Use the Content Ref Administration page (PORTAL_CREF_ADM) to create and maintain homepage definitions.

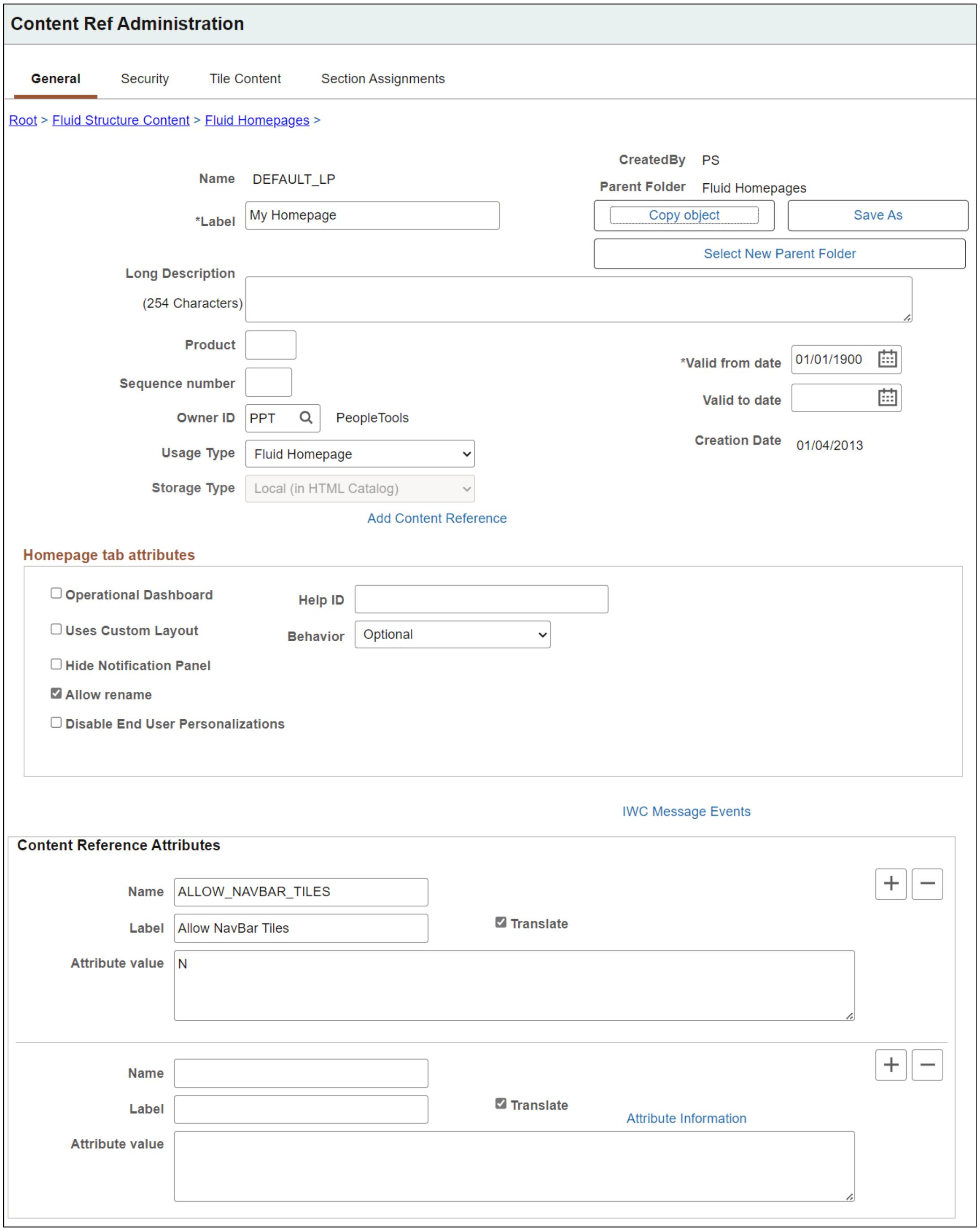
Navigation:

1. Select **PeopleTools** > **Portal** > **Structure and Content**.
2. Continue to navigate through the folder structure depending on the homepage type:
   - For classic homepages, select **Portal Objects** > **Homepage** > **Tabs**.
   - For fluid homepages, select **Fluid Structure Content** > **Fluid Homepages**.
3. Click the Add Content Reference link.

This example illustrates the fields and controls on the Content Ref Administration page for a classic homepage definition. You can find definitions for the fields and controls later on this page.



This example illustrates the fields and controls on the Content Ref Administration page for a fluid homepage definition. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Name	Enter a unique ID for the homepage definition. The value that you specify in the **Name** field can consist of up to 50 letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.

Field or Control	***Description***
Label	Enter the label that users will see for the homepage.
Long Description	Enter a long description for the homepage.
Product	Portal-aware applications can use this field for group processing. All PeopleTools objects have the product name *PT*.
Sequence number	Enter a value to designate the order of the homepage when multiple homepages are defined. The sequence number determines the order in which the homepages appear. The homepages appear with the lowest sequence number first. If any numbers are identical, then the order is alphanumeric. No value is equivalent to specifying *0*. **Important!** Therefore, the homepage with the lowest sequence number becomes the default homepage.
Save As	The Save As button is not available in Add mode. The button appears after the homepage CREF is saved. Use the Save As button to clone the delivered homepage or dashboard. It provides administrators a starting point to build upon the delivered definition. On saving the cloned homepage, administrators can navigate to the new definition.
Valid from date	Enter the date that the homepage becomes visible to users.
Valid to date	Enter an optional date that the homepage is removed from use.
Owner ID	Select the object owner ID.
Usage Type	Select a usage type depending on the definition: • *Homepage tab* for classic homepage definitions. When you select *Homepage tab,* two additional pages appear in the component: the **Tab Content** and **Tab Layout** pages. • *Fluid Homepage* for fluid homepage definitions. When you select *Fluid Homepage*, an additional page appears in the component: the **Section Assignments** page.
Storage Type	Displays *Local (in HTML Catalog),* which cannot be modified.

Note: The ALLOW_NAVBAR_TILES content reference attribute found on some fluid homepage definitions is not a valid attribute and is ignored.

Homepage Tab Attributes

Field or Control	*Description*
Uses Custom Layout	Select this setting when you create an advanced sectionalized homepage that uses the new model. This setting prohibits the adaptive changes performed by the system when running on a small form factor device making the responsibility of the custom layout to handle all visual changes for the small form factor device. Do not use this setting with pre-8.62 fixed tile size homepages or dashboards. For more information see Configuring Advanced Sectionalized Homepage and Dashboard.
Hide Notification Panel	Select to hide the notifications panel for all users when this homepage is displayed. You may use this option to provide more area to work with when laying out sections on a homepage. **Note:** The Notifications panel is displayed by default on fluid homepages unless this setting has been configured differently for this homepage.
Allow rename	Select to enable the user to personalize the homepage label. When you select the **Disable End User Personalizations** check box, the Allow rename check box is automatically unavailable for entry. **Note:** When a portal administrator updates the label for a fluid homepage or a fluid dashboard, the PeopleSoft system propagates the new label to homepages or dashboards that have not been personalized and to personalized homepages or dashboards that also retain the original label. If a user has personalized the label, then the user's personalization is retained.

Field or Control	***Description***
Help ID	Enter the help ID for the homepage. When users click the Help link, the PeopleSoft system opens a new window displaying online help documentation for this item. **Note:** You must also configure a help URL on the Web Profile Configuration - General page.
(Classic only) **Hide pagelet action bar**	Select to hide the pagelet title bars on all pagelets on the homepage tab. This option overrides all properties for pagelets used on this tab.
Operational Dashboard	Select to use the homepage as a dashboard page, which hides the homepage so that it does not appear when users sign in. **Important!** If you use the Dashboard Administration pages to create dashboards, then the system creates the homepage content reference, selects this check box, and creates a content reference link automatically. However, if you manually select this check box, you must manually create a content reference link to the dashboard, which you position in the menu structure.
(Fluid only) **Disable End User Personalizations**	Select to disable end-user personalizations for the homepage. When you select this check box, the **Allow rename** check box is automatically unavailable for entry because modifying the homepage label is an end-user personalization. When you create a section on a fluid homepage, this option is automatically selected. For more information on using this check box, see (Fluid only) Disabling User Personalization for Homepages.

Field or Control	***Description***
(Fluid only) **Behavior**	Select one of the following: • *Optional* – The homepage is available to all users that have permissions to the homepage, but it is not visible by default; users can add, move, or delete the homepage in their list of available homepages. • *Optional - Default* – The homepage is visible to all users by default for users that have permissions to the homepage; users can move or delete the homepage in their list of available homepages. • *Required* – The homepage is visible to all users by default for users that have permissions to the homepage; users can move the homepage in their list of available homepages; however, they cannot delete it. • *Required - Fixed* – This option is equivalent to *Required.* The fixed option is ignored.

Related Links

Managing Dashboard Pages

Setting Homepage Permissions

Use the Content Reference Security page (PORTAL_CREF_SEC) to define the portal registry security for the homepage definition.

Navigation:

From the Content Ref Administration page, select the Security page tab.

This example illustrates the fields and controls on the Content Reference Security page (a classic homepage definition is shown). You can find definitions for the fields and controls later on this page.

Field or Control	***Description***
Public	Select to make the homepage available to all users. When this check box is selected, all other fields except **Label** are hidden.
Author Access	Select to indicate that the creator of the homepage definition can access the homepage in the future, regardless of how other permissions are set.

Security Authorizations

You can base homepage security on roles or permission lists. You can specify any combination of roles and permission lists that allow homepage access.

Field or Control	***Description***
Type	Select from these types: • *Permission List:* Enables access for users based on permission list membership. You then specify the permission lists that should have access to this homepage. • *Role:* Enables access for users based on role membership. You then specify the roles that should have access to this homepage.
Name	Select the permission list or role that you want have access to this homepage.

Field or Control	Description
View Definition	Click to access the security administration component for the permission list or role specified on the current row.

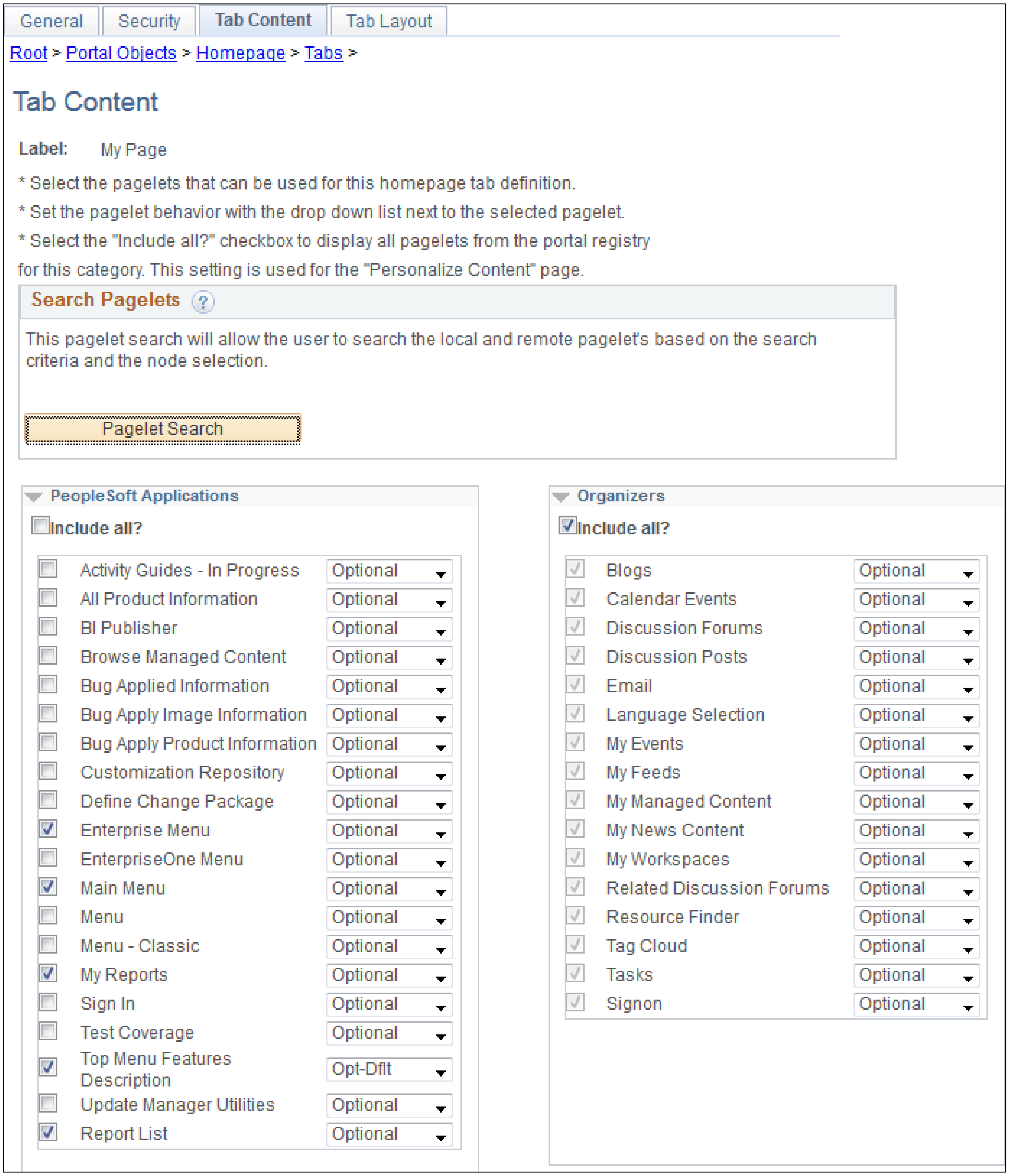
Selecting the Pagelets or Tiles That Appear on Homepages

Use the Tab Content - Tab Content page (PORTAL_TAB_CON_ADM) or the Tile Content: Tab Content page (PORTAL_TAB_LANDING) to select which pagelets or tiles are available to users who have access to this homepage.

Navigation:

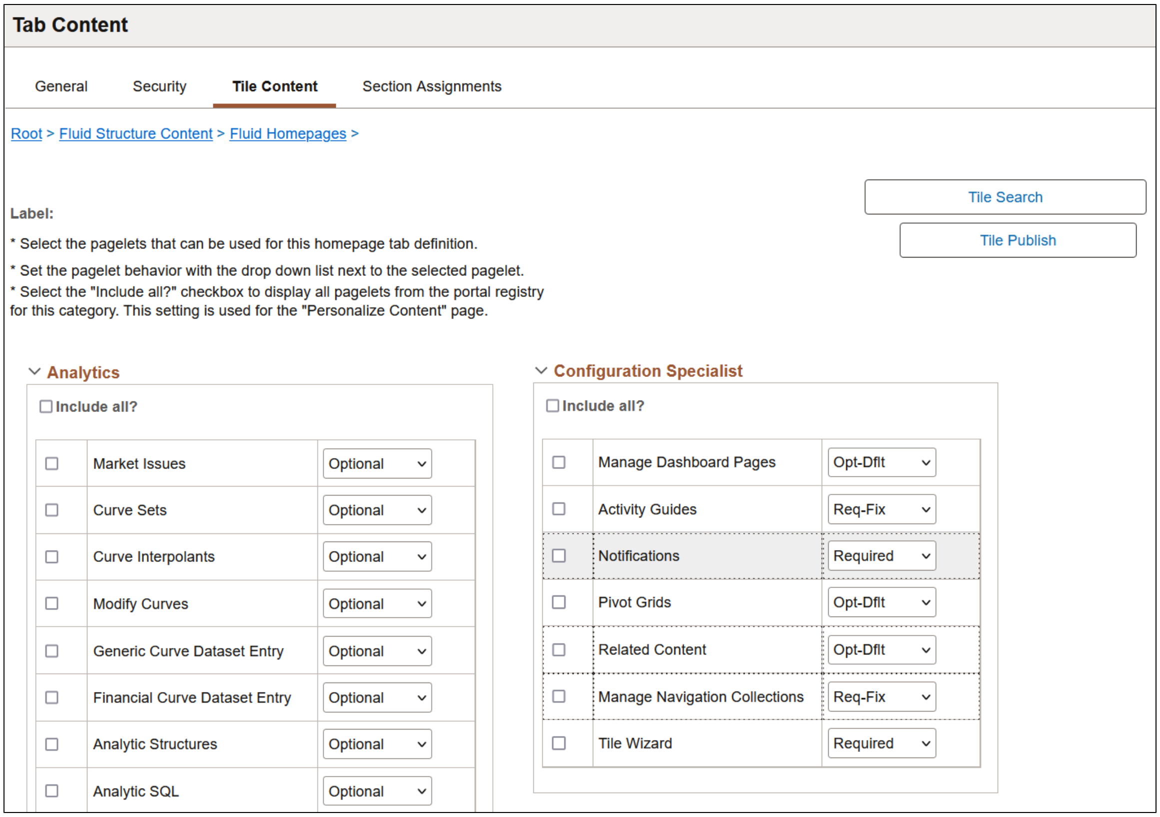
- For a classic homepage, from the Content Ref Administration page, select the Tab Content page tab.
- For a fluid homepage, from the Content Ref Administration page, select the Tile Content page tab.

This example illustrates the fields and controls on the Tab Content - Tab Content page. You can find definitions for the fields and controls later on this page.



This example illustrates the fields and controls on the Tile Content - Tab Content page. You can find definitions for the fields and controls later on this page.

Important! While this page still determines which tiles are displayed as required, fixed, or by default, the restrictive function of this page does not apply to fluid homepages—that is, all tiles can be added to any fluid homepage by any user with permissions to that tile regardless of whether the tile has been explicitly allowed on this page.



To select the pagelets or tiles that can be displayed on this homepage:

1. Expand any collapsed group boxes and then select the check box to make the pagelet or tile available for this homepage.

   Select the **Include All?** check box to select all pagelets or tiles in that category.

   Note: A user must also have security access to the pagelet or tile definition for the item to be displayed on the homepage.

   ---

2. In addition:

   - For classic homepages, you can click the Pagelet Search button and select a pagelet from the Pagelet Search page to add it to the homepage tab definition.

     See Searching for and Importing Pagelets and Tiles for more information on searching for pagelets on local and remote nodes.

   - For fluid homepages, you can click the Tile Search button and select a pagelet from the Tile Search page to add it to the homepage tab definition.

See Searching for and Importing Pagelets and Tiles for more information on searching for tiles on local and remote nodes.

3. Use the drop-down list box to select the pagelet or tile behavior. If you're creating a section, you must select the tile behavior type other than Optional.

Note: If you change the behavior type of a tile and the tile is assigned to a section on the Section Assignments page, the behavior type of the tile in the section is automatically updated to reflect the change.

Values are:

Field or Control	***Description***
Opt-Dflt (optional default)	The pagelet or tile is visible to all users by default for users that have permissions to the pagelet or tile; users can move or delete the pagelet or tile. **Note:** You must select the *Opt-Dflt* setting if you wish to publish additional tiles to users who have already personalized this homepage. See (Fluid only) Using the Tile Publish Page to Publish Additional Tiles and (Fluid only) Using a PeopleCode Program to Publish Additional Tiles for more information.
Optional	Optional means this pagelet or tile is available for a user to add to their homepage as long as the user has security permission to this pagelet or tile. Additionally, users can move or delete the pagelet or tile. When an administrator sets a pagelet or tile as optional in a homepage, it does not display automatically on the homepage. However, in fluid homepages, all tiles regardless whether they are set optional or not, are available for the user to add as long as the user has security permission to it. Only classic homepages, classic dashboards, and fluid dashboards allow only optional pagelets or tiles to be added by the user to their classic homepage, classic dashboard, and fluid dashboard. **Note:** However, since all tiles are already available to all users that have permissions to the tile, this selection is unnecessary for tiles. Setting a tile that is assigned in a section from non-Optional to Optional removes the tile from the section because Optional tiles cannot be assigned to sections.

Field or Control	Description
Req-Fix (required fixed)	The pagelet or tile is visible to all users by default for users that have permissions to the pagelet or tile; however, users cannot move or delete the pagelet or tile. **Note:** A tile that is designated as fixed cannot be moved directly. If a fixed tile is also the first tile, no other tiles can be moved in front of it. However, if non-fixed tiles precede the fixed tile, other tiles can be moved in front of the fixed tile thereby rendering the fixed setting moot.
Required	The pagelet or tile is visible to all users by default for users that have permissions to the pagelet or tile; users can move the pagelet or tile, however, they cannot delete it.

4. Click the **Save** button.

   If you're creating a standard fluid homepage, that is, a homepage without sections, then on clicking the **Save** button, the fluid homepage is ready for use.

   If you want to create a sectionalized fluid homepage, that is, a homepage with sections, then you should use the Section Assignments tab to create sections and assign tiles to it.

(Fluid only) Deselecting a Tile on an Existing Fluid Homepage

When you deselect a tile on an existing fluid homepage, the tile is either displayed or not displayed based on these conditions:

- If the fluid homepage is *not* personalized by a user, the deselected tile is not displayed on the homepage.
- If the fluid homepage is user personalized, the deselected tile continues to display on the user-personalized homepage. However, the user can delete this tile on the user-personalized homepage.
- If the deselected tile had the Required or Req-Fix (required fix) behavior, it is changed to the Opt-Dflt (optional-default) behavior, which enables the user to move or delete the tile from a personalized homepage.

To remove a tile from a user-personalized homepage or to disable a tile from being added by a user in a homepage, you should either set an expiry date for the tile or change the tile permission setting.

Related Links

"Personalizing Classic Homepages and Dashboards" (Applications User's Guide)

(Fluid only) Creating Sections on Fluid Homepages

Use the Section Assignments page (PORTAL_TAB_SECTION) to create sections and assign tiles to a fluid homepage or load tiles dynamically using an application class.

Note: Only Portal Administrators can create sections on a homepage.

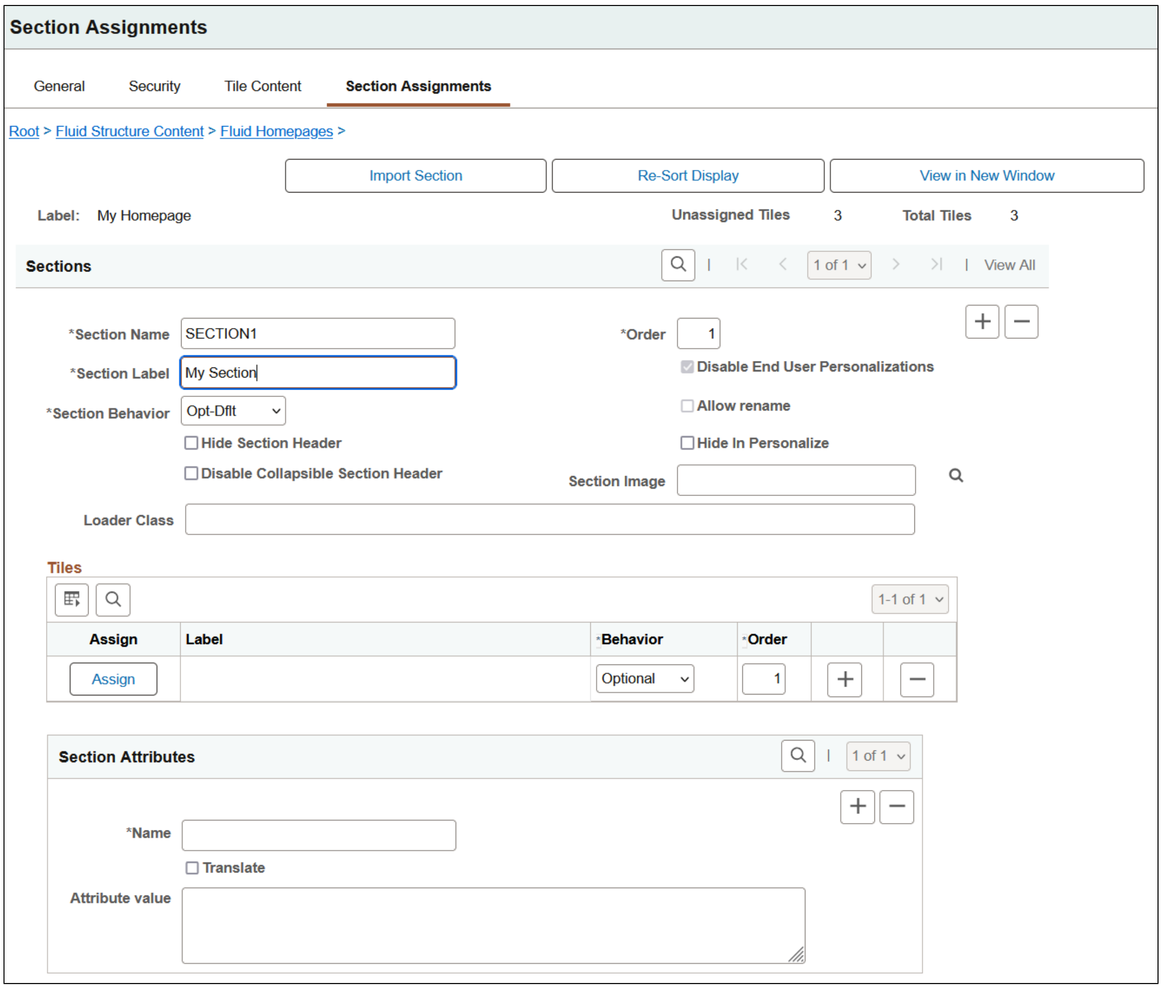
To improve the usability of homepages, Portal Administrators can group tiles into a section. The grouping of tiles can be based on your business requirements.

Note: PeopleTools 8.62 provides greater control on the styling and layout of homepages and sections. For more information, see Understanding Advanced Sectionalized Homepage Development.

Navigation:

From the Content Ref Administration page, select the Section Assignments page tab.

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



Field or Control	*Description*
Import Section	Select to copy a section definition from a section template. Section templates are saved in the **Fluid Section Templates** folder. For more information, see Creating a Section Template and Importing a Section in Homepages and Dashboards.

Field or Control	*Description*
Re-Sort Display	After you've created sections, assigned tiles to the sections, and set the order for the tiles, select the **Re-Sort Display** button to sort the display of sections and tiles by their order number.
View in New Window	The View in New Window button is not available in Add mode. The button appears after the Section Assignments page is saved. Administrators and application developers can use this option to see the developed homepage as is in a new tab on the browser. You should note that this is merely a quick navigational access to see the homepage as you set the layout of the homepage with its sections. When the button is selected, the homepage definition is first saved so that all changes made to the homepage can be viewed. The re-caching of the homepage may cause the performance to be slow. If any errors are encountered while saving the homepage definition, the view in new window operation is stopped. **Note:** If you're editing an existing homepage that is available to end users and select the View in New Window button, the updated homepage will be available to end users, so developers should use standard development practices to minimize the impact to end users who are using the existing homepage. The following prerequisites should be fulfilled to view the homepage: • Administrators and application developers must have permissions to access the homepage. • The current date must be within the valid from and valid to dates specified for the homepage. If not, an error message is displayed.
Unassigned Tiles	Displays the number of tiles that are not assigned to any sections; the number decreases as you assign tiles to sections. In order to see a list of unassigned tiles, you need to add a row to a section and select the **Assign** button. The unassigned tiles are displayed at the top of the list with no assigned section text. Unassigned tiles are placed in the standard area, that is, at the top of the homepage. These tiles are ordered in the standard area based on their portal sequence number.
Total Tiles	Displays the number of tiles that you selected on the **Tiles Content** tab.

Sections

Field or Control	*Description*
Section Name	Enter a unique ID for the section definition. It should be unique within a homepage. The value that you specify in the Name field can consist of up to 30 letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.
Section Label	Enter a label for the section. The label is visible to all users who can view this section.
Section Behavior	In PeopleTools 8.61, all sections act the same because users cannot modify or personalize sectionalized homepages. The behavior types that are applicable to homepages are also applicable to sections. The behavior types are as follows: • Optional: Optional sections are currently not available. • Opt-Dflt (optional default) • Req-Fix (required fix) • Required
Hide Section Header	Select to not display the text in the section header including the image, if specified.
Disable Collapsible Section Header	When you select this option, the section header cannot be collapsed, that is, the section header remains expanded. To display the section image correctly, this option should be set.
Loader Class	Enter the full local path of the application class used to load tiles dynamically in the associated section. It is an extension to the base class PTNUI:Dynamic:DynamicLoader. For more information, see “DynamicLoader Class Methods” (PeopleCode API Reference).

Field or Control	*Description*
Order	Enter a value to designate the order of sections on a homepage. The order number determines the sequence in which the sections appear on a homepage. The section with the lowest order number appears first. If any order numbers are identical, then the sections are sorted alphabetically by the section label. **Note:** The section order is solely controlled by the order number specified here.
Disable End User Personalizations	In PeopleTools 8.61, personalization of sections is not supported, so the **Disable End User Personalization** check box is automatically selected and cannot be changed.
Allow rename	When end-user personalization is disabled for a section, the renaming of a section is also automatically disabled.
Hide in Personalize	Select to hide the section on the Personalize Homepage page. The section with its tiles is hidden, that is, users cannot copy the tiles to another homepage. In effect, a section with this attribute is a system-owned section.
Section Image	Select an image to display in the section header. The image should be displayed beside the section title.

Rules for managing a section:

- A section can contain several tiles.
- If a section does not contain any tiles, the section is not displayed on the homepage. A section can be empty due to runtime characteristics, for example:
  - You do not have permissions to the tiles in a section or tiles are hidden through the TILEAPPFILTER attribute. In this scenario, the tiles are not displayed. Consequently, the section is not displayed.
  - Not all tiles in a section are available or optimized for small form factor devices. If you're using a small form factor device, then these tiles are not displayed and the section is hidden.

  You can use the empty-section tile, which is an administrator controlled tile, to ensure that the section is displayed. See Configuring Tiles for an Empty Section.
- If all the tiles in a section are marked as Hide in Personalize and contains the empty tile, then these tiles are administrator-controlled tiles, which users cannot copy to other homepages. In this case, it is recommended that the section should also have the Hide in Personalize attribute set.

  See Hiding Tiles in Sections on Homepage Personalization.

- Similar to the arrangement of tiles with the Req-Fix behavior at the beginning of a standard homepage, you must ensure that the sections with the Req-Fix behavior are grouped together at the beginning of the homepage. In a section, tiles with the Req-Fix behavior should be grouped together at the beginning of each section.

Tiles

The **Tiles** grid appears after you create a section, which enables you to assign tiles for the section.

Tiles have to selected on the Tile Content tab first in order for the tiles to be available for assignment into sections.

Field or Control	***Description***
Assign	Clicking this button displays the Tile Assignment page where you can select tiles to insert into the section.
Label	Displays the label of the tile.
Behavior	Displays the behavior of the tile that you specified for a tile on the Tile Content tab. However, you can modify the behavior type as required.
Order	Enter a value to designate the order of the tile within a section. The order number determines the sequence in which the tiles appear in a section. The tile with the lowest order number appears first in a section. If any order numbers are identical, then the tiles are sorted alphabetically by the tile label. **Note:** The tile order is solely controlled by the order number specified here.

Rules for assigning tiles to a section:

- A tile can appear only once on a homepage.
- Ensure that the tiles with the Req-Fix behavior are grouped together at the beginning of the section.
- Empty-section tile must be assigned the Req-Fix behavior and should be placed as the first tile in a section. See Configuring Tiles for an Empty Section.

Section Attributes

Use the Section Attributes area to specify stylesheet, style classes, and other attributes that are applicable at the section level.

See Styling Homepages and Dashboards Using CSS Variables and Applying Styles from Delivered Style Sheets for more information on attributes and its values for section level.

Related Links

Managing Section Templates and Sections

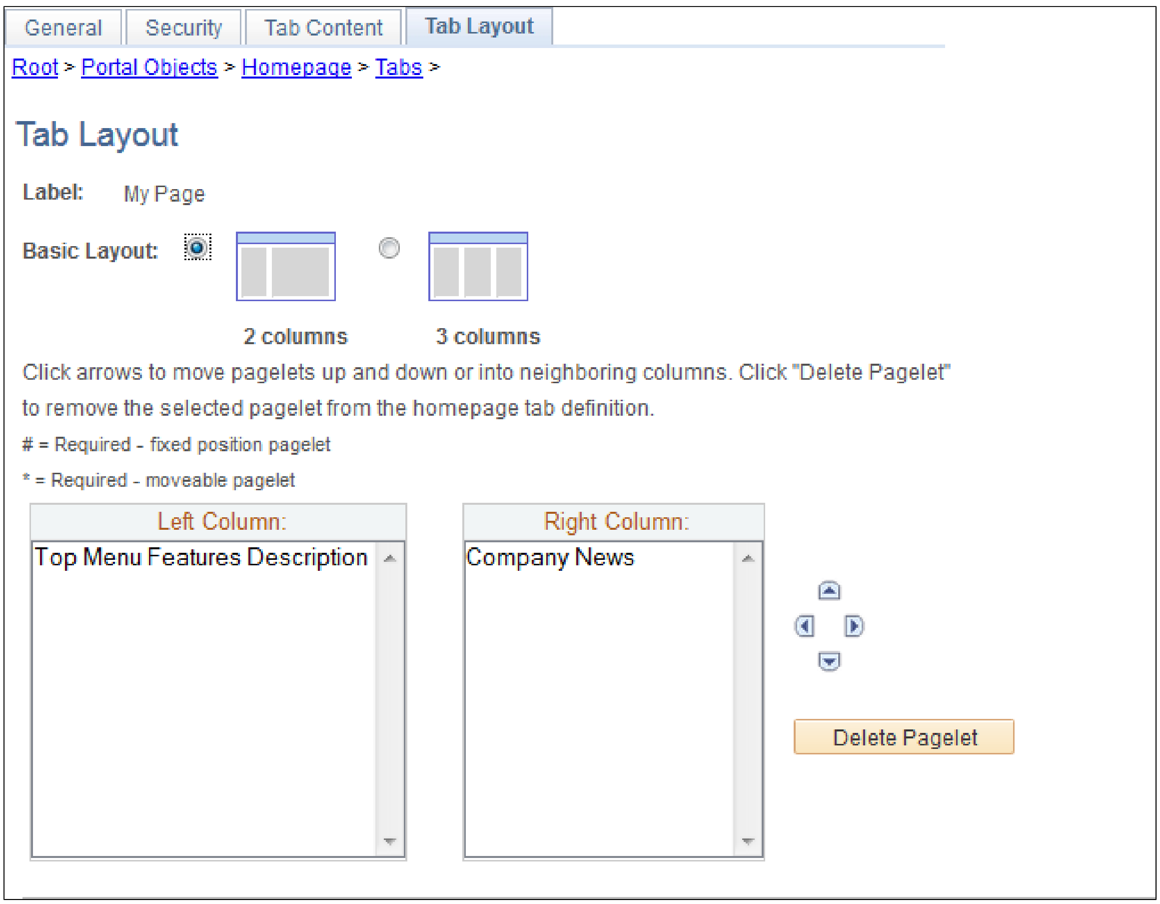
(Classic only) Arranging Pagelets on Classic Homepages

Use the Tab Layout page (PORTAL_TAB_LAY_ADM) to specify the initial layout of columns and pagelets for a classic homepage.

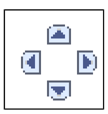
Navigation:

From the Content Ref Administration page, select the Tab Layout page tab.

This example illustrates the fields and controls on the Tab Layout page for classic homepages. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Label	Displays the label information from the Content Ref Administration page.

Field or Control	***Description***
Basic Layout	Select to display the pagelets in either two or three columns on the homepage. To specify the three-column layout, be sure that at least one pagelet does not have the *Req-Fix* behavior option selected on the Tab Content page.
Columns	Displays the pagelets selected on the Tab Content page in the column sections. If the basic layout is two columns, the pagelets are divided into left and right columns. If the basic layout is three columns, the pagelets are divided into left, center, and right columns. **Note:** A pound sign (#) indicates a pagelet with the *Req-Fix* behavior option selected on the Tab Content page. An asterisk (*) indicates a pagelet with the *Required* behavior option selected on the Tab Content page. For example, #Signon would indicate that the Signon pagelet cannot be moved to a different location on the homepage.
	Use the **Move** icons to position the pagelets. Highlight a pagelet, and then click the directional arrow buttons to move the selected pagelet above a pagelet, below a pagelet, to the next column to the right, or the next column to the left.
Delete Pagelet	Highlight a pagelet and then click to delete it from the homepage.

To arrange the tab layout:

1. Select a basic layout option.
2. Select the pagelet name.
3. Click the directional arrow buttons to position the pagelets in the columns.
4. Click the **Save** button.

Note: As the portal administrator, you determine the amount of flexibility that you give users to change the layout of the pagelets. The settings that you select on both the Tab Content and Tab Layout pages determine the degree of flexibility. Grant the end user as much or as little flexibility as your organization demands.

(Fluid only) Arranging Tiles on Fluid Homepages

There is no specific page that allows the portal administrator to arrange or order the tiles on a fluid homepage. Instead, the sequence number for each tile determines the initial order of tiles on a fluid homepage. Tiles appear in order with the lowest sequence number first. If any sequence numbers are identical, then the order is alphanumeric. If no sequence number is specified, the default value of zero is used for that tile.

To rearrange the tile order:

1. Navigate to **PeopleTools** > **Portal** > **Structure and Content.**
2. Continue to navigate through the folder structure; select **Fluid Structure Content** > **Fluid Pages**.
3. Click the link for the folder that contains the tile that you want to reorder.
4. Click the **Edit** link for each tile that you want to reorder.
5. Enter a number in the **Sequence** field.
6. Click Save.

Arranging Tiles in Sections on Fluid Homepages

For fluid homepages with sections, you can decide the sequence of tiles by specifying the order number for tiles.

Use the **Section Assignments** page to arrange the tiles in sections.

Related Links

Creating a Section Template

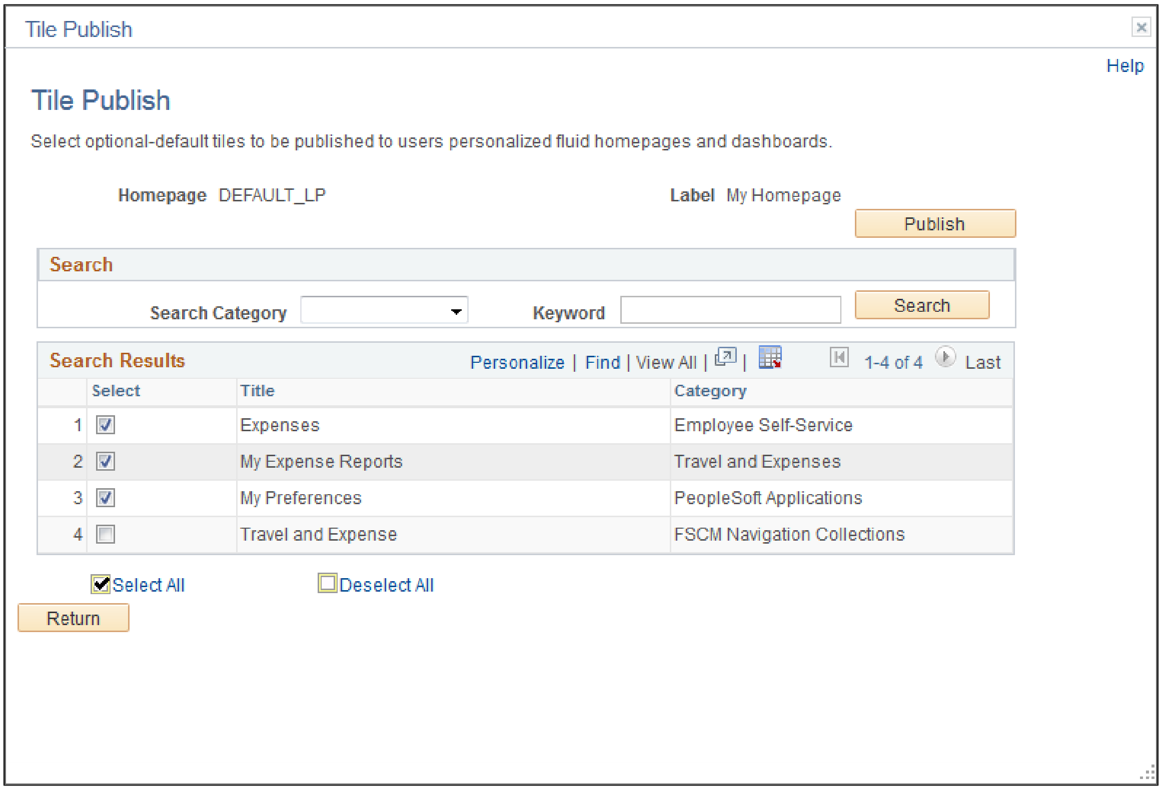
(Fluid only) Using the Tile Publish Page to Publish Additional Tiles

If you have the Portal Administrator role, you can use the Tile Publish page to publish tiles to already personalized homepages or dashboards. Otherwise, you can write a PeopleCode program to publish tiles to already personalized homepages or dashboards; for more information, see the following section.

Note: On homepages that are disabled for end-user personalizations, the **Tile Publish** button is not enabled.

Use the Tile Publish page to publish additional tiles to users who have already personalized this homepage or dashboard.

This example illustrates the fields and controls on the Tile Publish page.



To publish additional tiles to users who have already personalized this homepage or dashboard:

1. On the Tile Content - Tab Content page, select the additional tiles for this homepage or dashboard.
2. Select *Opt-Dflt* for each of these tiles.
3. Save the homepage definition (or dashboard definition).
4. Click the Tile Publish button.

   The Tile Publish page is displayed listing all new and existing tiles that have been marked as *Opt-Dflt*.
5. Select which tiles are to be published to users who have already personalized this homepage or dashboard.

   Note: Tiles that are not selected and tiles that have been previously published for this homepage or dashboard are not removed by this process.

   ---
6. Click the Publish button.

   A confirmation message indicates how many tiles have been published.
7. Click OK.
8. Make note of the process ID in case you need to use Process Monitor to review the status of this PT_TILE_PUB process.

Note: Process Scheduler must be running on your system to complete this process. After the process successfully completes, the updates will be visible to users when they sign in next or after they refresh the homepage or dashboard.

9. Click the Return button to return to the Tile Content - Tab Content page.

(Fluid only) Using a PeopleCode Program to Publish Additional Tiles

You can also write a PeopleCode program to call the delivered PublishToHomepage function, which runs the same PT_TILE_PUB Application Engine program that is invoked from the Tile Publish page. This methodology provides users that do not have the Portal Administrator role a way to publish tiles to already personalized fluid homepages or dashboards. If already personalized homepages or dashboards are found that do not include the tiles to be published, the initiator of this action will be prompted to continue before the PT_TILE_PUB program is submitted to the Process Scheduler; otherwise, the PT_TILE_PUB program is not submitted.

Note: To run the PT_TILE_PUB program, at least one of the user’s permission lists must also include the TLSALL process group.

Syntax

```
PublishToHomepage(HP_name, tile_list, behavior, run_now, run_DT)
```

Description

Use the PublishToHomepage function to add and then publish tiles to personalized fluid homepages or dashboards.

Parameters

Field or Control	***Description***
HP_name	Specifies the content reference name (ID) for the fluid homepage or dashboard as a string value.
tile_list	Specifies one or more content reference names (IDs) for tile definitions as a comma-delimited string value.

Field or Control	Description
behavior	Specifies the tile behavior for all tiles in the tile list as a string value: • *3DEF* – Optional default. • *4OPT* – Optional. This behavior pertains to fluid dashboards only. **Note:** If a specified tile is already configured for the homepage or dashboard and its current behavior differs from what is specified here, then the behavior will be updated when the program is executed.
run_now	Specifies a boolean value indicating whether to submit the Application Engine program to the Process Scheduler immediately.
run_DT	Specifies the date and time to submit the Application Engine program as a datetime value. If *run_now* is True, this value is ignored.

Returns

A Record object in the format of the PT_TILEPUB_WRK2 record definition with the following fields:

- PROCESSINSTANCE (number): The process request number if the request was submitted successfully.
- PTSCHEDULESTATUS (number): 0 indicates the process was submitted successfully; a non-zero number indicates that the submit function failed.
- STATUS (string): 0 indicates that the process was not required to be scheduled because there were no user personalizations to this homepage. Y indicates that the operator clicked Yes to submit the process; N indicates that the operator clicked No to decline submitting the process.

Example

```
Declare Function PublishToHomepage PeopleCode PT_TILE_PUB_WRK.FUNCLIB FieldFormula;

&rec = PublishToHomepage("DEFAULT_LP", "TILE_CREF_1, TILE_CREF_2", "3DEF", True, Da⇒

teTimeValue("2020-06-30 22:30:00"));

If &rec.STATUS = "Y" Then
   If &rec.PTSCHEDULESTATUS.Value = "0" Then
      MY_WRK.PROCESSINSTANCE = &rec.PROCESSINSTANCE;
      MessageBox(0, "", 99999, 101, "Publish process submitted. Process ID: " | &re⇒

c.PROCESSINSTANCE);
   Else
      MessageBox(0, "", 99999, 102, "Process could not be scheduled.");
```

```
   End-If;
End-If;
```

Specifying the Homepage Sequence

The sequence number determines the order in which the homepages appear. Homepages appear in order with the lowest sequence number first. If any sequence numbers are identical, then the order is alphanumeric. If no sequence number is specified, the default value of zero is used for that homepage.

Important! Therefore, the tab with the lowest sequence number becomes the default homepage.

To rearrange the homepage order:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**
2. Continue to navigate through the folder structure depending on the homepage type:
   - For classic homepages, select **Portal Objects** > **Homepage** > **Tabs**.
   - For fluid homepages, select **Fluid Structure Content** > **Fluid Homepages**.
3. Click the **Edit** link for each homepage that you want to reorder.
4. Enter a number in the **Sequence** field.
5. Click Save.

If caching is turned on for the portal, the tab order change does not take effect until the cache is recycled or the web server is restarted.

(Fluid only) Publishing Personalizations as a New Fluid Homepage

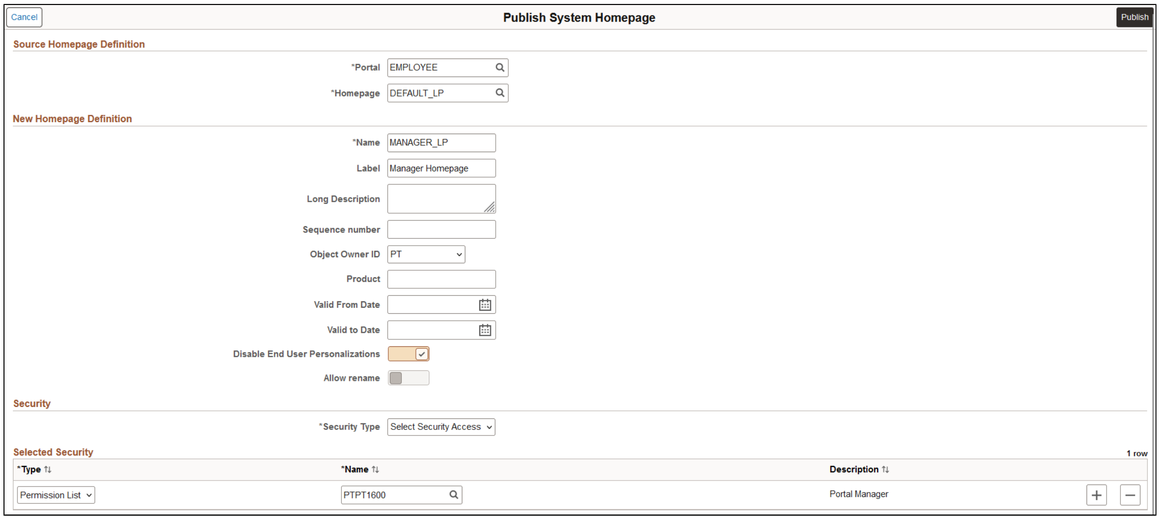
Use the Publish System Homepage page (PTPHP_PUBLISHHP) to publish an administrator created or personalized homepage as a new fluid homepage.

Note: Only users with the PTPT1600 (Portal Manager) permission list can access this page to publish a personalized homepage.

Navigation:

1. View the personalized fluid homepage that you want to publish as a new fluid homepage.
2. From the Actions menu, select Publish Homepage.

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



Source Homepage Definition

Field or Control	***Description***
Portal	Select the portal registry in which this definition is stored. The current portal name is displayed by default.
Homepage	Displays the ID of the copied homepage definition.

New Homepage Definition

Field or Control	***Description***
Name	Enter a unique ID for the homepage definition. The value that you specify in the **Name** field can consist of up to 50 letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.
Label	Enter the label that users will see for the homepage.
Long Description	Enter a long description for the homepage definition.

Field or Control	***Description***
Sequence number	Enter a value to designate the order of the homepage when multiple homepages are defined. The sequence number determines the order in which the homepages appear. The homepages appear with the lowest sequence number first. If any numbers are identical, then the order is alphanumeric. No value is equivalent to specifying *0*. **Important!** Therefore, the homepage with the lowest sequence number becomes the default homepage.
Object Owner ID	Select the object owner ID.
Product	Portal-aware applications can use this field for group processing. All PeopleTools objects have the product name *PT*.
Valid From Date	Select or enter the date that the homepage becomes visible to users.
Valid to Date	Select or enter an optional date that the homepage is removed from use.
Disable End User Personalizations	Select to disable end-user personalizations for the homepage. When you select this check box, the **Allow rename** check box is automatically unavailable for entry because modifying the homepage label is an end-user personalization. For more information on using this check box, see (Fluid only) Disabling User Personalization for Homepages.
Allow rename	Select to enable the user to personalize the homepage label. **Note:** When a portal administrator updates the label for a fluid homepage or a fluid dashboard, the PeopleSoft system propagates the new label to homepages or dashboards that have not been personalized and to personalized homepages or dashboards that also retain the original label. If a user has personalized the label, then the user's personalization is retained.

Security

Field or Control	*Description*
Security Type	Select the type of security that you want to assign to the homepage definition. Options are: • *Public Access:* Select to designate that all users can access the homepage. • *Select Security Access:* Select to assign role-based or permission list-based security to the homepage definition. The Selected Security scroll area is displayed.

Selected Security

Field or Control	*Description*
Type	Select from these types: • *Permission List:* Enables access for users based on permission list membership. You then specify the permission lists that should have access to this homepage. • *Role:* Enables access for users based on role membership. You then specify the roles that should have access to this homepage.
Name	Select the permission list or role that you want have access to this homepage.

Related Links

“Working with Fluid Homepages and Dashboards” (Applications User’s Guide)

(Fluid only) Disabling User Personalization for Homepages

Portal administrators can prohibit end users from personalizing homepages by using the **Disable End User Personalizations** check box that is available on fluid homepage definitions and the Publish System Homepage page. Sectionalized homepages automatically disable end-user personalization. You can apply this setting for standard homepages and dashboards (homepages or dashboards without sections) and for sectionalized homepages and dashboards.

Portal administrators can disable end-user personalization on fluid homepage and dashboard definitions by navigating through the Structure and Content component. Dashboard administrators cannot specify this setting through Manage Dashboard Pages.

Consider the following when you plan to disable end-user personalization:

- When you disable end-user personalization, personalization of a homepage is restricted except for the following personalization:
  - User can personalize the order of homepages.
  - User can copy a tile to another homepage that allows user personalization and to homepages where the tile is not present.

  For more information see “Managing Fluid Homepages” (Applications User’s Guide).
- When end-user personalization is disabled for a homepage, which has been personalized by users in the past, that personalization is no longer available for use (or the personalization data becomes stale) because end-user personalization data is not maintained, except for the homepage order. User personalization data that is no longer maintained includes removed tiles, added tiles from the tile repository, pin-to tiles, and tile order changes.
- Standard homepages can only be published if the homepage has been modified. With homepages set to disable end-user personalization, homepages cannot be modified. In this case, an administrator can perform a Save As operation on a non-modified homepage. This enables administrators to create a new version of a homepage using the current homepage as a starting point.

  The homepage publishing process provides you the option to disable end-user personalization for the newly published homepage.

  Refer to the Cloning of Publishing a Homepage as a New Homepage section later in this topic for more information.
- Homepages that are disabled for end-user personalization cannot be the target of copy tile or move tile actions.
- Homepages that are disabled for end-user personalization are not listed in the Add To Homepage menu from non-homepage components.

Cloning or Publishing a Homepage as a New Homepage

Homepages that are disabled for end-user personalization, both standard homepages and sectionalized homepages, can be published or cloned as a new homepage.

Use the Publish System Homepage page (PTPHP_PUBLISHHP) to publish or clone a homepage that is disabled for end-user personalization as a new fluid homepage.

Note: Only users with the PTPT1600 (Portal Manager) permission list can access this page to publish or clone a homepage that is disabled for end-user personalization and modify the cloned homepage.

To publish a homepage (which is disabled for end-user personalization) as a new homepage:

1. View the fluid (disabled for end-user personalization) homepage.
2. From the Actions menu, select Publish Homepage.
3. The following message is displayed:

   This homepage does not allow personalization and is unmodified. Do you wish to save a copy of this homepage under a different name?

Click OK on the message that is displayed.

For further instructions on cloning or publishing a homepage, see (Fluid only) Publishing Personalizations as a New Fluid Homepage.

(Fluid only) Restricting Fluid Homepage and NavBar Personalization

You can change the access to certain PeopleTools-delivered definitions to restrict certain ways in which users can personalize fluid homepages, the NavBar, or both. The delivered PTPT1400 (PeopleSoft Guest) permission list has some of these personalization features turned off by default. See the following table for a list of personalization features, the PeopleTools definition that provides access to that feature, and whether the feature is enabled in PTPT1400.

Important! Do not modify the delivered PTPT1400 permission list. Create a custom permission list or clone PTPT1400 as a starting point if you wish to restrict personalization for users other than guest users.

Feature	***PeopleTools Definition***	***Enabled in PTPT1400***
Personalize fluid homepages and dashboards: • Personalize Homepage link (on fluid homepages and dashboards). • Drag-and-drop tiles (on fluid homepages and dashboards).	Menu:Component:Page permissions — PORTAL_PERS_HOMEPAGE:PORTAL_HOMEPAGE:Personalize Content	N
Display the Add To links in the Actions menu: • Add To Homepage. • Add To NavBar. • Add To Favorites.	Web library:Function permissions — WEBLIB_PTNUI:PT_BUTTON_PIN.FieldFormula.IScript_SavePin	N
Hide the Sign Out link in the Actions menu.	Web library:Function permissions — WEBLIB_PORTAL.PT_PORTAL_HDRLINK.FieldFormula.IScript_HideSignOutLink	N
Display the Notifications feature (badge in the banner or link in the Actions menu).	Menu:Component:Page permissions — PTPNEVENTS:PTPN_POPUP_WINDOW:Notification Window	N
Display the NavBar.	Menu:Component:Page permissions — NUI_FRAMEWORK:PTNUI_NAVBAR:NavBar	Y
Personalize the NavBar.	Menu:Component:Page permissions — NUI_FRAMEWORK:PTNUI_NAVBAR:Personalization	N

Feature	***PeopleTools Definition***	***Enabled in PTPT1400***
Add tiles to the NavBar.	Menu:Component:Page permissions — NUI_FRAMEWORK:PTNUI_NAVBAR:Grouplet Search	N

Related Links

Implementing Guest User Access

Managing Dashboard Pages

This section discusses how to:

- Create and manage dashboards.
- Specify dashboard properties.
- Specify the menu location of a dashboard.
- Specify the pagelets or tiles that can appear on a dashboard.
- (Fluid only) Create sections on fluid dashboards.
- (Classic only) Arrange pagelets on classic dashboards.
- (Fluid only) Arrange tiles on fluid dashboards.
- (Fluid only) Publish additional tiles to already personalized dashboards.
- (Fluid only) Disable user personalization on fluid dashboards.

Note: In addition, you can create and manage WorkCenter dashboards (a type of classic dashboard), which are maintained separately from other classic dashboards. See Creating and Managing WorkCenter Dashboards.

Creating and Managing Dashboards

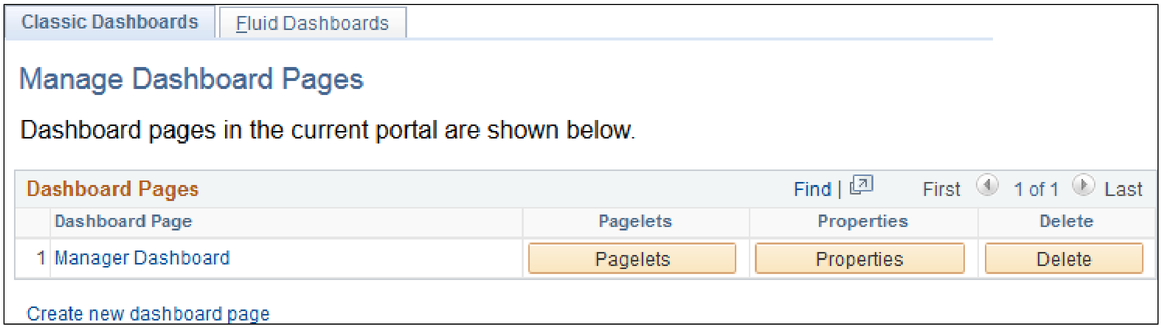
Use the Manage Dashboard Pages page to manage dashboard definitions:

- Use the Classic Dashboards page (PTAL_DASHBD_PAGES) to manage classic dashboard definitions.
- Use the Fluid Dashboards page (PTAL_FLDASHBDPAGES) to manage fluid dashboard definitions.

Navigation:

PeopleTools > **Portal** > **Dashboards** > **Manage Dashboard Pages**

This example illustrates the fields and controls on the Classic Dashboards: Manage Dashboard Pages page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Create new dashboard page	Click to access the Dashboard Page Administration page and configure the properties of a new dashboard definition.
Dashboard Page	Displays an active link to the dashboard itself. Click the link to access the dashboard.
(Classic only) **Pagelets**	Click to configure the content and layout of a classic dashboard.
(Fluid only) **Tiles**	Click to configure the content of a fluid dashboard.
Properties	Click to access the Dashboard Page Administration page and specify dashboard properties, location, and security details.
Delete	Click to remove the dashboard content reference definition from the portal registry.

Specifying Dashboard Properties

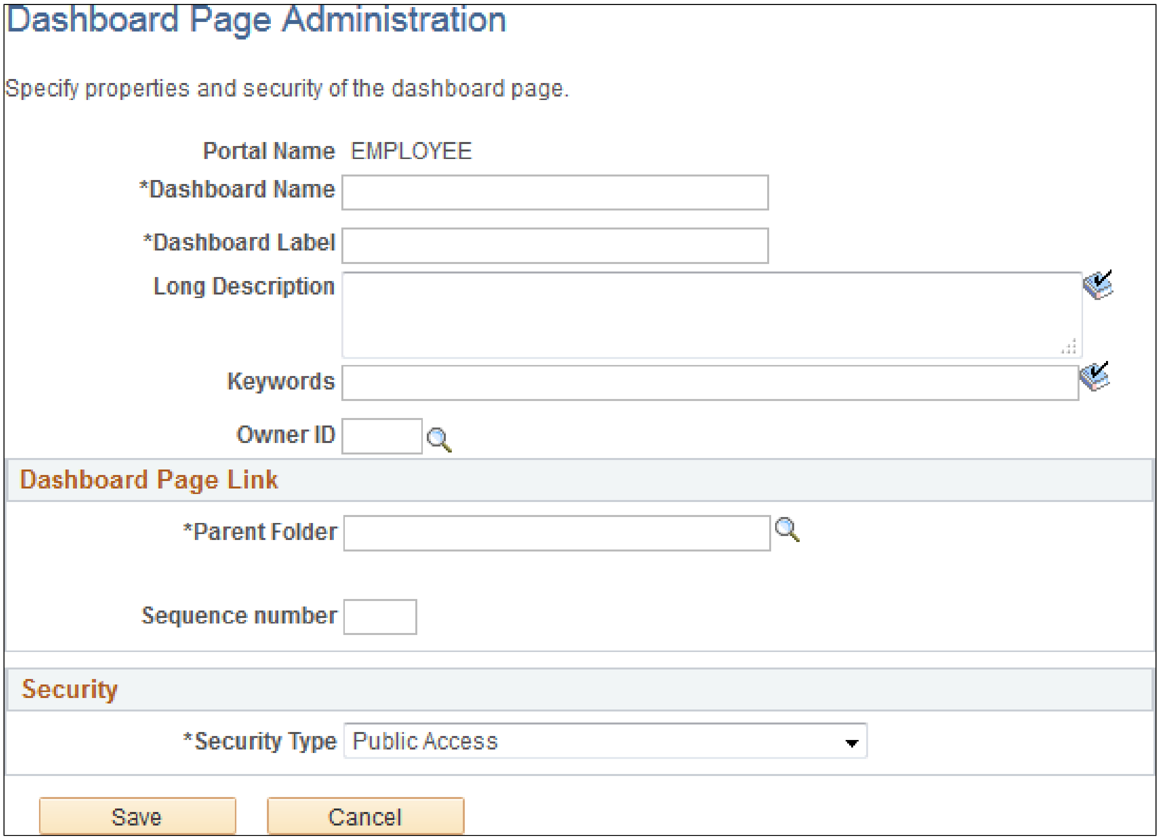
Use the Dashboard Page Administration page (PTAL_DASHBD_ADMIN) to specify dashboard properties including the dashboard's location in the menu structure.

Navigation:

On the Manage Dashboard Pages page:

- Click the **Create new dashboard page** link.
- Click the **Properties** button for an existing dashboard definition.

This example illustrates the fields and controls on the Dashboard Page Administration page. You can find definitions for the fields and controls later on this page.



Enter the following dashboard properties:

Field or Control	***Description***
Dashboard Name	Enter a unique identifier for the dashboard. This field is required.
Dashboard Label	Enter the label that appears in the navigation. This field is required.
Long Description	Enter a description that appears as the tool tip text.
Keywords	Enter a keyword or keyword phrase to be stored as a content reference attribute. **Note:** Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language). See Defining Content References for more information on content reference attributes.

Field or Control	***Description***
Owner ID	Select the organization that owns the object.

Dashboard Page Link

Use these fields to specify the location of the dashboard page link in the menu structure.

Field or Control	***Description***
Parent Folder	Enter the name of the navigation folder in which you want to place the dashboard page link. Click the look up button to access the Select the Dashboard Link Parent Folder page.
Sequence number	Enter a number to specify the order of the dashboard link within the parent folder.

Security

Use the fields in the Security group box to specify whether access to the dashboard is restricted by permission list and role or unrestricted and available to all users.

Field or Control	***Description***
Security Type	Select from these values: • *Public Access:* Select to allow all users to view and access the dashboard. • *Select Security Access:* Select to specify PeopleTools Security access, which is based on permission lists and roles. When you select this option, the Selected Security grid appears.

Selected Security

Insert rows into the **Selected Security** grid to specify the permission lists and roles that provide access to the dashboard. You can combine permission lists and roles by inserting multiple rows in the grid.

Field or Control	*Description*
Type	For each row, select from these values: • *Permission List:* Select to secure the dashboard by permission list. • *Role:* Select to secure the dashboard by role.
Name	Specify the permission list or role name that provides access to the dashboard. Click the look up button to see a list of available names.
Save	Click to save the configuration and return to the Manage Dashboard Pages page.
Cancel	Click to discard any changes and return to the Manage Dashboard Pages page.

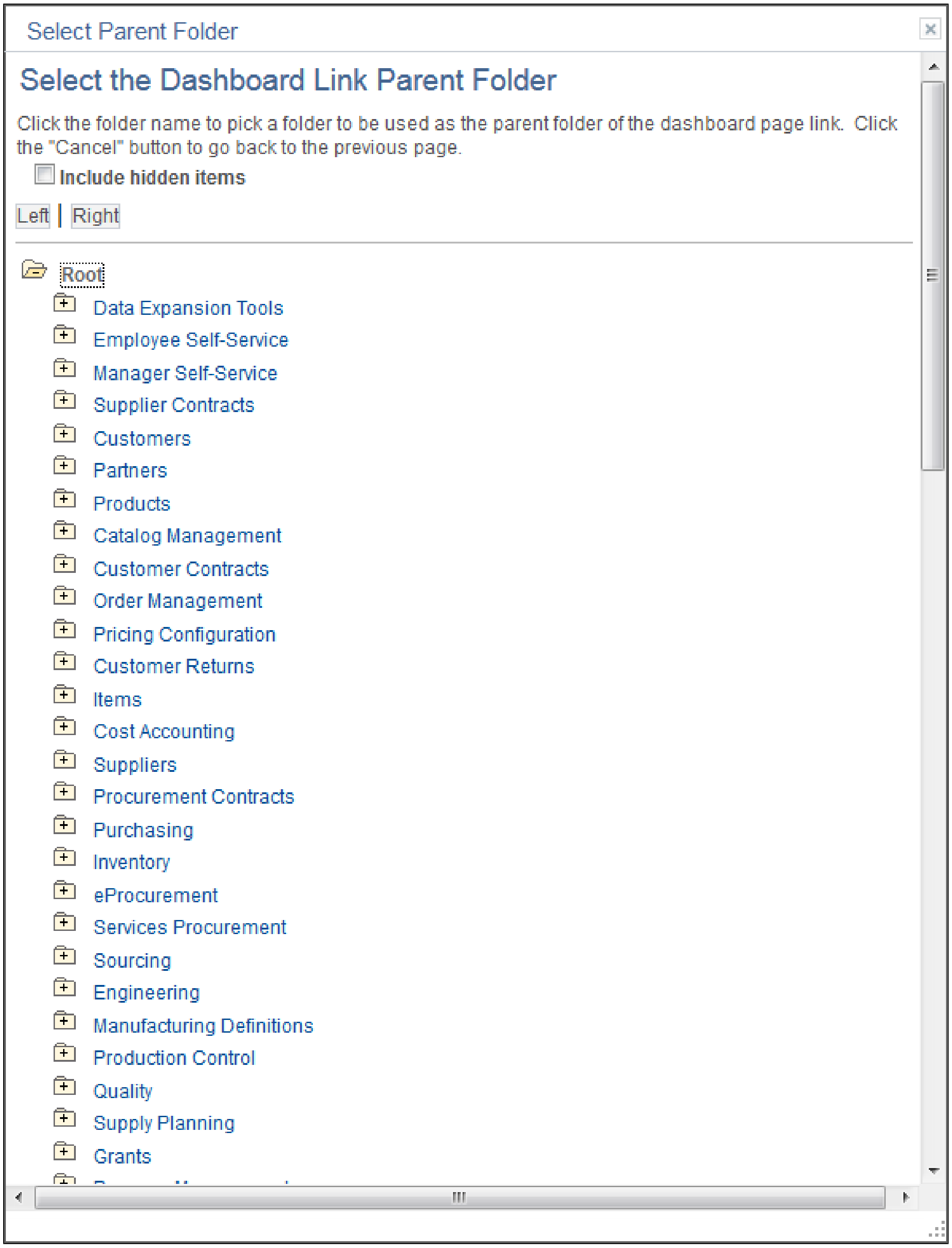
Specifying the Menu Location of a Dashboard

Use the Select the Dashboard Link Parent Folder page (PTAL_SELECT_FREF) to specify the location of the link to the dashboard in the portal menu structure.

Navigation:

On the Dashboard Page Administration page, click the look up button for the **Parent Folder** field.

This example illustrates the fields and controls on the Select the Dashboard Link Parent Folder page. You can find definitions for the fields and controls later on this page.



To select the parent folder of the link to the dashboard, you must navigate through the menu tree until you see the destination folder.

Specifying the Pagelets or Tiles That Can Appear on a Dashboard

Use the Tab Content page (PTSYSTABCONTENT) to specify which pagelets can appear on classic dashboards and which tiles can appear on fluid dashboards. (This page is similar to the pages that you use for the same purposes on homepages. The process is also the same.)

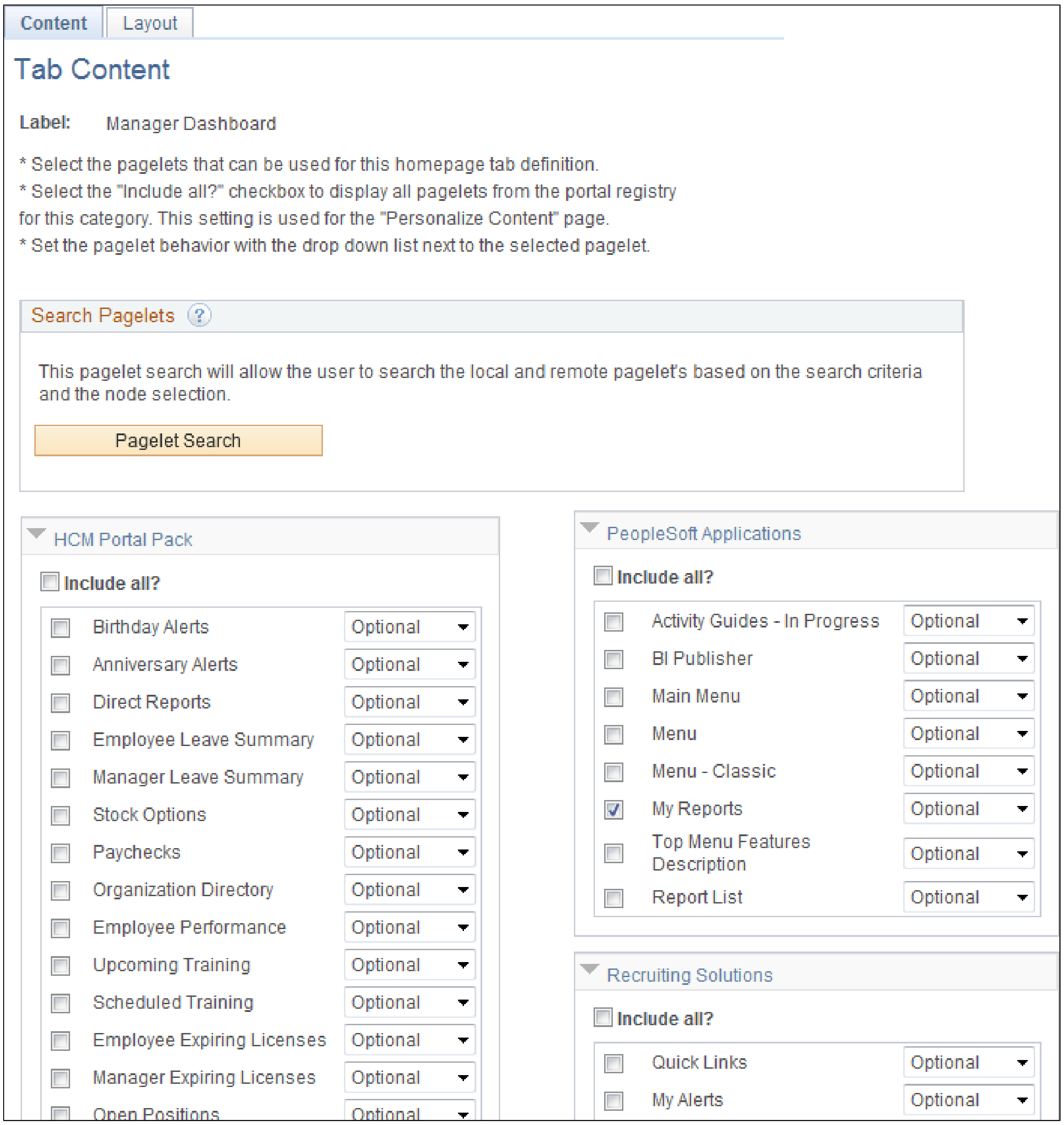
Navigation:

On the Manage Dashboard Pages page:

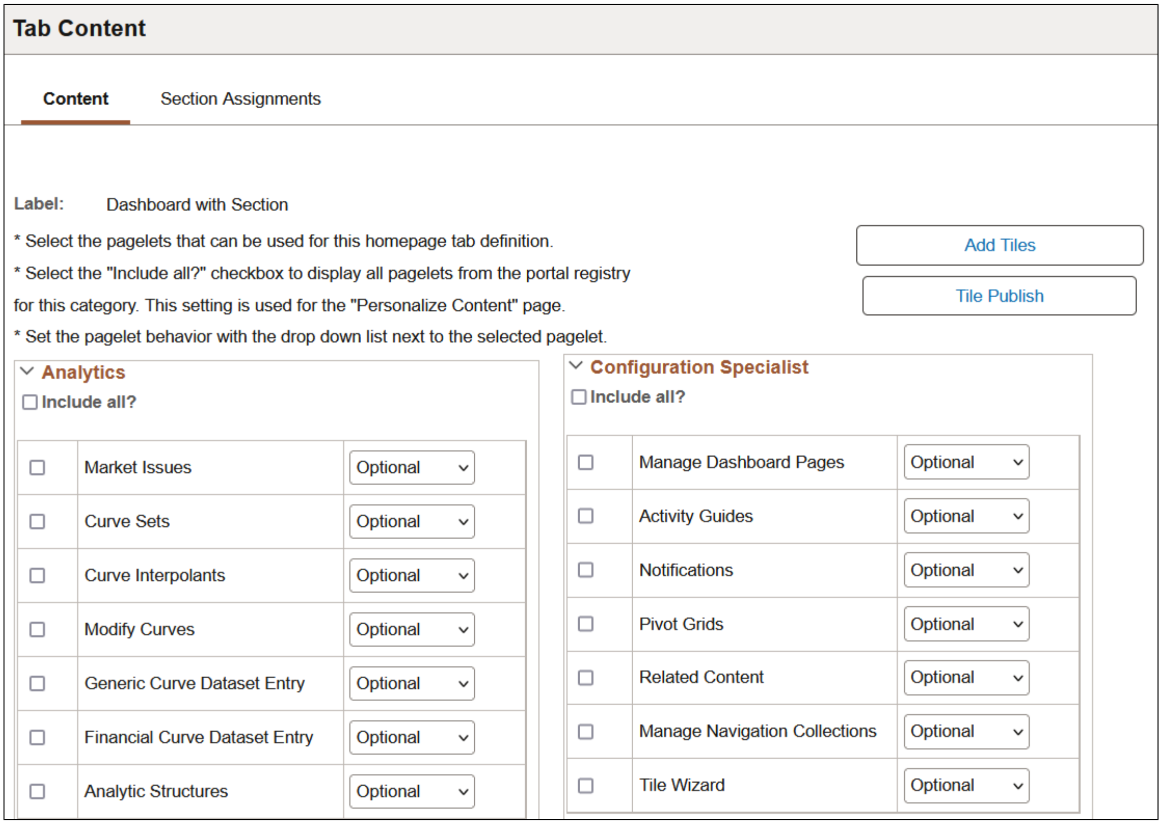
- Click the **Pagelets** button to specify pagelets for a classic dashboard.
- Click the **Tiles** button to specify tiles for a fluid dashboard.

This example illustrates the fields and controls on the Content - Tab Content page for pagelets. You can find definitions for the fields and controls later on this page.

Note: The Layout page tab is not displayed for fluid dashboards.



This example illustrates the fields and controls on the Content - Tab Content page for tiles. You can find definitions for the fields and controls later on this page.



To select the pagelets or tiles that can be displayed on this dashboard:

1. Expand any collapsed group boxes and then select the check box to make the pagelet or tile available for this dashboard.

   Select the **Include All?** check box to select all pagelets or tiles in the category.

   Note: The user must have security access to the pagelet or tile definition for the pagelet or tile to be displayed on the dashboard.

   ---

2. In addition:

   - For classic dashboards, you can click the Pagelet Search button and select a pagelet from the Pagelet Search page to add it to the dashboard definition.

     See Searching for and Importing Pagelets and Tiles for more information on searching for pagelets on local and remote nodes.

   - For fluid dashboards, you can click the Add Tiles button and select a tile from the Tile Search page to add it to the dashboard definition.

     See Searching for and Importing Pagelets and Tiles for more information on searching for tiles on local and remote nodes.

3. Use the drop-down list box to select the pagelet or tile behavior. If you're creating a section, you must select the tile behavior type other than Optional.

Note: If you change the behavior type of a tile and the tile is assigned to a section on the Section Assignments page, the behavior type of the tile in the section is automatically updated to reflect the change.

Values are:

Field or Control	***Description***
Optional	The pagelet or tile is available to all users that have permissions to the pagelet or tile. The pagelet or tile is not visible by default, but users have the option to add this tile or pagelet to their dashboard. In addition, once added to a dashboard, the user can move or delete this pagelet or tile. Only selected optional tiles are available for users to add later on to their dashboard. Setting a tile that is assigned in a section from non-Optional to Optional removes the tile from the section because Optional tiles cannot be assigned to sections.
Opt-Dflt (optional default)	The pagelet or tile is visible to all users by default for users that have permissions to the pagelet or tile; users can move or delete the pagelet or tile. **Note:** You must select the *Opt-Dflt* setting if you wish to publish additional tiles to users who have already personalized this homepage. See (Fluid only) Publishing Additional Tiles to Already Personalized Dashboards for more information.
Req-Fix (required fixed)	The pagelet or tile is visible to all users by default for users that have permissions to the pagelet or tile; however, users cannot move or delete the pagelet.
Required	The pagelet or tile is visible to all users by default for users that have permissions to the pagelet or tile; users can move the pagelet or tile, however, they cannot delete it.

4. Click the **Save** button.

   If you're creating a standard fluid dashboard, that is, a dashboard without sections, then on clicking the **Save** button, the fluid dashboard is ready for use.

   If you want to create a sectionalized fluid dashboard, that is, a dashboard with sections, then you should use the Section Assignments tab to create sections and assign tiles to it.

(Fluid only) Deselecting a Tile on an Existing Fluid Dashboard

When you deselect a tile on an existing fluid dashboard, the tile is *not* displayed on a user-personalized fluid dashboard and a non-personalized fluid dashboard.

(Fluid only) Creating Sections on Fluid Dashboards

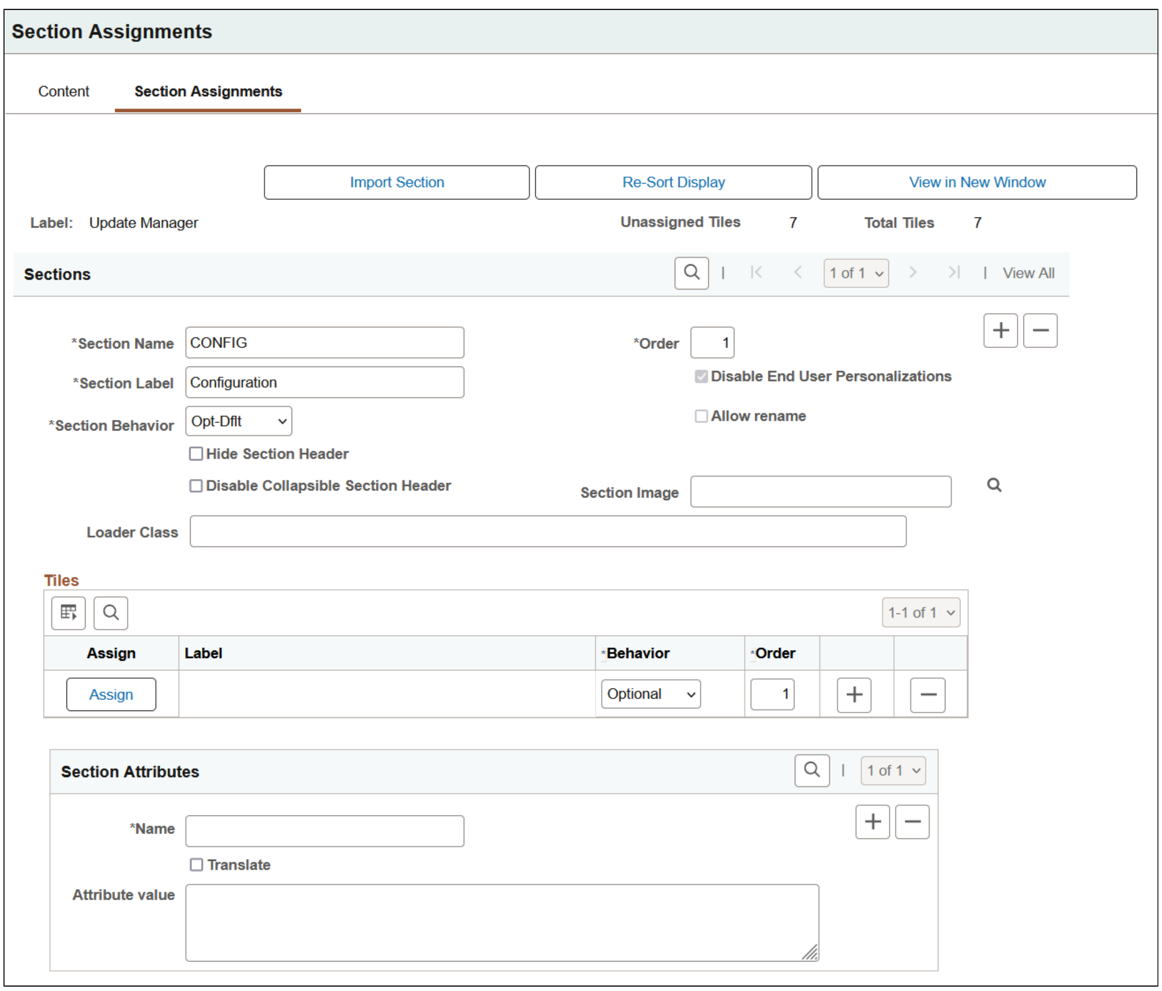
Use the Section Assignments page (PTSYSTABSECTIONS) to create sections and assign tiles to a fluid dashboard or load tiles dynamically using an application class. (This page is similar to the page for sections assignments on homepages. The process is also similar.)

Note: PeopleTools 8.62 provides greater control on the styling and layout of dashboards and sections. For more information, see Understanding Advanced Sectionalized Homepage Development.

Navigation:

On the Tab Content page, select the Section Assignments tab.

This example illustrates the fields and controls on the Section Assignments page for dashboards. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Import Section	Select to copy a section definition from a section template. The section template is saved in the Fluid Section Templates folder. For more information, see Importing a Section in Homepages and Dashboards.
Re-Sort Display	After you've created sections, assigned tiles to the sections, and set the order for the tiles, select the **Re-Sort Display** button to sort the display of sections and tiles by their order number.
View in New Window	The View in New Window button is not available in Add mode. The button appears after the Section Assignments page is saved. Administrators and application developers can use this button to see the developed homepage as is in a new tab on the browser. You should note that this is merely a quick navigational access to see the homepage as you set the layout of the homepage with its sections. When the button is selected, the homepage definition is first saved so that all changes made to the homepage can be viewed. The re-caching of the homepage may cause the performance to be slow. If any errors are encountered while saving the homepage definition, the view in new window operation is stopped. **Note:** If you're editing an existing homepage that is available to end users and select the View in New Window button, the updated homepage will be available to end users, so developers should use standard development practices to minimize the impact to end users who are using the existing homepage. The following prerequisites should be fulfilled to view the homepage: • Administrators and application developers must have permissions to access the homepage. • The current date must be within the valid from and valid to dates specified for the homepage. If not, an error message is displayed.
Unassigned Tiles	Displays the number of tiles that are not assigned to any sections; the number decreases as you assign tiles to sections. Unassigned tiles are placed in the standard homepage area, that is, at the top of the homepage. These tiles are ordered on the homepage based on their portal sequence number.
Total Tiles	Displays the number of tiles that you selected on the **Content** tab.

Sections

Rules for managing a section:

- A section can contain several tiles.
- If a section does not contain any tiles, the section is not displayed on the dashboard.

  You can use the empty-section tile, which is an administrator controlled tile, to ensure that the section is displayed. See Configuring Tiles for an Empty Section.
- Ensure that the sections with the Req-Fix behavior are grouped together at the beginning of the dashboard.

Field or Control	***Description***
Section Name	Enter a unique ID for the section definition. It should be unique within a dashboard. The value that you specify in the Name field can consist of up to 30 letters, digits, and underscores (_). It cannot contain any spaces and cannot begin with a digit.
Section Label	Enter a label for the section. The label is visible to all users who can view this section.
Section Behavior	The behavior types that are applicable to homepages are also applicable to sections. The behavior types are as follows: • Optional • Opt-Dflt (optional default) • Req-Fix (required fix) • Required
Hide Section Header	Select to not display the text in the section header including the image, if specified.
Disable Collapsible Section Header	When you select this option, the section header cannot be collapsed, that is, the section header remains expanded.
Loader Class	Enter the full local path of the application class used to load tiles dynamically in the associated section. It is an extension to the base class PTNUI:Dynamic:DynamicLoader. For more information, see “DynamicLoader Class Methods” (PeopleCode API Reference).

Field or Control	*Description*
Order	Enter a value to designate the order of the sections when multiple sections are defined on a homepage. The order number determines the sequence in which the sections appear on a homepage. The section with the lowest order number appears first. If any order numbers are identical, then the sequence is alphanumeric. **Note:** The section order is solely controlled by the order number specified here.
Disable End User Personalizations	By default, end-user personalization is disabled for sectionalized dashboards. See (Fluid only) Disabling User Personalization for Dashboards.
Allow Rename	When end-user personalization is disabled for a dashboard, the renaming of a dashboard is also automatically disabled.
Section Image	Select an image to display in the section header. The image should be displayed beside the section title.

Tiles

The **Tiles** grid appears after you create a section, which enables you to assign tiles for the section.

Rules for assigning tiles to a section:

- A tile can appear only once on a dashboard.
- Ensure that the tiles with the Req-Fix behavior are grouped together at the beginning of the section.
- Empty-section tile must be assigned the Req-Fix behavior and should be placed as the first tile in a section. See Configuring Tiles for an Empty Section.

Field or Control	*Description*
Assign	Clicking the **Assign** button displays the Tile Assignment page where you can select tiles to insert into the section.
Label	Displays the label of the tile.

Field or Control	*Description*
Behavior	Displays the behavior of the tile that you specified for a tile on the **Content** tab. However, you can modify the behavior type as required.
Order	Enter a value to designate the order of the tile within a section. The order number determines the sequence in which the tiles appear in a section. The tile with the lowest order number appears first in a section. If any order numbers are identical, then the tile sequence is alphanumeric. **Note:** The tile order is solely controlled by the order number specified here.

Section Attributes

Use the Section Attributes area to specify stylesheet, style classes, and other attributes that are applicable at the section level.

See Styling Homepages and Dashboards Using CSS Variables and Applying Styles from Delivered Style Sheets for more information on attributes and its values for section level.

Importing a Section

You can import the section templates that are stored in the Fluid Section Templates folder into a dashboard.

For instructions on creating a section template, see Creating a Section Template.

The steps to import a section into a dashboard are similar to importing a section into a homepage. See Importing a Section in Homepages and Dashboards.

Related Links

Managing Section Templates and Sections

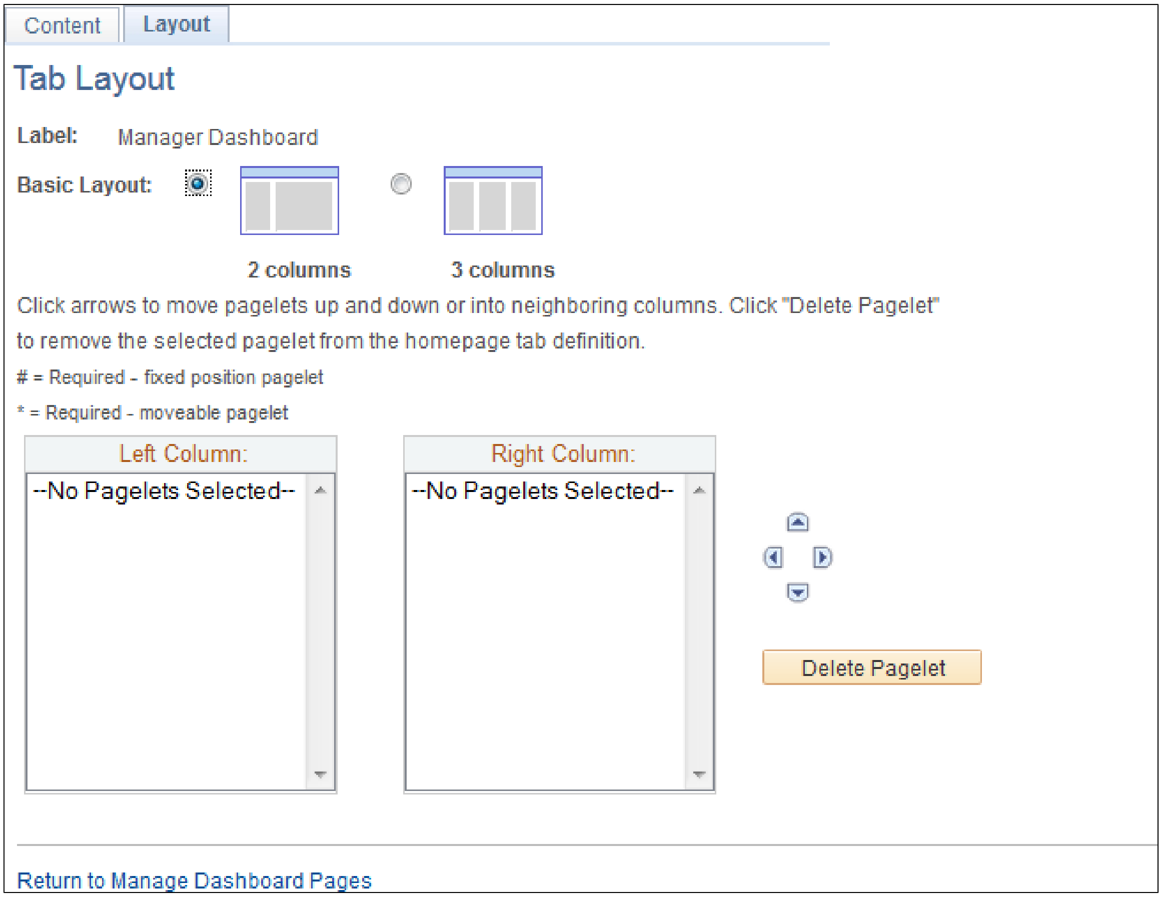
Arranging Pagelets on Classic Dashboards

Use the Tab Layout page (PTSYSTABLAYOUT) to specify the initial layout of columns and pagelets for a classic dashboard. (This page is similar to the page that you use for the same purpose on homepages. The process is also the same.)

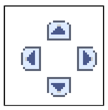
Navigation:

On the Tab Content page, select the Layout page tab.

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



Field or Control	***Description***
Label	Displays the label information from the Content Ref Administration page.
Basic Layout	Select to display the pagelets in either two or three columns on the dashboard. To specify the three-column layout, be sure that at least one pagelet does not have the *Req-Fix* behavior option selected on the Content page.

Field or Control	***Description***
Columns	Displays the pagelets selected on the Content page in the column sections. If the basic layout is two columns, the pagelets are divided into left and right columns. If the basic layout is three columns, the pagelets are divided into left, center, and right columns. **Note:** A pound sign (#) indicates a pagelet with the *Req-Fix* behavior option selected on the Content page. An asterisk (*) indicates a pagelet with the *Required* behavior option selected on the Content page. For example, #Signon would indicate that the Signon pagelet cannot be moved to a different location on the homepage.
	Use the **Move** icons to position the pagelets. Highlight a pagelet, and then click the directional arrow buttons to move the selected pagelet above a pagelet, below a pagelet, to the next column to the right, or the next column to the left.
Delete Pagelet	Highlight a pagelet and then click to delete it from the homepage.

To arrange the layout of the dashboard:

1. Select a basic layout option.
2. Select the pagelet name.
3. Click the directional arrow buttons to position the pagelets in the columns.
4. Click the **Save** button.

Note: As the portal administrator, you determine the amount of flexibility that you give users to change the layout of the pagelets. The settings that you select on both the Tab Content and Tab Layout pages determine the degree of flexibility. Grant the end user as much or as little flexibility as your organization demands.

Arranging Tiles on Fluid Dashboards

There is no specific page that allows the portal administrator to arrange or order the tiles on a fluid dashboard. Instead, the sequence number for each tile determines the initial order of tiles on a fluid dashboard similar to how tiles are ordered on a fluid homepage. Tiles appear in order with the lowest sequence number first. If any sequence numbers are identical, then the order is alphanumeric. If no sequence number is specified for a tile, the default value of zero is used for that tile.

To rearrange the tile order:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**
2. Continue to navigate through the folder structure; select **Fluid Structure Content** > **Fluid Pages**.

3. Click the link for the folder that contains the tile that you want to reorder.

4. Click the **Edit** link for each tile that you want to reorder.

5. Enter a number in the **Sequence** field.

6. Click Save.

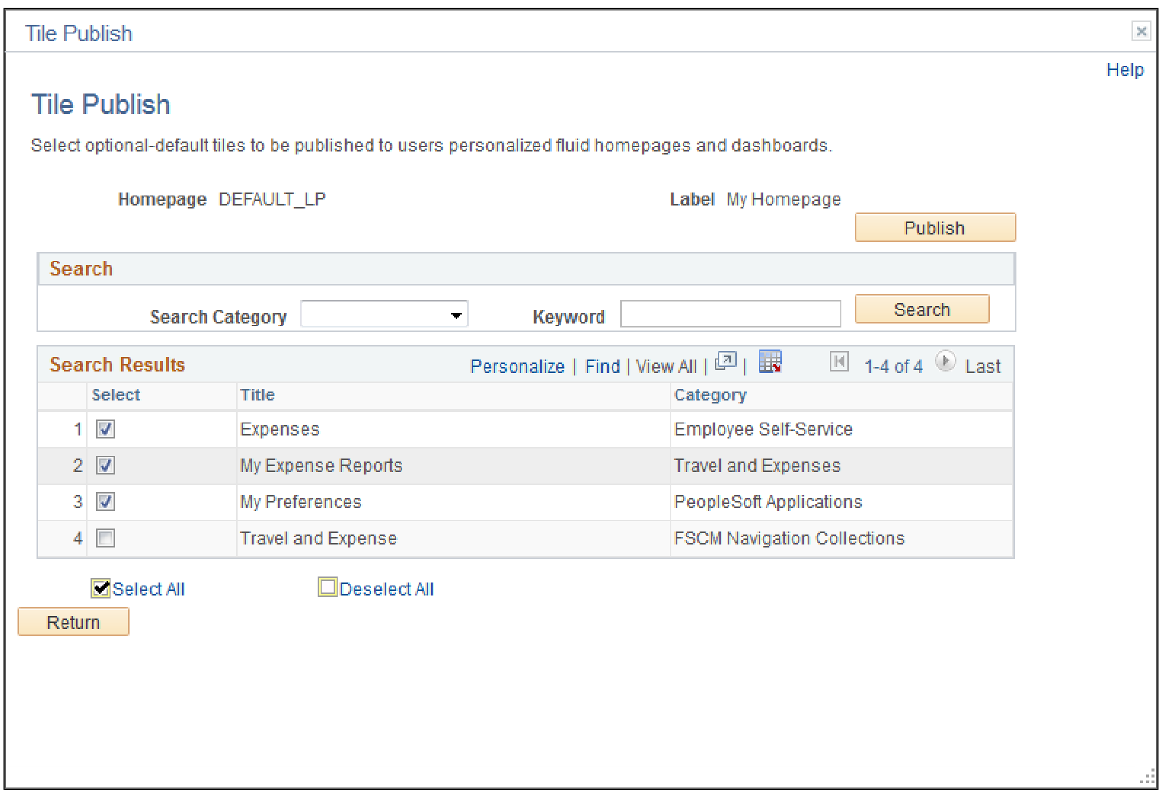
(Fluid only) Publishing Additional Tiles to Already Personalized Dashboards

If you have the Portal Administrator role, you can use the Tile Publish page to publish tiles to already personalized homepages or dashboards. Otherwise, you can write a PeopleCode program to publish tiles to already personalized homepages or dashboards; for more information, see the following section.

Note: On homepages that are disabled for end-user personalizations, the **Tile Publish** button is not enabled.

Use the Tile Publish page to publish additional tiles to users who have already personalized this homepage or dashboard.

This example illustrates the fields and controls on the Tile Publish page.



To publish additional tiles to users who have already personalized this homepage or dashboard:

1. On the Tile Content - Tab Content page, select the additional tiles for this homepage or dashboard.

2. Select *Opt-Dflt* for each of these tiles.

3. Save the homepage definition (or dashboard definition).
4. Click the Tile Publish button.

   The Tile Publish page is displayed listing all new and existing tiles that have been marked as *Opt-Dflt*.
5. Select which tiles are to be published to users who have already personalized this homepage or dashboard.

   Note: Tiles that are not selected and tiles that have been previously published for this homepage or dashboard are not removed by this process.

   ---
6. Click the Publish button.

   A confirmation message indicates how many tiles have been published.
7. Click OK.
8. Make note of the process ID in case you need to use Process Monitor to review the status of this PT_TILE_PUB process.

   Note: Process Scheduler must be running on your system to complete this process. After the process successfully completes, the updates will be visible to users when they sign in next or after they refresh the homepage or dashboard.

   ---
9. Click the Return button to return to the Tile Content - Tab Content page.

Related Links

(Fluid only) Using a PeopleCode Program to Publish Additional Tiles

(Fluid only) Disabling User Personalization for Dashboards

Portal administrators can disable end-user personalization on fluid dashboard definitions by navigating through the Structure and Content component. Dashboard administrators cannot specify this setting through Manage Dashboard Pages.

By default, sectionalized dashboards automatically disable end-user personalization. You can apply this setting for standard dashboards (dashboards without sections) and for sectionalized dashboards.

Consider the following when you plan to disable end-user personalization:

- When you disable end-user personalization, users cannot personalize a dashboard. In fact, the Personalize Dashboard option is not available in the dashboard Actions menu.
- When end-user personalization is disabled for a dashboard, which has been personalized by users in the past, that personalization is no longer available for use (or the personalization data becomes stale) because end-user personalization data is not maintained. User personalization data that is no longer maintained includes removed tiles, added tiles from the tile repository, pin-to tiles, and tile order changes.

Creating Contextual Fluid Dashboards

Understanding Contextual Fluid Dashboards

Contextual fluid dashboards allow application developers to design a fluid dashboard in which user-specified or fixed context values determine the content of the tiles displayed on the dashboard. To create a contextual fluid dashboard, an application developer must:

1. Create a dashboard header page and component definition.
2. Create a custom application class that implements the PTNUI:Banner:BannerTile base class.
3. Add PeopleCode to the header page and component.
4. Create one or more context-enabled tiles that can consume the context provided by the dashboard header component.
5. Create a fluid dashboard definition.

Creating a Dashboard Header Page and Component Definition

The dashboard header component defines the contextual data keys that are used to provide context to the tiles on the dashboard. This dashboard header component is normally displayed fixed at the top of the dashboard page except when the dashboard is displayed in a fluid activity guide, master/detail component, or on a small form factor device.

To create a dashboard header page and component:

1. Create a new, standard page definition. Do not select the Page (Fluid) definition type because that also requires selecting a layout type.
2. On the Use tab of the Page Properties dialog box, select the Fluid Page check box.
3. Add a group box to the page encompassing most of the width and height of the page.
   a. Select the Fluid tab of the Group Box Properties dialog box.
   b. In the Style Classes field, enter the following style: *psc_lpbanner-context*
   c. In the Group Box Type field, select *Layout Only*.
4. Add one or more input fields that represent the contextual data keys. Note the following design considerations:
   - Disable autocomplete for each field (record field properties, Use tab).
   - Do not use FieldChange PeopleCode on these input fields to invoke any BannerTile methods that would refresh the dashboard. See Adding PeopleCode to the Dashboard Header Page and Component for information on how to add an Update button to the page.
   - In certain circumstances, rather than allowing input directly on this page, create a secondary page for entering contextual data. Do this when there are more than two input fields on the page or if an input field is a date field or a popup group box field.

- Add an Edit button on the main page to open the secondary page.
- Set the input fields on the main page to display only.

5. In designing your page and component, note the "Limitations with Interactive Grouplets" in Fluid Related Content.
6. Create a standard fluid component definition. Note the following design considerations:
    - Specify a search record as is appropriate for your page and component design.
    - On the Fluid tab of the Component Properties dialog box, ensure the search page type is *None*.
    - Do not run the component registration wizard or register this component in the portal registry. Do not create a content reference definition for this component in the tile repository. Moreover, do not provide any direct menu navigation to this component.
7. Manually, add the dashboard header component to a menu definition in Application Designer.

    Note: If necessary, create a menu definition specifically for this and other dashboard header components.
8. In the browser, add the menu, component, and page to one or more permission lists. Use these same security settings when you create the fluid dashboard definition.

Warning! While the delivered PTNUI_BANNERTILE component provides an example of a dashboard header component, do not clone this component or the PTNUI_BANNERTILE page as a starting point for your implementation. Moreover, do not clone the PTNUI:Banner:DefaultBannerTile to create your custom application class.

Creating a Custom Application Class

The custom application class that you implement has a one-to-one relationship to your dashboard header component. The dashboard header page must instantiate an object of this class in the page's Activate event. Note the following design considerations for your custom application class:

- Your class implements the PTNUI:Banner:BannerTile base class. Use the **implements** keyword, not **extends**.
- The constructor method of your class must call "BannerTile" (PeopleCode API Reference) to instantiate the superclass by passing the dashboard ID. In your constructor, you can optionally:
    - Set the "AnnounceText" (PeopleCode API Reference) property.
    - Set the "IsPersistContext" (PeopleCode API Reference) property only if you do not wish to persist the context data.
- You must implement the "GetGroupletURL" (PeopleCode API Reference) abstract method to return the URL to the dashboard header component. The following example demonstrates the recommended approach using the GenerateComponentContentURL built-in function.

- Oracle recommends that you implement (override) the “GetBannerLabel” (PeopleCode API Reference) method to set a label for the dashboard header component for accessibility purposes only.
- Optionally, implement (override) the “SetDefaultContext” (PeopleCode API Reference) method to set default context values, possibly from persistent storage.
- Optionally, implement (override) the “OverrideDefaultProperties” (PeopleCode API Reference) method to override the default setting for certain properties—for example, the IsExpanded property.

The following example provides a simple implementation of PTNUI:Banner:BannerTile:

```
import PTNUI:Banner:BannerTile;

class MyBannerComp implements PTNUI:Banner:BannerTile
   method MyBannerComp(&hp_ID As string);
   method GetGroupletURL() Returns string;
   method SetDefaultContext();
   method GetBannerLabel() Returns string;
end-class;

method MyBannerComp
   /+ &hp_ID as String +/
   %Super = create PTNUI:Banner:BannerTile(&hp_ID);
   %This.AnnounceText = "One or more tiles has been updated.";
   rem %This.IsPersistContext = False;
end-method;

method GetGroupletURL
   /+ Returns String +/
   /+ Extends/implements PTNUI:Banner:BannerTile.GetGroupletURL +/
   Local string &url = GenerateComponentContentURL(%Portal, %Node, MenuName.MY_MENU⇒

, "GBL", Component.MY_DRIVERCOMP, "", "");
   Return &url;
end-method;

method SetDefaultContext
   /+ Extends/implements PTNUI:Banner:BannerTile.SetDefaultContext +/
   %This.SetKeyValue("SETID", "AUS01");
   %This.SetKeyValue("DESCR", "Default description");
end-method;

method GetBannerLabel
   /+ Returns String +/
   /+ Extends/implements PTNUI:Banner:BannerTile.GetBannerLabel +/
   Return "My Dashboard: Contextual Data Area";
end-method;
```

Warning! Do not clone or copy PTNUI:Banner:DefaultBannerTile to create your custom application class.

Related Links

“GenerateComponentContentURL” (PeopleCode Language Reference)

Adding PeopleCode to the Dashboard Header Page and Component

Update your page and component definition to include these PeopleCode programs:

1. Add page Activate PeopleCode to this page that links this component to your custom application class. There is a one-to-one relationship between a dashboard header component and a custom application class. In the Activate program:

- Get the TABID value from the Request object and use this value to instantiate an object of your custom application class.
- Display the *current* context values to the user.
- Invoke the required "ApplyVisuals" (PeopleCode API Reference) method.

For example:

```
import MYPKG:MyBannerComp;

Component MYPKG:MyBannerComp &oBanner;

Local string &hp_ID = %Request.GetParameter("TABID");

&oBanner = create MYPKG:MyBannerComp(&hp_ID);

/* Display the current context values to the user.                        */
/* When IsPersistContext is true, the keys exist; use GetKeyValue.          */
/* Else, you must assign the same values as specified in SetDefaultContext. */
If &oBanner.IsKeyExist("SETID") Then
   GetField(MY_REC.SETID).Value = &oBanner.GetKeyValue("SETID");
Else
   GetField(MY_REC.SETID).Value = "AUS01";
End-If;

If &oBanner.IsKeyExist("DESCR") Then
   GetField(MY_REC.DESCR).Value = &oBanner.GetKeyValue("DESCR");
Else
   GetField(MY_REC.DESCR).Value = "Default description";
End-If;

/* Required for correct rendering of the dashboard. */
&oBanner.ApplyVisuals();
```

2. If there are fields to update either on the main page or on a secondary page, add an explicit trigger field as a push button or hyperlink on the main page (even if the data entry occurs on the secondary page):

   - Add FieldChange PeopleCode to this component record field to invoke the "UpdateContext" (PeopleCode API Reference) method, which updates the dashboard with the context data values. For example:

     ```
     import MYPKG:MyBannerComp;

     Component MYPKG:MyBannerComp &oBanner;

     &oBanner.SetKeyValue("SETID", GetField(MY_REC.SETID).Value);
     &oBanner.SetKeyValue("DESCR", GetField(MY_REC.DESCR).Value);

     &oBanner.UpdateContext()
     ```

   - On the Use tab of the Page Field Properties dialog box, select the Activate by Enter Key check box.

Related Links

"GetParameter" (PeopleCode API Reference)

Creating Context-Enabled Tiles

The dashboard header component will pass all context values as query string parameters to all tiles on the fluid dashboard. Each tile can consume zero, one, or more of these query string parameters. In the component or iScript definition for the tile, use the QueryString property or the GetParameter method of the Request class to get the query string parameters passed by the dashboard header component.

Application developers can create tiles that are dynamically loaded into a section using the DynamicLoader application class. When a dynamic tile is created using the AddContentText() or AddContentHTML() method to provide content for the tile, this type of dynamic tile is *not* a context-enabled tile because it cannot consume the context values passed as query parameters from the dashboard header component. However, a dynamic tile can be a context-enabled tile if the tile's content is provided through a URL that uses the grouplet content URL setting that can consume the context values passed as query parameters from the dashboard header component.

In the following example, the dynamic tile uses the grouplet content URL to get the content:

```
&oTile = %This.CreateTileUsingTab("", "Information");
  &oTile.RowNum = %This.GetNextRowNumber();
  &oTile.GltURLType = "USCR";
  /* Tile content from URL */
  &oTile.GltContentUrl = GenerateScriptContentURL(%Portal, %Node, Record.WEBLIB_QEN⇒

UI, Field.FUNCLIB, "FieldFormula", "IScript_GroupletDiag");
  %This.AddTileToSection(&oTile);
```

Related Links

“QueryString” (PeopleCode API Reference)

“GetParameter” (PeopleCode API Reference)

Configuring the Fluid Dashboard

To configure a contextual fluid dashboard:

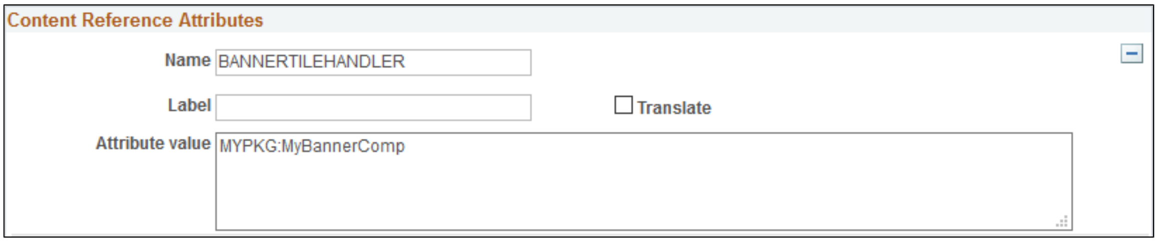
1. In the browser, select **PeopleTools** > **Portal** > **Dashboards** > **Manage Dashboard Pages**.
2. Select the Fluid Dashboards page.
3. Create a fluid dashboard definition.
4. Add contextually driven tiles to the dashboard.
5. In the portal registry, edit the content reference definition for this fluid dashboard.

   Note: The content reference definition can be found in Fluid Structure Content > Fluid Homepages.

6. Add a content reference attribute:
   - Name: BANNERTILEHANDLER
   - Translate: *deselected*
   - Attribute value: *The name of your custom application class.* For example: MYPKG:MyBannerComp

BANNERTILEHANDLER content reference attribute

The following example illustrates the definition of the BANNERTILEHANDLER content reference attribute.



Related Links

Managing Dashboard Pages

Defining Content References

Searching for and Importing Pagelets and Tiles

This section discusses how to search for and import individual pagelets and tiles. In addition, after unified navigation has been configured, you can import pagelets in bulk from content provider systems into the designated portal system.

Related Links

Importing Pagelets from Content Provider Systems

Pages Used to Search for and Import Pagelets and Tiles

Page Name	***Definition Name***	***Usage***
Pagelet Search Page	PTUN_PGLTSRCH_SEC	Search for pagelet definitions on local and remote content provider nodes.
Tile Search Page	PTUN_GPLTSRCH_SEC	Search for tile definitions on local and remote content provider nodes.

Pagelet Search Page

Use the Pagelet Search page (PTUN_PGLTSRCH_SEC) from an invoking definition (homepage, dashboard, WorkCenter, and so on) to search for pagelet definitions on local and remote content provider nodes.

Navigation:

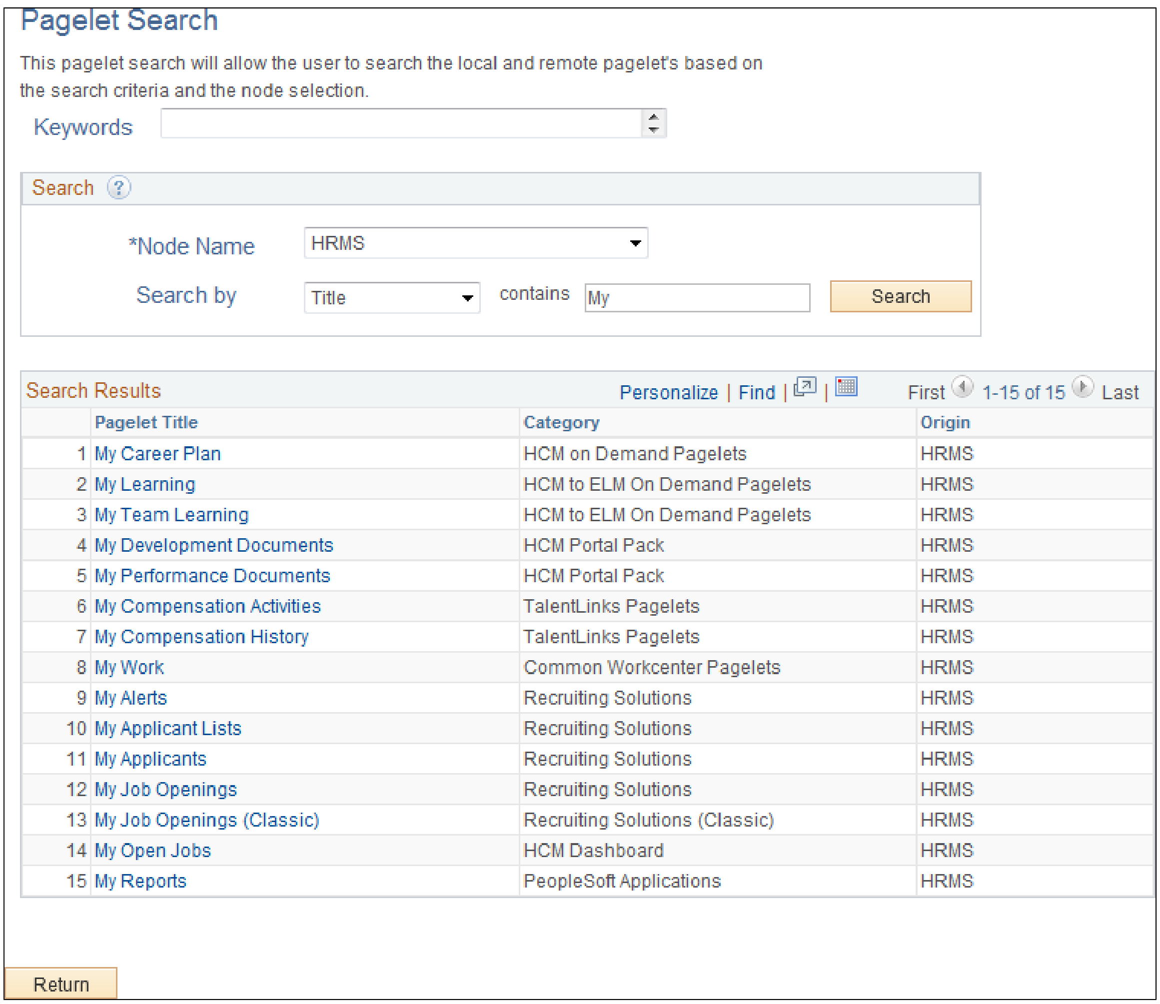
To access the Pagelet Search page:

- Click the Pagelet Search button on the:

- Dashboard’s Tab Content page.
- Homepage’s Tab Content page.
- WorkCenter dashboard’s Tab Content page.

- Click the pagelet Select icon button on the:
  - Assign Related Actions page.
  - Assign Related Content page.
  - Configure WorkCenter Page page.

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



Pagelet Search

This pagelet search will allow the user to search the local and remote pagelet's based on the search criteria and the node selection.

Keywords

Search

*Node Name HRMS

Search by Title contains My Search

Search Results Personalize | Find | First 1-15 of 15 Last

	Pagelet Title	Category	Origin
1	My Career Plan	HCM on Demand Pagelets	HRMS
2	My Learning	HCM to ELM On Demand Pagelets	HRMS
3	My Team Learning	HCM to ELM On Demand Pagelets	HRMS
4	My Development Documents	HCM Portal Pack	HRMS
5	My Performance Documents	HCM Portal Pack	HRMS
6	My Compensation Activities	TalentLinks Pagelets	HRMS
7	My Compensation History	TalentLinks Pagelets	HRMS
8	My Work	Common Workcenter Pagelets	HRMS
9	My Alerts	Recruiting Solutions	HRMS
10	My Applicant Lists	Recruiting Solutions	HRMS
11	My Applicants	Recruiting Solutions	HRMS
12	My Job Openings	Recruiting Solutions	HRMS
13	My Job Openings (Classic)	Recruiting Solutions (Classic)	HRMS
14	My Open Jobs	HCM Dashboard	HRMS
15	My Reports	PeopleSoft Applications	HRMS

Return

Field or Control	*Description*
Keywords	Displays any keywords defined on the invoking definition as a display-only value—for example, a WorkCenter definition can include keywords. **Note:** If keywords are defined on the invoking definition, then this field is populated, the Search by field is set to *Keywords,* the contains field is populated with the same keywords, and the search is automatically executed. You can update the search parameters, such as the contains field, to perform a different search after this initial keyword search.

Search

Field or Control	*Description*
Node Name	Select the local or remote content provider node. **Important!** While this field displays local and remote *default local nodes* and *portal host nodes,* you should select portal host nodes only (that is, content provider nodes) for pagelet searches.
Search by	Select one of the following: • *Category* – To search by the title of the pagelet folder. • *Keyword* – To search by keywords added to the pagelet definition. See Maintaining Pagelet Branding Attributes for more information about adding keywords to pagelet definitions. (See Defining Content References for information about adding keywords to any content reference definition.) • *Role* – To search by roles that have access to the pagelet definition. • *Title* – To search by the pagelet title.
contains	Enter a word, partial word, phrase, or partial phrase as the search term. **Note:** Clearing this field and performing a search will return all results displaying valid values for the selected Search by value.

Field or Control	*Description*
Search	Click the Search button to perform a search.
Return	Click the Return button to return to the invoking page without selecting a pagelet definition.

Search Results

Field or Control	*Description*
Pagelet Title	Click a pagelet title to select and add that definition to the invoking definition.

Tile Search Page

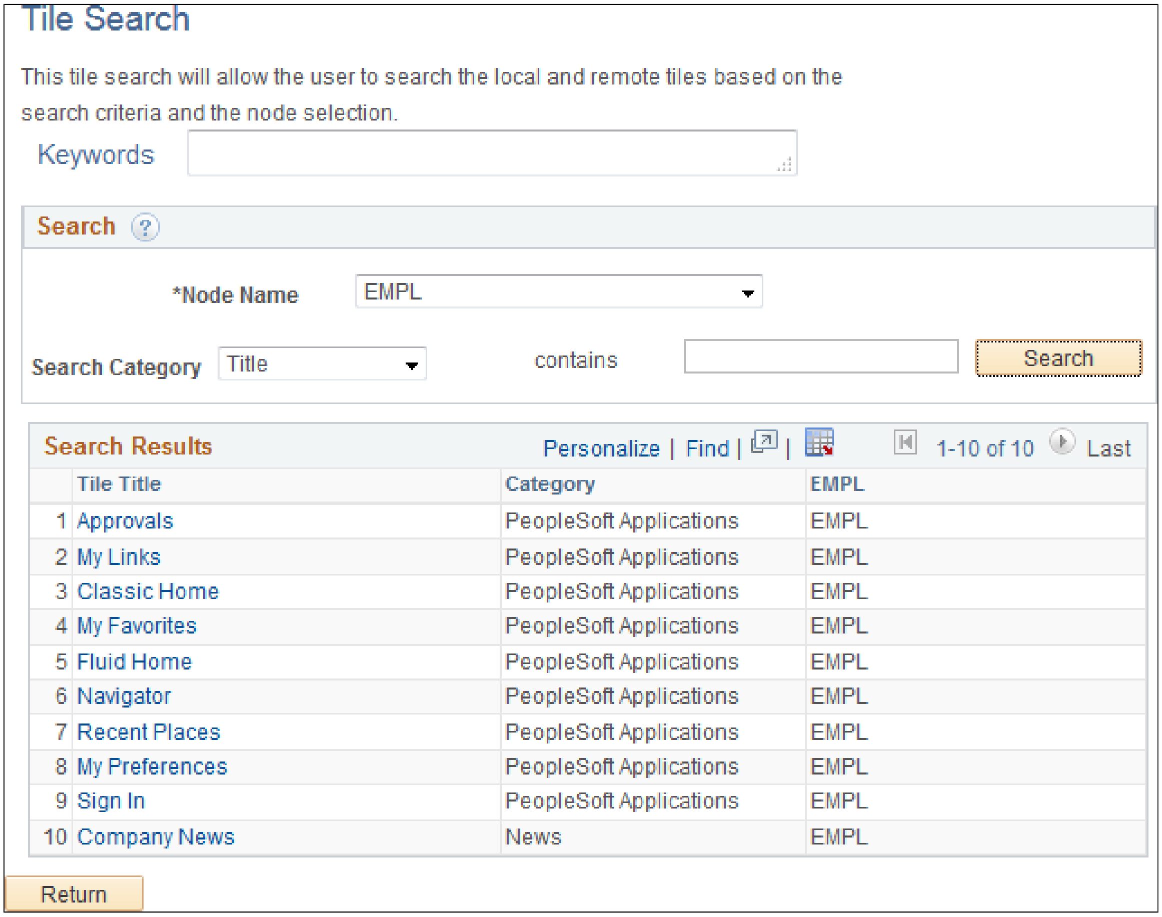
Use the Tile Search page (PTUN_GPLTSRCH_SEC) from an invoking definition (homepage or dashboard) to search for tile definitions on local and remote content provider nodes.

Navigation:

To access the Tile Search page:

- Click the Tile Search button on the:
  - Dashboard’s Tab Content page.
  - Homepage’s Tab Content page.
- Click the tile Select icon button on the:
  - Define Related Content Service page.
  - Assign Related Content page.

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



Field or Control	***Description***
Keywords	Displays any keywords defined on the invoking definition as a display-only value—for example, a dashboard definition can include keywords. **Note:** If keywords are defined on the invoking definition, then this field is populated, the Search by field is set to *Keywords,* the contains field is populated with the same keywords, and the search is automatically executed. You can update the search parameters, such as the contains field, to perform a different search after this initial keyword search.

Search

Field or Control	*Description*
Node Name	Select the local or remote content provider node. **Important!** While this field displays local and remote *default local nodes* and *portal host nodes,* you should select portal host nodes only (that is, content provider nodes) for tile searches.
Search by	Select one of the following: • *Category* – To search by the title of the tile folder. • *Title* – To search by the tile title.
contains	Enter a word, partial word, phrase, or partial phrase as the search term. **Note:** Clearing this field and performing a search will return all results displaying valid values for the selected Search by value.
Search	Click the Search button to perform a search.
Return	Click the Return button to return to the invoking page without selecting a tile definition.

Search Results

Field or Control	*Description*
Tile Title	Click a tile title to select and add that definition to the invoking definition.

Importing Pagelet and Tile Definitions

You can search for pagelets or tiles in other databases (remote content provider nodes) only after unified navigation has been configured. When you select a remote pagelet on the Pagelet Search page or a remote tile on the Tile Search page, you import the pagelet or tile definition into the local database. This imported definition is then available to all features that use pagelets or tiles, such as homepages, dashboards, WorkCenters, and so on.

For more information on unified navigation and importing pagelets and tiles:

- See Understanding Unified Navigation for an introduction to unified navigation.
- For information on other methodologies on importing pagelets and tiles including importing pagelets and tiles in bulk, see Searching for and Importing Pagelets and Tiles.

Managing Section Templates and Sections

Section templates and sections are available only for fluid homepages and dashboards. All the content discussed in this topic is applicable for fluid homepages and dashboards.

This section discusses how to:

- Create section templates.
- Import sections in homepages and dashboards.
- Hide tiles in homepage personalization.
- Configure tiles for an empty section.
- Guidelines for sections in a cluster.

Related Links

(Fluid only) Creating Sections on Fluid Homepages

Creating a Section Template

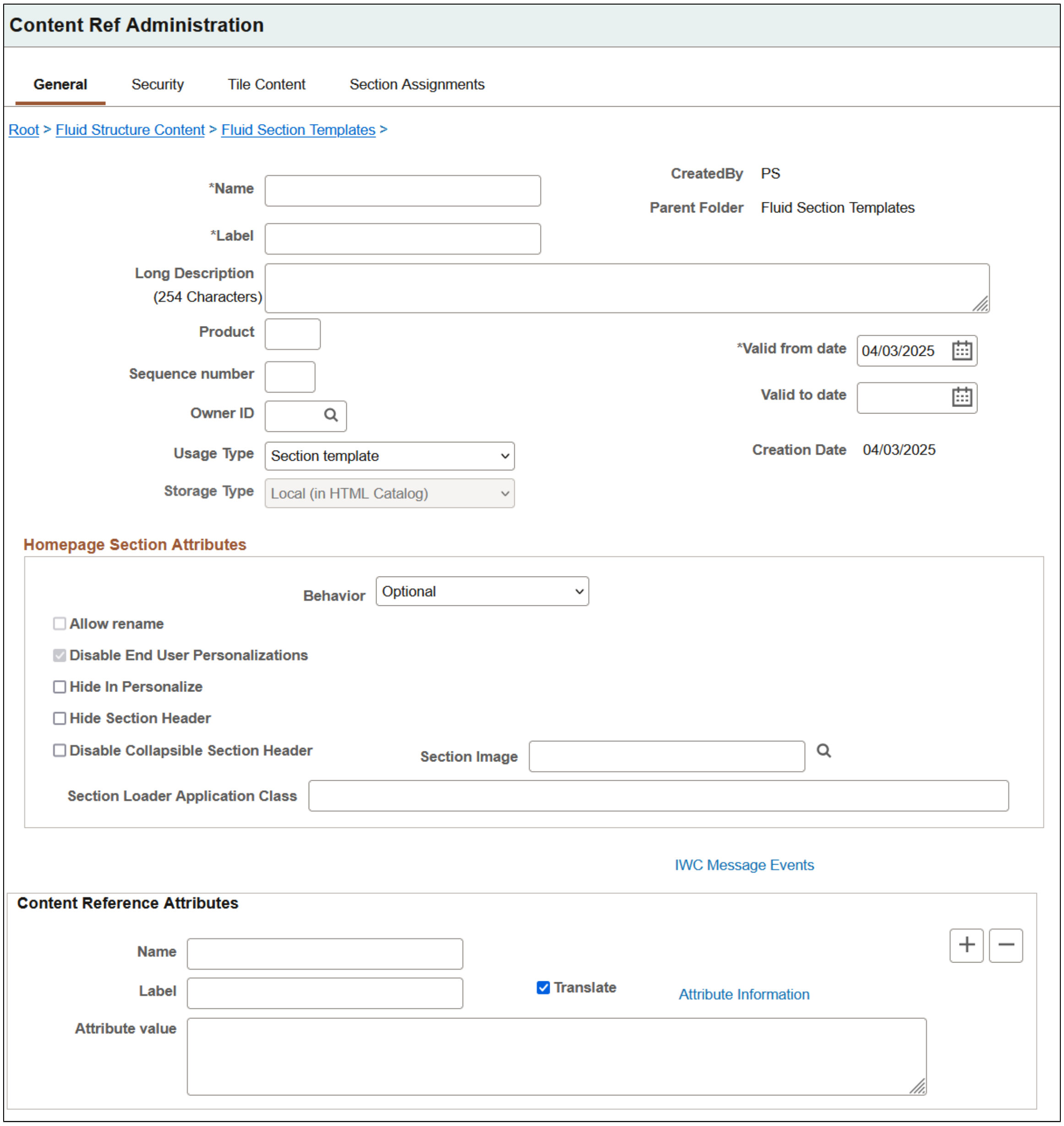
A section template enables you to copy a predefined section definition into a fluid homepage or dashboard. A section template defines a single section definition and is stored in the Fluid Section Templates folder (in the Fluid Structure Content folder).

You use the Content Ref Administration page (PORTAL_CREF_ADM) to create and maintain section template definitions.

Navigation:

1. Navigate to **PeopleTools** > **Portal** > **Structure and Content**.
2. Select **Fluid Structure Content** > **Fluid Section Templates**.
3. Click the Add Content Reference link.

This example illustrates the fields and controls on the Content Ref Administration page for a section template definition.



4. On the General page, enter the necessary information.

   In the **Usage Type** field, select Section template.

   In the Homepage Section Attributes area, the following options are available:

   - **Behavior**: You must select a behavior type other than Optional.
   - **Allow Rename**: This option is unavailable for entry.
   - **Disable End User Personalizations**: This option is selected automatically.

- **Hide in Personalize**: Select this option to prevent this section from appearing on the homepage personalization page.
- **Hide Section Header**: Select to not display the text in the section header including the image, if specified.
- **Disable Collapsible Section Header**: When you select this option, the section header cannot be collapsed, that is, the section header remains expanded. To display the section image correctly, this option should be set.
- **Section Loader Application Class**: Enter the full local path of the application class used to load tiles dynamically in the associated section. It is an extension to the base class PTNUI:Dynamic:DynamicLoader. For more information, see Designing Sections and Tiles and “DynamicLoader Class Methods” (PeopleCode API Reference).
- **Section Image**: Select an image to display in the section header. The image should be displayed beside the section title.

See Creating and Configuring Homepage Definitions.

5. In the Content Reference Attributes area, specify attributes that are applied at the section level. For a listing of attributes for section level, see Applying Styles from Delivered Style Sheets and Styling Homepages and Dashboards Using CSS Variables.
6. On the Security page, set the access permission.

   Note: Access permission is used to allow a section template to appear on the import list, and it determines who can search for section templates and import them. Access permission does not set up the permission of the section once imported.

   See Setting Homepage Permissions.
7. On the Tile Content page, select the tiles that you want for this section. You must select a behavior type other than Optional.

   See Selecting the Pagelets or Tiles That Appear on Homepages.

   If you're creating a section to dynamically load tiles through an application class, you do not have to select tiles on this page. See Creating Tiles for more information.
8. On the Section Assignments page, a section is created and all the selected tiles are automatically listed in the Tiles grid. The system attempts to place the tiles with the Req-Fix behavior type first in the section.

   Also note that the Allow Rename, Disable End User Personalizations, and Hide in Personalize options are disabled.

   Additionally, the Hide Section Header, Disable Collapsible Section Header, Section Image, and Loader Class options are disabled.
9. Save the section template. A section template defines a single section.

Note: Although you can save the section template without navigating to the Section Assignments page, it is not recommended because the tile order is based on the system. As an administrator, you may want to decide the tile order and the Section Assignments page enables you to specify the tile order.

The section template is stored in the Fluid Section Templates folder and can be imported in a homepage or a dashboard.

After the section template is saved, the **Save As** button appears on the General page. Use the Save As button to clone the section template. It provides administrators a starting point to build upon the delivered section template. On saving the cloned section template, administrators can navigate to the newly saved section template.

You may edit the section template to add tiles, remove tiles, set the tile behavior, or modify the order. However, these changes are not automatically propagated to the homepages where this section is used. If you want to propagate these changes in the homepages where the section is used, you have to import the section again in that homepage.

To import a section, see Importing a Section in Homepages and Dashboards.

Importing a Section in Homepages and Dashboards

Importing a section means copying a template's section definition into a fluid homepage or dashboard. You can either manually create a section and assign tiles or you can import a pre-defined section, which is stored in the Fluid Section Templates folder.

You should note that the import process creates a section with tiles within your homepage or dashboard based on the section template at the time of import. If changes are made to the template, you should re-import that section in each homepage where you had imported the section template, that is, there are no automatic updates. Once imported, tiles are not re-imported so that the settings specific to that homepage are maintained. However, new tiles are added.

If you change the section ID after importing the section into a homepage or dashboard, you can no longer re-import the section template because with the change in section ID it is an independent section.

Note: Prior to importing a section, you should ensure that section templates are available in the Fluid Section Templates folder. For more information, see Creating a Section Template.

Complete the following steps to import a section:

1. On the Section Assignments page, click the **Import Section** button.
2. The Find Section Template page appears, which enables you to search for section templates that are stored in the Fluid Section Template folder.

   Use the section template label, description, or wildcards such as % (percent) and _ (underscore) to search. Section template security is applied to display the search results based on the user's permissions. Section template security is not applied to a section after it has been imported. Sections are driven by the security applied to a tile definition. If you do not have permissions to a tile, that tile is not displayed in the section.

The search results display the label of the search template, indicates whether the section exists on a homepage, the number of tiles it contains, and the description of the search template.

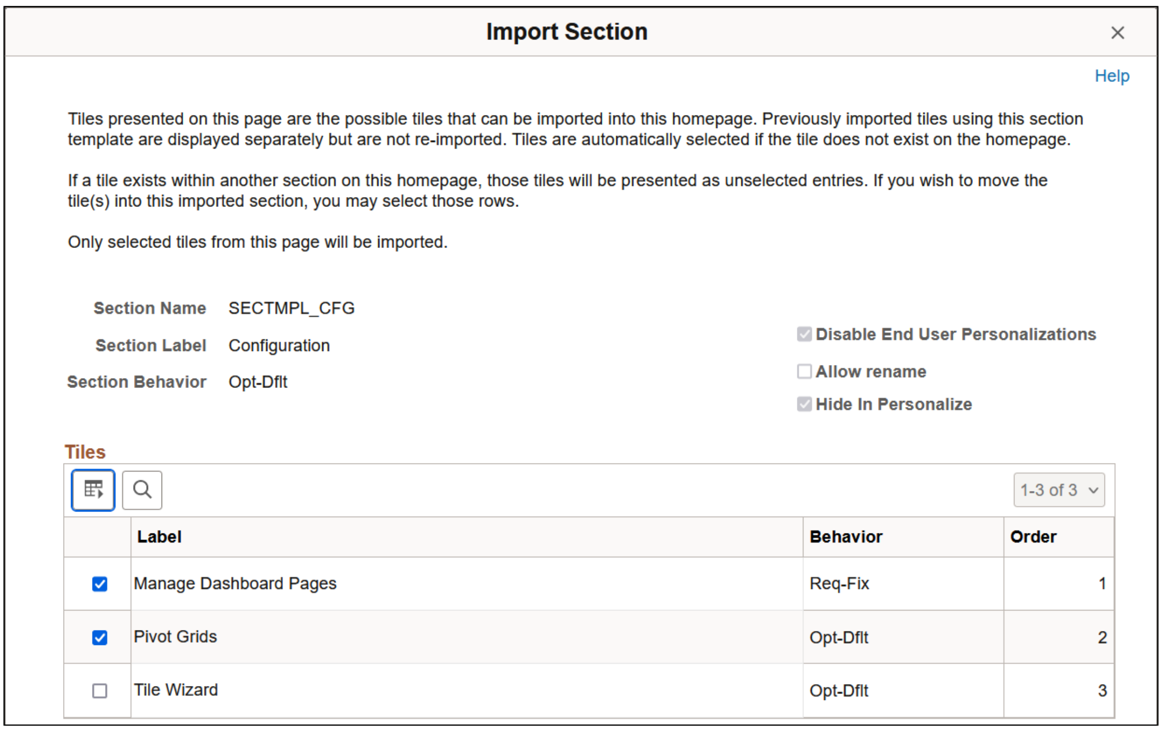
3. In the Section Template Results grid, click the section template you want to import.

4. The Import Section page lists the tiles that the section contains and includes tile information, such as tile behavior type and order number. However, you cannot edit the tile behavior or order number here.

   Note the following:

   - The system automatically selects the tiles that are not present on the target homepage or dashboard.
   - If the section template contains tiles that are already present in sections on the target homepage or dashboard, the tiles are not automatically selected for import. If you select tiles that are already part of a section, then the tiles are moved to the imported section.

   This example illustrates the fields and controls on the Import Section page.



5. Select the tiles that you require and click the **Import** button. Only the tiles that are selected and have a behavior type other than Optional are imported.

   If the homepage contains sections, the imported section is inserted after the existing sections. However, you can modify the section order number. When you import a section, the section order number is not imported.

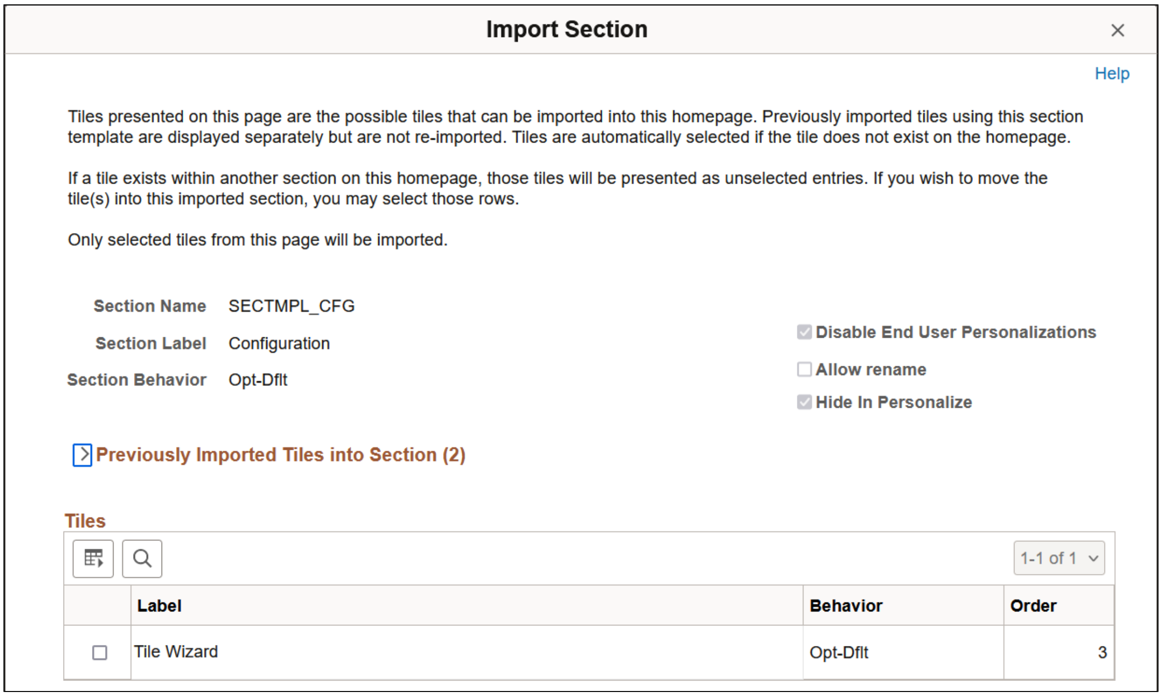
   In the imported section, you may change the tile behavior type and modify the order.

6. Save the homepage or dashboard.

   You can select the same section template to import tiles that you may not have imported. In this case, the Import Section page indicates the tiles that were previously imported in the **Previously Imported**

Tiles into Section group box, and you cannot import them again. Expand the **Previously Imported Tiles into Section** group box to see the list of previously imported tiles.

This example illustrates the re-import of section on the Import Section page.



Hiding Tiles in Sections on Homepage Personalization

You can hide a tile that is assigned to a section on the Personalize Homepage page.

Use the HIDE_IN_PERSONALIZE content reference attribute in the tile definition and set the value to true. This attribute enables an administrator to hide tiles so that users cannot copy these tiles to another homepage because it may not work in that context.

A tile with the HIDE_IN_PERSONALIZE attribute set to true possesses the following characteristics:

- It does not appear in sections on the Personalize Homepage page.
- It is not listed on the Add Tile page.
- It is not designed to work in a traditional homepage or in the traditional area of a sectionalized homepage.

Note: If you set the HIDE_IN_PERSONALIZE attribute for all the tiles in a section, it is recommended that you set the **Hide in Personalize** option for the section as well.

Configuring Tiles for an Empty Section

By default, an empty section or a section without any conventional tiles assigned to it is not displayed on a homepage. However, you can configure special tiles such that these special tiles are displayed in a section only when other conventional tiles are either not assigned to the section or the visibility of a tile is controlled through PeopleCode.

A section can be empty due to runtime characteristics, for example:

- You do not have permissions to the tiles in a section or tiles are hidden through the TILEAPPFILTER attribute. In this scenario, the tiles are not displayed. Consequently, the section is not displayed.
- Not all tiles in a section are available or optimized for small form factor devices. If you're using a small form factor device, then these tiles are not displayed and the section is hidden.

Use the EMPTY_SECTION_TILE content reference attribute in the tile definition and set the value to true. Administrators can use a tile with this attribute to display the tile as an actionable tile or a non-actionable tile in a section.

Note: When you use the EMPTY_SECTION_TILE content reference attribute in a tile definition, you do not have to set the HIDE_IN_PERSONALIZE content reference attribute for the tile because the tile with the EMPTY_SECTION_TILE attribute is hidden in the homepage personalization page and not listed on the Add Tile page.

A tile with the EMPTY_SECTION_TILE attribute set to true possesses the following characteristics:

- It does not appear in sections on the homepage personalization page.
- It is not listed on the Add Tile page.
- It should never be used in a traditional homepage or the traditional area of a sectionalized homepage.
- It can be an actionable tile or a non-actionable tile.
- It is mutually exclusive with other tiles in a section. That is, this tile appears only when no other tile is present in a section or the tiles are not visible in a section.
- The tile behavior type should be set to Req-Fix. It is recommended that this tile should be placed as the first tile in the section (as a best practice).
- The section that contains the empty-section tile should be also hidden from personalization, so you should select the Hide in Personalize check box at the section level on the Section Assignments page of the homepage definition.

Note: If you use the empty-section tile and you use the TileAppFilter.IsTileVisible method to control the visibility of conventional tiles in a section, you can configure only tiles of the local node; you cannot configure tiles of a remote node because of a limitation of the TileAppFilter.IsTileVisible method.
See “TileAppFilter Class Methods” (PeopleCode API Reference).

Configuring an Actionable Tile for an Empty Section

An actionable tile (a tile that can be clicked to perform navigation) is similar to a conventional tile, that is, it has all the characteristics of a conventional tile except the content reference attribute area, where you specify the EMPTY_SECTION_TILE attribute and set it to true.

Navigation:

1. Navigate to the tile repository folder in Fluid Structure Content.
2. Create a conventional tile.

3. On the General page, in the Content Reference Attributes area, in the **Name** field, enter EMPTY_SECTION_TILE.

4. In the **Attribute value** field, enter TRUE.

5. Deselect the **Translate** check box.

6. Similar to a conventional tile, you may set permissions or use the TILEAPPFILTER content reference attribute to hide or display the conventional tiles.

7. Save the tile content reference.

8. Assign this empty-section tile to a section.

   Note: Ensure that you assign this tile as the first tile in a section with the behavior type set as Req-Fix.

   If you want to configure multiple empty-section tiles for a homepage, you should configure different tiles because a tile can appear only once on a homepage.

Configuring a Non-actionable Tile for an Empty Section

A non-actionable tile (does not perform any action; used only for display purpose in an empty section) does not look or act like a conventional tile. The setup is similar to that of a conventional tile except the following:

1. Navigate to the tile repository folder in Fluid Structure Content.

2. Create a conventional tile.

3. On the General page, in the Content Reference Attributes area, in the **Name** field, enter EMPTY_SECTION_TILE.

4. In the **Attribute value** field, enter TRUE.

5. Deselect the **Translate** check box.

6. Similar to a conventional tile, you may set permissions or use the TILEAPPFILTER content reference attribute to display or hide the conventional tiles.

7. On the Fluid Attributes page, select the **Disable Main Hotspot** check box.

8. Optionally, in the **Image Name** field, choose PT_EMPTY_GRID_ICN.

9. Save the tile content reference.

10. Assign this empty-section tile to a section.

    Note: Ensure that you assign this tile as the first tile in a section with the behavior type set as Req-Fix.

    If you want to configure multiple empty-section tiles for a homepage, you should configure different tiles because a tile can appear only once on a homepage.

When these empty-section tiles are rendered to an end user, you can note the following characteristics:

Actionable Tile	***Non-actionable Tile***
The specified label of the tile is displayed.	The specified label of the tile is *not* displayed. The label is used as the name of the region that surrounds the empty section pattern. The label is used for screen reader purposes only as a navigational shortcut.
The border of the tile is displayed.	The border of the tile is *not* displayed.
Similar to conventional tiles, actionable tiles honor the width and height values.	While the defined height of the tile is honored, the width is ignored. The width of this empty-section tile spans the width of the tile area, that is, from the position of the first tile to the position of the last possible tile in the defined space.

Improvements to TILEAPPFILTER to Support Empty Sections on a Homepage

Displaying empty-section tiles depends, to a certain extent, on the ability of applications to control the visibility of conventional tiles through the TILEAPPFILTER content reference attribute. By default, the TILEAPPFILTER setting is only evaluated the first time the homepage is displayed during a login session. From that point, it is cached. However, if you want that the application class pointed to by the TILEAPPFILTER setting to be re-evaluated every time the homepage is opened, you should use the TILEAPPFILTER_REFRESH content reference attribute and set its value to True. You should use the TILEAPPFILTER_REFRESH attribute in conjunction with the TILEAPPFILTER attribute that controls the visibility of tiles.

Warning! When the TILEAPPFILTER_REFRESH attribute in a tile is implemented by the system due to a change in the visibility of the tile, the homepage cache is invalidated, and the system reloads *all* homepages, so you must use this attribute with caution. It may cause performance issues because all the homepages are reloaded similar to the loading of homepages when you sign in to an application.

Important! Do not use the TILEAPPFILTER_REFRESH attribute in fluid dashboards. TILEAPPFILTER_REFRESH is not needed on fluid dashboards because their values are not cached like homepages. Dashboards already re-evaluate the TILEAPPFILTER attribute every time the dashboard is opened, so the TILEAPPFILTER_REFRESH attribute is not necessary.

See “TileAppFilter Class Methods” (PeopleCode API Reference) for the limitations of the TileAppFilter.IsTileVisible method.

Guidelines for Sections in a Cluster

This section provides guidelines to administrators on the behavior of sections and tiles in a homepage when these homepages appear in a cluster environment.

This section discusses:

- Sections and tiles on merged homepages.
- Application nodes on PeopleTools versions prior to PeopleTools 8.61.

Sections and Tiles on Merged Homepages

The rules that govern the merging of homepages in a cluster also apply to sections. That is, in a merged homepage, if sections with the same label exist, then the sections are merged. The rules that apply to a traditional homepage (a homepage without sections) continue to exist.

For example, if the HRMS application contains a homepage with a section labeled as Benefits and the FSCM application contains a homepage with a section labeled as Benefits, then the merged homepage displays the Benefits section as a merged section with tiles from both the application nodes.

On a merged homepage, the section order defined on the signed-in application is honored. It is recommended that administrators pay particular attention to the order number of sections such that users experience consistency when they sign in to any of the application nodes.

When you set the CLUSTER_AWARE content reference attribute for a tile, the system uses the tile from the local application node instead of the tile from the remote application node. In a merged homepage, the location of the tile with the cluster aware attribute is as defined on the local application node.

Application Nodes on PeopleTools Versions Prior to PeopleTools 8.61

When an application node running at least PeopleTools 8.61 receives a request to get its homepages and tiles, some homepages and tiles are not pushed out to remote nodes if the remote nodes are running PeopleTools versions prior to PeopleTools 8.61.

The following are not pushed out to remote nodes running PeopleTools versions prior to PeopleTools 8.61:

- Sectionalized homepages and dashboards.
- Homepages and dashboards that are disabled for end-user personalization.
- Empty-section tiles, that is, tiles with the EMPTY_SECTION_TILE content reference attribute.
- Tiles hidden from homepage personalization, that is, tiles with the HIDE_IN_PERSONALIZE content reference attribute.

Troubleshooting Homepages, Dashboards, and WorkCenter Pages

Under most circumstances, PeopleTools portal technology assembles homepages, dashboards, and WorkCenters by simultaneously retrieving content for each of the pagelets referenced on the page. As a result, the portal renders the page itself as soon as the user accesses it and immediately begins the deferred loading of individual pagelets.

Pagelet Loading

Portal homepages, dashboards, and WorkCenter pages defer-load each pagelet individually after you access the page. Each pagelet appears as you see in this example, indicating that it's currently loading its data.

The following example illustrates a pagelet that is loading.

Modifying Default Pagelet Behaviors

In some cases, a pagelet may take awhile to load. Some reasons for this could be the URL for the pagelet is incorrect or has changed, DNS can't resolve the URL, the content server is down or unreachable, the content server is behind a firewall, or the content server is temporarily unavailable.

You can modify unavailable pagelets by specifying a time-out period during which the pagelet must be loaded. To specify a pagelet time-out, add the PSTIMEOUT attribute to the content reference for the pagelet in the portal registry.

See Configuring Pagelet Time-out Settings.

In other cases, when **document.write** or **document.writeln** commands are detected inside a pagelet, the whole page will be reloaded again and all pagelets are then loaded sequentially. In this case, to properly assemble the content of the pagelet in question and the whole page, none of the pagelets can be defer-loaded individually. To avoid reloading the page multiple times and to allow other pagelets to be defer-loaded, you must add a content reference attribute to the pagelet in question.

To specify that a specific pagelet with **document.write** or **document.writeln** commands be loaded in the non-default manner:

1. Select PeopleTools, Portal, Structure and Content, Portal Objects, and navigate to the Content Ref Administration page for the pagelet.
2. In the Content Reference Attributes region of the page, enter *PTPGLTDEFERLOAD* in the Name field.
3. Deselect the Translate option.
4. In the Attribute Value field, enter *FALSE.*
5. Save the content reference.

Implementing Guest User Access

While no guest user ID is defined and delivered with PeopleTools, PeopleTools does deliver the following definitions to assist with creating guest user access:

- PeopleSoft Guest role.
- PTPT1400 (PeopleSoft Guest) permission list.

The PTPT1400 permission list provides access to the following:

- The default fluid homepage (DEFAULT_LP: My Homepage).

  Note: Because permissions are set to public on the default classic homepage (DEFAULT: My Page), a guest user also has access to this classic homepage.

- The NavBar, the Classic Home button, the Fluid Home button, the Navigator, and any menu folders and menu items that PTPT1400 has been assigned to or that are designated as public.
- Any fluid dashboard that PTPT1400 has been assigned to or that is designated as public.

  Note: The PTPT1400 permission list has access to the underling fluid dashboard component (PT_FLDASHBOARD).

- The Sign In tile and the Sign In pagelet.

On the other hand, the PTPT1400 permission list does not provide access to:

- Personalizing any fluid or classic homepages including dragging and dropping tiles or pagelets, or personalizing the NavBar.
- Add To links, the Notifications feature, and the My Preferences link.
- The Forgot Password component (from the Sign In tile or Sign In pagelet).

While certain PeopleSoft applications are delivered with a defined guest user ID and guest user access enabled, and a guest homepage definition (PeopleSoft Interaction Hub, for example), others are not. In addition, you may want to implement your own guest user access and homepages.

To implement guest user access:

1. (Optional) Clone the PTPT1400 permission list if you wish to customize the guest user experience.
2. (Optional) Change access permissions in your custom guest permission list.

   For example, to hide the Sign Out link in the Actions menu for guest users, grant full access to the WEBLIB_PORTAL.PT_PORTAL_HDRLINK.FieldFormula.IScript_HideSignOutLink function.
3. (Optional) Clone a custom guest role if you created a custom guest permission list. Assign your custom guest permission list to this custom guest role.
4. Define a guest user ID—for example, MY_GUEST.
5. Add either the delivered PeopleSoft Guest role or your custom guest role to your guest user ID.
6. Update the active web profile to enable public user access:

   Note: Enabling public user access means that public (guest) users do not have to explicitly sign in using the guest user ID and password.

a. Select PeopleTools, Web Profile, Web Profile Configuration.

b. Select the active web profile.

c. Select the Security page.

d. Select the Allow Public Access option.

e. Set the user ID to your guest user ID.

f. Set the password to your guest user's password.

g. Set the HTTP Session Inactivity field to value that specifies the interval of inactivity after which time the web server releases the guest user's HTTP session. (This setting merely prevents an overload of web server resources for inactive public users, and does not sign guest users out.)

h. Stop and restart the web server.

7. Create a fluid homepage definition, a classic homepage definition, or both specifically for guest users. Add the PTPT1400 permission list or your custom guest permission list to each guest homepage definition.

   - Configure each guest homepage with the tiles and pagelets that are deemed suitable for public access.

   - Optionally, add the Sign In tile or Sign In pagelet to each guest homepage thereby allowing defined system users to sign in from the guest homepage.

8. Add guest user access to other public components of your site. For example, add the PTPT1400 permission list or your custom guest permission list to fluid dashboards that are suitable for public use.

9. Distribute the URL to the guest homepage for use by the public.

   - For example, for a classic guest homepage: http://myserver.example.com:8010/psp/MY_SYS/EMPLOYEE/MY_NODE/h/?tab=GUEST

   - For example, for a fluid guest homepage: http://myserver.example.com:8010/psp/MY_SYS/EMPLOYEE/MY_NODE/c/NUI_FRAMEWORK.PT_LANDINGPAGE.GBL

   Note: Unlike a classic guest homepage, there is not a distinct URL for each fluid homepage. The guest user will see the default homepage and any other fluid homepages to which the user has been granted access.

Note: When configured with the Sign In tile or Sign In pagelet, a guest homepage can also serve as an alternative signon page for defined system users.

Related Links

(Fluid only) Restricting Fluid Homepage and NavBar Personalization

Chapter 14

Developing Advanced Sectionalized Homepages and Dashboards

Designing PeopleSoft Landing Page

PeopleSoft Landing Page provides a tailored one-stop starting experience for every user and enables them to seamlessly navigate to relevant content and operational areas within the applications. This concept of a single landing page enables you to transition to a more simplified navigation paradigm, as an alternative to navigating across various traditional homepages, which are primarily role focused.

While a PeopleSoft Landing Page is a new and improved navigation concept, it is simply a homepage with sections, tiles, and additional styling taking advantage of the new layout and display capabilities offered through advanced sectionalized homepage development in PeopleTools 8.62. A landing page can be designed to present tailored content to the user based on their roles, recent activities, and so much more.

The development and design of PeopleSoft landing page is similar to the development of a sectionalized homepage, so you need to use the same pages (homepage CREF, tile CREF) in PIA that you have been using to create sectionalized homepages, section templates, and tiles. You can use styles and CSS variables for styling the homepage, sections, and tiles.

If the user has access to only a single landing page, the homepage selector (in PeopleTools 8.62.01 and previous versions it always displayed as a menu even for a single homepage) is converted to the simple text form of the landing page title. When the landing page is opened, initial focus is always placed on the homepage selector or title of the landing page for consistency of behaviors regardless of the number of homepages involved. However, unlike the homepage selector (which is an actionable menu) when a single landing page is involved, once the user navigates away from the landing page title, the user cannot set focus back to that static title. The homepage selector behavior when multiple homepages are involved does not change.

The following table lists the tasks that you need to complete and the corresponding documentation in the Online Help to guide you. It is not necessary to follow the sequence followed here because you may want to create your section templates first or create the tiles first. This table provides a high-level list of tasks that are typically needed to create a landing page.

Tasks	***Online Help***
Familiarize with Advanced Sectionalized Homepage Development including the styles, CSS variables, and so on delivered in PeopleTools 8.62.	Understanding Advanced Sectionalized Homepage Development and the subsequent topics.
Create a homepage.	Creating and Configuring Homepage Definitions

Tasks	***Online Help***
Create section template or sections.	Creating a Section Template (Fluid only) Creating Sections on Fluid Homepages
Create tiles statically using tile CREF.	Using Tile Wizard Creating and Maintaining Tile Definitions Manually
Create tiles dynamically using PeopleCode.	Creating Tiles "DynamicLoader Class Methods" (PeopleCode API Reference)

Understanding Advanced Sectionalized Homepage Development

PeopleTools 8.62 provides flexibility to design the layout of homepages and dashboards, which enables you to modernize the look and feel of the homepages and dashboards. Customers and application developers have greater control over the layout and styling of homepages, dashboards, sections, and tiles.

The foundation of advanced sectionalized homepage development is the sectionalized homepages and dashboards that were introduced in PeopleTools 8.61. The PIA pages that you have been using to create sectionalized homepages and dashboards can be used for the advanced sectionalized homepages and dashboards.

Important! The development of advanced sectionalized homepage is supported only on sectionalized homepages or dashboards. It is not supported for non-sectionalized areas of a homepage or dashboard. If it is used outside a sectionalized area, it may produce undesired results. Moreover, you cannot have a mix of sectionalized and non-sectionalized areas on advanced sectionalized homepages or dashboards.

The portal-like look and feel for homepage or dashboard is implemented at the homepage styling level and is called *psc_hpmodel-A*. To implement it, you *must* apply this style for homepages and dashboards through the content reference attribute of PS_STYLECLASSES. It is important to apply the style because all other styles and CSS variables work only in conjunction with this new style. It is important that the Uses Custom Layout check box is also enabled when applying *psc_hpmodel-A* on the homepage or dashboard content reference (CREF). Otherwise, some layout issues may occur on small form factor. More information on this style and usage of CSS variables is discussed in later topics.

Oracle recommends that you must be familiar with the following CSS concepts to design the layout of advanced sectionalized homepages:

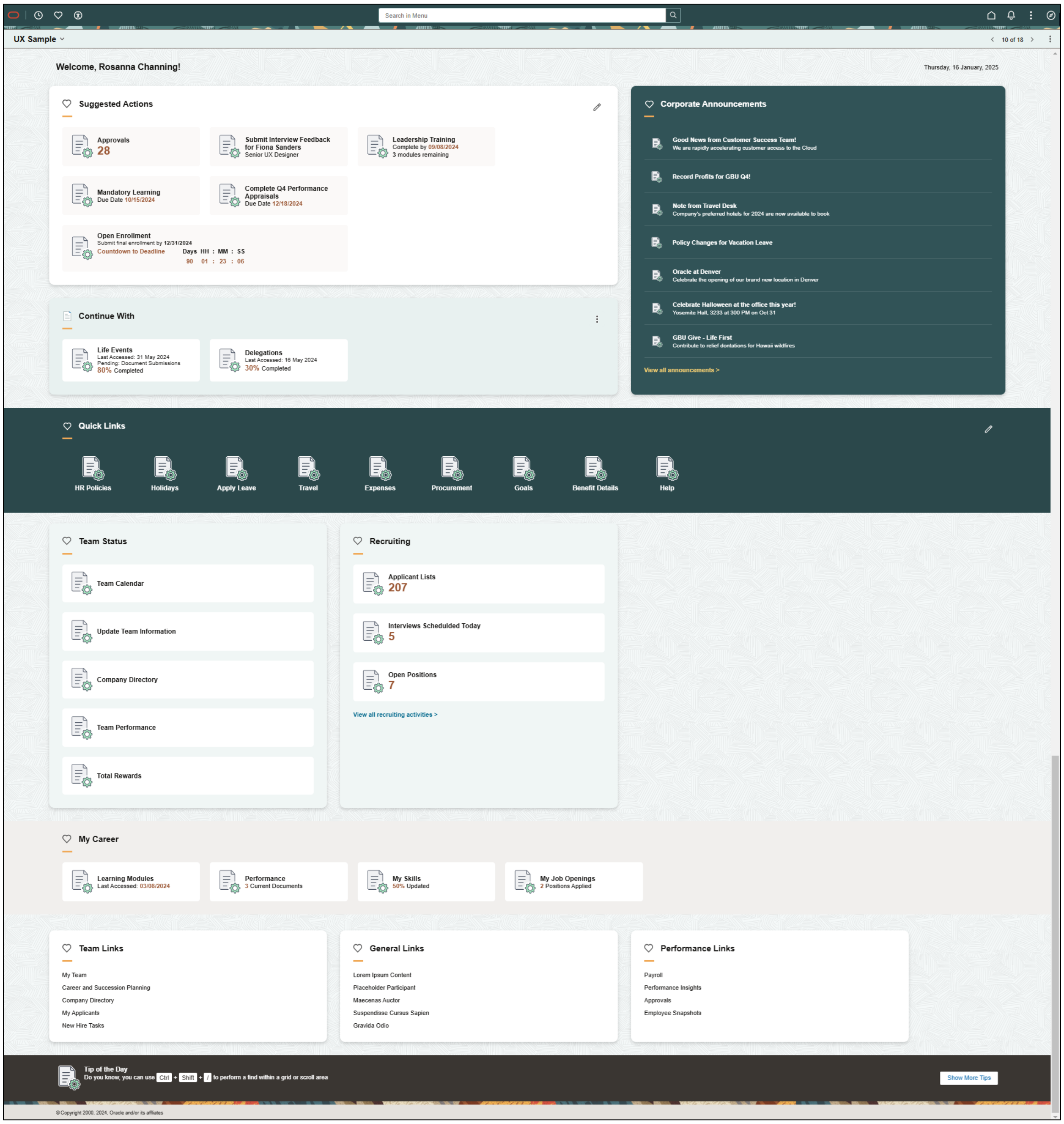
- Custom CSS Variables and Properties.
- CSS Grid and Subgrid Layouts.
- CSS Containers and Container Queries.
- Nested CSS Syntax.

The advanced sectionalized homepages and dashboards can contain sections, which are populated with tiles that are statically defined as content references (CREFs) and dynamic tiles, that is, tiles dynamically created and loaded into the associated section through the delivered PeopleCode class and methods. Oracle recommends that application developers create sections and tiles implementing the layout and styling that is delivered in PeopleTools 8.62 for the advanced sectionalized homepages or dashboards. It is also recommended that application developers create section templates with default styling and then import them into homepages or dashboards and apply specific styling as required for a homepage or dashboard.

If you plan to use tiles developed in previous releases in the advanced sectionalized homepages or dashboards, you must ensure that you implement the grouplet-based styling on these tiles as delivered in PeopleTools 8.62. For more information on grouplet-based styling, see Grouplet-Based Tile Styling.

The following screenshot is a sample of the advanced sectionalized homepage displaying the varied layouts of sections and different presentations of tiles.

Sample of advanced sectionalized homepage with sections and tiles



This topic discusses the following:

- Configure the advanced sectionalized homepages and dashboards.
- Apply styles from the delivered style sheets.
- Use CSS variables for the styling of advanced sectionalized homepages and dashboards.
- Design sections and tiles.
- Design PeopleSoft landing page.
- Apply guidelines for merging of homepages and sections in a cluster.

Configuring Advanced Sectionalized Homepage and Dashboard

To configure the advanced sectionalized homepage or dashboard, application developers and customers can apply stylesheets and style class attributes for homepages or dashboards and sections in the homepage or dashboard content reference (CREF). Styling for tiles is applied at the tile level in the tile CREF.

In addition to the attributes, you should also specify required configuration setting provided on the homepage or dashboard CREF and tile CREF pages.

Specifying Attributes for Homepages, Sections, and Tiles

You can use the following attributes at the appropriate levels to configure homepages, sections, and tiles.

Attributes	***Applicable Level***	***Description***	***Example***
PS_STYLESHEETS	• Homepage or Dashboard • Section Template • Section • Tile CREF	A semi-colon separated list of stylesheet names known by the local system to be loaded when attribute is found at the various levels. Note that stylesheets are loaded at homepage load time and will only be loaded once even if referenced by multiple sources.	CSS_SHEET1;CSS_SHEET2;CSS_SHEET3
PS_STYLECLASSES	• Homepage or Dashboard • Section Template • Section • Tile CREF	A semi-colon separated list of style classes to be applied at the assigned level, such as, homepage or dashboard, section, or tile. **Note:** At the homepage level, you must enter the *psc_hpmodel-A* value to apply the model delivered for advanced sectionalized homepage or dashboard.	psc_theme-primary-all-140;psc_theme-tile-transparent
PS_INCLUDE_SOURCE	Tile CREF	Use this attribute to implement a tile that needs to know the triggering source (homepage and section IDs) when the tile is selected (action URL). It is primarily used for implementing section action buttons on a section so that settings against a specific homepage or section can be stored by the application.	True

Specifying Settings on Homepage CREF and Tile CREF

The following settings enable you to set the behavior of certain aspects of homepages, sections, and tiles.

You should note that all the settings, which are introduced to support the advanced sectionalized homepage or dashboard, are not listed here. You can find descriptions for *all* the settings in the topics where the pages are described.

See Managing Homepages and Setting Fluid Attributes for Content References.

Control Name	***Applicable Level***	***Description***
Uses Custom Layout	Homepage CREF	Select this setting when you create an advanced sectionalized homepage. Selecting this setting prohibits the adaptive changes automatically performed by the system when running on a small form factor device making the responsibility of the custom layout to handle all visual changes for the small form factor device. Do not use this setting with pre-8.62 fixed tile size homepages or dashboards.
Hide Notification Panel	Homepage CREF	You can use this option to hide the notifications panel for all users when this homepage is displayed.
Section Loader Application Class or **Loader Class**	Section Template or Section	You have to specify the full local path of the application class used to load tiles dynamically in the associated section. It is an extension to the base class PTNUI:Dynamic:DynamicLoader. See Creating Tiles and “DynamicLoader Class Methods” (PeopleCode API Reference).
Identify Tile as a Button	Tile CREF	Use this setting to change a tile from being identified as a link to a button. This setting is typically used for a tile that you want to set as a section-action button. **Note:** This option applies to tiles within a section or outside a section on a homepage or dashboard. It does not apply to NavBar tiles. See Creating Different Presentations of Tiles

Control Name	***Applicable Level***	***Description***
Use Alternate Text on Tile Image	Tile CREF	Use this option to specify that the title of a tile (standard or long) will be used as the Title and Alt Text if a tile is rendered simply as a button with no text. For example, when you specify a tile as a button, you may want to display the alternate text when you hover on the button. **Note:** This option applies to tiles only within a section on a homepage or dashboard. It does not apply to NavBar tiles. See Creating Different Presentations of Tiles
Disable Accessible Description as Tile	Tile CREF	Select to disable the tile accessible description. This setting is typically used in conjunction with the Identify Tile as Button setting so that the tile is no longer identified as a tile. Also, you can use this setting when you are configuring a tile as a link. **Note:** This option applies to tiles within a section or outside a section on a homepage or dashboard. It does not apply to NavBar tiles. See Creating Different Presentations of Tiles.

Applying Styles from Delivered Style Sheets

Styles specific to *psc_hpmodel-A*, which is implemented for advanced sectionalized homepages and dashboards, are available in the PT_HP_CUSTOM_PROPS (PSSTYLEDEF_FMODE) and PTNUI_LANDING_CSS style sheets.

The following table lists the style naming conventions for the *psc_hpmodel-A* styling. You should note that any style that denotes being available for a section is also available within a section template. Similarly, any style that denotes being available for a homepage is also available for dashboards.

Style Pattern	***Description***
psc_theme-*	Typically involves section and tile colors.

Style Pattern	*Description*
psc_theme-primary-*	Maps the primary color palette from it's current value to the "*" color palette.
psc_theme-section-*	Assigns colors for a section.
psc_theme-section-primary-*	Assigns the colors for tiles that are found in the section having the color palette number of *.
psc_theme-section-all-primary-*	Assigns colors for both the section and the tiles found within the section or section template using the primary palette.
psc_theme-tile-*	Assigns colors for a specific tile (if applied at the tile level) or all tiles if applied at the section or section template level.
psc_theme-tile-for-primary-*	Assigns tile colors from primary which are the standard for given types of sections. The * (asterisk) in this case is for the palette color number of the section or section template.
psc_section-*	Typically defines some behavior or characteristic to be applied to a section or section template.
psc_tile-*	Typically defines some behavior or characteristic to be applied to a tile or all tiles within a section or section template.

Specific Color Based Styling

In continuation of the Redwood color scheme for PeopleSoft applications that was introduced in PeopleTools 8.60, new color palettes are added in the PT_CUSTOM_PROPS (PSSTYLEDEF_FMODE) style sheet.

A primary theme with the style pattern: *--pt-theme-primary-#* and a neutral theme with the style pattern: *--pt-theme-neutral-#* are added. In PeopleTools 8.62, the primary theme points to the equivalent teal color palette from Redwood.

The palette color ranges from 1 and 10 to 190 with the color numbers having a *p* suffix, which denotes that these are special PeopleSoft versions of the Redwood color.

The following table lists a few of the color numbers.

Style Class	*Applicable Level*	*Description*
psc_theme-section-primary-1	Section Homepage	Setting a section to the primary 1 color (which is typically white background)

Style Class	*Applicable Level*	*Description*
psc_theme-section-all-primary-1	Section Homepage	Setting a section and tile content to the white with neutral (10) tiles.
psc_theme-tile-for-primary-1	Section Tile Homepage	This style class provides the ability to use the tile color from a different section color. For example, if you want the tile color from the styling of primary-all-1 on a primary-130 section, you could specify psc_theme-section-primary-130 (only the section color specifications) and psc_theme-tile-for-primary-1 (use the neutral 10 color as if the section were in a primary-1 section).

Styles for Layout and Behavior

Use these styles for layout and behavior styling of homepages, sections, and tiles.

Style	*Applicable Level*	*Description*
• psc_branding-bypassall • psc_branding-bypass	Homepage	Use these styles for homepages that are *not* marked as custom layout. These styles block global homepage branding from being applied to the interior of the homepage. While *psc_branding-bypass* blocks branding only on sections and tiles, but allows branding on the homepage; the *psc_branding-bypassall* blocks branding settings on the homepage background itself. **Note:** By default, any homepage marked as custom layout does *not* get homepage and tile branding applied automatically.
• psc_branding-apply • psc_branding-backgroundonly	Homepage	Use these styles for homepages that are marked as custom layout. It applies the branding as specified for the custom-layout homepage. While *psc_branding-apply* applies all branding to the homepage, *psc _branding-backgroundonly* applies branding only to the homepage background.

Style	***Applicable Level***	***Description***
psc_homepage-no-bottom-spacing	Homepage	Removes the bottom spacing from the homepage by setting the CSS variable *--pt-homepage-bottom-spacing* to 0px. It is useful for creating a footer on a homepage where the bottom section is the footer section.
psc_section-compression-none	Homepage	Disables the automatic auto-adjustment of sections when the landing page falls below 453 pixels. This style allows a developer to create their own responsive behavior (typically when the breakpoint of 453 pixels is not correct for their landing page implementation). This section compression reduces the section internal padding, sets the gutters of the homepage to 0 pixel, makes all sections full width sections (meaning it spans the gutter, and is square, has no shadow, reduces interior section spacing, and reduces gaps between sections). For more information, see Automatic Section Adjustments on Small Form Factor Devices.
psc_section-height-fit	Homepage Section	Causes the visual bounding box of a section not to default to the size of the CSS grid style (so that sections are perfectly aligned for dependent sections but instead it shrinks the height of the section within the grid cell to only be large enough to show the content (empty area at the bottom of a section would not be shown). In this case, the alignment of dependent sections may not be consistent but it removes the blank area of the section. Note that this does not affect actual LAYOUT so there is no changes to how many rows the section consumes so no sections can be moved to recover the gap. If applied at the homepage level, all sections are set to FIT (shrink to content). If set at the section level, only that section is affected

Style	***Applicable Level***	***Description***
psc_tile-height-fit	Homepage Section Tile	By default, a tile's boundary (visual height) is defined by the grid layout. If other tiles are larger on the row, it will also affect the height of all tiles in that row. The psc_tile-height-fit style allows the tile height to fit the content (down to the minimum tile height as defined by tile collection used). Just as with sections, it potentially reduces the blank space found in the tile but it does not affect the layout of tiles among other tiles (other tiles cannot fit within the reclaimed space). It only affects the visual sizing of the tile. If applied at the homepage level, it affects all tiles on the page. If at the section level, it applies to all tiles within the section. If applied at the tile level, it affect the height of only one tile. This still will not affect autosizing tiles.
psc_section-noimage	Section	Hides the image for the section header (typically used only for sections that have the **Disable Collapse** check box selected).
psc_section-square	Section	Removes the rounded corners of the section.
psc_section-noshadow	Section	Removes the shadow from around the section.
psc_section-full-width	Section	Makes a section span the entire columnar width of the homepage's CSS grid consuming the default gutters on each side. Additionally, the rounded corners and shadows are removed for these type of sections automatically.
psc_section-padding-none	Section	Sets the --pt-section-padding variable to 0px to remove all internal padding to the section. Typically used when the header is not displayed for a section and the tile is responsible for the actual layout of the interior.
psc_section-height-auto	Section	If a homepage has been set to fit with psc_section-height-fit, the psc_section-height-auto allows for the reverting to the default sizing of the specific section to match the grid layout of the section.

Style	***Applicable Level***	***Description***
psc_section-content-scrollable	Section	Enables a developer to restrict the size of the content of a section (does not include the section header or padding) and make it scrollable given a maximum height (width is also possible but has fewer use cases). By default, merely adding this style to a section does nothing visually. You must also use new CSS variables specifically for setting the maximum size of the content. --Max Height: --pt-section-content-max-height (default is none)--Content Padding: --pt-section-content-padding (defaults to 0px)Max Height is the maximum display size of the content. Content padding is important for tiles configured with shadows because the shadows fall outsize the tile and can be clipped by this scrollable content setting. Should be set only if you style tiles with shadows (psc_tile-shadow).
psc_tile-full-width	Tile	Makes a tile span the entire columnar width of the section's CSS grid.
psc_section-elastic	Section	Makes all tiles within the section elastic, that is, the tiles can grow to consume the entire width of the section container. However, the width can be restricted if the application uses custom stylesheet and sets the *--pt-tile-max-width* CSS property to the desired value. At that point, the content will be centered within the container, by default.
psc_tile-morph-standard	Section	Makes tiles within the container to morph or behave like elastic tiles when the size of the tile specified in the tile width value attribute (either width 1 or width 2) is reached. This style should not be used with *psc_section-elastic* or *psc_section-simple-list-collection*. Tile morphing occurs on changes to the section container (not the overall window size). The break points are at 692px (for 2 wide tiles) and 358px (for 2 and 1 wide tiles). This allows tiles to shrink below the minimum defined size.
psc_tile-simple-link	Tile	Converts a single tile into a simple hyperlink. You can use the *--pt-tile-font-weight* CSS variable to set the bold font for the hyperlink.

Style	***Applicable Level***	***Description***
psc_section-link-collection	Section	Converts all tiles found within the section into hyperlinks; a collection of links displayed vertically.
psc_section-simple-tile-collection	Section	Changes layout of tile and only renders the tile image (on top) and tile label (below the image). By default, the tile label does not wrap. However, you can set the *--pt-tile-max-width* CSS variable on this section through custom styling to restrict the maximum width that any tile can grow and the label will wrap.
psc_section-simple-list-collection	Section	Converts all tiles to be elastic tiles and separates items by horizontal bottom border. By default, the background of the tile will be the same as the section, but the hover may not. Use *psc_theme-tile-transparent* at the section level to get background to stay same color. Similar to *psc_section-elastic*, the *--pt-tile-max-width* CSS variable can be set through custom styling to restrict the growth of the list (and the list will be centered within section by default).
psc_section-collapse-gap-all	Section	Consumes the section gap on all sides of a section using the *--pt-section-grid-gap* value making the section larger.
psc_section-collapse-gap-inline	Section	Consumes the section gap on the start/ left and end/right of a section. This is not a replacement for using full width section but may be used in other use cases.
psc_section-collapse-gap-block	Section	Consumes the section gap above and below the section. Very useful when a new full width section between two other full width sections so that there is no gap between full width sections. Typical use case for psc_section-collapse* styles.
psc_section-collapse-gap-top psc_section-collapse-gap-bottom psc_section-collapse-gap-start psc_section-collapse-gap-end	Section	Consumes the section gap on one side (top, bottom, start/left, end/right) making the section larger based on the value of *--pt-section-grid-gap*. Typically used when attempting to consume section gap between specific sections. Care must be taken when section possibly wrap as undesired layout may occur.

Style	***Applicable Level***	***Description***
psc_section-branding-strip	Section	Adds a thin themed branding strip to the top of a section. Uses the strip as defined by the theme. At this point, only neutral and teal themes have a branding strip value. All other themes (if specified using psc_theme-primary-PALETTENAME) will default to the neutral branding strip.
psc_section-spacing-adjust	Section	Applicable only Full Width sections (psc _section-full-width). This automatically modifies the section padding (--pt-section-padding) and section action button adjustment (--pt-section-action-button-adjust) values to match the visual alignment of non-full width sections so that the visual alignments match, even though the full width section consume. May not work for non-default settings of --pt-section-interior-spacing and --pt-section-gutter in some cases.
psc_tile-shadow	Section Tile	Enables tile shadow either for a section or for a single tile. By default, tiles do not have shadows. The shadow by default uses a new shadow color (--pt-tile-alt-shadow-color) to define the default shadow to be used. Shadows if enabled will not change on hover or activate. New styles are specifically created to allow control over the tile shadows other than color (--pt-tile-shadow, --pt-tile-hover-shadow, --pt-tile-active-shadow). If the tile is not actionable (has no main hotspot enabled), this style will not display a shadow.
psc_tile-shadow-none	Tile	Allows for a specific tile that is found within a content where all tiles have shadows to remove the shadow selectively. Has no effect if *psc_tile-shadow* has not been used at section level. Uses the value of *--pt-section-grid-gap* to consume the space and make the section larger on all sizes.

Style	***Applicable Level***	***Description***
psc_section-action-button	Tile	Creates an image only tile which is taken out of the section CSS grid layout and absolutely positioned on the same line as the section header, that is, the upper portion of the section. Image and padding of tile is specifically sized to look like a standard image button. Mostly styled as transparent using one of the styles previously described. Also select the Identify Tile as a Button and the Use Alternate Text on Tile Image check boxes on the Tile CREF (or dynamic tile). This type of style should be applied only to the first tile in the section for proper keyboard navigation.
psc_tile-override	Tile	This style applied to a tile bypasses the entire tile adjustment logic.
psc_tile-layout-override	Tile	Applies to tiles which are dynamic tiles that have content (not just a tile label and an image). These tiles have the *ps_tile-dynamic* and *ps_tile-has-content* styles applied to it. When this style applied, it causes the typical grid layout for the dynamic content to be bypassed so that application developers can style the interior of a dynamic tile content from scratch. This provides easier control over complex dynamic tile content.
psc_tile-image-invert	Tile	Sets the tile app filter variable (--pt-tile-image-filter which defaults no none for no filter) to "invert(1) contrast(3)" so that the primary icon of the image can be inverted to be visible on a dark background. This is primarily for section action buttons and can only be applied to the standard dark gray icons for the tile.
psc_tile-emphasis-text	Tile	Typically used to specify emphasis of text for content of a dynamic tile. You can use the *--pt-tile-emphasis-text-color*, which defaults to the current font size and bold font weight. These defaults can be changed by setting the *--pt-tile-emphasis-font-weight* and the *--pt-tile-emphasis-font-size* CSS variables.
psc_tile-emphasis-large-text	Tile	Similar to the *psc_tile-emphasis-text* style but sets the font-weight and font-size variables to be 28px and bold respectively.

Style	Applicable Level	Description
psc_tile-less-emphasis-text	Tile	Typically used for dynamic tile content to specify text that has emphasis but less emphasis then the *psc_tile-emphasis** style. The default color is the tile text color and the font weight is bold. These can be changed by using the *--pt-tile-less-emphasis-text-color* and *--pt-tile-less-emphasis-font-weight* CSS variables.
psc_tile-secondary-text	Tile	Typically used for dynamic tile content to specify a smaller secondary set of text. The default is to show this text as a font size of 13.667px and font-weight of normal (color is the tile text color). To change the font size or font weight, you can use the *--pt-tile-secondary-text-font-size* and *--pt-tile-secondary-text-font-weight* CSS variables.
psc_tile-height-auto	Tile	Allows the specified tile to be reverted to the default height sizing of a tile to consume the grid layout cell of the tile.

Styling Homepages and Dashboards Using CSS Variables

Instead of modifying the styles that are described in the previous topics, you can use CSS variables to style homepages, sections, and tiles.

The following table lists the various levels where CSS variables can be defined.

Level	Identified as Tag or Class/Selector	Description
Top of document	:root OR HTML tag	Top level of the HTML document (the HTML tag). Overall palette and theme definitions and functional CSS variables are defined at this level.
Homepage container	.pt_homepage	Overall homepage container of all homepages. Typically when styling you also want to avoid styling in personalization mode which would be done by this CSS selector - *.pt_homepage:not(.persmode)*. Many overall CSS variables are set here including the homepage background and texture.
Dashboard container	.pt_apphomepage	Same level as homepage container, but it is for dashboard specific styling.

Level	*Identified as Tag or Class/Selector*	*Description*
Individual homepage	.psc_hpmodel-A	There are a number of other style classes which are assigned at the individual homepage level (including portal-, node-, and tab- style classes). However, you typically apply the *psc_hpmodel-A* style at the overall homepage level (per expected usage). Defaults are set here for overall section and tile sizing or layout using CSS variables. You should note that the visible homepage will be the homepage that does not have the *psc_invisible* style associated at the same level as *psc_hpmodel-A*.
Container of all sections	.psc_hp-sections	This is the container section for all homepages.
Individual section	.psc_hpsection	Each section has this style class assigned along with the section name or ID as defined by the administrator.
Section tile collection container	• .psc_hpsection • .nuitilegrid .ps_grid-body	Within the section is a DIV grid which has a grid body. This grid body is defined as a CSS grid where the individual tiles are contained and laid out. You can define CSS variables here to control tile sizing and layout for a specific section.
Individual tile container	.nuitile	Within the collection of tiles, you'll find the tiles. Tiles have two different levels: the overall wrapper used to layout the tiles in the collection and actionable tile area. The outer container is where tile styles are applied as well as the tile sizing styling based on the tile definition.
Individual tile actionable area	.nuilp	Within the tile container (.nuitile) is the actionable area (if the main hot spot is enabled, which is the default). It is here where focus is actually set when dealing with tiles. From this level and down will be the internal layout of the tiles. For this model many tiles will be laid out in a CSS Grid form.

CSS Variables

The following CSS variables can be used in custom style sheets for a homepage, section, or tile to set values used within the *psc_hpmodel-A* styling. However, this is not an exhaustive list of CSS variables. This table lists the significant set of CSS variables.

CSS Variable	*Applicable Level*	*Description*
--pt-tile-max-width	• Section • Tile	When set for elastic, simple list, and simple tiles, it specifies the maximum width the tiles can grow. Since this is typically applied to a collection of tiles, the section level is the usual place to specify this variable. Example: `--pt-tile-max-width: calc (var(--pt-tile-spacing) * 2 + var(--pt-tile-image-size) + 35ch); /* tile padding plus image width plus 35 characters max */`
--pt-tile-font-weight	• Section • Tile	Undefined CSS variable by default, used to indicate the font-weight for tiles rendered as links. Typically used for special links rather than for an entire collection of links. Internally, the default for tiles rendered as link is font weight of 100 (or normal) due to fallback value since this CSS variable is not defined anywhere.
--pt-section-background	• Homepage • Section	Sets the background for the section (all sections if set at the homepage; if those sections do not specify a color) when set at the section level. Use this variable if you want a complex background on not a simple color from the color palette (for example an image with a background color). Example:`--pt-section-background: var(--pt-theme-neutral-texture) repeat var (--pt-theme-neutral-10);`
--pt-section-text-color	• Homepage • Section	Sets the header and general text color for a section (or sections if set at the homepage level). Since this model typically does not involve collapsible section headers, there should be no need for a separate hover color.
--pt-section-link-text-color	• Homepage • Section	Color to be used for links found in the section (or sections when defined at homepage level). Typically defaults to the same color as the section text color. Can be reset at the tile level by using *psc_tile-simple-link* for a special link text color.

CSS Variable	***Applicable Level***	***Description***
--pt-section-border-color	• Homepage • Section	Special decorative border color that appears below the section image. By default, there is not a border used around each section but a shadow around a section with this model. This variable must be used in conjunction with the --*pt-section-border-style* variable, which by default is set to None. Default value for this variable is the section text color. If the section is a transparent section, the default border color will also be transparent instead of the section text color.
--pt-section-border-style	• Homepage • Section	Sets the border style for the section. Default value is None.
--pt-section-border-radius	• Homepage • Section	Size of the border radius applied to a section.
--pt-section-shadow	• Homepage • Section	Complete specification for the shadow of a section.
--pt-tile-separator-color	• Homepage • Section	Special color used to separate tiles (typically used for sections that display tiles as a list).
--pt-hpsection-image-url	Section	While this property is typically populated through the section image check box on the non-collapsible section in the homepage definition, this variable can also be used within custom style sheets to provide dynamic control over a section image. For example, you can use this variable when the section background changes and you need to change the image for color contrast.
--pt-homepage-bottom-spacing	Homepage	Use to customize the lower spacing of the homepage. Normally, it is the same as the section spacing. It is used by the style *psc_homepage-no-bottom-spacing* to remove the gap for footer support.

CSS Variable	***Applicable Level***	***Description***
--pt-homepage-text-color	Homepage	This variable is used to set the default text color for a homepage in general. This is important for transparent sections so that the text color can have a contrasting color with the homepage background. This variable is *not* defined in any delivered stylesheet but is available for use by application developers and customers to set the overall default text for the homepage for transparent sections.
--pt-tile-image-filter	• Section • Tile	By default this variable is not defined (that is, set to None). This variable allows changes in a tile's primary image through CSS filter property. Useful for transparent tiles to attempt to manipulate images to support proper contrast against changes to the background without changing images (if possible). Example: `--pt-tile-image-filter: contrast(3) invert(1);`
--pt-custom-homepage-background	Homepage	By default, this variable is not defined (that is, set to transparent). When set at the homepage level, this variable allows applications to have different homepage backgrounds for different pages. Example: `--pt-custom-homepage-background: var(--pt-theme-neutral-texture) repeat var(--pt-theme-neutral-10);`

CSS Variables for Styling Tiles

The following table lists the CSS variables that are specific to tiles. While these may be set at the tile level, you should note that the width or height of a tile affects the layout of a section.

CSS Variable	***Applicable Level***	***Description***
--pt-tile-width	• :root • Homepage • Section	Defines the BASE width for a tile. It is the smallest unit for the layout using CSS grid.

CSS Variable	***Applicable Level***	***Description***
--pt-tile-height	• :root • Homepage • Section	Defines the BASE height for a tile. It is the smallest unit for the layout using CSS grid.
--pt-tile-spacing	• :root • Homepage • Section	Internal padding for a tile.
--pt-tile-gap	• :root • Homepage • Section	Spacing between tiles
--pt-tile-width-#--pt-tile-height-#	• :root • Homepage • Section	Where "#" is the dimension specified in the Tile CREF definition (or via the Tile API), specifies the width/height values of a given tile per the specified configuration. With this model, the width (unless it is a list, elastic, or simple tile) is FIXED but the height value is more of a minimum allowing the content to grow.
--pt-tile-shadow	• Homepage • Section Container • Section (typical) • Tile	Specifies the complete shadow specification for the default state of a tile using psc_tile-shadow styling. Example: `--pt-tile-shadow: 0px 2px 4px 0px var(--pt-tile-alt-shadow-color, transparent);`
--pt-tile-hover-shadow --pt-tile-active-shadow	Wherever --*pt-tile-shadow* is defined.	Specifies the complete shadow specification for hover and active states of a tile. The default is that hover and active shadows are the same as the default value for --*pt-tile-shadow*. If the default value for --*pt-tile-shadow* is changed, you should also change --*pt-tile-hover-shadow* and --*pt-tile-active-shadow* variables as well.

CSS Variable	Applicable Level	Description
--pt-tile-alt-shadow-color	• :root • Homepage • Section Container • Section • Tile	Specifies the default shadow color to use when applying the style psc_tile-shadow. This is specified in PT_CUSTOM_ PROPS (defined in :root) and matches the default section shadow color. Example: `--pt-tile-alt-shadow-color: rgba(255,255,255,0.3);`

CSS Variables for Styling Homepages and Sections

The following table lists the CSS variables for styling homepages and sections.

The default CSS grid for homepage is a 12 column layout with the outer two columns used for a gutter or margin. At the homepage level, the element style *psc_hp-sections* contains the section layout, which can be modified through custom style sheets at the same level. The section specific layout can be overridden at the individual section level.

CSS Variable	Applicable Level	Description
--pt-section-grid-column-count	Defined a section container level of psc_ hp-sections element.	Determines the number of columns (including gutters) to be used in lay out sections on a homepage. The default number of columns is 12.
--pt-section-template-columns	Defined a section container level of psc_ hp-sections element.	Specifies the actual layout of columns as far as sizing into which sections will be placed. The default is `--pt-section-template-columns: var(--pt-section-gutter) repeat(calc(var(--pt-section-grid-column-count) - 2), minmax(0, 1fr)) var (--pt-section-gutter);`
--pt-section-gutter	Defined a section container level of psc_ hp-sections element.	Column sizing for the side gutters for layout of the homepage. The default is 0px (meaning that the actual margin is implemented through the gap for sections) so the gutter found on the sides of the homepage will be the size of a section gap.
--pt-section-spacing	Defined a section container level of psc_ hp-sections element.	Specifies the spacing between sections (and the gutter). The default value is 32px.

CSS Variable	***Applicable Level***	***Description***
--pt-section-min-spacing	Defined a section container level of psc_ hp-sections element.	Specifies the minimum spacing for responsiveness. The default is 5cqi. *cqi* units are container units as a percentage of inline sizing (width).
--pt-section-grid-gap	Defined a section container level of psc_ hp-sections element.	Specifies the actual gap between sections (which is responsive). The default is as follows: `--pt-section-grid-gap: min(var(--pt-section-spacing, 32px),var(--pt-section-min-spacing,5cqi));`
--pt-section-padding	Section	Specifies the padding between the content and the bounding box container. The default is 32px. You can specify different paddings for each size using standard padding syntax. This variable is useful for full width tiles that need to have the contents moved to align better with the non-full width sections. Example: `--pt-section-padding: calc(2 * var(--pt-section-interior-spacing)) calc(4 * var(--pt-section-interior-spacing) + var(--pt-section-gutter));`
--pt-section-action-button-adjust	Section	Specifies the additional offset for the horizontal alignment of the section action button in addition to the default section padding. It is useful when the section is a full width section and the interior has been moved using the --pt-section-padding. Example: `--pt-section-action-button-adjust: calc(3 * var(--pt-section-interior-spacing) + var(--pt-section-gutter));`
--pt-section-row-start	Section	Specifies the grid row starting position for a section. The default is Auto.
--pt-section-row-end	Section	Specifies the grid row ending position for a section. The default is Auto.
--pt-section-column-start	Section	Specifies the grid column starting position for a section. The default is 2 (leaving the first column for a gutter).

CSS Variable	*Applicable Level*	*Description*
--pt-section-column-end	Section	Specifies the grid column ending position for a section. The default is -2 (spans all columns to the next to the last column so as to leave a gutter).
--pt-section-content-max-height	• Homepage • Section Container • Section (typical)	Use in conjunction with style *psc_ section-content-scrollable* to specify the maximum height for the content of a section before it starts to vertically scroll. This variable is not defined but will default to none (for not max-height) Example: `--pt-section-content-max-height: 17ch; /* for 17 charcters tall */`
--pt-section-content-padding	• Homepage • Section Container • Section (typical)	Use in conjuction with style *psc_section-content-scrollable* and *psc_tile-shadow*. Since shadows extend beyond the boundaries of the tile and therefore the section content, additional padding must be added to the section content to avoid clipping visual dressings that are external to the tile (like shadows). Typically, it would be the maximum spread size for a shadow. Example: `--pt-section-content-padding: 8px; /* size of shadow spread */`

Designing Sections and Tiles

Advanced sectionalized homepage development provides you greater control and flexibility in designing the sections and tiles on your homepage or dashboard.

Creating and Styling Sections

Sections are an important construct in advanced sectionalized homepage development. This topic briefly describes the process of creating sections and points you to existing documentation. Additionally, this topic describes the various visual forms for sections and points to the styles and CSS variables delivered for this purpose.

Creating a Section

You can create a section either using the Section Assignments tab on the Content Ref Administration page or you can create section templates and import them.

Refer to the following topics for more information on creating sections: (Fluid only) Creating Sections on Fluid Homepages and Managing Section Templates and Sections.

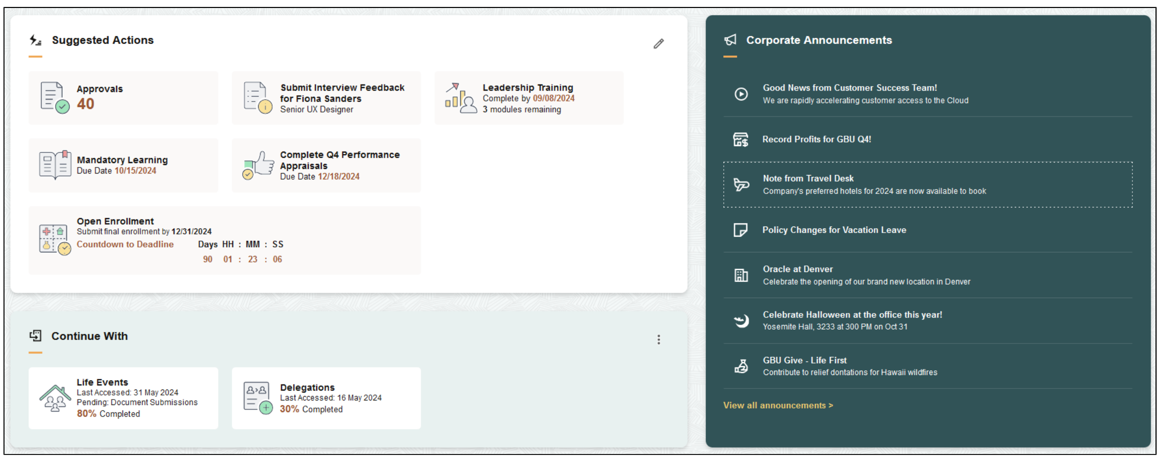
Designing Visual Forms of Sections

You use the delivered styles and CSS variables applicable at the level of a section to add granular styling to the section. The styles and CSS variables are described in the preceding topics. Based on your business requirements, you can style sections to span the entire width of the homepage or style sections such that they grow vertically or horizontally as you add tiles in the section, or you can lay the sections side-by-side.

Additionally, you can design each sections to be unique in its look and feel by applying specific background color, border, icon, and so on.

The following example shows some of the visual forms of sections.

This example illustrates some of the visual forms of sections.



Adjusting Section Height to Enable Vertical Scrolling

When you want to enable vertical scrolling for users to scroll through the tiles, you should complete the following steps:

- At the landing page or homepage content reference level (CREF), add a style in the PS_STYLECLASSES section attribute: *psc_section-content-scrollable*.
- In the custom landing page or section stylesheet, set the CSS variable *--pt-section-content-max-height* to the desired value.

  Sample CSS to set the value:

  This CSS, for example, allows the section content (the tiles within) to grow to a maximum height of 17 character units. Once that size is reached, the section presents a vertical scroll bar for access to the other tiles which are outside of the visible area. This allows the section content to only grow to a given height.

```
.pt_homepage:not(.persmode) .tab-PTDMO_LANDING_SAMPLE {
    & .sect-PTFAVORITES { /*Section: Favorites */
        --pt-section-content-max-height: 17ch;
    }
}
```

Creating Tiles

This topic briefly describes the process of creating tiles and points you to existing documentation. In addition to creating tiles using the tile wizard or creating tile definitions manually, you can create tiles dynamically and load them into a section.

Creating Tiles Statically

You can create a tile using the Tile Wizard or you can manually create tile content references.

Refer to the following topics for more information on creating tiles: Using Tile Wizard and Creating and Maintaining Tile Definitions Manually.

Creating Tiles Dynamically

You can create tiles dynamically and load these tiles into a section using the delivered base class: PTNUI:Dynamic:DynamicLoader. As these tiles are dynamically created, they do not have a tile content reference.

Typically, you would create tiles dynamically in this manner:

- Create an application class, which is an extension to the base class PTNUI:Dynamic:DynamicLoader.
- Use the LoadSection method to load a section.
- Use the CreateTile or CreateTileUsingTab methods to define a tile.
- Use the AddTileToSection method to add the tile to the section.

The following code snippet creates a standard tile with title and image:

```
method LoadSection
   /+ Extends/implements PTNUI:Dynamic:DynamicLoader.LoadSection +/
   Local PTNUI:Model:Tile &oTile;

   &oTile = %This.CreateTileUsingTab("", "Team Calendar");
   &oTile.RowNum = %This.GetNextRowNumber();
   &oTile.GltImageName = Image.TEAM_CALENDAR;
   %This.SetupFluidTarget(&oTile, GenerateComponentContentURL(%This.Tab.PortalName,⇒

 %This.Tab.NodeName, MenuName.MYMENU, %Market, Component.MYCOMPONENT, Page.MYPAGE, ⇒

%Action_UpdateDisplay));

   %This.AddTileToSection(&oTile);
end-method;
```

For a detailed description of the DynamicLoader class and its methods, see “DynamicLoader Class Methods” (PeopleCode API Reference).

Creating Different Presentations of Tiles

PeopleTools supports creating different presentations of tiles.

You can create the following tiles:

- Tile presented as a button.
- Tile presented as a link.
- Standard tile.

Tile Presented as a Button

This is a special tile that can be configured as a button instead of the default rendering as a link. You must ensure that it is the first tile in a section for tab ordering. You can use the tile to open a modeless window, which enables users to configure or personalize the section. Special styling is applied to the tile to make it appear in the upper-right corner of a section.

Special properties must be applied to a tile for the tile to behave as a button. These properties are applicable to both statically defined tiles and dynamically generated tiles.

To configure a tile as a button, complete these steps:

1. If you're using an application class to dynamically create the tile by extending the DynamicLoader class, set the following properties:
   - Use the Tile.IsButtonRole property and set it to True.
   - Use the Tile.IsNotTile property and set it to True.
   - Use the Tile.StyleClasses property and specify the *psc_section-action-button* attribute.
   - Use the Tile.UseAltText property and set it to True. If you are using only an image for the button, ensure that the alternate text and title are populated with the text of the tile label.

   If you are creating a statically defined tile, in the tile definition, select the following check boxes on the Fluid Attributes tab:

   - Identify Tile as a Button
   - Use Alternate Text on Tile Image

2. Update the homepage content reference to include the *psc_section-action-button* value for the PS_STYLECLASSES attribute in the Section Attributes area on the Section Assignments page.

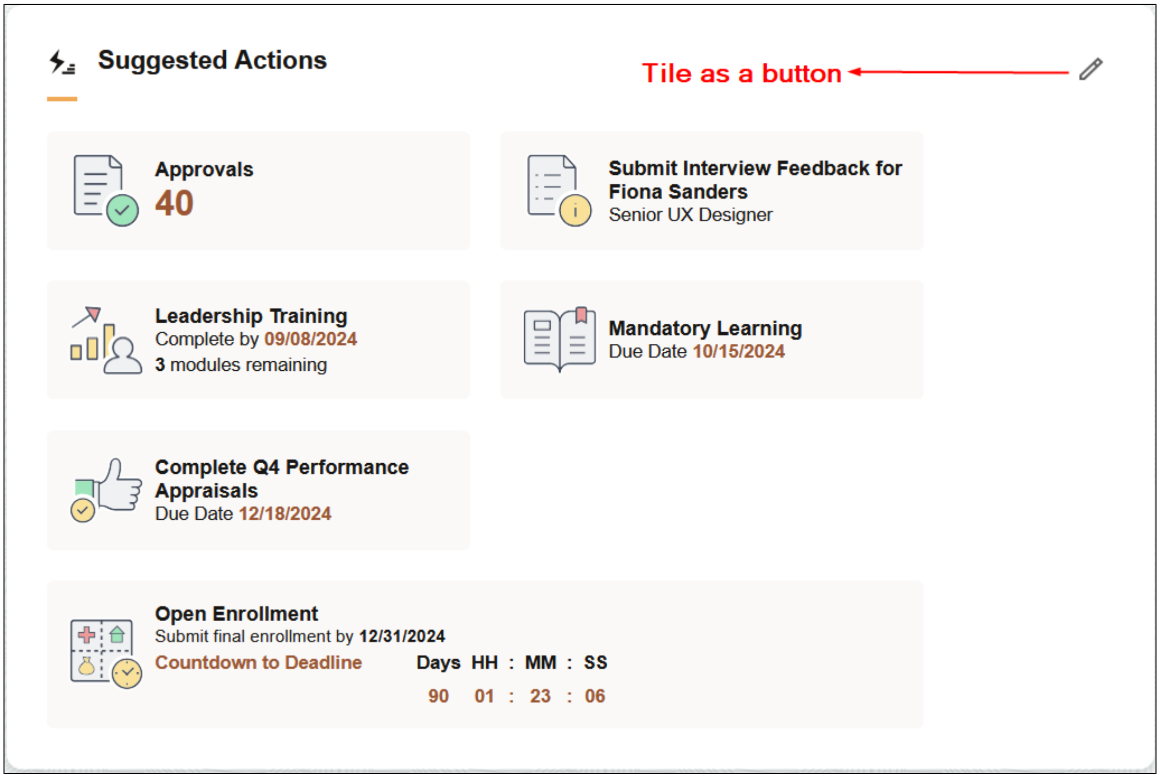
The following sample code shows the properties for the tile.

```
&oTile = %This.CreateTileUsingTab("", "Edit Suggested Actions");
   &oTile.RowNum = 1;
   &oTile.GltImageName = Image.MYIMAGE;
   &oTile.UseAltText = True;
   &oTile.IsButtonRole = True;
   &oTile.IsNotTile = True;
   &oTile.DisplayIn = %GroupletActionInModal;
   &oTile.StyleClasses = "psc_theme-tile-transparent;psc_section-action-button";
   &oTile.IncludeSourceInUrl = True;
   %This.SetupFluidTarget(&oTile, GenerateComponentContentURL(%This.Tab.PortalName,⇒
```

```
 %This.Tab.NodeName, MenuName.MYMENU, %Market, Component.MYCOMPONENT, Page.MYPAGE, ⇒

%Action_UpdateDisplay));
   %This.AddTileToSection(&oTile);
```

The sample PeopleCode creates a tile presented as a button:

Example of a tile displayed as a button.



See Applying Styles from Delivered Style Sheets and Styling Homepages and Dashboards Using CSS Variables for information on styles and CSS variables that can be applied to sections and tiles.

Tile Presented as a Link

You can create a tile to display as a link (hyperlink). Special properties have to be applied to the tile to behave as a link. These properties and styles are discussed in preceding topics.

- At the tile level, use the *psc_tile-simple-link* style class to convert a tile to a hyperlink.
- At the section level, use the *psc_section-link-collection* style class, which converts all tiles in the section into a collection of links displayed vertically.
- Use the Tile.IsNotTile property and set it to True.

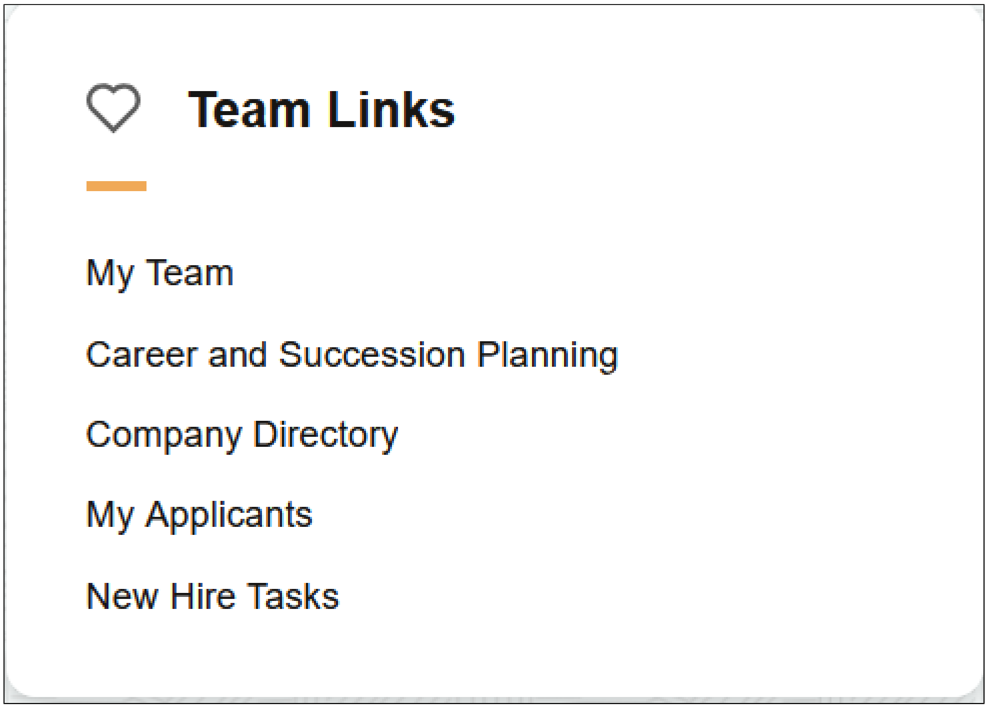
The following sample code creates a tile that is presented as a link. At the section level, the *psc_section-link-collection* style class is applied which presents the tiles as links.

```
&oTile = %This.CreateTileUsingTab("", "My Team");
   &oTile.RowNum = %This.GetNextRowNumber();
   &oTile.IsNotTile = True;
   %This.SetupFluidTarget(&oTile, GenerateComponentContentURL(%This.Tab.PortalName,⇒
```



```
 %This.Tab.NodeName, MenuName.MYMENU, %Market, Component.MYCOMPONENT, Page.MYPAGE, ⇒
%Action_UpdateDisplay));
   %This.AddTileToSection(&oTile);
```

Example of tiles displayed as links.



See Applying Styles from Delivered Style Sheets and Styling Homepages and Dashboards Using CSS Variables for information on styles and CSS variables that can be applied to sections and tiles.

Standard Tiles

Standard tiles are tiles with image and text or tiles with live data. However, the visual presentation of these tiles can be changed by applying appropriate styles. For example, tiles can be displayed as a collection with a separator line, a tile can span the width of a section; a tile can display secondary text in addition to the main text.

See Applying Styles from Delivered Style Sheets and Styling Homepages and Dashboards Using CSS Variables for information on styles and CSS variables that can be applied to sections and tiles.

Example of standard tiles.

Related Links

"Tile Class Properties" (PeopleCode API Reference)

Grouplet-Based Tile Styling

Grouplet-based tile styling provides the ability to apply new styling to tiles created in pre-8.62 releases such that the styling affects only the content of the tile without modifying the tile content reference (CREF). The grouplet-based styling requires a change only to the grouplet component content or grouplet iScript content. The grouplet-based styling does not require any modifications to the tile CREF, so Applications do not need to redeliver tile CREFs when they implement advanced sectionalized homepage.

Note: The grouplet-based styling is applicable only within the context of advanced sectionalized homepage, that is, the *psc_hpmodel-A* style must be implemented.

When grouplet-based styling is implemented, two types of tiles are automatically supported without requiring any modifications by Applications.

- Tile grouplet content containing only an image (not the image defined in the tile CREF definition).
- Tile grouplet content containing an image and a live data count.

However, these tiles are automatically supported only if the tiles are delivered with a specific structure and styling used in the pre-8.62 implementation of tile content. The following structure and styling is expected:

Type of Tile	***Structure and Styling***
Image only grouplet	`DIV.ps_box-group` with either `psc_tile_content` or `psc_tile-box-img` (this is a groupbox and it must be the top level to be rendered) `DIV.psc_tile-img` (any type of image based control) `IMG`
Image and live data count grouplet	`DIV.ps_box-group` with either `psc_tile_content` or `psc_tile-box-img` (this is a groupbox and it must be the top level to be rendered) `DIV.psc_tile-img` (any type of image based control) `IMG` `DIV.psc_tile_livedata-count.psc_nolabel` (display only text control like edit box with no label displayed) `DIV.ps_box-value` (count value here) `DIV.ps_box-label` (could come before the ps_box-value since it is not displayed)

All other tile structures and styling require modification for the grouplet-based styling to render correctly on the pre-8.62 tiles. Application developers must consider the following aspects of the tiles and apply the styles appropriately.

- Identify that this tile conforms to a general standard tile content, that is, the tile has an image, tile label, and one or more lines of content.
- Ignore the layers of structure of the tiles, such as group boxes and HTML areas.
- Identify the lines of content data directly to be displayed within the tile. Optionally, use new styles to control the rendering of the lines of content.

Style Name	***Typical Structure Usage***	***Description***
psc_tile-standard	Top-level Group box	Identifies that this tile has been tested by the application and works per standards defined by these styles. Do not use this style if the tile does not conform to tile standard content (for example - no image)
psc_tile-structure-ignore	Top-level Group box	Ignore the current structure (a groupbox) and all groupbox and HTML structures found within this structure. This styling uses display:contents which makes the CSS grid layout focus on only the important parts of the tile content (tile image, tile label, lines of data). This is the most commonly used style.
psc_tile-groupbox-ignore	Any group box level	Not necessary whether *psc_tile-structure-ignore* is used on an ancestor element. Allows ignoring of only groupbox structures from this element down to expose the data. Typically used for tiles requiring custom changes due to the structure, meaning that *psc_tile-structure-ignore* could not be used.
psc_tile-htmlarea-ignore	Any HTML Area level	Not necessary whether *psc_tile-structure-ignore* is used on an ancestor element. Allows ignoring of only HTML area structures from this element down to expose the data. Typically used for tiles requiring custom changes due to the structure, meaning that *psc_tile-structure-ignore* could not be used.
psc_tile-ignore	Any Container level	Applies to only a single element to be ignored for layout purposes (does not affect any children elements)
psc_tile-img-swap-dim	Image Control having psc_tile-img	Typically, tile images use the width as the major width allowing the height to be derived (allowing aspect ratio to be maintained). Typical tile image is 60x60 pixels so it actually does not matter. This style says to use the height instead of the width for the dimension.

Style Name	***Typical Structure Usage***	***Description***
psc_tile-data	Edit or Text Control	Indicates that this structure includes the live data information to be displayed as a line of data.
psc_tile-data-1 through psc_tile-data-6	Used in conjunction with psc_tile-data level	Indicates the line of information. Do not duplicate information as these tiles are used to properly center the tile image vertically.
psc_tile-emphasis-text psc_tile-label-emphasis-text psc_tile-value-emphasis-text	Tile line data	Shows the appropriate data with emphasis (typically the live data count coloring). Notice the first style applies to all contents of the control. The next two styles apply to either the label or value of the display-only textual data field.
psc_tile-emphasis-large-text psc_tile-label-emphasis-large-text psc_tile-value-emphasis-large-text	Tile line data	Same as previous row except larger text.
psc_tile-less-emphasis-text psc_tile-label-less-emphasis-text psc_tile-value-less-emphasis-text	Tile line data	Shows the appropriate data with less emphasis but not normal text (typically bolded). Notice the first style applies to all contents of the control. The next two styles apply to either the label or value of the display-only textual data field.
psc_tile-normal-text psc_tile-label-normal-text psc_tile-value-normal-text	Tile line data	Shows the appropriate data with normal text. Notice the first style applies to all contents of the control. The next two styles apply to either the label or value of the display-only textual data field.
psc_tile-secondary-text psc_tile-label-secondary-text psc_tile-value-secondary-text	Tile line data	Shows the appropriate data with smaller font as with secondary data. Notice the first style applies to all contents of the control. The next two styles apply to either the label or value of the display-only textual data field.

Automatic Section Adjustments on Small Form Factor Devices

From PeopleTools 8.62.02, a section automatically adjusts to narrow browser windows (browser window widths below a given breakpoint) to handle small form factor sizes without having to adapt based on being identified as a small form factor at login. The breakpoint for section adjustment is set to occur below 453 pixels.

Section adjustment or compression below 453 pixels includes the following changes:

- All sections behave as full-width sections. Full-width sections have the following characteristics:
  - Sections do not have rounded corners (squared corners).
  - Sections do not have shadows.
  - Sections span entire width of the homepage.
- Homepage gutter is set to 0 pixels.
- Internal spacing of section is reduced; adjusts the section action button spacing if it exists.
- Section gaps are reduced.

Note: You can disable the auto-adjustment feature by applying the *psc_section-compression-none* style at the homepage level (in the PS_STYLECLASSES homepage attribute properties of the homepage content reference definition).

Some reasons why you may decide not to use the auto-adjustment feature:

- 453 pixels is not the desired breakpoint for your landing page implementation.
- The behavior performed by the auto-adjustment feature is not the desired behavior. Behaviors may not work as expected because default styling of sections, gaps, spacing or padding, or tile collections were not used.
- Implementors did not use styling provided by PeopleTools to layout the landing page in which case all auto-adjustment needs to be owned by the implementor.
- You may have your own design for responsiveness, which conflicts with the auto-adjustment behaviors.

Using Delivered PeopleTools Section Templates

A section template enables you to import a predefined section definition into a fluid homepage or dashboard. A section template defines a single section definition and is stored in the Fluid Section Templates folder (in the Fluid Structure Content folder), which you can import in your landing page, homepage, or dashboard.

PeopleTools delivers the following section templates:

- Recently Visited (PTRECENTLYVISITED).
- Favorites (PTFAVORITES).
- Navigation Collections (PT_NAVCOL_TEMPLATE).

In PeopleTools, the recently visited items and favorites are available in Quick Access Bar (QAB) and in the NavBar. From PeopleTools 8.62.06, you can add the recently visited items, favorites, and navigation collections to a landing page, sectionalized homepage, or sectionalized dashboard as a section. You should note that these section templates work correctly in landing pages, sectionalized homepages and dashboards, which implement the new styling model described in the preceding topics.

For instructions on importing section templates into a landing page, homepage, or dashboard, see Importing a Section in Homepages and Dashboards.

Recently Visited Section Template

The Recently Visited section template renders users' recently accessed navigation items as links by default. Administrators can import this section into any PeopleSoft landing page, homepage, or dashboard. The template leverages an application class that extends the DynamicLoader application class, ensuring dynamic and efficient data sourcing.

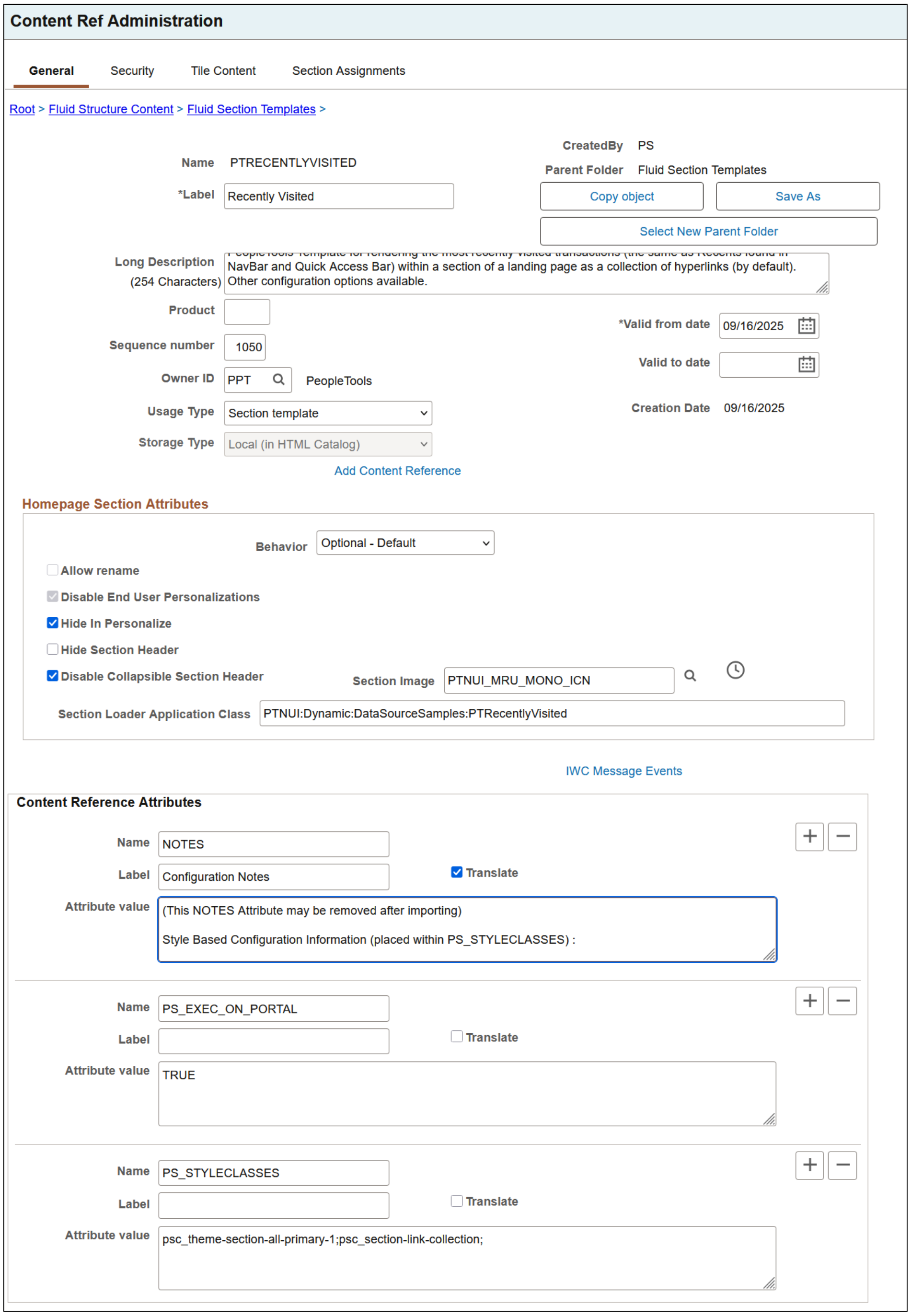
The recently visited items are available in the Quick Access Bar (QAB) and in the NavBar. In addition to these, PeopleTools enables you to add the recently visited items to a landing page, homepage, or dashboard as a section.

When you plan to use the Recently Visited section, you should note the following:

- User permissions for the Recently Visited section match those required to view recently used items in QAB and NavBar.
- If a user does not have access to the recently used items, the section does not appear.
- If there are no recent items, for example, on the user's first login, the section does not appear.
- The number of items displayed adheres to the configured limit for recently used items. However, you can specify the number of items to be displayed, which is described later in this topic.

To view the delivered Recently Visited section template, navigate to **PeopleTools** > **Portal** > **Structure and Content** > **Fluid Structure Content** > **Fluid Section Templates**, and select the corresponding Edit link.

This example illustrates the settings in the delivered Recently Visited section template. You can find definitions for the fields and controls later in this page.



For descriptions of all the fields and controls on the Recently Visited section template page, see Creating a Section Template.

The following table describes the fields that you should take note of and change, if required, after you import this section template to a target landing page, homepage, or dashboard.

<table>
<tr><th>Field or Control</th><th>Description</th></tr>
<tr><td>Section Loader Application Class</td><td>Use the PTRecentlyVisited application class to dynamically load the recent items in the section. It is an extension to the base class PTNUI:Dynamic:DynamicLoader.</td></tr>
<tr><td>Content Reference Attributes</td><td>The PS_EXEC_ON_PORTAL attribute is introduced in PeopleTools 8.62.06. Setting it to True ensures that the recently visited data displayed in the section is found on the portal or login node so that the data is consistent with the recently visited items in QAB and NavBar (which are both cluster-aware features).

Note: It is recommended that you do not change the value set in the template.

For more information on the PS_EXEC_ON_PORTAL attribute, see <u>Understanding the PS_EXEC_ON_PORTAL Attribute</u>.
PS_STYLECLASSES

After importing this section template to the target landing page, you may change the styling of the section as required by your business needs.

For a list of styles that are delivered by PeopleTools, see <u>Styles for Layout and Behavior</u>.

In addition to the styling attributes, PeopleTools also delivers attributes to configure the data that is displayed in the section. For a list of attributes, see the Style Class-Based Data Configurations for Recently Visited Section later in this topic.</td></tr>
</table>

Style Class-Based Data Configurations for Recently Visited Section

The style class-based data-configurations should be placed in PS_STYLECLASSES and should be prefixed with data_. It should follow this format: data_NAME-VALUE. The name (alphanumeric) and value (alphanumeric) pair should be separated with a hyphen (-). For example, data_IMAGENAME-DEFAULT.

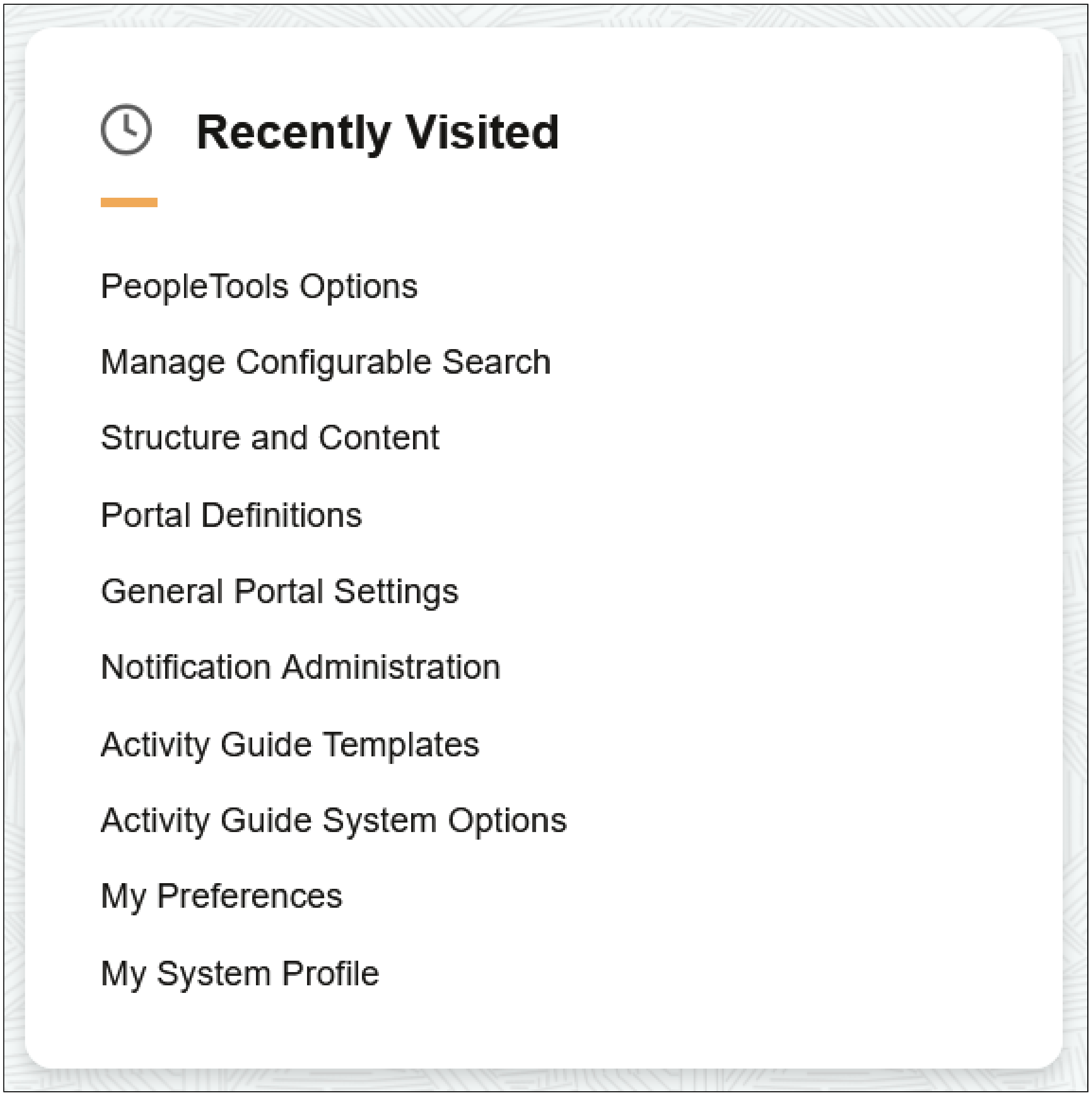
Style	***Example***	***Description***
data_IMAGENAME-?	data_IMAGENAME-DEFAULT	Replace the "?" with the name of the icon to be used for all items within this section, however you cannot use an icon for tile presentation as a hyperlink; you should choose an alternate tile presentation to use an icon. A value of DEFAULT means that the same icon used within NavBar and QAB will be used for all recent items. This setting is optional.
data_MAXRESULTS-?	data_MAXRESULTS-5	Replace the "?" with the maximum recently visited items (numeric value) to be displayed within the section. The default is all items up to the configured limit of maximum recent items storage. This setting is optional.
data_TILEPREFIX-?	data_TILEPREFIX-MYMRU	Replace the "?" with the string to be used as the prefix for the tile names (ID's) created dynamically in the section. The default is PTMRU. The alphanumeric string should not exceed 8 characters and should not contain special characters. This setting is optional. You would typically use a prefix to differentiate one set of recently visited tiles from another set. For example, if you plan to create two recently visited sections with different tile presentation, using a prefix helps you to differentiate the tile names.

Examples of the Recently Visited Section

The Recently Visited section with the default settings is rendered as shown in this screenshot.

- psc_theme-section-all-primary-1
- psc_section-link-collection
- psc_sectgroup-C

This example illustrates the Recently Visited section with default settings.



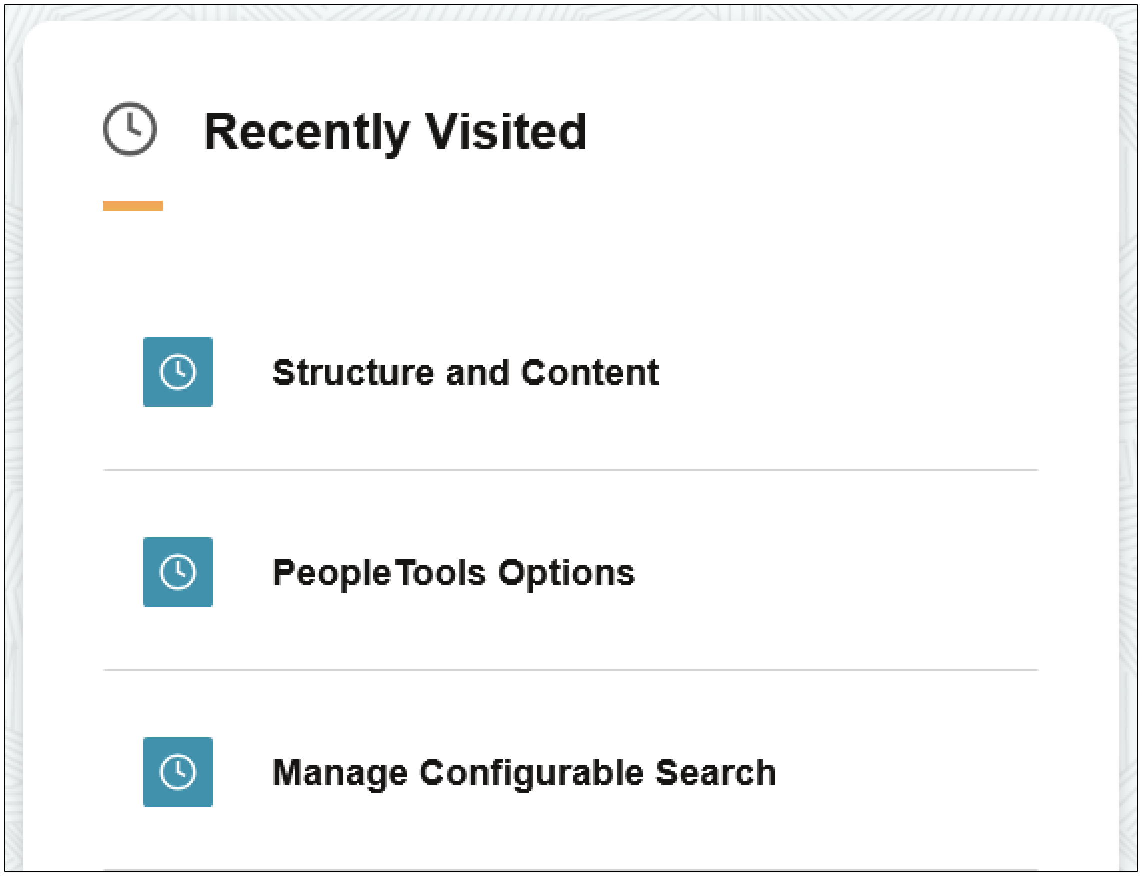
After importing the section template, you may modify the style classes and style class-based data configurations as shown in the following example:

In PS_STYLECLASSES attribute, enter style classes separated by semi-colon (;). In this example, the single style class-based configuration sets the image name for all entries to be PTNUI_MRUITEM_ICN.

- psc_theme-section-primary-1
- psc_theme-tile-transparent
- psc_section-simple-list-collection
- psc_sectgroup-C
- data_IMAGENAME-PTNUI_MRUITEM_ICN

Based on styling and data configurations from the example, the Recently Visited section is rendered as shown here:

This example illustrates the Recently Visited section post import with modified style classes and style class-based data configurations.



Favorites Section Template

The Favorites section template renders users' favorite items as links by default. Administrators can import this section into any PeopleSoft landing page, homepage, or dashboard. The template leverages an application class that extends the DynamicLoader application class.

The Favorites items are available in the Quick Access Bar (QAB) and in the NavBar. In addition to these, PeopleTools enables you to add the Favorites items to a landing page, sectionalized homepage, or sectionalized dashboard as a section.

When you plan to use the Favorites section, you should note the following:

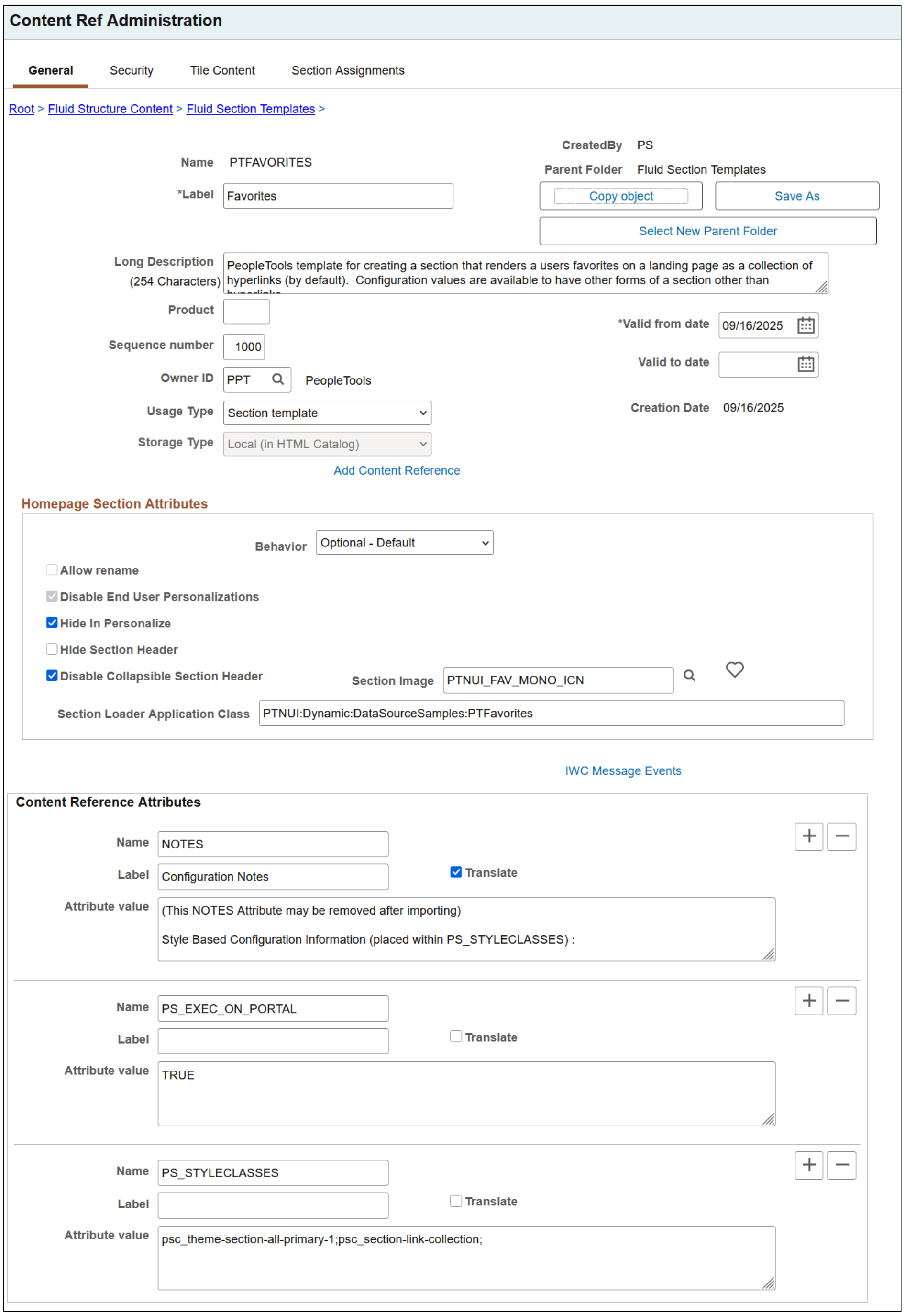
- If a user has not saved any favorites, the section does not appear.
- Permissions to access favorites data are the same as that for favorites within QAB and NavBar. If the user does not have permissions to access favorites data, the section does not appear.
- All favorites items are displayed. You can use style class-based data configuration (*data_MAXRESULTS-?*) to limit the number of favorites items displayed in the section.

  If you want to design the section such that a scroll bar appears for users to view all the items, application developers can enable vertical scrolling by limiting the height of a section. For information on adjusting section height to enable vertical scrolling, see Creating and Styling Sections.

- An edit button (pencil icon) is available on the Favorites section to navigate to the Edit Favorites component. For more information on the Edit Favorites page, see "Working with Favorites" (Applications User's Guide).

To view the delivered Favorites section template, navigate to **PeopleTools** > **Portal** > **Structure and Content** > **Fluid Structure Content** > **Fluid Section Templates**, and select the corresponding Edit link.

This example illustrates the settings in the delivered Favorites section template. You can find definitions for the fields and controls later in this page.



For descriptions of all the fields and controls on the Favorites section template page, see Creating a Section Template.

The following table describes the fields that you should take note of and change, if required, after you import this section template to a target landing page, sectionalized homepage, or sectionalized dashboard.

Field or Control	*Description*
Section Loader Application Class	Use the PTFavorites application class to dynamically load the Favorites items in the section. It is an extension to the base class PTNUI:Dynamic:DynamicLoader.
Content Reference Attributes	• The PS_EXEC_ON_PORTAL attribute is introduced in PeopleTools 8.62.06. Setting it to True ensures that the favorites displayed in the section is found on the portal or login node so that the data is consistent with the favorites items in QAB and NavBar (which are both cluster-aware features). **Note:** It is recommended that you do *not* change the value set in the template. For more information on the PS_EXEC_ON_PORTAL attribute, see Understanding the PS_EXEC_ON_PORTAL Attribute. • PS_STYLECLASSES After importing this section template to the target landing page, you may change the styling of the section as required by your business needs. For a list of styles that are delivered by PeopleTools, see Styles for Layout and Behavior. In addition to the style classes, PeopleTools also delivers style class-based configuration to configure the data that is displayed in the section. For a list of style class-based configurations, see the Style Class-Based Data Configurations for the Favorites Section later in this topic.

Style Class-Based Data Configurations for the Favorites Section

The style class-based data configurations should be placed in PS_STYLECLASSES and should be prefixed with data_. It should follow this format: data_NAME-VALUE. The name (alphanumeric) and value (alphanumeric) pair should be separated with a hyphen (-). For example, data_IMAGENAME-DEFAULT.

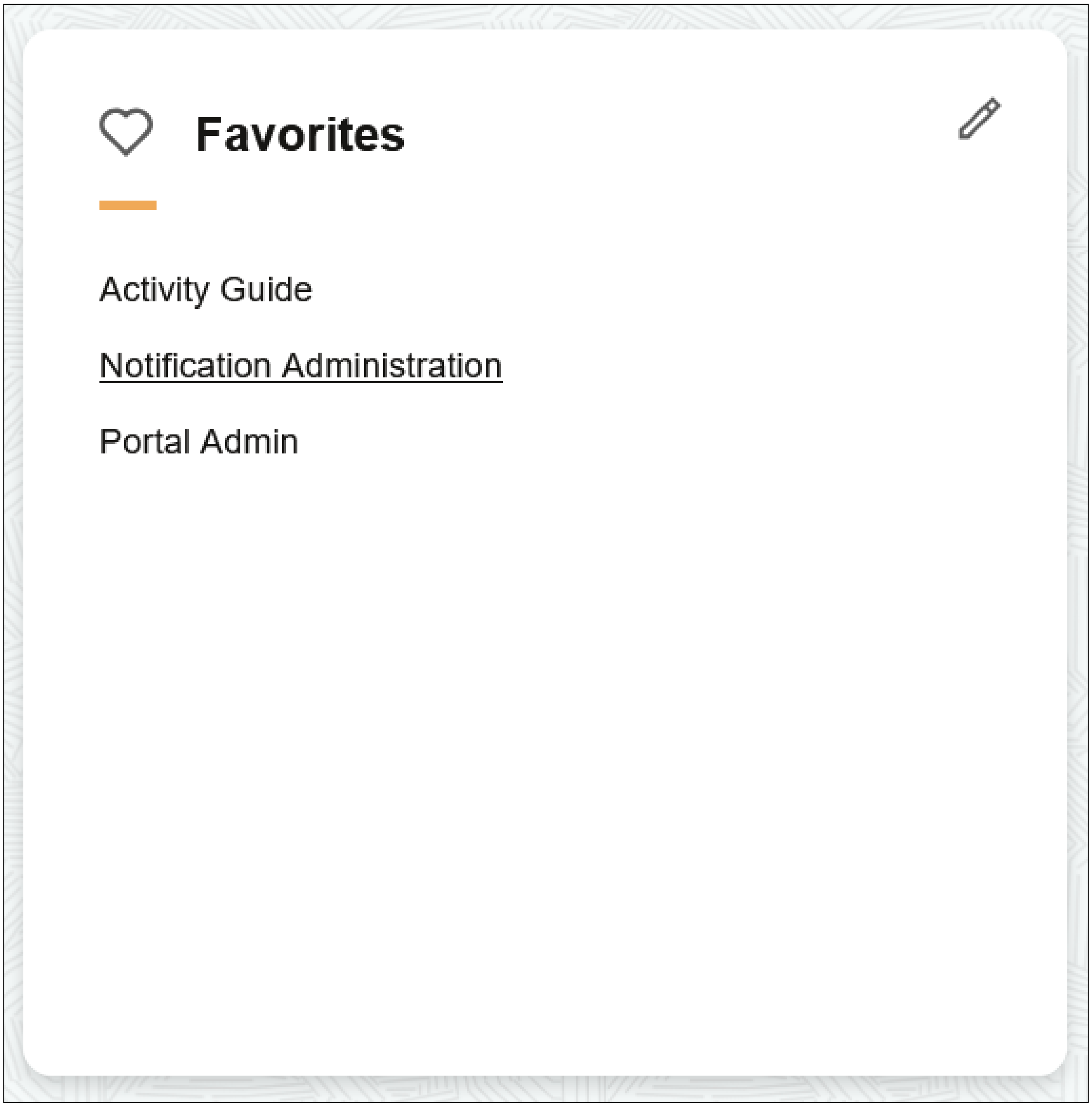
Style	***Example***	***Description***
data_IMAGENAME-?	data_IMAGENAME-DEFAULT	Replace the "?" with the name of the icon to be used for all items within this section, however you cannot use an icon for tile presentation as a hyperlink; you should choose an alternate tile presentation to use an icon. A value of DEFAULT means that the same icon used within NavBar and QAB will be used for all Favorites items. This setting is optional.
data_MAXRESULTS-?	data_MAXRESULTS-5	Replace the "?" with the maximum favorite items (numeric value) to be displayed within the section. The default is all items or a value of 0. This setting is optional.
data_TILEPREFIX-?	data_TILEPREFIX-MYFAV	Replace the "?" with the string to be used as the prefix for the tile names (ID's) created dynamically in the section. The value for this property should not exceed 8 characters. The default is PTFAV. This setting is optional. You would typically use a prefix to differentiate one set of favorites tiles from another set. For example, if you plan to create two favorites sections with different tile presentation, using a prefix helps you to differentiate the tile names.

Examples of the Favorites Section

The Favorites section with the default settings is rendered as shown in this screenshot.

- psc_theme-section-all-primary-1
- psc_section-link-collection
- psc_sectgroup-C

This example illustrates the Favorites section with default settings.



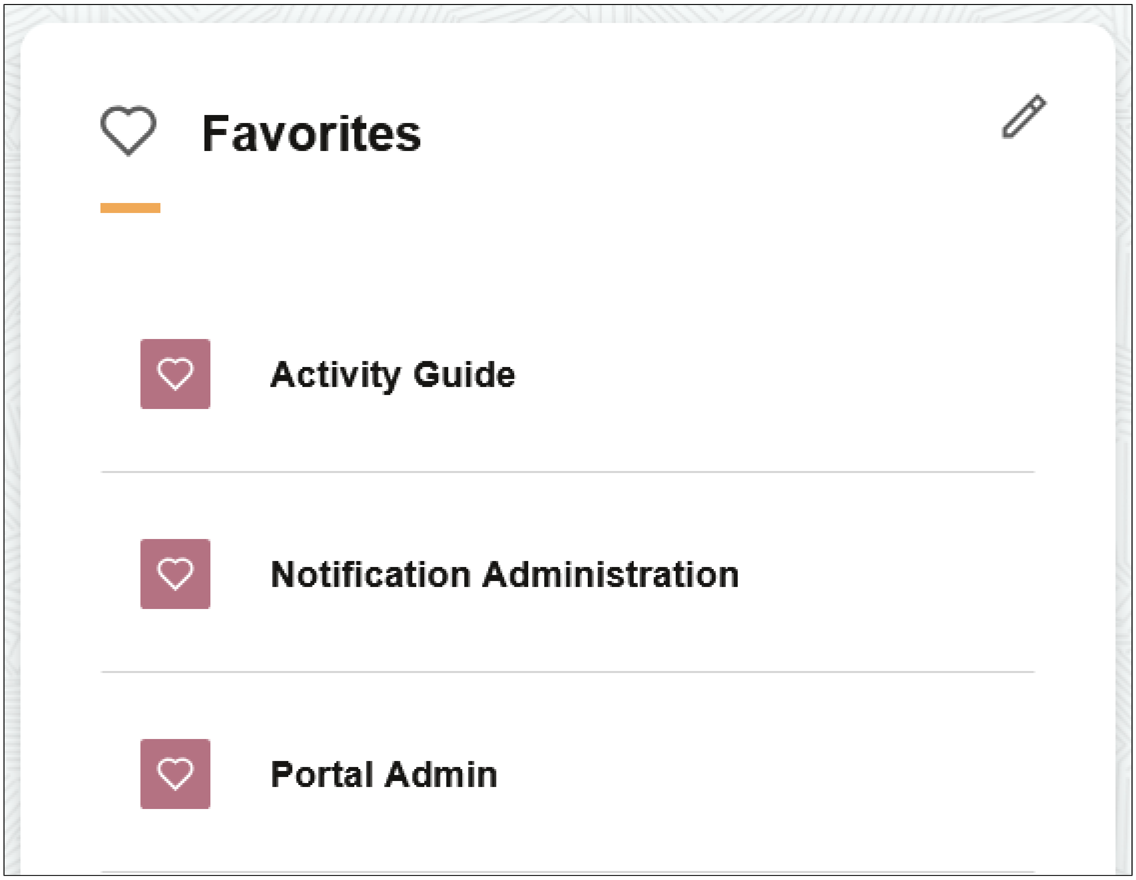
After importing the section template, you may modify the style classes and style class-based data configurations as shown in the following example:

In PS_STYLECLASSES, enter the style classes separated by a semi-colon (;). In this example, the single style class-based configuration sets the image name for all entries to be PTNUI_FAVITEM_ICN.

- psc_theme-section-primary-1
- psc_theme-tile-transparent
- psc_section-simple-list-collection
- data_IMAGENAME-PTNUI_FAVITEM_ICN

Based on styling and data configurations from the example, the Favorites section is rendered as shown here:

This example illustrates the Favorites section post import with modified style classes and style class-based data configurations.



Navigation Collection Section Template

The Navigation Collection section template renders specified navigation collection as links by default. Administrators can import this section into any PeopleSoft landing page, sectionalized homepage, or sectionalized dashboard. The template leverages an application class that extends the DynamicLoader application class, ensuring dynamic and efficient data sourcing.

The navigation collection section template is not a ready-to-use section, unlike the recently visited and favorites section templates. After you import the section into a sectionalized homepage or dashboard, you must manually configure it for your navigation collection.

- Section **Name** - Change from the default PT_NAVCOL_TEMPLATE to a unique name that describes your collection's purpose.

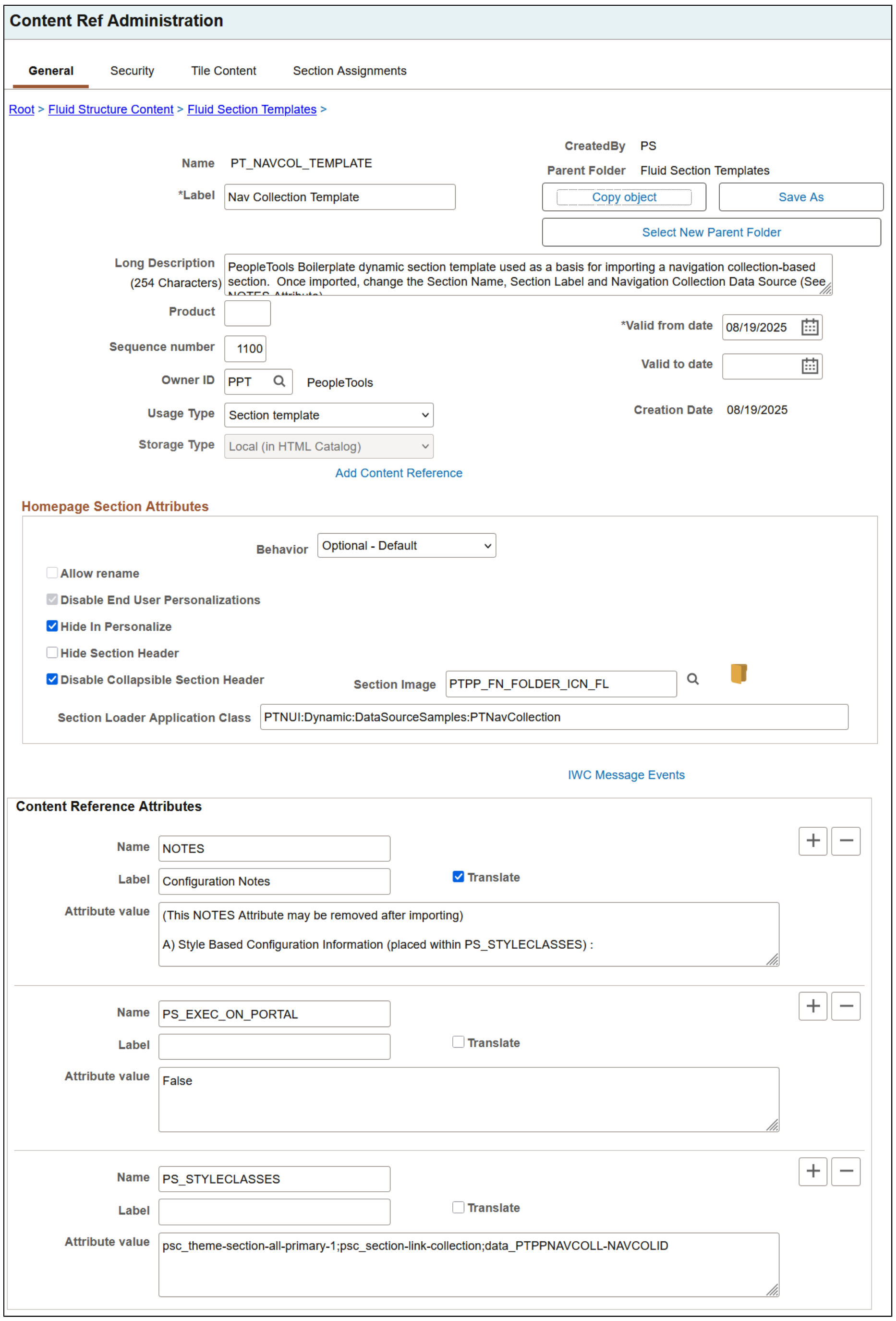
  Note: Avoid the *PT_* prefix as it is reserved for PeopleTools-delivered objects.

- Section **Label** - Set a meaningful label that reflects the navigation collection for end-users.
- **Section Image** - Select an appropriate image or icon for the navigation section.

- Data configuration - In the PS_STYLECLASSES attribute, locate the style class-based data configuration, specifically the text: *data_PTPPNAVCOLL-NAVCOLID*. Replace *NAVCOLID* with the actual navigation collection ID to drive the section tiles.

To view the delivered Navigation Collection section template, navigate to **PeopleTools** > **Portal** > **Structure and Content** > **Fluid Structure Content** > **Fluid Section Templates**, and select the corresponding Edit link.

This example illustrates the settings in the delivered Nav Collection section template. You can find definitions for the fields and controls later in this page.



For descriptions of all the fields and controls on the Favorites section template page, see Creating a Section Template.

The following table describes the fields that you should take note of and change, if required, after you import this section template to a target landing page, homepage, or dashboard.

Field or Control	*Description*
Section Loader Application Class	Use the PTNavCollection application class to dynamically load the specified navigation collection items in the section. It is an extension to the base class PTNUI:Dynamic:DynamicLoader.
Content Reference Attributes	• The PS_EXEC_ON_PORTAL attribute is introduced in PeopleTools 8.62.06. In this section template, it is set to false to retrieve information from the node where the section is defined. For more information on the PS_EXEC_ON_PORTAL attribute, see Understanding the PS_EXEC_ON_PORTAL Attribute. • PS_STYLECLASSES After importing this section template to the target landing page, you may change the styling of the section as required by your business needs. For a list of styles that are delivered by PeopleTools, see Styles for Layout and Behavior. In addition to the style classes, PeopleTools also delivers style class-based configuration to configure the data that is displayed in the section. For a list of style class-based configurations, see the Style Class-Based Data Configurations for Navigation Collection Section later in this topic.

Style Class-Based Data Configurations for the Navigation Collection Section

The style class-based data configurations should be placed in PS_STYLECLASSES and it should follow this format: data_NAME-VALUE. For example, data_IMAGENAME-DEFAULT.

Style	*Example*	*Description*
data_PTPPNAVCOLL-?	data_PTPPNAVCOLL-ADMN_ BRANDING	(Required) Replace the "?" with the CREF name of the desired navigation collection under the Portal Objects folder.

Style	***Example***	***Description***
data_IMAGENAME-?	data_IMAGENAME-PTNUI_EXPAND_ICN	Replace the "?" with the name of the default image to be used for all items within a navigation collection. The default display for this section is hyperlinks which does not require an image. If a tile presentation other than hyperlink is used, the default image will initially be pulled from how the navigation collection renders items in the panel form. If you specify an image name, it replaces the default image supplied by the navigation collection. This setting is optional.
data_MAXDEPTH-?	data_MAXDEPTH-2	Replace the "?" with the maximum levels (numeric value greater than zero) to recurse in the navigation collection hierarchy. The default is unlimited (that is, the full depth of the navigation collection). The navigation collection rendered on a panel presentation has a limit of 2 levels only. This can improve performance especially for those navigation collections using Menu links. This setting is optional.
data_MAXRESULTS-?	data_MAXRESULTS-5	Replace the "?" with the maximum navigation collection items (numeric value) to be displayed within the section. The default is all items or a value of 0. Because navigation collection items are fetched and then sorted for proper display (based on folder and sequence numbers), this configuration does not improve performance, but merely limits how many items are to be displayed within the section. This setting is optional.

Style	*Example*	*Description*
data_MAXFETCH-?	data_MAXFETCH-200	Replace the "?" with the maximum navigation collection items (numeric value greater than 0) to be displayed within the section. The default is a maximum of 100 items given usability within a section. Because navigation collection items are fetched and then sorted for proper display (based on folder and sequence numbers), this configuration improves performance, but does not guarantee the order of the items to be similar to data_MAXRESULTS. This setting is optional.
data_TILEPREFIX-?	data_TILEPREFIX-MYNAV	Replace the "?" with the string to be used as the prefix for the tile names (ID's) created dynamically in the section. The default is PTNVC. The alphanumeric string should be short (8 characters limit) and should not contain special characters. This configuration is only needed if more than one section is being used for navigation collection rendering on the same landing page. This setting is optional.

Best Practices

Follow these best practices for better performance:

- Use navigation collection-based sections for small collections; large collections can impact performance, especially when loaded remotely.

  You must exercise caution if you're including Menu folders in navigation collections because they can cause excessively large number of tiles to be created.
- Avoid configuring navigation collections with excessive depth or size.
- Some navigation collections may not operate properly within a section (especially those that load remotely), but the majority of the navigation collections should work as expected. You should remember to use navigation collections with a small number of items so as not to overload the homepage.

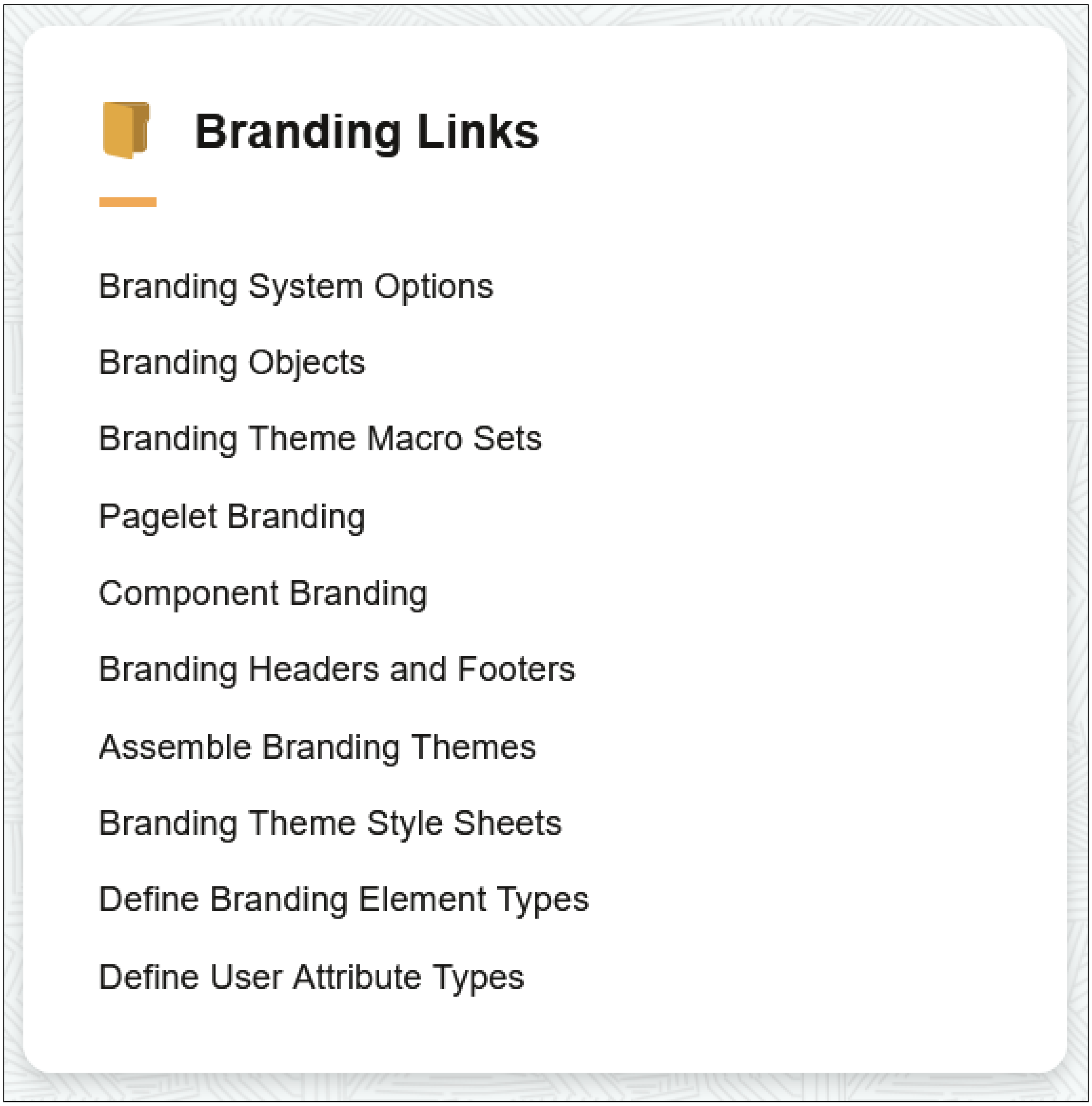
Examples of the Navigation Collections Section

The navigation collections section with the default settings is rendered as shown in this screenshot.

- psc_theme-section-all-primary-1

- psc_section-link-collection
- data_PTPPNAVCOLL-ADMN_BRANDING (The branding navigation collection is used only as an example to render the section; it is not the default value.)
- psc_sectgroup-C

This example illustrates the navigation collections section with default settings.



After importing the section template, in addition to the mandatory changes described previously in this topic, you may modify the style classes and style class-based data configurations as shown in the following example:

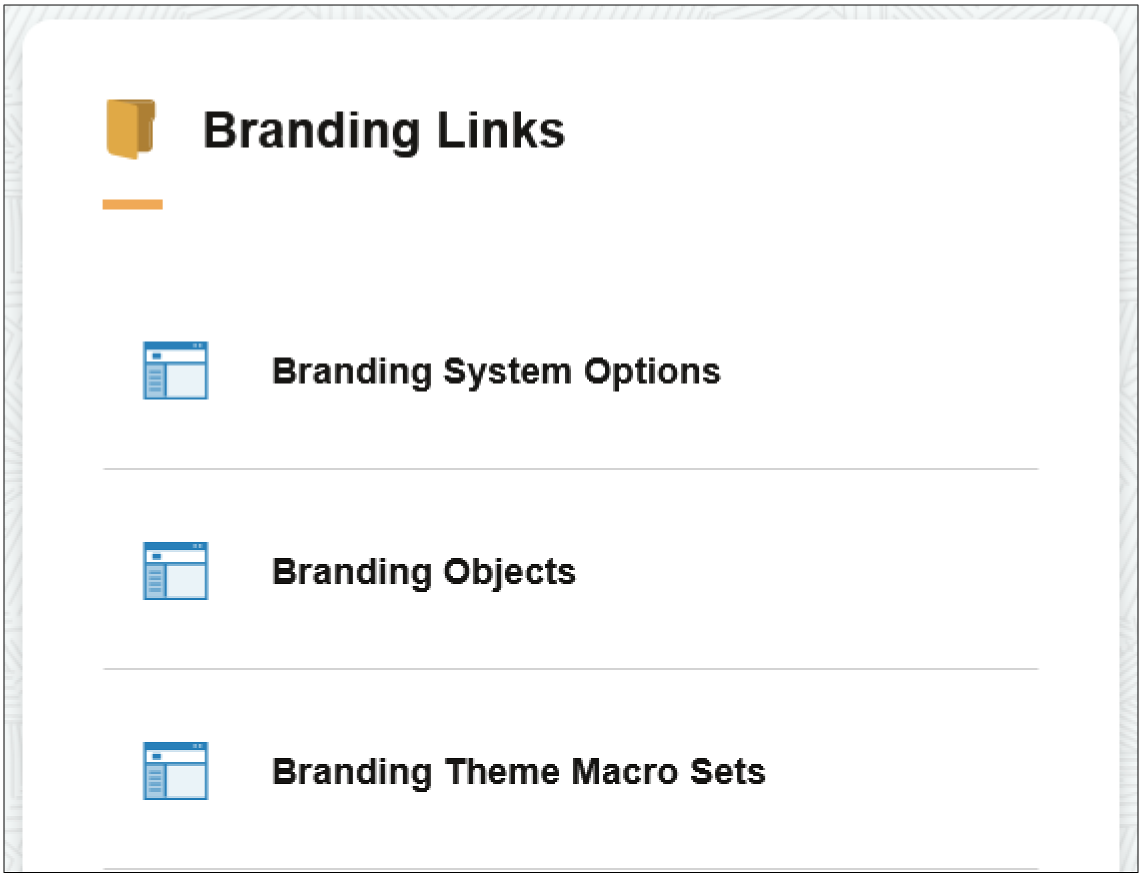
In PS_STYLECLASSES, enter the style classes separated by a semi-colon (;).

- psc_theme-section-primary-1
- psc_theme-tile-transparent

- psc_section-simple-list-collection
- data_PTPPNAVCOLL-ADMN_BRANDING
- psc_sectgroup-C

Based on styling and data configurations from the example, the navigation collections section is rendered as shown here:

This example illustrates the navigation collections section post import with modified style classes and style class-based data configurations.



Understanding the PS_EXEC_ON_PORTAL Attribute

The PS_EXEC_ON_PORTAL content reference attribute is introduced in PeopleTools 8.62.06.

The PS_EXEC_ON_PORTAL attribute controls the node location where the DynamicLoader application class is run for PeopleSoft portal content. By managing this attribute, administrators define whether data is consistently retrieved from the login (Portal) node or from the remote node.

This attribute is only applicable under the following conditions

- The environment is running in a cluster.
- The section is dynamic and defined exclusively on a remote node (either as an entirely remote homepage or as a section only present on a merged homepage through a remote definition).

Also note the following considerations when using the PS_EXEC_ON_PORTAL attribute:

- The term local refers to the login node (the node through which the user accessed the portal) where cluster aware data would be retrieved. Any other node in the cluster is considered remote.
- If a section definition exists locally, it will always be executed on the local (login) node, regardless of the attribute's value.
- If the PS_EXEC_ON_PORTAL attribute is not set, its default value is False. This means that the DynamicLoader application class for a remote section runs on the remote node where it is defined.

The PS_EXEC_ON_PORTAL attribute takes either of the two values: True or False.

- When PS_EXEC_ON_PORTAL is set to True, the DynamicLoader application class is run on the login node, making the rendered data consistent regardless of whether the landing page or section is local or remote. This approach ensures that items in Recently Visited and Favorites reflect user activity from the login node, supporting users with a unified view across clustered environments.

  In the Recently Visited and Favorites section templates, the default configuration is to run the DynamicLoader application class on the login node so that the retrieved data is consistent with the Recently Visited and Favorites items displayed in QAB and NavBar. This setting is not typically used with navigation collections.
- When PS_EXEC_ON_PORTAL is set to False, the DynamicLoader application class is run on the remote node where the landing page or section originates.

The DynamicLoader application class is a PeopleTools delivered class, so the PS_EXEC_ON_PORTAL works seamlessly as long as both the Portal and the remote nodes are running the same patch level of PeopleTools (8.62.06 or later), and the administrator does not have to ensure that the application class is present in both the nodes. However, in the case of custom application class, administrators must ensure that the custom application class is present on the login node and the remote node when using the PS_EXEC_ON_PORTAL attribute.

The PS_EXEC_ON_PORTAL attribute is primarily used for Recently Visited and Favorites, ensuring that this data is consistently retrieved from the Portal node. However, it is also relevant in clustered environments for other types of cluster-aware data, where information should be sourced specifically from the login node. While its main application is for Recently Visited and Favorites, administrators and application developers can use this attribute for other applications or custom solutions that require data to be pulled exclusively from the login node.

Guidelines for Advanced Sectionalized Homepages in a Cluster

For homepages and sections to appear correctly in a clustered environment, you must ensure that the custom style sheets (defined using the PS_STYLESHEETS content reference attribute) and section images are available on all application nodes participating in the cluster. You should manually copy the custom style sheets and section images on all application nodes that are part of the cluster.

Guidelines for Merged Homepages in a Cluster

The rules that govern the merging of homepages in a cluster also apply to the advanced sectionalized homepages and sections.

In a merged homepage, the following guidelines apply:

- If the Uses Custom Layout setting is specified for any of the homepages, the Uses Custom Layout setting is automatically applied to the merged homepage.
- If the Hide Notification Panel setting is specified for any of the homepages, the Hide Notification Panel setting is automatically applied to the merged homepage.
- The custom style sheets (defined using the PS_STYLESHEETS content reference attribute) from all the homepages are collected and appended to the HTML <head> tag of the merged homepage HTML document.
- The custom style classes (defined using the PS_STYLECLASSES content reference attribute) from all homepages are merged together and set in the HTML <div> tag of the merged homepage. Duplicate style classes are automatically removed.
- PeopleTools automatically adds a style class called node-<name of the node>, for example, node-HRMS, depending upon the application node where the first homepage was processed. In a merged homepage where the participating application nodes are HRMS, ERP, CRM and the acting portal node is HRMS, then the style class name is node-HRMS because the first homepage was processed on this node.

Guidelines for Merged Sections in a Cluster

Similar to the process of merging homepages, the sections with the same label are also merge.

The following guidelines apply for merged sections in a merged homepage:

- If any of the sections has the Hide Section Header attribute set, the Hide Section Header attribute is automatically applied to the merged section.
- If any of the sections has the Disable Collapse attribute set, the Disable Collapse attribute is automatically applied to the merged section.
- In a merged section, the first section image encountered during the merge process is displayed in the merged section based on these conditions:
    - If the section on the acting portal node has the section image set, then that section image is displayed.
    - If the section on the acting portal node does not have the section image set, then whichever remote section image is encountered first during the merge process is displayed.
- The custom style sheets (defined using the PS_STYLESHEETS content reference attribute) from all the sections are collected and appended to the HTML <head> tag of the merged homepage HTML document.
- The custom style classes (defined using the PS_STYLECLASSES content reference attribute) from all sections are merged together and set in the HTML <div> tag of the merged section. Duplicate style classes are automatically removed.
- Dynamic Section: This is the section that has an associated application package peoplecode class that loads the dynamic tiles. Admin can also add cref based tiles in this dynamic section. If Tools encounter dynamic sections with the same labels, only cref based tiles are merged together. Dynamic

tiles are not merged together. Only the local dynamic loader application package class is run. The remote loader app package class is not run.

- Remote Dynamic Section: If Tools encounter a dynamic section that is not locally defined or not defined in the local homepage, then this remote app package loader class will be run and the remote dynamic tiles are loaded.

Guidelines for Remote Homepages in a Cluster

The acting portal node retrieves remote homepages from application nodes.

The following guidelines are used when retrieving remote homepages:

- If a homepage has a dynamic section, its associated application package dynamic loader class is run to obtain the dynamic tiles.
- The custom style sheet of the homepage is appended to the HTML <head> tag of the fluid homepage HTML document.
- Any custom style sheet set for the section or tiles of the homepage is appended to the HTML <head> tag of the fluid homepage HTML document.
- PeopleTools automatically adds a style class called node-<name of the node>, for example, node-HRMS. The style class name indicates the application node of the remote homepage. In this example, the homepage is from the HRMS application node.

Guidelines for Content Reference Tiles

The following guidelines are used when retrieving content reference tiles:

- The custom style sheets (defined using the PS_STYLESHEETS content reference attribute) are appended to the HTML <head> tag of the overall fluid homepage HTML document.
- The custom style classes (defined using the PS_STYLECLASSES content reference attribute) are set in the HTML <div> tag of the tile. Duplicate style classes are automatically removed.

Application Nodes on PeopleTools Versions Prior to PeopleTools 8.62

When an application node running at least PeopleTools 8.62 receives a request to get its homepages and tiles, some homepages and tiles are not pushed out to remote nodes if the remote nodes are running PeopleTools versions prior to PeopleTools 8.62.

The following guidelines are applied when retrieving homepages and tiles on lower PeopleTools versions:

- Homepages and dashboards with sections are never pushed out to application nodes running PeopleTools versions 8.60 or lower.
- Application nodes running PeopleTools release 8.61 will display regular section normally, but no custom style sheets and no style classes will be applied. Custom style sheets and style classes defined in CREF tiles will also not apply in PeopleTools 8.61. Dynamic sections will display like a regular section. Dynamic tiles will look like regular tiles. No styles sheet and style classes are applied.

- In PeopleTools 8.62, the PT_MIN_VERSION CREF attribute can be used in the homepage or dashboard CREF, section CREF, and tile CREF to define the minimum PeopleTools version this object (that is, homepage, dashboard, section, or tile) can be pushed out to. If the requesting application node is running at a lower PeopleTools version than the specified PT_MIN_VERSION, then this application node will not receive this object.

  Consider the following scenario of a cluster and the use of the PT_MIN_VERSION CREF attribute. Application nodes HRMS and CRM form a cluster. While HRMS is running on PeopleTools 8.62, CRM is running on PeopleTools 8.61. Homepage A CREF of HRMS has the PT_MIN_VERSION attribute set to 8.62. On CRM, Homepage B CREF contains Section C and Section D, but Section D has the PT_MIN_VERSION attribute set to 8.62. If a user logs in directly to CRM (CRM is acting as portal gateway), the user will only see Homepage B with only Section C. Section D is not displayed because CRM is running on PeopleTools 8.61 and Section D requires PeopleTools version at 8.62. Homepage A is not displayed because it requires PeopleTools version at 8.62.

- Fluid dashboards can be made available to other remote environments through Add To Dashboard (or PinTo) where a user may be able to choose which fluid dashboard to pin a tile. The PT_MIN_VERSION CREF attribute can also be used to filter this dashboard out so remote nodes of lower PeopleTools release do not get this dashboard listed as an option on the Add To... dialog page (or Pin To).

Chapter 15

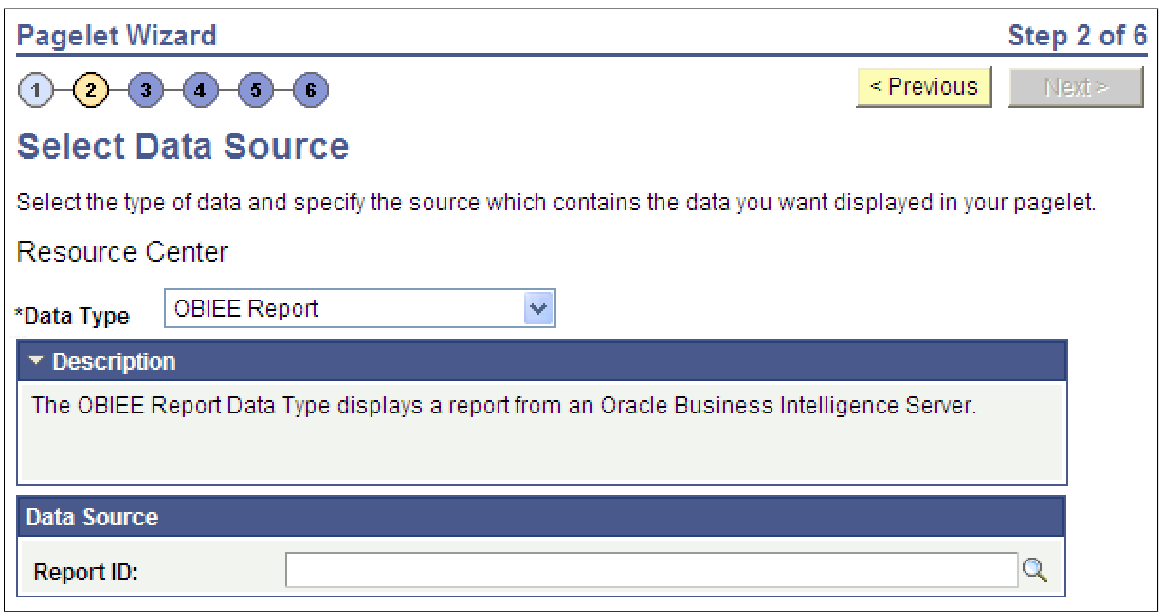
Using Pagelet Wizard to Create and Manage Pagelets

Understanding Pagelet Wizard

Pagelet Wizard provides a user-friendly, browser-based graphical user interface that leads you through the steps involved in creating and publishing a pagelet. Portal administrators and nontechnical users can use Pagelet Wizard to integrate and transform data from a variety of data sources.

You do not need to have PeopleSoft-specific application development tools or skills to use Pagelet Wizard to create pagelets. During the pagelet-creation process, Pagelet Wizard presents a series of numbered steps. Each step appears in a numbered path at the top of the page to indicate where you are in the pagelet creation process as shown in the following example:

The following example helps us understand the numbered steps in Pagelet Wizard.



This labeled path assists you in navigating forward and back through the steps in the wizard, which enables you to review or edit pagelet definition values. Once you've completed a step, Pagelet Wizard automatically takes you to the next appropriate step.

Pagelet Wizard leads you through the following steps that are typically used to create a pagelet:

1. Specify pagelet information.

2. Select data source.

3. Specify data source parameters.

4. Select display format.
5. Specify display options.
6. Specify publishing options.

Note: Before you start building a pagelet with Pagelet Wizard, you should familiarize yourself with all data types supported by Pagelet Wizard.

Related Links

Understanding Pagelet Wizard Data Sources

Understanding Pagelet Wizard Data Sources

Pagelet Wizard supports the creation of pagelets using the following data sources:

Term	***Definition***
Activity guides	Use to create activity guide pagelets for deployment in WorkCenters. The pagelet can be generated from an activity guide template and instance, from an activity guide template only, or without a template producing a generic activity guide pagelet.
Free text	Use the free text data type to present free-form HTML or simple text in a tile.
HTML	Use to present simple formatted HTML in a pagelet. HTML data source pagelets are compatible with the passthru and custom display formats. Because the HTML data type relies on the rich text editor (RTE) for text entry and formatting, complex HTML code might not appear correctly in a pagelet after RTE post-processing of the entered data. **Note:** The Secure PW Administrator role is required to be able to create or update pagelet definitions that use an HTML data source.
IB connectors (Integration Broker connectors)	Use to create pagelets that interact directly with secured, registered Integration Broker connectors within Pagelet Wizard. See “Connectors” (Integration Broker) for more information.

Term	***Definition***
Integration Broker	Use to create pagelets based on Integration Broker service operations. Integration Broker facilitates synchronous and asynchronous messaging among internal systems and with trading partners, while managing message structure, message format, and transport disparities. See “Understanding Managing Service Operations” (Integration Broker) for more information.
Navigation collections	Use to create pagelets from existing sets of folders and links stored in navigation collections. Navigation collection data source pagelets are compatible with the menu and custom display formats.
OBIEE reports (Oracle Business Intelligence Enterprise Edition reports)	Use to create pagelets from existing reports on the OBIEE server.
PeopleSoft queries	Use to access PeopleSoft application data or any tables within the PeopleSoft environment through Pagelet Wizard pagelets. PeopleSoft Query data source pagelets are compatible with the table, list, chart, and custom display formats. **Note:** If data masking for personally identifiable information or other sensitive data has been enabled on a record field that is part of the criteria for the PeopleSoft query, then these data values are masked when displayed via Pagelet Wizard pagelets.
Pivot grids	Use to create pagelets from existing pivot grid instances, which use interactive pivot tables and charts that display application data that you can filter and pivot across multiple dimensions.
Rowsets	Use to access a rowset record for which the data is supplied at runtime. This data source is geared toward supplying data to embedded pagelets. Rowset data source pagelets are compatible with the table, list, chart, and custom display formats. You designate a record as the rowset data source when you design your pagelet using Pagelet Wizard. When the pagelet is run for display, the data populated for the rowset must contain that record in the first level of the rowset. If the runtime rowset does not contain the record specified in Pagelet Wizard, no data appears in the pagelet.

Term	***Definition***
Search records	Use to create Pagelet Wizard pagelets that provide quick access to specific transactions on a target transaction page. The search record data type enables you to select a basic search record or a component interface. Use the basic search record data type for simple search lookup pages. Use the component interface search record data type when the basic search record does not provide the required functionality, for example, when security or PeopleCode considerations need to be taken into account. Search record data source pagelets are compatible with the search list display format.
URLs	The URL data type retrieves data from external sources such as the internet. This includes sources such as Real Simple Syndication (RSS) news feeds. Use to display appropriately sized dynamic HTML pages on Pagelet Wizard pagelets. URL data-source pagelets are compatible with the custom and passthru display formats.

Note: The Service Oriented Architecture Protocol (SOAP) data type used in previous releases is now included in the functionality of the IB data type. As of PeopleTools 8.50, SOAP services are consumed using Integration Broker, so use the IB data type to create pagelets going forward.
You can define your own data types on the Define Data Types page.
When you use PeopleSoft Interaction Hub, Pagelet Wizard includes other data sources that are specific to PeopleSoft Interaction Hub installations.

Related Links

Defining Display Formats
Defining Data Types
Step 2: Selecting a Pagelet Data Source

Understanding Pagelet Wizard Display Formats

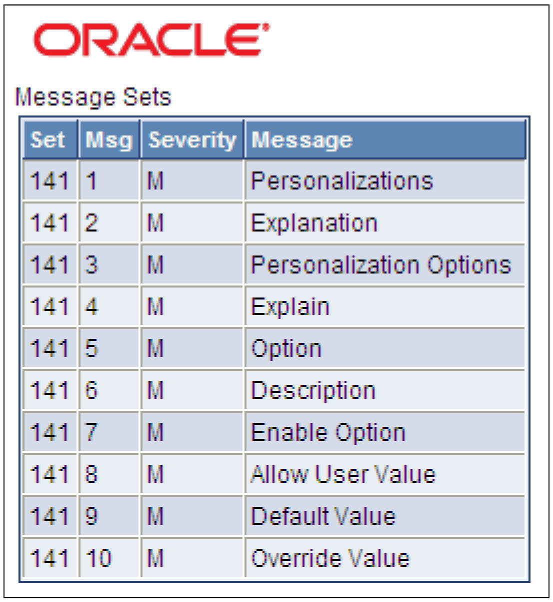
Pagelet Wizard supports the creation of pagelets based on the following display formats:

- Table
- List
- Chart
- Menu
- Search list

- Passthru
- Custom (Extensible Stylesheet Language, or XSL)

Table Display Format

The table display format is appropriate for creating a pagelet that displays data from the PeopleSoft query and rowset data sources. This example shows a table pagelet displaying data from a PeopleSoft query data source:



Set	Msg	Severity	Message
141	1	M	Personalizations
141	2	M	Explanation
141	3	M	Personalization Options
141	4	M	Explain
141	5	M	Option
141	6	M	Description
141	7	M	Enable Option
141	8	M	Allow User Value
141	9	M	Default Value
141	10	M	Override Value

List Display Format

The list display format is appropriate for creating a pagelet that displays data from the PeopleSoft query and rowset data sources. This example illustrates a list pagelet that displays data from a PeopleSoft query data source:

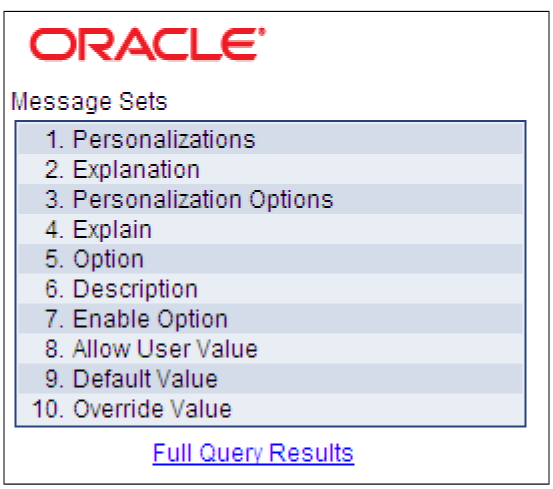
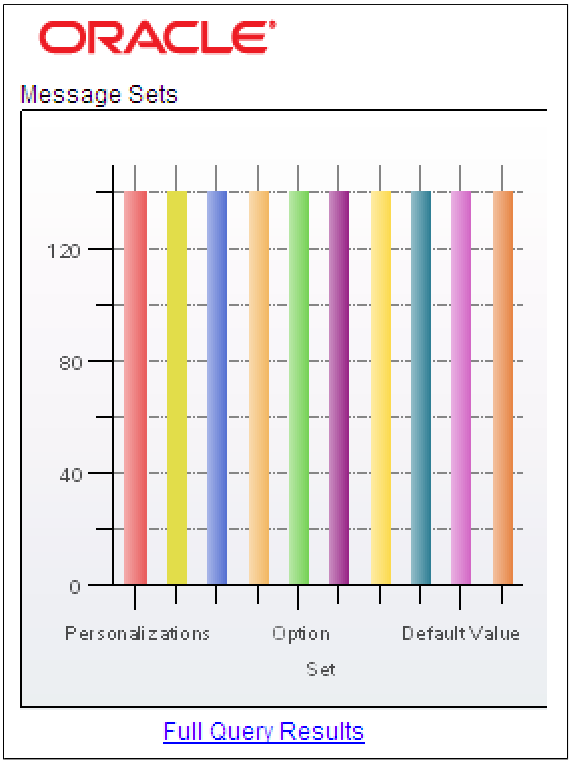


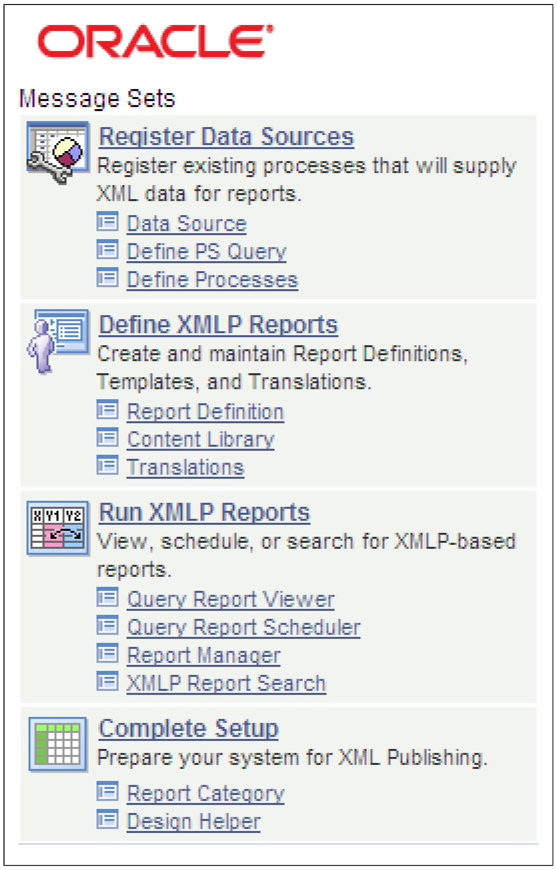
Chart Display Format

The chart display format is appropriate for creating a pagelet that displays data from the PeopleSoft query and rowset data sources. This example shows a chart pagelet displaying data from a PeopleSoft query data source.



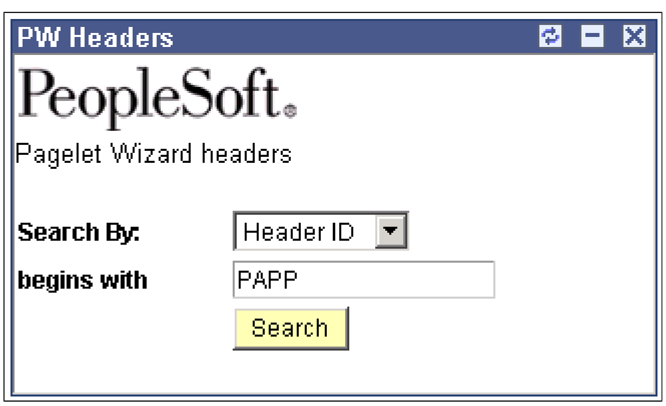
Menu Display Format

The menu display format is appropriate for creating a pagelet that displays data from a navigation collection data source. This example illustrates a menu pagelet displaying data from a navigation collection data source.

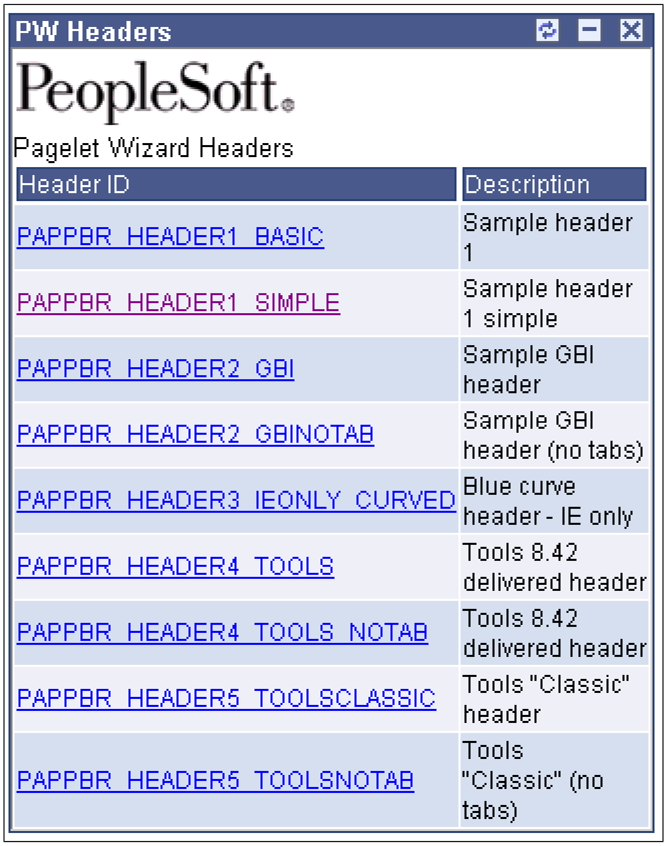


Search List Display Format

The search list display format is appropriate for creating pagelets that display data from the search record data source as shown in this example:

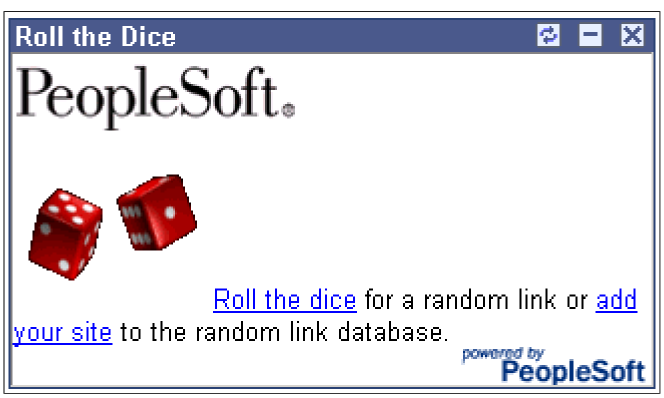


This example shows the same search list pagelet when the Show Results option has been set to yes.



Passthru Display Format

The passthru display format is appropriate for creating a pagelet that displays data from the HTML, IB Connector, and URL data sources. Because the passthru display format performs no further transformations on data from the data source, it should be used only with data sources that return HTML that can be directly displayed as a pagelet. The following example shows a passthru pagelet displaying data from an HTML data source:



Custom Display Format

The custom display format is appropriate for creating a pagelet using all data sources that return XML. Generally, most data types return XML, but circumstances potentially occur in which IB Connector, URL, and HTML data types do not return XML. In those instances, the custom display format would not be appropriate.

Use the custom display format when you want to use a display format other than the ones expressly listed in Pagelet Wizard. When you select the custom display format, Pagelet Wizard enables you to select a format provided by one of the delivered XSL prototypes, or you can select an XSL template that you created on the Define XSL page.

The following example shows a custom pagelet displaying data from a PeopleSoft query data source:

This custom pagelet uses the same PeopleSoft Query data source as the table pagelet example shown previously; however, this example has a custom XSL template applied.

Related Links

"Understanding the Charting Classes" (PeopleCode API Reference)

Defining Pagelet Wizard XSL Prototypes

Step 4: Selecting a Pagelet Display Format

Understanding Pagelet Wizard Display Sizes

In addition to the display format, pagelet sizing is also important during pagelet creation. Pagelet Wizard uses pagelet sizing that follows a basic set of rules so that pagelets are sized appropriately for PeopleSoft portal homepages. Pagelet size corresponds to the homepage layout that is specified. You can specify either a two-column layout that displays one narrow pagelet and one wide pagelet, or a three-column layout that displays three narrow pagelets.

Pagelets that you design with Pagelet Wizard should conform to the dimensions of the narrow column pagelet and (optionally) the wide column pagelet. A narrow pagelet should be 240 pixels wide. Subtracting the border and the internal margin provides 218 pixels for content. A wide pagelet should be 490 pixels wide. Subtracting the border and the internal margin provides 468 pixels for content.

If you display a pagelet that is wider than the homepage column that contains it, an error does not occur. The column expands to accommodate the content; however, this may require horizontal scrolling on the homepage, which can be inconvenient for the end user. In accordance with general pagelet design principles, make the pagelet height as short as possible.

Understanding Pagelet Wizard Pagelet Types

Pagelet Wizard supports the publication of the following pagelet types:

- Homepage pagelets.
- Template pagelets.
- Embeddable pagelets.

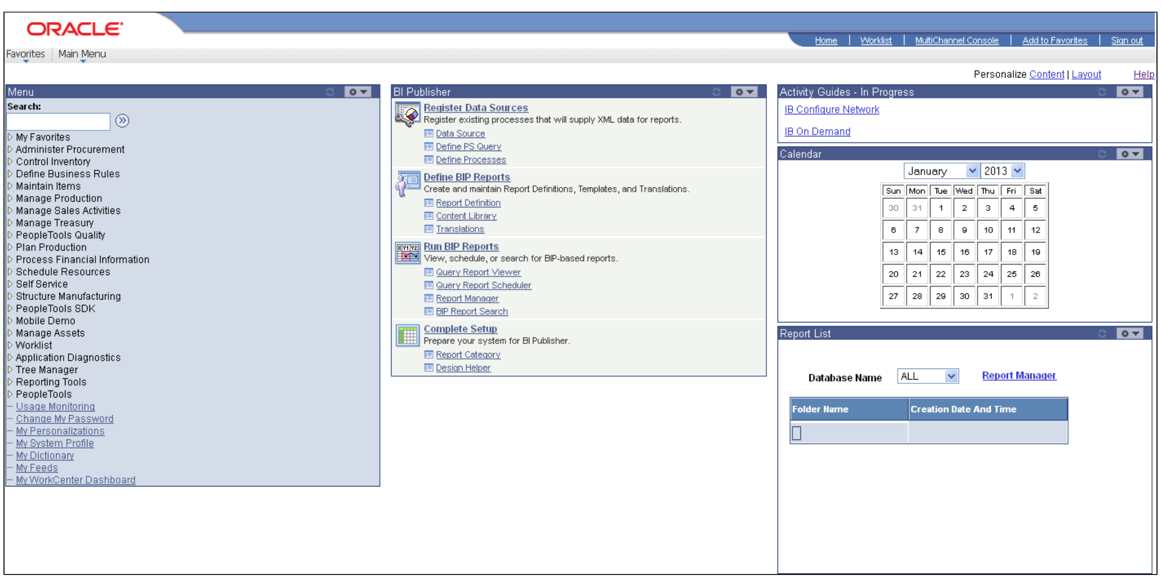
Homepage Pagelets

Homepage pagelets are pagelets that are specifically for use on portal homepages. The layout and content of the homepage is usually configurable by the end user, and homepage pagelets can be designated as optional or required.

Note: On any PeopleSoft homepage, only one pagelet or portlet should use the calendar prompt. If more than one pagelet or portlet on a homepage uses the calendar prompt, unexpected results may occur.

See Selecting the Pagelets or Tiles That Appear on Homepages.

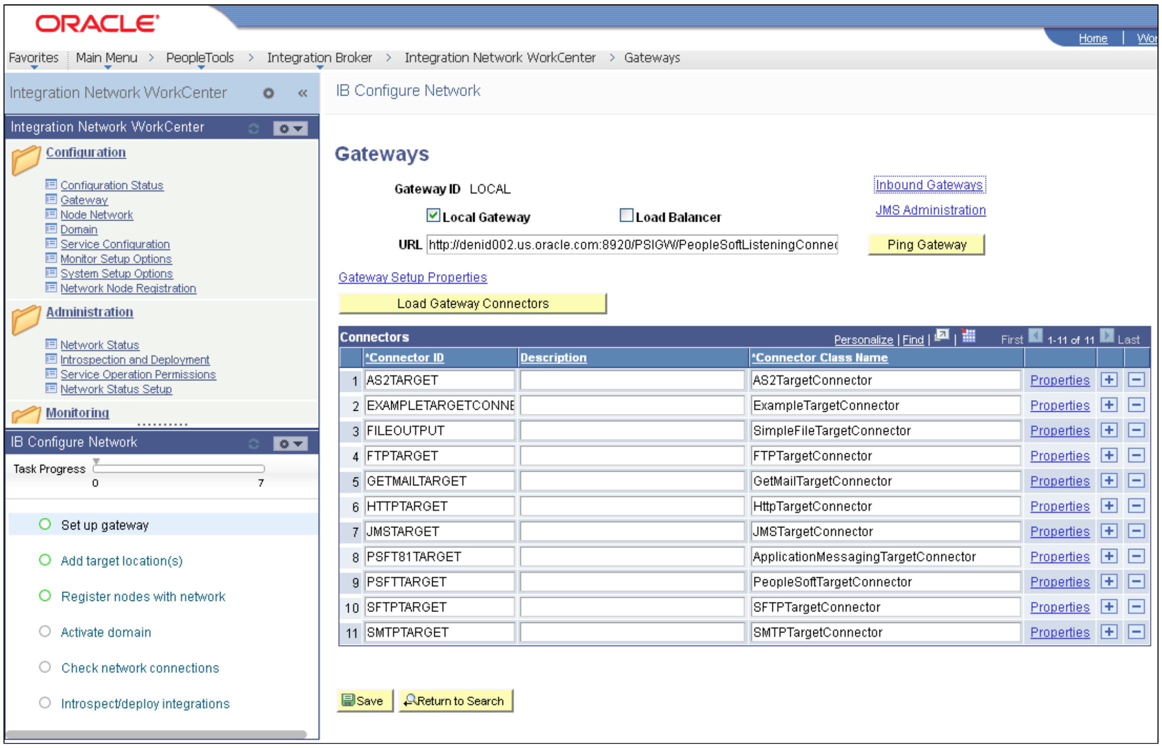
The layout and content of the homepage is usually configurable by the end user. As shown in this example, in addition to the Menu pagelet, the user has added four additional pagelets to his or her portal homepage.



Template Pagelets

Template pagelets are pagelets that can be added to a PeopleSoft portal template. These pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful when you use Context Manager to display context-sensitive data in the template pagelet. Pagelets that you intend to use in the task panel of WorkCenter pages can also be template pagelets.

Each pagelet in the Integration Network WorkCenter is a template pagelet:



Embeddable Pagelets

Embeddable pagelets are pagelets that you can embed in PeopleSoft transaction pages or iScripts within an HTML area. For example, you might create an OBIEE-based pagelet that displays a report of sales by region to embed on a Regional Sales page so that you could compare the sales of individual regions or each region to the whole.

Related Links

[Step 6: Specifying Pagelet Publishing Options]
[Configuring WorkCenter Pages]
[Understanding Web Libraries]

Defining Pagelet Wizard Setup Values

This section discusses how to:

- Set up the PeopleSoft Integration Broker (IB) gateway for Pagelet Wizard.

- Set up the default gateway for use with Pagelet Wizard data sources.
- Set up the Pagelet Wizard IB Connector data source security.
- Set up the Pagelet Wizard Integration Broker data source.

Setting Up the PeopleSoft Integration Broker Gateway for Pagelet Wizard

This section provides an overview of setting up the PeopleSoft Integration Broker Gateway for Pagelet Wizard and discusses how to set up the default gateway for use with Pagelet Wizard data sources.

When you use Pagelet Wizard with PeopleSoft Interaction Hub installed, the IB Connector and Integration Broker data sources become available for use. These data sources rely on PeopleSoft Integration Broker for web interactivity. You must set up the default gateway before these data sources are usable with your system.

Setting Up the Default Gateway for Use with Pagelet Wizard Data Sources

Access the Gateways page. (Select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**.)

Set up the default gateway according to the instructions in the Integration Broker PeopleBook.

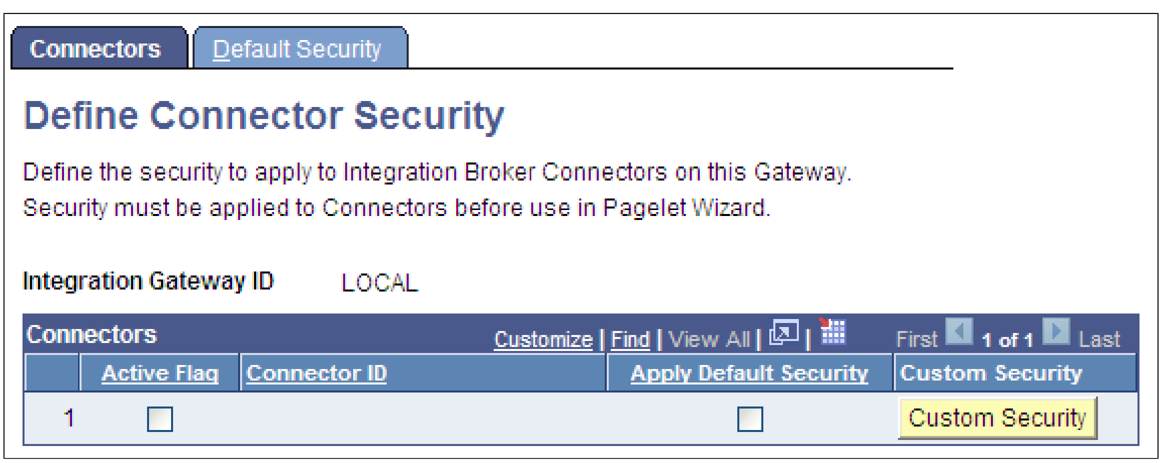
See "Understanding Managing Integration Gateways" (Integration Broker Administration).

1. Click **Save.**
2. Click **Load Gateway Connectors.**
3. Click **Save.**

Setting Up the Pagelet Wizard IB Connector Data Source Security

Access the Define Connector Security - Connectors page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define IB Connector Security** > **Connectors**.)

This example illustrates the fields and controls on the Define Connector Security - Connectors page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Integration Gateway ID	Identifies the Integration Gateway.
Active Flag	Select to activate the connector and make it available for use with the Pagelet Wizard IB Connector data source.
Apply Default Security	Select to apply the default security settings that are defined on the Define IB Connector Security - Default Security page to the connector.
Custom Security	Click to access the Define Connector Security page, where you can define custom security settings for the connector.

Related Links

“Configuring the Integration Network” (Integration Broker Administration)

Setting Up the Pagelet Wizard Integration Broker Data Source

When you use Pagelet Wizard with PeopleSoft Interaction Hub installed, the Integration Broker data source becomes available for use. You must perform the following tasks before the Integration Broker data source is usable with your system.

1. Set up the Integration Broker listening connector.

   See “Configuring the Integration Network” (Integration Broker Administration).

2. Set up the appropriate service, service operation, handlers, and routings for the operation that you want to use.

   See “Configuring the Integration Network” (Integration Broker Administration).

Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types

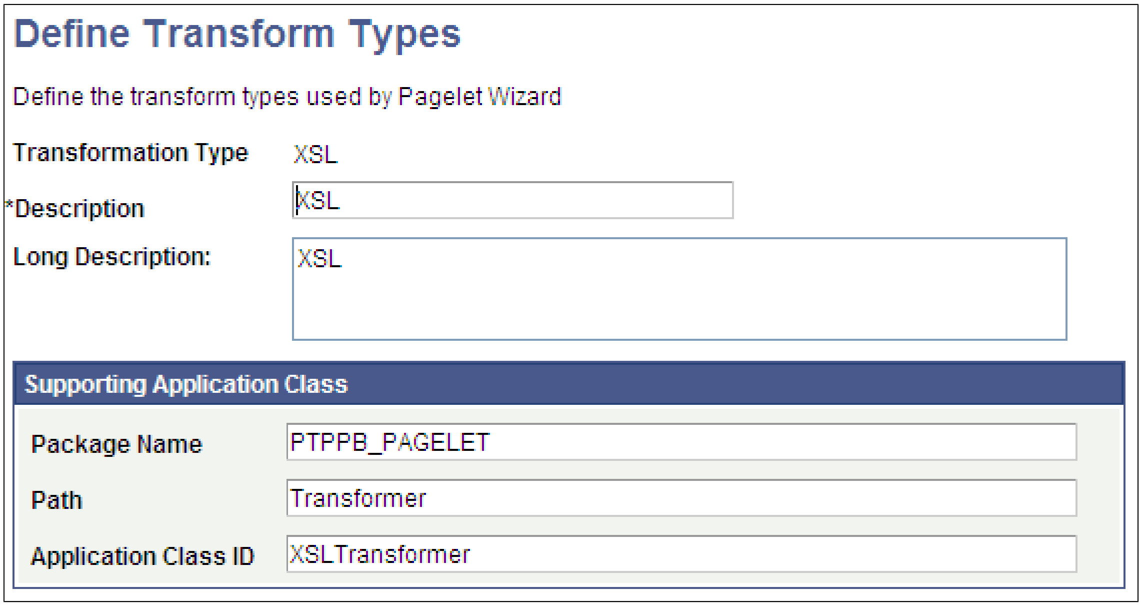
This section discusses how to:

- Define transformation types.
- Define display formats.
- Define data types.

Defining Transformation Types

Access the Define Transform Types page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Transform Types**.)

This example illustrates the fields and controls on the Define Transform Types page. You can find definitions for the fields and controls later on this page.



Pagelet Wizard is delivered with the following transformation type definitions to support their associated display formats:

- PASSTHRU
- XSL

Warning! Do not alter delivered field values for these transformation types.

The Define Transform Types page enables you to build your own transformation types. For example, you could create a new transformation type called ABC123. You would then extend Pagelet Wizard to perform ABC123 transformations by building a supporting application class and registering it as a transformation type. You could then create builders for assembling pagelet displays that create ABC123 transformation definitions.

See Understanding Pagelet Wizard Display Formats.

Field or Control	***Description***
Transformation Type	Enter the internal name that you want to assign to the transformation type.
Description	For reference purposes only, enter a short description of the transformation type.
Long Description	For reference purposes only, enter a more detailed description of the transformation type.

Supporting Application Class

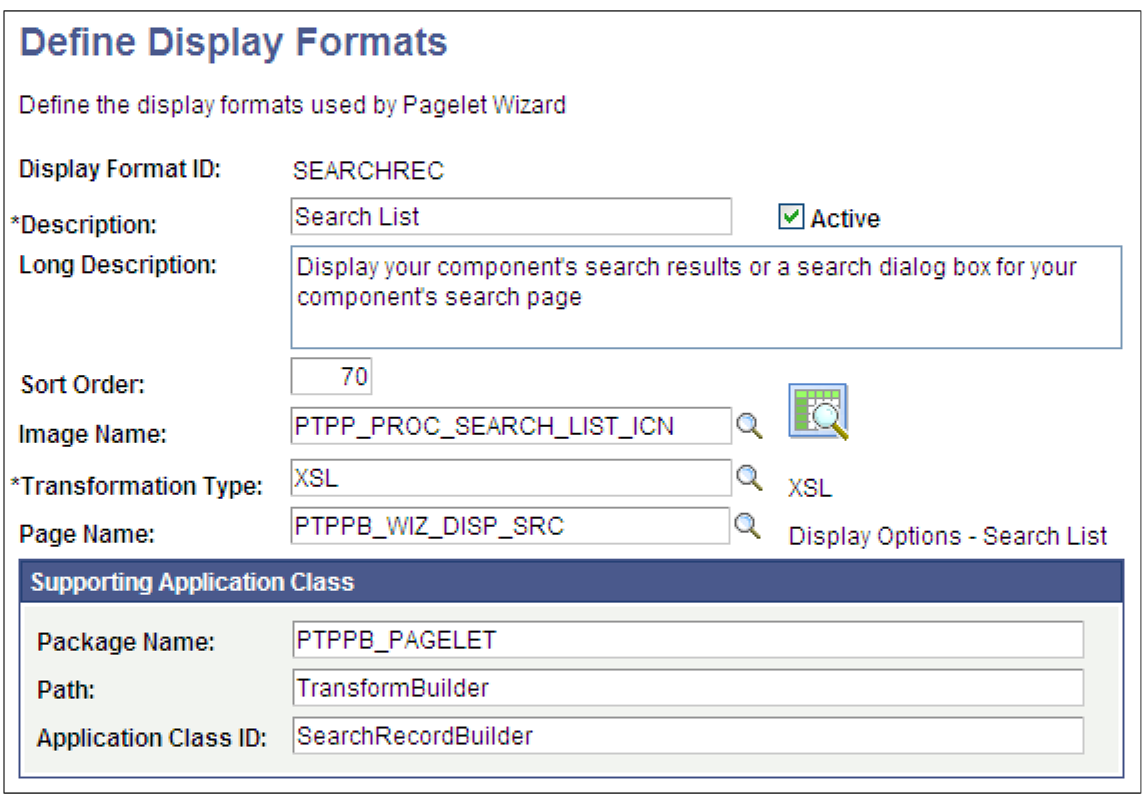
The application class specified here refers to the PeopleCode that implements the transformation type.

Field or Control	***Description***
Package Name	Enter the name of the application class package that you want to use for the transformation type.
Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method to use for the transformation type.
Application Class ID	Enter the name of the application class that you want to use for the transformation type. The class must exist in the application package name that you have specified.

Defining Display Formats

Access the Define Display Formats page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Display Formats**.)

This example illustrates the fields and controls on the Define Display Formats page. You can find definitions for the fields and controls later on this page.



A display format performs the following tasks:

- Defines a transformation builder, which is code that builds a particular type of display format (chart, menu, list, and so forth).
- Identifies the proper transformation to use with pagelets created with a specific display format builder.

For example, the chart display format defines the code that builds charts, and it specifies that Pagelet Wizard must use the XSL transformation for pagelets created using this display format.

Pagelet Wizard is delivered with the following display format definitions to support their associated display formats:

- CHART
- CUSTOM
- LIST
- MENU
- PASSTHRU
- SEARCHREC
- TABLE

Warning! Do not alter delivered field values for these display formats.

See Understanding Pagelet Wizard Display Formats.

Field or Control	***Description***
Display Format ID	Enter the name that you want to assign to the display format.
Description	Enter a short description of the display format. This short description appears as the display format name value on the Select Display Format page in Pagelet Wizard.
Active	Select to activate the display format and make it a selectable value in Pagelet Wizard. If you deactivate a display format, the display format is no longer available for selection on the Select Display Format page when accessed for associated data types. Existing pagelets that have been created by means of a deactivated display format are unaffected; however, you can no longer use Pagelet Wizard to change the display settings for those pagelets.
Long Description	Enter a detailed description of the display format. This long description appears as the display format description on the Select Display Format page in Pagelet Wizard.

Field or Control	***Description***
Sort Order	Enter a numeric value to specify the order in which you want the display format to appear as a selectable field value relative to other display formats on the Select Display Format page. See Step 4: Selecting a Pagelet Display Format.
Image Name	Select the image that you want to appear with the display format name and description on the Select Display Format page.
Transformation Type	Display formats use a transformation technology. Select the transformation type that you want to use for the display format. Available transformation types are defined on the Define Transform Types page. See Defining Transformation Types.
Page Name	Enter the object name of the page to be used to define display options for this display format. This page must exist in the PTPPB_WIZARD component in PeopleSoft Application Designer.

Supporting Application Class

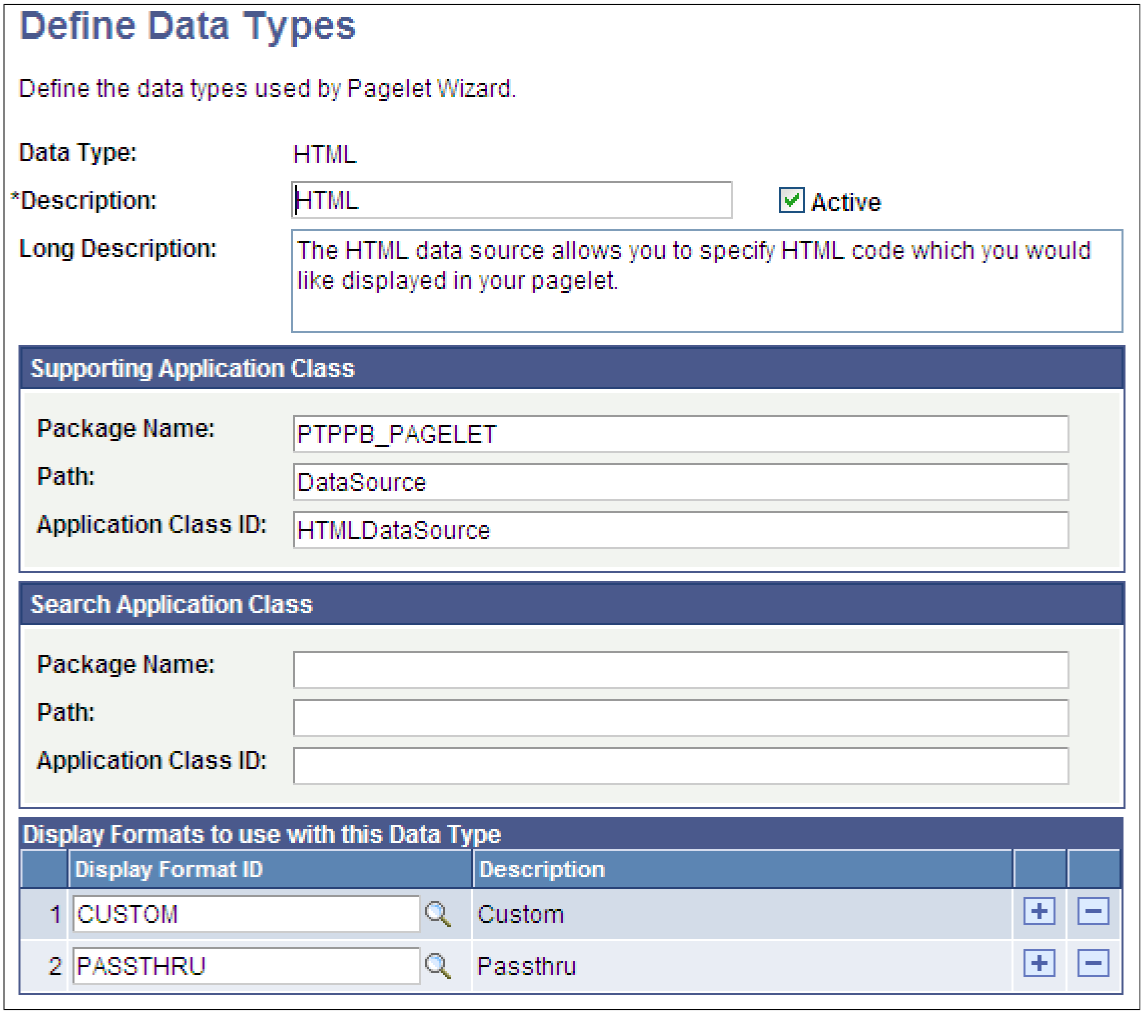
The application class specified here refers to the PeopleCode that implements the display format. This code is the builder that creates the definition that Pagelet Wizard uses (along with the transformation) to display the pagelet.

Field or Control	***Description***
Package Name	Enter the application class package name to use for the display format.
Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method used to create the display format.
Application Class ID	Enter the name of the application class to use for the display format. The class must exist in the application package name that you specify.

Defining Data Types

Access the Define Data Types page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Data Types**.)

This example illustrates the fields and controls on the Define Data Types page. You can find definitions for the fields and controls later on this page.



Pagelet Wizard is delivered with the following data type definitions to support their associated data sources:

- ACTIVITY_GUIDE data type enables you to generate activity guide pagelets.
- FREE_TEXT data type allows you to generate pagelets from any complex, free-form HTML or text.
- HTML data type allows you to generate pagelets from any simple, formatted HTML code.
- IBCONNECTOR data type interacts directly with connectors registered to the Integration Broker and secured using Define IB Connector Security within Pagelet Wizard. This data type should be used only when it is not desirable to use the Integration Broker data type.
- IBDATASOURCE data type retrieves messages using Integration Broker. Specifically, it uses outbound synchronous messages to retrieve data from external sources.
- NAVIGATION_COLLECTION data type enables you to display navigation collections within the context of a pagelet.
- OBIEE data type interacts directly with the analytics server.

- PIVOTGRID data type enables you to display pivot tables and charts in pagelets.
- PSQUERY data type enables you to use PeopleSoft Query to create database queries on records and fields within the PeopleSoft system.
- ROWSET data type is used with embedded pagelets. Rowsets can exist within components.
- SEARCHREC data type displays the search box or the search results from a given component.
- URL data type retrieves data from external sources such as the internet. This includes sources such as news feeds.

Warning! Do not alter delivered field values for these data types.

See Understanding Pagelet Wizard Data Sources.

Field or Control	***Description***
Data Type	Enter the internal name that you want to assign to the data type.
Description	Enter a short description of the data type. This short description appears as the prompt list value when you select data sources in Pagelet Wizard.
Active	Select to activate the data type and make it a selectable value in Pagelet Wizard. If you deactivate a data type, the data type is no longer available for selection on the Select Data Source page. If you deactivate a data type that is in use, associated pagelets do not display data. The pagelets display an error message instead.
Long Description	Enter a detailed description of the data type. This long description appears as the data type description on the Select Data Source page in Pagelet Wizard.

Supporting Application Class

The application class specified here refers to the PeopleCode that implements the data type. This is the code that Pagelet Wizard calls to retrieve the data for the pagelet.

This table describes the Supporting Application Class fields:

Field or Control	***Description***
Package Name	Enter the name of the application class package that you want to use for the data type.

Field or Control	Description
Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method that defines the data type.
Application Class ID	Enter the name of the application class that you want to use for the data type. The class must exist in the application package name that you specify.

Note: The search option of each pagelet has to be explicitly declared through this page or Pagelet Wizard to activate the search box, even for *Inline XSL* type of search box.

Search Application Class

The application class specified here refers to the PeopleCode that implements a default pagelet search for this data type. This is the code that Pagelet Wizard calls to build and place the search box in the pagelet, perform the search, and render and format the search results.

Changes to the default search application class at the data type level appear in any pagelet of this data type automatically the next time the pagelet is rendered, with the exception of the pagelets for which you define a custom search application class. Examples of these changes are included in the subsequent list.

- Changing the code in the search class.
- Adding a new search class.
- Removing a search class.
- Replacing a search class.

The *PTPPB_PAGELET:PageletSearch:PortalSearch* sample search application class is part of a standard PeopleTools installation. This class uses the portal registry index to perform the search. You can use this sample search class to test the pagelet search functionality for any pagelet or you can associate this search class with either a pagelet data type or with a pagelet.

Note: Pagelet search is supported for homepage and embedded pagelets only.

This table describes the Search Application Class fields:

Field or Control	Description
Package Name	Enter the Search application class package name that you want to use for the data type.
Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method to define the search.

Field or Control	*Description*
Application Class ID	Enter the name of the Search application class that you want to use for the data type. The class must exist in the application package name that you specify.

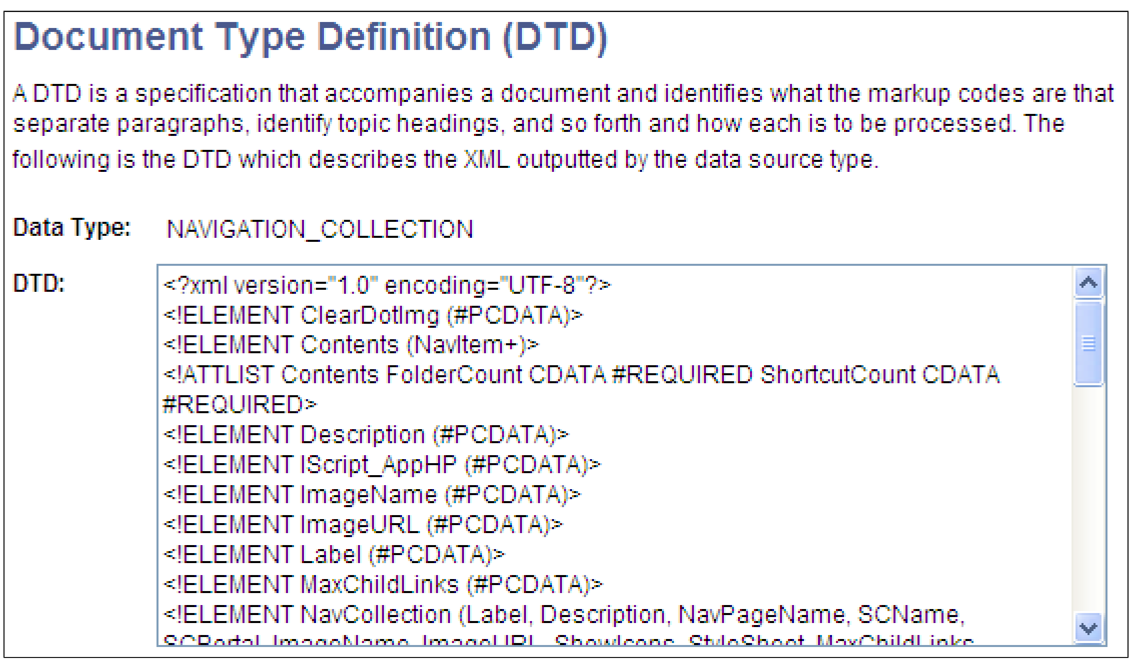
Display Formats to Use with This Data Type

Field or Control	*Description*
Display Format ID	Enter the display formats that are supported by the data type. You can select from existing display formats defined on the Define Display Formats page. See Defining Display Formats.
View Document Type Definition (DTD)	Click to access the Document Type Definition (DTD) page, where you can access the DTD that describes the XML generated by the selected data type.

Viewing the Document Type Definition (DTD)

The Document Type Definition (DTD) page enables you to view the tags and attributes that are used to describe content in the XML from the data source type. This information is useful when you use third-party tools to generate XSL templates for the data type.

This example illustrates a DTD specification on the Document Type Definition (DTD) page.



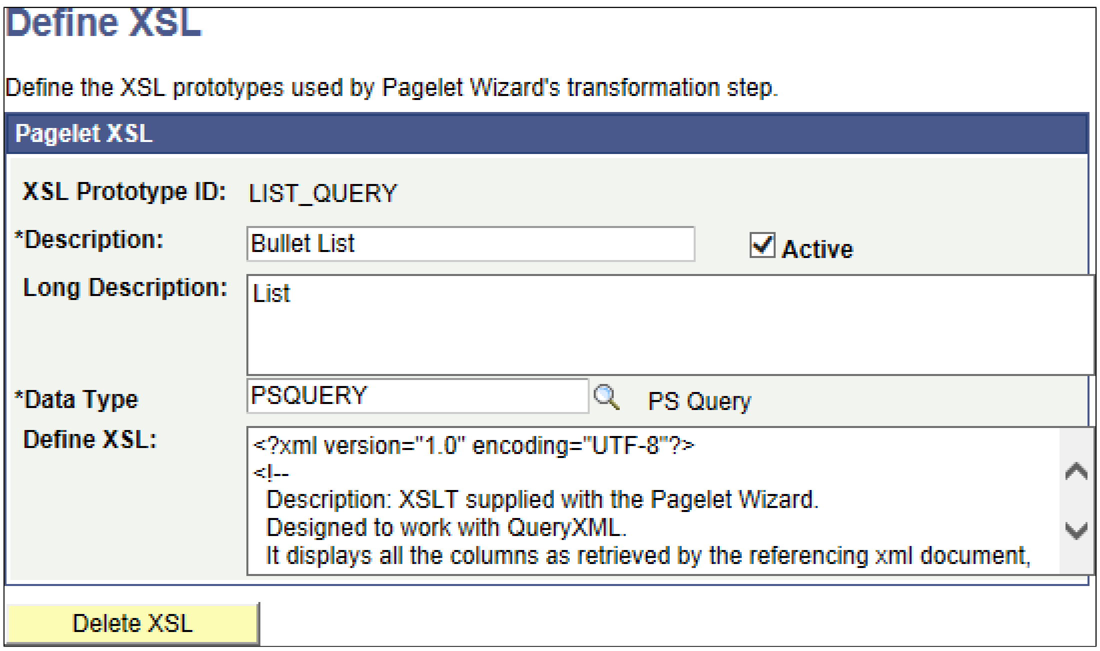
Defining Pagelet Wizard XSL Prototypes

This section discusses how to define XSL prototypes for use in building Pagelet Wizard pagelets.

Defining XSL Prototypes

Access the Define XSL page. (**PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet XSL**).

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



Pagelet Wizard is delivered with the following XSL prototypes to support their associated data sources and display formats:

XSL Prototype Name	***Data Type***	***Description***
LIST_QUERY	PSQUERY	Bulleted list for query data sources.
NAV_SMALLIMAGES	NAVIGATION_ COLLECTION	Displays the navigation collection with small images for folders and links. Descriptions are displayed as hover text for links only. **Note:** Only the parent folders and not subfolders will be displayed as expanded when clicked. Use the PTPP_SELFSERVICE data source as an alternative.
NUMBERED_LIST_ QUERY	PSQUERY	Numbered list for query data sources.

XSL Prototype Name	***Data Type***	***Description***
PTPPB_NAVCOLL_ ACCORDION	NAVIGATION_ COLLECTION	Displays an accordion menu.
PAPP_IB_BULLET_LIST	IBDATASOURCE	Bulleted list.
PAPP_IB_CUSTOM_LIST	IBDATASOURCE	Customizable list.
PAPP_IB_CUSTOM_ TABLE	IBDATASOURCE	Customizable table.
PAPP_IB_NUMBER_LIST	IBDATASOURCE	Numbered list.
PAPP_IB_NUMBER_LIST _H	IBDATASOURCE	Numbered list with a header.
PAPP_IB_SKELETON	IBDATASOURCE	Basic XSL.
PAPP_IB_TABLE	IBDATASOURCE	Table.
PTPP_SELFSERVICE	NAVIGATION_ COLLECTION	Displays the navigation collection with large images for folders only; links are displayed without images. All folder and link items are underlined only on hover over. Descriptions are displayed as hover text for links only.
PAPP_URL_ATOM03	URL	ATOM 0.3.
PAPP_URL_ATOM10	URL	ATOM 1.0.
PAPP_URL_RSS	URL	RSS.
PAPP_URL_RSS091	URL	RSS 0.91-0.93.
PAPP_URL_RSS10	URL	RSS 1.0.
PAPP_URL_RSS20	URL	RSS 2.0.
PAPP_URL_SKELETON	URL	Basic XSL.
TABLE_QUERY	PSQUERY	Table

Warning! Do not alter the delivered XSL for these XSL prototypes. Instead, clone and then modify your copy of the XSL prototypes.

Field or Control	***Description***
XSL Prototype ID	Enter an ID for the XSL prototype.
Description	Enter a description of the XSL prototype. The text that you enter here appears as available XSL template field values on the Specify Display Options page when you create a custom display format pagelet. See Specifying Custom Display Options.
Active	Select to activate the XSL prototype and make it an available XSL template field value on the Specify Display Options page for custom display format pagelets. If you deactivate an XSL prototype, the prototype is no longer available for selection on the Specify Display Options page for the custom display format. Existing pagelets that have been created by means of the deactivated XSL prototype are unaffected.
Long Description	For reference purposes only, enter a long description of the XSL prototype.
Data Type	Select the applicable data type for the XSL prototype. You can assign only one data type per XSL prototype.
Define XSL	Enter the XSL code for formatting pagelet XML data in the desired pagelet format.
Delete XSL	Click to delete the XSL prototype definition. You will be prompted to confirm your deletion.

Defining Pagelet Wizard Headers and Footers

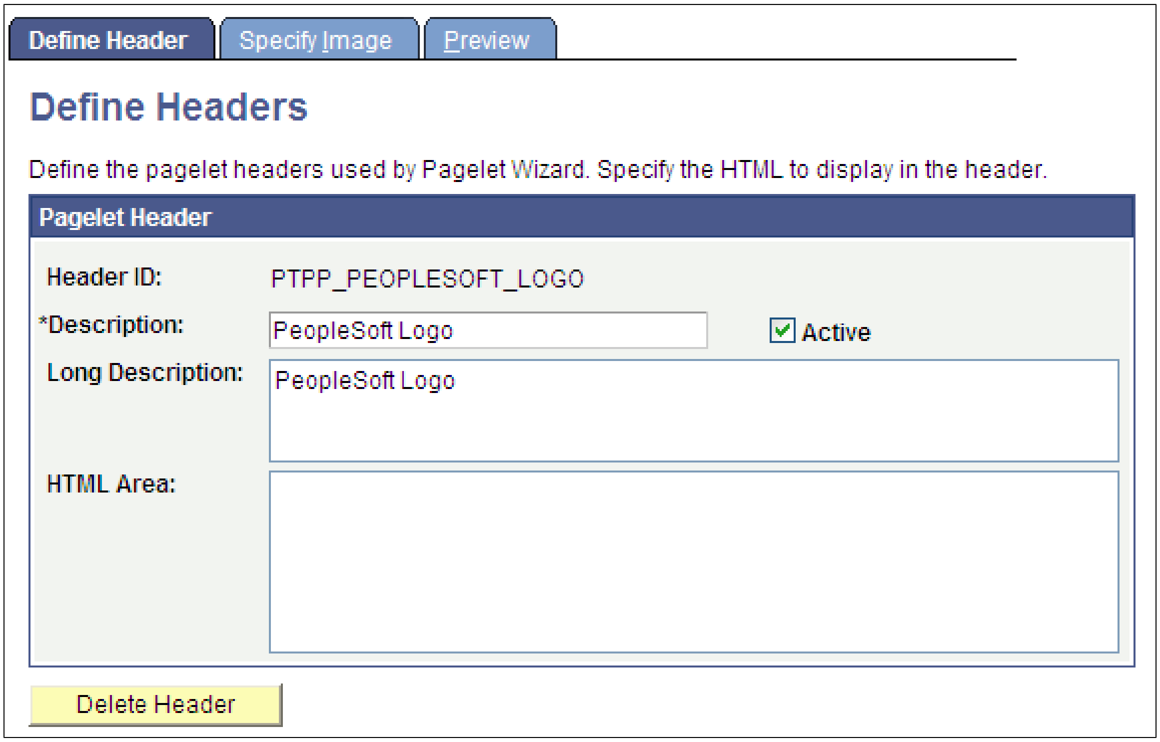
This section discusses how to:

- Define headers.
- Define header images.
- Define footers.
- Define footer images.

Defining Headers

Access the Define Header page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Headers** > **Define Header**.)

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



Pagelet Header

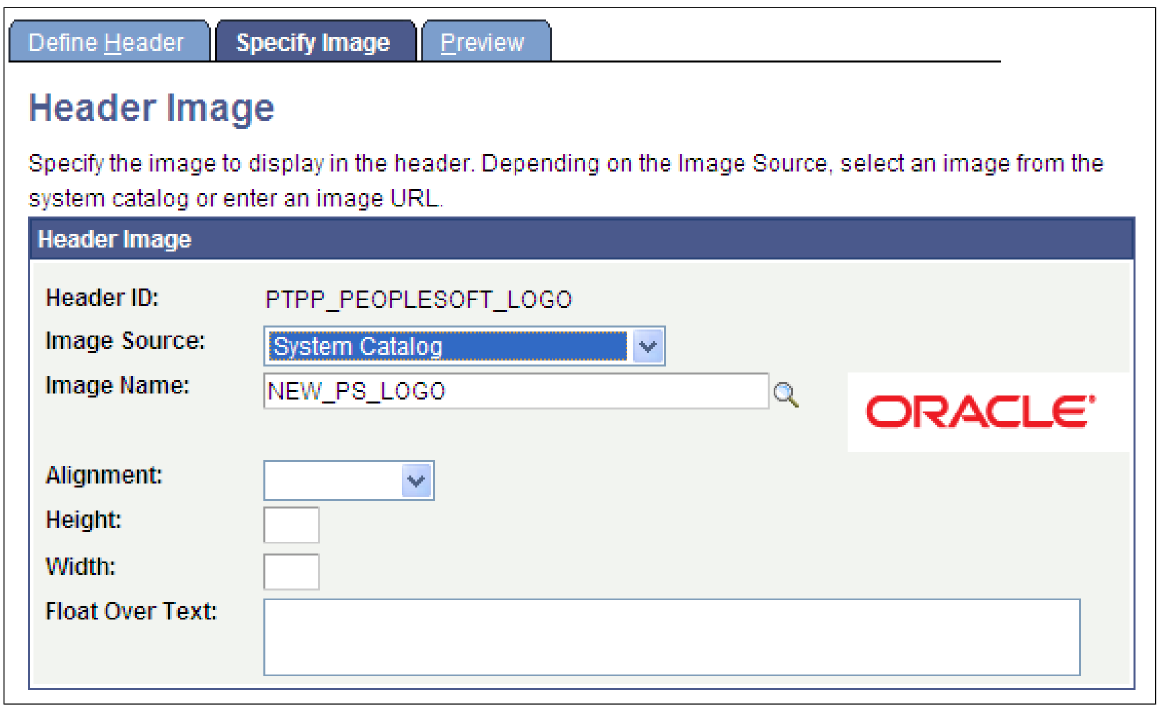
Field or Control	*Description*
Header ID	Enter an ID for the pagelet header.
Description	Enter a description of the pagelet header. The text that you enter here appears as the selectable field value in the **Header** field when users assign headers to pagelets on the Specify Display Options page.
Active	Select to activate the pagelet header and make it a selectable field value in the **Header** field on the Specify Display Options page. If you deactivate a header that is in use, the header no longer appears on associated pagelets. In addition, the header is not available for selection on the Specify Display Options page.
Long Description	For reference purposes only, enter a long description of the pagelet header.

Field or Control	*Description*
HTML Area	Enter the HTML code that you want to use to generate the pagelet header. You can enter both HTML text and image references. Alternatively, you can include an image by selecting it on the Specify Image page.
Delete Header	Click to delete the pagelet header design. You will be prompted to confirm your deletion.

Defining Header Images

Access the Header Image page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Headers** > **Define Header** > **Specify Image**.)

This example illustrates the fields and controls on the Define Headers - Specify Image page. You can find definitions for the fields and controls later on this page.



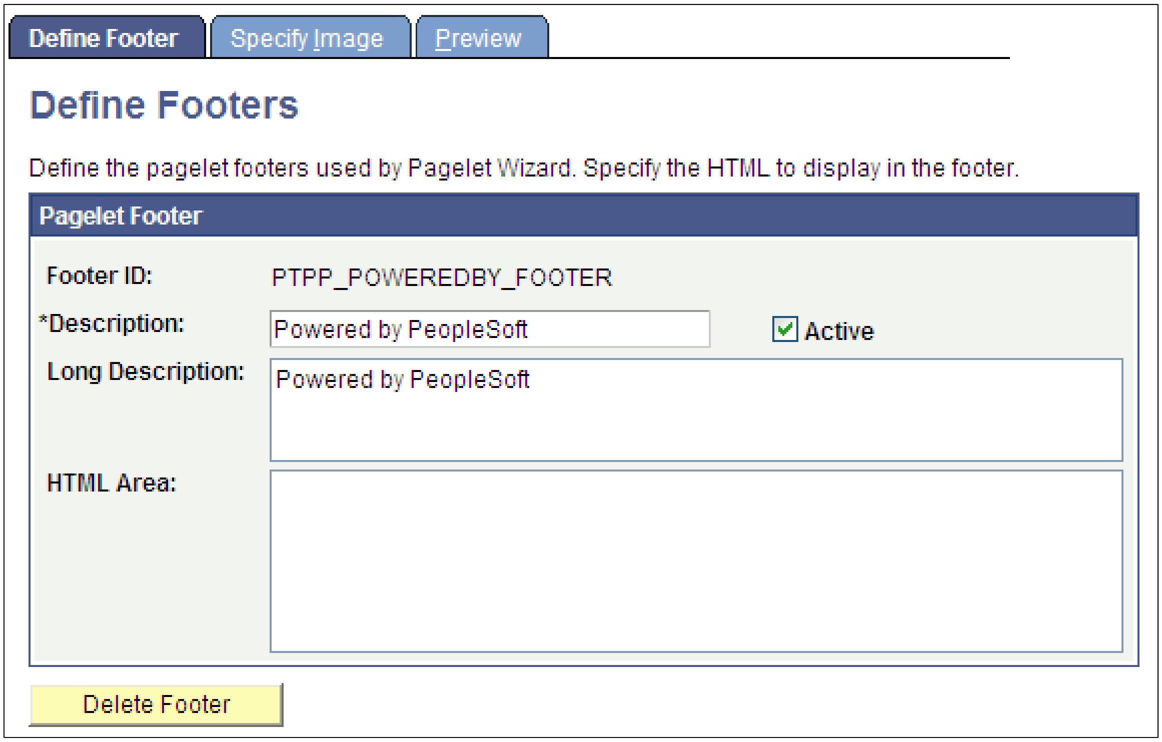
Header Image

Field or Control	***Description***
Image Source	Select the image source. Options are: *System Catalog:* Select to specify an image from the PeopleTools system image catalog. *External Source:* Select to specify an image from a source external to the PeopleTools system image catalog.
Image Name	If you select the *System Catalog* image source, click the **Search** button to access a list of existing images.
Image URL	If you select the *External Source* image source, enter a URL for the desired image file. All image types are supported. Ensure that the location of the image provides reliable access.
Alignment	Select the alignment of the image. Options are: *Left:* Select to align the image with the left margin of the pagelet. *Right:* Select to align the image with the right margin of the pagelet.
Height	Enter the image height in pixels if the image needs to be resized. If you enter an image height, the system adjusts the image width to fit the image height.
Width	Enter the image width in pixels if the image needs to be resized. If you enter an image width, the system adjusts the image height to fit the image width.
Float Over Text	Enter text that you want to appear when the user moves the cursor over the image.

Defining Footers

Access the Define Footer page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Footers** > **Define Footer**.)

This example illustrates the fields and controls on the Define Footers - Define Footer page. You can find definitions for the fields and controls later on this page.



Pagelet Footer

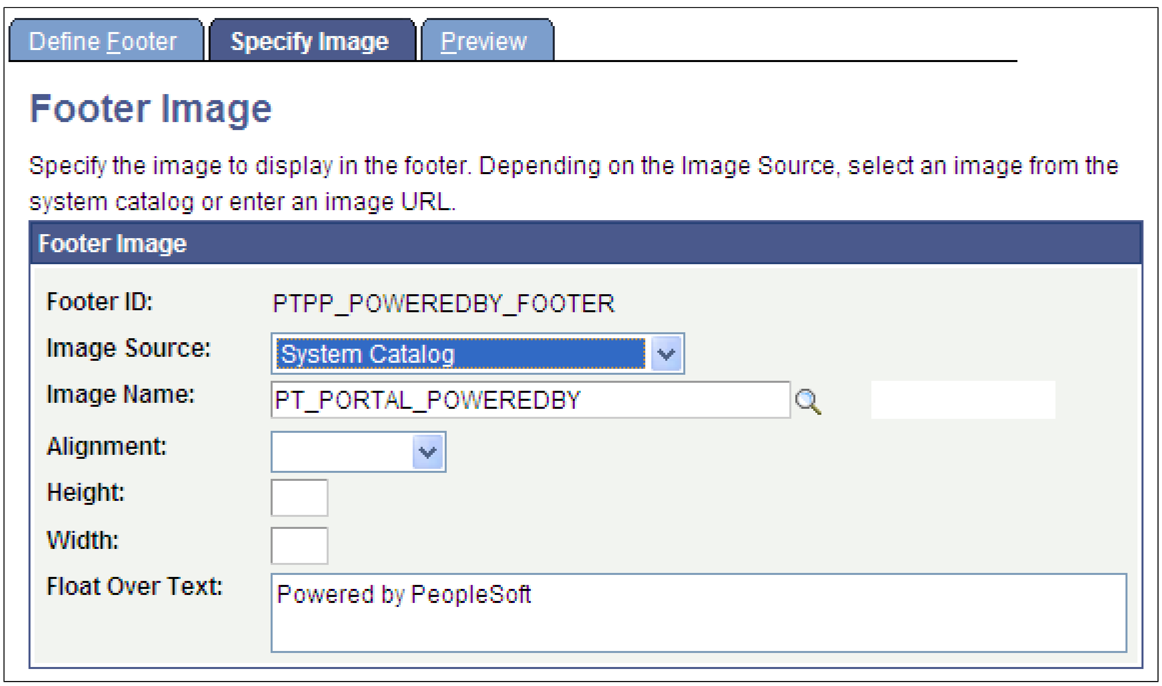
Field or Control	***Description***
Footer ID	Enter an ID for the pagelet footer.
Description	Enter a description of the pagelet footer. The text that you enter here appears as the selectable value in the **Footer** field when users assign footers to pagelets on the Specify Display Options page.
Active	Select to activate the pagelet footer and make it a selectable value in the **Footer** field on the Specify Display Options page. If you deactivate a footer that is in use, the footer no longer appears on associated pagelets. In addition, the footer is not available for selection on the Specify Display Options page.
Long Description	For reference purposes only, enter a long description of the pagelet footer.

Field or Control	***Description***
HTML Area	Enter the HTML code that you want to use to generate the pagelet footer. You can enter both HTML text and image references. Alternatively, you can include an image by selecting it on the Specify Image page.
Delete Footer	Click to delete the pagelet footer design. You will be prompted to confirm your deletion.

Defining Footer Images

Access the Specify Image page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Footers** > **Specify Image**.)

This example illustrates the fields and controls on the Define Footers - Specify Image page. You can find definitions for the fields and controls later on this page.



Footer Image

Field or Control	*Description*
Image Source	Select the image source. Options are: *System Catalog:* Select to specify an image from the PeopleTools system image catalog. *External Source:* Select to specify an image from a source external to the PeopleTools system image catalog.
Image Name	If you select *System Catalog* as the image source, click the **Search** button to access a list of existing images.
Image URL	If you select *External Source* as the image source, enter a URL for the desired image file. All image types are supported. Ensure that the location of the image provides reliable access.
Alignment	Select the alignment of the image. Options are: *Left:* Select to align the image with the left margin of the pagelet. *Right:* Select to align the image with the right margin of the pagelet.
Height	Enter the image height in pixels if the image needs to be resized. If you enter an image height, the system adjusts the image width to fit the image height.
Width	Enter the image width in pixels if the image needs to be resized. If you enter an image width, the system adjusts the image height to fit the image width.
Float Over Text	Enter text that you want to appear when the user moves the cursor over the image.

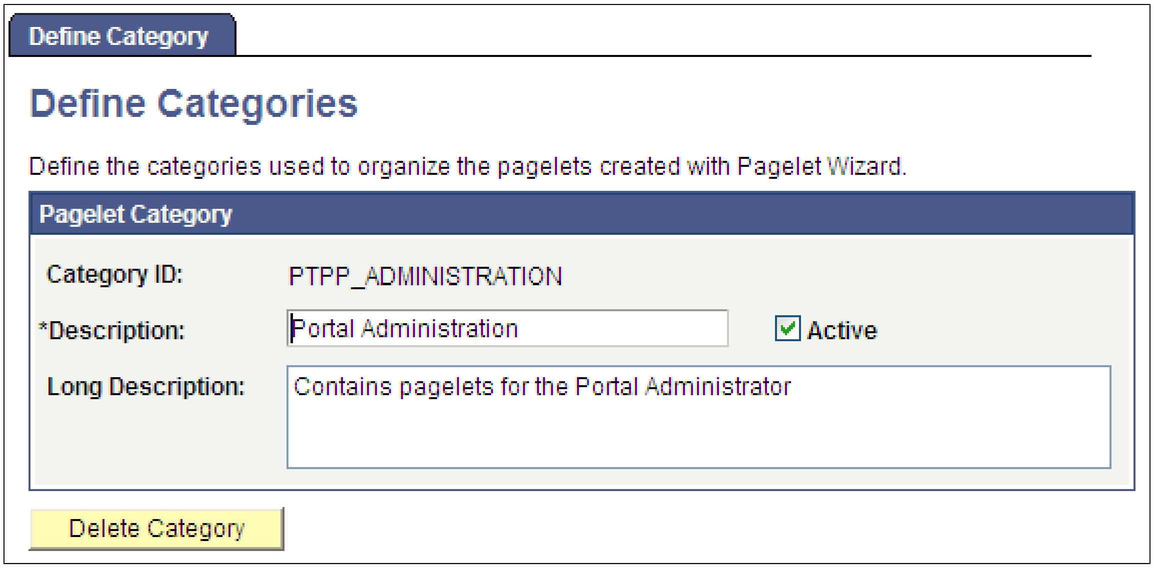
Defining Pagelet Wizard Pagelet Categories

This section discusses how to define Pagelet Wizard pagelet categories.

Defining Pagelet Categories

Access the Define Categories page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Define Pagelet Categories**.)

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



Field or Control	*Description*
Category ID	Enter a Pagelet Wizard pagelet category ID. You can use category IDs to organize Pagelet Wizard pagelets. The portal administrator can use this value when running pagelet reports and when searching for pagelets that need to be moved between different portal sites. Categories provide another level of organization in addition to the owner ID. For example, you can use a departmental owner ID, and within that, categorize your pagelets by pagelet type.
Description	Enter a description of the pagelet category. The text that you enter here appears as a field value in the **Category ID** field when users assign pagelets to categories on the Specify Pagelet Information page.
Active	Select to activate the pagelet category and make it a selectable field value in the **Category ID** field on the Specify Pagelet Information page. If you deactivate a category, the category is no longer available for selection on the Specify Pagelet Information page. Existing pagelets that have been created using the deactivated category are unaffected.
Long Description	For reference purposes only, enter a long description of the pagelet category.

Field or Control	***Description***
Delete Category	Click to delete a pagelet category. You are prompted to confirm the deletion. If you delete a category, the category is no longer available for selection on the Specify Pagelet Information page. Existing pagelets that have been created using the deactivated category still appear, but the pagelet category is removed from the pagelet definition in Pagelet Wizard.

Step 1: Entering Pagelet Identifying Information

This section discusses how to specify pagelet information. This is step 1 in the Pagelet Wizard pagelet creation process.

Note: Before you start building a pagelet with Pagelet Wizard, familiarize yourself with all data sources. This enables you to evaluate the best way to retrieve the data that you want to display in the pagelet.

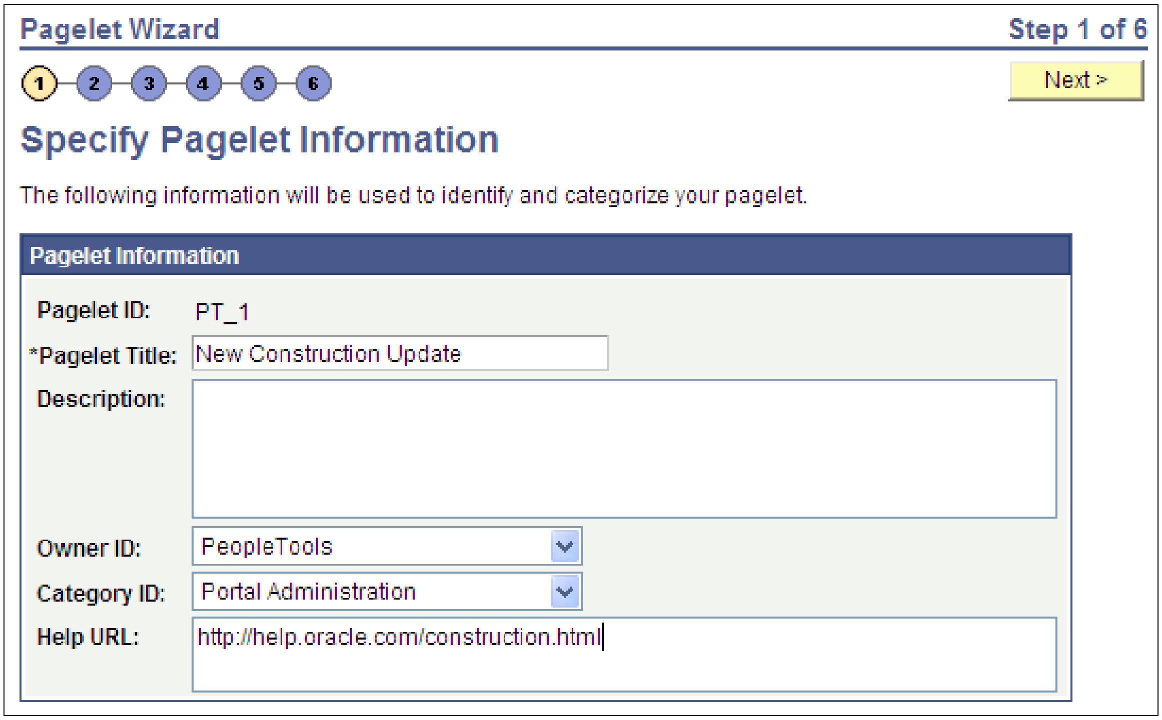
Related Links

Understanding Pagelet Wizard Data Sources

Specifying Pagelet Information

Access the Specify Pagelet Information page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard**.)

This example illustrates the fields and controls on the Specify Pagelet Information page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Pagelet ID	Enter an ID for the pagelet. Each pagelet that you build with Pagelet Wizard must have a unique ID. **Note:** Pagelet IDs can contain only the characters A–Z, a–z, and _. All other characters put into a pagelet ID will be removed by the system. If all characters are removed from the user-entered pagelet ID, then the Registry Object Prefix will be used instead. If after you remove illegal characters from the pagelet ID the name is found to match an existing pagelet ID, then Pagelet Wizard appends numbers to the end of the pagelet ID to make the new pagelet ID unique.
Pagelet Title	Enter a unique title for the pagelet. This value appears in the pagelet title bar, which is the solid bar at the top of the pagelet. If you change the pagelet title value for a published pagelet, you must unpublish and republish the pagelet for the change to take effect.
Description	Enter additional descriptive information to identify the pagelet.

Field or Control	*Description*
Owner ID	Select the value that you want to use to identify pagelets created by the organization. The portal administrator can use this value when searching for pagelets through PeopleSoft Application Designer or SQL queries on the portal registry table. Do not select a delivered PeopleSoft owner ID. You can define unique owner IDs for the organization by entering field translate values for the OBJECTOWNERID field using PeopleSoft Application Designer.
Category ID	To organize Pagelet Wizard pagelets, select a category assignment for the pagelet. You create pagelet categories on the Define Categories page. The portal administrator can use this value when running pagelet reports and when searching for pagelets that need to be moved between different portal sites. Categories provide another level of organization in addition to the owner ID. For example, you can use a departmental owner ID, and within that, categorize pagelets for that ID by pagelet type. See Defining Pagelet Wizard Pagelet Categories.
Help URL	Enter the URL of the help document to associate with the pagelet. Entering a URL causes the **Help** button to appear in the pagelet header bar. When the pagelet is published as a homepage pagelet, a PTPPB_PAGELET_HELP_LINK content reference attribute is added to the pagelet content reference definition corresponding to the pagelet. The value of this content reference attribute will be the help URL entered by the user. **Note:** The **Help** button appears only when the pagelet is on the homepage inside the PeopleSoft Portal; it does not appear when the pagelet is displayed as a WSRP portlet. **Note:** The pagelet help URL takes precedence over the help ID, when both attributes are set for a pagelet.

Step 2: Selecting a Pagelet Data Source

This section discusses selecting a pagelet data source. This is step 2 in the Pagelet Wizard pagelet creation process. Specifically, this section discusses how to :

- Select an activity guide data source.
- Select a free text data source.
- Select an HTML data source.

- Select an IB connector data source.
- Select an Integration Broker data source.
- Select a navigation collection data source.
- Select an OBIEE report data source.
- Select a PeopleSoft Query data source.
- Select a pivot grid data source.
- Select a rowset data source.
- Select a search record data source.
- Select a URL data source.

You can define other data sources by using the Define Data Types page.

Related Links

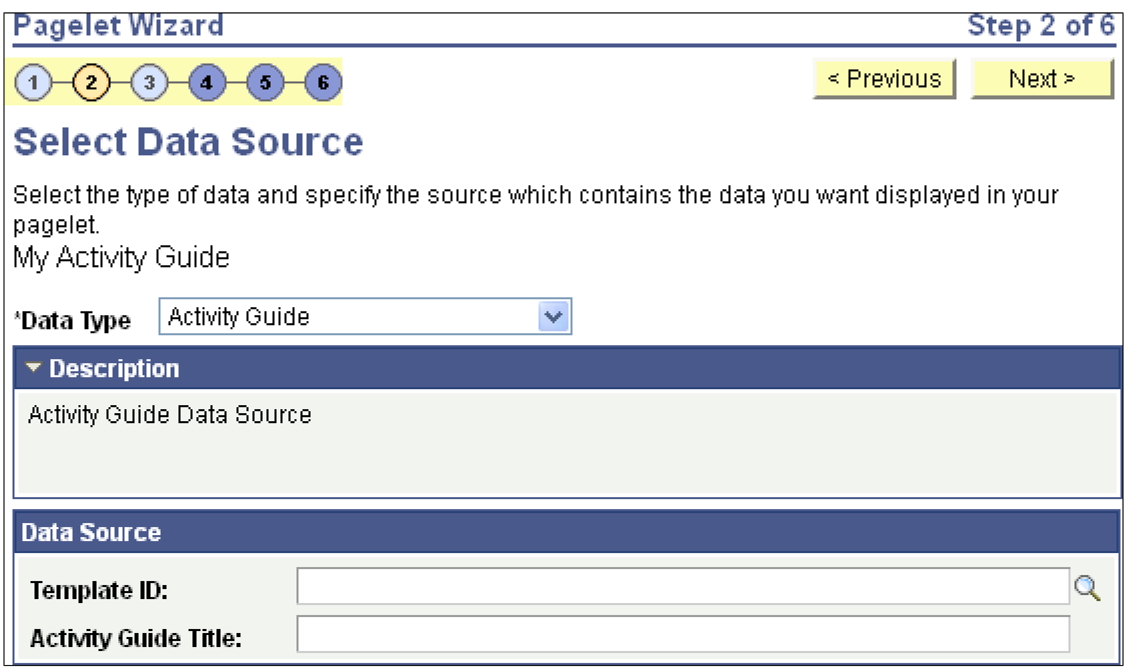
Understanding Pagelet Wizard Data Sources
Defining Data Types

Selecting an Activity Guide Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *Activity Guide* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for an Activity Guide data source. You can find definitions for the fields and controls later on this page.



Depending on whether you want to tie the activity guide pagelet to a specific activity guide instance or to dynamically create the activity guide instance, there are three approaches for generating the activity guide pagelet:

- Provide the template ID, but not the instance ID, which allows the instance to be obtained or generated at runtime. At runtime, the required context data is passed as *query string parameters* in the WorkCenter URL as name-value pairs. If an activity guide instance exists for this context data, then it is retrieved. Otherwise, a new activity guide instance will be created dynamically.

  You can use the Publish as Pagelet task in the Setup Activity Guide WorkCenter to generate an activity guide pagelet of this type, which is typically all that is required in most circumstances. Alternatively, you can use Pagelet Wizard to generate this type of activity guide pagelet.

- Provide both the template ID and the activity guide instance ID, which creates a pagelet tied to an existing activity guide instance.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. The template ID is specified as a data source identifier in step 2 of Pagelet Wizard; the instance ID is specified as a data source parameter in step 3.

- Do not provide either the template ID or the activity guide instance ID, which generates an empty, generic activity guide pagelet that can be used to generate any instance from any activity guide template at run time.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. At run time, the template ID is passed as a data source parameter to the pagelet and the context data is provided as query string parameters in the WorkCenter URL as name-value pairs. This will create the activity guide instance dynamically.

Data Source

Field or Control	***Description***
Template ID	Select a specific activity guide template or leave this field blank to generate a generic activity guide pagelet.
Activity Guide Title	Specify the title for the activity guide pagelet.

Related Links

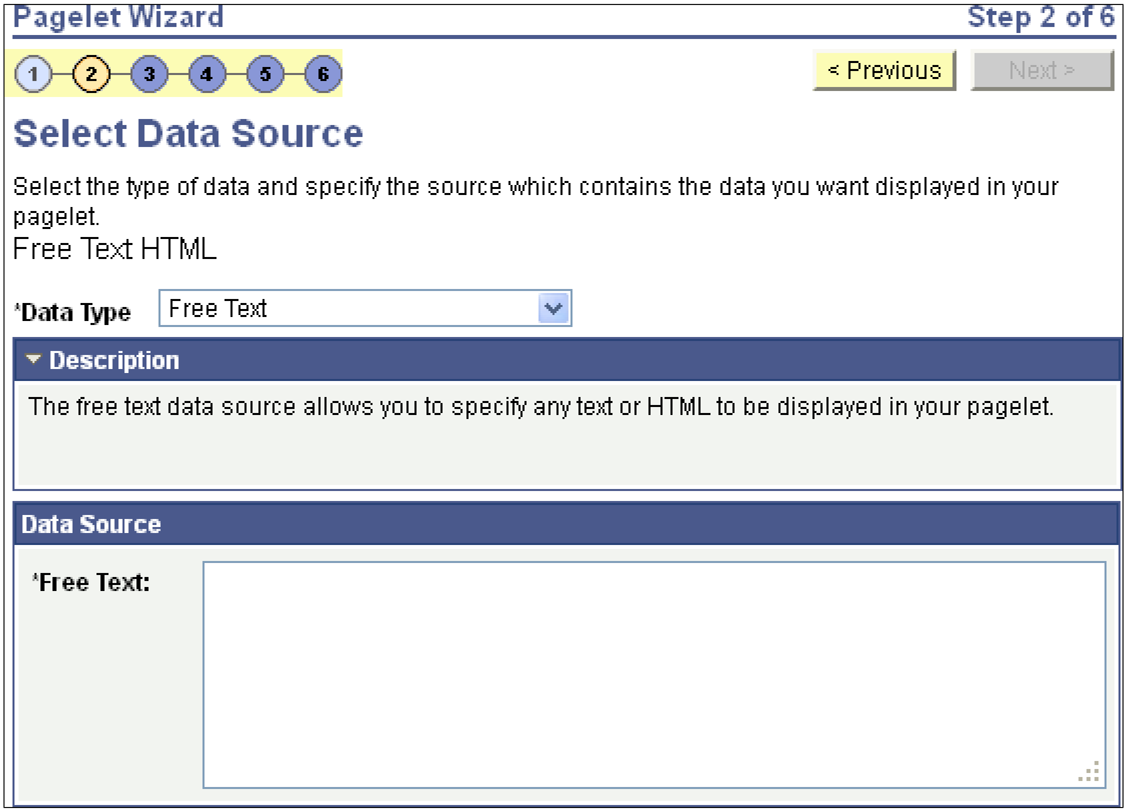
Deploying Classic-Only Activity Guides

Selecting a Free Text Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *Free Text* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for a Free Text data source. You can find definitions for the fields and controls later on this page.



Use the free text data source to present complex formatted HTML in a pagelet. Select this data type instead of the HTML data type when you want to control some aspects of the HTML code.

Data Source

Field or Control	***Description***
Free Text	Use the free text data type to present free-form HTML or simple text in a tile. You can use the following HTML tags only: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , <p>, <b>, <i>, and <img>. Within the <img> tag, src, width, height, and alt are allowable attributes. Use the %Image meta-HTML element to identify an image to include within the HTML: <img src="%Image(*PS_IMAGE_ID*)"> **Note:** To activate the Next button, tab out of the Free Text field.

When you use free text as the data source for a pagelet, there are no configurable data source parameters. Consequently, no end-user personalization exists for free text data source pagelets. When creating an free text data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

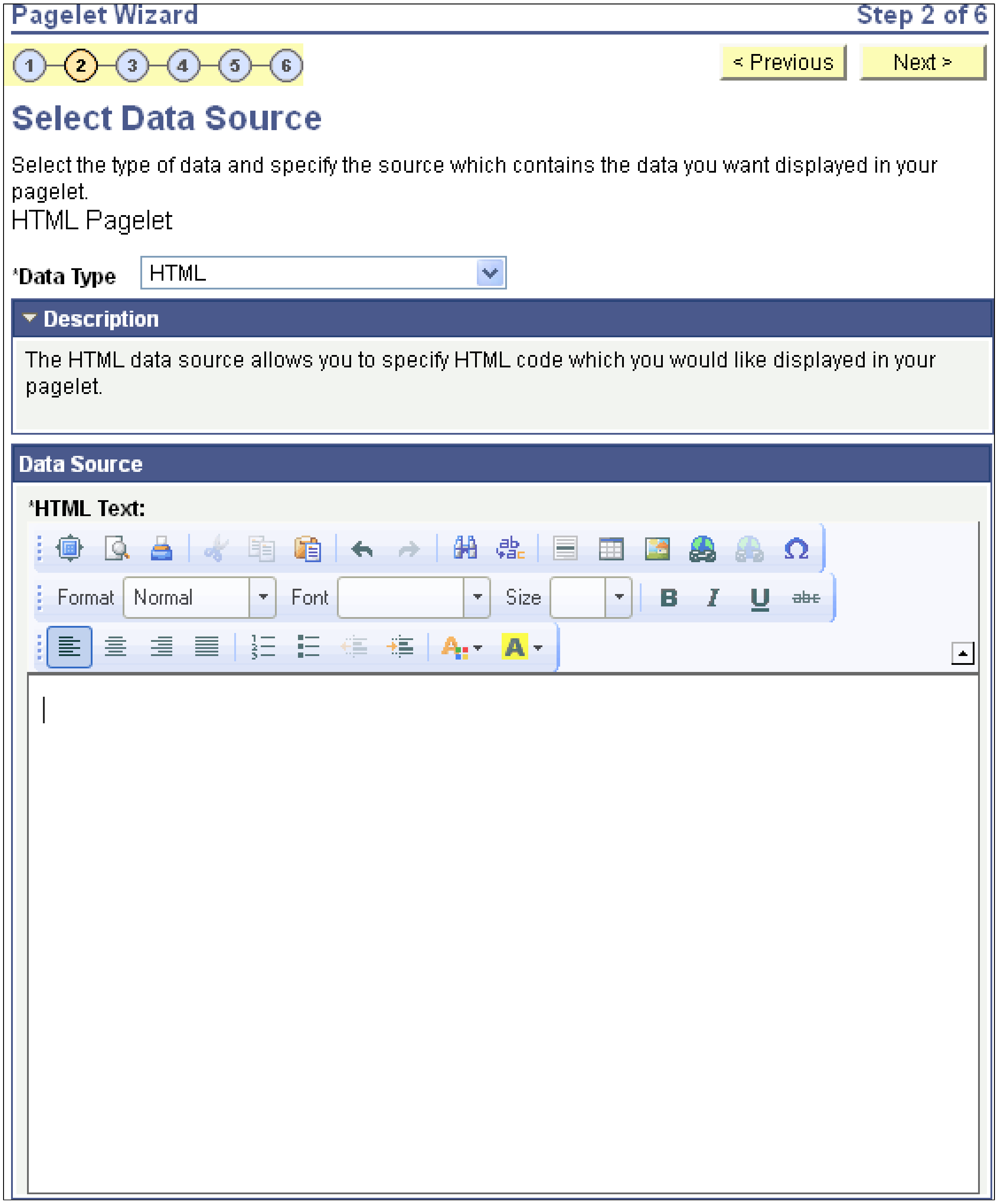
Selecting an HTML Data Source

Note: The Secure PW Administrator role is required to be able to create or update pagelet definitions that use an HTML data source.

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *HTML* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for an HTML data source. You can find definitions for the fields and controls later on this page.



Use the HTML data source to present simple, formatted HTML in a pagelet.

Data Source

Field or Control	***Description***
HTML Text	Enter the HTML code that you want to use as the data source for the pagelet. You can enter an HTML document or fragment. However, complex HTML code that specifies custom CSS, JavaScript, or embedded media might not appear correctly in a pagelet after the rich text editor post-processes the entered data. Ensure that the HTML that you enter generates content that is appropriately sized for pagelet display. **Note:** Source code mode is disabled in the rich text editor on this field. This is for security reasons because malicious code could be inserted in source code mode. **Note:** To activate the Next button, click the Save button.

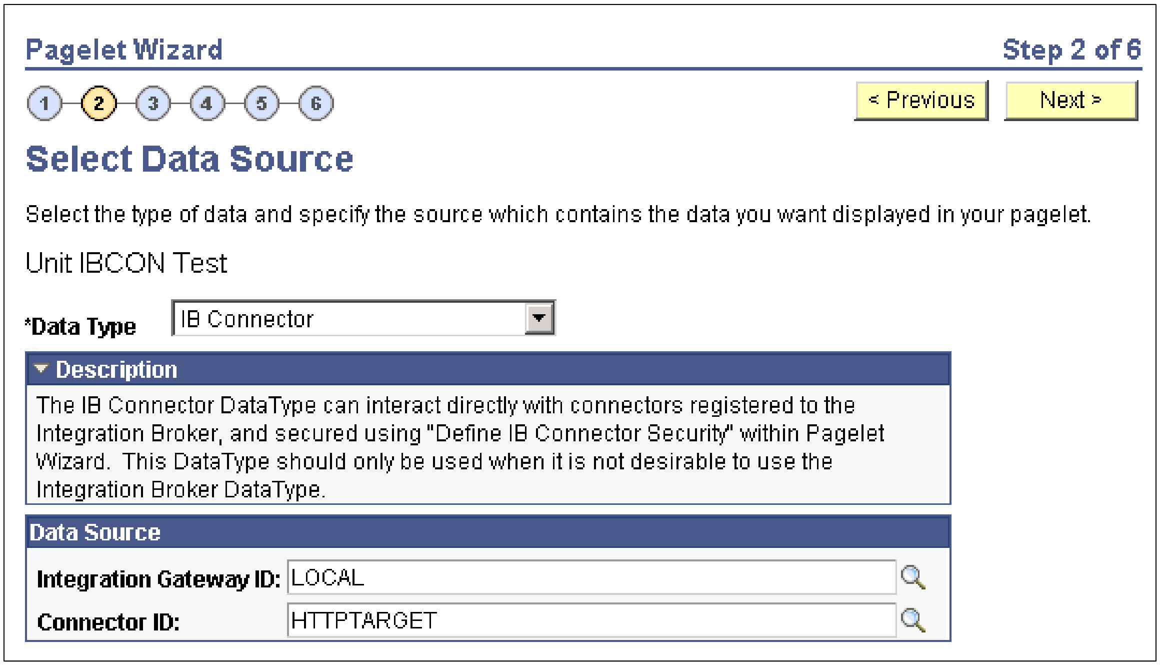
When you use HTML as the data source for a pagelet, there are no configurable data source parameters. Consequently, no end-user personalization exists for HTML data source pagelets. When creating an HTML data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

Selecting an IB Connector Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.)** On the Specify Pagelet Information page, click the Next button

Select *IB Connector* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for an IB Connector data source. You can find definitions for the fields and controls later on this page.



Data Source

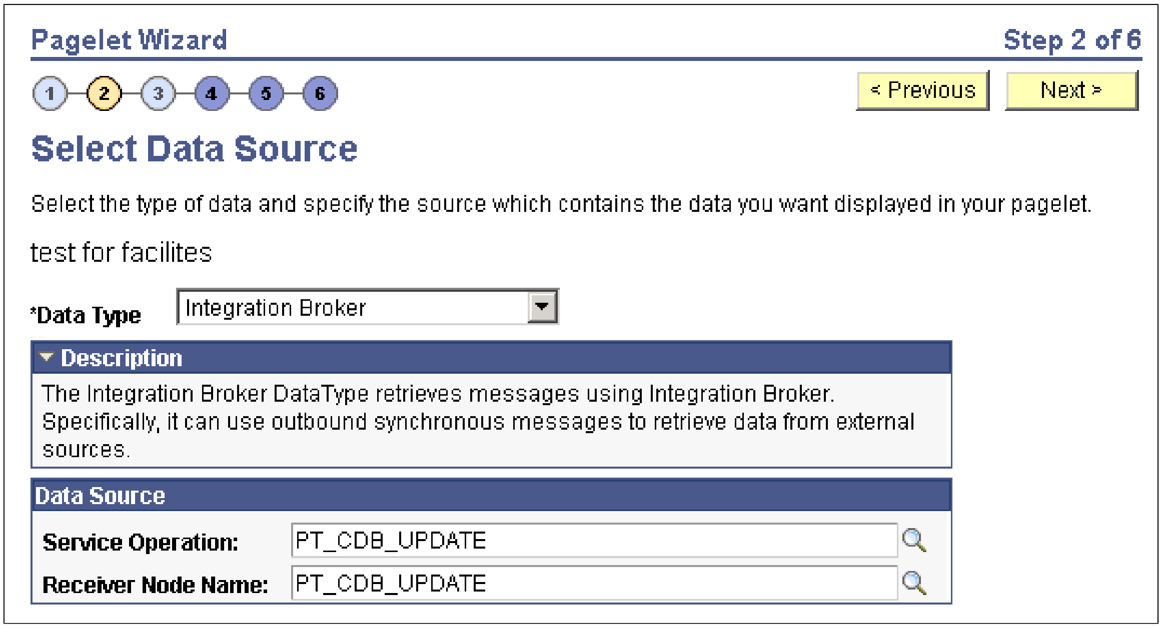
Field or Control	Description
Integration Gateway ID	Select the Integration Broker gateway that contains the connector that you want to use to supply content to the pagelet.
Connector ID	Select the connector that you want to use to supply content to the pagelet.

Selecting an Integration Broker Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *Integration Broker* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for an Integration Broker data source. You can find definitions for the fields and controls later on this page.



You can use an outbound synchronous message defined in the local database as a data source for a Pagelet Wizard pagelet. Outbound, synchronous messages are messages for which a request is made to a target system and a response is expected in real time.

The message can be either structured or unstructured. If the message is structured, then a record structure has been associated with the outbound message and that record structure is translated into data source parameters during step 3. If the message is unstructured, then no record structure has been associated with the outbound message, so no data source parameters can be supplied and Pagelet Wizard bypasses step 3.

Typically, because outbound synchronous messages are designed to run in real time, they are associated with transactions with good performance records. As with all pagelet data sources, you should understand that pagelet performance and homepage performance are tied to the performance of the underlying Integration Broker message and target transaction. You should review the performance of the target transaction before creating an Integration Broker message-based pagelet.

Data Source

Field or Control	*Description*
Service Operation	Select the requesting service operation that you want to be the data source for the pagelet.
Receiver Node Name	Select the node that is associated with the selected requesting service operation.

Data Source Details

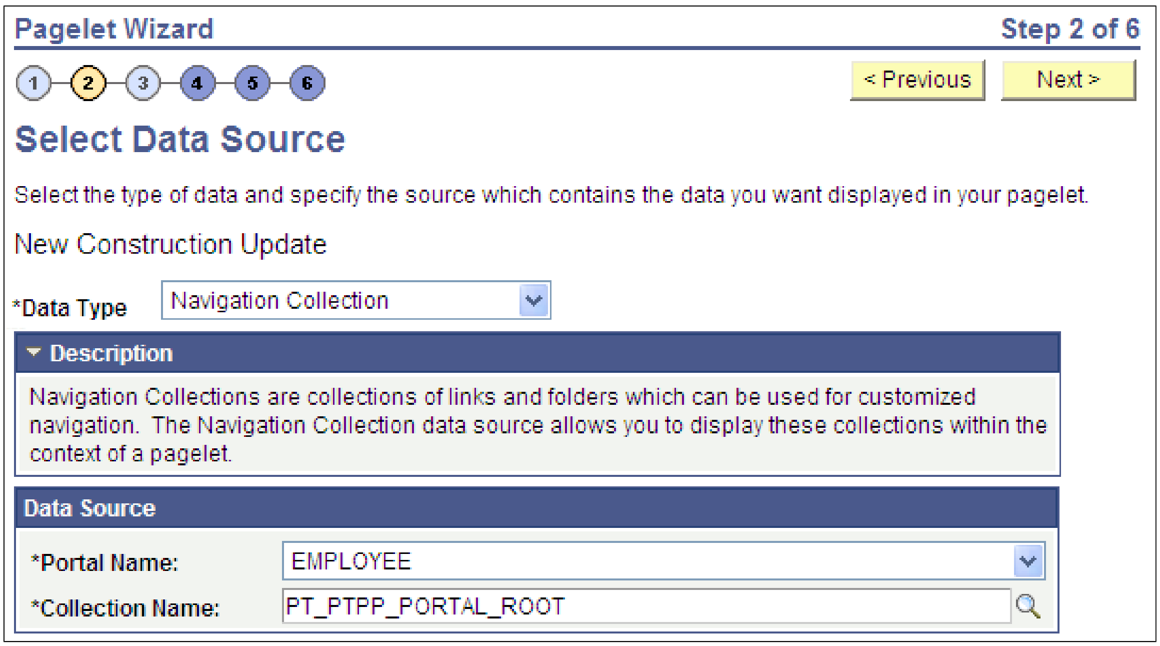
If the selected message is structured, the **Data Source Details** group box displays input and output records and fields that are defined in the message structure.

Selecting a Navigation Collection Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *Navigation Collection* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for a navigation collection data source. You can find definitions for the fields and controls later on this page.



The navigation collection that you use as the data source must already have been created using the navigation collections pages. However, the navigation collection does not need to exist in its final form. Any changes that you make to the underlying navigation collection used as the data source are immediately reflected on any pagelets created from that navigation collection.

Note: Although you can create a navigation collection pagelet directly from the navigation collections component, pagelets created using this component cannot be maintained in Pagelet Wizard. Only navigation collection pagelets created specifically with Pagelet Wizard can be maintained in Pagelet Wizard.

See Understanding Navigation Collections.

Data Source

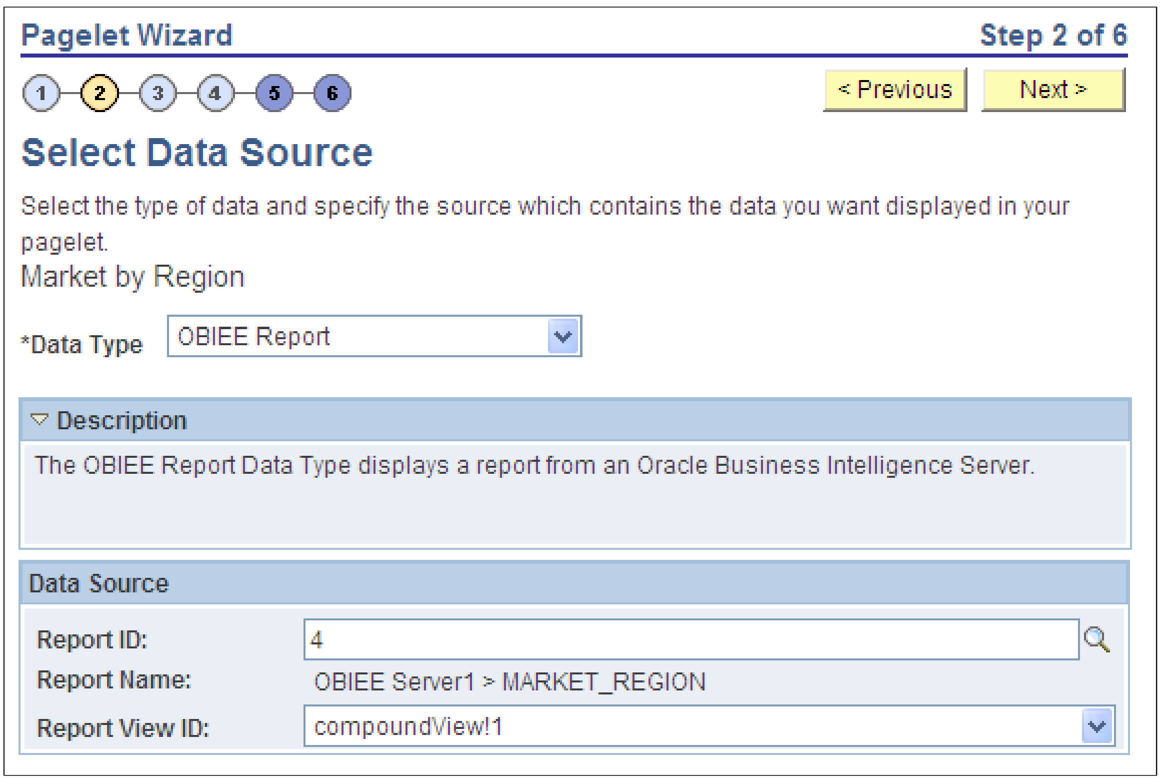
Field or Control	***Description***
Portal Name	Select the portal that contains the navigation collection for the pagelet.
Collection Name	Select the name of the navigation collection that you want to use to create the pagelet.

Selecting an OBIEE Report Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *OBIEE Report* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for an OBIEE report data source. You can find definitions for the fields and controls later on this page.



Data Source

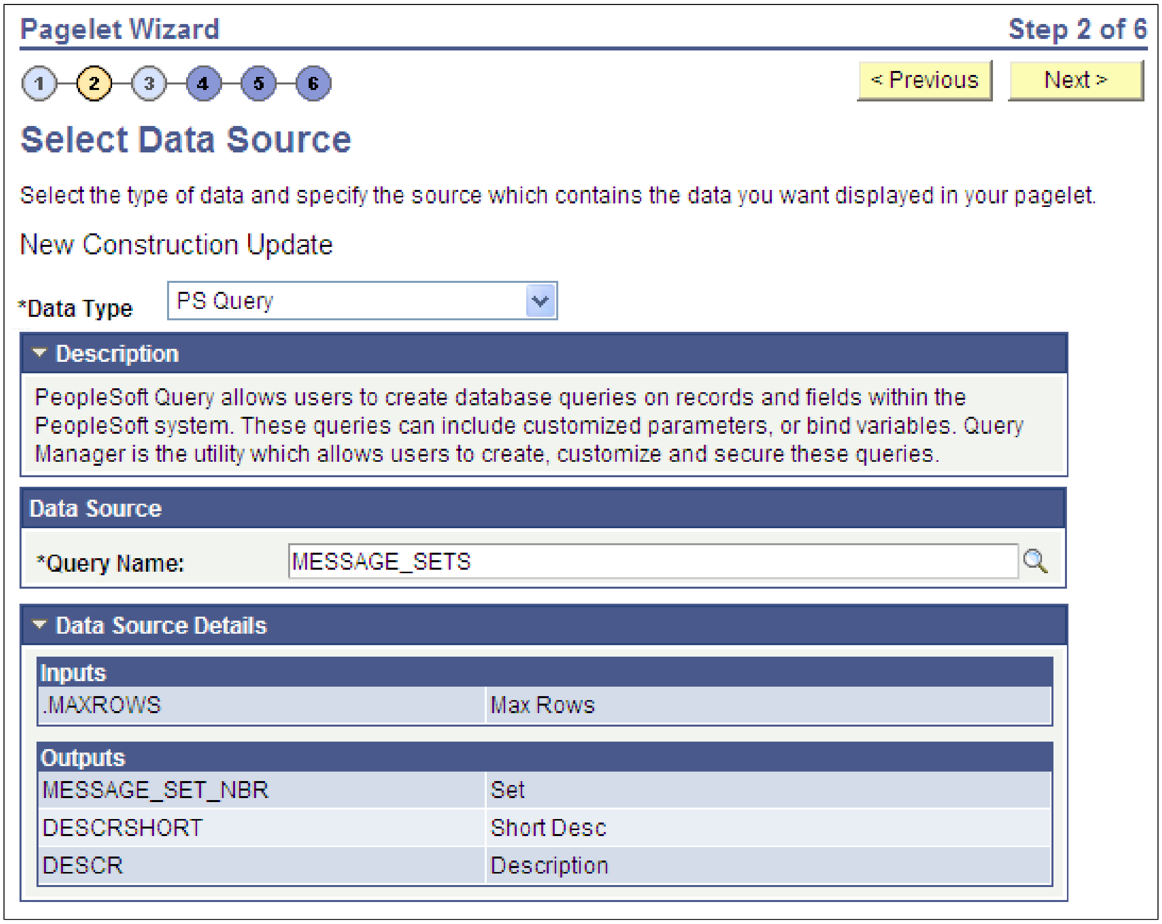
Field or Control	***Description***
Report ID	Select the report that you want to be the data source. Only reports that have been loaded into the PeopleSoft application appear in this list. See Loading OBIEE Report Definitions.
Report Name	This field displays the OBIEE server name and the report name. This field is display-only.
Report View ID	Select the view or report type as created in the OBIEE application.

Selecting a PeopleSoft Query Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *PS Query* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for a PeopleSoft query data source. You can find definitions for the fields and controls later on this page.



Queries typically return a number of columns and rows. Queries for pagelets should be selected based on available display space on the homepage. In addition, avoid selecting long-running queries that can delay the display of the homepage.

Note that public query results are available to portal viewers only if they have appropriate access to the underlying data record. Permission lists in the users' role definitions must contain query access groups that include the records in the query definition.

Note: If data masking for personally identifiable information or other sensitive data has been enabled on a record field that is part of the criteria for the PeopleSoft query, then these data values are masked when displayed via Pagelet Wizard pagelets.

Data Source

Field or Control	***Description***
Query Name	Select a query to serve as the data source for the pagelet. You can select any public query in the local database. PeopleSoft queries provide SQL access to any table recognized by the PeopleSoft system. This includes external, remote tables that are recognized by the PeopleSoft system by way of a database link. See "Creating New Queries" (Query).

Note: Even though a query is marked as public, PeopleSoft Query still applies security to the underlying records. Ensure that the users of this pagelet have query security access to the underlying records to avoid runtime errors.

Data Source Details

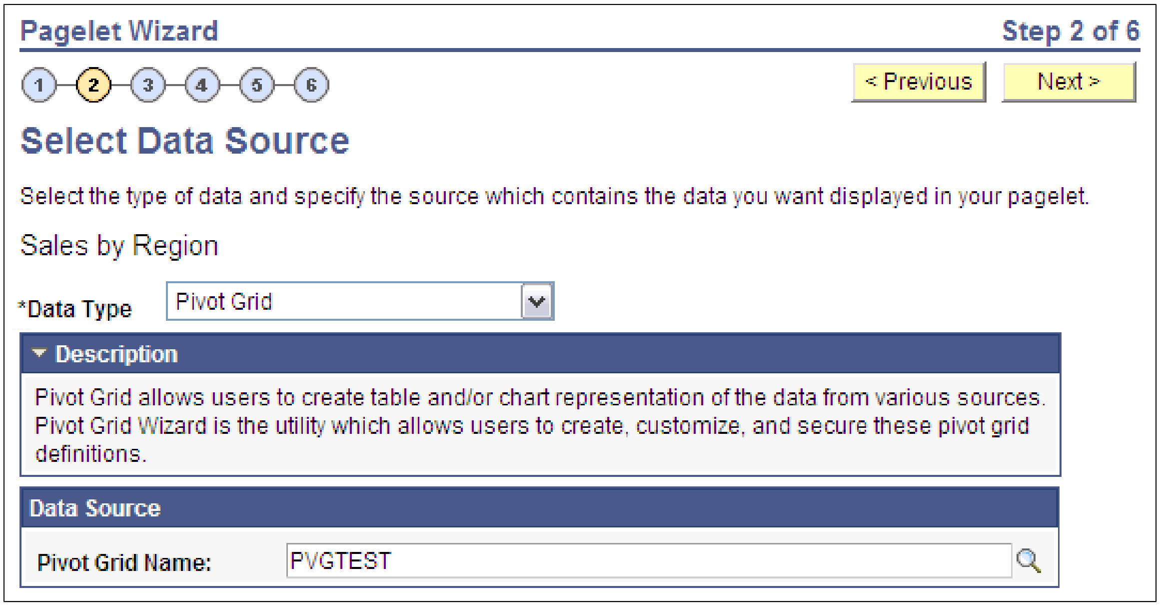
Field or Control	***Description***
Inputs	Displays the data source parameter fields that correspond to the query's input prompts. This field also displays the required system-supplied .MAXROWS query data source parameter that you can use to limit the amount of data returned to the pagelet.
Outputs	Displays the data source parameter fields that correspond to the query's output.

Selecting a Pivot Grid Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *Pivot Grid* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for a pivot grid data source. You can find definitions for the fields and controls later on this page.



Pivot Grid is a PeopleSoft component that enables you to display application data in interactive pivot tables and charts so that you can pivot, slice, and dice your application data across multiple dimensions. Pivot grids are ideal data sources for pagelets that you use on dashboards and homepages as a form of operational reporting.

Data Source

Field or Control	***Description***
Pivot Grid Name	Select the pivot grid that you want the pagelet to display.

Related Links

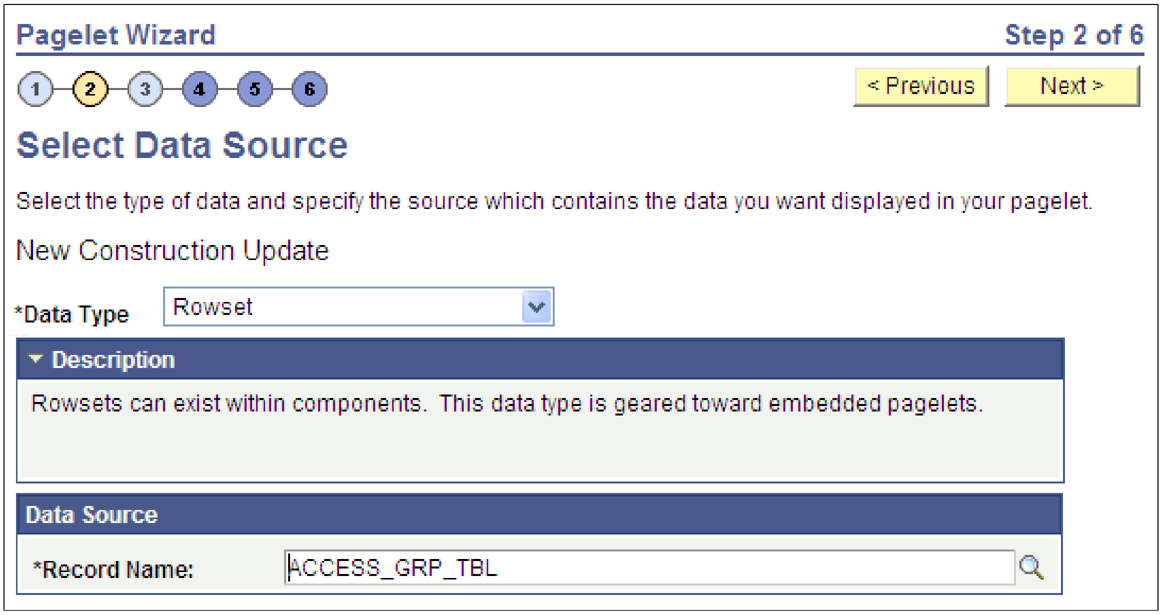
“PeopleSoft Pivot Grid Implementation” (Pivot Grid)

Selecting a Rowset Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *Rowset* in the **Data Type** field.

The followings example illustrates the fields and controls on the Select Data Source page for a Rowset data source.



A rowset is a record that collects rows of data and can contain data from one or more records. The rowset data source is geared towards supplying data to embedded pagelets. This is because embedded pagelets require that a populated rowset be passed into the data source for it to be able to display the data as a chart or table, for example.

Embedded pagelets do not allow for customizable parameters that enable criteria selection. Consequently also, no end-user personalization for rowset data source pagelets.

When creating a rowset data source pagelet, you skip step 3 and proceed to step 4, in which you select your pagelet display format.

Data Source

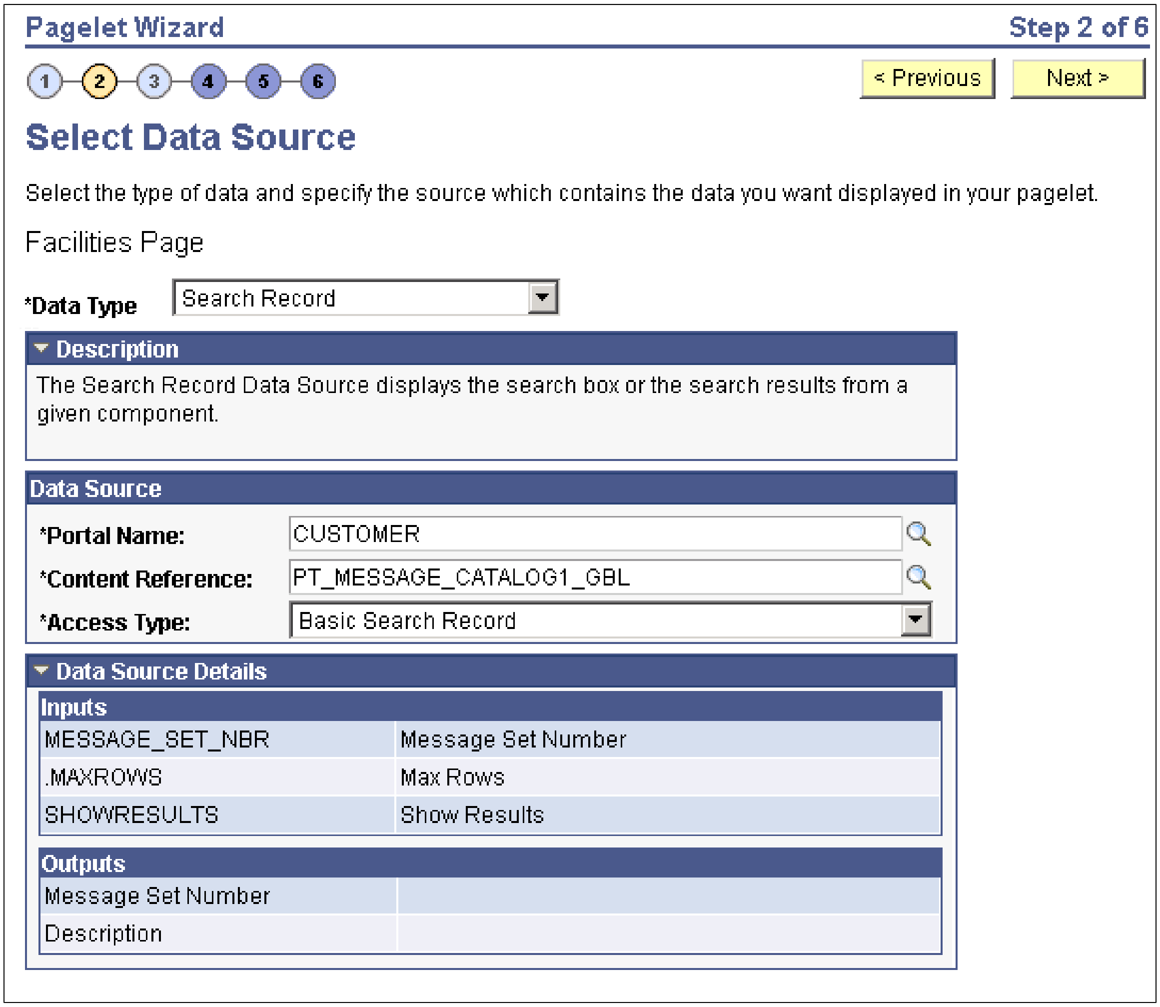
Field or Control	***Description***
Record Name	Select the record that contains the layout of the rowset that you want to use to supply your pagelet with data. Records available for selection are records that are tables or views.

Selecting a Search Record Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *Search Record* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for a search record data source. You can find definitions for the fields and controls later on this page.



Search records typically return a number of columns and rows. When you select search records to be displayed on pagelets, consider the available display space on the homepage.

Data Source

Field or Control	*Description*
Portal Name	Select the portal in which the search record for the pagelet resides.
Content Reference	Select the content reference for the search record.

Field or Control	Description
Access Type	Select the type of access for the search record data. Options are: *Basic Search Record:* Select to have the search record pagelet retrieve data based on the basic search record defined in the selected portal. *CI: <component interface name>:* Select when the basic search record does not provide the required functionality, for example, when security or PeopleCode considerations need to be taken into account. This option is available if the selected search record has an associated component interface. Select to have the search record pagelet retrieve data by way of the component interface. See "Creating Component Interface Definitions" (Component Interfaces)"Implementing PeopleSoft Component Interfaces" (Component Interfaces).

Data Source Details

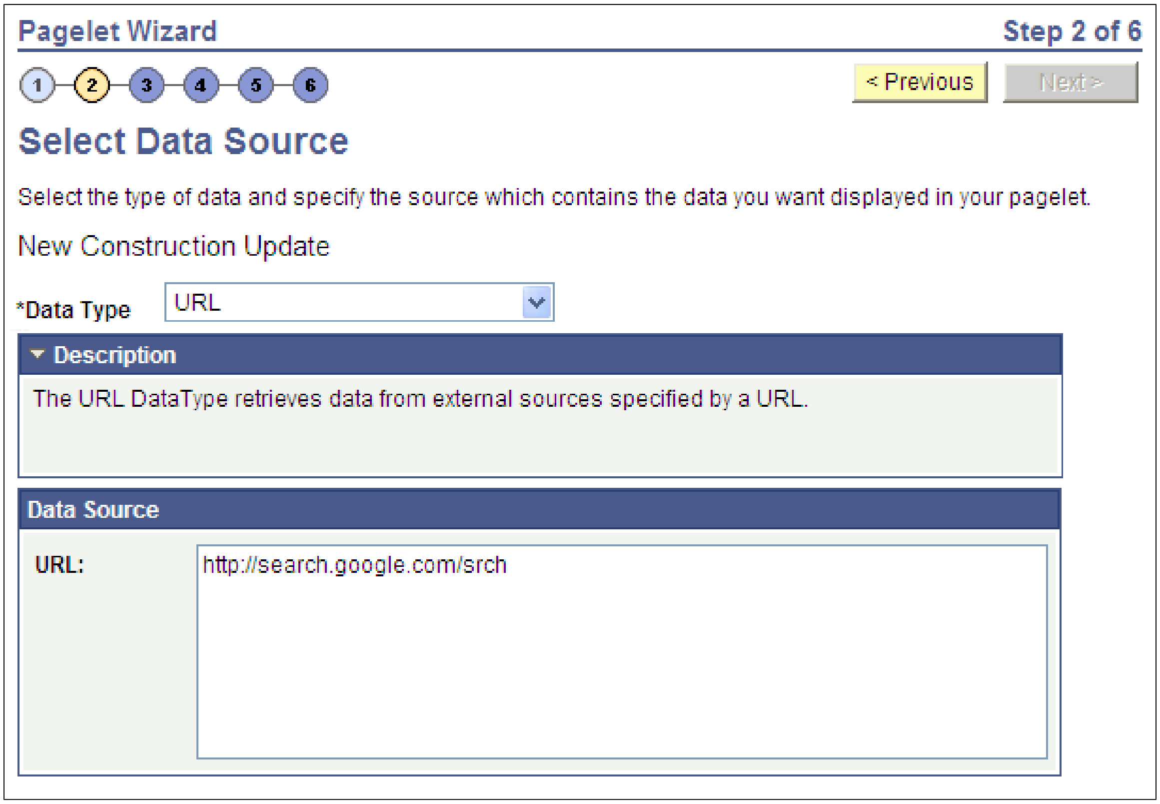
Field or Control	Description
Inputs	Displays search record key input fields, as well as other system-required input fields that are used to derive data that is displayed on the search record pagelet.
Outputs	Displays search record key output fields, as well as other output fields for which search results are displayed on the search record pagelet. These output fields are derived from the drop-down list box fields on the search record. If the SHOWRESULTS data source parameter field for the pagelet is set to *N,* the search key output fields appear as **Search By** drop-down list values on the pagelet.

Selecting a URL Data Source

Access the Select Data Source page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Pagelet Information page, click the Next button.)

Select *URL* in the **Data Type** field.

This example illustrates the fields and controls on the Select Data Source page for a URL data source. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
URL	Enter any HTTP-accessible URL as the data source for the pagelet. The URL may or may not require query string parameters that can be specified and personalized as end-user prompts within Pagelet Wizard. For example, the Yahoo! search page URL takes the form http://search.yahoo.com/search?p=*searchterm*. If you want to make *searchterm* a user-specified data parameter, enter a **URL** field value of *http://search.yahoo.com/search.* In step 3 on the Specify Data Source Parameters page, add a data source parameter **Field Name** value of *p* and set its **Usage Type** field value to *User Specified* with a default value. Pagelet Wizard then adds the *?p=searchterm* value, with the *searchterm* value able to be personalized by the user, to the URL before it retrieves the content to the pagelet.

Step 3: Specifying Pagelet Data Source Parameters

This section provides an overview of system variables supported as data source parameters, lists common elements, and discusses specifying pagelet data source parameters. This is step 3 in the Pagelet Wizard pagelet creation process. Specifically, this section discusses how to:

- Specify activity guide data source parameter values.

- Specify free text data source parameter values.
- Specify HTML data source parameter values.
- Specify IB connector data source parameters.
- Specify Integration Broker data source parameters.
- Specify navigation collection data source parameters.
- Specify OBIEE data source parameters.
- Specify pivot grid data source parameters.
- Specify query data source parameters.
- Specify rowset data source parameters.
- Specify search record data source parameters.
- Specify URL data source parameters.
- Specify search record data source parameters.

Understanding System Variables Supported as Data Source Parameters

When you set the **Usage Type** field to *System Variable* or *User Specified* on the Specify Data Source Parameters page, you can enter a system variable such as *%UserId,* which the system substitutes at runtime with the actual value represented by the variable. The following system variables are supported as data source parameters.

- %AuthenticationToken
- %ClientDate
- %ClientTimeZone
- %ContentID
- %ContentType
- %Copyright
- %Currency
- %Date
- %DateTime
- %DbName
- %DbServerName
- %DbType
- %EmailAddress

- %EmployeeId
- %HPTabName
- %IsMultiLanguageEnabled
- %Language
- %Language_Base
- %LocalNode
- %Market
- %MaxMessageSize
- %NavigatorHomePermissionList
- %Node
- %PasswordExpired
- %Portal
- %PrimaryPermissionList
- %ProcessProfilePermissionList
- %RowSecurityPermissionList
- %RunningInPortal
- %ServerTimeZone
- %Time
- %UserDescription
- %UserId

Common Elements Used in This Section

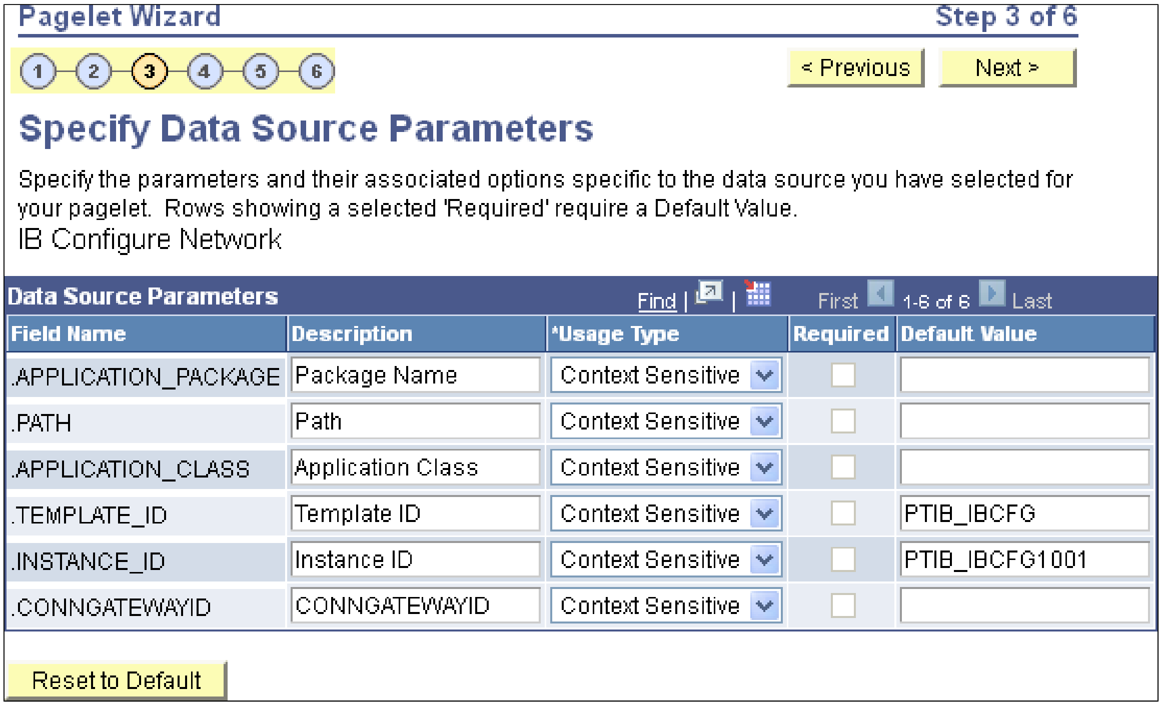
Field or Control	***Description***
Description	Displays a description of the data source parameter.

Field or Control	***Description***
Usage Type	Select the type of end-user accessibility that you want to grant for the Data Source Parameter field when it appears on the pagelet. Options are: • *Not Used:* Select if it will not be used on the pagelet. • *Fixed:* Select to enter a fixed value for the data source parameter that the pagelet end user cannot modify. • *System Variable:* Select to assign a system variable as the data source parameter value. The value of the system variable is automatically inserted into the parameter when the pagelet appears. When you select this option, the pagelet end user cannot modify the data source parameter. For example, suppose that you specify *%UserId* as the system variable for a **User** data source parameter field. When the pagelet appears on a user's homepage, the **User** field is populated by the *%UserId* system variable, which is the user ID used to access the pagelet. • *User Specified:* Select to enable end users to specify a data source parameter value for this field. When a pagelet contains a user-specified field, the **Customize** button appears on the pagelet title bar. End users can click this button to access a personalization page, where they can select a data source parameter value that they want to appear by default on the pagelet when they access it. They can select a value from a prompt, or they can manually enter their own value if no prompt values are available. If you change the usage type from or to *User Specified* for a data source parameter on a published homepage pagelet, you must unpublish and republish the pagelet. • *Admin Specified:* Enables users with administrative privileges to specify variables for this field as well as access the **Configure** link on the published pagelet and select from those parameters for users. • *Context Sensitive:* Enables users to specify context sensitivity for activity guide, OBIEE, and embedded pagelets.

Field or Control	***Description***
Default Value	If you select *User Specified* or *Admin Specified* as the usage type and the data source parameter is a required value, you must enter the default value that should appear before a user enters a value. If the data source parameter is not a required value, you do not have to enter a default value. If you select a default value when defining available user-specified values on the Specify Data Source Parameter Values page, this field is populated with that default value. If you select *User Specified* as the usage type, you can enter a system variable as the default value. For example, to make the current date the default value for a user-specified parameter, enter a default value of *%Date*. If you select *System Variable* as the usage type, you must enter the system variable to use as the data source parameter value. See Understanding System Variables Supported as Data Source Parameters. If you select *Fixed* as the usage type, you must enter the fixed value. You can enter a value that includes the % and * wildcards at the beginning or end of a value in the **Default Value** field.
	If you select *Fixed* as the usage type and the data source parameter has associated prompt values, you can use the **Look up Value** button to search for available data source parameter values. If you select *System Variable* as the usage type, you can use the **Look up Value** button to access a list of valid system variables. If you select *User Specified* as the usage type, this button does not appear.
User Values	If you select *User Specified* as the usage type, click to access the Specify Data Source Parameter Values page. See Step 3: Specifying Pagelet Data Source Parameters.

Specifying Activity Guide Data Source Parameters

This example illustrates the fields and controls on the Specify Data Source Parameters for an activity guide data source *when generating a pagelet for a specific activity guide instance.* You can find definitions for the fields and controls later on this page.



When creating an activity guide pagelet, do one of the following:

- If you are creating an activity guide pagelet for a specific activity guide instance, specify the template ID at step 2 and specify the instance ID on this page.
- If you are creating an activity guide pagelet for an activity guide template with the instance ID to be supplied at run time, specify the template ID at step 2 and skip this page.
- If you are creating a generic activity guide pagelet with both the template ID and instance ID to be supplied at run time, specify no template ID at step 2 and skip this page.

Data Source Parameters

Field or Control	***Description***
.TEMPLATE_ID	Specified in step 2 when specifying the data source.
.INSTANCE_ID	Specify an instance ID *only when you want to generate a pagelet for a specific activity guide instance.*

Related Links

Deploying Classic-Only Activity Guides

Specifying Free Text Data Source Parameters

When creating an free text data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

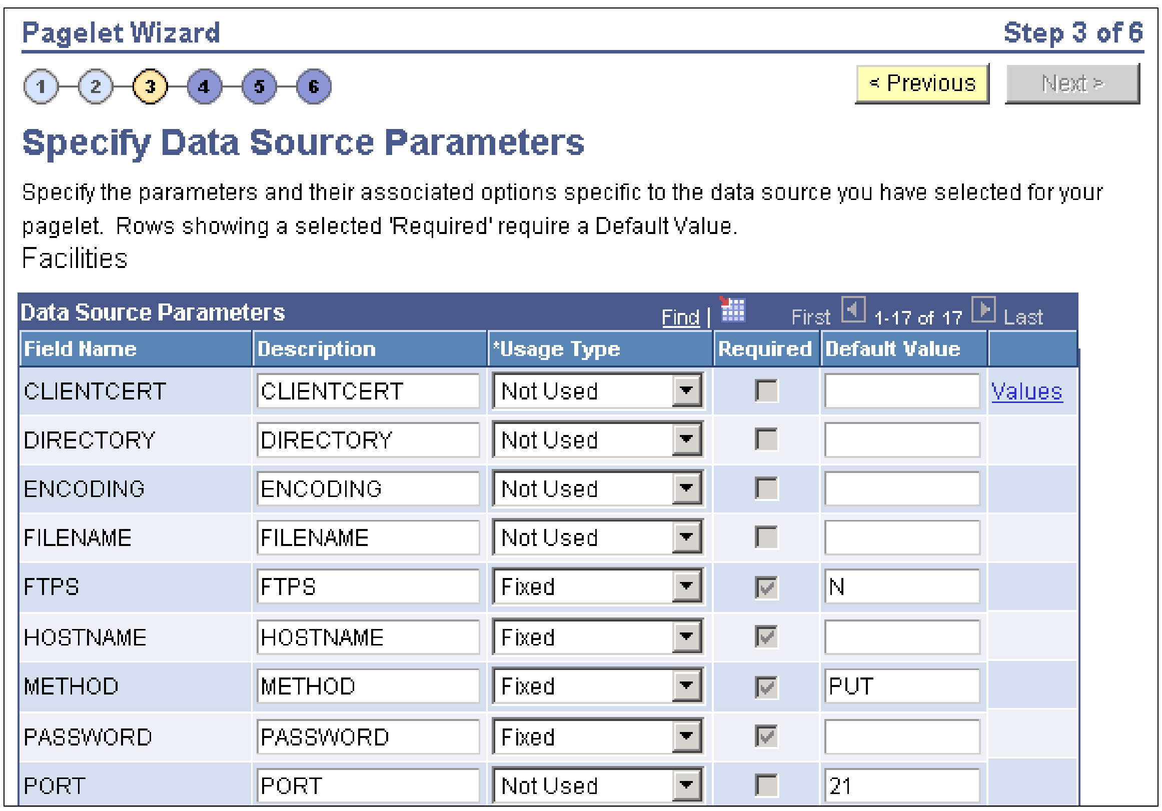
Specifying HTML Data Source Parameters

When creating an HTML data source pagelet, you skip step 3 and proceed to step 4, in which you select the pagelet display format.

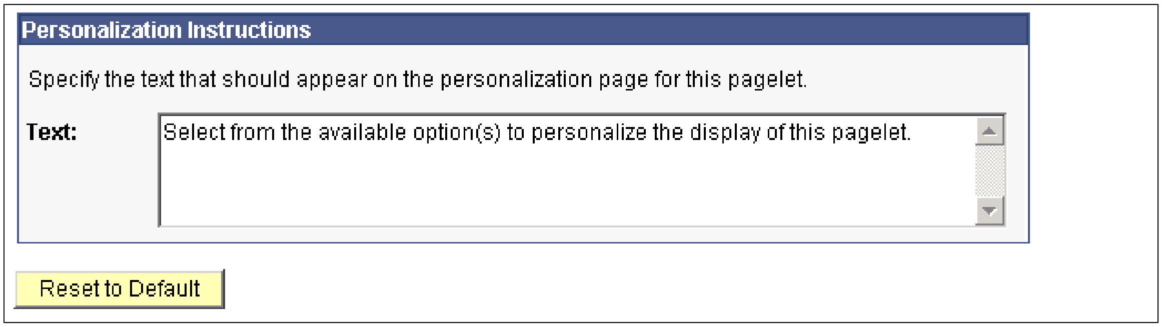
Specifying IB Connector Data Source Parameters

After selecting *IB Connector* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.)

This example illustrates a portion of the fields and controls on the Specify Data Source Parameters for an IB connector data source.



This example illustrates the remainder of the fields and controls on the Specify Data Source Parameters for an IB connector data source.



Define the data source parameters for the IB Connector data source. Click the **User Values** button to set the default values for users.

See Specifying User-Defined Data Source Parameters.

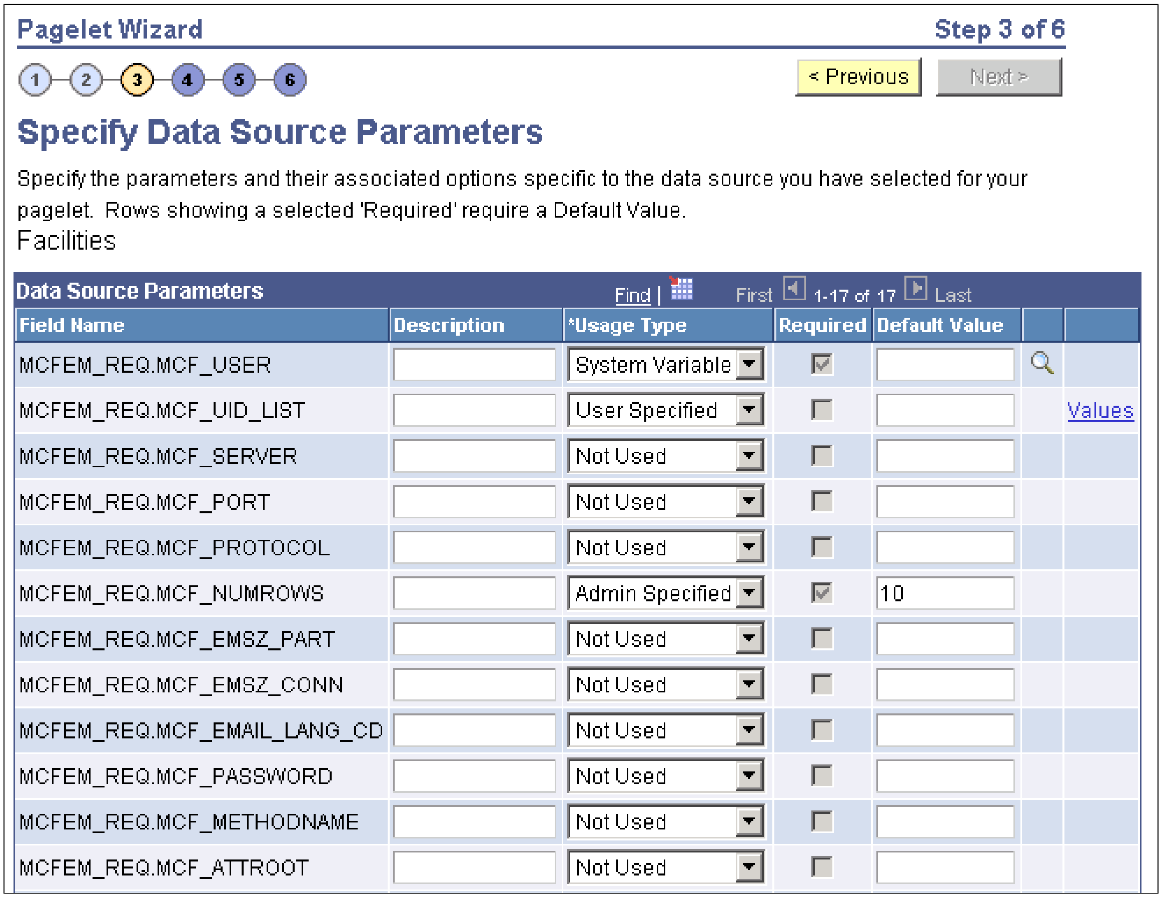
Related Links

“Connectors” (Integration Broker)

Specifying Integration Broker Data Source Parameters

After selecting *Integration Broker* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.).

This example illustrates a portion of the fields and controls on the Specify Data Source Parameters for an Integration Broker data source.



This step is applicable for the Integration Broker data source if the service operation is based on a structured message or if the message is structured but contains a schema. The structure that is associated with the message is translated into data source parameters on this page.

If the message is unstructured, then no schema structure has been associated with the message. Therefore, no data source parameters can be supplied and Pagelet Wizard bypasses this step.

You must supply values for all rows with the **Required** option selected.

Personalization Instructions

Use the **Personalization Instructions** text box . These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

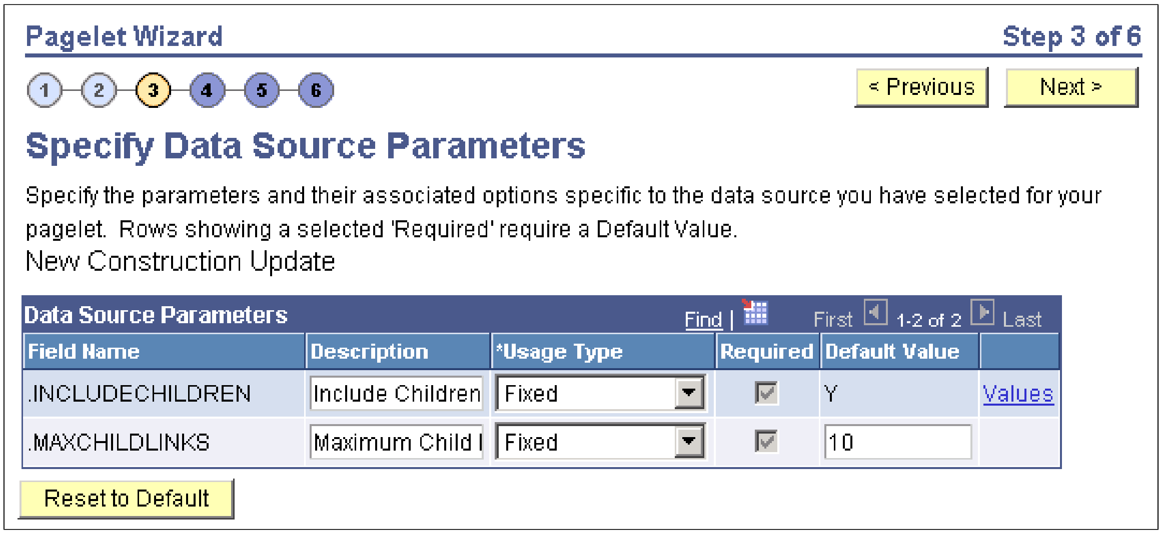
Related Links

"Understanding Managing Service Operations" (Integration Broker)

Specifying Navigation Collection Data Source Parameters

After selecting *Navigation Collection* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.)

This example illustrates the fields and controls on the Specify Data Source Parameters for a navigation collection data source. You can find definitions for the fields and controls later on this page.



Data Source Parameters

Pagelet Wizard analyzes the selected navigation collection and retrieves a list of predefined parameters and any related prompt list values. You cannot add to the list of parameters.

Field or Control	***Description***
Field Name	Displays the following data source parameters required by the navigation collection data source. **.INCLUDECHILDREN:** Determines whether the pagelet displays navigation collection child links. **.MAXCHILDLINKS:** If the pagelet is defined to display child links, determines the maximum number of navigation collection child links to display.
Required	This option is selected if the data source parameter must contain a value on the pagelet to retrieve data successfully. For example, the **.INCLUDECHILDREN** field is required. If this option is selected and is display-only, the value is set in the application class at the API level.

Field or Control	***Description***
Reset to Default	Click to reset the values on this page to the default values specified by the navigation collection data source API.
Values	Click the **Values** link to set the default values for users.

Personalization Instructions

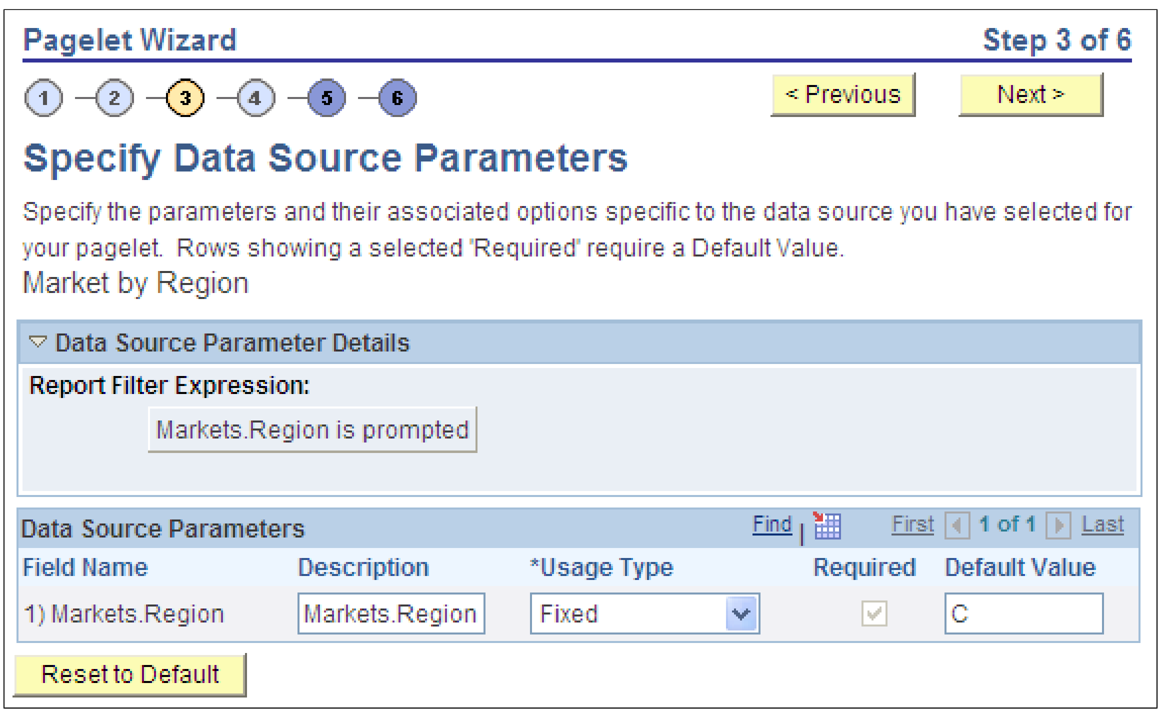
Use the **Personalization Instructions** text box to enter instructional informs for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

Specifying OBIEE Data Source Parameters

After selecting *OBIEE Report* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.)

This example illustrates the fields and controls on the Specify Data Source Parameters page for an OBIEE report data source. You can find definitions for the fields and controls later on this page.



Data Source Parameters

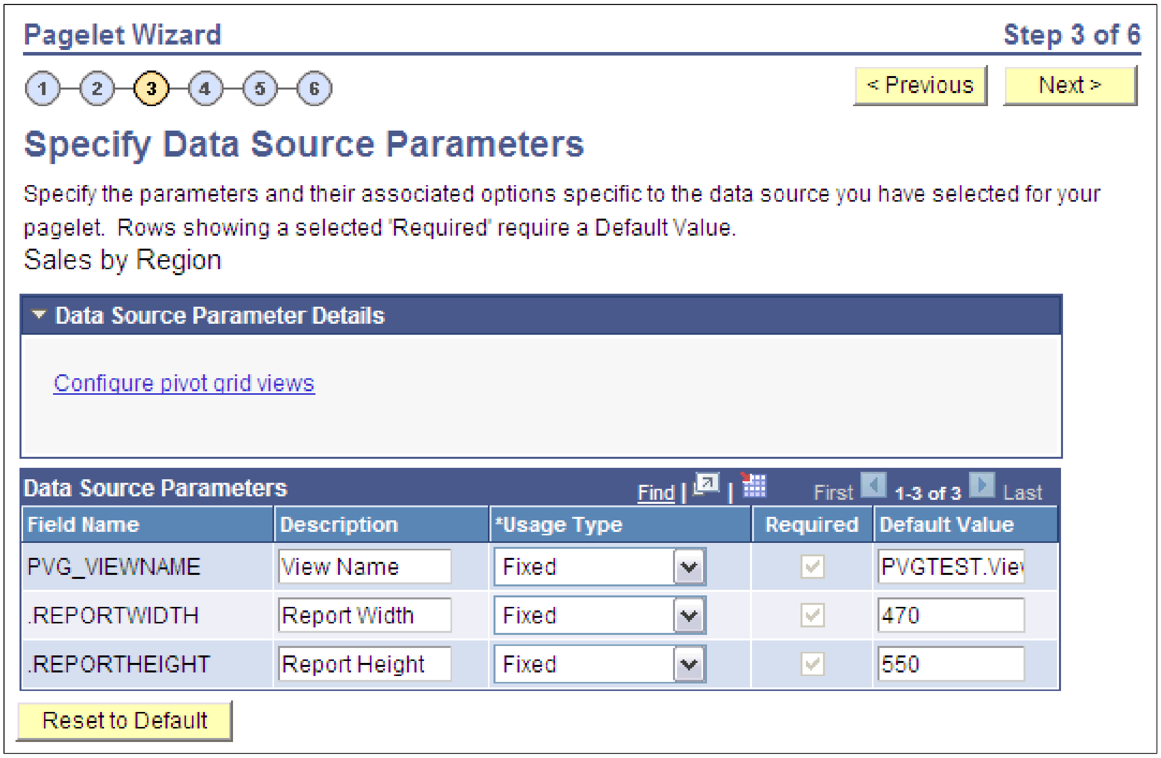
Pagelet Wizard analyzes the selected OBIEE report and retrieves a list of predefined parameters and any related prompt list values. You cannot add to the list of parameters.

Field or Control	***Description***
Field Name	Displays the data source parameters that correspond to the OBIEE prompts associated with the selected OBIEE report. This field also displays the required system-supplied .MAXROWS query data source parameter that you can use to limit the amount of data returned on the pagelet.
Description	This field contains a description that identifies the purpose or location of the OBIEE field.
Required	This option is selected if the data source parameter must have a value entered on the pagelet to retrieve data successfully. For example, the **.MAXROWS** field is required. If this option is selected and is display-only, the value is set in the application class at the application programming interface (API) level. If you leave this option deselected, the data source parameter value can be cleared at runtime, or it can be entered to refine the data selection criteria.
Reset to Default	Click to reset the values on this page to the default values specified by the OBIEE data source API.

Specifying Pivot Grid Data Source Parameters

After selecting *Pivot Grid* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.)

This example illustrates the fields and controls on the Specify Data Source Parameters page for a pivot grid data source. You can find definitions for the fields and controls later on this page.



Data Source Parameter Details

Displays the **Configure pivot grid views** link, which you click to open the Pivot Grid Views Component page where you can define the pivot grid options.

See “Specifying Display Options” (Pivot Grid).

Data Source Parameters

Pagelet Wizard analyzes the selected pivot grid and generates the list of pagelet parameters shown in this step. You cannot add to the list of parameters.

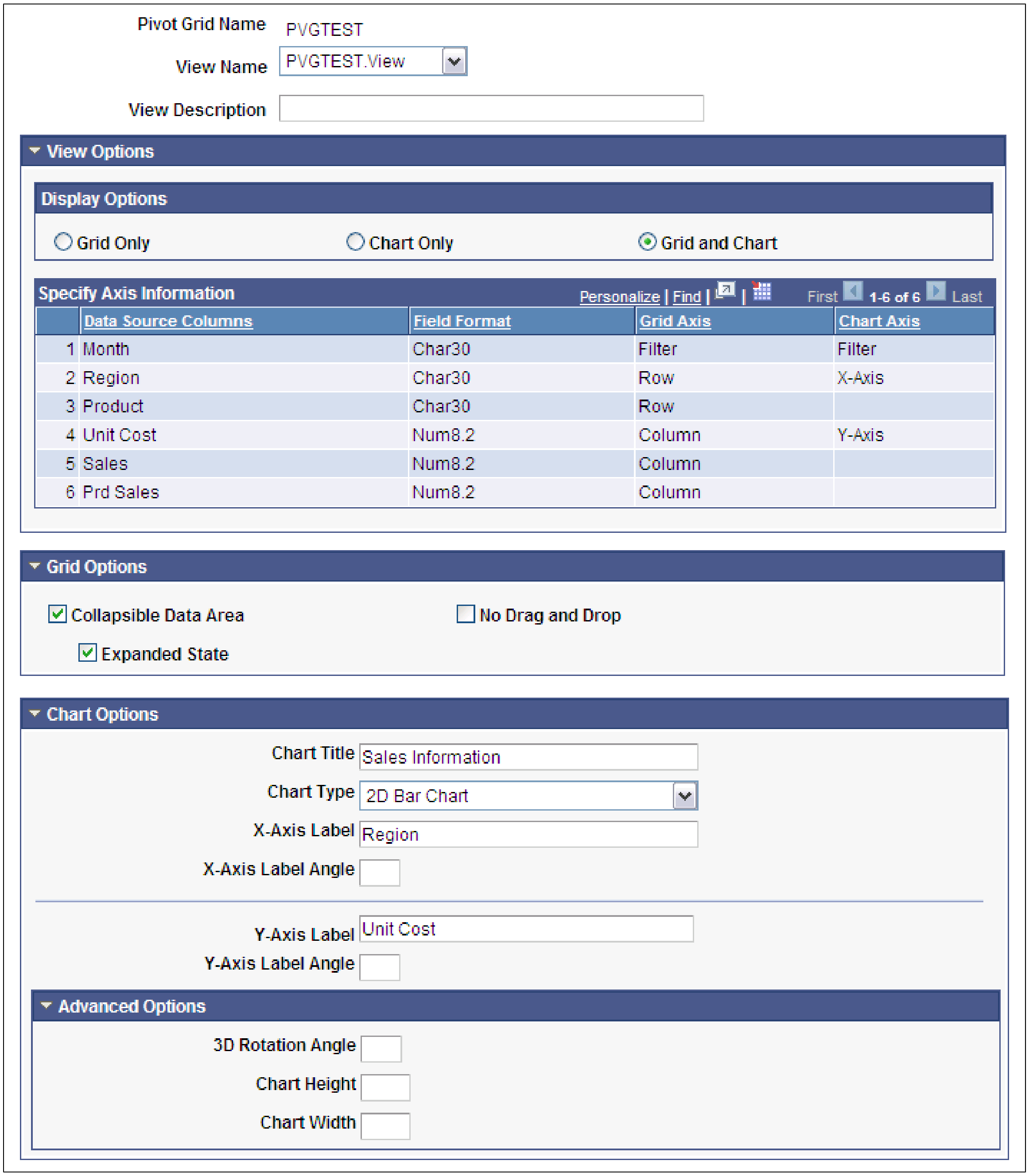
Field or Control	***Description***
Description	Enter text that describes the field.

Field or Control	***Description***
Usage Type	Select the way the system uses the parameter. Select one of these options: • *Admin Specified* • *Context Sensitive* • *Fixed* • *Not Used* • *System Variable* • *User Specified*
Required	Select to require a value for the parameter.
Default Value	Enter the parameter value that the system uses by default.
Reset to Default	Click to set all values on this page to the default values specified by the pivot grid data source API.

Pivot Grid Views Component

Access the Pivot Grid Views page (Click the **Configure pivot grid views** link on the Specify Data Source Parameters page).

This example illustrates the fields and controls on the Pivot Grid Views page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
View Name	Select the pivot grid view to use for this pagelet.
View Description	Enter text that describes the pivot grid view.

Field or Control	***Description***
Display Options	Select which elements appear on the pagelet. Choose one of these values: • *Grid Only* • *Chart Only* • *Grid and Chart*
Grid Options	• **Collapsible Data Area**- select to enable the user to collapse the pivot grid into the grid header. If you select this option, the system automatically selects the **Expanded State** check box. • **Expanded State** - select to display the grid in the expanded state when the pagelet initially appears on the homepage or dashboard. • **No Drag and Drop** - select to disable the drag and drop functionality of elements within the pivot grid.
Chart Options	Enter or select values for the title, type, x and y-axis labels and x and y-axis angles (in degrees) of the chart.
Advanced Options	Enter values for the three-dimensional rotation angle (in degrees), height (in pixels), and width (in pixels) of the chart.
Save As	Click to open the Save View As dialog box, where you can either save the current pivot grid model as a new view or update an existing view.

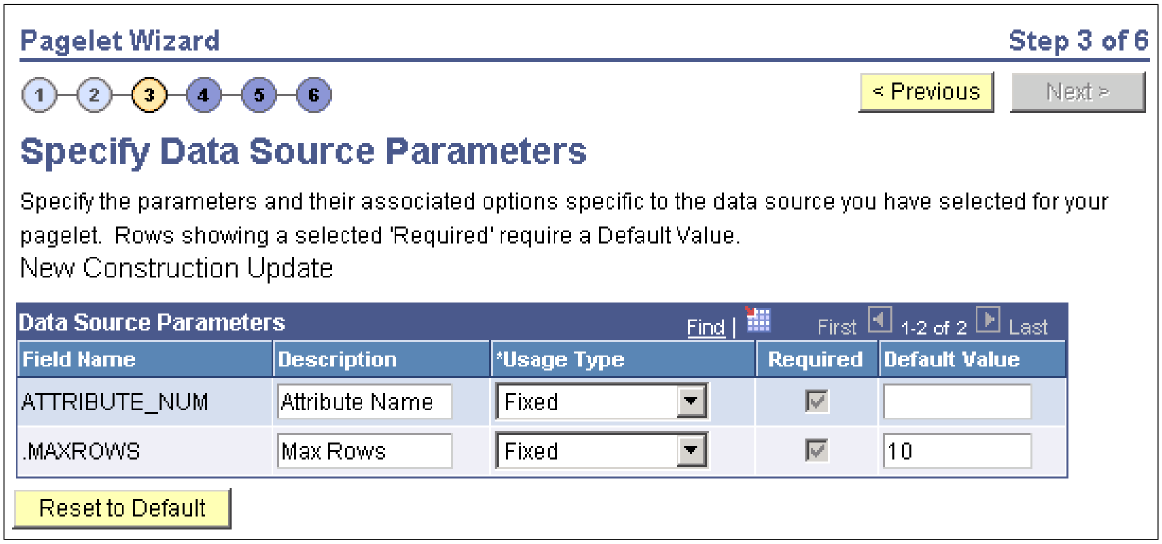
Related Links

“Pivot Grid Viewer Overview” (Pivot Grid)
“Specifying Data Source Parameters” (Pivot Grid)

Specifying Query Data Source Parameters

After selecting *PS Query* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.)

This example illustrates the fields and controls on the Specify Data Source Parameters page for a PeopleSoft query data source. You can find definitions for the fields and controls later on this page.



Data Source Parameters

Pagelet Wizard analyzes the selected query and translates the query prompts into the list of pagelet parameters shown in this step. You cannot add to the list of parameters. Most prompts will be marked as required; however, any prompts that are used exclusively in the IN clause of the query will not be marked as required.

Important! Query Prompts that are based on dynamic views should not be used as User Specified data source parameters. Unexpected results will occur.

You can attain advanced functionality with proper design of the query and the Pagelet Wizard pagelet. For example, you might want to create a pagelet that will show open purchase orders for a list of vendors, and allow each user to specify up to five vendors. To achieve this functionality, you would first create a query to retrieve the open purchase orders for up to five vendors. When you create this query, ensure that the five vendor ID prompts are used only in the IN clause of the query. You can then create a pagelet using the PS Query data type and choose the query that you have created. The five vendor ID prompts will be shown as data source parameters. However, they will be marked as not required. End users will then be able to specify any number of vendors (up to five) when they personalize the pagelet. Finally, update the personalization instructions on the pagelet to explain to the end users that they can supply up to five vendor IDs.

Field or Control	***Description***
Field Name	Displays the data source parameters that correspond to the query prompts associated with the selected query. This field also displays the required system-supplied .MAXROWS query data source parameter that you can use to limit the amount of data returned on the pagelet.

Field or Control	***Description***
Required	This option is selected if the data source parameter must have a value entered on the pagelet to retrieve data successfully. For example, the **.MAXROWS** field is required. If this option is selected and display-only, the value is set in the application class at the application programming interface (API) level. If you leave this option deselected, the data source parameter value can be cleared at runtime, or it can be entered to refine the data selection criteria. Prompts tied to the IN clause of the query will not be marked as required.
Reset to Default	Click to reset the values on this page to the default values specified by the query data source API.
Values	Click the **Values** link to set the default values for users.

Personalization Instructions

Use the **Personalization Instructions** text box to enter custom personalization instructions for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

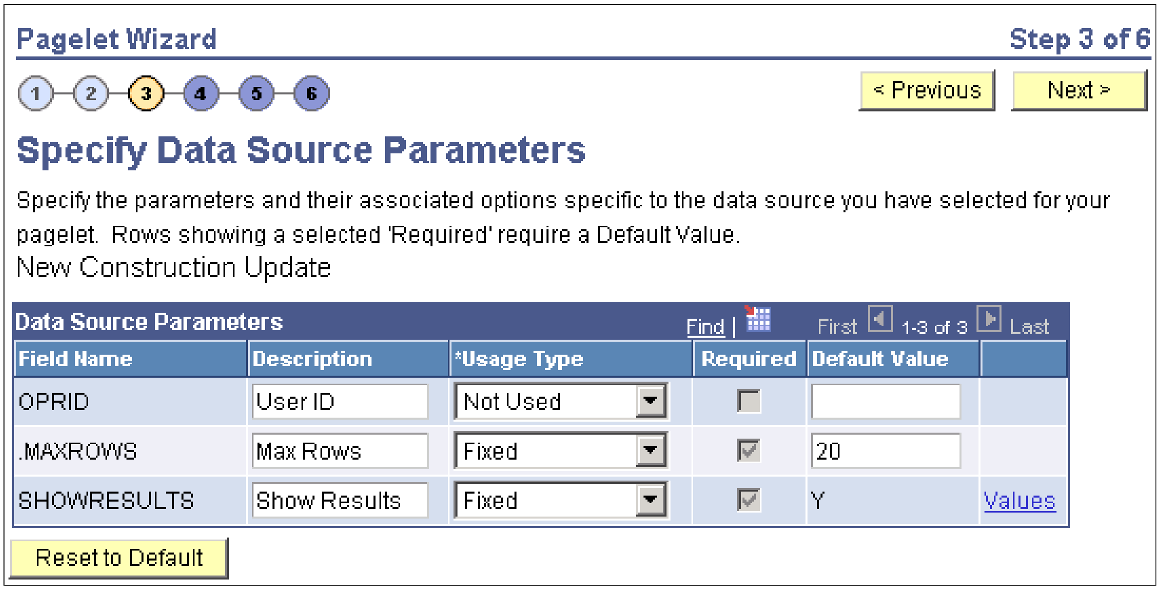
Specifying Rowset Data Source Parameters

When creating a rowset data source pagelet, you skip step 3 and proceed to step 4, in which you select your pagelet display format.

Specifying Search Record Data Source Parameters

After selecting *Search Record* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.)

This example illustrates the fields and controls on the Specify Data Source Parameters for a search record data source. You can find definitions for the fields and controls later on this page.



Data Source Parameters

Pagelet Wizard analyzes the selected search record and retrieves a list of search keys required for the search record. You cannot add to the list of data source parameters.

For fields that are search keys but that are not displayed on the search prompt page or in the result set for the search record (for example, the SETID and BUSINESS_UNIT fields), Pagelet Wizard assimilates the PeopleTools behavior and does not display the keys during design or at runtime. It simply binds in the same values that PeopleTools would use. These values are derived from the default field values established by the user in the PSRECDEFN table.

Field or Control	***Description***
Field Name	Displays the data source parameters that correspond to the search keys required for the search record. This field also displays the required system-supplied .MAXROWS and SHOWRESULTS search record data source parameters. Use the **.MAXROWS** field to limit the amount of data returned to the pagelet. Use the **SHOWRESULTS** field to determine whether search results appear on the pagelet. If this field value is set to *Y,* search record results appear directly on the pagelet with links to transaction data. In addition, if this field value is set to *Y,* the search results grid displays results based on the supplied data source parameter default key field values. If no defaults are supplied, all results appear, up to the .MAXROWS value defined for the pagelet. If this field value is set to *N,* the pagelet displays a search prompt for the selected search record. The **Search By** drop-down list box values that are available on the pagelet are the search keys for the search record. If data source parameter default key field values have been defined, then when the pagelet is initially accessed, the default value appears in the **begins with** pagelet field.
Required	This option is selected if the data source parameter must have a value entered on the pagelet to retrieve data successfully. For example, the **.MAXROWS** and **SHOWRESULTS** fields are required. If this option is selected and display-only, the value is set in the application class at the API level. Search key values are not required because the search record pagelet supports partially populated keys so that you can define the pagelet to return all possible results. If you leave this option deselected, the data source parameter value can be omitted at runtime, or the value can be provided to refine the data selection criteria.
Reset to Default	Click to reset the values on this page to the default values specified by the search record data source API.
Values	Click the **Values** link to set the default values for users.

Personalization Instructions

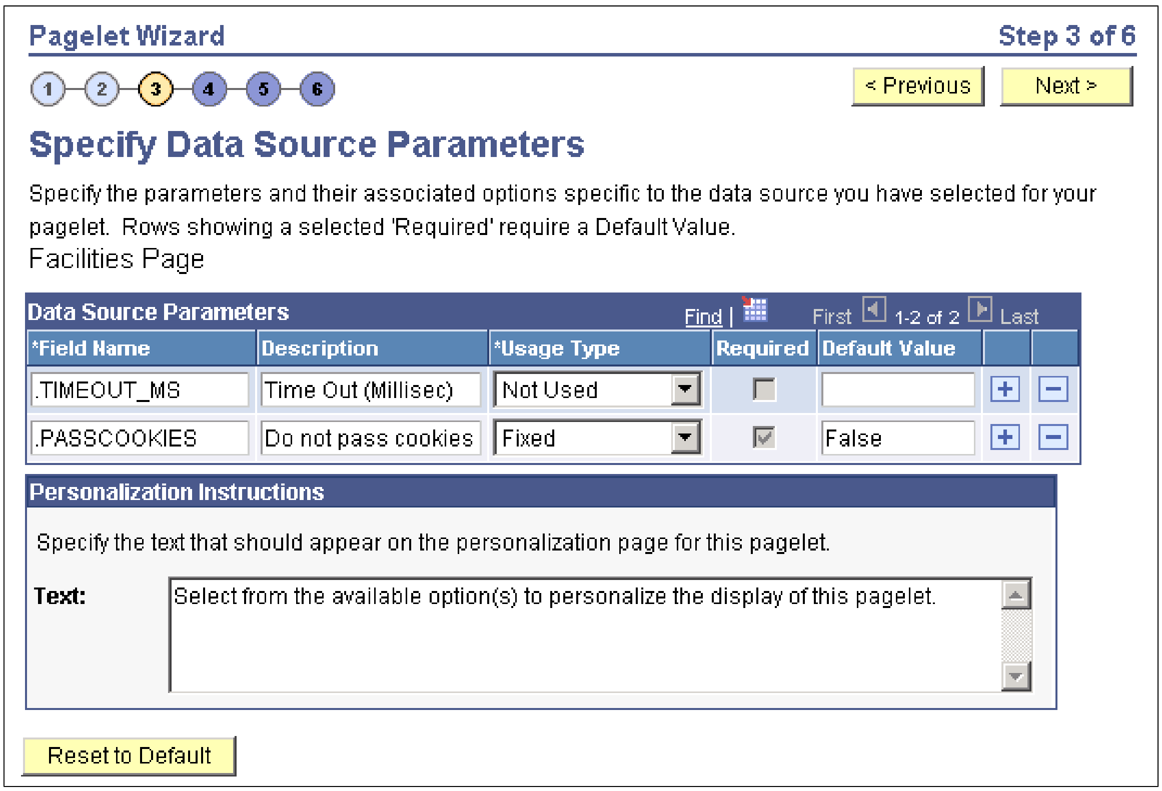
Use the **Personalization Instructions** text box to enter custom personalization instructions for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

Specifying URL Data Source Parameters

After selecting *URL* as the data type on the Select Data Source page, access the Specify Data Source Parameters page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Data Source page, click the Next button.)

This example illustrates the fields and controls on the Specify Data Source Parameters page for a URL data source. You can find definitions for the fields and controls later on this page.



The data source parameters that you can define on this page correspond to URL query parameters, such as an HTTP GET request.

An HTTP GET request is a URL type for which parameters are placed on the query string separated by ? and &, such as http://search.yahoo.com/search?p=searchterm.

For example, the Yahoo! search page URL takes the form http://search.yahoo.com/search?p=searchterm. If you want to make searchterm a user-specified data parameter, enter a **URL** field value of *http://search.yahoo.com/search* in step 2 on the Select Data Source page. In step 3 on this page, add a data source parameter **Field Name** value of *p* and set its **Usage Type** field value to *User Specified* with a default value. Pagelet Wizard then adds the *?p=searchterm* value, with the *searchterm* value able to be personalized by the user, to the URL before it retrieves the content to the pagelet.

You can specify a parameter for the number of milliseconds that a data source will wait for a response before timing out on the homepage. When you set *.timeout_ms,* you can then enter the number of milliseconds that the system will wait.

Passing Cookies

When the URL request is in the AuthenticationToken domain, then the URL data source will pass all cookies on the request to retrieve the content. (The only exception to this rule is if the request is being made back to the local web server. In that instance, only the PS_TOKEN cookie will be passed.) Any requests to servers outside of the AuthTokenDomain will not have any cookies passed.

To turn off the passing of cookies other than PS_TOKEN, add the parameter .passcookies as a Fixed usage type with a default value of False. Adding this parameter with a value of False will instruct pagelet wizard to pass only the PS_TOKEN cookie (and no other cookies) for requests within the AuthenticationToken domain. Adding the .passcookies parameter with a value of True is not recommended because this is the same as the default behavior.

Note: Make sure to examine any security impact that passing cookies might present in your configuration before setting this parameter.

Personalization Instructions

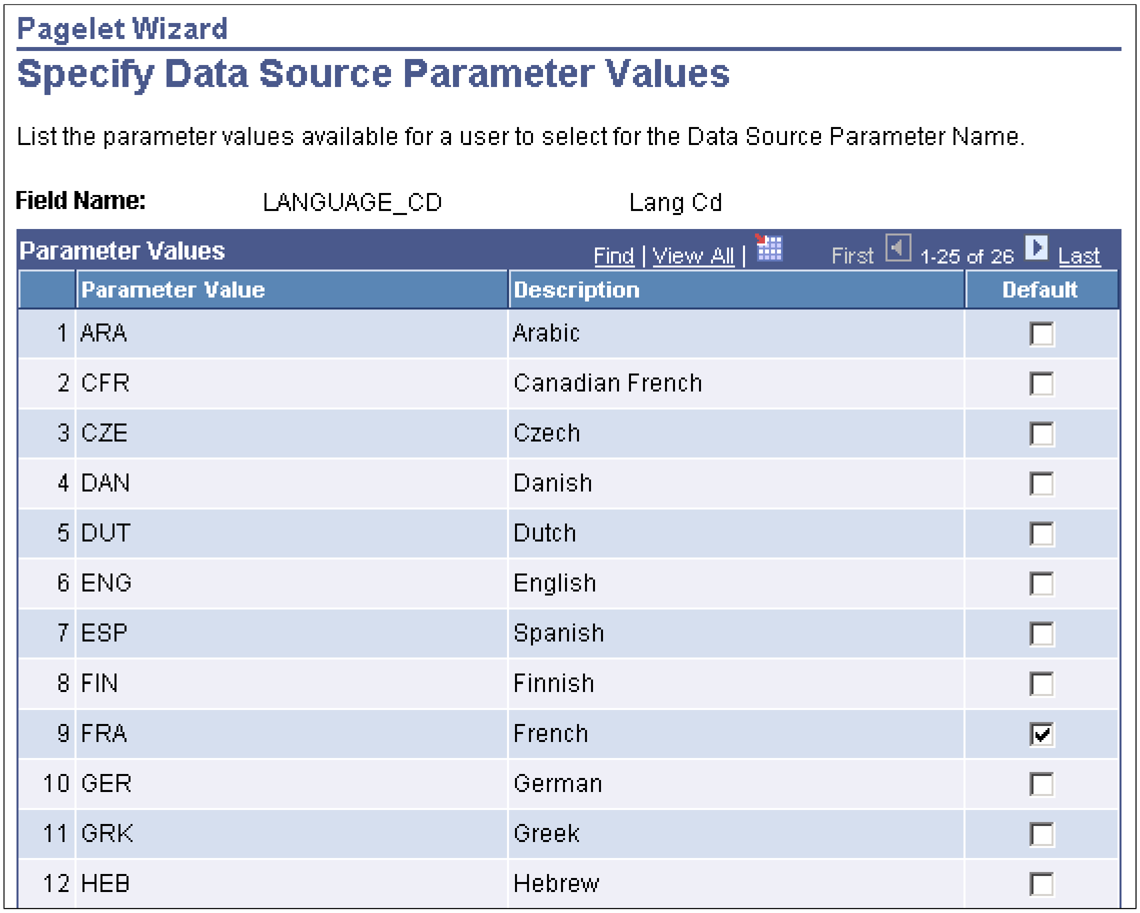
Use the **Personalization Instructions** text box to enter custom personalization instructions for the pagelet. These personalization instructions appear on the Personalization page of the pagelet.

Note: Personalization instructions must be translatable.

Specifying User-Defined Data Source Parameters

Click the **User Values** button on the Specify Data Source Parameters page to display the Specify Data Source Parameters Values page.

This example illustrates the fields and controls on the Specify Data Source Parameter Values page.



Select the parameter value to be displayed to the user as the default value for the field. Users can select from this list of valid values when they personalize the pagelet.

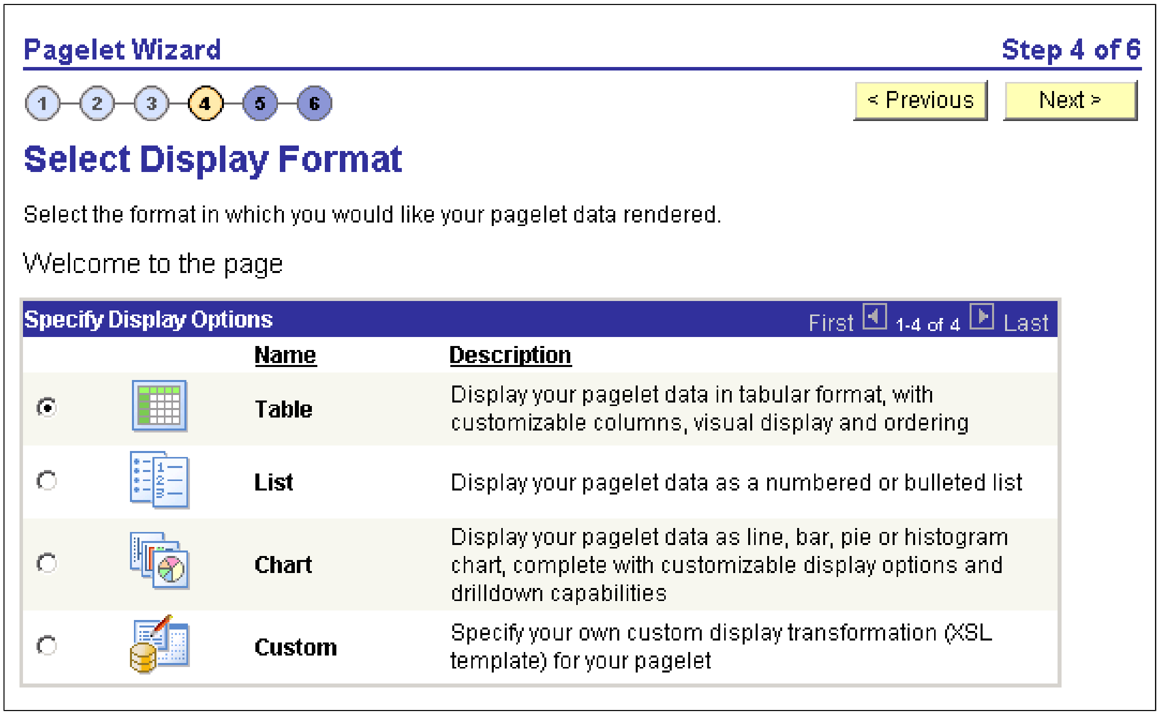
Step 4: Selecting a Pagelet Display Format

This section discusses how to select a display format. This is step 4 in the Pagelet Wizard pagelet creation process.

Selecting a Display Format

Access the Select Display Format page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Specify Data Source Parameters page, click the Next button.)

This example illustrates the fields and controls on the Select Display Format page. You can find definitions for the fields and controls later on this page.



Select the format to use for the pagelet. The display formats that are available for selection on this page depend on the data source that you have selected on the Select Data Source page. Options are:

Field or Control	***Description***
Table	Available for query and rowset data source pagelets.
List	Available for query and rowset data source pagelets.
Chart	Available for query and rowset data source pagelets.
Menu	Available for Navigation Collection data source pagelets.
Search List	Available for search record data source pagelets.
Passthru	Available for activity guide, free text, HTML, OBIEE, and pivot grid data source pagelets.
Custom	Available for activity guide, free text, HTML, navigation collection, query, rowset, and search record data source pagelets.

Related Links

Step 5: Specifying Pagelet Display Options

Step 5: Specifying Pagelet Display Options

This section lists common elements and discusses how to specify pagelet display options. This is step 5 in the Pagelet Wizard pagelet creation process. Specifically, this section discusses how to:

- Specify table display options.
- Specify passthru display options.
- Specify search list display options.
- Specify menu display options.
- Specify list display options.
- Specify chart display options.
- Specify custom display options.
- Specify data link details.

Common Elements Used in This Section

This section discusses common elements used in this section.

Common Links

This table describes the common links used in this section.

Field or Control	***Description***
	Click the **Link Options** icon to display the Link Details page, where you can associate URLs with field values, which at runtime appear on the pagelet as active links that you click to access additional relevant information. See Specifying Data Link Details.

Additional Text

The **Additional Text** group box contains options to add headers, footers, opening text, and closing text to a pagelet.

Field or Control	***Description***
Header	Select a pagelet header. You can select from headers that are defined on the Define Header page. See Defining Headers.
Opening Text	Select the type of opening text that you want to appear on the pagelet. Options are: *Freeform Text/HTML:* The **Text** field appears. Enter the opening text that appears on the pagelet. *Message Catalog Entry:* The **Message Set** and **Number** fields appear. Select the message set and message number that appear as opening text on the pagelet. You can select from the messages defined in the database. See "Message Catalog" (System and Server Administration).
Show "View Source Data" Link	Displays when *PS Query* is selected as the data source for a pagelet. Select to display the **Full Query Results** link on the pagelet. Click the **Full Query Results** link to access the full results of the query used for the pagelet.
Closing Text	Select the type of closing text to appear on the pagelet. Options are: *Freeform Text/HTML:* The **Text** field appears. Enter the closing text that appears on the pagelet. *Message Catalog Entry:* The **Message Set** and **Number** fields appear. Select the message set and message number that appear as closing text on the pagelet. You can select from the messages defined in the database. See "Using Administration Utilities" (System and Server Administration).
Footer	Select a pagelet footer. You can select from footers that are defined on the Define Footer page. See Defining Footers.

Search Options

The **Search Options** group box contains the options to override the default search functionality for the current pagelet only.

See Defining Pagelet Wizard Transformation Types, Display Formats, and Data Types.

Field or Control	*Description*
Search Box	Select a value from these options to determine the location of the search box field: • *None:* Do not show the search box. This is the default option. • *Top Left:* The search box appears at the top left corner of the pagelet, above the pagelet content but below any header and opening text. • *Top Right:* The search box appears at the top right corner of the pagelet, above the pagelet content but below the header and opening text. • *Bottom Left:* The search box appears at the bottom left corner of the pagelet, below the pagelet content but above any footer and closing text. • *Bottom Right:* The search box appears at the bottom right corner of the pagelet, below the pagelet content but above any footer and closing text. • *Inline XSL:* A PSSEARCHBOX node must be written into the transformation XSL to indicate where the search box appears. If multiple PSSEARCHBOX nodes are in the pagelet content after the transformation, only the first node is replaced by the search box. The PASSTHRU display format does not support this type of search box. If the display format is PASSTHRU, no search box appears in the pagelet.
Package	Enter the application class package name that you want to use to define and build the search for the pagelet.
Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method used to define and build the search for the pagelet.
Class ID	Enter the application class name that you want to use to define and build the search for the pagelet. The class must exist in the application package and subpackage that you specify.

Note: If you have not defined default **Search Option** values for the data type on the Define Data Types page (PeopleTools, Portal, Pagelet Wizard, Define Pagelet Data Types), the **Package, Path,** and **Class ID** fields have no default values. You must enter values if you want to activate the pagelet search feature.

Customization

The **Customization** group box contains the option to customize your pagelet.

Field or Control	*Description*
Customize	standardClick to customize the pagelet by applying a custom XSL template, or by directly editing the XSL resulting from the pagelet display options that you have defined so far. When you click **Customize,** you switch from a standard display format selected on the Select Display Format page to a custom format of your definition. The XSL field is populated based on preexisting display options that are set for the original standard display format. You can edit the XSL directly in the XSL field and retain as much of the original standard display options as you want. Remember that if you apply an XSL template using the **XSL Template** field, the XSL is regenerated based on the selected template, and the original standard display options that you have set may be overridden. Making any edits to the XSL, either directly or by assigning a template, disconnects the pagelet definition from the original standard display format. When you click the **Customize** button, the current pagelet definition is saved for restoration purposes. You can revert to this standard pagelet definition by clicking the **Undo Customizations** button. Clicking this button assigns your pagelet to the Custom design format on the Select Display Format page - Step 4. When you return to the Specify Publishing Options page - Step 5, you access the version of the page for the Custom display format. See Specifying Custom Display Options.

Specifying Table Display Options

After selecting the *Table* option on the Select Display Format page, select the Specify Display Options page (**PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Format page, click the Next button.)

This example illustrates the fields and controls on the Specify Display Options page for the table display format. You can find definitions for the fields and controls later on this page.

Table Options

Select options that will display grid lines, alternate row shading, and row numbers on the table-formatted pagelet.

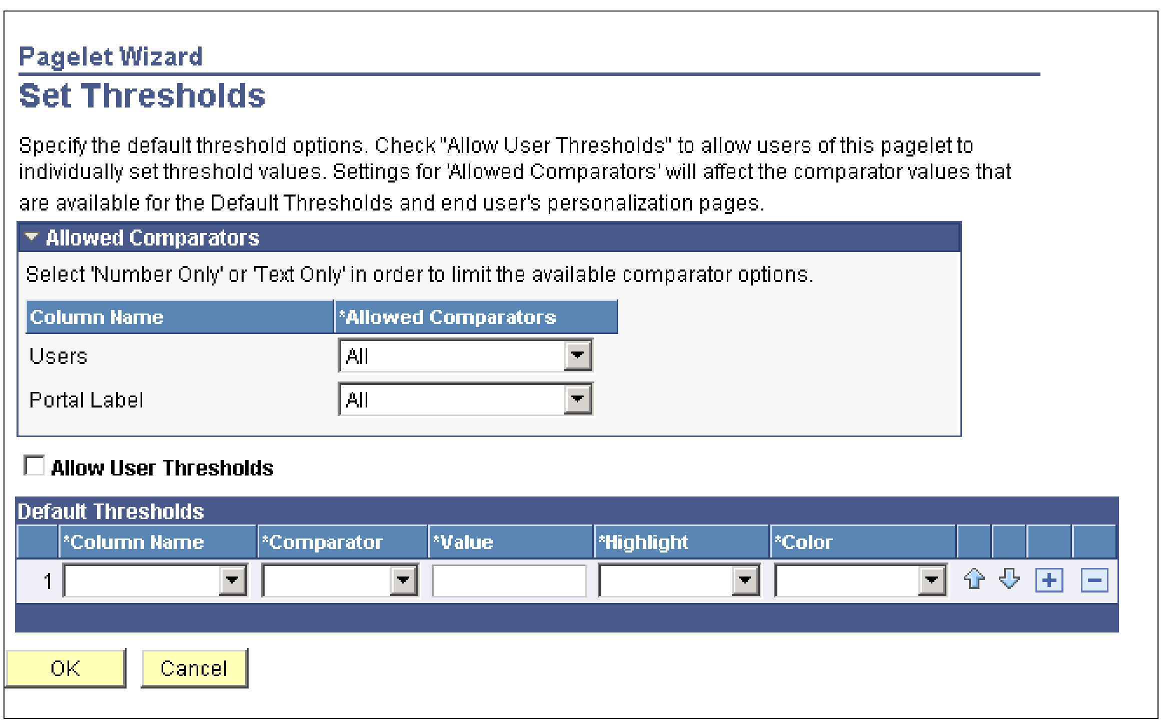
Columns

Field or Control	*Description*
Column Name	Displays possible column headings provided by the selected data source parameters. Use the drop-down list boxes to set the order in which the columns appear on the table-formatted pagelet. Use the **Add** button or the **Delete** button to add or remove columns.
Alignment	Specify the alignment of the text in the column. Options are *Center, Left,* and *Right.*
	Click the **Link Options** icon to display the Link Details page, where you can associate URLs with field values, which at runtime appear on the pagelet as active links that you click to access additional relevant information. See Specifying Data Link Details.

Setting Thresholds

Access the Set Thresholds page by clicking the **Set Thresholds** button in the Table Options of step 5 in Pagelet Wizard.

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



Use the Set Thresholds page to set the column alignment and threshold options.

Field or Control	***Description***
Allowed Comparators	The system displays the column names and enables you to set the comparators for that column based on numbers, text, or both.
Allow User Thresholds	Select to enable users to set their own threshold values for the table from the Personalization page.

Set the display order and attributes of each column. Use the up and down arrows to change the display order of the columns, if necessary. Each column threshold can have a unique color associated with it—duplicates are allowed. If more than one threshold rule applies for a given value, the rule that is positioned highest in the list is the one that is applied.

Field or Control	***Description***
Column Name	Select the column name from the drop-down list box.
Comparator	Select a comparator value for each column.
Value	Enter the threshold value for the column. When this value is reached, the highlight and color attributes become active.
Highlight	If you choose to have highlighting for the threshold, select either the entire row or just a single cell.
Color	Select the color for the column.
	Use the arrows to move each individual column up or down in the table.

Pagelet Preview

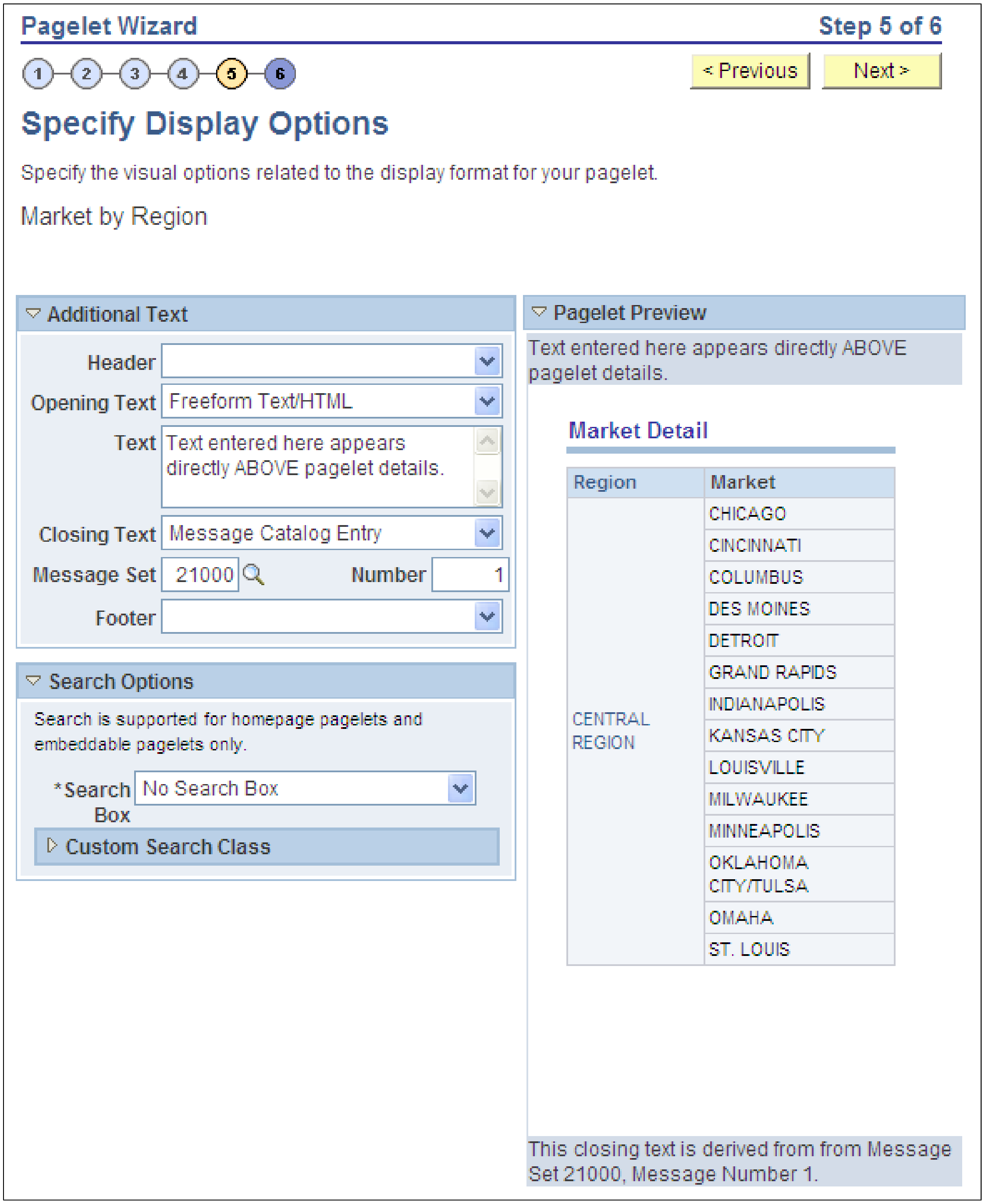
Displays a preview of the pagelet based on most display option settings on this page.

Specifying Passthru Display Options

After selecting the *Passthru* option on the Select Display Format page, select the Specify Display Options page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Format page, click the Next button.)

Note: For OBIEE Reports and pivot grids, *Passthru* is the only display format option.

This example illustrates the fields and controls on the Specify Display Options page for the passthru display format. You can find definitions for the fields and controls later on this page.



Additional Text

Field or Control	***Description***
Header	Select a value from this field to display a header below the pagelet header, but above any opening text.

Field or Control	***Description***
Opening Text	Select a value to include text *above* the pagelet details. Select from these values: • *Freeform Text/HTML* Select to enable the **Text** field and manually enter static text or HTML. • *Message Catalog* Select to enter the message set and message number from which to derive the opening text from the message catalog.
Text	Enter text or HTML. This field appears when you select *Freeform Text/HTML* as the opening or closing text.
Closing Text	Select a value to include text *below* the pagelet details. Select from these values: • *Freeform Text/HTML* Select to enable the **Text** field and manually enter static text or HTML. • *Message Catalog* Select to enter the message set and message number from which to derive the opening text from the message catalog.
Footer	Select a value from this field to display a footer above the Footer bar, but below any closing text.

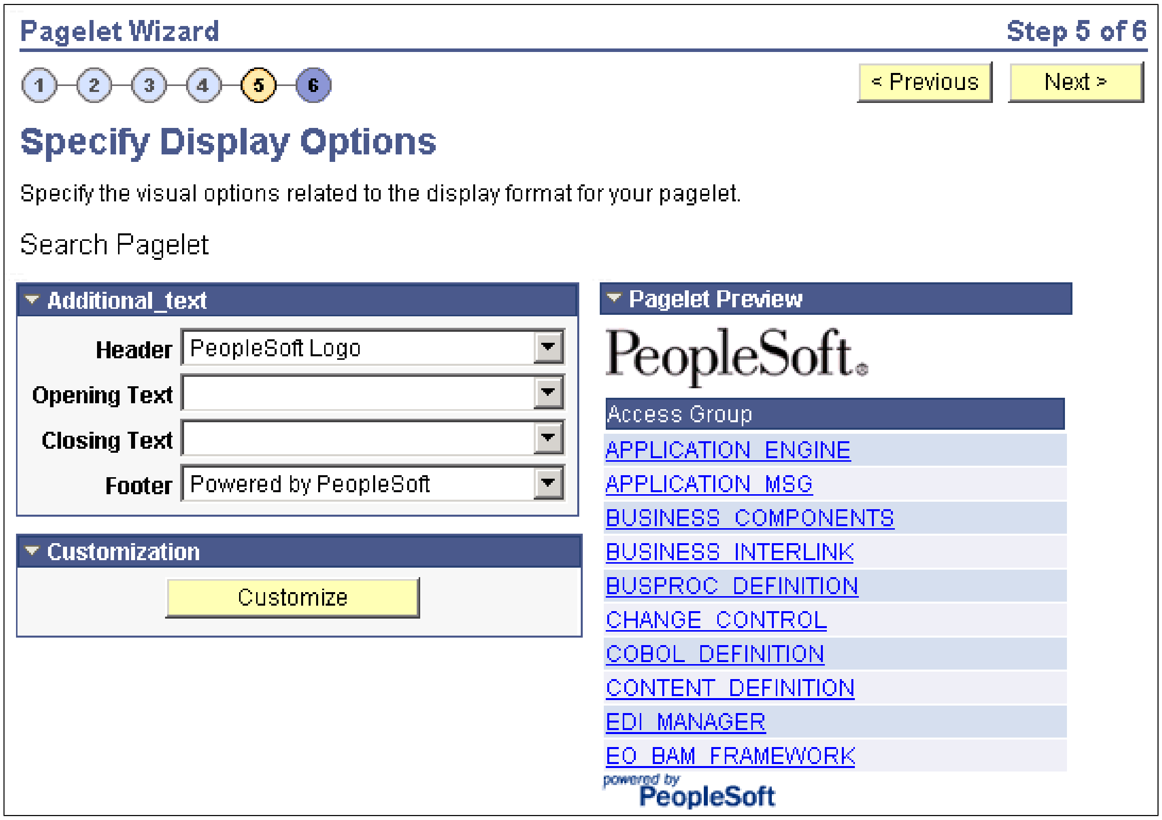
Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Search List Display Options

After selecting the *Search List* option on the Select Display Format page, select the Specify Display Options page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Format page, click the Next button.)

This example illustrates the fields and controls on the Specify Display Options page for the search list display format. You can find definitions for the fields and controls later on this page.



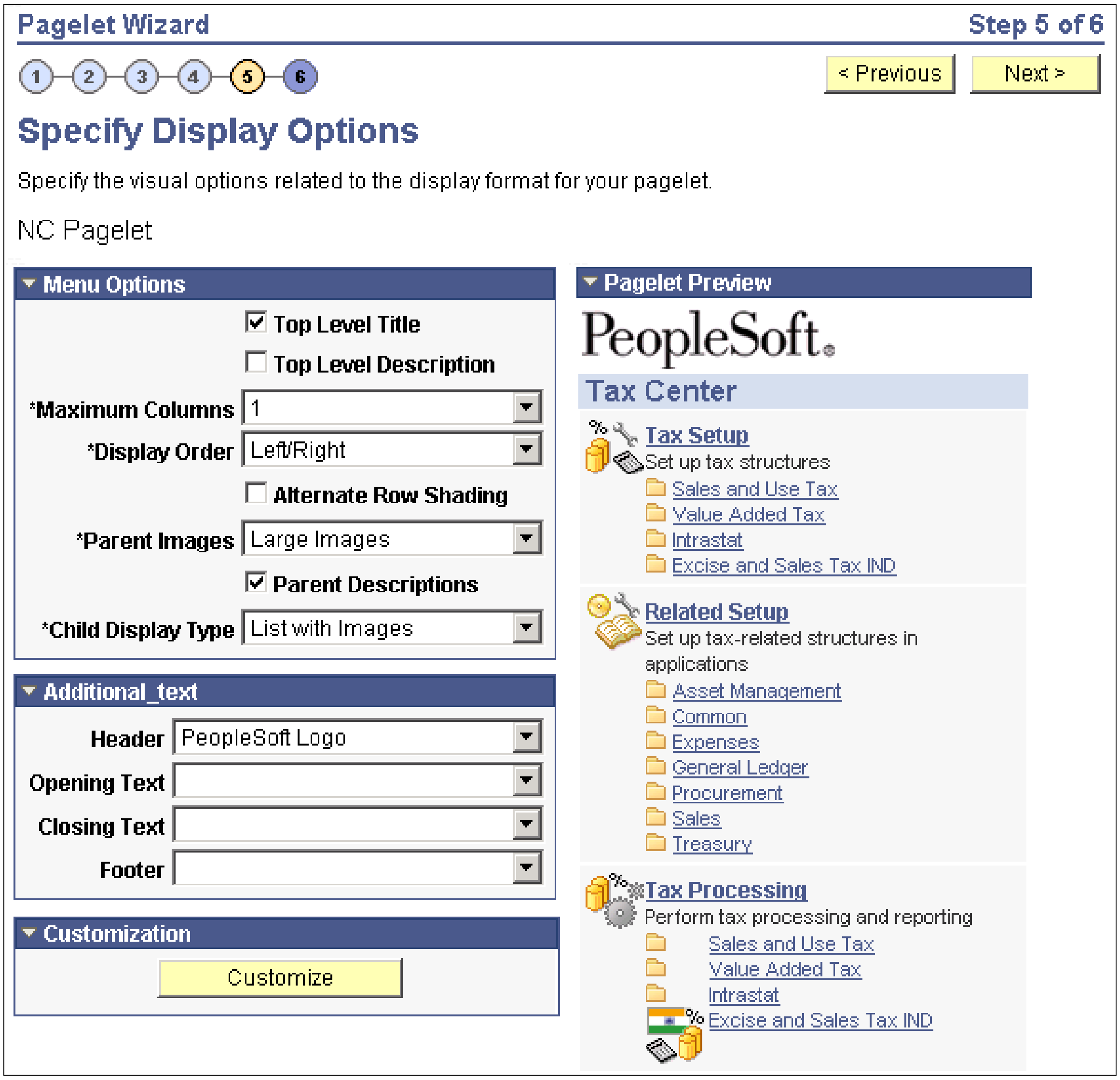
Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Menu Display Options

After selecting the *Menu* option on the Select Display Format page, select the Specify Display Options page (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Format page, click the Next button.)

This example illustrates the fields and controls on the Specify Display Options page for the menu display format. You can find definitions for the fields and controls later on this page.



Menu Options

Field or Control	***Description***
Top Level Title	Select to have the menu pagelet display the top-level title defined in the underlying navigation collection.
Top Level Description	Select to have the menu pagelet display the top-level description defined in the underlying navigation collection.

Field or Control	***Description***
Maximum Columns	Select the number of columns to display (options are *1* through *4*) on the navigation collection menu pagelet. Narrow pagelets display contents in one column. Wide pagelets can display contents in up to four columns. See Understanding Pagelet Wizard Display Sizes.
Display Order	Select the order in which you want navigation collection elements to appear on the menu pagelet. When you create a navigation collection, you define the order of folders and links in the collection using options in the **Placement Properties** group box on the Edit Folder page and the Edit Link page. Options are: *Left to Right:* Select to display the elements from left to right on the menu pagelet. *Top to Bottom.* Select to display the elements from top to bottom on the menu pagelet.
Alternate Row Shading	Select to display alternate row shading on the menu-formatted pagelet.
Parent Images	Select the type of images that you want to display for parent-level navigation collection elements (folders) on the menu pagelet. When you create a navigation collection, the collection definition includes specified large and small images that appear for parent-level elements in the navigation collection. Options are: *Large Images:* Select to have the menu pagelet display the large folder image defined for the underlying navigation collection. *No Images:* Select to prevent parent-level images from appearing on the menu pagelet. *Small Images:* Select to have the menu pagelet display the small folder image defined for the underlying navigation collection.
Parent Descriptions	Select to have the menu pagelet display parent-level folder descriptions defined in the underlying navigation collection.

Field or Control	*Description*
Child Display Type	Select the display type for child links on the menu pagelet. Options are: *Comma Delimited:* Displays the maximum number of child links separated by commas. *List with Images:* Displays the maximum number of child links accompanied by icons. This icon is the small content image selected in the navigation collection system or registry options for the underlying navigation collection. *Simple List:* Displays the maximum number of child links, one child link per row. No bullets, numbers, or icons appear in the list.

Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

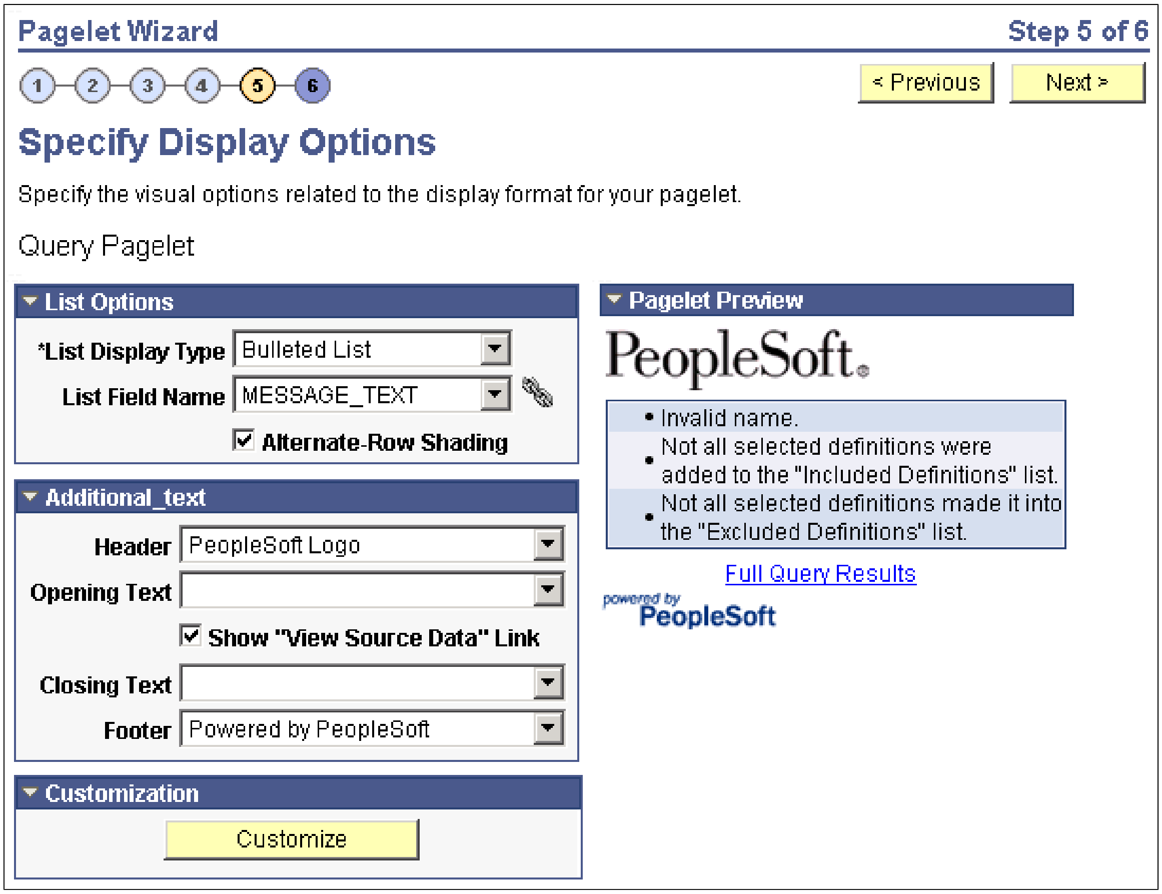
Related Links

Understanding Navigation Collections

Specifying List Display Options

After selecting the *List* option on the Select Display Format page, select the Specify Display Options page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Format page, click the Next button.)

This example illustrates the fields and controls on the Specify Display Options page for the list display format. You can find definitions for the fields and controls later on this page.



List Options

Field or Control	***Description***
List Display Type	Select the list format for displaying query results on the list pagelet. Options are: *Bulleted List:* Displays query results in a bullet list. *Numbered List:* Displays query results in a numbered list. *Simple List:* Displays query results in a list with no bullets or numbers.
List Field Name	Select the field for which values are displayed in the list on the pagelet.
Alternate-Row Shading	Select to use alternate-row shading on the list pagelet.

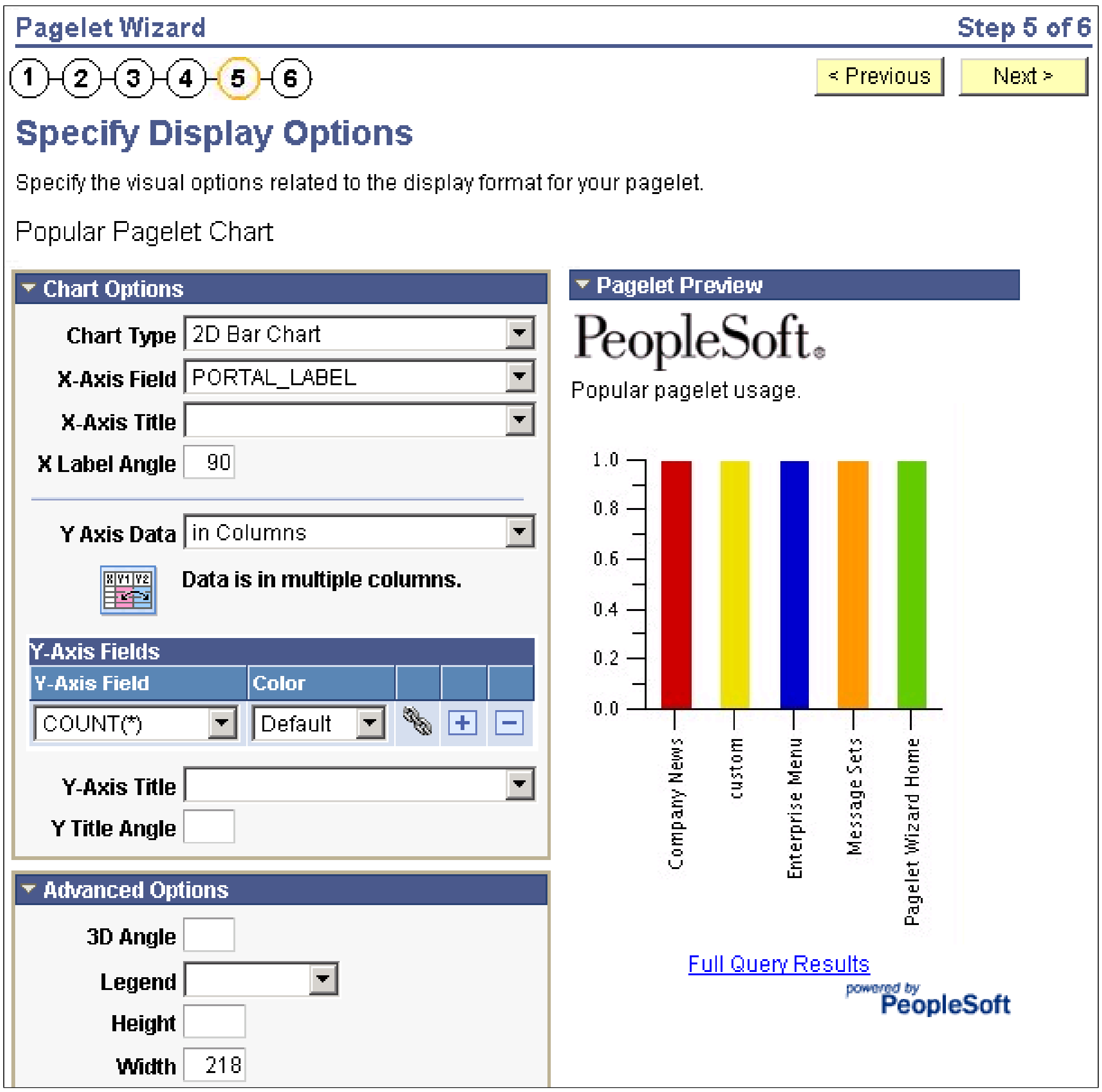
Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

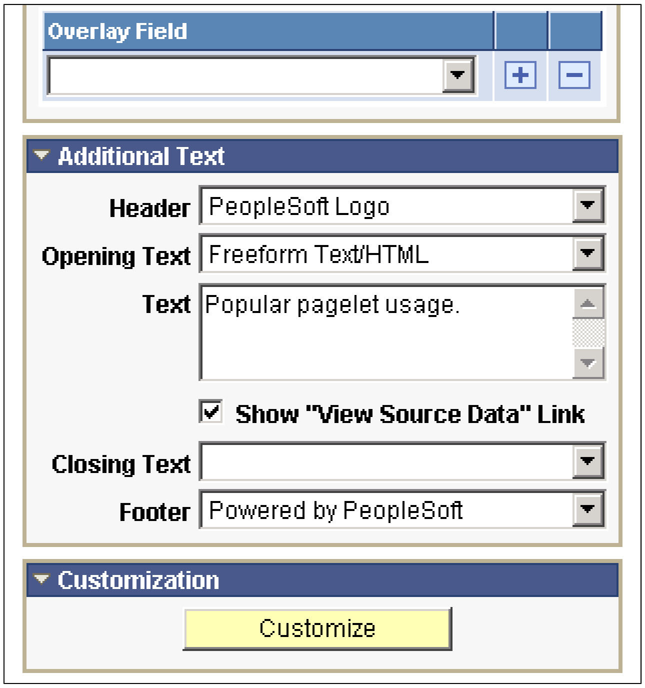
Specifying Chart Display Options

After selecting the *Chart* option on the Select Display Format page, select the Specify Display Options page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Format page, click the Next button.)

This example illustrates a portion of the fields and controls on the Specify Display Options page for the chart display format. You can find definitions for the fields and controls later on this page.



This example illustrates the remainder of the fields and controls on the Specify Display Options page for the chart display format. You can find definitions for the fields and controls later on this page



To generate a chart pagelet using Pagelet Wizard, you include information related to the chart's *x* and *y* axes. The *y* axis is the axis that contains the query data. In most charts, the *y* axis is the vertical axis. However, in a horizontal bar chart, the *y* axis is the horizontal axis. The *x* axis is the axis against which the *y*-axis data is measured.

For example, a chart pagelet might have a *y* axis that displays the number of times that a pagelet has been accessed. This access count could be displayed against an *x* axis that shows the name of the pagelet accessed.

Chart Options

Field or Control	***Description***
Chart Type	Select the type of chart to use to display the query results on the pagelet. Available chart formats are those supported by the PeopleCode Chart class. See "Understanding the Charting Classes" (PeopleCode API Reference).
X-Axis Field	Select the field to use to derive *x*-axis values for the chart. Available values are determined by the output fields of the data source.

Field or Control	***Description***
X-Axis Label	Select the field that you want to use to derive the labels for the x-axis values for the chart. Available values are determined by the output fields of the data source.
X Label Angle	Enter the angle at which you want the text of the *x*-axis labels to appear along the *x* axis on the pagelet.
Y Axis Data	Select to indicate whether the *y*-axis data is stored in rows or columns in the selected query. This enables Pagelet Wizard to read query data and generate the chart. *in Columns:* Select if the *y*-axis data is stored in columns.

In this example, the query data is stored in columns (labeled Revenue and Profit) as shown here:

Quarter	***Revenue***	***Profit***
Q1	3562	1010
Q2	3473	1290
Q3	4568	1490
Q4	5668	2769

Field or Control	***Description***
Y Axis Data (continued)	*in Rows:* Select if the *y*-axis data to be used to generate the chart is stored in rows.

In this example, the same data is stored in rows with Revenue and Profit stored as data values in the Type column as shown here:

Quarter	***Type***	***Value***
Q1	Revenue	3562
Q1	Profit	1010

Quarter	***Type***	***Value***
Q2	Revenue	3473
Q2	Profit	1290
Q3	Revenue	4568
Q3	Profit	1490
Q4	Revenue	5668
Q4	Profit	2769

Field or Control	***Description***
Y-Axis Field	Select the field that you want to use to derive *y*-axis values for the chart. Available values are determined by the output fields of the data source. If you set the **Y Axis Data** field to *in Columns,* you can select up to four *y*-axis fields to create a chart pagelet that displays data for multiple fields. For example, if you track sales figures for several departments over multiple years, you can assign each department its own *y*-axis value.
Color	Select the color for the *y*-axis data. If you select multiple *y*-axis fields, selecting a different color for each field improves pagelet usability. This option is available only if the *y*-axis data is in stored columns.
Y-Axis Series	This field is available if the **Y Axis Data** field is set to *Rows.* Select the field to use to generate a color-coded series of *y*-axis data. Colors are automatically assigned, but you can override them by supplying custom XSL.

Advanced Options

Field or Control	***Description***
3D Angle	If you select a 3D chart type, enter the angle at which you want the 3D data to be displayed.

Field or Control	*Description*
Legend	Select the area of the pagelet for the chart legend to appear. Options are: • *Bottom* • *Left* • *None* • *Right* • *Top*
Height	Enter the height of the chart in pixels.
Width	Enter the width of the chart in pixels. If you do not enter height or width values, the chart is automatically sized based on the pagelet position and size (narrow or wide). See Understanding Pagelet Wizard Display Sizes.
Overlay Fields	Select up to three fields for which you want to display data using a line drawn over the background chart. Overlays apply to all chart types except 2D and 3D pie charts.

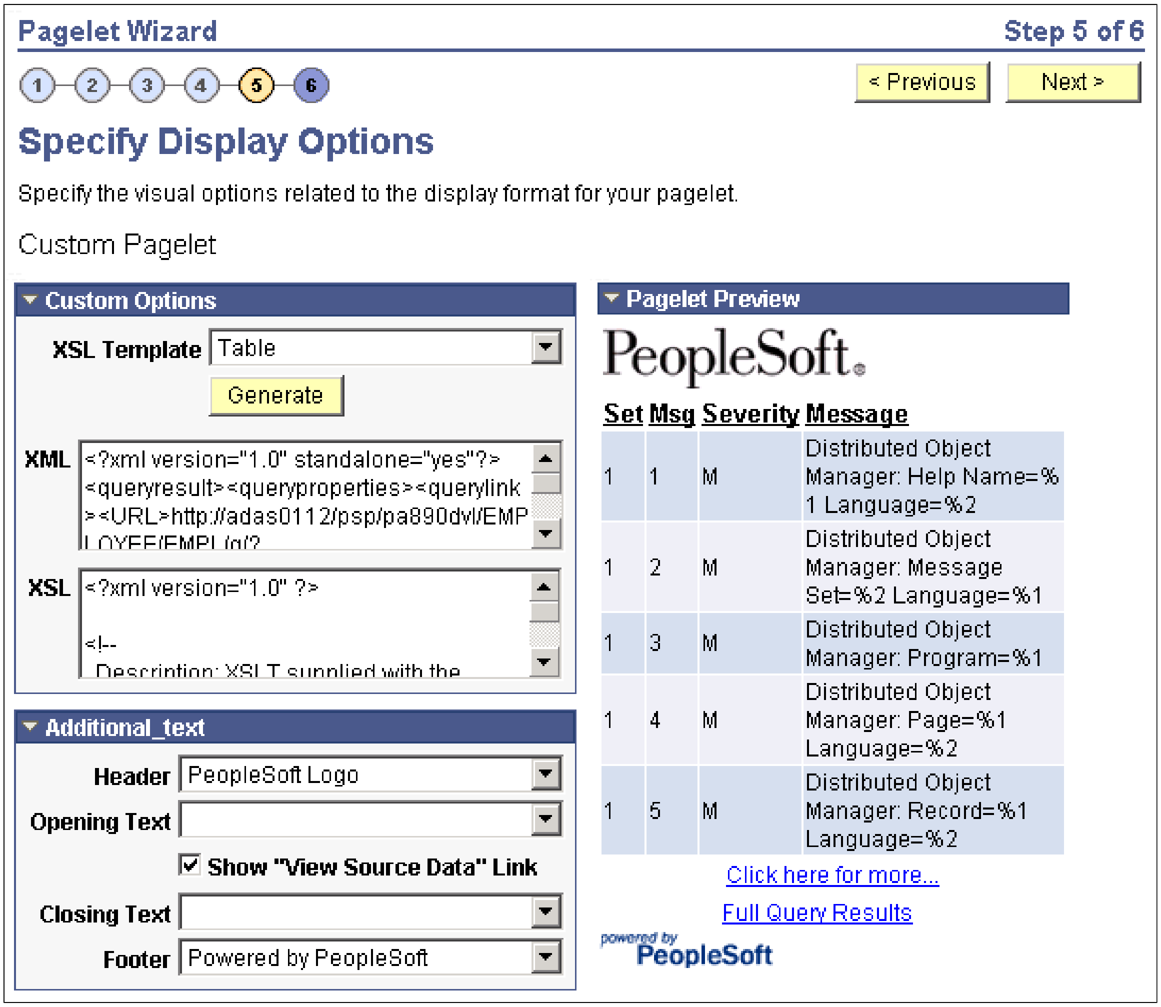
Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Custom Display Options

After selecting the *Custom* option on the Select Display Format page, select the Specify Display Options page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Format page, click the Next button.)

This example illustrates the fields and controls on the Specify Display Options page for the custom display format. You can find definitions for the fields and controls later on this page.



Custom Options

Field or Control	***Description***
XSL Template	Select the XSL template to use to generate the custom format pagelet. Available XSL templates are defined on the Define XSL page. Only XSL templates defined for the selected data source are displayed. See Defining Pagelet Wizard XSL Prototypes.
Generate	Click to generate custom XSL for the pagelet based on the selected XSL template. This XSL is displayed in the XSL text box and is applied to the XML in the XML text box to generate the pagelet preview.

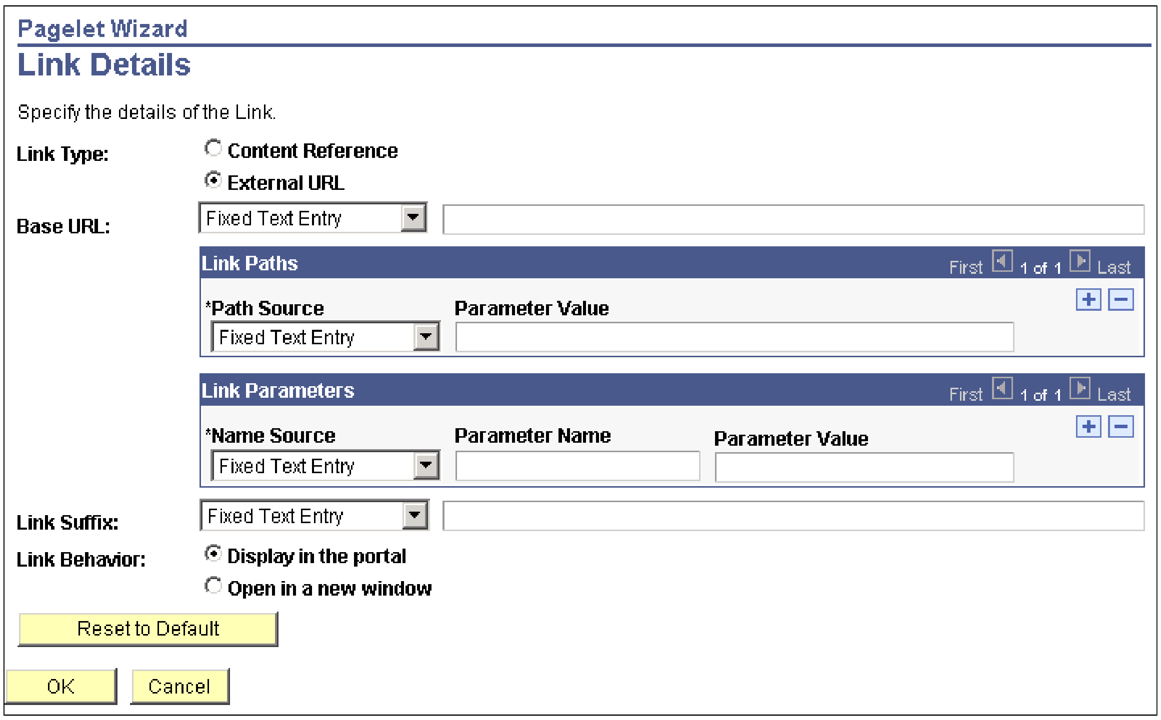
Pagelet Preview

Displays a preview of the pagelet based on most display option settings on this page.

Specifying Data Link Details

Access the Link Details page. (Click the **Link Options** icon on the Specify Display Options page.)

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



You can access this page for field or column names that appear on the Specify Display Options page if you select the table, rowset, list, and chart formats. Entering information on the Link Details page enables you to associate URLs with field values, which at runtime appear on the pagelet as active links that you click to access additional relevant information.

Pagelet Wizard builds the column data link in the follow way using link path, parameter, and suffix values:

http://baseURL/linkpath1/linkpath2/…/linkpathN/?parm1=val1&parm2=val2&…&parmN=valNlinksuffix

Field or Control	***Description***
Link Type	Select the type of link destination to define for the data. Options are: *Content Reference:* Select to define data links to a content reference. *External URL:* Select to define data links to an external URL.
Portal Name	Select the portal in which the content reference resides. This field appears when you select *Content Reference* as the link type.
Content Reference	Select the content reference to which the link points. This field appears when you select *Content Reference* as the link type.
Base URL	Enter the URL to which you want the link to point. This field appears when you select *External URL* as the link type. Options are: *Specified by Data:* Select to have the system dynamically generate the link information based on the selected data field. *Fixed Text Entry:* Select to hard-code the link information by entering a value.

Link Paths

Field or Control	***Description***
Path Source	Select the method to use to determine the link destination. Options are: *Specified by Data:* Select to have the system dynamically generate the link information based on the selected data field. *Fixed Text Entry:* Select to hard-code the link information by entering a value.
Link Path	This field appears if you select *Specified by Data* as the path source. Select the data field to use to generate the link.
Parameter Value	This field appears if you select *Fixed Text Entry* as the path source. Enter a parameter value.

Link Parameters

Field or Control	***Description***
Name Source	Select the method to use to determine the link destination. Options are: *Specified by Data:* Select to have the system dynamically generate the link information based on the selected data field. *Fixed Text Entry:* Select to hard-code the link information by entering a value.
Parameter Name	Enter a value that corresponds to the value in the **Parameter Value** field.
Parameter Value	If you have selected *Specified by Data* as the name source, select a data field. If you selected *Fixed Text Entry* as the name source, enter a parameter value.
Link Suffix	Select the method to use to determine the link destination. Options are: *Specified by Data:* Select to have the system dynamically generate the link information based on the selected data field. *Fixed Text Entry:* Select to hard-code the link information by entering a value. If you selected *Specified by Data* as the link suffix, select a data field. If you selected *Fixed Text Entry* as the link suffix, enter a parameter value.
Link Behavior	Options are: *Display in portal:* Select if you want the content accessed by the data link to be displayed within the portal. *Open in a new window:* Select if you want the content accessed by the data link to be displayed in a new window.
Reset to Default	Click to clear the values on this page and remove the link.

PeopleTools Portal Technologies XSL Templates

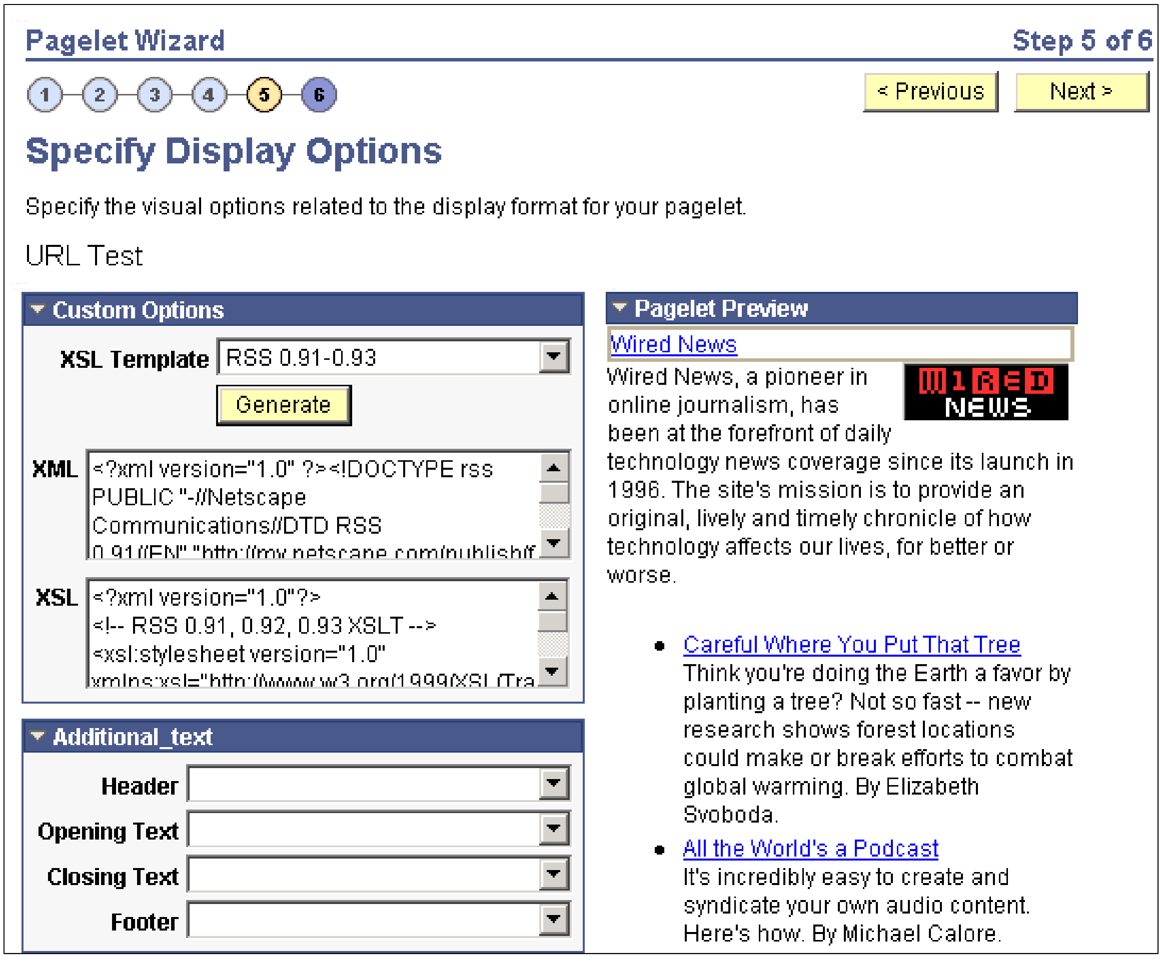
PeopleTools Portal Technologies delivers seven XSL templates that enable you to transform content from internet news sources that use industry-standard formats such as RSS and ATOM. When you select the URL data source and the custom display format, the following additional templates appear in the drop-down list box in Step 5 of Pagelet Wizard.

These include:

- ATOM 0.3
- ATOM 1.0
- RSS
- RSS 0.91 - 0.93
- RSS 1.0
- RSS 2.0
- Skeletal XSL

Select the appropriate XSL template from the list and click the **Generate** button. The formatted news articles appear in the pagelet preview on the right side.

This example illustrates the fields and controls on the Specify Display Options page (using an XSL template).



Step 6: Specifying Pagelet Publishing Options

This section provides an overview of caching options and discusses how to specify pagelet publication options. This is step 6 in the Pagelet Wizard pagelet creation process.

Understanding Caching Options

Enabling caching improves performance because after the pagelet is stored in the cache memory, the next time the pagelet is accessed, the information is retrieved and displayed from cache memory instead of requiring a trip to the server. Three caching options exist:

- Private.

  Private caching is user-based; pagelet caching is stored per user. For example, if user A accesses the pagelet, only user A will access the pagelet from the cache.

- Public.

Public caching is application-based; pagelet caching is stored per web server instance. For example, if a user accesses the pagelet using web server instance 1, only other users accessing the pagelets using web server instance 1 will access the pagelet from the cache.

- Role.

  Role caching is role-based; pagelet caching is stored per *user role combination*. For example, if user A has membership in roles 1, 2, and 3, only another user with all roles 1, 2, and 3 will access the pagelet from the cache.

Content Reference Attributes

If you enable caching, Pagelet Wizard adds the following attribute information to the pagelet content reference:

- Name: *PSCACHECONTROL*
- Label: *PSCache-Control*
- Attribute value: *Caching Type field value;max-age=Minutes to Cache field value*

  For example, *Public;max-age=30*.

After the pagelet is saved and registered, you can use Pagelet Wizard to alter caching options in the same way that you can alter other registration information. However, altering caching option values requires that you reregister the pagelet.

See Administering Content References.

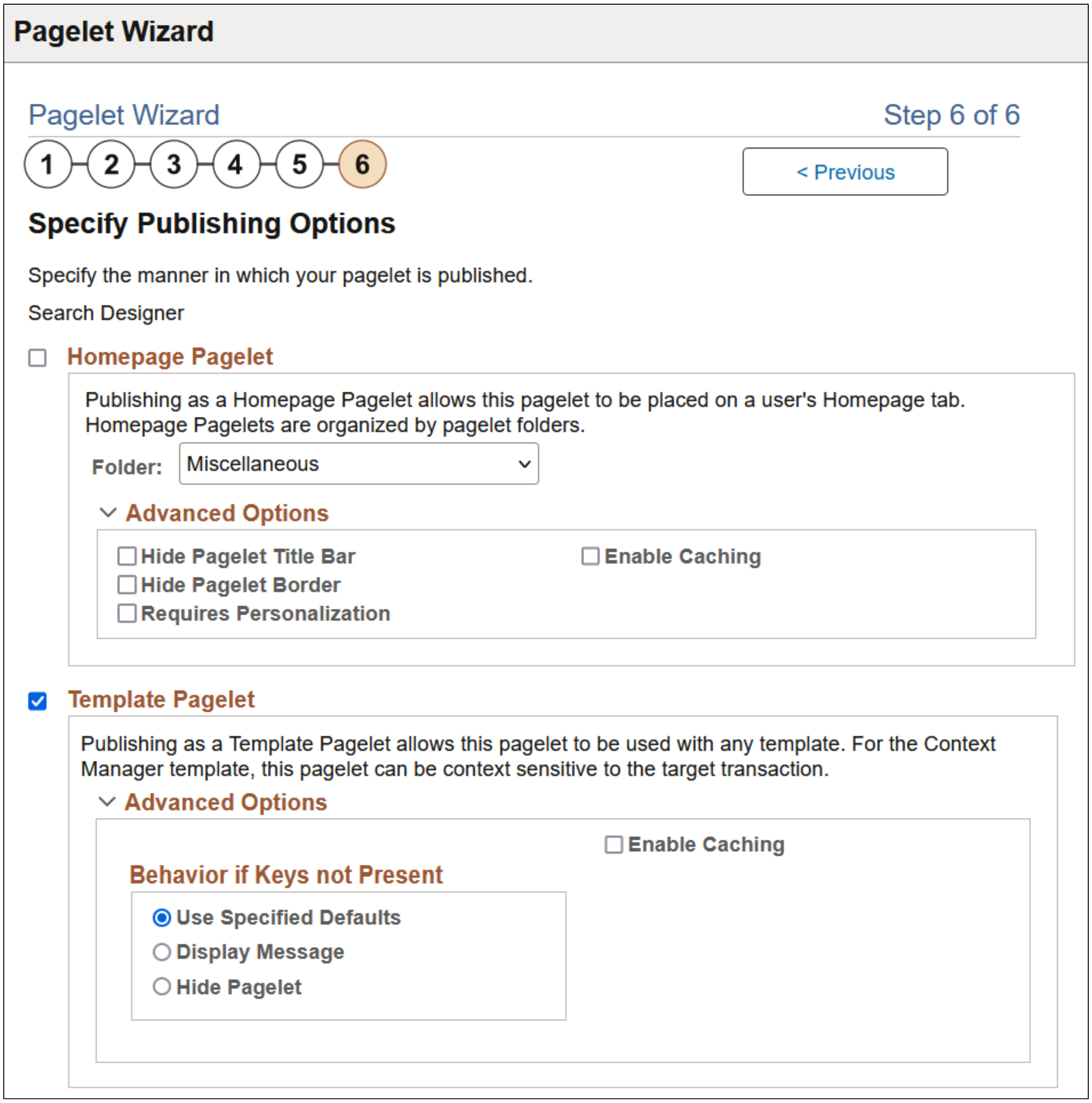
Common Elements Used in This Section

Field or Control	***Description***
Enable Caching	Select to enable caching for the homepage or template pagelet. Selecting this option displays the **Caching Options** group box.
Caching Type	*Private:* Select to enable user-based caching. *Public:* Select to enable application-based caching. *Role:* Select to enable role-based caching.
Minutes to Cache	Enter the number of minutes that the pagelet remains available in the cache. If you do not enter a value, a default value of 30 minutes is used.

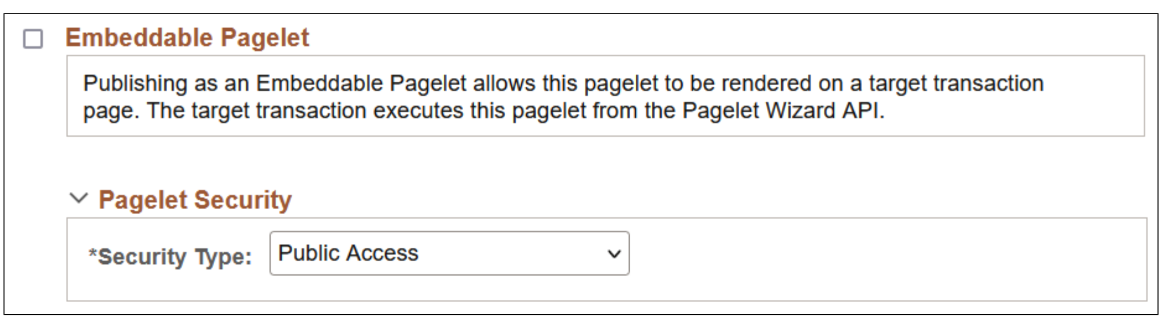
Specifying Pagelet Publication Options

Access the Specify Publishing Options page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.** On the Select Display Options page, click the Next button.)

This example illustrates the fields and controls on the Specify Publishing Options page. You can find definitions for the fields and controls later on this page.



This example illustrates additional fields and controls on the Specify Publishing Options page. You can find definitions for the fields and controls later on this page.



When you access a pagelet definition on this page, the pagelet registration options (**Homepage Pagelet, Template Pagelet, Embeddable Pagelet,**) are reflected only as they exist in the portal that you are currently signed in to. Select the registration option to publish the pagelet to the portal where you are currently signed in.

For example, if you access ABC123 pagelet, the **Homepage Pagelet** option is selected only if the portal where you are currently signed in contains a content reference for the ABC123 pagelet. If no content reference for this pagelet is in the current portal, even if a content reference for this pagelet is in another portal on the same database, the Homepage Pagelet option will be deselected. Select the **Homepage Pagelet** option to publish the pagelet to the current portal.

Homepage Pagelet

Select to publish the pagelet as a homepage pagelet that can be added to a WorkCenter page, homepage or dashboard page.

Field or Control	***Description***
Folder	Select the portal registry folder in which you want to create the pagelet content reference. The pagelet also appears in this pagelet category on the Personalize Content page, which administrators and users access to manage pagelets on homepages and dashboards.

Advanced Options

Field or Control	***Description***
Hide Pagelet Title Bar	Select to display the pagelet without a title bar.
Hide Pagelet Border	Select to display the pagelet without a border.
Requires Personalization	Select to specify that the pagelet requires end-user personalization. The pagelet displays a personalization button that accesses a personalization page. With this option selected, you are requiring that the end user access the personalization page to personalize and specify data source parameter values before it displays data. A message notifying the user of the personalization requirement appears on the pagelet. If this option is deselected, the pagelet displays data using the default data source parameter values provided.
More options	After a pagelet has been published as a homepage pagelet, the **More options** link is displayed. Click the link to access the Pagelet Branding page, on which you can set additional pagelet branding attributes. See Maintaining Pagelet Branding Attributes.

Template Pagelet

Select to publish the pagelet as a pagelet that can be added to any template. Template pagelets can extract and leverage keys specified by the target transaction for the template. Publishing a pagelet as a template pagelet is useful when you use the PeopleTools Context Manager or WorkCenter pages to display the template pagelet.

When a pagelet is published as a template pagelet, any user-specified parameters are automatically available as parameters that can be passed in Context Manager. For the keys to pass correctly, the key name in Context Manager (the key name on the page) must be identical to the name of the parameter in the pagelet. If the names match, then any key values passed are used by the template pagelet in place of the default values. If the names are not an exact match, the default values appear. Because users cannot customize template pagelets, user-specified parameters can show only the default values or the values that come through Context Manager key passing.

See Configuring WorkCenter Pages.

Advanced Options

Field or Control	***Description***
Behavior if Keys not Present	Select one of the following options to indicate how pagelet data is displayed if target transaction keys are not present to populate the pagelet's parameters. Options are: *Use Specified Defaults:* If target transaction keys are not present, the pagelet will display data using the defaults specified in Step 3: Specify Pagelet Data Source Parameters. *Display Message:* If target transaction keys are not present, the pagelet does not display data but instead displays a message. *Hide Pagelet:* If target transaction keys are not present, the pagelet does not appear.

Embeddable Pagelet

Select this option to make your pagelet available as an embeddable pagelet that can be generated by the Pagelet Wizard API for rendering within an HTML area of a target page or using an iScript.

Pagelet Security

If you change any pagelet security options for a registered pagelet, you must reregister the pagelet.

Field or Control	*Description*
Security Type	Select the type of security that you want to assign to the pagelet. Options are: *Public Access:* Select to designate that all users can access the pagelet. *Select Security Access:* Select to assign role-based or permission list-based security to the pagelet. The **Selected Security** scroll area appears.
Author Access	Select to provide the pagelet author with access to the pagelet regardless of any security restrictions assigned to the pagelet. This access is granted based on the author's user ID.
Update Security	This button appears when Pagelet Wizard determines that an inconsistency exists between the security defined for the pagelet on this page and the security on one or more of the content references defined for the pagelet. When you click this button, the content reference security is updated for all of the content references for Homepage Pagelet, Template Pagelet, and WSRP Portlet options.

Selected Security

Field or Control	*Description*
Type	Select the type of security to assign to the pagelet. Specified security is applied to the content reference entries when the pagelets are registered. Options are: *Permission List:* Select to assign permission list-based security to the pagelet. Select a permission list in the **Name** field. *Role:* Select to assign role-based security to the pagelet. Select a role in the **Name** field.
Name	Enter the name of the permission list or role to which to authorize access to the pagelet.

Buttons

Field or Control	*Description*
Previous	Click to move backwards one step in the pagelet creation process in Pagelet Wizard.
Finish	Click to complete the creation of the pagelet. The Pagelet Creation Confirmed page appears. After you click the **Finish** button for a pagelet, the button no longer appears when you access the pagelet definition in Pagelet Wizard. Once a pagelet has been created, you can apply any modifications to the pagelet definition by clicking the **Save** button.

Using Pagelet Wizard Post-Transformation Processing

This section provides an overview of Pagelet Wizard post-transformation processing and discusses how to:

- Insert message catalog text.
- Format Date, Time, DateTime, Currency, and Number elements.
- Disable WSRP style substitution.
- Insert a content-reference-based link.
- Insert a content reference URL as text.
- Generate charts.
- Insert pagelet headers and footers in specific locations.

Understanding Pagelet Wizard Post-Transformation Processing

Pagelet Wizard enables you to use delivered Pagelet Wizard builder transformation tasks in custom and passthru transformations.

Listed in the order in which they are run in post-transformation processing, the following transformation tasks are not automatically generated for custom and passthru transformations, but they can be added by means of embedded tags:

1. Insert Message Catalog text.
2. Format Date, Time, DateTime, Currency, and Number.
3. Insert content-reference-based links.

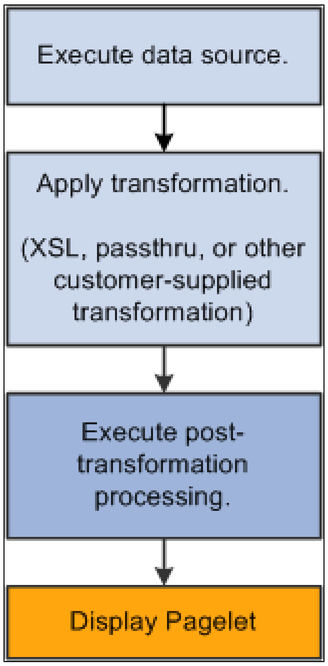
4. Insert content reference URLs as text.

5. Generate charts.

6. Insert pagelet headers and footers in specific locations.

These tasks are handled by the Post Transformation step, which is built into Pagelet Wizard.

The following diagram illustrates the point at which the Post Transformation step runs in the process flow for displaying a Pagelet Wizard pagelet.



The Post Transformation step runs after the completion of the Transformation step. Following the Post Transformation step, the Pagelet Wizard pagelet appears for the user. Pagelet Wizard uses embedded tags to identify the type of Post Transformation processing that occurs.

Tags that perform these post-transformation tasks can be placed in the HTML for HTML pagelets using the passthru transformation, or in the XSL for pagelets using a custom transformation. For all other transformations—such as chart, list, table, and so forth—Pagelet Wizard automatically generates the tags when necessary. These tags are primarily meant to be used by pagelet designers who want to create custom pagelets using functionality similar to what our delivered Pagelet Wizard builders provide.

Inserting Message Catalog Text

Use the following syntax to insert the text of a message catalog entry. Variables that are user-supplied are italicized.

Note: This processing step occurs only if the output of the transformation step is XHTML-compliant.

See XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	***XHTML Tag***	***HTML After Processing***
Message Catalog Text	<PSMSGTEXT MSGSET="*message set*" MSGNBR="*message number*" DEFAULT="*defaulttext*"/>	Message text.
Message Catalog Explain Text	<PSMSGEXPLAINTEXT MSGSET="*message set*" MSGNBR="*message number*" DEFAULT="*defaulttext*"/>	Message explanation text.

Formatting Date, Time, DateTime, Currency, and Number Elements

Use the following syntax to format Date, Time, DateTime, Currency, or Number elements. Variables that are user-supplied are set in italics.

Note: This processing step occurs only if the output of the Transformation step is XHTML-compliant.

See XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	***XHTML Tag***	***HTML After Processing***
Date	<PSFORMAT TYPE="DATE" >DateToFormat</PSFORMAT> **Note:** FORMAT has the same syntax as the DateTimeToLocalizedString() PeopleCode Function.	Formatted Date. Language-specific Formatted Date.
Time	<PSFORMAT TYPE="TIME" [SOURCETZ="*Source Time Zone*"][DESTTZ="*Destination Time Zone*"][FORMAT="*format string*"]>TimeToFormat</PSFORMAT> **Note:** SOURCETZ and DESTTZ have the same format as the DateTimeToTimeZone() PeopleCode function. FORMAT has the same syntax as the DateTimeToLocalizedString() PeopleCode Function.	Formatted Time.
DateTime	<PSFORMAT TYPE="DATETIME" [SOURCETZ="*Source Time Zone*"][DESTTZ="*Destination Time Zone*"] [FORMAT="*format string*"]>DateTimeToFormat</PSFORMAT> **Note:** SOURCETZ and DESTTZ have the same format as the DateTimeToTimeZone() PeopleCode function. FORMAT has the same syntax as the DateTimeToLocalizedString() PeopleCode Function.	Formatted DateTime.

Item to Insert	***XHTML Tag***	***HTML After Processing***
Number	<PSFORMAT TYPE="NUMBER" FORMAT="*format string*" [WIDTH="*width*"] [PRECISION="*precision*"]> NumberToFormat</PSFORMAT> **Note:** FORMAT, WIDTH, and PRECISION have the same format as the NumberToDisplayString() PeopleCode Function.	Formatted Number.
Currency	<PSFORMAT TYPE="CURRENCY" FORMAT="*format string*" [WIDTH="*width*"] [PRECISION="*precision*"] [CURRENCY_CODE="*CODE* "] [CURRENCY_ DECORATION="*SYMBOL*/*CODE*/*BOTH* "]> NumberToFormat</PSFORMAT> **Note:** FORMAT, WIDTH, and PRECISION have the same format as the NumberToDisplayString() PeopleCode Function. CURRENCY_CODE can be one of the supported PeopleSoft currency codes, or if left blank, the default is the user's currency code. CURRENCY_DECORATION can have the value of *SYMBOL* (currency symbol appears before the number), *CODE* (currency code appears after the number), or *BOTH* (the default).	Formatted Number with currency symbol and/or currency code.

Inserting Content Reference-Based Links

Use the following syntax to insert content-reference-based links into a Pagelet Wizard pagelet. Variables that are user-supplied are set in italics.

Note: This processing step occurs only if the output of the Transformation step is XHTML compliant.

See XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	***XHTML Tag***	***HTML After Processing***
Content Reference Link	<PSREGISTEREDLINK PORTAL="*portal name*" CREF="*content reference name*" target="*target*" APPEND="*append*">Text of Link</ PSREGISTEREDLINK>	<A HREF="http…URL of content reference…[?/&]append" target="target">Text of Link</A>

Inserting a Content Reference URL as Text

Use the following syntax to insert a content reference URL as text into a Pagelet Wizard pagelet. Variables that are user-supplied are set in italics.

Note: This processing step occurs only if the output of the Transformation step is XHTML-compliant.

See XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

Item to Insert	***XHTML Tag***	***HTML After Processing***
Content Reference URL	<PSREGISTEREDURL PORTAL="*portal name*" CREF="*content reference name*" APPEND="*append*"/>	http…URL of content reference…[?/&]append

Generating Charts

Embedded charts are generated in the Post Transformation step.

For details about chart generation and its syntax, refer to the ChartXSLBuilder included in the EOPPB_PAGELET application class. You can access the ChartXSLBuilder application class code using PeopleSoft Application Designer.

Inserting Pagelet Headers and Footers in Specific Locations

Pagelet Wizard automatically inserts a pagelet's header in front of the output of the Transformation step, and it inserts the pagelet footer after the output of the Transformation step.

To alter this behavior, you must use special HTML comment tags. Here is the syntax of these tags:

Item to Insert	***HTML Tag***	***XSL Equivalent***
Header	<!--PWHEADER-->	<xsl:comment>PWHEADER</xsl:comment>
Footer	<!--PWFOOTER-->	<xsl:comment>PWFOOTER</xsl:comment>

Administering Pagelet Wizard Pagelets

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

- Delete and copy Pagelet Wizard pagelets.
- Generate export and import scripts for Pagelet Wizard pagelets.
- Review Pagelet Wizard pagelet details.

Common Elements Used in This Section

This section discusses common elements used in this section.

Filter Pagelets

If appropriate, use the options in the **Filter Pagelets** group box to narrow the range of pagelets displayed in the **Pagelets** scroll area.

Field or Control	*Description*
Data Type	Select the data type of the pagelet that you want to appear in the **Pagelets** scroll area. Options are: • *Activity Guide.* • *All Data Types.* • *Free Text.* • *HTML.* • *IB Connector.* • *Integration Broker.* • *Navigation Collection.* • *OBIEE Report.* • *PS Query.* • *Pivot Grid.* • *Rowset.* • *Search Record.* • *URL.*
Published Type	Select the pagelet type assigned to the pagelet that you want to appear in the **Pagelets** scroll area. Options are: • *All Pagelets.* • *Embeddable Pagelet.* • *Homepage Pagelet.* • *Non-published Pagelet.* • *Template Pagelet.* • *WSRP.*

Field or Control	***Description***
Portal Name	The portal name that you select here does not filter results, but rather serves to inform you which pagelets displayed in the **Pagelet** scroll area have been published in the selected portal. Pagelets are automatically published to the portal where the pagelet author is signed in when creating the pagelet.
Pagelet Keyword	Enter keywords to narrow the range of pagelets that appear in the **Pagelets** scroll area. Pagelet elements that are used as keywords include pagelet ID, title, and description.
Search	Click to perform a pagelet search using the criteria entered in the **Filter Pagelets** group box.
Reset	Click to clear the **Pagelet Keyword** field.

Pagelets

The **Pagelets** scroll area provides the following pagelet information.

Field or Control	***Description***
Pagelet ID	Displays the pagelet ID. Click to access the pagelet definition in Pagelet Wizard.
Pagelet Title	Displays the pagelet title.
Homepage	If selected, indicates that the pagelet has been published as a homepage page.
Template	If selected, indicates that the pagelet has been published as a template pagelet.
WSRP	If selected, indicates that the pagelet has been published as a WSRP portlet.
Embeddable	If selected, indicates that the pagelet has been published as an embeddable pagelet.

Deleting and Copying Pagelet Wizard Pagelets

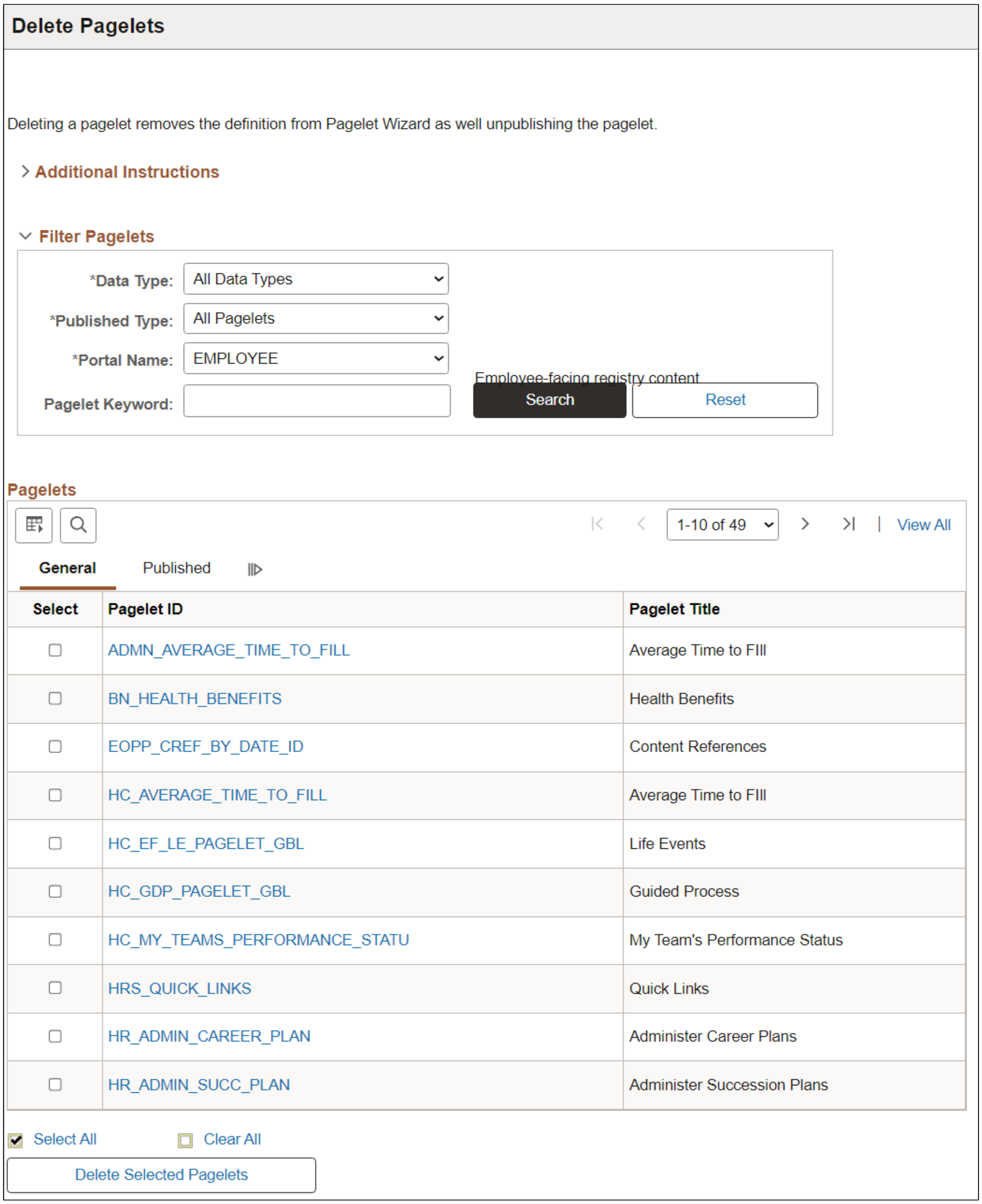
This section discusses how to:

- Delete pagelets.
- Copy pagelets.

Deleting Pagelets

Access the Delete Pagelets page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Delete Pagelets**.)

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



Deleting a pagelet removes the pagelet definition from Pagelet Wizard. In addition, associated metadata and registry entries are deleted from the database and portal registry.

Field or Control	***Description***
Select	Select the pagelets to delete.
Delete Selected Pagelets	Click to delete selected pagelets. The system prompts you to confirm the deletion.

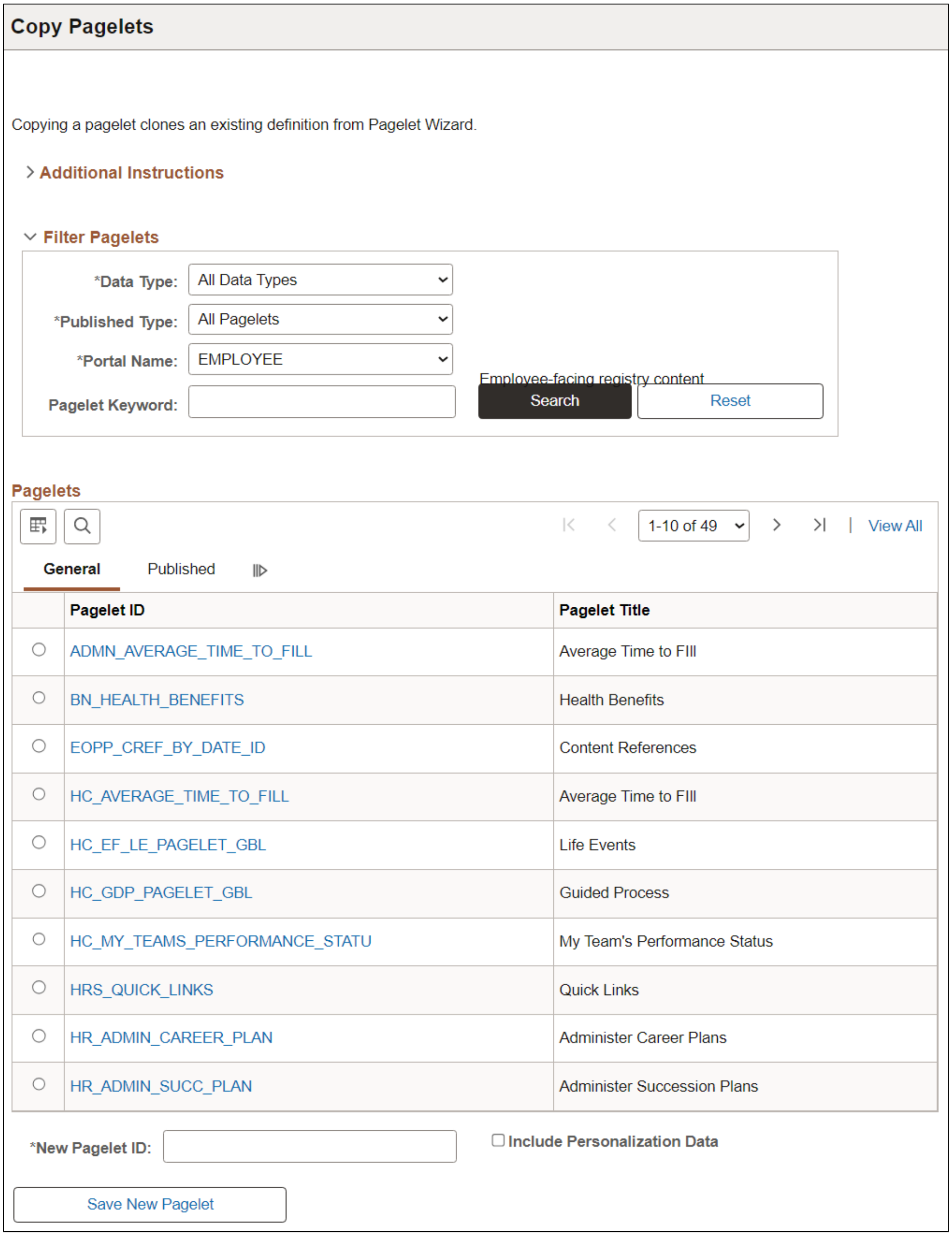
The Published tab shows information about how the pagelet has been published.

Field or Control	***Description***
Homepage	If selected, indicates that the pagelet has been published as a homepage page.
Template	If selected, indicates that the pagelet has been published as a template pagelet.
WSRP	If selected, indicates that the pagelet has been published as a WSRP portlet.
Embeddable	If selected, indicates that the pagelet has been published as an embeddable pagelet.

Copying Pagelets

Access the Copy Pagelets page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Copy Pagelets**.)

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



The pagelet copy inherits all aspects of the original pagelet, with the exception of registration information.

Field or Control	***Description***
Select	Select the pagelet to copy.

Field or Control	***Description***
New Pagelet ID	Enter a unique pagelet ID for the new pagelet that will be copied from the selected existing pagelet. **Note:** Pagelet IDs can contain only the characters A–Z, a–z, and _. All other characters put into a pagelet ID will be removed by the system. If all characters are removed from the user-entered pagelet ID, then the Registry Object Prefix will be used instead. If, after removing illegal characters from the pagelet ID, the name is found to match an existing pagelet ID, then Pagelet Wizard will append numbers to the end of the pagelet ID to make the new pagelet ID unique.
Include Personalization Data	Select to copy any end-user personalizations defined for the existing pagelet along with the new pagelet.
Save New Pagelet	Click to create the cloned pagelet with the new pagelet ID. Once you create the copied pagelet, you must publish it before it can be used as a homepage pagelet or a template pagelet. You can access the new pagelet in Pagelet Wizard and publish the pagelet during Step 6 on the Specify Publishing Options page.

The Published tab shows information about how the pagelet has been published.

Field or Control	***Description***
Homepage	If selected, indicates that the pagelet has been published as a homepage page.
Template	If selected, indicates that the pagelet has been published as a template pagelet.
WSRP	If selected, indicates that the pagelet has been published as a WSRP portlet.
Embeddable	If selected, indicates that the pagelet has been published as an embeddable pagelet.

Generating Export and Import Scripts for Pagelet Wizard Pagelets

This section provides an overview of export and import scripts for Pagelet Wizard pagelets and discusses how to:

- Generate export and import pagelet scripts.
- Generate export and import scripts for Pagelet Wizard setup data.

Understanding Export and Import Scripts for Pagelet Wizard Pagelets

You can use these export and import scripts to copy Pagelet Wizard pagelet definitions from one database to another.

When you export or import a pagelet, the imported pagelet is not automatically published. You must access the pagelet definition in Pagelet Wizard in the target portal in the target database and publish the pagelet on the Specify Publishing Options page in Step 6.

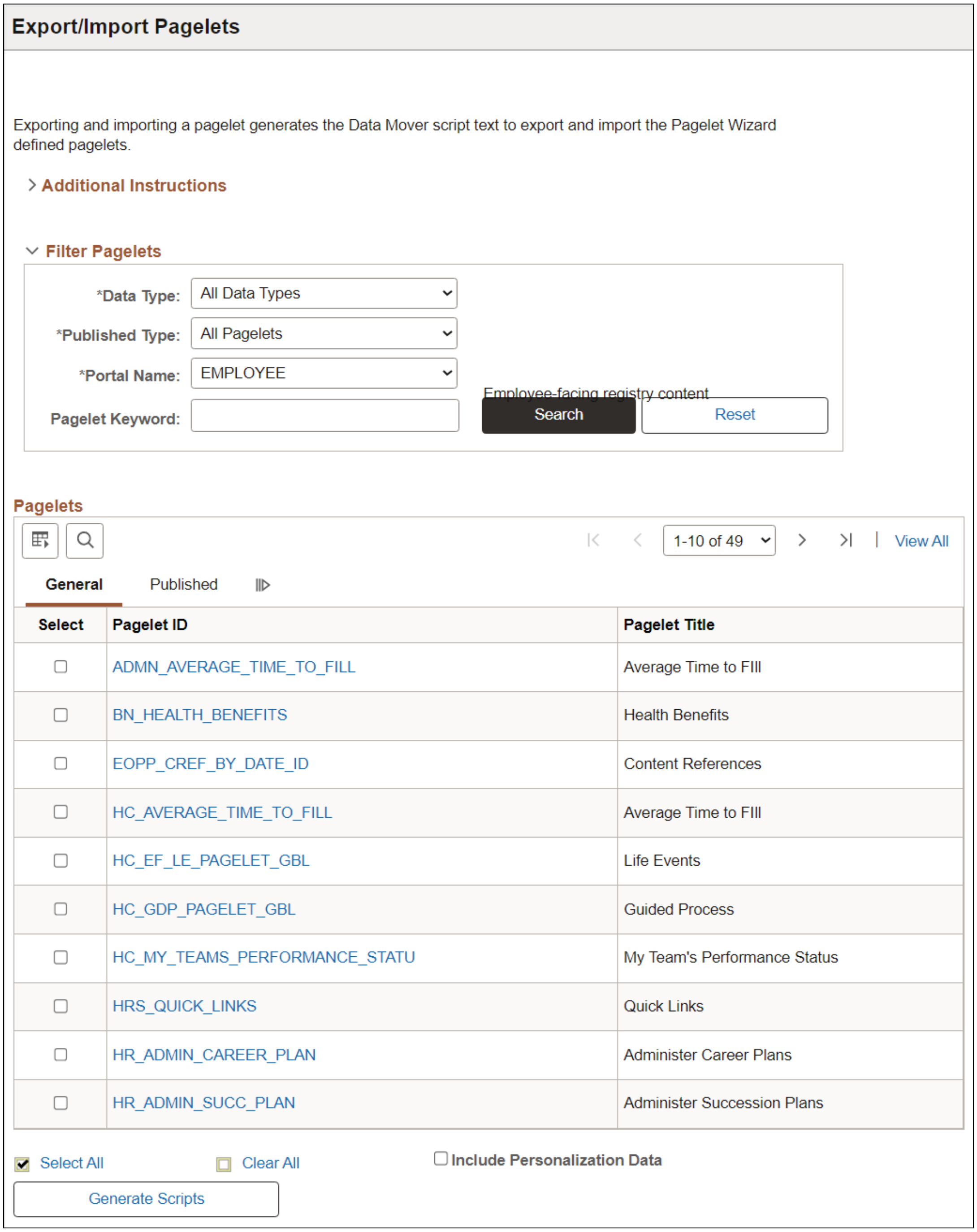
Alternatively, you can run the generated export and import Data Mover scripts (.DMS scripts) and copy the pagelet content reference by placing it into a PeopleSoft Application Designer project.

Note: These export and import scripts do not include the ability to export or import Pagelet Wizard setup data associated with the pagelet definitions being exported or imported.

Generating Export and Import Pagelet Scripts

Access the Export/Import Pagelets page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Export/ Import Pagelets**.)

This example illustrates the fields and controls on the Export/Import Pagelets page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Select	Select the pagelets for which you want to generate export and import scripts.
Include Personalize Data	Select to include any end-user personalizations defined for the selected pagelets in the export and import.

Field or Control	***Description***
Generate Scripts	Click to generate export and import scripts for the selected pagelets. The scripts appear on the Data Mover Scripts page. Copy the generated scripts into PeopleSoft Data Mover to export the selected pagelet definitions from a source database and import them into a target database. See "Understanding PeopleSoft Data Mover" (Lifecycle Management Guide).

Note: Other Pagelet Wizard-related Data Mover scripts are delivered with PeopleSoft Enterprise Components for users who need more customizable scripting for pagelet migration between databases. You can find these scripts delivered in the PS_HOME scripts directory.

Generating Export and Import Scripts for Pagelet Wizard Setup Data

PeopleTools Portal Technologies delivers the following Data Mover scripts to export and import Pagelet Wizard setup data:

- EOPPB_SETUP_DEFN_IMP.DMS
- EOPPB_SETUP_DEFN_EXP.DMS

Use these scripts to copy the following types of Pagelet Wizard setup data from one database to another, for example, from a development database to a production database:

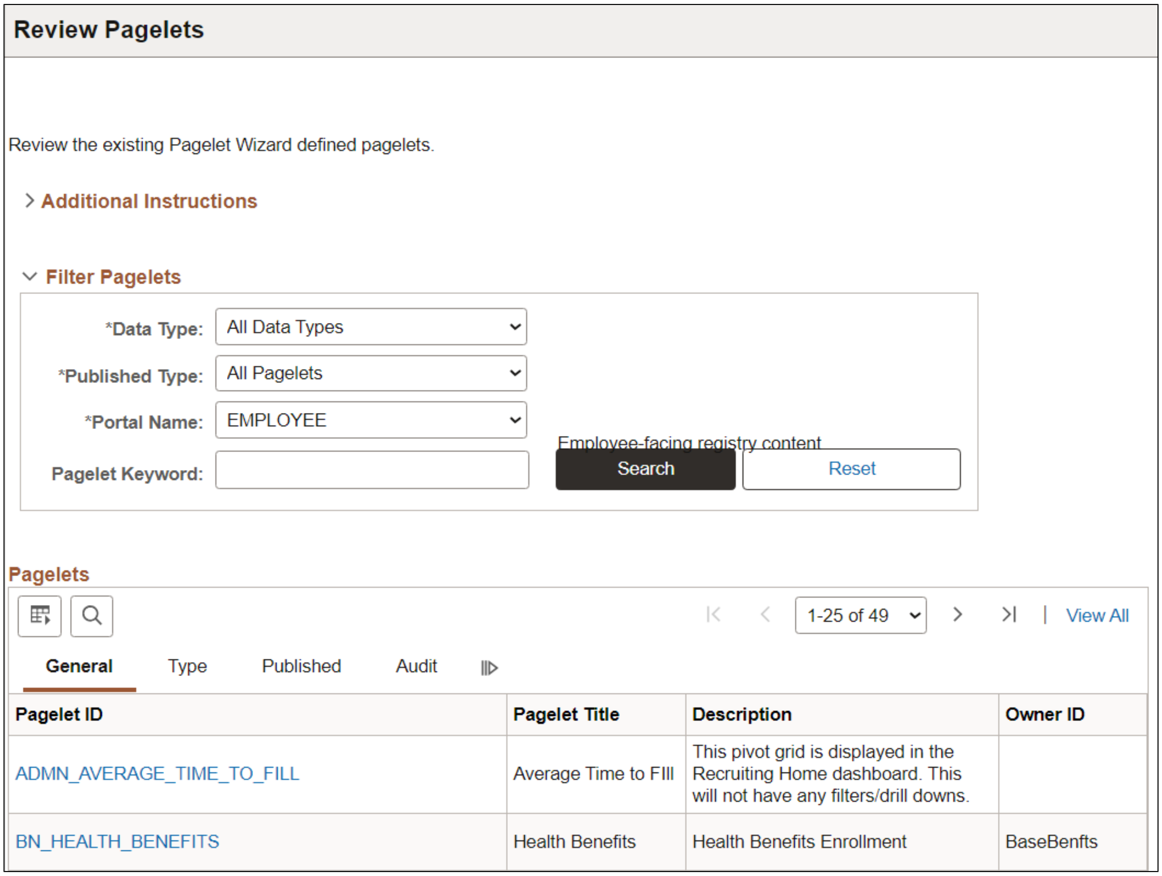
- Footers.
- Headers.
- Categories.
- XSL prototype templates.

You can find these scripts delivered in the PS_HOME scripts directory.

Reviewing Pagelet Wizard Pagelet Details

Access the Review Pagelets page. (Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Review Pagelets**.)

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



On any of the tabs for this component, you can click the **Pagelet ID** link to open Pagelet Wizard and the definition of the pagelet.

General Pagelet Information

Click the **Pagelet ID** link to open Pagelet Wizard and the definition of the pagelet.

Pagelet Type Information

Field or Control	***Description***
Category	Displays the category ID from step 1 of Pagelet Wizard.
Data Type	Displays the pagelet data type.
Display Format ID	Displays the format ID of the pagelet.

Pagelet Publishing Information

Field or Control	***Description***
Homepage	If selected, indicates that the pagelet has been published as a homepage page.
Template	If selected, indicates that the pagelet has been published as a template pagelet.
WSRP	If selected, indicates that the pagelet has been published as a WSRP portlet.
Embeddable	If selected, indicates that the pagelet has been published as an embeddable pagelet.
Cache	Displays a check box if caching is enabled for this pagelet.
Time	Displays the time the pagelet is held in the cache.

Pagelet Audit Information

This tab includes the timestamp and user ID of the person who created the pagelet as well as information about any updates to the pagelet.

Chapter 16

Developing and Registering Pagelets

Understanding Pagelets

This section discusses:

- Pagelets.
- Pagelet development.
- Pagelet extensions.
- Pagelet personalizations.

Prerequisites

This document provides a practical guide for technical users and programmers who develop pagelets (or portlets) for operation within a PeopleSoft portal. To take full advantage of the information covered in this document, Oracle recommends that you have a basic understanding of the PeopleSoft Pure Internet Architecture and portal technology. Extensive information can be found in the PeopleBooks documentation. Many of the fundamental concepts related to the portal are also discussed in the following PeopleTools document: *PeopleTools: System and Server Administration*.

Pagelets

Pagelets are small pages that provide snapshots of useful, relevant and typically read-only content from PeopleSoft or non-PeopleSoft applications or websites. In other portal products, pagelets are sometimes referred to as portlets, gadgets, widgets, or modules.

Homepage pagelets are designed to appear on tabbed homepages or dashboard pages. Homepage pagelets typically present general, high-level information; however, many are designed to provide the user with easy access to more detailed information either in another pagelet on the homepage or a target transaction page. Users can select the homepage pagelets that appear on his or her homepages by way of the **Personalize Content** link that is in the upper left corner of the homepage.

Template pagelets are small, narrow-format pagelets that are designed to appear in a template iframe, much like the Menu pagelet. Usually, you incorporate template pagelets into a portal template when you create and configure the template. You also use template pagelets when you configure the task panel of WorkCenter pages or context-sensitive pagelets that use Context Manager. You should design template pagelets so that they display data that is related in some way to the target transaction.

Pagelets follow a basic set of rules so that they can be displayed properly on PeopleTools portal homepages, dashboard pages, related content frames, or in the pagelet area of WorkCenter pages. Homepage pagelet size should correspond to the homepage layout specified by the user. A user can specify either a two-column layout of one narrow pagelet and one wide pagelet, or a three-column layout of three narrow pagelets.

Template pagelets should use the narrow column format of a pagelet. Banner pagelets span the width of the homepage or dashboard page. Any pagelet that you design must conform to the dimensions of the narrow column and optionally, of the wide column. Column width is the only restriction on pagelet dimensions, although design principles suggest keeping pagelets as compact as possible.

The homepage pagelets that appear on the portal homepage depend on several factors, including the type of portal that you implement, the permissions that you grant to the user, and the manner in which the user personalizes his or her homepage.

PeopleSoft pagelets are URL-accessible HTML documents that are designed to be size-appropriate for display on a homepage. Pagelets can also originate from remote, non-PeopleSoft systems such as static HTML files or HTML that is generated dynamically from a variety of technologies.

Pagelet Development

You can use PeopleTools to develop pagelets as smaller-than-normal pages, just as you would any other PeopleSoft transaction page. PeopleSoft portals are delivered with a suite of pagelets, primarily built using PeopleTools technology. However, you can also base the design of a pagelet on PeopleCode internet scripts (iScripts). Use iScripts only when you can't accomplish the same task with a PeopleSoft transaction page, for example, if you are developing a pagelet that uses external content (content that does not originate in a PeopleSoft application). You can also create pagelets using other leading web-enabling technology such as JavaServer pages (JSP), Java servlets, Microsoft Active Server Pages (ASP), Perl scripts, and common gateway interface (CGI) programs.

A well-designed and well-developed pagelet serves these primary functions:

- Summarizes data that is relevant, personalized, and actionable. Page space is valuable; pagelets must be more than attractive.
- Provides links to more detailed information, such as content in another application.

In addition, pagelets should also:

- Aggregate data from multiple sources.
- Be simple and intuitive.
- Be sized appropriately.

  The size of pagelets corresponds to the homepage layout that a user chooses. Any pagelet that you design must first conform to the dimensions of the narrow column and optionally, of the wide column. Pagelet width is the only dimension to which you must adhere. The height of pagelets can vary; however, good design principles suggest keeping them as short as possible.
- Avoid the use of *scrollable grids.*

  Grids that you insert on pagelets are not scrollable by default. If the grid appears on a pagelet and on a standard transaction page, Oracle recommends that you use a non-scrollable grid type to ensure consistent user experiences.

Pagelet Header Icons

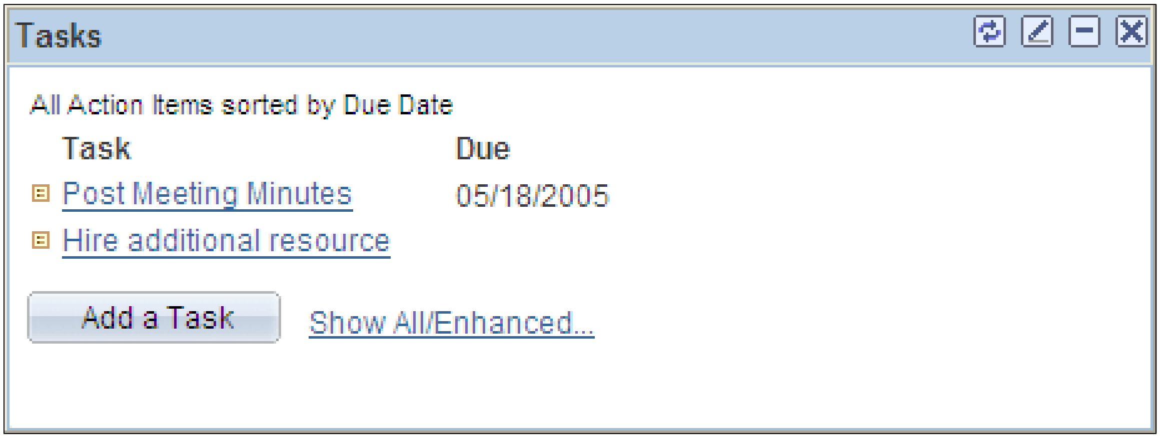
The icons that you see in the pagelet header function similarly across most pagelet types. You do not have to use all header icons on all pagelets. You can vary the icons that appear in the header based on business rules or technical constraints.

This table lists and describes the icons that you can use in pagelet headers.

Term	***Definition***
Expand	Click to show the content of the pagelet so that more than just the pagelet header is visible. When you click this icon, in its place you will see the **Hide** icon.
Help	Click to view pagelet help. When you click this icon, any documentation that you configure appears. Not all pagelets provide documentation.
Personalize	Click to personalize the pagelet. When you click this icon, a page appears so that you can change any pagelet settings that you are allowed to configure. The options that you can personalize vary among users and pagelets, and not all pagelets have options that you can personalize.
Minimize	Click to hide the body of the pagelet so that only the header is visible. When you click this icon, in its place you will see the **Show** icon.
Refresh	Click to force an application server trip so that the pagelet data and settings are current.
Remove	Click to remove the pagelet from the page. You will be prompted to confirm the action and must accept or dismiss the confirmation.

Pagelet Example

This example demonstrates many characteristics of a well-designed pagelet and the pagelet header icons in the Tasks pagelet.



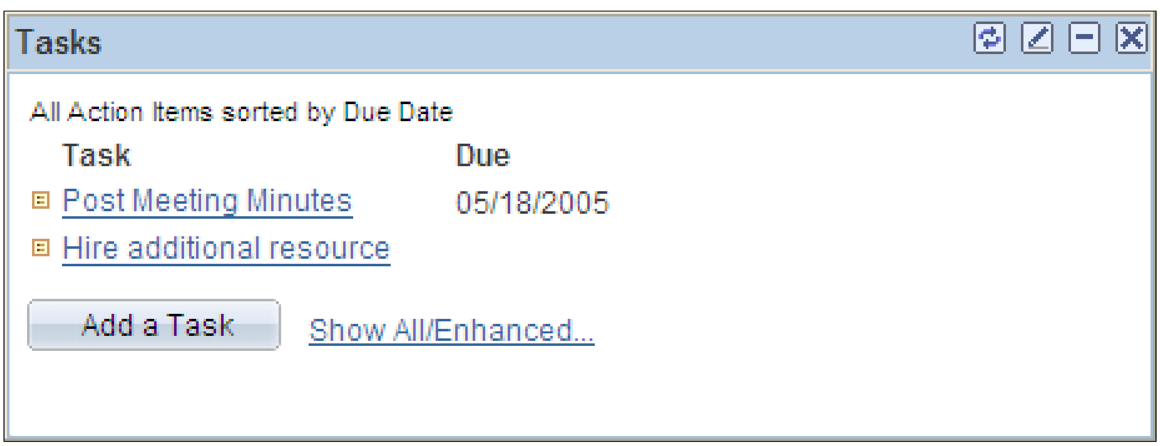
Pagelet Extensions

Some pagelets use pagelet extensions, which are the supporting or supplementary components for a pagelet. They are not displayed on your homepage; you access pagelet extensions by associating them with links and buttons on pagelets. Pagelet extensions are optional. They are a means of providing additional information or editing capability that is too cumbersome to display on the pagelet.

You could configure a homepage pagelet so that each item is a link to the worklist item in another database, or to the personal task defined in the portal, or to a variety of resources.

The Tasks pagelet is a homepage pagelet that is designed to display a short list of relevant tasks. If dozens of tasks were open, which is likely over time, the pagelet would grow to be too long and possibly crowd out relevant, actionable data on other pagelets.

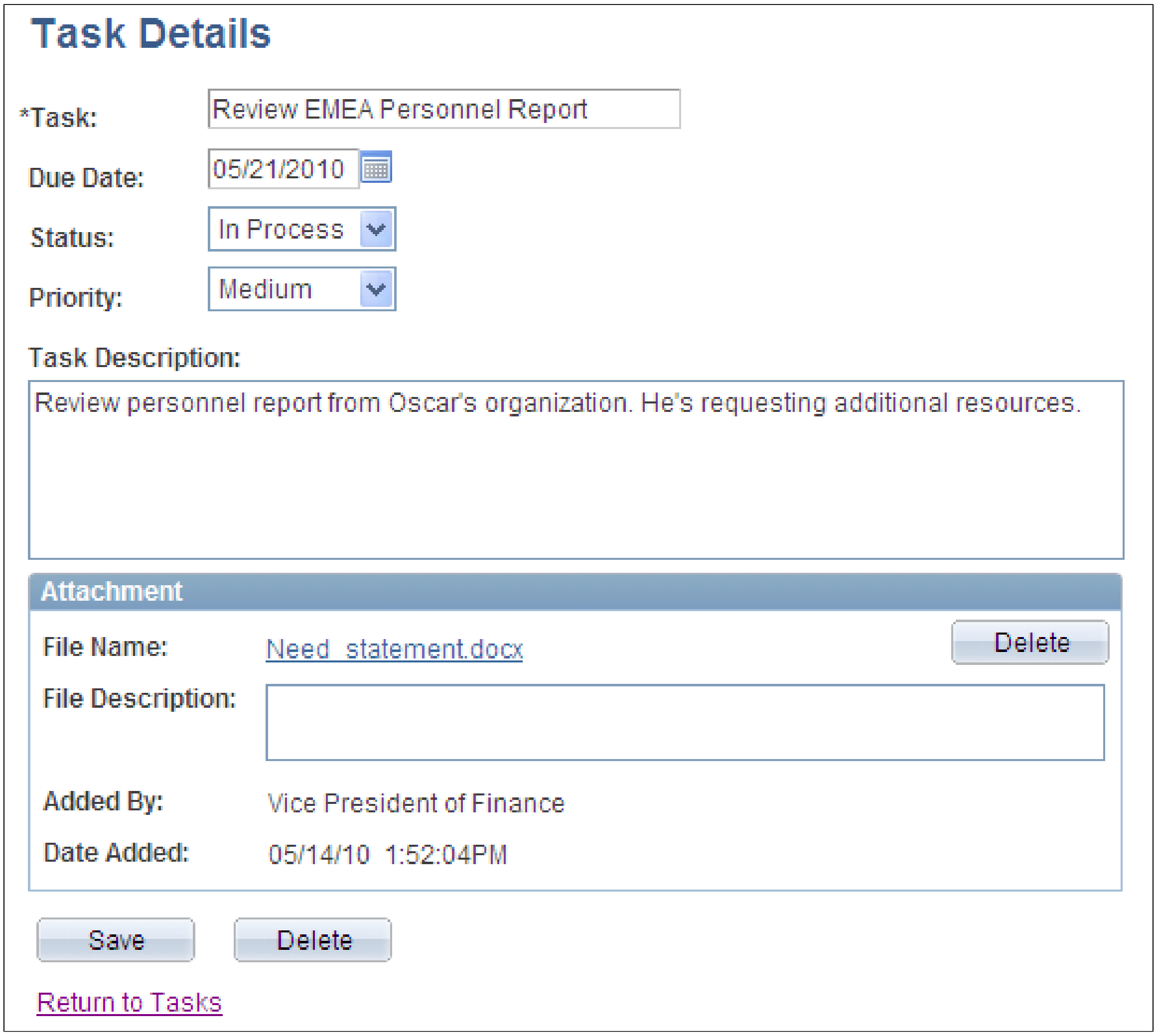
This example illustrates that the Tasks pagelet is configured with active links and buttons that access pagelet extensions, which enhance the functionality of the pagelet.



Here you see two extension pages that you access from the Tasks pagelet:

- The Task Details page.

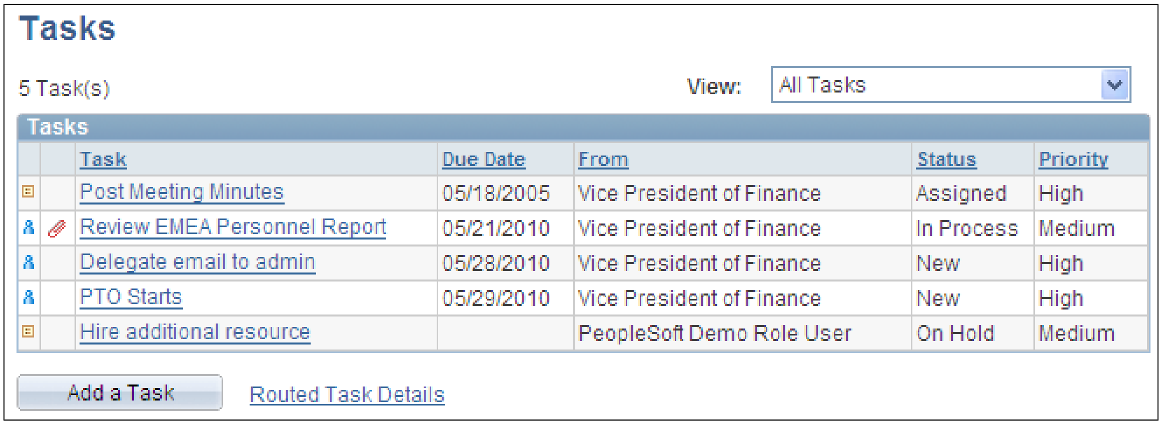
Click a link for a task name to access the Task Details page, which you use to view or modify personal task information. The Task Details page enhances the pagelet by showing more detail and enabling you to perform further actions, such as attaching a file or deleting the task.



- The Tasks page.

Click the **Show All/Enhanced** link to access the Tasks page, which provides the user with more options such as the ability to display all or certain types of tasks. Because it is a standard page, the extension page can use a grid to retrieve and display a longer list of tasks. In addition, the extension page is also better suited to display more details because it is wider than the pagelet and can show more columns

The following example displays more details on the Tasks page.



Pagelet Personalizations

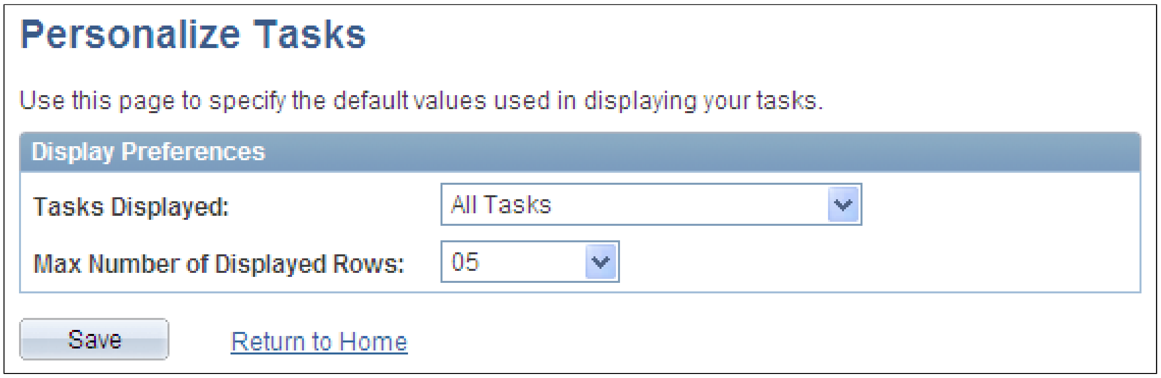
Like homepages, pagelets can be personalized in different ways. The data displayed can be filtered automatically to show only data that is relevant to a particular user. The user can also explicitly personalize the data shown by setting options on a personalization page.

Personalization pages are another type of pagelet extension. However, personalization page pagelet extensions are different from previously discussed pagelet extensions in the following ways:

- You do not click a button or link on the pagelet to access personalization pages. Instead, clicking the **Personalize** icon on the pagelet header accesses a personalization page where you can define user options that are specific to that pagelet.
- Setting values on the personalization page will change the way data appears on the pagelet for the user. Detail pages usually display data in read-only mode or allow you to change application data.

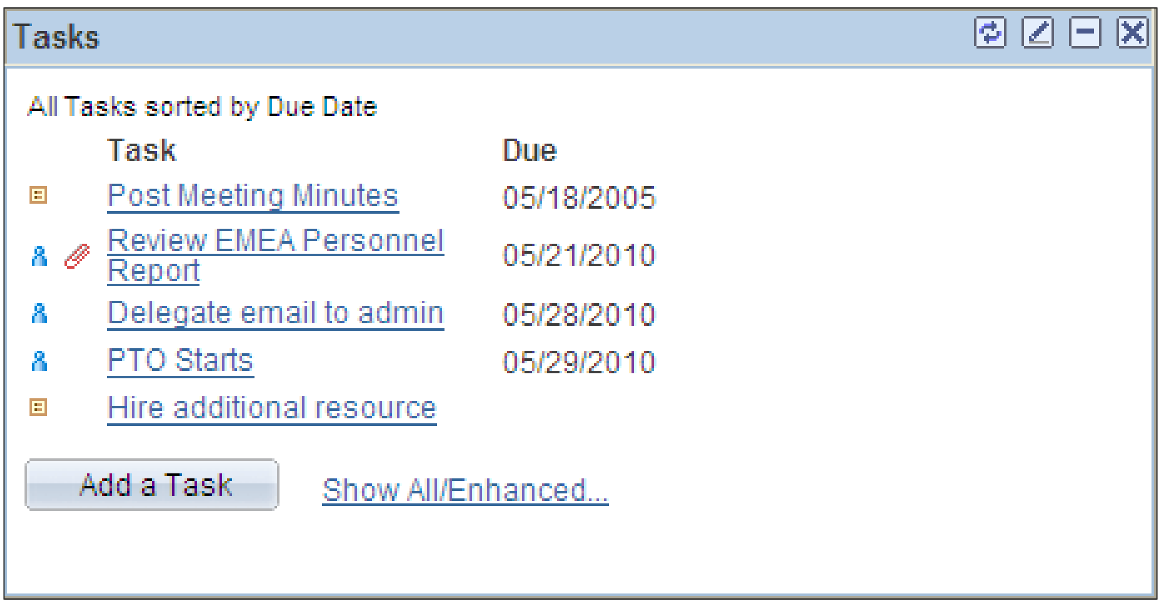
Note: Any text that the pagelet developer enters during Step 3 in Pagelet Wizard appears as instructions on the Personalize page.

The Personalize Tasks page provides an example of a pagelet personalization page.



You can use the Personalize Tasks personalization page for the Tasks pagelet to change the type and number of tasks shown on the pagelet. Clicking **Save** stores these values and returns you to the homepage. The Tasks pagelet now reflects these changes.

This example illustrates how personalization changes what is displayed in the Task pagelet.



Understanding PeopleSoft Portal Architecture

This section discusses:

- PeopleSoft portal architecture.
- Client access.
- The web server.
- The application server.
- The database server.

- The portal registry.
- Portal components.
- Page assembly.

PeopleSoft Portal Architecture

When you develop pagelets, you need to understand the overall architecture of the PeopleSoft portal, as well as the PeopleSoft Pure Internet Architecture. This will enable you to integrate your pagelets in the most efficient manner.

The PeopleSoft Pure Internet Architecture is the server-centric architecture used to deploy PeopleSoft internet applications to end users who access the applications through a web browser. This next generation architecture leverages a number of internet technologies and concepts to deliver simple, ubiquitous access to PeopleSoft applications and to enable the open flow of information between systems.

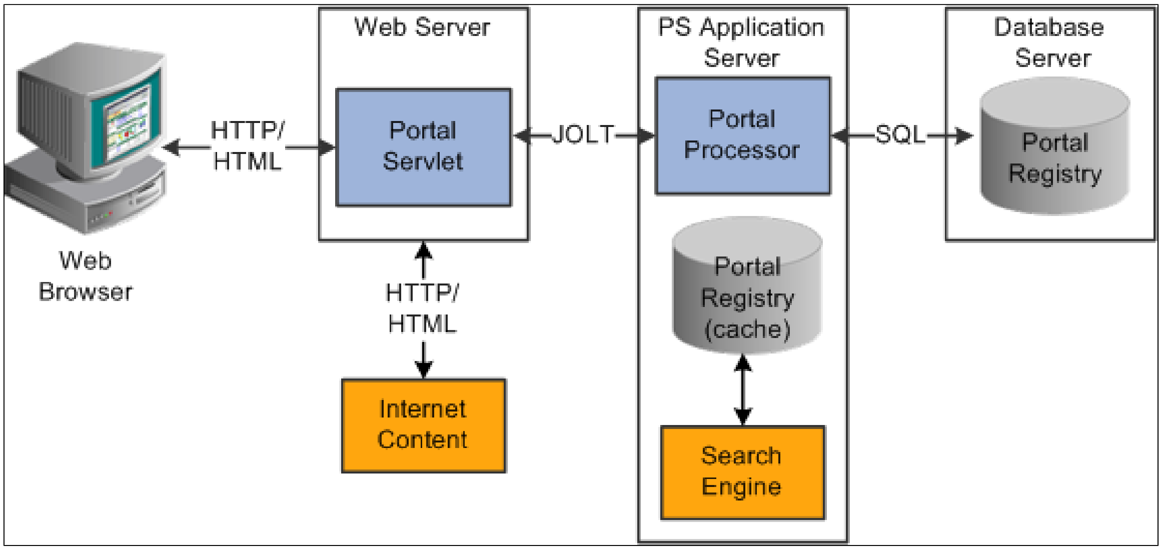
Using PeopleSoft Pure Internet Architecture as the foundation, you can provide a wide range of end users with access to PeopleSoft applications over the web, as well as more easily integrate your PeopleSoft applications with existing internal systems and external trading partner systems.

The following diagram highlights the systems and the primary portal components of the PeopleSoft Pure Internet Architecture:

- The web browser communicates with the web server over HTTP/HTTPS.
- The portal servlet runs on the web server. The web server communicates with the application server over Jolt. The web server also communicates with other internet content providers over HTTP/HTTPS.
- The portal processor runs on the application server. The search engine retrieves index entries from the portal registry cache on the application server. The application server communicates with the database server through SQL.
- The portal registry data resides on the database server.

Warning! Always refer to the certifications section of My Oracle Support for supported configurations. Versions and products supported can change frequently.

The following diagram highlights these primary portal components of the PeopleSoft Pure Internet Architecture: the portal servlet (web server); the portal processor, portal registry cache, and search engine (application server); and the portal registry (database server).



Client Access

PeopleSoft Pure Internet Architecture is a completely server-based architecture. Clients to this architecture can be nearly any kind of internet access device, such as:

- A web browser running on supported operating system.
- A wireless device or cell phone.
- An external or third-party system with extensible markup language (XML)/hypertext transfer protocol (HTTP) protocols.

No PeopleSoft executable files are on the client; thus the client can be any internet device that uses standard internet technologies such as HTTP, hypertext markup language (HTML), and XML to communicate with the PeopleSoft internet application server.

A web browser running on a PC is the most common internet client. The PeopleSoft internet application server simply serves HTML and JavaScript to the web browser and the end user navigates through the PeopleSoft application as if he or she were navigating any other website.

A key concept of PeopleSoft Pure Internet Architecture is that no complex client software installation is required. The internet client device accessing the internet architecture already has all of the software and configuration that it needs. No additional software must be installed on the client for interaction with PeopleSoft applications. For example, no Java applets, Windows .DLLs, or browser plug-ins are needed.

The Web Server

The web server acts as the front end of PeopleSoft Pure Internet Architecture. When a client connects to the server by way of a URL, the system displays a sign-in screen sent to the browser in HTML. The web server manages communication with the browser.

The following web server products can be configured to deploy your PeopleSoft applications: Oracle WebLogic server.

Two key PeopleSoft servlets run on the web server—the presentation relay servlet and portal servlet.

Presentation Relay Servlet

The presentation relay servlet is used to process all inbound and outbound HTTP requests for PeopleSoft transactions and queries. This very thin servlet acts as a relay between the client device and the core back-end services on the application server. It receives and serves HTML, XML, and wireless markup language (WML) requests over HTTP.

Portal Servlet

The portal servlet is a Java servlet that runs on the portal web server. It intercepts user requests for HTML pages, retrieves the requested page, wraps additional content around it, and then sends the resulting page to the user's browser. The servlet acts as an invisible browser that sits between the user's browser and requested content.

The portal servlet:

- Provides a consistent user interface.

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

- Ensures that PeopleSoft-specific tags are processed correctly.

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

One of the most important aspects of portal technology is its role in integrating content from a wide variety of data sources and displaying that content on a single page in a coherent, understandable, and presentable way. We refer to this complex process as “page assembly and proxying.” Portal processing assembles the page to be displayed based on the content retrieved from various data sources. It uses portal templates to wrap the contents of the assembled document into a single page that fits into the context of the site.

- For page-based templates, the portal servlet assembles pages for the browser.

  It ensures that all URL references in the HTML on the assembled page are references back to the portal servlet itself. In some cases, each URL in the HTML document assembled by the portal servlet must be rewritten to reference the portal servlet, not the originally requested URL. This process of redirecting URLs so that they point to the portal servlet is called proxying.

- For frame-based templates, the portal server updates the src tags in the frameset with the target content and sends it to the browser.

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

The Application Server

The application server is the core of PeopleSoft Pure Internet Architecture; it carries out business logic and issues SQL to the database server. The application processing logic that ran on the client in previous releases now runs on the application server. The application server consists of numerous PeopleSoft services and server processes that handle transaction requests. These include requests to:

- Authenticate users.
- Build application pages.
- Save application pages.
- Run some PeopleCode.
- Run SQL (prompts, page-specific SQL, validations, and so forth).

The application server is responsible for maintaining the SQL connection to the database for the browser requests and the Windows development environment. PeopleSoft uses Tuxedo to manage database transactions and Jolt, Tuxedo's counterpart, to facilitate transaction requests issued from the internet. Both Tuxedo and Jolt are products of Oracle Systems.

The Portal Processor runs as an application service of the PeopleSoft application server. It runs with the other application services, such as the Component Processor, Security Manager, and SQL Access Manager. Portal Processor tasks include:

- Fetching content templates from the database.
- Fetching content references from the database portal registry and caching them in the application server portal registry.
- Processing personalizations.
- Interacting with other application services (lightweight directory access protocol (LDAP), role-based security, and so forth).

The Database Server

The PeopleSoft database is the repository for all information managed by PeopleSoft applications. Not only is application data stored in the database, but PeopleSoft metadata is also maintained in the database. PeopleSoft Application Designer enables you to define and maintain this metadata that the system uses to drive the runtime architecture. The application server carries out business logic based on the PeopleSoft metadata.

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

Note: In general, PeopleSoft Interaction Hub 9.x can retrieve content from any RDBMS-based application as long as the content that is being retrieved is URL-accessible and is HTML-based. Always refer to the certifications section of My Oracle Support for supported RDBMS products.

The Portal Registry

The portal registry is a key administrative component within the metadata of a PeopleSoft database. A portal registry is a hierarchical structure in which URLs accessed by way of the portal are organized, classified, and registered. Each portal registry consists of the following objects:

- Folders.

  Folders group and organize content references into a hierarchy. Each folder can contain content references as well as other folders.

  Every portal registry contains a root folder and a Portal Objects folder. The Portal Objects folder contains administrative objects specific to the portal and includes the following folders: Homepage, Navigation Collections, Pagelets, Template Pagelets, and Templates.

  In addition to these standard folders, several folders typically are located directly under the root folder: one folder for PeopleTools (administrative references) and other main folders for each PeopleSoft application. These main application folders contain the folders and content references associated with each PeopleSoft application that you've licensed.

- Content references.

  Content references are URLs that have been registered in a portal registry. They can be PeopleSoft application pages or external web pages. Content references fall into four main categories: pagelets, target content, templates, and homepage tabs.

  In addition to specifying a URL, each content reference includes additional information such as its creator, effective dates, associated template, search keywords, and so forth. Registry URLs can point to any website that responds to HTTP requests with an HTML response—that is, a static or dynamic web page.

  For example, a content reference could be a URL pointing to a page in a PeopleSoft application. Or a content reference could point to static HTML pages on an intranet site, such as a procurement or expense policy document, or dynamic pages created by an analytic or reporting system. Access to content references is controlled by security permission lists assigned to each content reference. Any portal content can be limited to a specified group of users, or made public and available to any authorized portal users.

- Nodes.

  Nodes refer to a source of HTML content. They primarily identify the universal resource indicator (URI). A URI is a subset of the URL that points to the location of the resource. It does not include the content information, such as the target file or application and any parameters passed to that resource.

The portal registry's hierarchical structure enables a portal administrator to more easily create a classification and navigation system in which content references can be registered and managed for all portal users. PeopleTools provides portal administration pages for this purpose. Additionally, the portal includes a registry application programming interface (API) for maintaining the portal registry from PeopleCode.

To improve performance, a copy of the registry is also stored on the application server in cache files. This is how the portal servlet accesses the registry at runtime.

Portal Components

The portal technology framework includes the following services:

- Personalization.

  Pagelets optionally can be assigned a user personalization settings page. This allows the user of the pagelet to specify selections, settings, or attributes specific to the pagelet's content. These personalizations commonly alter the content of the pagelet. An example of this would be setting the city for which a weather pagelet displays forecast information. A personalization page is accessible to the user by means of clicking the **Personalize** button on the pagelet's frame header.

- Search.

  The portal uses a search engine to quickly search for registered content in the portal registry. This is a popular means of portal navigation. PeopleTools search capabilities were built to assume multi-language support, including double-byte languages.

- Navigation.

  Important! Breadcrumbs, drop-down menus, menu pagelets and other forms of classic navigation are no longer supported. The information in this section pertains to a prior version of PeopleSoft classic applications only. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see "Working with Fluid Pages and Controls" (Applications User's Guide).

  PeopleTools portal technology provides a set of navigation components based on the portal registry. These components are the drop-down menu and Favorites. Navigation has been engineered to provide rapid access to complex information based on the role of the user.

  - Menu navigation.

    PeopleTools portal menu navigation provides a consistent method of content access, categorization, and organization. The menu navigation presents a dynamic hierarchy-based view of the folders and references within the portal registry. Menu navigation is available through the Menu pagelet or from the Main Menu drop-down navigation depending on your portal settings.

  - Favorites.

    The PeopleTools portal includes a My Favorites folder that you use to store frequently accessed pages. In the Menu pagelet, this folder is at the root level. In drop-down navigation, My Favorites appears under the Favorites drop-down menu.

  - Recently Used.

    The PeopleTools portal include a **Recently Used** list that includes up to five of your most recently accessed components. This list appears under the Favorites drop-down menu.

Related Links

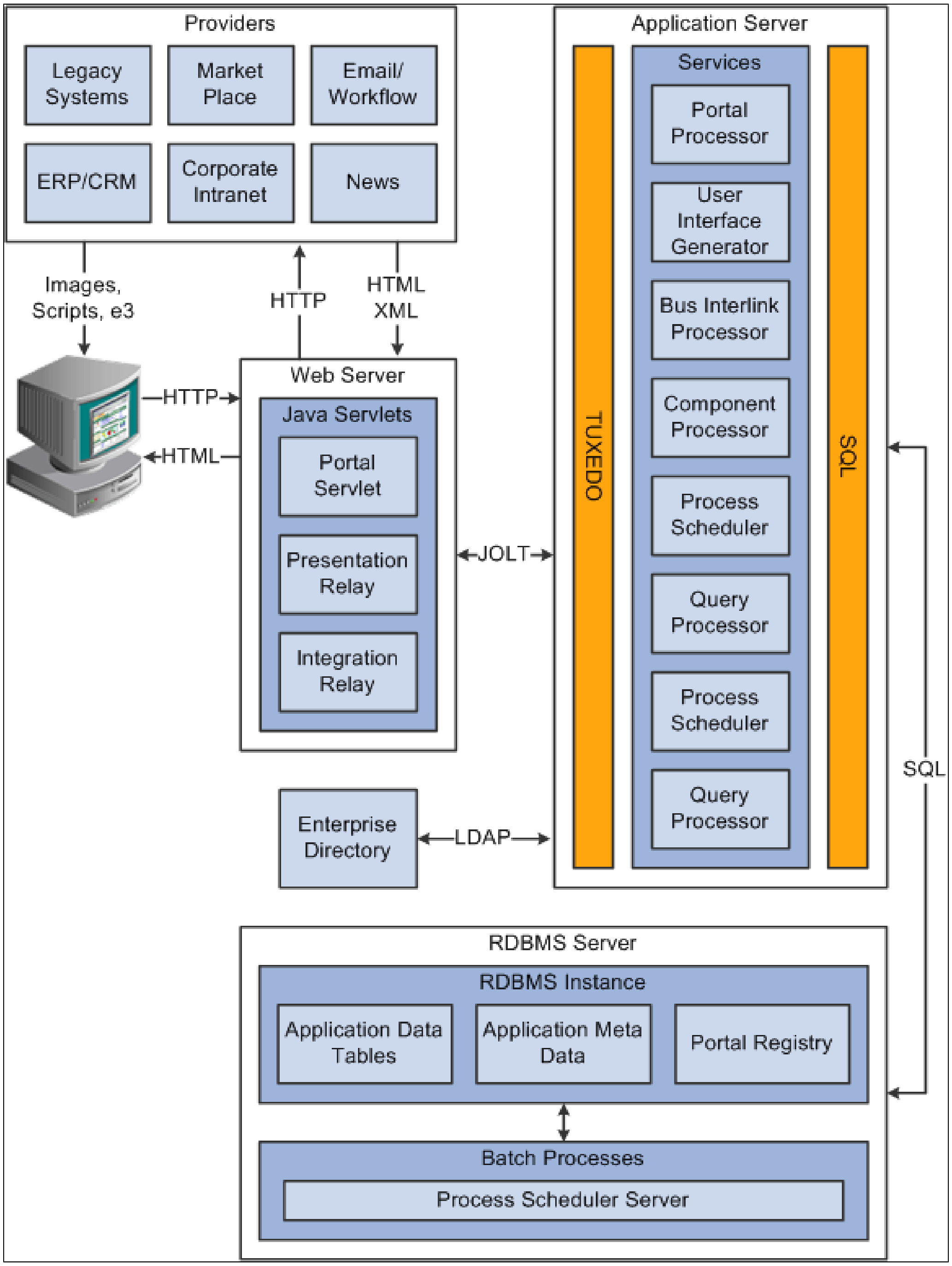
Managing General Settings for Portals

Page Assembly

Previous sections in this PeopleBook have described the individual components of PeopleSoft Pure Internet Architecture. This section provides a high-level overview.

1. An HTTP request is sent from the user's browser to the portal's web server, which invokes the portal servlet. Assume that the browser is requesting a PeopleSoft application page.
2. The portal servlet makes a request to the Portal Processor to fetch the content reference for the link selected by the user. The Portal Processor fetches the content reference from the portal registry and passes a partially completed HTML document back to the portal servlet.
3. The content reference could be pointing to any or several of the content providers (specified by a node). Each content reference is referenced in a partially completed HTML document. The portal servlet issues the HTTP request to the appropriate web server and retrieves the HTML document.
4. The portal servlet then completes the page assembly process with the template, merging the content from the HTML document, and then serves the completed page back to the user's web browser.

The following diagram illustrates the numerous sources of data for page assembly in the PeopleSoft Pure Internet Architecture:



Understanding Pagelet Development

This section discusses:

- Pagelet development guidelines.
- Pagelet development considerations.

- Pagelet development options.
- Size considerations.
- HTML design considerations.
- Branding considerations.

Pagelet Development Guidelines

Follow these guidelines when developing new pagelets:

- Adhere to object naming standards.
- Do not provide any filter, sort, or refresh buttons because they also require trips to the application server.
- If you enable the cache homepage option in the web profile, the pagelet must be display-only and cannot be interactive.

Because the portal servlet performs page assembly and proxying only on blocks of HTML, a pagelet must:

- Be URL-accessible.

  The portal servlet will issue an HTTP request to the registered URL.
- Be HTML 4.0 compliant.

  The HTML returned will be combined with HTML from other pagelets to form the entire web page.
- Be sized appropriately.

  The pagelet's width should conform to the narrow or wide guidelines discussed in the Sizing section in this document.

  See Understanding Pagelet Wizard Display Sizes.
- Use JavaScript name spaces.

  If you write custom JavaScript code, ensure that JavaScript from multiple portlets can coexist on the same page. Only one namespace is shared by all portlets on a portal page. For example, if portlet A and portlet B both use the GETURL function, only one definition will exist for GETURL defined by the most recently processed portlet. If you need a different GETURL function for each portlet, then you must uniquely name the GETURL function for each portlet, for example, portlet A calls GETURL1 and portlet B calls GETURL2. If the same GETURL function is called by both portlets, then you don't need to rename GETURL because the most recent definition of GETURL will be used by both functions.

Pages created by way of PeopleSoft Pure Internet Architecture conform to these requirements. However, these requirements are not exclusive to PeopleSoft. As mentioned, pagelets can also be from remote, non-PeopleSoft systems, such as static HTML files or dynamically generated HTML from a variety of technologies such as JavaServer pages (JSPs), Java servlets, Microsoft active server pages (ASPs), Perl scripts, and CGI programs.

Pagelet Development Considerations

When developing pagelets, you should also consider the following topics.

Single Signon

Oracle delivers a component interface named PRTL_SS_CI that enables external applications to support a single signon solution with PeopleSoft portal applications. This enables users to sign in once when accessing the portal and not have additional sign-ins for each system that is referenced in your portal.

See Understanding Single Signon and Pagelets.

Layout

Observe the following rules for layouts:

- Avoid horizontal scrolling.
- Do not use page titles.

  The name of the content reference in the portal registry is used as the pagelet title.
- Pagelet instructions should not be necessary.
- Avoid group boxes whenever possible.

  Note: They may be necessary to separate sections.

- Left align text wherever possible.
- Deselect the **Show Borders** option for group boxes and scroll areas.
- Select **View** > **Internet Options** in PeopleSoft Application Designer to ensure that you can access all the styles available.
- Leave more than 20 percent spacing between field labels and field edit boxes because the rendered styles are larger in the browser than they appear in PeopleSoft Application Designer.
- Before bringing your pagelet into the portal, view it in PeopleSoft Application Designer. Select **Layout** > **View in Browser.**

Appearance and Character

The appearance of objects on pagelets, such as text, links, tables, and so forth, should be modified by means of the PeopleSoft style classes to retain a consistent appearance and character between your pagelet and the rest of the portal. PeopleSoft uses cascading style sheets (CSS). Style sheets are useful because they enable developers to change page attributes across multiple pages quickly and easily.

Each style sheet is composed of individual classes, which affect the formatting of each page. Classes control a broad range of characteristics, including font size, spacing, alignment, border width, weight, and color.

When creating the page, select the *Use Default Style Sheet* option. For any controls on the page, select the *Use Default Style* option.

All style sheets used in each pagelet on the page are referenced on the final page that is assembled by the portal. Therefore, a pagelet should not redefine any class that might be shared by other pagelets on the assembled page. If a class needs to be changed, define a new one.

The order of precedence for style sheets is:

1. Template.
2. Target component.
3. Pagelets – in the order that they are assigned.

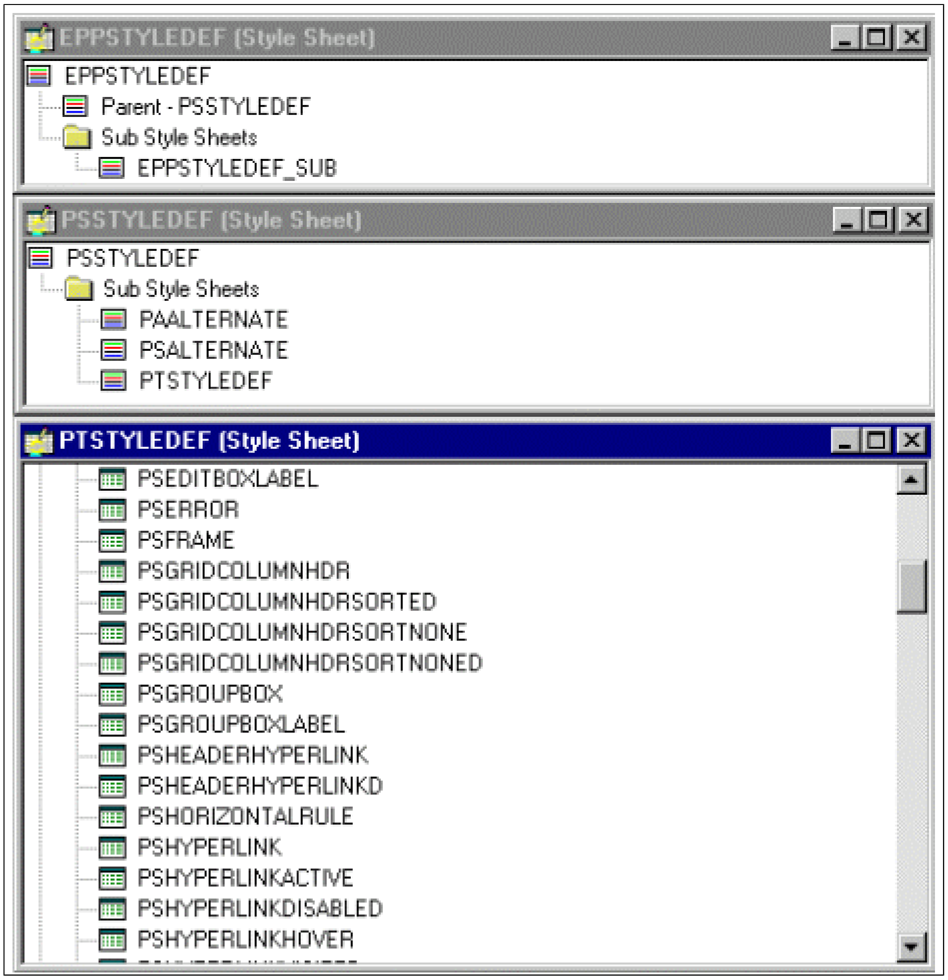
Note: In the same way that PeopleTools supports subrecords to enable shared definitions between records, you can define sub style sheets to share a common set of classes. A sub style sheet has all of the properties of a style sheet.

Style sheet definitions are stored in the database. They are accessed and modified by means of PeopleSoft Application Designer. The default style sheet used by PeopleTools depends on your application release, but is either PSSTYLEDEF or PSSTYLEDEF_SWAN. The PeopleSoft Interaction Hub uses a PSSTYLEDEF or PSSTYLEDEF_SWAN derivative.

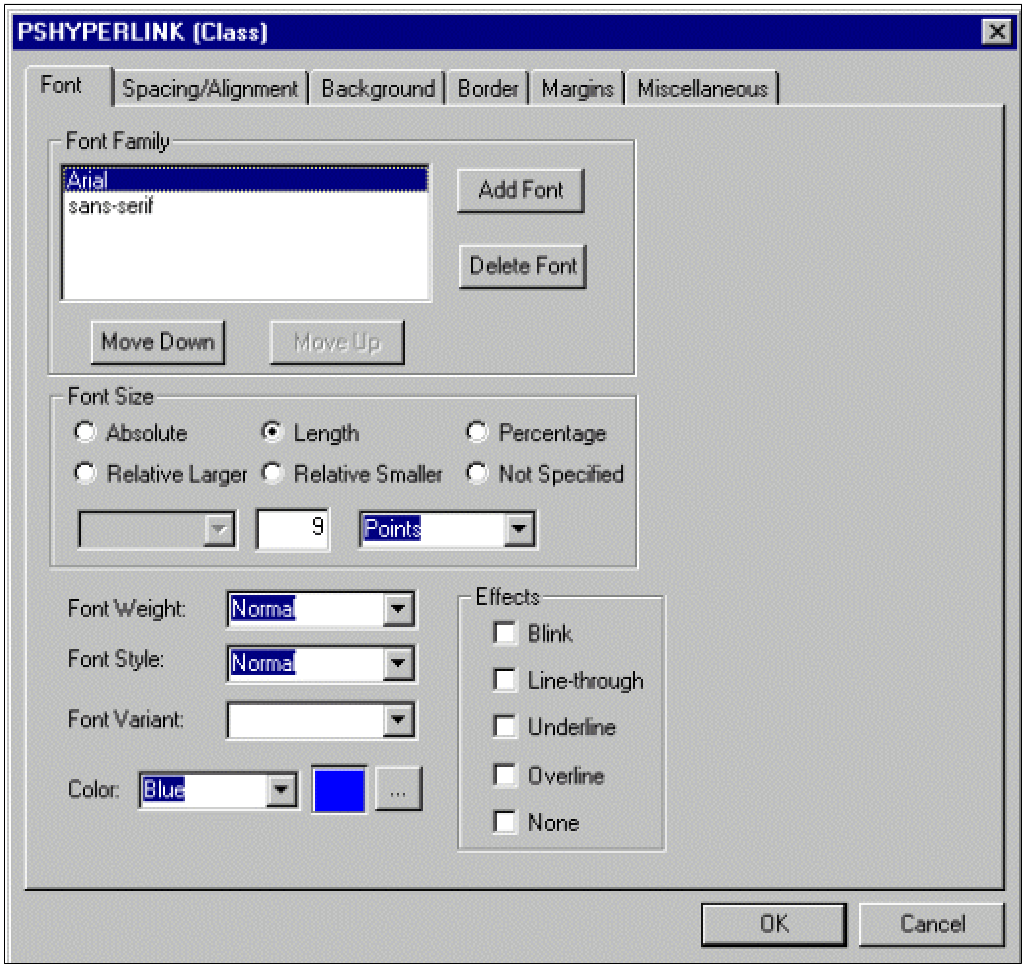
You can determine which style sheet is being used by selecting **PeopleTools** > **Utilities** > **Administration** > **PeopleTools Options.** The value in the **Style Sheet Name** field designates the style sheet used by the database.

Two ways are available by which you can see the style classes defined in a style sheet.

If you access PeopleSoft Application Designer, you can open the designated style sheet and view its definitions in a graphical interface.



Double-click any style class to view and modify its definition.



If you don't have access to PeopleSoft Application Designer and want to see the CSS file, you can access the file cached for the CSS on the web server. For example, on WebLogic you'll find the file in this directory:

`....\peoplesoft\applications\PORTAL\`*`site_name`*`\cache`

The file is generally named PSSTYLEDEF_1.css or PSSTYLEDEF_SWAN_1; however, it may have a slightly different name. For example, it might have a different numbered suffix.

Some of the most common styles used by pagelets include:

- PSTEXT

  General text label.

- PAADDITIONALINSTRUCTIONS

  Instructional text.

- PABOLDTEXT or PABOLDBLUETEXT

  Commonly used for group and category titles.

- PSPSMALLTEXT

Commonly used for footers and branding (credits).

- PSPAGE

  Commonly used for background colors.

- PSLEVEL1GRIDODDROW

  Shaded row, which is typically light gray or light blue.

- PSLEVEL1GRIDEVENROW

  Lighter-colored row, which is typically white.

- PSCHECKBOX
- PSDROPDOWNLIST
- PSEDITBOX
- PSGROUPBOX

  Commonly used to create groupings or dividers.

- PSHYPERLINK
- PSPUSHBUTTON
- PSRADIOBUTTON

The following examples show ways in which these styles are used in HTML and translated into elements in the PeopleSoft Pure Internet Architecture interface:

This HTML generates the following interface:

```
<table border='0' cellpadding='0' cellspacing='0' cols='1' CLASS='PSGROUPBOX' style⇒

="border-style:none;" width='639'>  <tr>
   <td class='PSGROUPBOXLABEL' align='LEFT'>Language Settings</td>
  </tr>
   <td colspan='2'  valign='top' align='LEFT'>
    <label for='PSOPTIONS_LANGUAGE_CD' class='PSDROPDOWNLABEL'>Language Code:
    </label>
   </td>
   <td  valign='top' align='LEFT'>
    <span class='PSDROPDOWNLIST' >English</span>
   </td>
...
<input type='check box' name='PSOPTIONS_TRANS_CHG_LASTUPD' id='PSOPTIONS_TRANS_CHG_⇒

LASTUPD' tabindex='14' value="Y" />
 <label for='PSOPTIONS_TRANS_CHG_LASTUPD' class='PSCHECKBOX'>
   Translations Change Last Update
 </label>
```


This example illustrates the fields and controls in the Language Settings group box of the PeopleTools Options page.

This HTML generates the following interface:

```
<input type='text' name='STYLESHEET' id='STYLESHEET' tabindex='28' value="" class='⇒

PSEDITBOX' style="width:23px; text-align:RIGHT; " maxlength='2'/>
  <label for='PSOPTIONS_TEMPTBLINSTANCES' class='PSEDITBOXLABEL'>    Style Sheet Na⇒

me:
  </label>
```

This example illustrates the Style Sheet Name field of the PeopleTools Options page.

Links and URLs

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

When a user signs in to a PeopleSoft portal, he or she signs in to a web server on which the portal servlet is running. The portal servlet processes all HTML that isn't in the simple URL format, converting all URL references to point to the portal web server rather than the original URL. The original URL is still necessary to retrieve the requested content; it is stored in the new URL in the URL query string parameter. The portal servlet proxies all links and all form actions in this manner.

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

Note: If URLs are included in your HTML, you must use absolute URLs as opposed to relative URLs.

When a URL is invoked on a target page, as opposed to the homepage, the content associated with the URL is rendered within the target frame. The PeopleSoft Interaction Hub and header and left-hand area will remain. Therefore, proxying isn't required to have the new content rendered in the context of the PeopleSoft Interaction Hub.

Performance

Homepages are cached on each user's browser. The browser does not access the web server after the homepage is initially retrieved. You can enable or disable this feature, and also adjust the time interval before the web server is accessed again to get a fresh homepage. In any case, if a user clicks the browser

Refresh button, the homepage is accessed from the web server again, overwriting the homepage cached on the browser.

Important! If any homepage pagelet in your implementation uses interactive mode, you must disable browser homepage caching in the web profile.

The following PeopleCode function is used to trigger a refresh of the homepage:

`FUNCLIB_PORTAL.TEMPLATE_FUNC.FieldFormula.ForceRefreshHomePage().`

Additionally, the following configuration properties are associated with homepage caching. Any changes to these settings are applied to all users signing in to the web server.

- PortalCacheHomepageOnBrowser=<*True* or *False*>

  If set to *True,* the homepage is cached on the browser. If set to *False,* the homepage is not cached on the browser.

- PortalHomepageStaleInterval=<*seconds until stale*>

  A homepage cached on the browser is considered stale after the specified number of seconds. The next time a user accesses the homepage by clicking a link, the web server is accessed and the homepage is refreshed on the browser.

Because different browser versions do not process HTML in exactly the same way, the browserprops.xml file on the web server on which the portal servlet is installed enables you to disable homepage caching for selected browser versions.

This can be useful if you have one or two supported browsers and want to disable cache for nonstandard browsers that could pose an administration problem. Follow the instructions in the file to disable caching for certain browser types.

As with homepages, navigation pages are cached on each user’s browser. You can set options for navigation caching by using the **Time page held in cache** (METAXP) option.

Set the **Time page held in cache** option by navigating to **My Personalizations** and clicking the **Personalize Option** button for the General Options personalization category. Note that this option is set in minutes, not seconds. A change to this option is picked up by the application server immediately. However, because the users' browsers already have cache control set by the previous value of the option, you must delete the browser cache for the new **Time page held in cache** value to take effect.

Oracle enables you to prevent the system from caching PeopleSoft pages to the browser. You control browser caching using the EnableBrowserCache property in the configuration.properties file.

Being able to control browser caching is useful for situations in which PeopleSoft applications are deployed to kiosk workstations where multiple users access the applications. Enabling you to prevent caching means that users can't click the **Back** button to view another individual's transaction or view any other sensitive data.

The side effect of completely disabling caching is degraded performance. For each new page, the system must access the database. However, PeopleTools offers a compromise related to browser caching in the form of the **Time page held in cache** (METAXP) option discussed earlier.

You can enable browser caching for the navigation pages that remain relatively static. This option applies to PeopleSoft Pure Internet Architecture navigation pages, portal homepages, and navigation pages. Use it to take advantage of the performance gains of caching while limiting the amount of time that navigation pages—the menu pages—remain in cache.

The **Time page held in cache** option is set to 20 by default. To disable this option, enter *0* in the **Override Value** edit box. The minimum value in minutes for this option is *0* (disabled) and the maximum value is *525600,* which is one year.

If the **Time page held in cache** option is set to *20,* and if you assume that the time is 7 p.m. on August 24, then the header information in the HTML is dated like this:

```
Wed 24 Aug 2011 07:20:00 PM
```

This header information indicates that in 20 minutes the system needs to check for a new page. This reduces the performance degradation when no caching occurs at all.

By default, the EnableBrowserCache property is set to *True.*

If the EnableBrowserCache property is set to *False*:

- The system never caches pages. When a user clicks the browser **Back** button, she receives a Page Expired message in .
- The setting overrides any date and time header settings.

The following table helps illustrate the way in which the EnableBrowserCache option works with the METAXP option.

EnableBrowserCache	***METAXP***	***Caching Behavior***
True	*0*	No caching due to the *0* value in METAXP.
True	*> 1*	Pages are cached with expiration values set in Greenwich Mean Time (GMT) based on the **Time page held in cache** value (METAXP).
False	*0*	No caching.
False	*>1*	No caching. The EnableBrowserCache option setting overrides the **Time page held in cache** value (METAXP).

Multi-language Support

PeopleSoft applications currently support the following 33 languages:

- Arabic.
- Bahasa Malay.

- Bulgarian.
- Canadian French.
- Chinese (Simplified).
- Chinese (Traditional).
- Croatian.
- Czech.
- Danish.
- Dutch.
- English.
- Finnish.
- French.
- German.
- Greek.
- Hebrew.
- Hungarian.
- Italian.
- Japanese.
- Korean.
- Norwegian.
- Polish.
- Portuguese.
- Romanian.
- Russian.
- Serbian.
- Slovak.
- Slovenian.
- Spanish.
- Swedish.
- Thai.
- Turkish.

- UK English.

Pagelet Wizard can access the PeopleSoft runtime environment to determine details of a user's profile, such as his or her language. For more information about how you can use this Pagelet Wizard feature to facilitate passing runtime parameters, such as language, see Pagelet Wizard documentation in this PeopleBook.

See Understanding Pagelet Wizard.

In addition, Java classes delivered with PeopleTools enable you to call PeopleCode from your Java program and access contextual information from the runtime system. If needed, language can be retrieved through a PeopleCode function that is accessible from Java. Further information about this feature can be found in this document.

See Developing Java-Based Pagelets.

Pagelet Development Tools

You can create a pagelet using several ways. Ultimately, the portal servlet assembles HTML, so the key is determining how to generate the HTML. Some methods leverage PeopleTools, while other options allow pagelet creation without PeopleTools.

One set of options is to develop pagelets with PeopleTools. This is the most straightforward approach if you are dealing solely with data from PeopleSoft applications. The two types of PeopleTools-based pagelets are:

- PeopleSoft pages.

  PeopleTools dynamically generates the appropriate HTML to render the page data based on the definitions created within PeopleSoft Application Designer. This is the most straightforward approach when the data being rendered is in a PeopleSoft application database.

- iScripts.

  You can write a PeopleCode function that renders HTML using %Request and %Response objects, which are similar to ASP or JSP programs. This allows more control over data retrieved and enables you to conditionally render HTML. This approach gives you maximum flexibility, but unlike the PeopleSoft Pure Internet Architecture page approach, you must code for things such as multi-language and browser support.

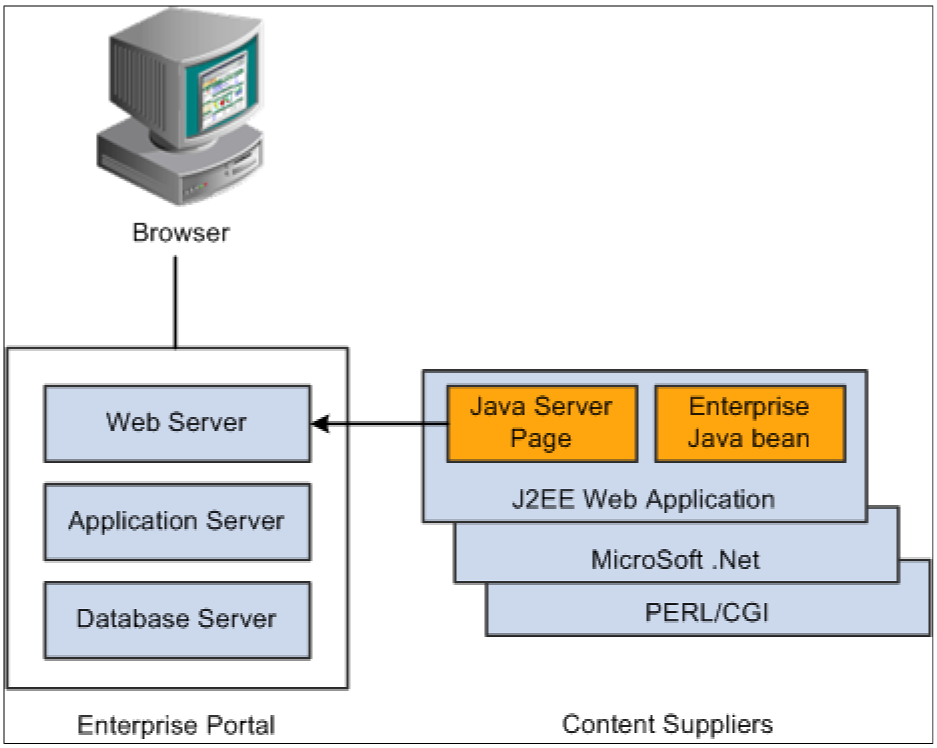
The focus of this document is a set of options that can be used to create a pagelet application with one of the many technologies that generate HTML. This may be the approach you take if you're integrating a non-PeopleSoft system with the PeopleSoft Interaction Hub. For example, you're applying business logic to data from a non-PeopleSoft system, or you might be combining data from a PeopleSoft application with other systems.

The pagelet can be written in a variety of technologies, including:

- Adobe ColdFusion.
- AJAX.
- ASP.
- CGI.

- HTML.
- Internet server API (ISAPI).
- JavaScript.
- JSP.
- Perl.
- Servlets.
- Tool Command Language (Tcl).

The following diagram illustrates some of the ways that you develop in a PeopleSoft environment using technology other than PeopleTools:



You can convert your own HTML into a pagelet in several ways:

- Pagelet Wizard.

  Pagelet Wizard walks you through a series of steps involved in creating and publishing a pagelet. Portal administrators and nontechnical users can integrate and transform data from a wide variety of data sources, both internal and external to PeopleSoft applications.

  External sources include web applications that can be referenced with a URL, HTML block (such as a form or table), and Java classes. You do not need to have PeopleSoft-specific application development tools or skills to be able to create pagelets.

- Registered URL.

  You can also choose to register a URL, such as a JSP or ASP, directly in the portal registry.

Pagelet Sizes

The homepage layout can vary based on end-user personalization. These two layout choices are available for homepages and dashboard pages:

- The three-column layout divides the homepage into three equally-sized columns. In PeopleSoft applications, these are called narrow columns.
- The two-column layout divides the homepage into two differently-sized columns: one narrow column and one wide column, which is twice the width of the narrow column.

Because pagelets must function in two possible homepage layouts, it is column size that drives pagelet size. This fact is evident in the three categories of pagelet widths that are listed here:

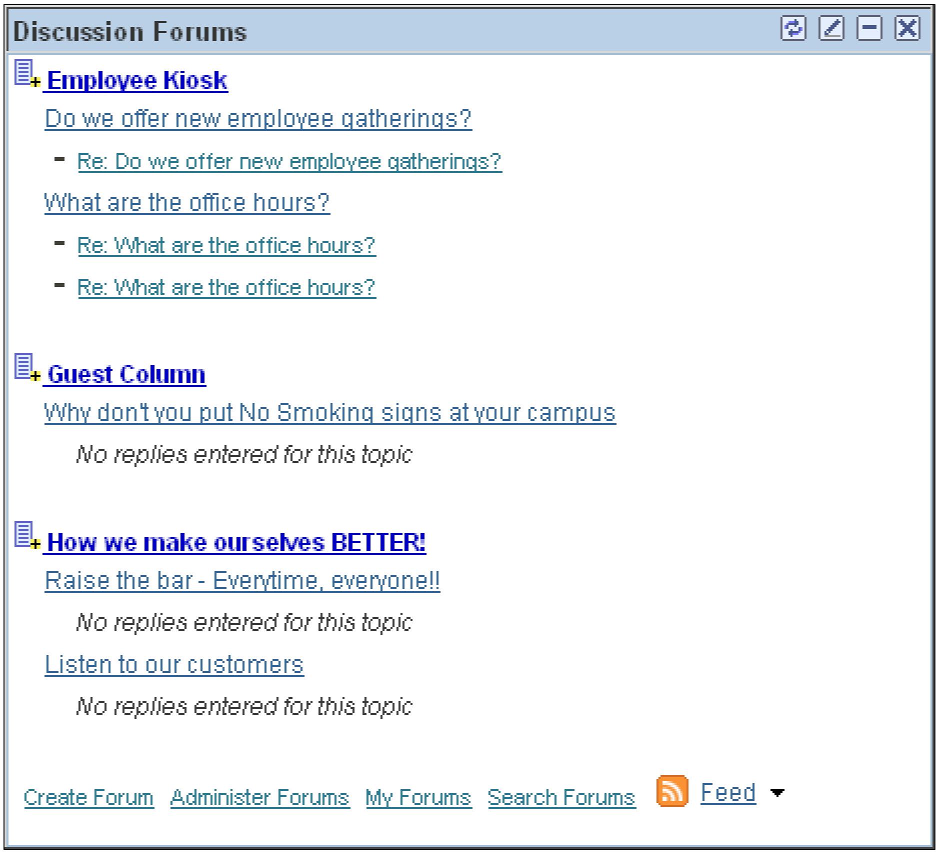
- Narrow pagelets.
- Wide pagelets.
- Banner pagelets.

Narrow Pagelets

With three-column layout, all pagelets are rendered as narrow pagelets. Narrow pagelets are designed to be approximately 300 pixels wide. Subtracting the border and the internal margin leaves 285 pixels for the content. A narrow pagelet appears in a column that is one-third the width of the portal homepage. Because the pagelet is narrow, you should provide a succinct list of values that users can quickly traverse. Select the minimum set of data that best encapsulate the pagelet content. A narrow pagelet typically accommodates a single 30-character field.

A pagelet that has been designed to fit in a narrow column can be rendered in a wide column as well. However, user interface issues can occur when a pagelet that has been designed to render in a wide column only is rendered in a narrow column. Therefore, a pagelet should always be able to operate in a narrow format.

This example shows the Discussion Forums pagelet as a narrow pagelet. The essential links and information are displayed in the narrow format.

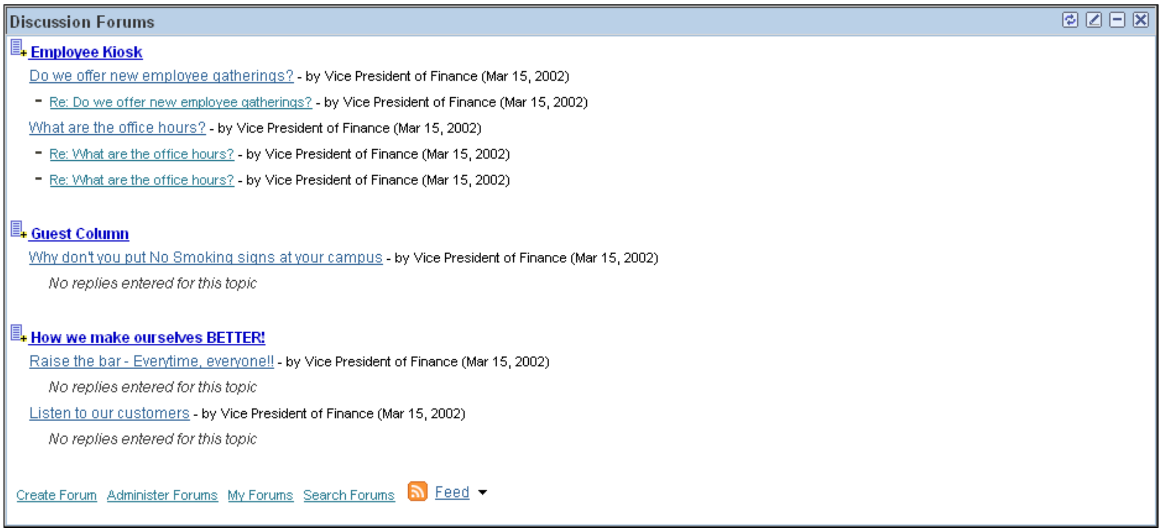


Generally, all pagelets should be developed with the narrow column size as the default. However, a pagelet can be designed to take advantage of the additional space when it is rendered in a wide column.

Wide Pagelets

With two-column layout, the first column allows spacing for a narrow pagelet and the second column allows spacing for a wide pagelet. Wide pagelets are designed to span two-thirds of the width of the homepage. Subtracting the border and the internal margin leaves 650 pixels for the content. Although you can fit more data on the pagelet, the data must remain meaningful.

This example shows the Discussion Forums pagelet as a wid pagelet. In addition to the essential links and information that are also displayed in the narrow format, the wide format includes who posted a discussion topic or reply and when.



When the layout contains the wide column, the **PORTALPARAM_COMPWIDTH** query string parameter is set to wide and passed to the HTML denote the wide version. Developing a wide version of a pagelet is optional.

The PORTALPARAM_COMPWIDTH Parameter

The parameter `PORTALPARAM_COMPWIDTH` can contain the value *Wide* or *Narrow* and can be used to specify whether a pagelet is being rendered in a narrow or wide column. The function **pgltAtion** does not check or use this parameter, the narrow pagelet will just be rendered in a wider area.

This sample PeopleCode references the query string parameter `PORTALPARAM_COMPWIDTH:`

```
Component string &CompWidth;

&CompWidth = %Request.GetParameter("PORTALPARAM_COMPWIDTH");

If &CompWidth = "Wide" Then;
  TransferPage(Page.PT_CONTRK_PGLT_W);
End-If;
```

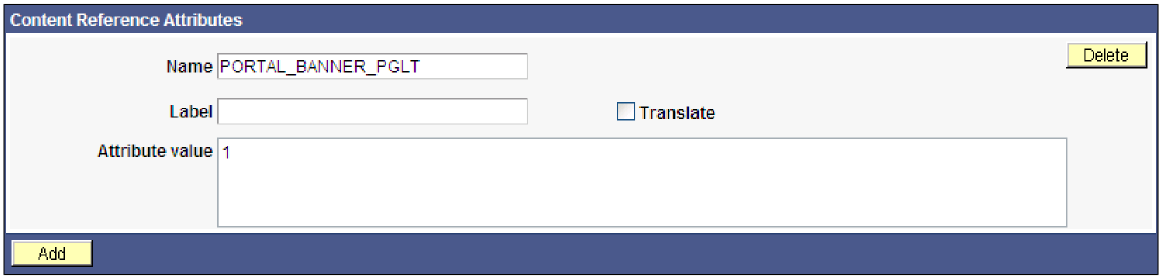
Banner Pagelets

Banner pagelets are the widest of all pagelets. Banner pagelets extend across the entire width of the homepage or dashboard page and span all columns regardless of layout. Both the two-column and three-column layouts enable you to add one banner pagelet that is positioned at the top of the page. Banner pagelets are designed to be 1240 pixels wide.

Note: The actual number of pixels that are available for content varies based on the resolution of your display and the rendering engine of your browser.

The portal uses the content reference attribute, *PORTAL_BANNER_PGLT* to appropriately position banner pagelets. You must manually enter this attribute name and its accompanying value.

This example shows the entry for the PORTAL_BANNER_PGLT content reference attribute:



Important! Homepages and dashboard pages can display one banner pagelet only. If more than one pagelet on a homepage includes the PORTAL_BANNER_PGLT attribute, the system recognizes the attribute for the first pagelet and ignores the attribute for any other pagelets.

See Creating Banner Pagelets.

Pagelet Extensions

Pagelet extensions are standard pages. They can be registered with a template that allows it to use the entire width of the browser with no side frame for navigation, or with a template that includes a side frame.

In the former case, no inherent sizing requirements are imposed. PeopleSoft pages are designed for browsers running with a resolution of at least 1280 x 1024. Therefore, when a side frame is not included, a pagelet extension should not be wider than 1190 pixels, accounting for borders and so forth.

In the latter case, a pagelet extension should not be wider than 375 pixels.

HTML-Related Design Considerations

Consider these HTML-related guidelines when you are designing pagelets:

- Ensure that the pagelet encapsulates data to provide at-a-glance summary information.
- Ensure that the pagelet provides links to detailed application pages.
- Avoid large borders or a design that creates extraneous white space. Screen area on a homepage is valuable and data should be maximized.
- Avoid designs that make the pagelet wider than the prescribed size. This undesirable design forces the user to scroll horizontally, so you should design your HTML to be vertically oriented. For instance, radio buttons should be arranged vertically, not side-by-side. When you create a form, buttons should appear below a text box, not next to it.
- Because this is a business-oriented portal, avoid extravagant or extraneous graphics. If you use graphics, they should be purposeful and unobtrusive. No graphic should be wider than 218 pixels, which would force the pagelet to be wider than a narrow pagelet.
- Avoid using any text or HTML tags that force the pagelet to a width that is greater than 218 pixels.

- In general, avoid explicit sizing. Let the browser render content in the space that is available, which should allow your text and graphics to fit appropriately. For example, place text in an HTML table to enable the table to wrap any long strings.
- When appropriate, use the PeopleSoft style sheet to ensure that the appearance and character of your pagelet is consistent with the rest of the PeopleTools portal content.
- When your pagelet does not use the entire width of the column, center the pagelet content.
- If a link on the pagelet takes the user to another website and you want to display the content in a new window, rather than using *target=_blank* in the link tag, use the following code:

```
<a href="javascript:void window.open
('http://www.example.com/cgi-bin?article3');" class="PSHYPERLINK">
```

- If a pagelet has personalization options, the pagelet should have a default mode. In default mode, ensure that the pagelet includes a standard set of data and a message conveying that default data can be personalized.

PeopleSoft Interaction Hub Guidelines

This section lists some HTML-related development guidelines for pagelets that you use in the PeopleSoft Interaction Hub.

- Within PeopleTools, attributes from all content are merged into the single <BODY> tag; be aware of the order of attribute precedence.

  The order of attribute precedence is:

  1. Template.
  2. Target content.
  3. Template components.

  Events beginning with *on* are naively combined, so if one of them issues a *return,* the following events will not run. All other attributes are not examined; rather they are used in order of precedence. For example, if both the template and a content component contain *topmargin,* the template attribute will be applied.
- Avoid large borders or anything that creates extraneous white space. Screen area on a homepage is valuable and data should be maximized.
- Avoid implementing anything that will make the pagelet width wider than the prescribed size. Making the user scroll horizontally is undesirable.

  Make your HTML vertically oriented. For instance, radio buttons should be placed vertically, not side-by-side. When you create a form, any buttons should appear under a text box, not next to it.
- Avoid using graphics that do not have thoughtful purpose. This is a business-oriented portal, so extravagant or extraneous images should not be used. If any are used, they should be purposeful and unobtrusive. No graphic should be wider than 218 pixels. A graphic larger than this size will force the width of the pagelet to exceed the size of a narrow pagelet.
- Avoid using any text or directives that force the pagelet to a width greater than 218 pixels.

- In general, avoid explicit sizing and allow the portal to set the width. Your text and graphics should then fit appropriately. For instance, placing text within an HTML table will allow it to wrap any long strings.
- When appropriate, use the PeopleSoft style sheet to ensure that the appearance and character of your pagelet is consistent with the rest of the PeopleSoft Interaction Hub content.
- When your pagelet does not use the entire width of the column, center the pagelet content.
- If a link takes the user to another website and you want it to display the content in a new window, rather than using *target=_blank* within the link tag, use:`<a href="javascript:void window.open ('http://www.example.com/cgi-bin?article3');" class="PSHYPERLINK">`

Branding Considerations

As mentioned earlier, this is a business-oriented portal. Although content and name recognition are important, they are secondary to the primary goal of enabling end users to perform their work in a more efficient manner. Thus, any branding should be subtle and never detract from the pagelet content or the rest of the portal.

Thus, for a user to operate within the PeopleSoft Interaction Hub product, a pagelet developed outside of PeopleSoft can credit the source developer, but must follow these standards:

- Branding on a homepage pagelet should be done by means of text only and should be placed at the bottom of the pagelet. No organization logos can be placed on the pagelet without permission from Oracle. You can use “Provided by *XXX*,” where *XXX* is the name of your company. Also, *XXX* can be a link to an appropriate web page.
- Organization logos and further information about your organization, products, and services should be located on the pagelet personalization page or other pagelet extensions.
- Generally, graphics for organization logos should not be larger than 100 pixels wide x 40 pixels high.

If you wanted to create a pagelet that an associate could use to search for keywords on another website, it might look like the following example.

The HTML used is shown here:

```
<table border=0 cellspacing=1 cellpadding=0>
  <tr>
   <td><!-- start of search form -->
    <form method="POST" action="http://www.example.com/cgi-bin/dofind.cgi">
     <center>
     <font class="PSEDITBOXLABEL">Enter keyword:</font>
     <input name="word" value="">
     <br>
     <input type="submit" value="Search"></center>
    </form>
<!-- end of search form --></td>
</tr>

<tr>
  <td><!-- start of optional directions -->
   <font class="PAADDITIONALINSTRUCTIONS">General directions can be provided here.
    Remember that pagelets should be as easy and intuitive to use
    as possible.<br>
    <br>
   </font>
<!-- end of optional directions --></td>
</tr>

<tr>
  <td><!-- start of footer for branding -->
   <center>
   <font class="PSPSMALLTEXT">Provided by
    <a href="http://www.example.com/">Content Source Company</a>
   </font>
   </center>
<!-- end of footer for branding --></td>
</tr>
</table>
```

Note how this sample pagelet adheres to the standards mentioned previously:

- It uses a table to help wrap words, rather than allowing long strings to dictate the width of the pagelet.
- All text uses the PeopleSoft styles, such as PSEDITBOXLABEL, PAADDITIONALINSTRUCTIONS, and PSPSMALLTEXT.
- The branding is small and no graphics are used, thus adhering to the branding standards mentioned previously.

Understanding Single Signon and Pagelets

This section discusses:

- Single signon.
- The PeopleSoft authentication process.
- The PeopleSoft authentication API.

Single Signon

The examples shown thus far have used publicly-available URLs. Even if the examples represent third-party applications, the discussion has concentrated on retrieving data and rendering a pagelet. The topic has not yet discussed the possible need to sign in to a non-PeopleSoft system.

When two or more systems need separate authentication, automating the sign in process is preferable. Needing to manually sign in to several different systems each day is inconvenient and annoying. Users often expect a business portal to be similar to accessing a variety of internet websites. Once signed in to the portal, a user should rarely (if ever) need to sign in to another system.

Implementing single signon between PeopleSoft and other systems can be done in several ways. First, you need to determine the primary (or master) and secondary authentication systems.

- PeopleSoft system as primary.

  Once a PeopleSoft user has signed in, an authentication cookie is sent to the browser's memory. Other applications can choose to authenticate using this cookie. Oracle provides an API that other applications can leverage. This is the option that is discussed in detail in this section.

- PeopleSoft system as secondary.

  The PeopleSoft authentication process is flexible enough to enable accessing another system.

  See “Enabling Signon PeopleCode for LDAP Authentication” (Security Administration).

- PeopleSoft system and other applications as secondary to third-party authentication system.

  A variant of the previous option would be for all applications (including PeopleSoft) to leverage, or trust, a third-party authentication system such as Netegrity, Oblix, or Securant.

  If you are writing a pagelet for the PeopleSoft portal, you have no guarantee that all possible customers for the pagelet will have access to a third-party authentication system. Thus, this option is not discussed in this PeopleBook.

PeopleSoft Authentication Process

Before discussing how your pagelet could leverage PeopleSoft authentication, you need to understand the authentication process.

After the first application server or node authenticates a user, the PeopleSoft application delivers a web browser cookie containing an authentication token. PeopleSoft Pure Internet Architecture uses web browser cookies to store a unique access token for all users after they are initially authenticated. When the user connects to another PeopleSoft application server/node, the second application server uses the token in the browser cookie to reauthenticate the user in the background so that they do not have to complete the sign-in process again. Your non-PeopleSoft application could do something similar.

Single signon is critical for PeopleSoft portal implementations because the portal integrates content from various data sources and application servers and presents them in a unified interface. When users sign in through the portal, they always take advantage of single signon. Users need to sign in once and be able to navigate freely without encountering numerous sign-in screens.

The following table presents the fields that appear in the PeopleSoft authentication token.

Field	***Description***
User ID	Contains the user ID of the user to which the server issued the token. When the browser submits this token for single sign-on, this is the user that the application server signs in to the system.
Language Code	Specifies the language code of the user. When the system uses this token for single sign-on, it sets the language code for the session based on this value.
Date and Time Issued	Specifies the date and time that the token was first issued. The system uses this field to enforce a time-out interval for the single sign-on token. Any application server that accepts tokens for sign-in has a *time-out minutes* parameter configured at the system level. A system administrator sets this parameter using the Single Signon page. The value is in Greenwich Mean Time (GMT), so the application server's time zone is irrelevant.
Issuing System	Specifies the name of the system that issued the token. When it creates the token, the application server retrieves this value from the database. Specifically, it retrieves the defined local node. Single sign-on is not related to PeopleSoft Integration Broker messaging, except that single sign-on functionality leverages the messaging concept of nodes and local nodes. You configure only a node to trust single sign-on tokens from specific nodes. Consequently, an application server needs an Issuing System value so that it can check against its list of trusted nodes to determine whether it trusts the issued token.

Field	***Description***
Signature	Contains a digital signature that enables the application server using a token for single sign-on to ensure that the token hasn't been tampered with after it was originally issued. The system issuing the token generates the signature by concatenating the contents of the token (all of the fields that appear in this table) with the message node password for the local node. The system then hashes the resulting string using the SHA1 hash algorithm for example, `signature = SHA1_Hash ( UserID + Lang + Date Time issued + Issuing System + Local Node Pswd )` **Note:** The "+" indicates concatenation. Only one way is available to derive the 160 bits of data that make up the signature, and that is by hashing exactly the same user ID, language, date time, issuing system, and node password. **Note:** If you are using digital certificate authentication, the signature of the digital certificate occupies this space. The preceding description applies only to using password authentication.

Note: Single signon does not depend on the use of an LDAP (Lightweight Directory Access Protocol) directory. You can store user credentials in an LDAP directory if desired, but it is not required.

These are the key security features of the cookie authentication token:

- The cookie exists in memory; it is not written to disk.
- No password is stored in the cookie.
- You can set the expiration of the cookie to be a matter of minutes or hours. This expiration option is a useful security feature.
- The cookie is encrypted and digitally signed using a checksum to prevent tampering.

The PeopleSoft Authentication API

Oracle delivers a component interface named PRTL_SS_CI, which enables external applications to seamlessly integrate a single sign-on solution with the PeopleSoft portal applications. This component interface helps ensure that users who have already signed in to the portal don't have to sign in again for every system that you reference in your portal.

Component interfaces are the focal points for externalizing access to existing PeopleSoft components. They provide real-time synchronous access to the PeopleSoft business rules and data associated with a component outside the PeopleSoft online system. Component interfaces can be viewed as *black boxes* that encapsulate PeopleSoft data and business processes, and hide the details of the structure and implementation of the underlying page and data.

To take advantage of the Single Signon API, you need to create a custom API, which includes building the dynamic link libraries, classes, and registry settings necessary to enable an external application to communicate with the portal. This can be done automatically through PeopleTools. More information about building dynamic link libraries, classes, and registry settings, as well as other details about PeopleSoft component interfaces, can be found in *PeopleTools: Component Interfaces*

See Understanding the Authentication DomainSecurity Properties.

Only external applications, such as COM, C/C++, or Java programs require a component interface API. PeopleCode programs do not require a component interface API and, in fact, Oracle does not recommend building a component interface API if the component interface is to be accessed from PeopleCode only.

The files of your custom API need to reside on the client machine, that is, the web server for ASP and the machine running the Java program for Java. The registry file may also need to be run to update the registry with the new libraries.

The Signon Process with the API

The PRTL_SS_CI component interface contains two user-defined methods:

- **Authenticate( ).**

  Your external authentication program distributes an authentication token that can be retrieved from a cookie in the browser. The Authenticate function determines whether an authentication token is valid.

- **GetUserID( )**.

  If the token is valid, you use the GetUserID function to retrieve the user ID associated with the authentication token.

Before reading about the development requirements of your API, examine the steps that occur internally when you use the API in conjunction with the delivered PRTL_SS_CI:

1. The user enters the user ID and password into the PeopleSoft portal sign-in page.
2. If the sign-in to the portal application server is successful, the server generates a single signon token. The web server receives the single sign-on token from the application server and issues a cookie to the browser.
3. The user navigates in the portal and encounters a link to the external system. The user clicks the link.
4. The browser passes the PS_TOKEN cookie to the external web server.
5. The external web server checks for the PS_TOKEN cookie before displaying a sign-in page.
6. After the system determines that the user is accessing the application through the PeopleSoft portal, you retrieve the authentication token and send it to the PRTL_SS_CI component interface to verify authentication. For instance, it calls PRTL_SS_CI.Authenticate(Auth. token string).
7. After the system authenticates the token, it can then make calls to the PRTL_SS_CI.Get_UserID() function to return the appropriate user ID.

In general, cookies are not transferable across domains. The only domain that can access the cookie is the domain that created it. Therefore, the web server for the non-PeopleSoft system must be on the same domain as the PeopleSoft system so that the cookies are passed appropriately.

External Application Support for Single Sign-on

Developers of external applications need to alter the sign-on process to conform to the following requirements:

1. Check for the PS_TOKEN cookie. If the cookie doesn't exist, continue with the normal sign-in process. Otherwise, bypass the sign-in page.
2. Retrieve the authentication token from the PS_TOKEN cookie.
3. Connect to the PeopleSoft portal through the PRTL_SS_CI API.
4. Pass the authentication token to the **Authenticate()** function of the API.
5. If **Authenticate()** returns *True,* retrieve the user ID associated with the authentication token using the Get_UserID() function.

Authentication API PeopleCode Example

The following PeopleCode example shows the process of validating your authentication token and retrieving the user ID. This example is designed to provide a general idea of the process involved and to help you incorporate the PRTL_SS_CI API into your sign-in process.

```
Local ApiObject &THISSESSION;
Local ApiObject &THISCI;
Local string &AUTHTKN;
/* Assigns the Authentication Token to a variable */
  &AUTHTKN = %AuthenticationToken;
/* Open a session and make a connection */
  &THISSESSION = GetSession();
   If &THISSESSION.connect(1, "EXISTING", "", "", 0) <> True Then
    WinMessage(MsgGet(30000, 1, "Session Connect Failed."));
    Exit (1);
   End-If;
/* Retrieves the component interface PRTL_SS_CI */
  &THISCI = &THISSESSION.GetCompIntfc(CompIntfc.PRTL_SS_CI);
/* Checks to see if the component interface is NULL */
   If &THISCI = Null Then
    WinMessage("Component Interface PRTL_SS_CI not found.
    Please ensure Component Interface Security access is granted
    to this user.");
    Exit (1);
   End-If;
/* Key fields would usually be set before the Get() function is
called in order to map the component interface to a particular
set of data. This component interface is not mapped to data.
* Therefore, the component interface is retrieved and then the
user defined methods are retrieved */
  &THISCI.get();
  PRTL_AUTH = &THISCI.Authenticate(&AUTHTKN);
  PRTL_USER_ID = &THISCI.Get_UserID();
```

Note: The component interface is not mapped to data because the key field for the data would be the authentication token. This token is dynamically assigned when the user signs in to the portal, and it is not stored as data anywhere in the system. Therefore, no key fields are needed and the token is passed directly to the user-defined functions.

Authentication API Java Example

Here is an example of a similar operation written in Java. This is a file named SingleSignon.java.

```
package examples.migration.sso;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import psft.pt8.joa.*;
import PeopleSoft.Generated.CompIntfc.*;

public class SingleSignon extends HttpServlet {
  public static ISession oSession;
  String psfttoken;
  public static void ErrorHandler() {
    //***** Display PeopleSoft Error Messages *****
if (oSession.getErrorPending() || oSession.getWarningPending()) {
     IPSMessageCollection oPSMessageCollection;
     IPSMessage oPSMessage;

     oPSMessageCollection = oSession.getPSMessages();
for (int i = 0; i < oPSMessageCollection.getCount(); i++) {
      oPSMessage = oPSMessageCollection.item(i);
      if (oPSMessage != null)
       System.out.println("(" + oPSMessage.getMessageSetNumber() + ","
 + oPSMessage.getMessageSetNumber() + ") : " + oPSMessage.getText());
     }
     //***** Done processing messages in the collection;
OK to delete *****
     oPSMessageCollection.deleteAll();
    }
 }

  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    try {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    Cookie[] cookies = request.getCookies();
     if (cookies == null) {
       out.println("<TR><TH COLSPAN=2>No cookies");
     } else {
       Cookie cookie;
       for(int i=0; i<cookies.length; i++) {
         cookie = cookies[i];

         String pstoken = cookie.getName();
            psfttoken = cookie.getValue();
            if (pstoken.equals ("PS_TOKEN"))

               out.println("<TR>\n" +
                   "  <TD>" + pstoken + "\n" +
                   "  <TD>" + psfttoken);
       }

     }

     String strServerName, strServerPort, strAppServerPath;
  String strUserID, strPassword;
  strServerName = "jfinnon09";
                strServerPort = "9500";
                strUserID = "VP1";
                strPassword = "password";

  //Build Application Server Path
  strAppServerPath = strServerName + ":" + strServerPort;
```



```
        //***** Create PeopleSoft Session Object *****
        //       ISession oSession;

     oSession = API.createSession();

     //***** Connect to the App Server *****
     if (!oSession.connect(1, strAppServerPath, strUserID, strPassword,
    null)) {
      out.println("\nUnable for Jason to Connect to
  Application Server.");
      ErrorHandler();
      return;
     }

        //***** Get Component Interface *****
     IPrtlSsCi oPrtlSsCi;
     String ciName;
      ciName = "PRTL_SS_CI";
     oPrtlSsCi = (IPrtlSsCi) oSession.getCompIntfc(ciName);
     if (oPrtlSsCi == null) {
      out.println("\nUnable to Get Component Interface " + ciName);
      ErrorHandler();
      return;
     }

     //***** Set the Component Interface Mode *****
       oPrtlSsCi.setInteractiveMode(false);
       oPrtlSsCi.setGetHistoryItems(true);
       oPrtlSsCi.setEditHistoryItems(false);

     //***** Set Component Interface Get/Create Keys *****

       //***** Execute Get *****
       if (!oPrtlSsCi.get()) {
        out.println("\nNo rows exist for the specified keys.
    \nFailed to get the Component Interface.");
        ErrorHandler();
        return;
       }

       //***** BEGIN:  Set/Get Component Interface Properties *****

       //***** Set Level 0 Properties *****
       //out.println("oPrtlSsCi.PrtlToken: " +
    oPrtlSsCi.getPrtlToken());
       //out.println("Checking token: " + psfttoken);

       //oPrtlSsCi.setPrtlToken(<*>);
                    //String psfttoken;
                    //psfttoken = oPrtlSsCi.getPrtlToken();
                    //System.out.println("oPrtlSsCi.PrtlToken: "
    + psfttoken);

       //***** END:  Set Component Interface Properties *****

       //***** Execute Standard and Custom Methods *****
       //***** Execute Authenticate *****
       boolean auth;

              auth = oPrtlSsCi.authenticate(psfttoken);
              //out.println("Auth: " + auth);
       //Execute Get_UserID
                    String psftuser;
       psftuser = oPrtlSsCi.getUserid();
                  //out.println("Psftuserid: " + psftuser);

                   String title = "Welcome  " + psftuser;

                  //****** Get HTML ***********
                   out.println(ServletUtilities.headWithTitle(title));
```

```
                 out.println("</BODY></HTML>");

                //***** Disconnect from the App Server *****
          oSession.disconnect();
          return;
  }


  catch (Exception e) {
    e.printStackTrace();
    System.out.println("An error occurred: ");
    ErrorHandler();
   }
 }


  /** Let the same servlet handle both GET and POST. */

  public void doPost(HttpServletRequest request,
                     HttpServletResponse response)
      throws ServletException, IOException {
    doGet(request, response);
  }

}
```

Developing PeopleSoft Component-Based Pagelets

To develop pagelets that are based on PeopleSoft components:

1. Open PeopleSoft Application Designer.
2. Select **File** > **Open** to display the Open Definition dialog box.
3. Use the drop-down list box to select *Page* as the definition.
4. Enter selection criteria and open the desired page.
5. Select **File** > **Definition Properties** and then select the Use tab.
6. Select the page size.

   For narrow pagelets, select *240xVar portal home page comp.*

   For wide pagelets, select *490xVar portal home page comp.*

   For banner pagelets, select *Custom size* then enter these width and height dimensions: *1193* x *259.*

   Note: You can adjust the pagelet dimensions at any time in the process.

7. Design the new pagelet using the guidelines described previously and similar design techniques that are used to design a page.
8. Save the page.

Note: When the page is added to a component, the search record is specified. As noted in the preceding guidelines, you should avoid a search interaction so that the pagelet can render its data on the homepage without any prompting for keys. If keys are needed, then the pagelet should be coded to use some default key values and the personalization options should initially reflect these defaults.

---

9. Register the pagelet in the portal registry. This step also involves setting up security access to the component and the page.

   See Administering Pagelets in the Portal Registry.

Developing iScript-Based Pagelets

Use an iScript only for pagelets that reference non-PeopleSoft data or if a PeopleSoft page does not provide the necessary functionality. When you develop pagelets that are based on iScripts, you should be aware of the advantages and disadvantages.

Advantages

These are some advantages of iScripts:

- You may be familiar with the technique because it's similar to creating Active Server Pages (ASP) or Java Server Pages (JSP). Basically, you are using a script to produce HTML and JavaScript.
- Greater flexibility is available in pagelet appearance because you have direct control of the HTML or JavaScript.
- Greater flexibility is available in pagelet sizing. PeopleSoft Internet Architecture pages tend to be fairly static in their object placement and sizing.
- Many PeopleSoft Application Designer definitions and their attributes are available for you to take advantage of programmatically. For example, almost all PeopleSoft pages share a consistent appearance through the use of style sheets. Your pagelet, too, can use these styles and thus maintain uniformity of appearance with your PeopleSoft application because iScript-based pagelets can access the same style sheet definitions and their properties,

Disadvantages

These are some disadvantages of iScripts:

- You must manually program all aspects of the transaction. You must handle and account for support for multiple browsers and international considerations, such as language, date and currency formatting, and so on. PeopleTools handles these details automatically on transaction pages and component-based pagelets.
- You must be extremely meticulous and thoughtful. The poor design and implementation of many complex and interrelated programs can lead to loss of functionality and degradation of performance.
- When you develop a template pagelet based on an iScript, some of the functions provided by PeopleSoft Application Designer, such as currency codes, language support, and multiple browser

support, are not automatically available in your iScript. Therefore, if you want any such functionality, you must develop it.

Template Pagelets

To create an iScript-based template pagelet:

Note: Use steps 1 and 2 when you are getting HTML from another website or if you want to create some relatively static HTML that is used by the iScript to render a pagelet. If you are creating a pagelet iScript that is going to render highly dynamic HTML, then begin with step 3 and create any necessary or complex PeopleCode to conditionally render the HTML or JavaScript.

---

1. Create the HTML code for the pagelet.

   In many cases you can navigate to a Uniform Resource Locator (URL) that you want to turn into a pagelet and copy the HTML associated with that web page. For form-based web pages, copy all of the HTML code located between the Form tags of an existing HTML document.

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

3. Create an iScript that calls the new HTML definition in a web library.

   A web library is a derived record that is created in PeopleTools for storing iScripts. The name of all web library records begin with WEBLIB_.

   Note: All iScripts should be coded in the FieldFormula PeopleCode event of a derived record called WEBLIB_*XX,* where *XX* is the product code. Only functions that start with *IScript_,* such as IScript_iTracker, can be invoked from a URL.

   ---

4. Find to the appropriate field name and edit the FieldFormula PeopleCode that is associated with the field.

   This is where you create an iScript that calls the HTML definition that you created.

5. Create a new iScript and give it a meaningful name.

   An easy way to create a new iScript is to copy and paste an existing iScript that is located in the web library and then make the desired changes. All iScript names must begin with *IScript_*.

6. Save the web library.

7. Register the iScript as a pagelet in the portal registry. This step also involves setting up security access to the iScript.

   See Administering Pagelets in the Portal Registry.

Modifying Security for the Web Library

Security for the web library can be set up initially by means of the Registration Wizard. You can use the following steps to modify the security settings created by the wizard.

To modify security:

1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists.**

2. Open the desired permission list, such as *ALLPAGES* or *PAPP9000.*

3. Go to the Web Libraries tab.

4. Select the **Edit** link next to the appropriate web library.

5. Modify the web library permissions as desired by changing the **Access Permissions** drop-down option. The choices are *Full Access* and *No Access.* Click **OK** to return to the main Permission List page.

6. Click **Save** to save your changes to the permission list.

Related Links

“Setting Web Library Permissions” (Security Administration)
“Understanding the Registration Wizard” (Application Designer Developer’s Guide)
“iScript Security” (PeopleCode API Reference)

Developing Java-Based Pagelets

This section provides overviews of how to develop pagelets in Java and Java pagelet development considerations and discusses how to:

- Write business logic in Java.
- Use Java to write business logic for a PeopleSoft or non-PeopleSoft system.
- Invoke PeopleSoft components from Java.

Understanding Java Development

Java is an extremely popular programming language today, especially for internet applications. PeopleTools provides support in several key areas to enable developers to create Java-based applications. Though the PeopleSoft development environment allows you to write PeopleSoft business logic in Java, the focus of the following sections is to provide information that is relevant to the developer who is integrating applications with the PeopleSoft Interaction Hub. An example is development that is being done to integrate your non-PeopleSoft application with the PeopleSoft Interaction Hub. Another example is a scenario in which you want to pull together data from your non-PeopleSoft system along with data from your PeopleSoft system for integration with the PeopleSoft Interaction Hub.

Two primary areas are available in which you can program in Java within the PeopleTools environment:

1. Writing business logic in Java.

   You can use Java used to write business logic for PeopleSoft or non-PeopleSoft applications that are to be rendered in the PeopleSoft Interaction Hub. Java can be called from PeopleCode and the Java program can then reference the PeopleSoft runtime environment as needed.

   In addition, Pagelet Wizard facilitates Java development in that it enables the Java programmer to concentrate on business logic, while Pagelet Wizard handles all the development aspects of rendering the application in the portal framework without any coding necessary.

2. Invoking PeopleSoft components from Java.

   All PeopleSoft components can be invoked from Java programs through the component interface technology. This is useful if you want to create Java servlets, applets, JSP programs, or Enterprise JavaBeans that invoke PeopleSoft business logic.

Another consideration when developing applications with Java is that any application developed by means of Java that is URL-addressable and produces HTML can be easily integrated into the PeopleSoft Interaction Hub through the PeopleSoft Interaction Hub registry functionality. This enables you to distribute your development and runtime environments making for better scalability for both.

For more information about leveraging the PeopleSoft Interaction Hub registry functionality to integrate your Java application with the PeopleSoft Interaction Hub, see the following documentation.

See Administering Pagelets in the Portal Registry.

These areas of Java programming within the PeopleTools development environment are discussed further in the following sections.

Describing Java Pagelet Development Considerations

This section discusses the following Java development elements:

- Business rules.
- User interface.
- User personalization.
- Navigation.
- Navigation on extended pages.
- Portal integration.

Business Rules

Java can be used to apply business logic to data to create a result set. Pagelet Wizard can then be used to invoke the Java program. Parameters that are required by the Java program can be defined, managed, and passed to the Java program through Pagelet Wizard. These parameters may be derived from user personalization parameters, as well as administrator-defined parameters and system variables.

For example, the Java program may need the user's ID, the user's language, and the user's personalization selection regarding whether to include historical transaction information or just current information. These parameters can be defined in Pagelet Wizard in relation to a Java class. Pagelet Wizard can access the PeopleSoft runtime environment to determine the user's ID as well as other profile information, such as the user's language.

See Understanding System Variables Supported as Data Source Parameters.

Pagelet Wizard will also manage the user interface and storage of user personalization data. This topic is discussed in detail in the subsequent User Personalization section. When the Java program is invoked, Pagelet Wizard will handle passing all required parameters to the Java program.

In addition, Java classes delivered with PeopleTools enable you to call PeopleCode from your Java program and access contextual information from the runtime system. If needed, business data can be retrieved, as necessary, through the numerous PeopleCode functions that are accessible from Java.

The result set created by the Java program may be in an HTML format or in an XML format. Pagelet Wizard will accept the output from the Java program and will manage the presentation of the information within the portal. This topic is discussed in the following User Interface section.

User Interface

Pagelet Wizard will manage all aspects of a pagelet's user interface. No user-interface-related programming for the pagelet is required in the Java program. Pagelet Wizard can accept an HTML- or XML-based result set from a Java program. The HTML from the Java program can be rendered directly. Alternatively, XML can be transformed with XSL (Extensible Stylesheet Language) to create an appropriate pagelet user interface.

In either case, Pagelet Wizard will manage the XSL and invoke the transformation at runtime. XSL templates are provided with Pagelet Wizard for general use scenarios. XSL development may be required, depending on the specific user interface required for a pagelet.

See Defining Pagelet Wizard XSL Prototypes.

User Personalization

Pagelet Wizard supports the persistence of user personalization selections for a pagelet application and handles all aspects of the user interface that enable users to enter their preferences and selections. Personalization parameters are defined within Pagelet Wizard administration pages. No programming is required, Java or otherwise.

User personalization selections can occur at two levels. From the homepage, users can select which pagelet applications they want to include on their homepage. After they've selected a pagelet application and have decided where they want the pagelets to appear on their homepage, they can personalize a particular pagelet application.

Examples of user personalizations at the second level can include scenarios in which a user selects transactions from a particular region or chooses to include transactions using a particular date as opposed to same-day transactions.

The user interface for all of these activities, as well as the storage of the personalization data for persistence, is managed by Pagelet Wizard without any additional programming, Java or otherwise.

Pagelet Wizard supports functionality that enables an end user to select a pagelet application to appear on his or her homepage. When a pagelet application is registered with Pagelet Wizard, a folder, which represents a portal application category, can be selected. When users personalize their homepage by selecting specific pagelet applications that they want to appear on their homepage, the selection is presented within the category (folder) that is defined when they register the pagelet application through Pagelet Wizard.

See Step 6: Specifying Pagelet Publishing Options.

Pagelet Wizard also supports functionality that enables an end user to select values that allow a specific pagelet application to be personalized. The functionality may dictate that some input parameters that are required by a Java program be entered by an end user, while other parameters can be set by an administrator. Pagelet Wizard manages parameter passing and the user interface. Pagelet Wizard inspects the Java source code for input parameters. Pagelet Wizard administrators can then select which parameters can be accessed by an end user for personalization. The user interface to enables an end user to personalize a pagelet application and update the appropriate Java input parameters is automatically generated and managed by Pagelet Wizard.

Navigation

When creating links in your pagelet application that enable a user to navigate to a related page for details and so forth, the link may need to be proxied by the portal to keep the user within the portal. For example, if a pagelet link is not proxied, then the pagelet can take a user to a site that is completely independent from the portal. In this scenario, the navigational links on the portal header, the homepage menu pagelet, and the left-hand menu pagelet will not be available to the user.

If the link is proxied by the portal, then the link can access and display content from a site that is independent of the portal. However, the content will be rendered within the portal. For example, the portal's header and left-hand menu will be available.

With the PeopleSoft Interaction Hub, links that are included in a pagelet will be altered automatically to support proxying by the portal. No programming is required, Java or otherwise.

Note: Content that is referenced by the link must be HTML-based to be rendered appropriately in the portal.

Navigation on Extended Pages

Clicking a link on an extended page displays content within the target frame. The PeopleSoft Interaction Hub, header, and left-hand menu area will remain. Therefore, the new content will be rendered in the context of the PeopleSoft Interaction Hub.

Note: Content that is referenced by the link must be HTML-based to be rendered appropriately in the portal.

The **Return to Home** link, which returns the user to the portal homepage, may be required from an extended page. The GenerateHomepagePortalURL function creates a URL string that represents an absolute reference to the portal homepage. Because you can access the PeopleSoft runtime environment from a Java program and you can access PeopleCode built-in functions, you can invoke the GenerateHomepagePortalURL function from your Java program.

See Developing Java-Based Pagelets.

See "GenerateHomepagePortalURL" (PeopleCode Language Reference)

Portal Integration

The pagelet application needs to be integrated with the portal registry so that the pagelet application is managed, secured, and presented through the PeopleTools portal. Registration makes the portal aware of a pagelet. Information included in registration determines the name and security for your pagelet application. In addition, registration data determines the category in which the pagelet application resides. Users view pagelet applications within these categories when selecting items that they want to see on their homepage. No programming is required, Java or otherwise, to complete portal registration.

Writing Business Logic in Java

Many reasons exist why you would want to write business logic for your non-PeopleSoft applications (and even your PeopleSoft applications) in Java. Perhaps you have licensed a third-party set of Java classes to do some very specific processing (tax calculation, for example). Or perhaps you have developed some internal Enterprise JavaBeans for your specific business processes. Or perhaps you simply like to code in Java. Whatever the reason, you can easily integrate Java code with your applications through the Java PeopleCode functions.

Invoking Java from PeopleCode

Three primary Java PeopleCode functions are used to invoke Java from PeopleCode:

- GetJavaClass
- CreateJavaObject
- CreateJavaArray

Java PeopleCode Function	***Usage***	***Example***

GetJavaClass	Finds a Java class that you can manipulate in PeopleCode. This is used for those classes that have static members for which instantiating an object of the class isn't appropriate. You can call only static methods, that is, class methods, with the object created with this function.	In Java, you access such static members of a class by using the class name:
		
		`result = java.class.name.SomeStaticMethod();`
		
		In PeopleCode, you use the following code:
		
		`&Result = GetJavaClass("java.class.name").SomeStaticMethod();`
		
		The following example is simple PeopleCode that uses GetJavaClass to get a system class:
		
		`&Sys = GetJavaClass("java.lang.System");`
		`&Sys.setProperty("java.security.policy", "C:⇒`
		`\java\policy");`
		`WinMessage("The security property is: " | &Sys.getProperty("java.security.policy"));`
		`&Props = &Sys.getProperties();`
		`&Props.put("java.security.policy", "C:\java\⇒`
		`policy");`
		`&Sys.setProperties(&Props);`
		`WinMessage("The security property is: " | &Sys.getProperty("java.security.policy"));`

<table>
<tr><th>Java PeopleCode Function</th><th>Usage</th><th>Example</th></tr>
<tr><td>CreateJavaObject</td><td>Creates a Java object that can be manipulated in your PeopleCode. You can use the CreateJavaObject function to create a Java array. If ClassName is the name of an array class (ending with [ ]), ConstructorParams are used to initialize the array.</td><td>In Java, use the following code to initialize an array:

<code>intArray = new int[]{1, 2, 3, 5, 8, 13};</code>

In PeopleCode, you use the following code to initialize a Java array:

<code>&IntArray = CreateJavaObject("int[]", 1, 2, ⇒</code>

<code>3, 5, 8, 13);</code>

If you want to initialize a Java array without knowing the number of parameters until runtime, use the CreateJavaArray function.

The following example is a simple PeopleCode program that creates a Java object from a sample program that generates a random password:

<code>/* Example to return Random Passwords from a⇒</code>

<code> Java class */</code>
<code>Local JavaObject &oGpw;</code>
<code>/* Create an instance of the object */</code>
<code>&oGpw = CreateJavaObject("com.PeopleSoft.Ran⇒</code>

<code>dom.Gpw_Demo");</code>
<code>&Q = "1";</code>
<code>/* Call the method within the class */</code>
<code>&NEW_VALUE = &oGpw.getNewPassword(&Q, PSRNDM⇒</code>

<code>PSWD.LENGTH);</code>
<code>/* This is just returning one value for now ⇒</code>

<code>*/</code>
<code>PSRNDMPSWD.PSWD = &NEW_VALUE;</code></td></tr>
</table>

<table>
<tr><th>Java PeopleCode Function</th><th>Usage</th><th>Example</th></tr>
<tr><td>CreateJavaArray</td><td>Enables you to create a Java array without knowing the number or value of the elements.

When you create an array in Java, you already know the number of elements in the array. If you don't know the number of elements in the array, but you want to use a Java array, use the CreateJavaArray function in PeopleCode. This will create a Java object that is a Java array, and you can pass in the number of elements that are to be in the array.</td><td>The following PeopleCode example passes a PeopleCode array of strings (&Parms) to a Java method xyz of class Abc. This example assumes that when you are writing the code, you don't know how many parameters you will have.

<code>Local JavaObject &Abc, &RefArray;
Local array of String &Parms;
&Parms = CreateArray();
/* Populate array how ever you want to popul⇒

ate it */
&Abc = GetJavaObject("com.peoplesoft.def.Abc⇒

");
/* Create the java array object. */
&JavaParms = CreateJavaArray("java.lang.Stri⇒

ng[]", &Parms.Len);
/* Populate the java array from the PeopleCo⇒

de array. */
&RefArray = GetJavaClass("java.lang.reflect.⇒

Array");
For &i = 1 to &Parms.Len
 &RefArray.set(&JavaParms, &i - 1, &Parms[&i⇒

]);
End-For;
/* Call the method. */
&Abc.xyz(&JavaParms);</code></td></tr>
</table>

Accessing the PeopleSoft Runtime System from Java

After a Java class has been invoked, the class can access the PeopleSoft runtime system. Java classes delivered with PeopleTools enable you to call PeopleCode from your Java program and access contextual information from the runtime system, such as the current user’s role and language preference. By importing the PeopleTools-delivered Java classes in your Java program, you can have access to PeopleCode objects and methods. Hundreds of PeopleSoft system variables, constants, and built-in functions are available for use with this approach. Discussions of the various methods follow.

Accessing the runtime system works only from a Java program that was initially called from PeopleCode. You must call PeopleCode facilities only from the same thread that was used for the call into Java. You cannot call any PeopleCode facility that would cause the server to return to the browser for an end-user action because the state of the Java computation cannot be saved and restored when the action is complete.

Java Class	***Usage***
SysVar	Use to refer to PeopleSoft system variables, such as %Language or %Oprid. For example, %Session, becomes SysVar.Session(). See "System Variables" (PeopleCode Developer's Guide).
SysCon	Use to refer to system constants, such as %SQLStatus_OK or %FilePath_Absolute. For example, %CharType_Matched becomes SysCon.CharType_Matched. See "Constants" (PeopleCode Developer's Guide).

Name	Enables you to use the PeopleSoft-reserved item references. This enables you to reference pages, components, records, fields, and so forth.
	
	For example, you can refer to a record field using the following PeopleCode:
	
	`REC_NAME.FIELD_NAME`
	
	With the Name class, you can use a similar construct:
	
	`new PeopleSoft.PeopleCode.Name("REC_NAME", "FIELD_NAME");`
	
	As another example, you can refer to a PeopleSoft page using the following PeopleCode:
	
	`Page.PAGE_NAME`
	
	In Java, it would be:
	
	`new PeopleSoft.PeopleCode.Name("Page", "PAGE_NAME");`
	
	See "Definition Name References" (PeopleCode Developer's Guide).

Func	Use to refer to built-in PeopleCode functions, such as CreateRowset or GetFile. For example, SetLanguage(LANG_CD) becomes Func.SetLanguage(LANG_CD). The existing PeopleCode classes (Array, Rowset, and so forth) have properties and methods that you can access from Java. PeopleCode classes have the same name, so Record becomes Record, SQL becomes SQL, and so forth. Methods are accessed by the method name. The name of a property is prefixed with either *get* or *set,* depending on whether you're reading or writing to the property. For example, to get the IsChanged property would be getIsChanged. To set the value for a field would be &MyField.setValue.

<table>
<tr><th>Java Class</th><th>Usage</th></tr>
<tr><td>Func (continued)</td><td>
Here is an example of a Java program that uses PeopleCode objects to access the database:
<pre><code>/* This code is used to test the Java/PeopleCode interface. */
import PeopleSoft.PeopleCode.*;
public class Test {
/*
* Add up and return the length of all the
* item labels on the UTILITIES menu,
* found two different ways.
*/
public static int Test() {
/* Get a Rowset to hold all the menu item records. */
Rowset rs = Func.CreateRowset(new Name("RECORD", "PSMENUITEM"), n⇒

ew Object[]{});
String menuName = "UTILITIES";
int nRecs = rs.Fill(new Object[]{"WHERE FILL.MENUNAME = :1", menu⇒

Name});
int i;
int nFillChars = 0;
for (i = 1; i <= rs.getActiveRowCount(); i++) {
String itemLabel = (String)rs.GetRow(i)
.GetRecord(new Name("RECORD", "PSMENUITEM"))
.GetField(new Name("FIELD", "ITEMLABEL"))
.getValue();
nFillChars += itemLabel.length();
}
/* Do this a different way - use the SQL object to read each menu
item record. */
int nSQLChars = 0;
Record menuRec = Func.CreateRecord(new Name("RECORD", "PSMENUITEM⇒

"));
SQL menuSQL = Func.CreateSQL("%SelectAll(:1) WHERE MENUNAME = :2"⇒

,
new Object[]{menuRec, menuName});
while (menuSQL.Fetch(new Object[]{menuRec})) {
String itemLabel = (String)menuRec
.GetField(new Name("FIELD", "ITEMLABEL"))
.getValue();
nSQLChars += itemLabel.length();
}
return nFillChars + 100000 * nSQLChars;
}
}</code></pre>
</td></tr>
<tr><td>Func (continued)</td><td>
This can be run from PeopleCode in the following way:
<pre><code>Local JavaObject &Test;
Local number &chars;
&Test = GetJavaClass("Test");
&chars = &Test.Test();
&Test = Null;
WinMessage("The character counts found are: " | &chars, 0);</code></pre>
See PeopleCode API Reference and PeopleCode Language Reference.
</td></tr>
</table>

Using Java to Write Business Logic for a PeopleSoft or Non-PeopleSoft System

This section presents an example of pagelet application development for which the intention is that all programming be done in Java. Business logic might be for a PeopleSoft system or a non-PeopleSoft system. In this development example, PeopleTools programming is not required. All programming is done with Java.

The PeopleTools infrastructure can be leveraged without any PeopleTools programming through the use of PeopleSoft Interaction Hub features that support administration, integration, and implementation for the portal. The ability to leverage the PeopleTools infrastructure can significantly reduce the amount of Java programming that is required to develop a pagelet application for use in the PeopleSoft Interaction Hub.

Some of the information in this section is covered in other Pagelet Wizard documentation. The main difference in this section is that it presents the information as a development scenario as opposed to a feature description. Also, the ability to access the PeopleSoft runtime environment from Java is discussed in this section.

The following development scenario addresses the following elements of portal application development, integration, and implementation:

- Business rules.

  Apply business logic and rules to data to create a result set.

- User interface.

  Create the appropriate presentation format for a set of business data.

- User personalization.

  - User personalization data.

    Allow pagelet applications to be selected by the end user to be placed on his or her homepage. Also, allow a user to personalize a specific pagelet application. These selections should persist.

  - User interface.

    Create the appropriate presentation format for personalization data.

- Navigation for a pagelet.

  Create links on your pagelet to enable a user to navigate to a related page for details and so forth.

- Navigation on extended pages.

  An extended page is a page that a user can get to from a homepage pagelet. Extended pages might show details as well as have links to enable a user to navigate to other related pages.

  - Links.

    Create links from your extended page to enable a user to navigate to a related page for further details and so forth.

- Return links.

  Create links that enable the user to return to the portal's homepage.

- Portal integration and implementation.

  Integrate the pagelet application with the portal registry so that the pagelet application is managed, secured, and presented through the PeopleTools portal.

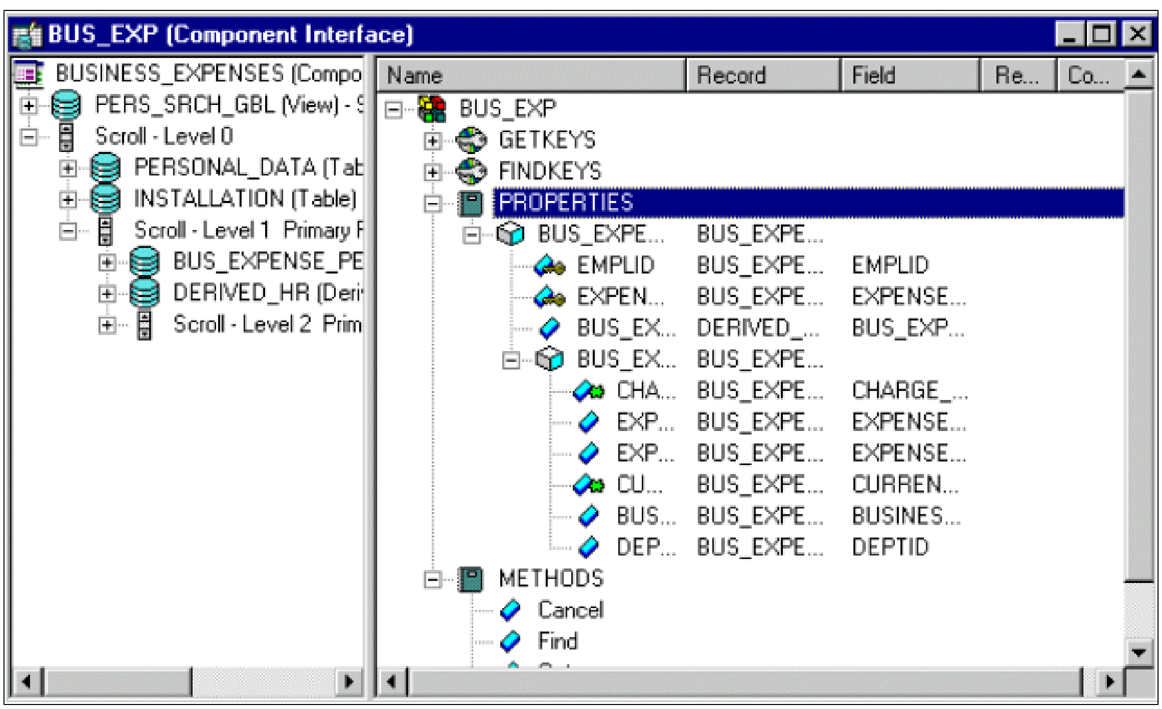
Invoking PeopleSoft Components from Java

All PeopleSoft components can be invoked from Java programs through component interface technology. This is useful for developers who want to create Java servlets, applets, JSP programs, or Enterprise JavaBeans that invoke PeopleSoft business logic. This section provides an example of how to invoke a PeopleSoft component from Java. A Business Expense component is used as the example.

Creating the Component Interface

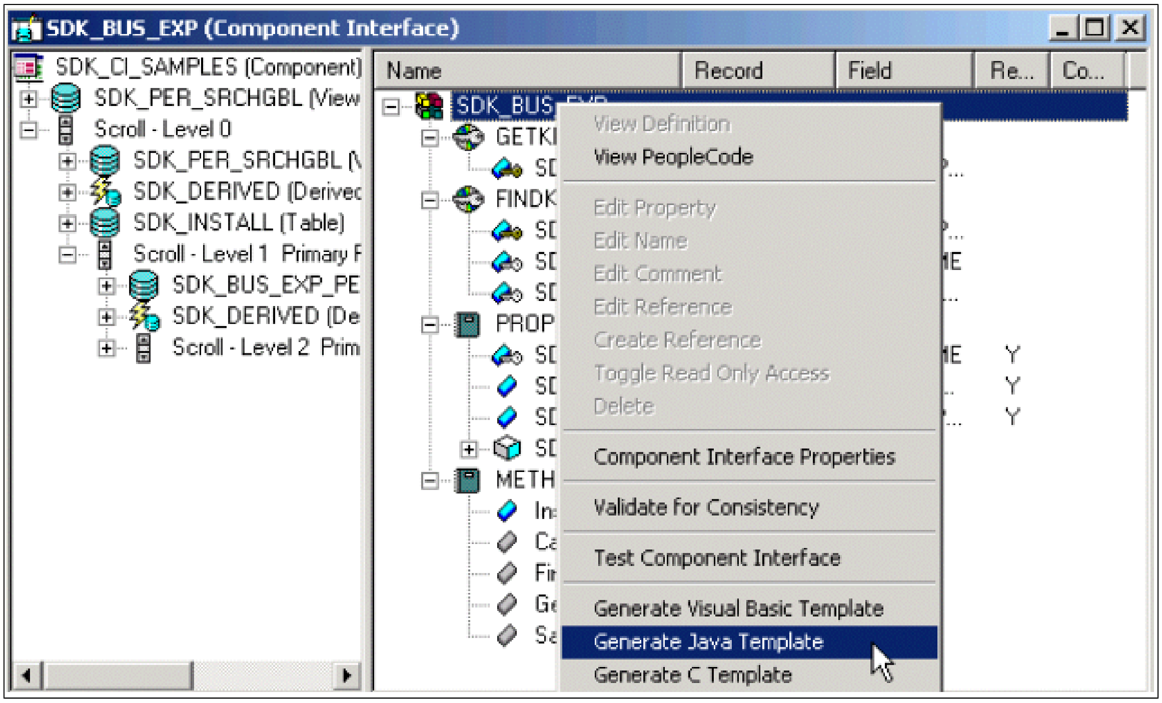
To expose a component to a third party, you must first create a component interface definition. You do this through the PeopleSoft Application Designer. Using drag-and-drop functionality, you can specify the properties and methods of the component that you want to expose. Numerous component interface definitions are delivered with the each PeopleSoft application.

The BUSINESS_EXP component definition appears in the left frame in the following example. The properties and methods that are exposed through this interface are displayed in the right frame.



After the component interface definition is saved, you can then generate the Java classes for invoking this interface. You do this using the Generate Java Template command in PeopleSoft Application Designer:

This example illustrates generating the Java classes for the SDK_BUS_EXP component interface.



See “Understanding Component Interface Class” (PeopleCode API Reference).

Invoking the Component Interface from Java

To invoke the Business Expense component interface from Java.

1. Connect to the application server.

   To access a component interface, you need to establish a PeopleSoft session. To create a session object, use the Session.Connect () method. The Connect method, which takes five parameters, actually signs in to a PeopleSoft session. The **Connect()** method connects a session object to a PeopleSoft application server. Note that various options are available for using an existing connection and disconnecting and switching connections.

```
import PeopleSoft.ObjectAdapter.*;
import PeopleSoft.Generated.PeopleSoft.*;
import PeopleSoft.Generated.CompIntfc.*;
private ISession oSession;
private CAdapter oAdapter;
oAdapter = new CAdapter();
oSession = new CSession(oAdapter.getSession());
oSession.Connect(1,"//EHARRIS032000:9000","PTDMO","PTDMO",new byte[0]);
```

2. Get an instance of the component interface.

   Use the GetComponent() method with a session object to get an instance of a previously created component interface.

```
busExpense = new CBusExp( oSession.GetComponent( "BUS_EXP" ));
```


3. Find an existing record.

   You can query a component interface to find relevant data instances based on primary and alternate search keys.

```
busExpense.setName( searchDialogStrings[ 0 ]);
busExpense.setLastNameSrch( searchDialogStrings[ 1 ]);
busExpense.setEmplid( searchDialogStrings[ 2 ] );
return( busExpense.Find() );
```

4. Get an instance of data.

   GetKeys are the key values required to return a unique instance of existing data. GetKeys can be set by means of simple assignment to the properties of the component interface and then the Get() method can be invoked. This populates the component interface with data based on the key values that you set; this has been referred to here as a data instance.

```
busExpense.setEmplid( getKey );
boolean result = busExpense.Get();
```

5. Migrate through collections of data.

   After getting a data instance, get access to the data in the component interface. PeopleSoft organizes component interface data within collections. Rows of data in a collection are called items.

   The following code creates a connection to the application server, gets the component interface, and fetches the first item in a collection.

```
oAdapter = new CAdapter();
oSession = new CSession(oAdapter.getSession());
oSession.Connect(1,"//EHARRIS032000:9000","PTDMO","PTDMO",new byte[0]);
busExpense = new CBusExp( oSession.GetComponent( "BUS_EXP" ));
busExpense.setEmplid( getKey );
boolean result = busExpense.Get();
busExpenseFirstScrollItemCollection = busExpense.getBusExpensePer();
busExpenseFirstScrollItem = firstScrollCollection.Item
( firstScrollIndex );
return( busExpenseFirstScrollItem.getBusExpenseDtl() );
```

6. Edit and access data in an item.

   Editing and accessing component interface data in Java is rather straightforward. The following Java code accesses the various public members of the class.

```
long j = busExpenseSecondScrollCollection.getCount();
Object [][] data = new Object[ ((int)j + 1) ][ 7 ];
for( int i = 1; i < j + 1 ; i++ )
{
busExpenseSecondScrollItem = busExpenseSecondScrollCollection.Item( i );
data[(i - 1)][0] = busExpenseSecondScrollItem.getBusinessPurpose();
data[(i - 1)][1] = busExpenseSecondScrollItem.getChargeDt();
data[(i - 1)][2] = busExpenseSecondScrollItem.getCurrencyCd();
data[(i - 1)][3] = busExpenseSecondScrollItem.getDeptid();
data[(i - 1)][4] = busExpenseSecondScrollItem.getExpenseAmt();
data[(i - 1)][5] = busExpenseSecondScrollItem.GetPropertyByName("ExpenseCd");
data[(i - 1)][6] = busExpenseSecondScrollItem.GetPropertyByName("CurrencyCd");
}return( data );
```

   In the following example, data is accessed by means of the **get***Property_Name***()** method of an item or by means of the generic **getPropertyByName()** method. This code illustrates the way in which an entire collection of data can be captured and packaged into an object for transfer to a calling object.

```
busExpenseFirstScrollItem.setEmplid( emplid );
busExpenseFirstScrollItem.setExpensePeriodDt( expensePeriodDt );
```

```
return( busExpense.Save() );
```

Just as before, data is edited by means of item objects and the setNameOfPropery() method of those items. Also note that you needed to call the Save() method on the component interface to commit the changes.

7. Insert an item into a collection and delete an item from a collection.

   Collection objects in Java contain the InsertItem() method in which the return value is the item that has just been inserted. After a new item is created, simply edit data in it and then remember to call the Save() method to commit the changes.

   ```
   busExpenseSecondScrollItem = busExpenseSecondScrollCollection.
   InsertItem( secondScrollIndex );
   ```

   Similarly, a DeleteItem() method is available:

   ```
   busExpenseSecondScrollCollection.DeleteItem( secondScrollIndex );
   ```

8. Disconnect from a session.

   After a session is no longer needed, it should be disconnected from the application server. This is done by calling the Disconnect() method on the session object.

   ```
   oSession.Disconnect();
   ```

Developing Contextual Embeddable Pagelets

This section discusses how to develop contextually relevant pagelets that you embed in transaction pages.

1. Create an embeddable pagelet using Pagelet Wizard.

   See Understanding Pagelet Wizard Pagelet Types.

2. Create a transaction page definition using Application Designer.

   See “Creating New Page Definitions” (Application Designer Developer’s Guide).

3. Place an HTML area on the page definition.

   See “Inserting HTML Areas” (Application Designer Developer’s Guide).

4. Write a PeopleCode function and map the pagelet parameters with any available values from the component buffer.

PeopleCode Sample for Rendering the Context-Based Embeddable Pagelet

This PeopleCode example uses application packages to create an embeddable pagelet.

```
 /* Import the Pagelet Wizard application package to create the embeddable pagelet*⇒

/

      import PTPPB_PAGELET:*;
       import PTPPB_PAGELET:UTILITY:*;
           Component object &Pagelet, &myDataSource;
```



```
           /* Create the Pagelet Wizard application pakage and assign the pagelet I⇒

D*/
           &PWAPI = create PTPPB_PAGELET:PageletWizard();
           &PageletID = "EMBEDED_PAGELET";

           /* Get the pagelet's pointer by passing the pagelet id*/
           &Pagelet = &PWAPI.getPageletByID(&PageletID, False);
           &myDataSource = &Pagelet.DataSource;

           /* Set the pagelet parameters to default values*/
           &Pagelet.PopulateInputsWithDefaults();

           /* Read the pagelet parameters */
           &DSParamColl = &myDataSource.getParameterCollection();
           &CollectionParamArray = &DSParamColl.getCollectionAsArray();

           /* To override the pagelet parameter default values, */
     /*  read the CollectionParamArray and set the parameter values */
     /*  from the component buffer based on the business requirement */
           If &CollectionParamArray.Len > 0 Then
            For &i = 1 To &CollectionParamArray.Len
              If (&i = 1) Then
                &DSParameter = &CollectionParamArray [&i];
                &CollectionParamArray [&i].value = PSOPRDEFN.OPRDEFNDESC;
              End-If;
            End-For;
           End-If;

           /* Get the Embeddable Pagelet HTML */
           &PgltHTML = &Pagelet.Execute();

           /* Associate the Pagelet HTML with the HTML Area */
           PSUSRPRFL_WRK.HTMLAREA = "<div class='PSTEXT'>" | &PgltHTML | "</div>";
```

Administering Pagelets in the Portal Registry

This section discusses how to:

- Register homepage and template pagelets.
- Modify pagelet attributes and security.
- Register URL-based pagelets.
- Register pagelet extensions.

Note: When you use Pagelet Wizard to register pagelets in the portal registry, Pagelet Wizard automates much of the process by interacting with the portal registry. You supply some key information in the pagelet definition, such as pagelet name, title, folder, and so on; Pagelet Wizard passes this information on to the portal registry.
However, the portal registry encapsulates other metadata about URLs that are accessed by way of the portal. Even if you use Pagelet Wizard to define and initially register a pagelet, you may still need to access the portal registry to update an attribute or to register additional entries.

Related Links

Understanding Portal Administration

Registering Homepage and Template Pagelets

Before you can access a pagelet through the portal or associate a template pagelet with a target or WorkCenter page, you must register the pagelet in the portal registry. You can register pagelets by using the Component Registration Wizard, or by using the following procedure:

To register a new homepage pagelet or template pagelet:

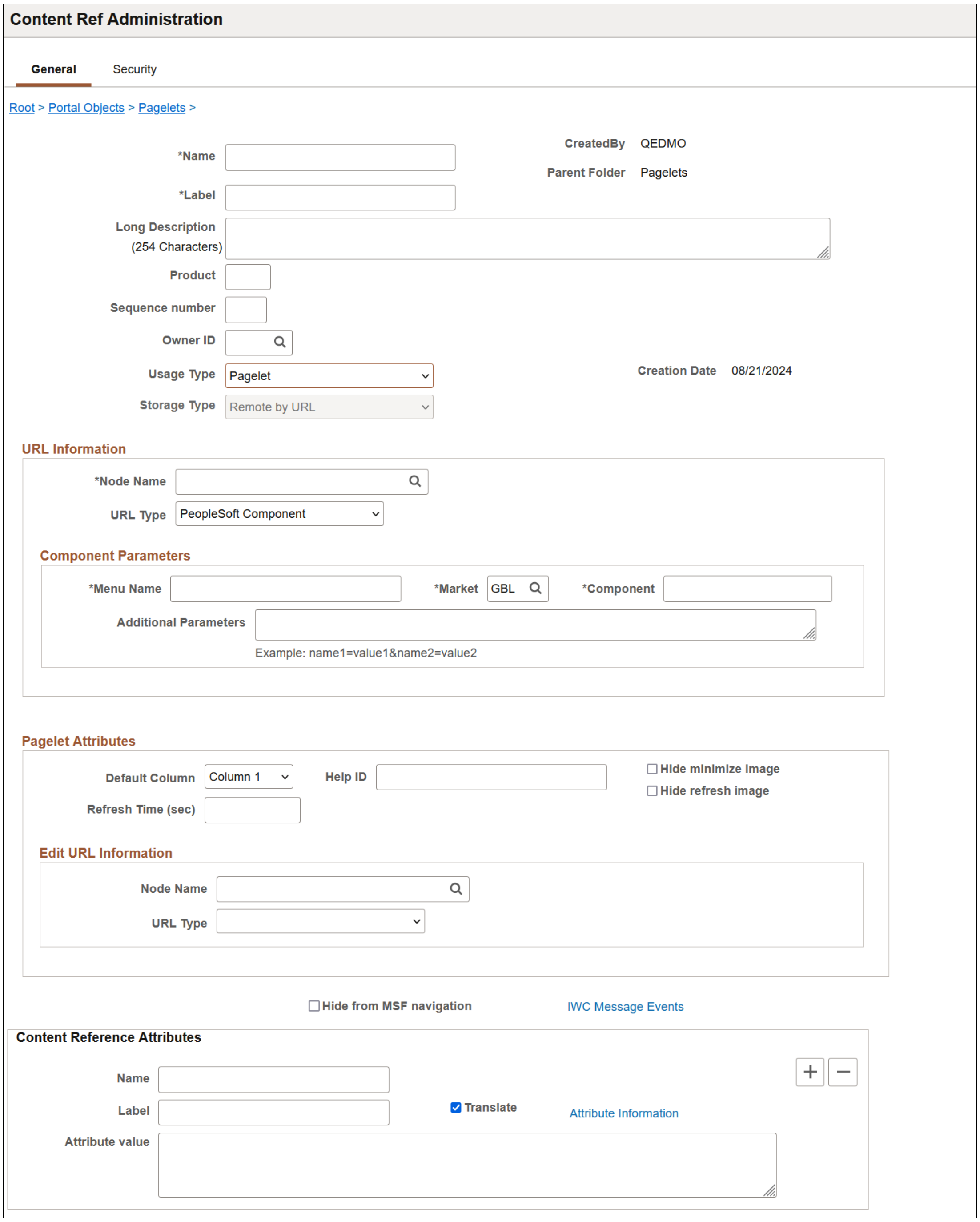
1. Select **PeopleTools** > **Portal** > **Structure and Content.** Then:
   - For template pagelets, click these links in this order: **Portal Objects, Template Pagelets.**

     Template pagelets must be registered in this folder.
   - For homepage pagelets, click these links in this order: **Portal Objects, Pagelets.** Then select any subfolder; for example, you can register a homepage pagelet in the Portal Objects, Pagelets, Organizers folder.
2. Click the **Add Content Reference** link.
   - For template pagelets, select *Target* as the **usage type**.
   - For homepage pagelets, select *Pagelet* as the usage type.
3. Select *Always use Local* as the node name.
4. Select the URL type based on the location of the pagelet.
   - Select *PeopleSoft Component* for component-based template pagelets.
   - Select *PeopleSoft Script* for iScript-based template pagelets.
   - Select *PeopleSoft Generic URL* for Pagelet Wizard-based pagelets.
5. Set pagelet attributes as necessary.
6. Set the content reference attributes as necessary.
7. Set additional parameters, if needed.
8. Save the content reference.

Modifying Pagelet Attributes and Security

Access the Content Ref Administration page. (Select **PeopleTools** > **Portal** > **Structure and Content.** Click the **Edit** link for a content reference.)

This example illustrates the fields and controls on the Content Ref Administration page for a pagelet.

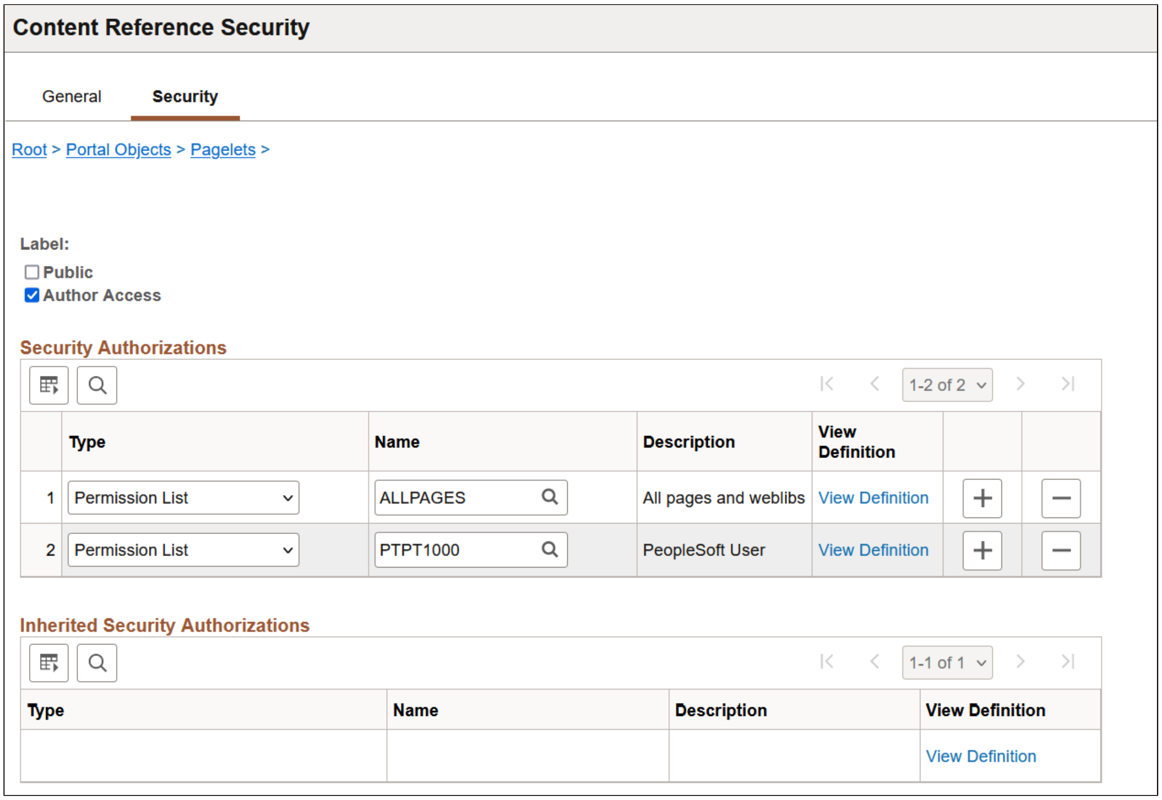


The portal registry stores every content reference that is available through the portal. The label and description defined here affect the appearance of the label and description in the navigation menu. Other attributes on this page affect the URL that will be used to reference the content associated with a definition.

See Understanding Portal Administration.

Access the Content Reference Security page. (Select **PeopleTools** > **Portal** > **Structure and Content.** Click the **Edit** link for a content reference. Select the Security tab.)

This example illustrates the fields and controls on the Content Reference Security page for a pagelet.

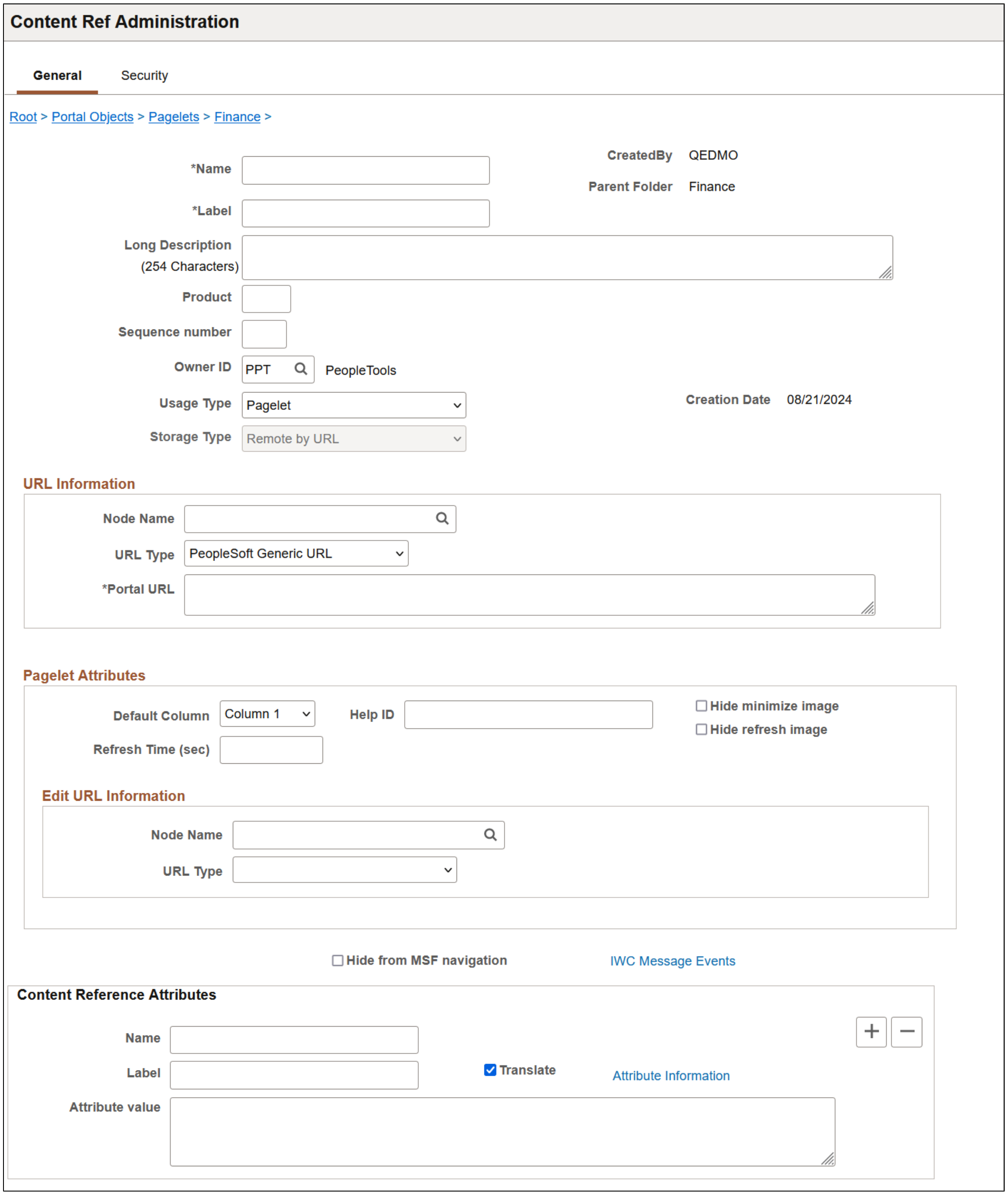


You can grant and review the permission lists and roles that provide access to homepage tabs and template pagelets on the Content Reference Security page. The page allows for any combination of permission lists and roles. You can also make the object publicly available or available to the author only, which can be useful for testing purposes. Permission lists and roles can be used to grant access to the content item associated with this definition.

See Understanding Portal Administration.

Registering iScript-Based Pagelets

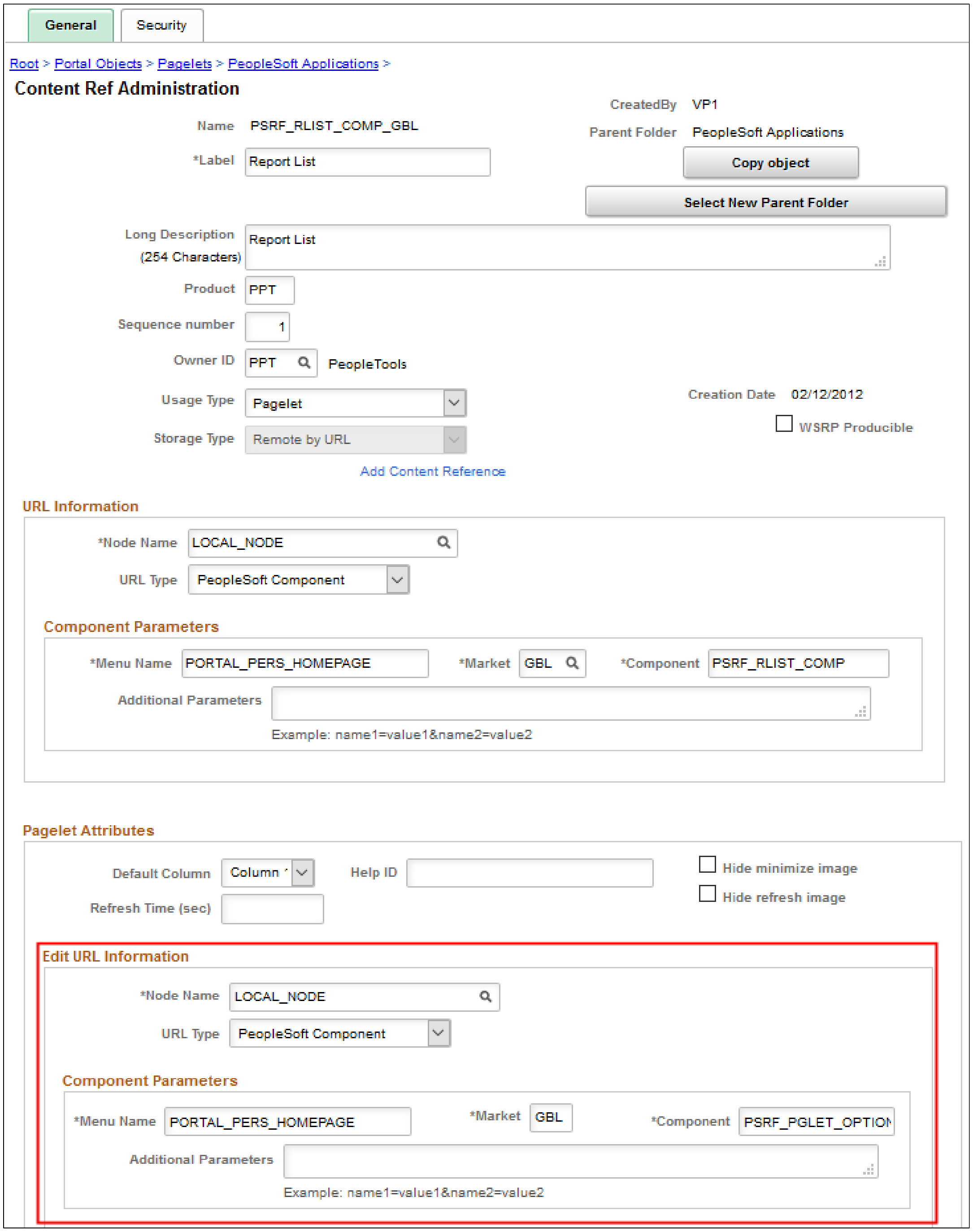
This example illustrates the fields and controls on the Content Ref Administration page for an iScript-based pagelet.



Specifying a Component for Pagelet Personalization

In the content reference definition for a pagelet, you can also specify a component to be used for pagelet personalization, which allows users to explicitly personalize the data shown in the pagelet by setting options on a personalization page.

In this example of the content reference definition for the Report List pagelet, the edit URL information within the Pagelet Attributes group box identify the node, URL type, and component containing the personalization page for this pagelet.



Using Attributes to Enhance Pagelets

This section provides an overview of pagelet and content reference attributes and discusses how to:

- Manage pagelet attributes.

- Configure pagelet Help links.
- Configure pagelet time-outs.
- Configure automatic pagelet refresh attributes.
- Create banner pagelets.
- Add images to pagelet headers.

Understanding Pagelet and Content Reference Attributes

Pagelet and content reference attributes provide a framework to specify and store free-form information about a content reference or pagelet. Attributes also enable you to manage pagelet behavior and appearance.

When you click a content reference, the portal retrieves the URL for the pagelet or content reference, it reads and processes the attribute and its value, and renders the page or pagelet incorporating the attribute.

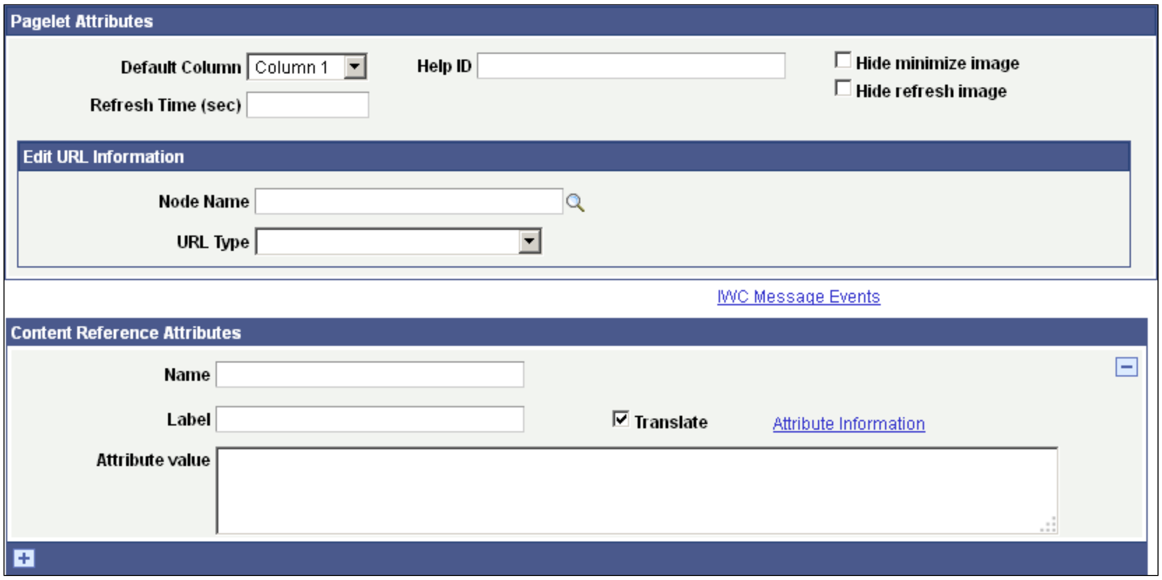
You add attributes by entering Name-Value pairs. For example, an attribute name is PORTAL_BANNER_PGLT. You use this attribute to define a pagelet as a banner pagelet, which is the only pagelet that can span the enter width of the transaction page that appears in the browser in your PeopleSoft applications. The possible attribute values for this attribute are 1, 2, and 3. Each number represents different options regarding the appearance of the banner pagelet.

Managing Pagelet Attributes

Access the Content Ref Administration page. (Select **PeopleTools** > **Portal** > **Structure and Content.** Click the **Edit** link for the appropriate content reference.)

1. Select *Pagelet* in the **Usage Type** drop-down list box.
2. Use the **Pagelet Attributes** group box to select the default column for a pagelet, associate a help topic with the pagelet, set refresh time, and hide certain buttons from users.
3. Use the **Content Reference Attributes** group box to create banner pagelets and add images to pagelet headers

Use the Pagelet Attributes and Content Reference Attributes group boxes to enhance pagelets.



Pagelet Attributes

Field or Control	***Description***
Default Column	Select the column in which the pagelet appears.
Help ID	Enter the help ID for the pagelet. When users click the Help link, the PeopleSoft system opens a new window displaying online help documentation for this item. **Note:** You must also configure a help URL on the Web Profile Configuration - General page.
Refresh Time (sec)	Enter the number of seconds after which the system auto-refreshes the pagelet. To activate the refresh timer, enter a value of *1* or greater. The timer refreshes the pagelet as soon as the user accesses the homepage and then automatically refreshes the pagelet every *n* number of seconds. **Note:** Homepages and dashboards use automatic refresh.
Hide minimize image	Select to hide the minimize button that normally appears in the pagelet header so that users are prevented from minimizing the pagelet.

Field or Control	*Description*
Hide refresh image	Select to hide the refresh button to prevent users from refreshing the pagelet. If you implemented pagelet caching for this pagelet, a refresh button automatically appears in the pagelet header.
Force Pagelet Refresh	Some components can render themselves differently based on the column width. When a homepage has a two-column layout and the user moves a component-based pagelet to a different sized column by using the drag-and-drop method, this pagelet needs to be refreshed so that the content displays correctly. The drag-and-drop iScript that persists the pagelet changes forces a homepage refresh only when all of the following conditions are met: • A pagelet changes columns. • The homepage layout is two-column. • The pagelet is component-based. • You select the **Force page refresh** property for the pagelet.

Note: When `document.write` or `document.writeln` commands are detected inside a pagelet, the whole homepage will be refreshed instead of just that pagelet. This is required to properly assemble that pagelet's content.

You can use a combination of a web profile setting and the PORTAL_NOBROWSERCACHE content reference attribute to determine whether an individual pagelet can be cached. See Configuring Caching and Implementing Homepage Caching.

Separately, you can also administer server-based caching. See Implementing Pagelet Caching.

Edit URL Information

Use the **Node Name**, **URL Type**, and additional URL-specific fields to specify a page to be used for personalizing this pagelet and to make the personalization button appear in the pagelet header. See Specifying a Component for Pagelet Personalization for an example.

Field or Control	*Description*
Node Name	Select the node name. This node name can be different from the pagelet node name. **Note:** In a clustered environment, the node name should not be defined as *LOCAL_NODE* and must be set to the actual portal host node for the pagelet.
URL Type	Select the type of edit URL: • *Non-PeopleSoft URL:* The **Edit URL** field appears. Enter the URL of the personalization page to use. • *PeopleSoft Component:* The **Component Parameters** group box appears, containing the same fields as the **Component Parameters** group box for the content reference. Use these fields to identify the personalization page to use. • *PeopleSoft Script:* The **iScript Parameters** group box appears, containing the same fields as the **iScript Parameters** group box for the content reference. Use these fields to identify the personalization page to use. The additional URL entry fields appear based on the setting of this field.

Content Reference Attributes

You can use the content reference attributes of a pagelet to change its character and appearance and to improve its usability.

Use content reference attributes to:

- Configure pagelet help links.
- Configure pagelet time-out settings.
- Configure automatic pagelet refresh.
- Suppress parallel pagelet loading (for pagelets with global variables).
- Create banner pagelets.
- Add icons to pagelet headers.

Related Links

Defining Content References
Configuring Pagelet Help Links
Configuring Pagelet Time-out Settings

Configuring Automatic Pagelet Refresh
Creating Banner Pagelets
Adding Images to Pagelet Headers

Configuring Pagelet Help Links

To specify pagelet help:

1. Select **PeopleTools** > **Portal** > **Structure and Content**.
2. Click the Portal Objects link.
3. Click the Pagelets link.
4. Navigate to the Content Ref Administration page for the pagelet.

   You define the pagelet help in the **Content Reference Attributes** region of the page.
5. Enter *PTPPB_PAGELET_HELP_LINK* in the **Name** field.
6. Leave the **Label** field blank.
7. Deselect the **Translate** check box.

   Note: You *must* deselect the **Translate** check box or unexpected results might occur.
8. In the **Attribute Value** field, enter the URL of the location of your help document, for example *http://www.example.com/help_doc1.html*
9. Save the content reference.

Note: The pagelet Help URL takes precedence over the Help ID when both attributes are set for a pagelet.

Note: The Help button appears only when the pagelet is on the homepage inside the PeopleSoft Portal; it does not appear when the pagelet is displayed as a WSRP portlet.

Related Links

Administering Content References

Configuring Pagelet Time-out Settings

To specify a pagelet time-out:

1. Select **PeopleTools** > **Portal** > **Structure and Content**.
2. Click the Portal Objects link.
3. Click the Pagelets link.
4. Navigate to the Content Ref Administration page for the pagelet.

   You define the pagelet help in the **Content Reference Attributes** region of the page.

5. Enter *PSTIMEOUT* in the **Name** field.

6. Enter *PSTimeout* in the **Label** field.

7. In the **Attribute Value** field, enter the number of seconds before the pagelet is considered unavailable. This can be any positive integer.

   Note: The portal ignores this attribute if you enter *0* or a negative value, or if the content reference isn't a pagelet.

   ---

8. Save the content reference.

When your specified time-out expires, the portal stops attempting to load the pagelet and generates an error message.

Related Links

Administering Content References
Configuring Look and Feel

Configuring Automatic Pagelet Refresh

To specify a pagelet refresh time in seconds:

1. Select **PeopleTools** > **Portal** > **Structure and Content**.

2. Click the Portal Objects link.

3. Click the Pagelets link.

4. Navigate to the Content Ref Administration page for the pagelet.

   You define the pagelet help in the **Content Reference Attributes** region of the page.

5. Enter the number of seconds for automatic refresh in the **Refresh Time (sec)** field.

6. (Optional) Select the **Hide refresh image** check box to prevent manual pagelet refresh by the user.

7. Save the content reference.

Note: Pagelets with document.write or document.writeln must not have the refresh timer set (to automatically refresh every *n* seconds) because the whole homepage will also be refreshed every *n* seconds.

Creating Banner Pagelets

To create banner pagelets:

1. Create and register your pagelet.

   See Administering Pagelets in the Portal Registry.

   See Understanding Pagelet Development.

2. Access the Structure and Content page (Select **PeopleTools** > **Portal** > **Structure and Content**).
3. Select **Portal Objects**, then **Pagelets** in the portal registry folder structure and click the **Add Content Reference** link.
4. If you have already registered the pagelet, navigate to the Pagelets folder and click the **Edit** link that is associated with the pagelet.
5. In the **Content Reference Attributes** group box, enter *PORTAL_BANNER_PGLT* in the **Name** field.
6. In the **Attribute value** field, enter one of these values:

   1 - The banner pagelet will display a pagelet header and a border around the pagelet content like a standard pagelet.

   2 - The banner pagelet will display a border around the pagelet content, but *no* pagelet header.

   3 - The banner pagelet will display *neither* a pagelet header *nor* a border around the pagelet content.
7. Add the pagelet to the homepage.

   See Selecting the Pagelets or Tiles That Appear on Homepages.
8. Click the **Save** button.

Adding Images to Pagelet Headers

To add images to pagelet headers:

1. In PeopleSoft Application Designer, create an image definition and store the image in the image catalog.
2. In your browser, access the Structure and Content page (Select **PeopleTools** > **Portal** > **Structure and Content.**)
3. Navigate through the Portal Objects folder structure to the Pagelets folder until you find the pagelet content reference. Then, click the **Edit** link.
4. In the **Content Reference Attributes** group box, enter *PT_PORTAL_PGLT_HDR_IMG* in the **Name** field.
5. In the **Attribute value** field, enter the name of the image definition that you want to appear in the pagelet header.

   Important! The image definition must exist in the image catalog of the PeopleTools portal database. Use PeopleSoft Application Designer to manage the image definitions that comprise the image catalog.
6. Click the **Save** button.

Related Links

“Understanding Image Definitions” (Application Designer Developer’s Guide)

Handling Special Situations

This section discusses how to:

- Determine pagelet running location.
- Use refresh tags in pagelets.

Determining Pagelet Running Location

You can determine whether your code is running within the portal environment (that is being invoked by means of an https request coming from a portal servlet) as opposed to running in on the system outside of the portal environment. The PeopleCode **%RunningInPortal** system variable returns a Boolean value that identifies the environment. This variable works in both iframe templates and HTML templates.

Using Refresh Tags in Pagelets

The order of precedence of refresh tags in pagelets is:

1. Template
2. Target content
3. Pagelet

Among pagelets, the first pagelet in the HTML to include a refresh tag is included. Subsequently found refresh tags are not included in the HTML.

Related Links

Understanding Portal Caching

Applying Changes

Chapter 17

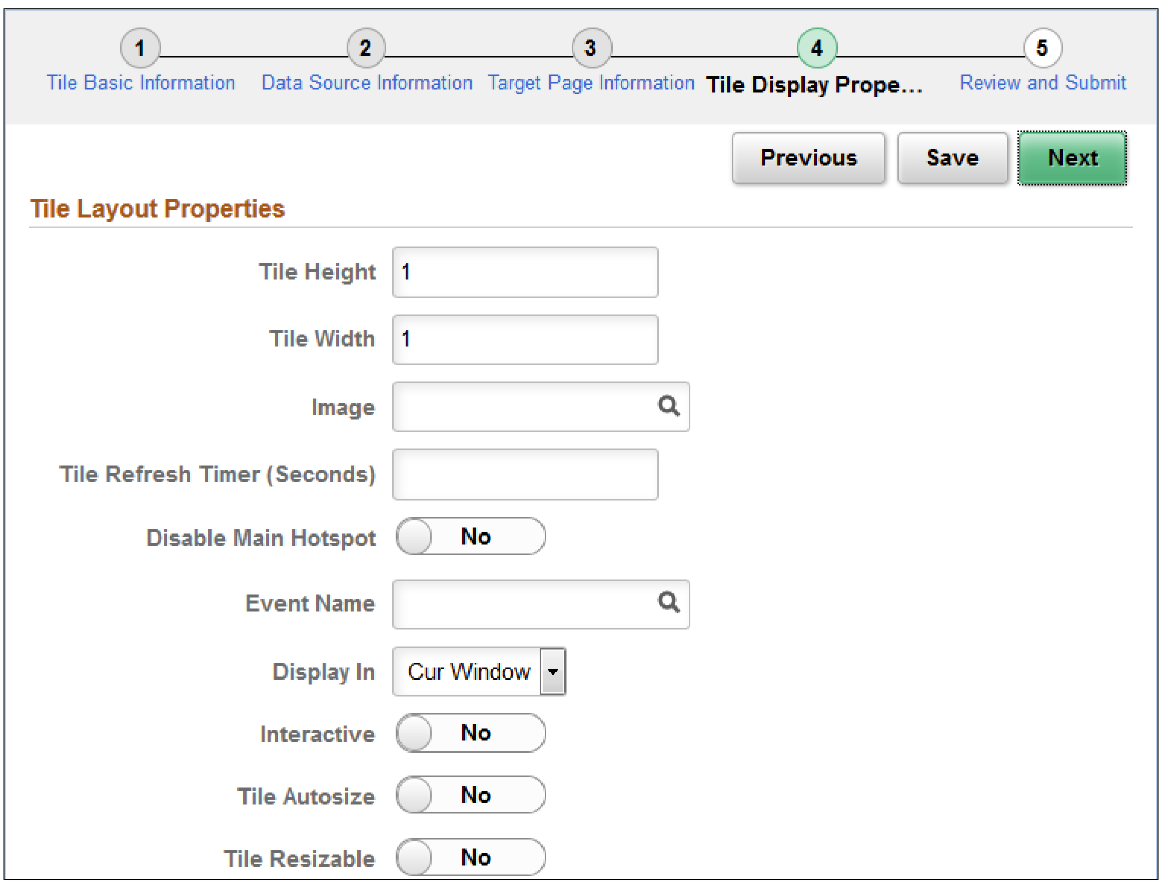
Creating and Managing Tiles

Using Tile Wizard

Understanding Tile Wizard

Tile Wizard provides a user-friendly, browser-based graphical user interface that leads you through the steps involved in creating and publishing a tile. Portal administrators and nontechnical users can use Tile Wizard to integrate and transform data from a variety of data sources.

During the tile creation process, Tile Wizard presents a series of numbered steps. Each step appears in a numbered path at the top of the page to indicate where you are in the tile creation process as shown in the following example:



This labeled path assists you in navigating forward and back through the steps in the wizard, which enables you to review or edit tile definition values. Once you've completed a step, Tile Wizard automatically takes you to the next appropriate step.

Tile Wizard leads you through the following steps that are typically used to create a tile:

1. Specify tile information.
2. Select the data source (and specify data source parameters when applicable).
3. Select the target page and tile repository location.
4. Specify display properties.
5. Review and publish the tile definition.

Related Links

Understanding Tile Definitions and the Tile Repository
PeopleSoft Fluid UX Standards: Tiles

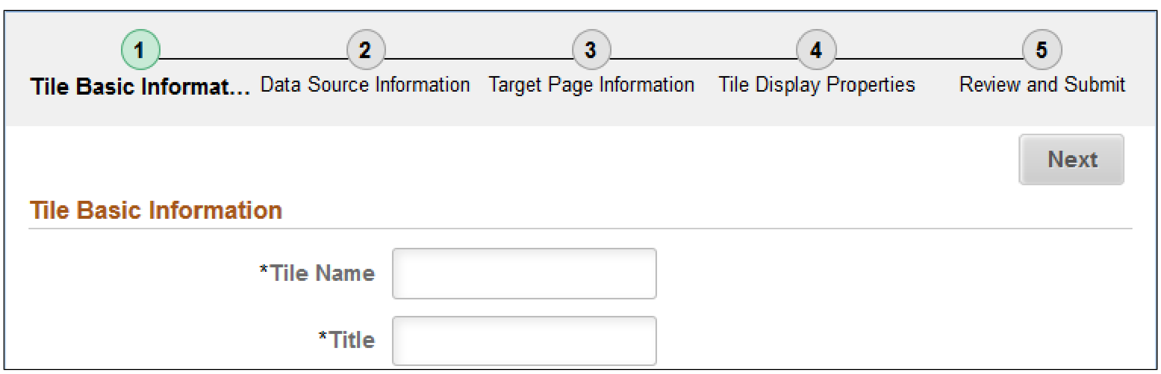
Step 1: Tile Basic Information

Use the Tile Basic Information page to specify the ID and title for the tile.

Navigation:

1. **PeopleTools** > **Portal** > **Tile Wizard**.
2. Click the:
   - Create a new Tile button to create a new tile definition.
   - Edit button for an existing tile to edit the definition.

This example illustrates the fields and controls on the Tile Basic Information page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Tile Name	Enter an ID for the tile definition. Each tile that you build with Tile Wizard must have a unique ID.
Title	Enter a descriptive title for the tile.

Step 2: Data Source Information

Tile Wizard supports the following data types:

Field or Control	***Description***
Application classes	Use the application class data type to specify a custom application class that implements the PTGP_APPCLASS _TILE:Tiles:Tile base class and one of its required SetTileContentAs methods to show charts, HTML, key performance indicators (KPIs), or images as tile content.
Free text	Use the free text data type to present free-form HTML or simple text in a tile.
Images	Use the image data type to create an image-only tile for an existing content reference.
Navigation collections	Use the navigation collections data type to select an existing navigation collection to be displayed as a "fluid navigation collection" in an activity guide or in the NavBar.
OBIEE (Oracle Business Intelligence Enterprise Edition)	Use the OBIEE data type to select an existing OBIEE report or dashboard from the OBIEE server as the data source for the tile.
PeopleSoft queries and SQL definitions	Use the PSQuery/SQL data type to select PeopleSoft queries, stored SQL statements, or a combination to specify the data displayed on the tile. **Note:** If data masking for personally identifiable information or other sensitive data has been enabled on a record field that is part of the criteria for the PeopleSoft query, then these data values are masked when displayed via Tile Wizard tiles.
Pivot grids	Use the pivot grid data type to select existing pivot grid instances, which use interactive pivot tables and charts that display application data that you can filter and pivot across multiple dimensions.

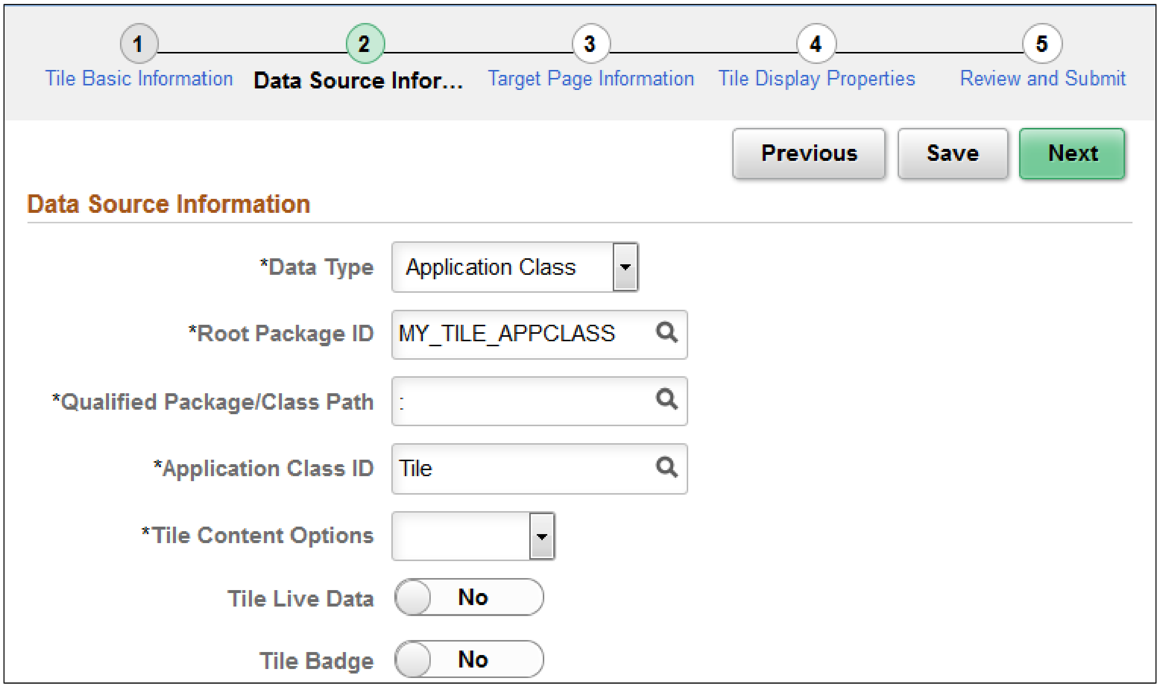
Step 2: Data Source Information - Application Class

Use the Data Source Information page for the Application Class data type to specify which application class data source.

Navigation:

1. On the Tile Basic Information page, click the Next button.
2. Select the *Application Class* data type.

This example illustrates the fields and controls on the Data Source Information page for the Application Class data type. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Root Package ID	Select a custom application package that includes the class that implements the PTGP_APPCLASS_TILE:Tiles:Tile base class and one of its required SetTileContentAs methods.
Qualified Package/Class Path	Select the package name or package path to the implemented class.
Application Class ID	Select the class that implements the required getTileLiveData method.

Field or Control	Description
Tile Content Options	Select the tile content type: • *Chart* – Corresponds to the SetTileContentAsChart method. • *Chart & KPI* – Corresponds to the SetTileContentAsChartAndKPI method. • *Data-1-KPI* – Corresponds to the SetTileContentAsOneKPI method. • *Data-2-KPIs* – Corresponds to the SetTileContentAsTwoKPIs method or the SetTileContentAsTwoKPIsTopBottom method. • *HTML Area* – Corresponds to the SetTileContentAsHTML method. • *Image* – Corresponds to the SetTileContentAsImage method. **Note:** This setting can be overridden in the getTileLiveData method.
Tile Live Data	Select this option to indicate whether the tile displays the live data region. **Note:** This setting can be overridden in the getTileLiveData method.
Tile Badge	Select this option to indicate whether the tile displays the badge region. **Note:** This setting can be overridden in the getTileLiveData method.

Related Links

“Implementing an Application Class for Tile Content” (PeopleCode API Reference)
“Understanding Application Class Tile Content” (PeopleCode API Reference)
“getTileLiveData” (PeopleCode API Reference)

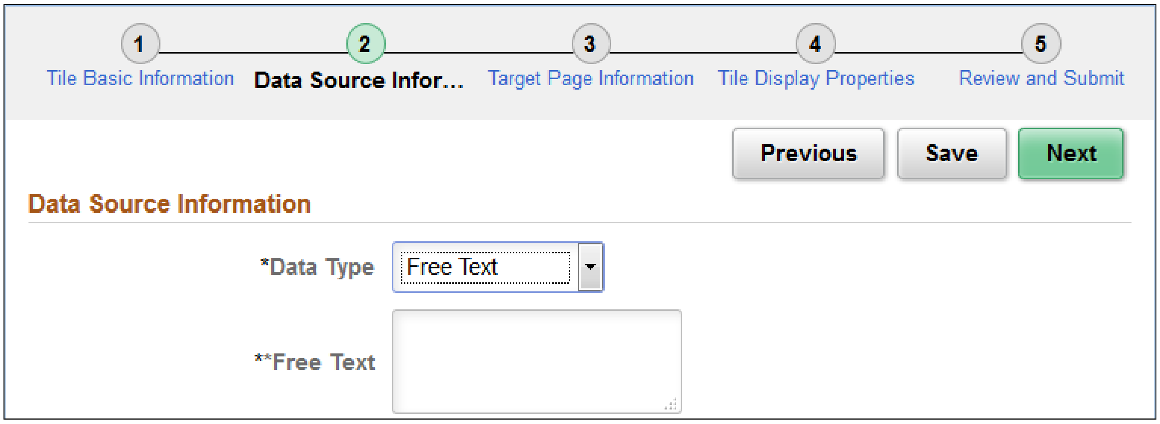
Step 2: Data Source Information - Free Text

Use the Data Source Information page for the Free Text data type to enter free text as the data source.

Navigation:

1. On the Tile Basic Information page, click the Next button.
2. Select the *Free Text* data type.

This example illustrates the fields and controls on the Data Source Information page for the Free Text data type. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Free Text	Use the free text data type to present free-form HTML or simple text in a tile. You can use the following HTML tags only: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , <p>, <b>, <i>, and <img>. Within the <img> tag, src, width, height, and alt are allowable attributes. Use the %Image meta-HTML element to identify an image to include within the HTML: `<img src="%Image(PS_IMAGE_ID)">` **Note:** For translatable text in the free text data type, use %Message(message_set,message_num) MetaHTML for the text content.

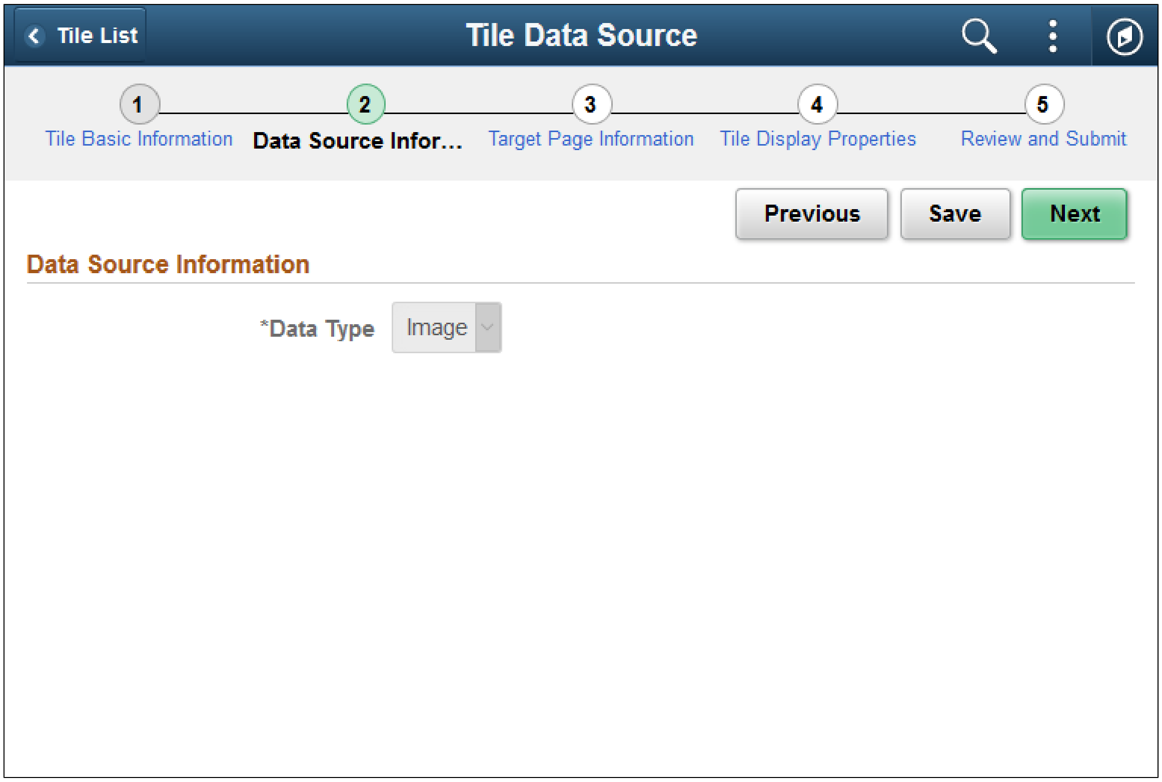
Step 2: Data Source Information - Image

Use the Data Source Information page to select Image as the data type.

Navigation:

1. On the Tile Basic Information page, click the Next button.
2. Select the *Image* data type.

This example illustrates the fields and controls on the Data Source Information page for the Image data type. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Image	Use the image data type to create an image-only tile for an existing content reference.

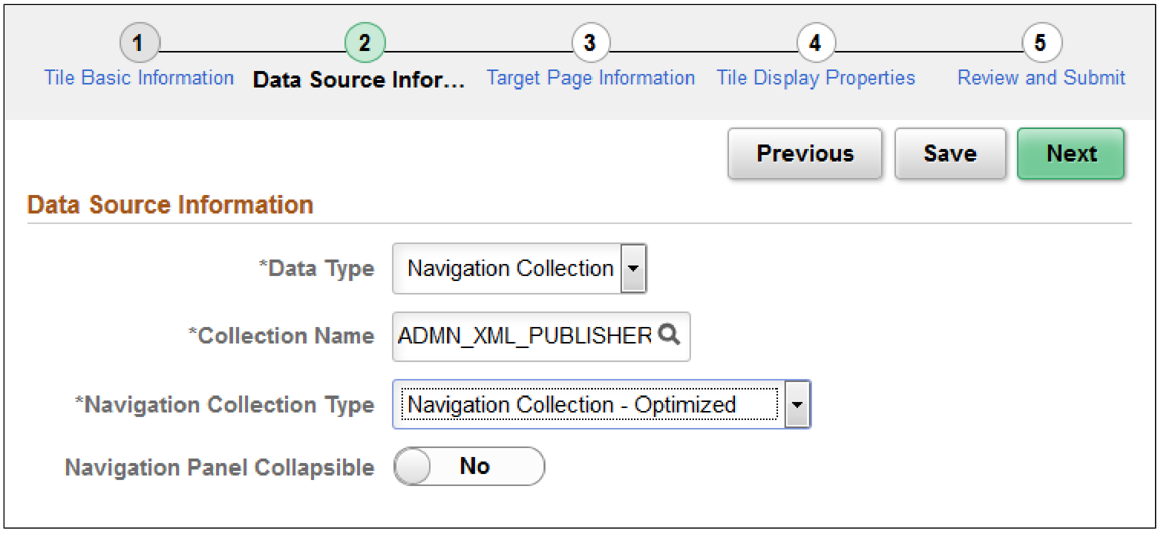
Step 2: Data Source Information - Navigation Collection

Use the Data Source Information page for the Navigation Collection data type to specify which navigation collection data source is to be displayed as a “fluid navigation collection.”

Navigation:

1. On the Tile Basic Information page, click the Next button.
2. Select the *Navigation Collection* data type.

This example illustrates the fields and controls on the Data Source Information page for the Navigation Collection data type. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Collection Name	Select the navigation collection.
Navigation Collection Type	Select the manner in which the navigation collection content will be displayed when a user clicks on the tile: • *Navigation Bar* – Displays the navigation collection links directly in the NavBar. • *Navigation Collection - Non-Optimized* – Displays the navigation collection in a *non-optimized* non-guided non-sequential activity guide. • *Navigation Collection - Optimized* – Displays the navigation collection in an *optimized* non-guided non-sequential activity guide.
Navigation Panel Collapsible	Select *Yes* to present the left panel as collapsible for all users. Otherwise, the left panel only becomes collapsible when the browser width falls below a threshold. **Note:** This field is displayed only when *Navigation Collection Optimized* is selected.

Fluid navigation collections support additional URL parameters that cannot be set in Tile Wizard—specifically, `&JumpPage`, `&NewWindow`, and `&AgInHistory`. See Manually Creating Tile Definitions to Launch Fluid Navigation Collections for information on how to modify the content reference definition to include additional URL parameters.

Related Links

Fluid Navigation Collections

Creating and Maintaining Navigation Collections

Understanding Activity Guide Templates and Template Types

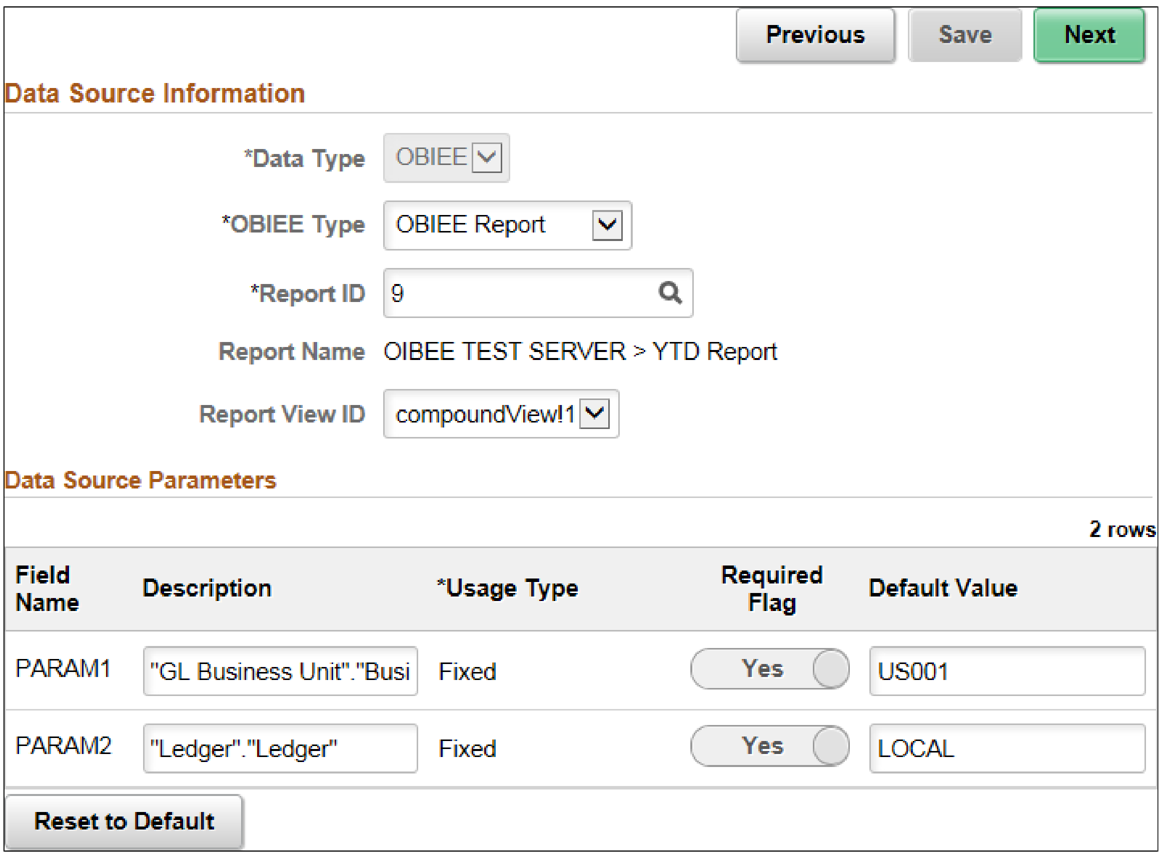
Step 2: Data Source Information - OBIEE

Use the Data Source Information page for the OBIEE data type to specify which OBIEE report or OBIEE dashboard data source.

Navigation:

1. On the Tile Basic Information page, click the Next button.
2. Select the *OBIEE* data type.

This example illustrates the fields and controls on the Data Source Information page for the OBIEE data type. You can find definitions for the fields and controls later on this page.



Data Source Information

Field or Control	*Description*
OBIEE Type	Select either: • *OBIEE Report* • *OBIEE Dashboard*

Field or Control	***Description***
Report ID	Select the ID of the report or dashboard that you want to be the data source. Only reports or dashboards that have been loaded into the PeopleSoft system appear in this list.
Report Name	This field displays the OBIEE server name and the report name. This field is display-only.
Report View ID	For OBIEE reports only, select the view or report type as created in the OBIEE application.

Data Source Parameters

Tile Wizard analyzes the selected OBIEE data source and generates the list of parameters shown in this step. You cannot add to the list of parameters.

Field or Control	***Description***
Description	Displays a description of the parameter.
Usage Type	All data source parameters are *Fixed*, meaning the data source parameter cannot be modified by users of the tile.
Required	Select to require a value for the parameter.
Default Value	Enter the parameter value that the system uses by default.
Reset to Default	Click to set all values on this page to the default values specified by the data source API.

Related Links

Loading OBIEE Report Definitions

Step 2: Data Source Information - PSQuery/SQL

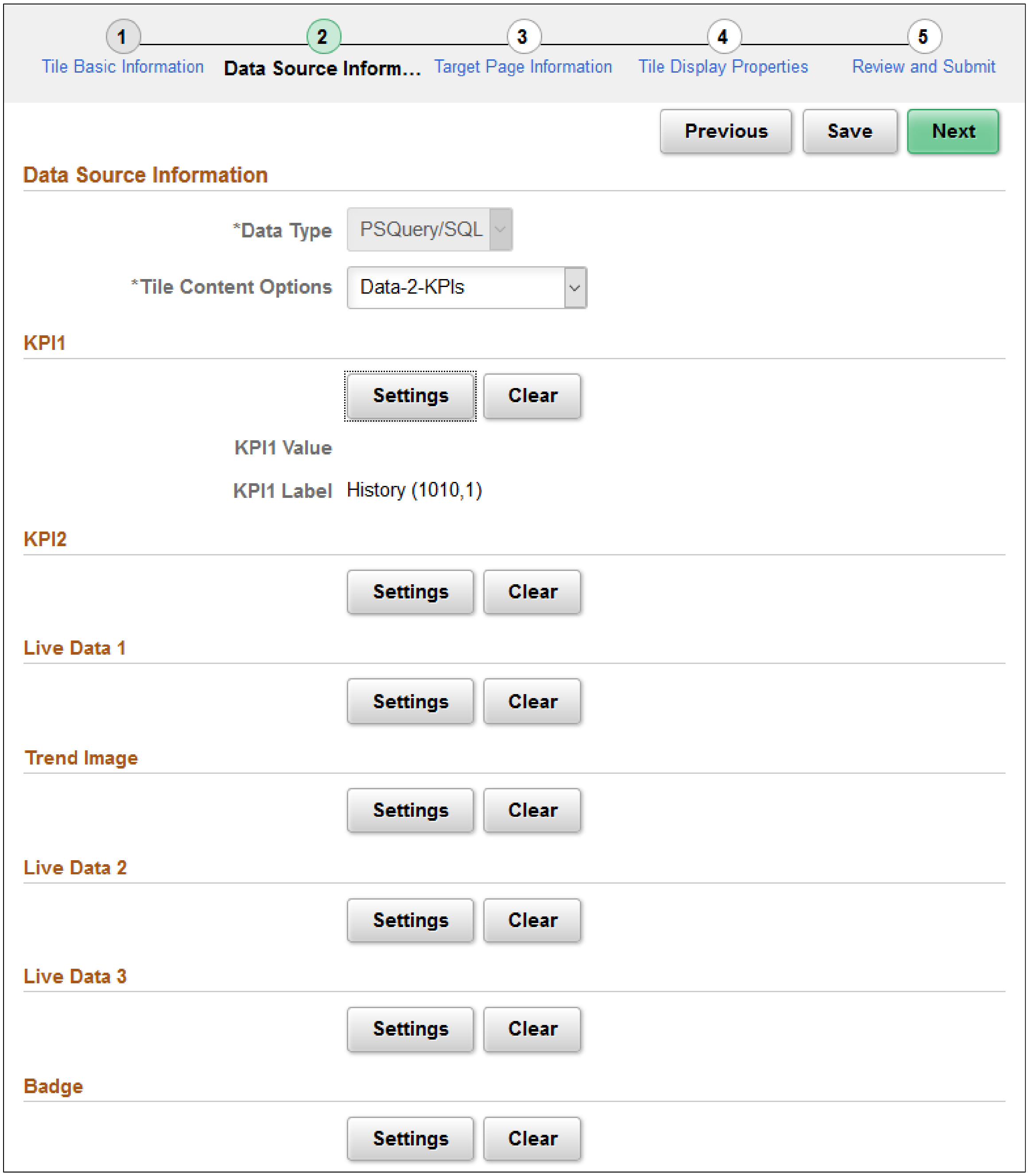
Use the Data Source Information page for the PSQuery/SQL data type to specify which PeopleSoft queries, stored SQL statements, or a combination to specify the data displayed on the tile.

Navigation:

1. On the Tile Basic Information page, click the Next button.

2. Select the *PSQuery/SQL* data type.

This example illustrates the fields and controls on the Data Source Information page for the PSQuery/SQL data type. In this example, settings for KP1 have already been selected. You can find definitions for the fields and controls later on this page.



The PSQuery/SQL data type provides an interface for using a PeopleTools-delivered implementation of the PTGP_APPCLASS_TILE:Tiles:Tile base class that allows you to select one or more PeopleSoft queries, SQL definitions, and Message Catalog entries to provide the data and text to display on the face of a tile. Using this data type does not require you to write or understand PeopleCode. However, you do need to understand the elements that can be displayed on a tile (for example, key performance indicators (KPIs), live data, badge data, and so on), the data types for these elements, and where these elements are displayed. See “Understanding Application Class Tile Content” (PeopleCode API Reference) for more information.

When selecting PeopleSoft queries, SQL definitions, and Message Catalog entries to configure a tile, note the following limitations:

- Regardless of the number rows and columns of data returned by a query or SQL, only the first row is used.
- Only Message Catalog entries that do not use bind variables can be used.

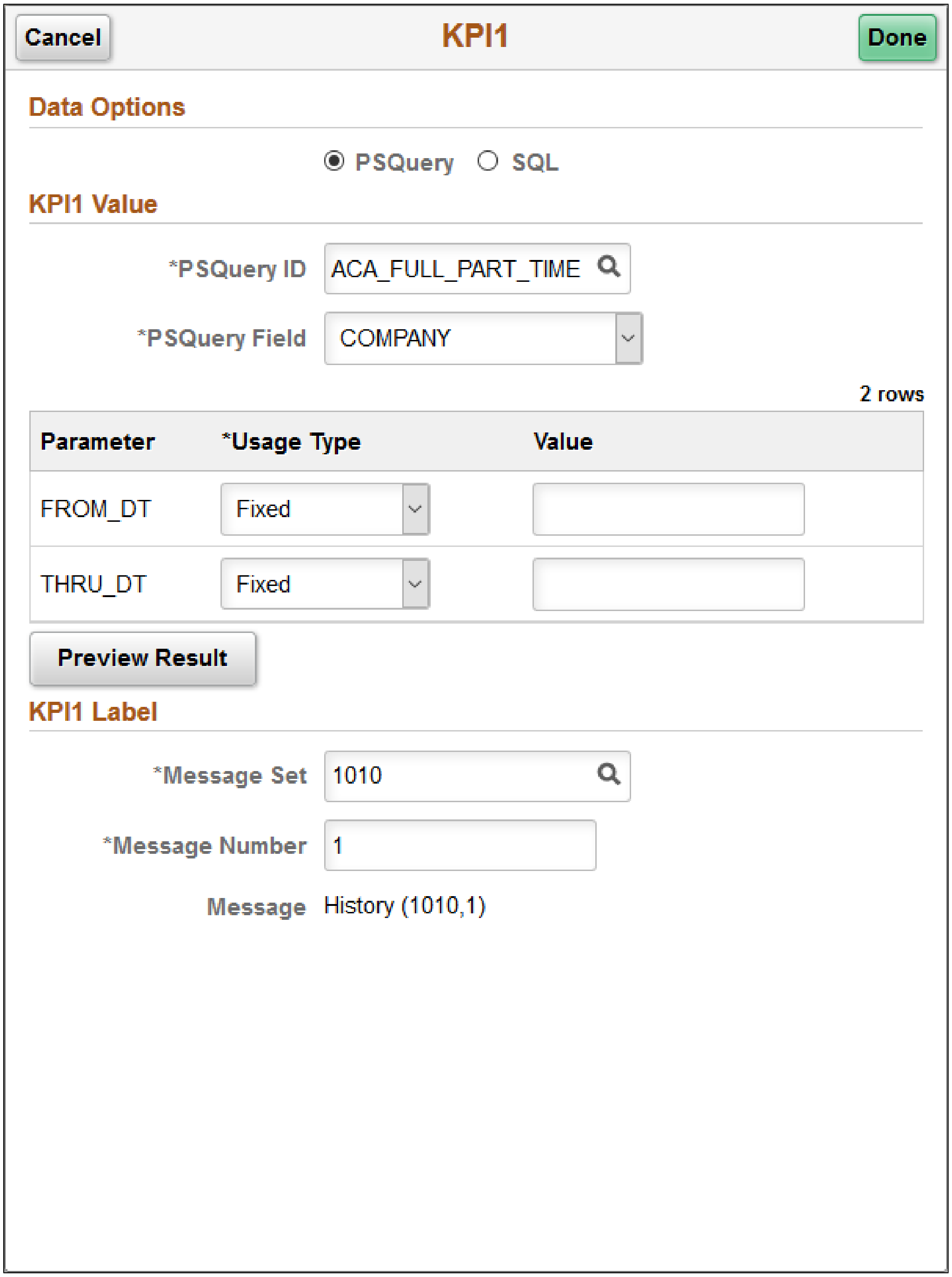
Note: If data masking for personally identifiable information or other sensitive data has been enabled on a record field that is part of the criteria for the PeopleSoft query, then these data values are masked when displayed via Tile Wizard tiles.

Field or Control	***Description***
Tile Content Options	Select one of the following: • *Data-1–KPI* to display one KPI value on the tile. Defining the settings for KPI1 is required. The other values are optional. • *Data-2–KPIs* to display two KPI values side-by-side on the tile. Defining the settings for KPI1 and KPI2 is required. The other values are optional. • *Data-2–KPIs-TopBottom* to display two KPI values stacked on the tile. Defining the settings for KPI1 and KPI2 is required. The other values are optional.
Settings	Click the Settings button to configure the values for an element.
Clear	Click the Clear button to clear the values for an element.

Configuring Settings

Click the Settings button to configure the values for an element.

The following example illustrates the settings page for the KPI1 element using a PeopleSoft query as the data source.



Field or Control	***Description***
Data Options	Select either *PSQuery* or *SQL*.
PSQuery ID	When *PSQuery* has been selected as the data option, select the PeopleSoft query.

Field or Control	***Description***
PSQuery Field	When *PSQuery* has been selected as the data option, select the column from the PeopleSoft query to return the data value.
SQL ID	When *SQL* has been selected as the data option, select the SQL definition.
SQL Field	When *SQL* has been selected as the data option, select the column from the SQL to return the data value.
Parameter	Displays input parameters as is appropriate for the selected query.
Message SetMessage Number	Select a message set and number to return a label for the data value.

Step 2: Data Source Information - Pivot Grid

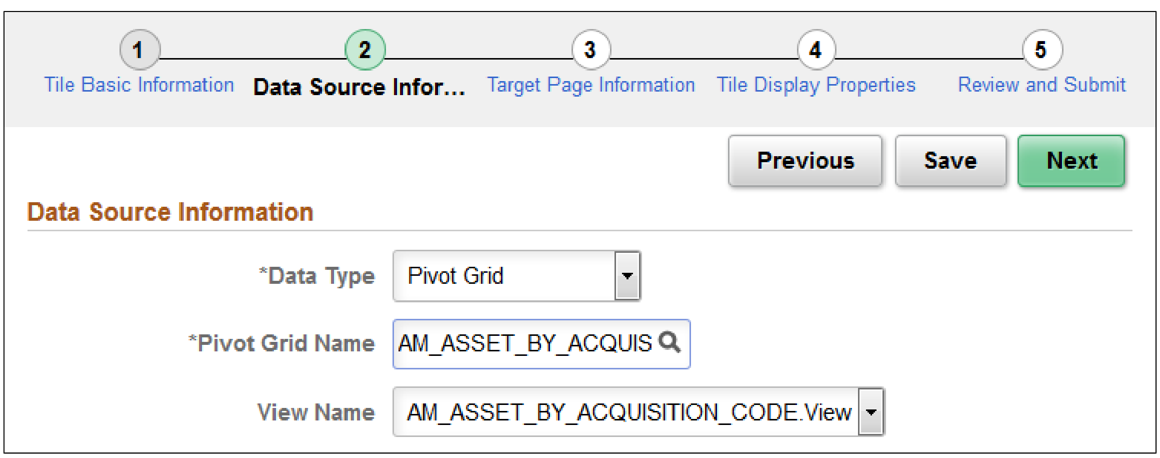
Use the Data Source Information page for the Pivot Grid data type to specify which pivot grid data source.

Navigation:

1. On the Tile Basic Information page, click the Next button.

2. Select the *Pivot Grid* data type.

This example illustrates the fields and controls on the Data Source Information page for the Pivot Grid data type. You can find definitions for the fields and controls later on this page.



Data Source Information

Field or Control	***Description***
Pivot Grid Name	Select the pivot grid that you want the tile to display.
View Name	The view name is generated and selected automatically by the system.

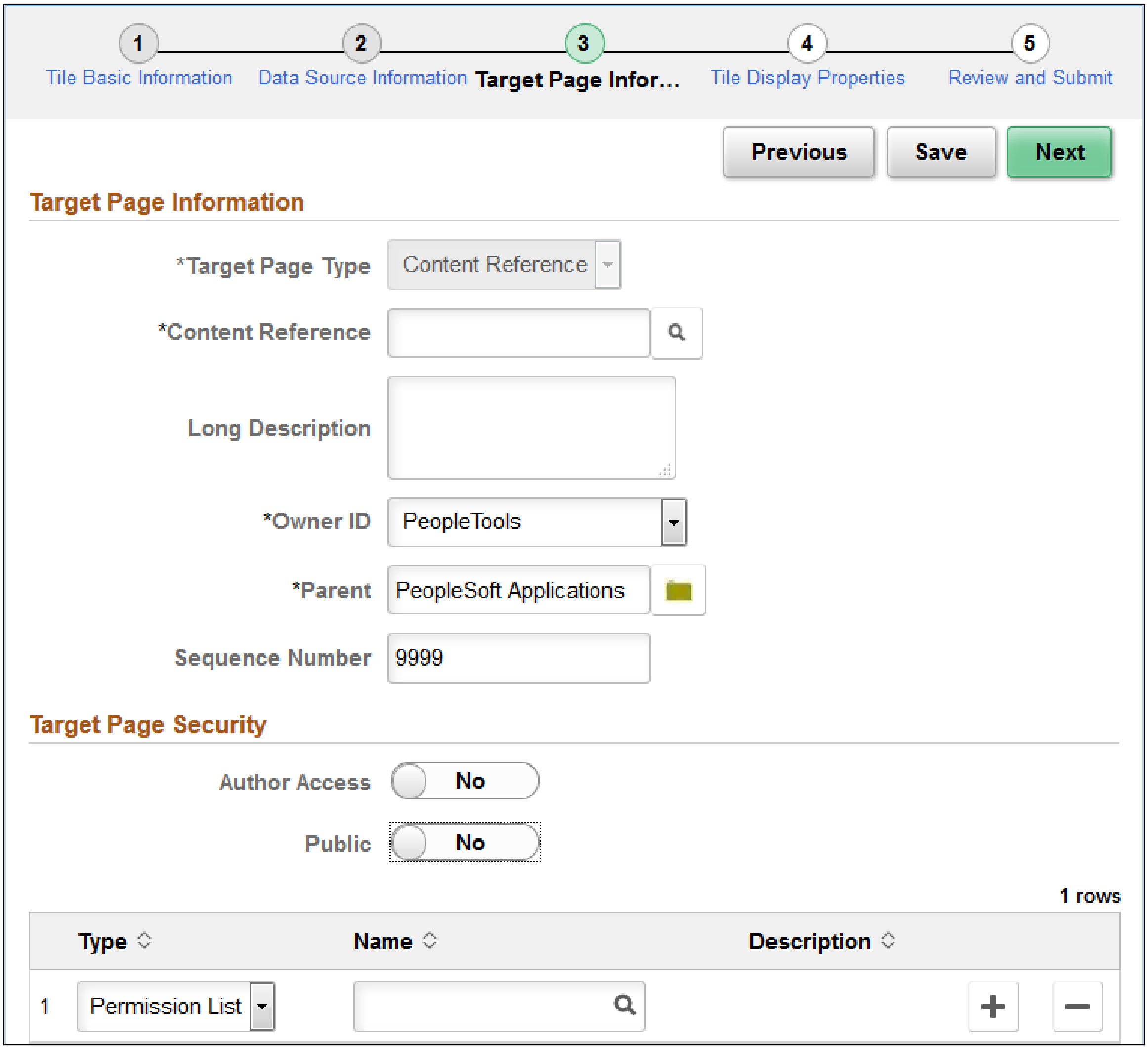
Step 3: Target Page Information

Use the Target Page Information page to select the target page and specify certain attributes of the tile's content reference definition.

Navigation:

On the Data Source Information page, click the Next button.

This example illustrates the fields and controls on the Target Page Information page. You can find definitions for the fields and controls later on this page.



Target Page Information

Field or Control	Description

Target Page Type	Select from the following options:
	• *Content Reference* – For all data sources, select this option to select an existing content reference as the target.
	  • If the selected content reference is an existing tile definition, the other fields on this page are read-only.
	  • Otherwise, when the tile definition is published, a content reference link will be created in the tile repository for this tile.
	• *OBIEE Report* – For OBIEE data sources only, select this option to use the PeopleTools OBIEE report viewer.
	When the tile definition is published, a new content reference definition will be created in the tile repository for this tile.
	Note: When this tile definition is published, the target page type is automatically updated to *Content Reference*.
	• *Pivot Grid Viewer* – For pivot grid data sources only, select this option to use the PeopleTools pivot grid viewer.
	When the tile definition is published, a new content reference definition will be created in the tile repository for this tile.
	Note: When this tile definition is published, the target page type is automatically updated to *Content Reference*.

Content Reference	When *Content Reference* is selected as the target page type, select an existing content reference from the portal registry on the Select CRef page.
Target Page Name	When either *OBIEE Report* or *Pivot Grid Viewer* is selected as the target page type, the value in this field is generated automatically. **Note:** When the field is editable, you can modify this to any unique name.
Long Description	Enter a description for the content reference (or content reference link) definition to be created.

Field or Control	*Description*
Owner ID	Select an owner ID for the content reference (or content reference link) definition to be created.
Parent	Select a folder in the tile repository for the content reference (or content reference link) definition to be created.
Sequence Number	Enter a sequence number for the content reference (or content reference link) definition to be created.

Target Page Security

Note: For a content reference link, security is defined on the target content reference. The fields displayed are read-only.

Field or Control	*Description*
Public	Select whether to allow public or selected security access: • *Yes:* Select to allow public access to the target page; all other security options are hidden. • *No:* Select to assign role-based or permission list-based security to the target page and whether to provide author access override to this selected security.
Author Access	Select *Yes* to provide the tile author with access to the target page regardless of any security restrictions assigned to the tile. This access is granted based on the author's user ID.
Type	Select the type of security to assign to the target page definition. Options are: • *Permission List:* Select to assign permission list-based security to the tile. Select a permission list in the **Name** field. • *Role:* Select to assign role-based security to the tile. Select a role in the **Name** field.
Name	Select the name of the permission list or role to which to authorize access to the tile.

Related Links

Understanding Tile Definitions and the Tile Repository

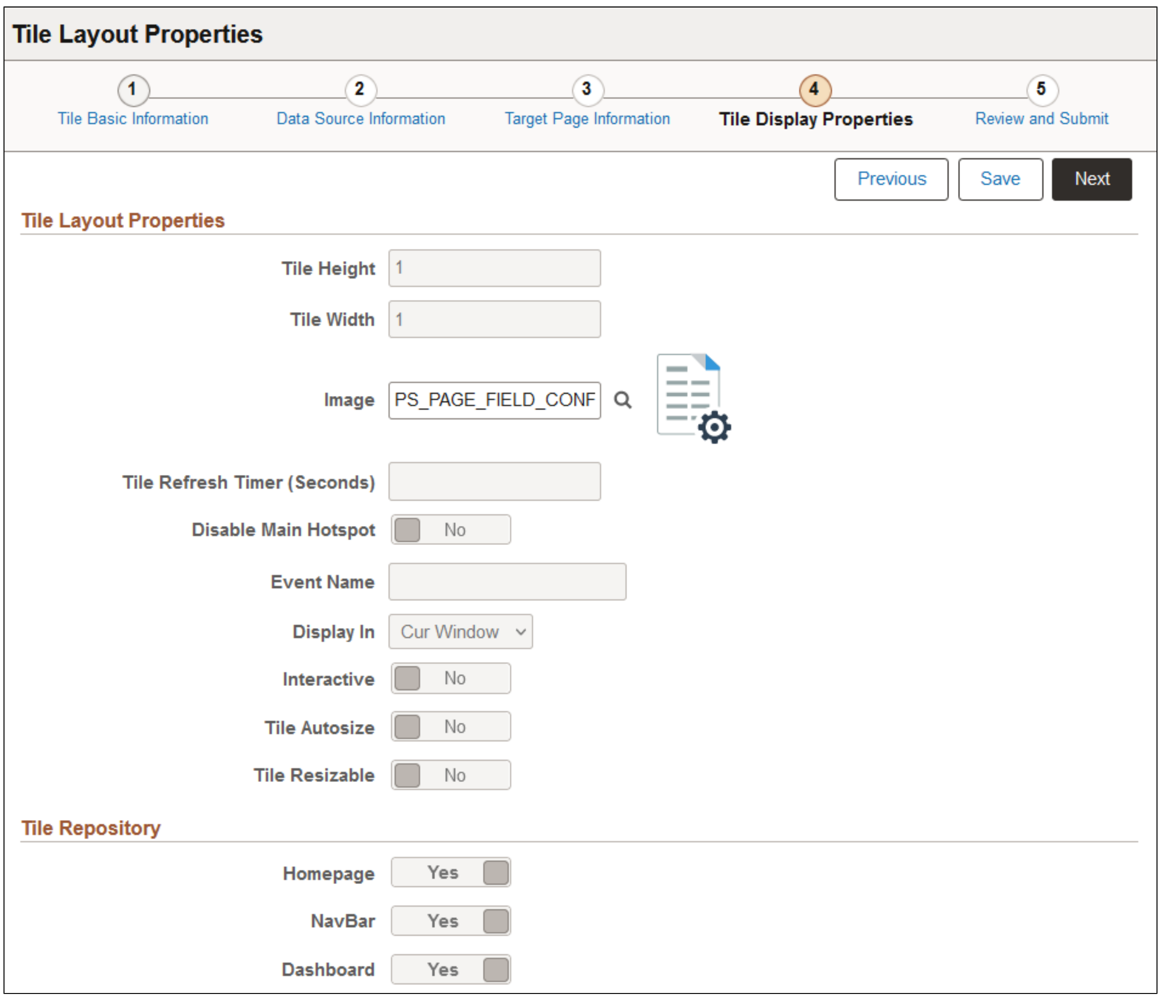
Step 4: Tile Layout Properties

Use the Tile Layout Properties page to specify the fluid attributes of the tile definition.

Navigation:

On the Target Page Information page, click the Next button.

This example illustrates the fields and controls on the Tile Layout Properties page. You can find definitions for the fields and controls later on this page.



Tile Layout Properties

<table>
<tr><th>Field or Control</th><th>Description</th></tr>
<tr><td>Tile Height</td><td>Set the height of the tile.

The default dimensions of a tile are 1 tile unit by 1 tile unit. Tiles up to 8x8 are supported. Note the following conditions and limitations with respect to the height and width of a tile:

• A tile can never be larger than the width of the device. For example, if the width of a device is 2 in portrait mode, and 3 in landscape; the tile can only be maximum of width 2 on this device.

• The valid size is checked only when the tile is initially loaded. If the device is rotated, or if the browser is resized, the tile is not resized.

• Any tile sized larger than 2x2 is not guaranteed to display at the specified size depending on the device used to access the PeopleSoft system. For example, a 4x4 tile on a small form factor device in portrait mode would only show as a 2x3 tile.

• Therefore, tiles larger than 2x2 need to implement variable sizing to accommodate different device sizes. Use large size tiles sparingly.

See the information on the Fluid Attribute page tab for more information including for a table of tile widths and heights from 1 to 8.</td></tr>
<tr><td>Tile Width</td><td>Set the width of the tile.</td></tr>
<tr><td>Image</td><td>Click the Image lookup button to display the Browse Image Catalog page, which allows you to select an image (in SVG format only) from the database to display a custom static image as the fluid content. Otherwise the default image is displayed.

See the following section, "Using the Browse Image Catalog Page" for more information.

Note: When dynamic content is also configured for a tile, this static image is displayed only until the dynamic content is loaded.</td></tr>
</table>

Field or Control	*Description*
Tile Refresh Timer (Seconds)	Enter the time in seconds to set an automatic refresh period for dynamic content on a tile. When the timer limit has been reached, the system re-draws the tile so that it displays the current data, such as in the case with chart. The default value of 0 disables any automatic refresh. **Note:** The system enforces a 10 second minimum limit. Any value entered less than 10 seconds is ignored and is treated as 10 seconds. When setting this value, be sure to monitor performance of the page refreshes and adjust accordingly. **Note:** When an event name is selected, this field is disabled.
Disable Main Hotspot	Select this option to allow links that are displayed dynamically within the tile to be available and usable. Selecting this option disables clicking the tile to display the target content defined for the tile.
Event Name	Select the event name to subscribe to a push notification event defined on the server. Subscribing to an event allows the tile content to be updated by the event. See “Understanding the Push Notification Framework” (Fluid User Interface Developer’s Guide) and “Defining Events” (Fluid User Interface Developer’s Guide)for more information on defining push notification events.
Display In	Control how the target transaction or content appears once a user taps the tile. Options are: • *Cur Window*: Target content appears in the current browser window. • *Modal*: Target content appears in a modal window. • *NavBar*: Target content appears in the NavBar. **Note:** This behavior applies only when the tile is added to the NavBar. If this is selected and the tile is added to a homepage or dashboard, the target content displays in the current browser window when tapped. • *New Window*: Target content appears in a new tab in the browser.

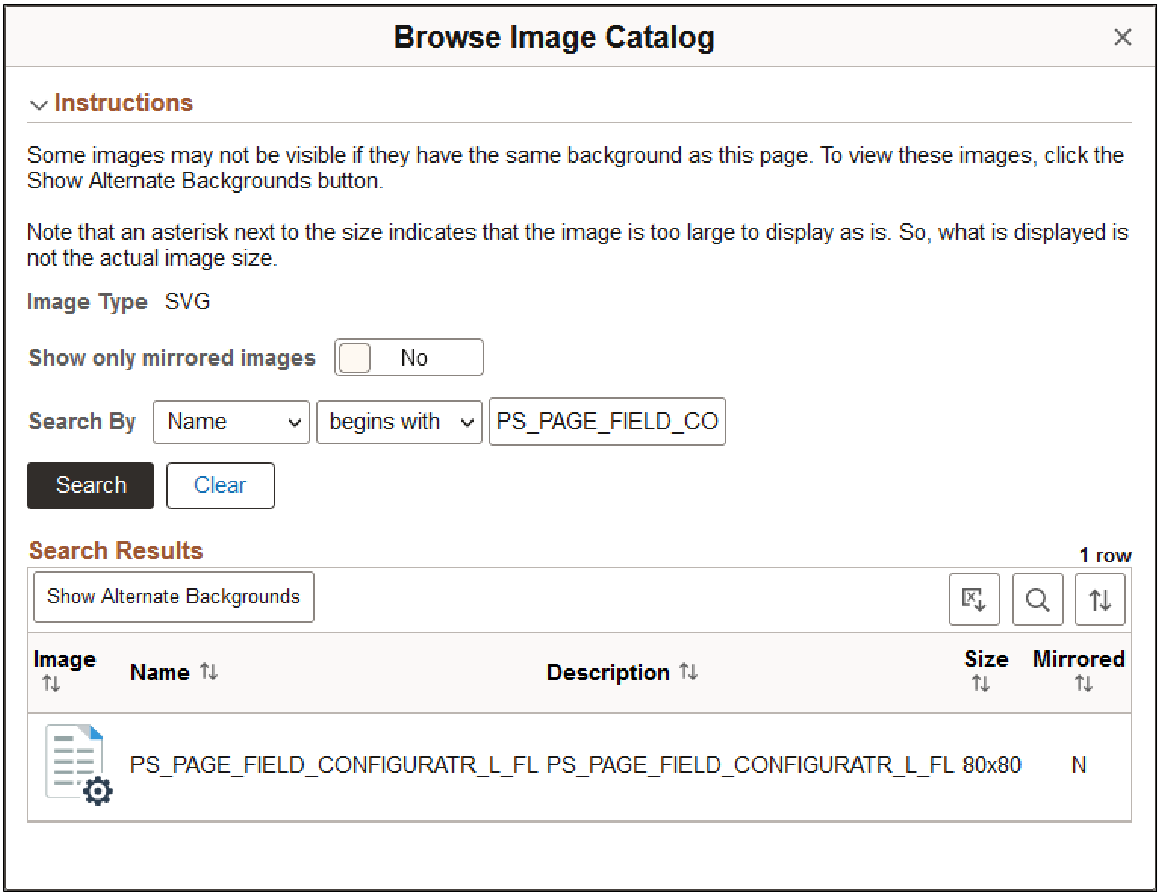
Field or Control	***Description***
Modal Parameters	Enter custom modal options. See "Modal Options" (Fluid User Interface Developer's Guide) for more information.
Interactive	Select this option to display the fluid content as interactive, which allows users to enter data or click buttons within the fluid content. Since interactive fluid content require more resources than non-interactive ones, select this option only as needed. **Note:** The Interactive option is disabled for pivot grid tiles and OBIEE tiles.
Tile Autosize	Select this option to allow the fluid content to dynamically resize itself based on the size of the content. Note the following conditions and limitations with respect to the autosize option: • The autosize option pertains to tiles and related information, but not to embedded related content. • The autosize option is best used on dashboards with few tiles. Be aware that layout changes can occur from the autosized content. • Autosize automatically accounts for resizing of the browser and rotation of the device. • The autosize and resizable options are mutually exclusive; both cannot be selected simultaneously. • The tile will not resize itself to larger than the dimensions of the device. • If the content is larger than the size of the tile, the tile will display scroll bars. • Due to the same-origin policy implemented by web browsers, for the autosize option to work with external content, the external content and the PeopleSoft application must either reside within the same domain or they must both set the document.domain property to the same value. Otherwise, the autosize option is ignored. For more information on how the domain and same-origin is determined, see Same-origin policy.

Field or Control	*Description*
Tile Resizable	Select this option to allow the tile to be redisplayed at the specified alternate dimensions. Note the following conditions and limitations with respect to the resizable option: • The resizable option is best used for tiles that can provide both a summary view and detailed view. When a tile is resized by a user, the content is refreshed to the show content that is optimized for the alternate size (and vice versa when the size is toggled back to the original size). • The autosize and resizable options are mutually exclusive; both cannot be selected simultaneously. • The state of the resize is not saved; therefore, whenever the user returns to the dashboard, the tile is displayed at its original size. • If the alternate size of the tile is the same as the original size (either by definition or by computation of the maximum size allowed by the device), no resize icon is displayed.
Alternate Height	Set the alternate height for a resizable tile.
Alternate Width	Set the alternate width for a resizable tile.

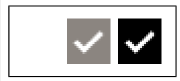
Using the Browse Image Catalog Page

Use the Browse Image Catalog page to select an SVG from the database to display a custom static image as the fluid content.

This example illustrates the fields and controls on the Browse Image Catalog page.



Field or Control	***Description***
Image Type	*SVG* is selected by default and cannot be changed.
Show only mirrored images	Select *Yes* to only retrieve images that are mirrored. Otherwise, both mirrored and unmirrored images are retrieved.
Search By	Select *Name* or *Description* and additional search criteria to limit the search results.
Search	Click the Search button to retrieve the images meeting the search criteria from the database.
Clear	Click the Clear button to clear both the search criteria and the grid.

Field or Control	***Description***
Show Alternate Backgrounds	Click the Show Alternate Backgrounds button to redisplay each image in the current grid on three background colors: white, gray, and black. This option allows you to view white images, which are not visible on the default white background. In this example, the white check mark is visible on the gray and black backgrounds: 

Select an image in the grid to use it as the custom static image.

Tile Repository

Field or Control	***Description***
Homepage	Select this option to allow the definition to be listed when the Add Tile dialog box is accessed from fluid homepages.
Dashboard	Select this option to allow the definition to be listed when the Add Tile dialog box is accessed from fluid dashboards.
NavBar	Select this option to allow the definition to be listed when the Add Tile dialog box is accessed from the NavBar.

Related Links

Setting Fluid Attributes for Content References

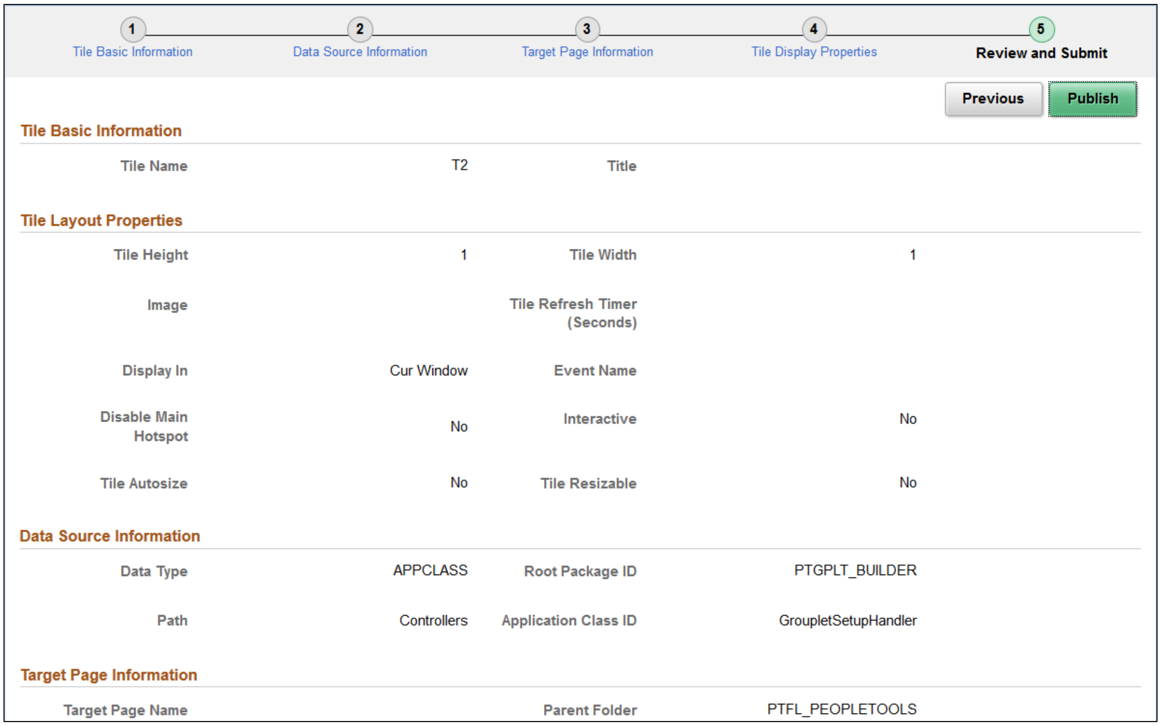
Step 5: Review and Submit

Use the Review and Submit page to review and publish the tile definition.

Navigation:

On the Tile Display Properties page, click the Next button.

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



Field or Control	*Description*
Publish	Click the Publish button to initially publish the tile definition. **Note:** After the tile definition has been published, subsequent modifications can be saved by clicking the Save button on any Tile Wizard page when the Save button is displayed and is active. However, you must click the Publish button and re-publish the changes to update the behavior of the tile.

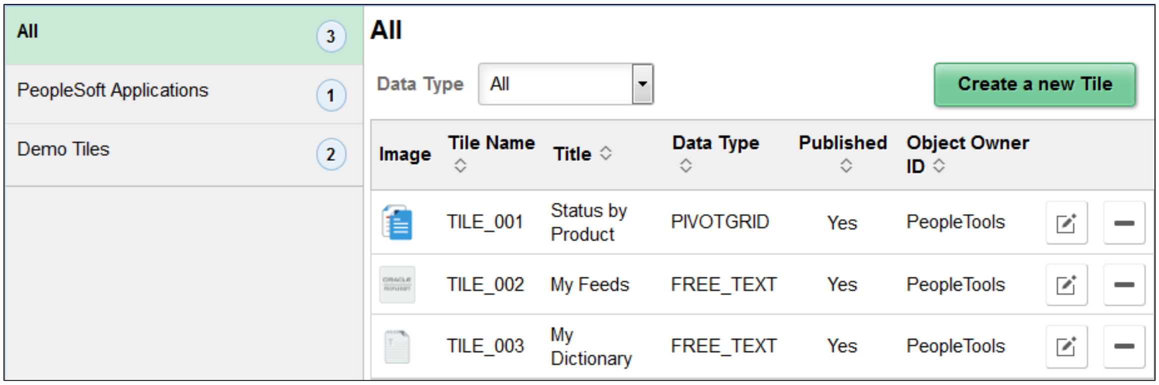
Tile List Page

Use the Tile List page to review, edit, or delete existing tile definitions that have been created with Tile Wizard.

Note: While the tile repository may contain additional tile definitions, only those created through Tile Wizard are listed here.

Navigation:

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



Use the categories in the left panel to filter the tiles displayed in the tile list.

Note: The left panel is collapsible depending on the width of the device used to access the page.

Field or Control	***Description***
Data Type	Select a data type to filter the tiles displayed in the tile list.
Create a new Tile	Click this button to create access Tile Wizard to create a new tile definition.
Edit	Click the Edit button to edit an existing tile definition. **Note:** Alternatively, click anywhere in the row (except on the Delete button) to edit an existing definition.
Delete	Click the Delete button to delete an existing tile definition.
Add to Project	Click the Add to Project button to access the Add To Project page to insert one or more tile definitions into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) for more information.

Creating and Maintaining Tile Definitions Manually

In certain circumstances, you must create or maintain tile definitions manually—for example, to create a tile for a fluid activity guide, to create a tile to access a navigation collection using a PeopleTools-delivered activity guide template, or to update an existing tile definition.

Related Links

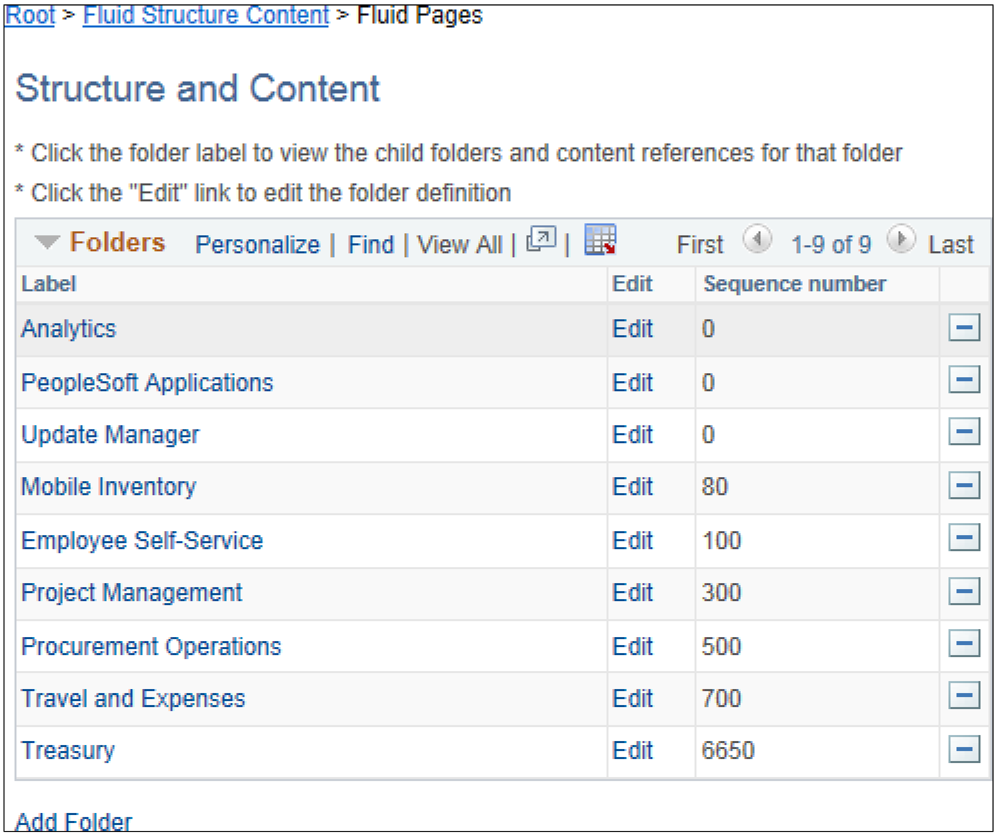
PeopleSoft Fluid UX Standards: Tiles

Understanding Tile Definitions and the Tile Repository

All tile definitions are stored in folders within the portal registry referred to as the *tile repository.* Specifically, the subfolders under Fluid Structure Content > Fluid Pages comprise the tile repository.

Important! Only one level of subfolders under the Fluid Pages folder is supported in the tile repository.

The following example displays subfolders in the tile repository:



Tile definitions are created in the tile repository either manually or by using Tile Wizard. In either case, the tile definition is either a content reference or a content reference link:

- When the target page is an existing content reference, modify the Fluid Attributes page tab for the existing content reference definition to create a content reference link in the tile repository.

  Note: When you use Tile Wizard to create the tile definition, the content reference link is created automatically when you select *Content Reference* as the target content type.

- When the target page is a delivered viewer component or other specialized component (for example, the pivot grid viewer, the activity guide start page, or others), create a new content reference definition in the tile repository with the appropriate URL definition.

Related Links

Setting Fluid Attributes for Content References
Using Tile Wizard

Manually Creating Tile Definitions to Access Fluid Activity Guides

To deploy optimized and non-optimized fluid activity guides, you must manually create an access mechanism that invokes the fluid activity guide start page component (PT_AGSTARTPAGE_NUI). Typically, a tile definition will provide this access mechanism. See Deploying Optimized and Non-Optimized Fluid Activity Guides for more information.

Manually Creating Tile Definitions to Launch Fluid Navigation Collections

You can create tile definitions to launch what are referred to as “fluid navigation collections.” See Fluid Navigation Collections for a definition of “fluid navigation collections” and guidelines on how to define these navigation collections.

You can create these tile definitions one of two ways:

- Using Tile Wizard, which provides a simple step-by-step guide. See Step 2: Data Source Information - Navigation Collection for more information.
- Manually by following the procedures in this section.

To create a tile definition to launch a fluid navigation collection:

1. Obtain the ID for the navigation collection:
   a. Go to the portal registry (**PeopleTools** > **Portal** > **Structure and Content**).
   b. Go to **Portal Objects** > **Navigation Collections**.
   c. Click the Edit link for a navigation collection.
   d. Copy or make note of the Name field.
2. Navigate to the tile repository in the portal registry (Fluid Structure Content > Fluid Pages).
3. Click the link for a subfolder.
4. Create a new content reference definition and follow the specific instructions depending on how you want to display the navigation collection.

In an Optimized Activity Guide

Note: Only links under the root folder or the first level of subfolders are displayed in an activity guide. If the navigation collection includes additional levels of subfolders and links, they are not displayed.

To display the navigation collection and links as an optimized non-guided non-sequential activity guide:

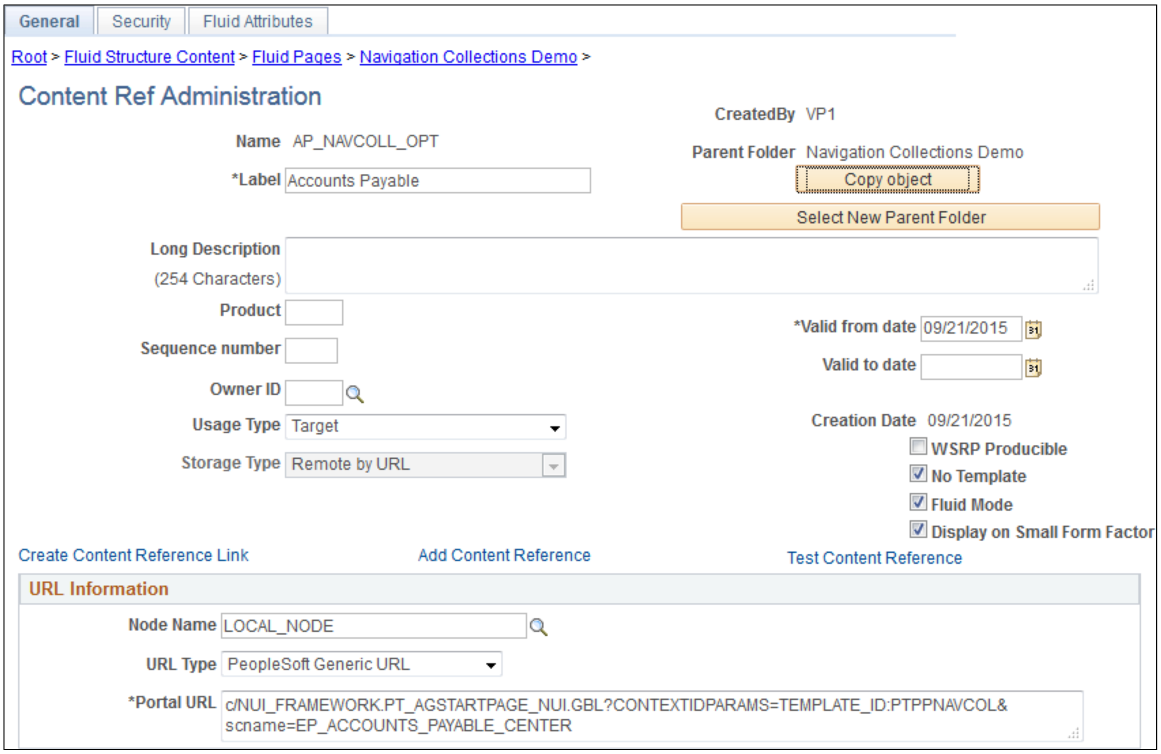
1. Set the following on the General page:
   - **No Template**: *selected*
   - **Fluid Mode**: *selected*
   - **Display on Small Form Factor** : *selected*

- **Node Name**: *LOCAL_NODE*
- **URL Type**: *PeopleSoft Generic URL*
- **Portal URL**:

  `c/NUI_FRAMEWORK.PT_AGSTARTPAGE_NUI.GBL?CONTEXTIDPARAMS=TEMPLATE_ID:PTPPNA⇒`

  `VCOL&scname=`*`NAVCOLL_ID`*

The following example displays a tile definition to display a navigation collection and links as an optimized non-guided non-sequential activity guide.



2. In the Portal URL field, add one or more optional query string parameters to override the default behavior of fluid navigation collections:

   - `&PanelCollapsible=Y`: Displays the left panel as collapsible rather than as fixed.
   - `&JumpPage=N`: Do not display a “jump page” on smartphones.

     On smartphones, a fluid navigation collection is displayed as the links only (that is, what is displayed in the left panel on other devices) by default. Also by default, clicking a link transfers you to the target component and exits the fluid navigation collection. You can return to the navigation collection using the Back button.
   - `&NewWindow=N`: Do not display a New Window menu item in the Actions menu.
   - `&AgInHistory=N`: Do not add the fluid navigation collection to the Back button history stack.

3. On the Security page, set the security to be the same as the security for the navigation collection.

4. On the Fluid Attributes page accept the defaults and set any other desired fluid attributes.

In a Non-Optimized Activity Guide

Note: Only links under the root folder or the first level of subfolders are displayed in an activity guide. If the navigation collection includes additional levels of subfolders and links, they are not displayed.

To display the navigation collection and links as an non-optimized non-guided non-sequential activity guide:

1. Set the following on the General page:
   - **No Template**: *selected*
   - **Fluid Mode**: *selected*
   - **Display on Small Form Factor** : *selected*
   - **Node Name**: *LOCAL_NODE*
   - **URL Type**: *PeopleSoft Generic URL*
   - **Portal URL**:

     `c/NUI_FRAMEWORK.PT_AGSTARTPAGE_NUI.GBL?CONTEXTIDPARAMS=TEMPLATE_ID:PTPPNO⇒`

     `NOPT&scname=`*`NAVCOLL_ID`*

2. In the Portal URL field, add one or more optional query string parameters to override the default behavior of fluid navigation collections:
   - `&PanelCollapsible=Y`: Displays the left panel as collapsible rather than as fixed.
   - `&JumpPage=N`: Do not display a “jump page” on smartphones.

     On smartphones, a fluid navigation collection is displayed as the links only (that is, what is displayed in the left panel on other devices) by default. Also by default, clicking a link transfers you to the target component and exits the fluid navigation collection. You can return to the navigation collection using the Back button.
   - `&NewWindow=N`: Do not display a New Window link the Actions menu.
   - `&AgInHistory=N`: Do not add the fluid navigation collection to the Back button history stack.

3. On the Security page, set the security to be the same as the security for the navigation collection.
4. On the Fluid Attributes page accept the defaults and set any other desired fluid attributes.

As Folders and Links in the NavBar

Note: When a navigation collection is displayed in the NavBar, the contents of one folder are displayed at a time. However, there is no restriction on subfolders beneath the first level.

To display the navigation collection and links directly in the NavBar:

1. Set the following on the General page:
   - **No Template**: *selected*
   - **Fluid Mode**: *selected*
   - **Display on Small Form Factor** : *selected*
   - **Node Name**: *LOCAL_NODE*
   - **URL Type**: *PeopleSoft Generic URL*
   - **Portal URL**:

     `c/NUI_FRAMEWORK.PTNUI_NAVCOLL_COMP.GBL?sa=n&scname=NAVCOLL_ID`
2. On the Security page, set the security to be the same as the security for the navigation collection.
3. On the Fluid Attributes page set the following:
   - **Display In**: *NavBar*
   - **Include In Repository: Homepage**: *deselected*
   - **Include In Repository: Dashboard**: *deselected*
   - **Include In Repository: NavBar**: *selected*
4. Set any other desired fluid attributes.

Related Links

Creating and Maintaining Navigation Collections
Setting Fluid Attributes for Content References
Understanding Activity Guide Templates and Template Types
“Working with Two-Panel Implementations” (Fluid User Interface Developer’s Guide)

Hiding or Displaying Tiles Using an Application Class

You can define a custom application class to hide or display tiles on fluid homepages and dashboards.

Note: Once a tile has been configured with this custom application class, the tile is no longer available on the Add Tile page for users to search for or add the tile. In addition, the tile can no longer be moved or deleted when configured on a homepage in this manner.

To hide or display tiles using a custom application class:

1. Write application class PeopleCode that extends the PTGP_APPCLASS_TILE:API:TileAppFilter base class.
2. Implement the “IsTileVisible” (PeopleCode API Reference) method to contain application-specific logic that hides or displays tiles based on your business requirements. To the extent possible, write

lightweight application logic to avoid issues with loading fluid homepages or dashboards that include multiple tiles that employ a tile filter.

The following example provides a simple implementation of PTGP_APPCLASS_TILE:API:TileAppFilter:

```
import PTGP_APPCLASS_TILE:API:TileAppFilter;

class MyFilter extends PTGP_APPCLASS_TILE:API:TileAppFilter
   method IsTileVisible(&cref As ApiObject) Returns boolean;

end-class;

method IsTileVisible
   /+ &cref as ApiObject +/
   /+ Returns Boolean +/
   /+ Extends/implements PTGP_APPCLASS_TILE:API:TileAppFilter.IsTileVisible +/

   /* Application-specific logic to hide or show a tile. */
   If IsUserInPermissionList("PTPT1000", "PTPT1600") And
         &cref.Name <> "PORTAL_ROOT_OBJECT" Then
      Return True;
   Else
      Return False;
   End-If;
end-method;
```

Update the content reference definition for the tile to include TILEAPPFILTER as a content reference attribute.

Set the attribute value to your custom application class. For example:

This example demonstrates how to set the TILEAPPFILTER content reference attribute.

3. Update the fluid homepage definition in **Fluid Structure Content** > **Fluid Homepages**. On the Tile Content page:

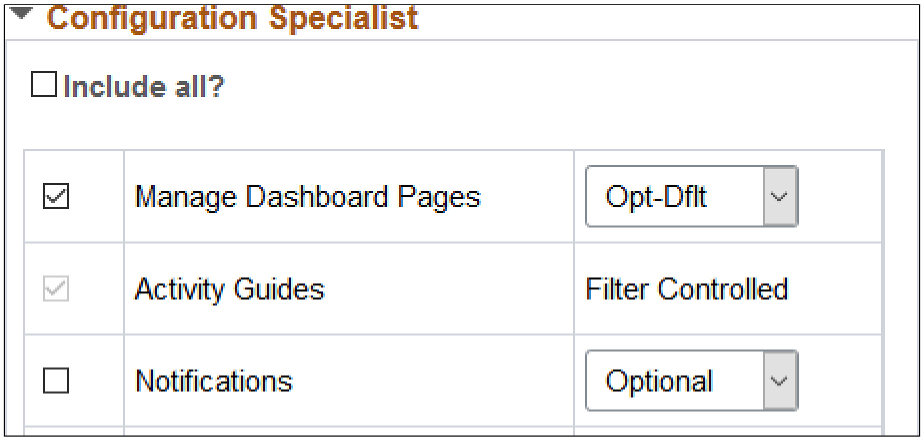
   a. Select the check box for the tile in question.

   b. Select one of these options only: *Opt-Dflt*, *Req-Fix*, or *Required*.

   Important! Do not select *Optional*. The tile will not be displayed automatically on user's homepages and users will not be able to search for and add the tile.

   c. Save the homepage definition.

Once a tile has been configured this way on a homepage, it cannot be reconfigured:



To restore the default behavior of this tile, remove the TILEAPPFILTER content reference attribute on the tile's content reference definition.

The use of the IsTileVisible method has the following limitations:

- In a cluster environment, the TileAppFilter.IsTileVisible logic is only available if you are logged in to the node where the tile is defined directly. If the tile is added as a remote node, it will behave like a conventional tile, and the TileAppFilter.IsTileVisible logic is not executed.
- In fluid homepages, the TileAppFilter.IsTileVisible logic is only executed once during a login session when the tile is initially loaded (typically at login) as the homepages are cached. However, dashboards evaluate the TileAppFilter.IsTileVisible logic every time the dashboard is loaded.

Chapter 18

Managing OBIEE Reports and Dashboards

Understanding OBIEE Integration Within the PeopleSoft Framework

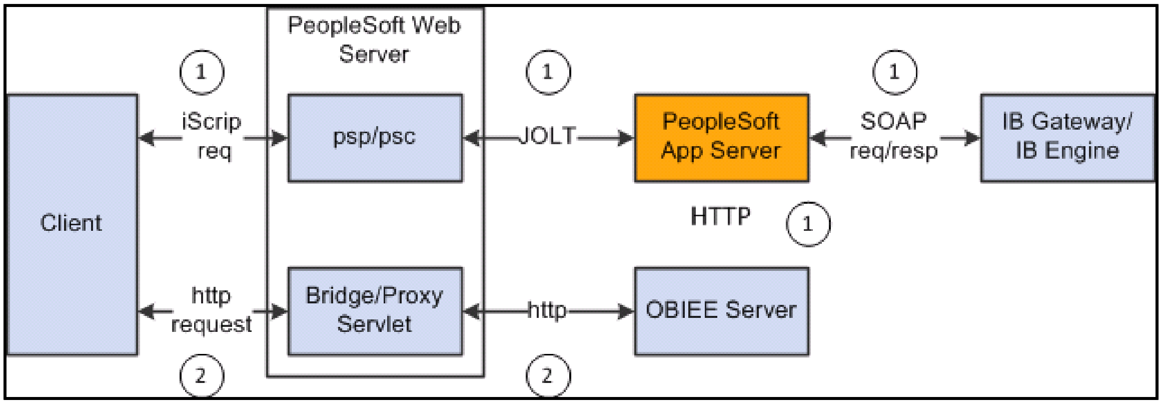
The OBIEE analytics servlet enables application developers to integrate with OBIEE analytics and display OBIEE reports as embedded pagelets on application pages, portal homepages, and dashboards. In addition, OBIEE reports and dashboards can be published as tile definitions for display on fluid homepages and dashboards. This functionality defines the OBIEE reports and dashboards iScript-based content reference definitions in PeopleSoft applications. The iScript essentially acts as a client for the report request and sends Service Oriented Architecture Protocol (SOAP) requests to the OBIEE server.

PeopleSoft applications store the credentials of an OBIEE administrator—a user with adequate privileges—in the PeopleSoft database and use these credentials to sign in to the OBIEE server, impersonate a real OBIEE user, and establish a session. The OBIEE server returns a session ID for this user session, which is sent as a query parameter with each SOAP/HTTP request being made to the OBIEE server. The OBIEE server responds to all valid requests. If the session ID is expired or is invalid, a sign-in screen appears to the user. The OBIEE server provides a session ID and then appends a session ID in each HTTP request that is part of the HTML markup provided by the OBIEE server. If the user ID value in the PeopleSoft application is not found on the OBIEE server, an authorization error message appears to the user in the PeopleSoft application. Similarly, if the user exists in OBIEE but does not have authorization to the requested report, an authorization error message appears to the user.

If authentication passes, then the OBIEE server generates the basic HTML markup for the PeopleSoft application to display. The application uses this basic HTML to send HTTP requests to the OBIEE server from the browser. Before reaching the OBIEE server, the HTTP requests pass through the PeopleSoft bridge servlet. The bridge servlet:

1. Acts as a reverse proxy server to avoid any cross domain issues.
2. Checks the validity of the OBIEE server for each request
3. Forwards each request to the OBIEE server by copying the PeopleSoft headers—including PeopleSoft cookies.
4. Receives and unpacks the OBIEE server response.
5. Loads the unpacked server response into the client.
6. Renders the OBIEE report in the PeopleSoft system

This diagram shows the architecture and the flow of information among the PeopleSoft web server, the PeopleSoft application server, the OBIEE server, and the Integration Broker gateway.

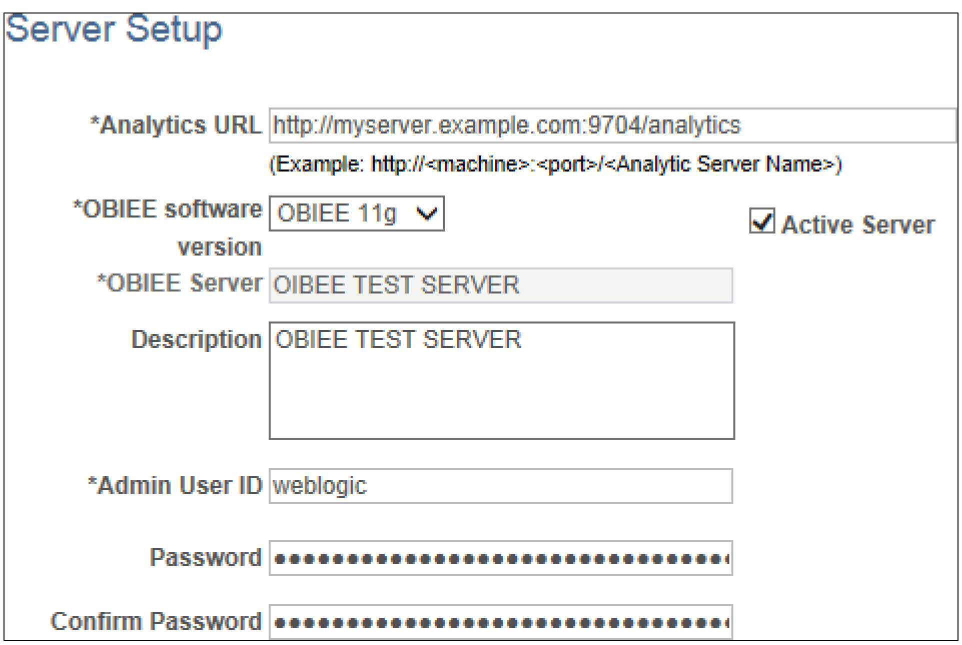


Note: OBIEE developers design and create reports and dashboards in the OBIEE system. The OBIEE server generates the complete HTML markup for the report. The PeopleSoft system does not format the report; rather the HTML passes through the PeopleSoft system directly to the pagelet or tile to be displayed as formatted by the OBIEE developer.
You should include only one OBIEE report on any PeopleSoft page or homepage. PeopleSoft applications do not currently support the appearance of multiple OBIEE reports on the same page, whether from the same OBIEE server or from multiple OBIEE servers.

Configuring OBIEE Server Setup

Access the OBIEE Server Setup page. (Select **PeopleTools** > **Portal** > **OBIEE Setup** > **OBIEE Server Setup**.)

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



Use the OBIEE Server Setup page to define the location and administrative information that is necessary to access the OBIEE reports.

Field or Control	***Description***
Analytics URL	Enter the URL of the OBIEE server, for example, http://srvr12.us.example.com:8739/AnalyticServer1
OBIEE software ver.	Specify the OBIEE software version. PeopleSoft applications support OBIEE 10g and 11g. The default value is *OBIEE 11g*.
OBIEE Server	This value is supplied by default from the OBIEE server name that you define when you add a server.
Active Server	Toggle this check box to make this server available or to take it offline in the PeopleSoft application.
Description	Enter a description of the server definition. This field works well for documentation purposes and when you are using multiple analytic servers.
Admin User ID	Enter the user ID of an OBIEE server administrator. Enter a value to ensure that you do not have to sign in when the system displays the OBIEE pagelet. **Note:** This user should already exist in the OBIEE server.
Password	Enter the OBIEE administrator's password.
Confirm Password	Reenter the OBIEE administrator's password for confirmation.

Configure the Web Server Files

You must include the OBIEE server URL in the configuration.properties file.

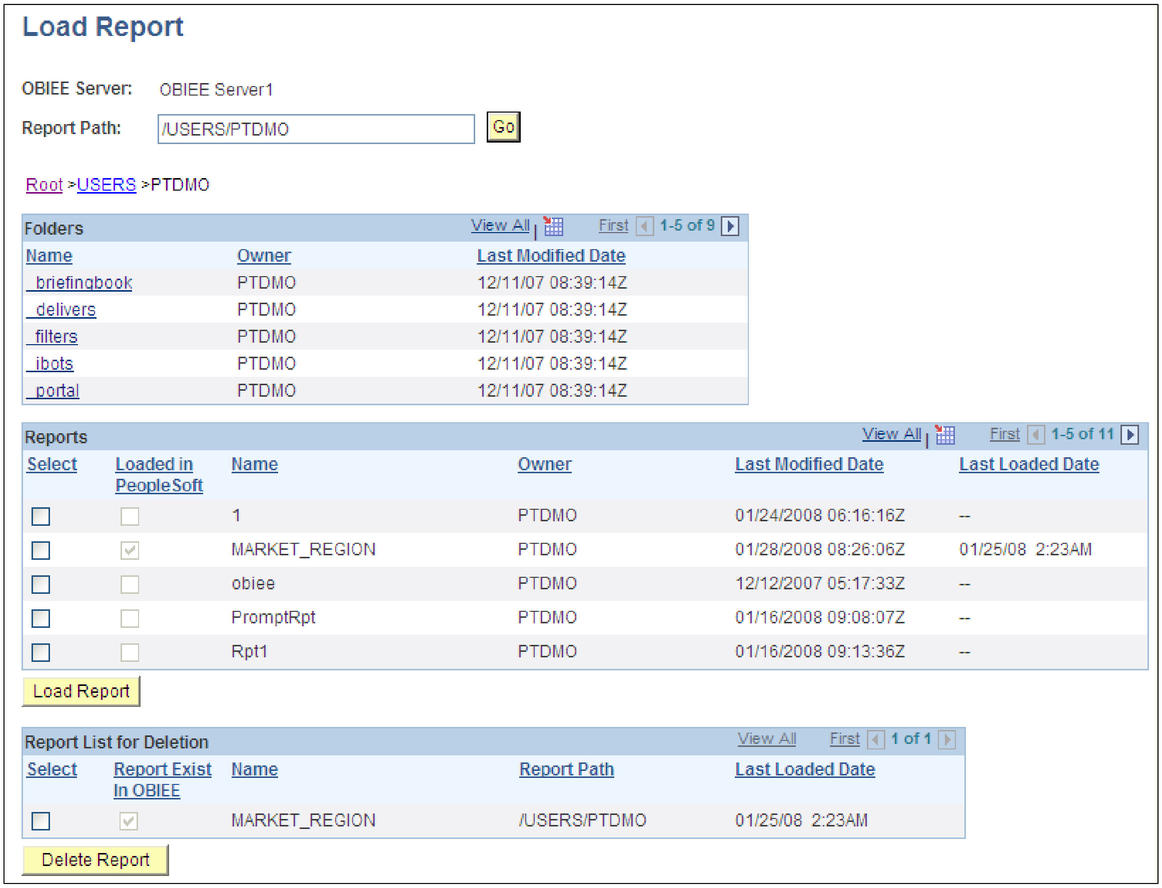
To configure the web server files:

1. Using a text editor, open the configuration.properties file in the web server deployment folder located here: <*Web_server_deployment_home*>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\ps\configuration.properties.
2. Find OBIEEServers= and enter the server URL. If you have multiple OBIEE servers, enter the URLs as a comma separated list.
3. Save the file.
4. Stop and restart the web server.

Loading OBIEE Report Definitions

Access the Load Report page. (Select **PeopleTools** > **Portal** > **OBIEE Setup** > **OBIEE Report Definitions**.)

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



Field or Control	***Description***
Report Path	Enter the directory from which to retrieve OBIEE reports. The format should be */USERS/<PS_UserID>/,* for example */USERS/PTDMO.*
Go	Click to retrieve the OBIEE reports defined in the report path directory

Folders

The PeopleSoft application retrieves this information from the OBIEE system.

Field or Control	Description
Name	This display-only field displays the names of folders within the root directory.
Owner	This display-only field displays the user ID of the owner as indicated in the OBIEE server.
Last Modified Date	This display-only field displays the date that the OBIEE report folder was last modified.

Reports

Field or Control	Description
Select	Select to choose the OBIEE report to load into the PeopleSoft application.
Loaded in PeopleSoft	This display-only field indicates whether the report has been loaded into the PeopleSoft application.
Name	This display-only field indicates the name of the OBIEE report as entered in the OBIEE system.
Owner	This display-only field displays the user ID of the owner as indicated in the OBIEE server.
Last Modified Date	This display-only field displays the date that the OBIEE report was last modified. This information is helpful in determining whether the version of the report loaded into the PeopleSoft application is the most current.
Last Loaded Date	This display-only field displays the date that the OBIEE report was last imported into the PeopleSoft application.

Report List for Deletion

Field or Control	***Description***
Select	Select to choose the OBIEE report to delete from the PeopleSoft application.
Report Exist In OBIEE	This display-only field indicates whether the report has been deleted from the OBIEE server. Pagelets based on reports that you remove from the OBIEE server will *not* produce results in the pagelet.
Name	This display-only field indicates the name of the OBIEE report as entered in the OBIEE system.

Loading OBIEE Reports into PeopleSoft Applications

To load OBIEE reports into PeopleSoft applications:

1. Select the **Select** check box for the reports that you want to load into the PeopleSoft application.

   Note: You can select multiple check boxes.

2. Click the **Load Report** button.

Note: Loading OBIEE reports into the PeopleSoft application does *not* remove the report from the OBIEE server.

Deleting OBIEE Reports from PeopleSoft Applications

To delete OBIEE reports from PeopleSoft applications:

1. Select the **Select** check box for the reports that you want to delete from the PeopleSoft application.

   Note: You can select multiple check boxes.

2. Click the **Delete Report** button.

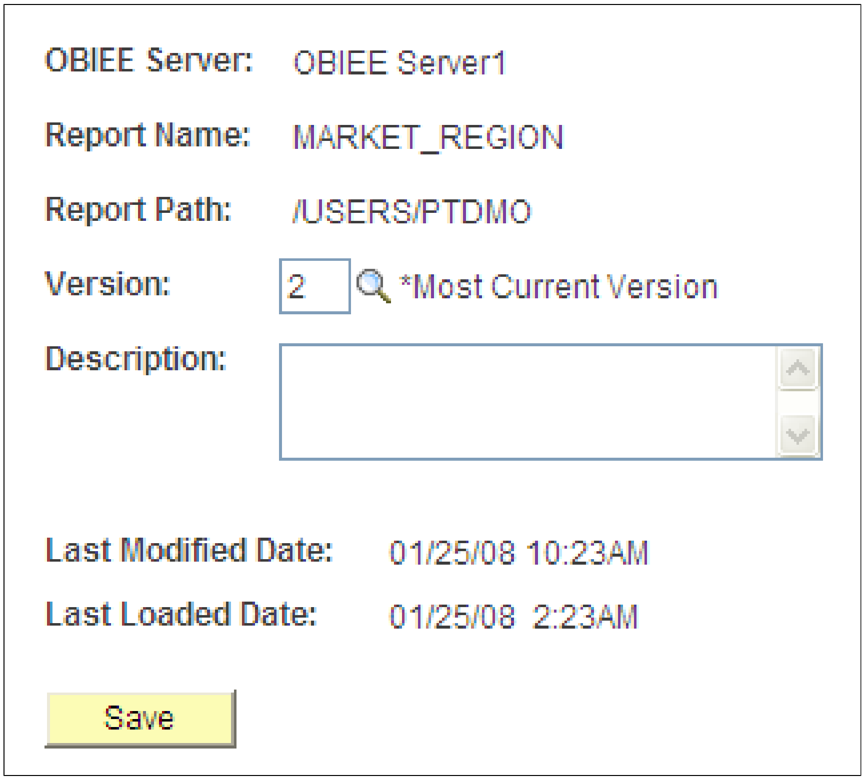
   Note: Deleting OBIEE reports from the PeopleSoft application does *not* delete the report from the OBIEE system.

Viewing OBIEE Report Details

Use the Report Details pages to view information about OBIEE reports.

Access the Report Detail page. (Select **PeopleTools** > **Portal** > **OBIEE Setup** > **View OBIEE Report Details**.)

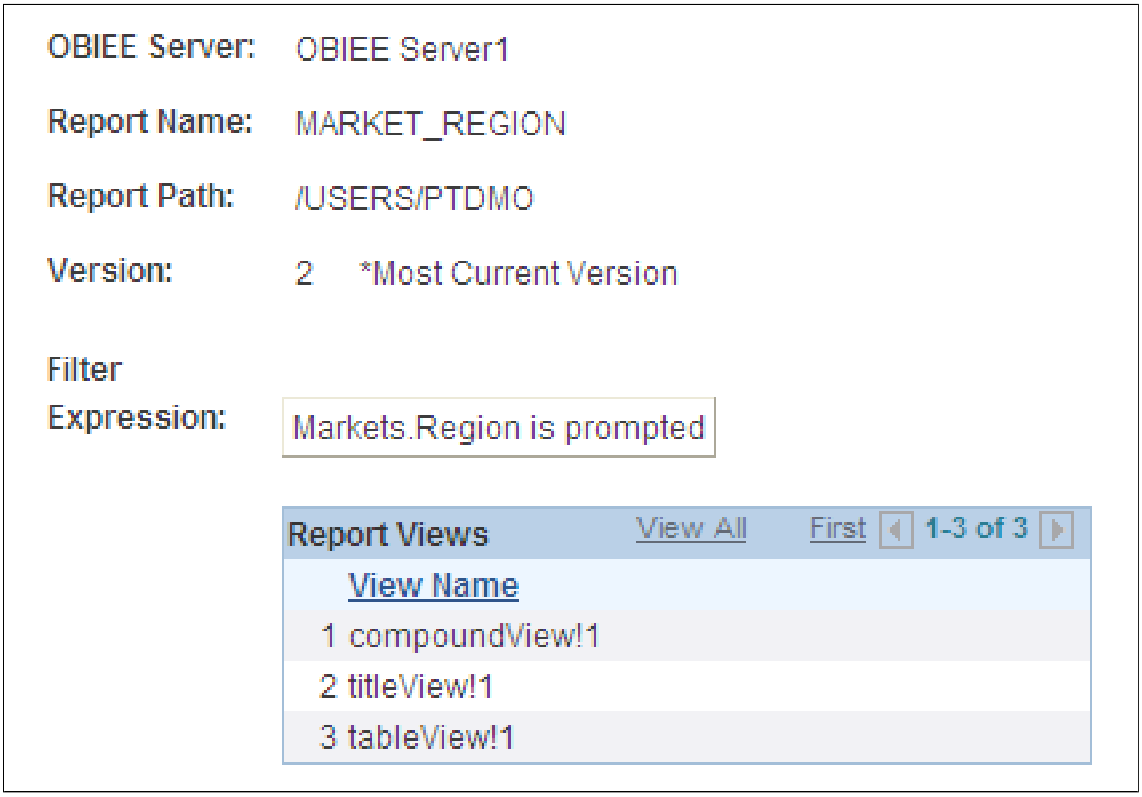
This example illustrates the fields and controls on the Report Detail page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Version	Click the prompt to select the version of the OBIEE report to view. **Note:** Each time you load the same report from the OBIEE server into the PeopleSoft application—regardless of whether the report has changed since last being loaded— the report version number increases by 1.

Access the Report Expression page. (Select the Report Expression tab.)

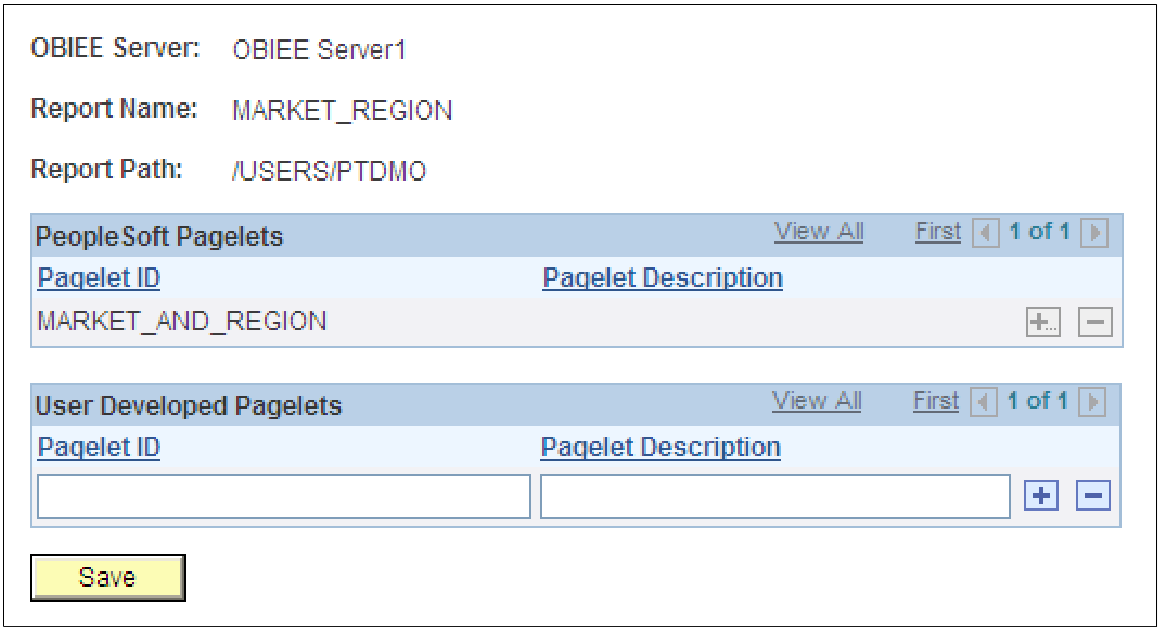
This example illustrates the fields and controls on the Report Expression page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Filter Expression	This display-only field indicates whether the OBIEE report has embedded filters. The *is prompted* filter indicates that the preceding field—in this case **Markets.Region**—is context-sensitive to the target page on the PeopleSoft application.
View Name	This display-only field indicates the types of views created by the OBIEE report builder as stored on the OBIEE server. See Oracle Business Intelligence documentation for explanations of the available types of views.

Access the Report References page. (Select the Report References tab.)

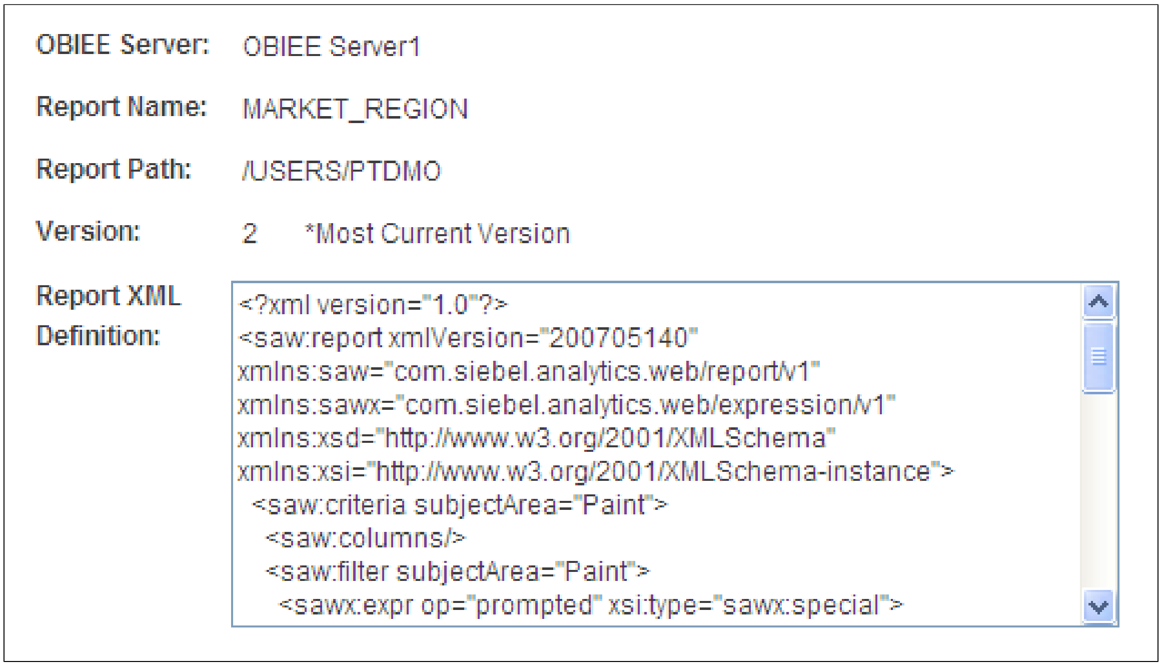
This example illustrates the fields and controls on the Report References page.



Field or Control	***Description***
PeopleSoft Pagelet ID	This value is supplied by default from the *PTPPB_PAGELET_ID* content reference attribute value that you enter on the pagelet Content Ref Administration page.
User Developed Pagelet ID	Enter the pagelet ID of any user-developed pagelets that you want to track. This field is informational only and has no validation behind it.

Access the Xml Definition page. (Select the Xml Definition tab.)

This example illustrates an XML report definition.



Field or Control	*Description*
Report XML Definition	This field displays the XML tags and embedded data for the version of the report as it is currently loaded. Move the cursor inside the field and press **Ctrl + A** to select the text in its entirety.

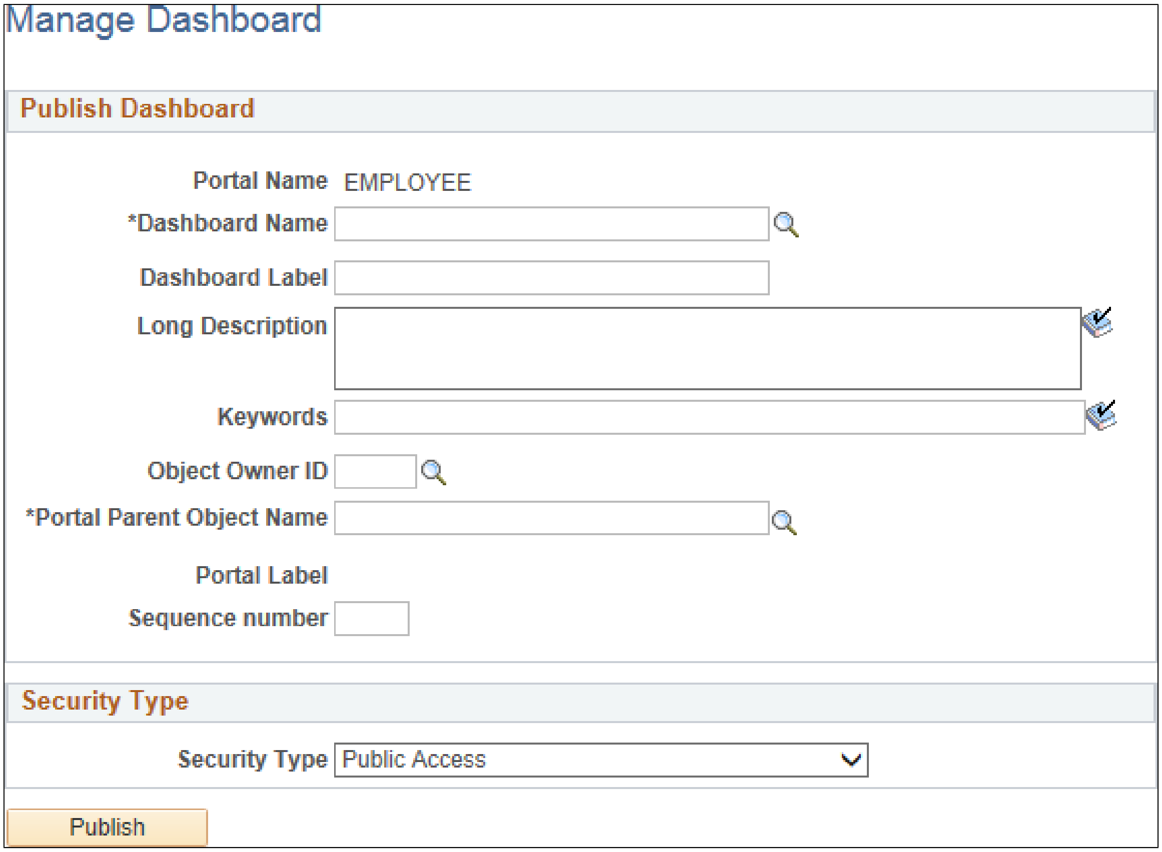
Managing OBIEE Dashboards

Use the Manage Dashboard page to create and publish the definition for an OBIEE dashboard.

Navigation:

PeopleTools > **Portal** > **OBIEE Setup** > **Manage OBIEE Dashboards**

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



Publish Dashboard

Field or Control	***Description***
Portal Name	Displays the name of the current portal.
Dashboard Name	Select an existing dashboard definition from the OBIEE server.
Dashboard Label	Enter a label for this dashboard definition.
Long Description	Enter a description that appears as the tool tip text.

Field or Control	***Description***
Keywords	Enter a keyword or keyword phrase to be stored as a content reference attribute. **Note:** Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language). See Defining Content References for more information on content reference attributes.
Object Owner ID	Select the organization that owns the object.
Portal Parent Object Name	Select the portal registry folder in which you want to place the dashboard definition. Click the look up button to access the Select the Dashboard Link Parent Folder page.
Portal Label	Displays the label for the selected folder.
Sequence number	Enter a number to specify the order of the dashboard link within the parent folder.
Publish	Click the Publish button to save the OBIEE dashboard definition.

Security Type

Use the fields in the Security group box to specify whether access to the dashboard is restricted by permission list and role or unrestricted and available to all users.

Field or Control	***Description***
Security Type	Select from these values: • *Public Access:* Select to allow all users to view and access the dashboard. • *Select Security Access:* Select to specify PeopleTools Security access, which is based on permission lists and roles. When you select this option, the Selected Security grid appears.

Selected Security

Insert rows into the **Selected Security** grid to specify the permission lists and roles that provide access to the dashboard. You can combine permission lists and roles by inserting multiple rows in the grid.

Field or Control	***Description***
Authorization Type	For each row, select from these values: • *Permission List:* Select to secure the dashboard by permission list. • *Role:* Select to secure the dashboard by role.
Authorization Name	Specify the permission list or role name that provides access to the dashboard. Click the look up button to see a list of available names.

Creating OBIEE Pagelets and Tiles

This section discusses how to use Pagelet Wizard and Tile Wizard to create pagelets from OBIEE reports and tiles from OBIEE reports and dashboards:

Using Pagelet Wizard to Create OBIEE Pagelets

To create OBIEE report pagelets:

1. Load the OBIEE report into the PeopleSoft application.
2. Select **PeopleTools** > **Portal** > **Pagelet Wizard** > **Pagelet Wizard.**

   Pagelet Wizard guides you through the process, by displaying a sequence of pages. Each page corresponds to a step in the process. A guide appears in a header above the page to indicate your progress.
3. Specify the pagelet information in step 1.

   See Step 1: Entering Pagelet Identifying Information.
4. Specify *OBIEE* as the data type in step 2. In addition, select the report ID and report view ID from which the pagelet will read data.

   Note: If you have not loaded the OBIEE report into the PeopleSoft application, this step fails.

   See Selecting an OBIEE Report Data Source.
5. Specify the data source parameters.

   See Specifying OBIEE Data Source Parameters.

6. Select *Passthru* as the display format.
7. Specify display options, such as additional text and a search capability.

   See Step 5: Specifying Pagelet Display Options.
8. Specify the publishing options.

   See Specifying Pagelet Publication Options.

Using Tile Wizard to Create OBIEE Tiles

To create tiles from OBIEE reports or dashboards:

1. Load the OBIEE report into the PeopleSoft application.
2. Select **PeopleTools** > **Portal** > **Tile Wizard**.

   Tile Wizard guides you through the process, by displaying a sequence of pages. Each page corresponds to a step in the process. A guide appears in a header above the page to indicate your progress.
3. Specify the basic tile information in step 1.

   See Step 1: Tile Basic Information.
4. Specify *OBIEE* as the data type in step 2. In addition, select the report ID or dashboard ID from which the tile will read data. In addition, you will specify data source parameters at this step.

   Note: If you have not loaded the OBIEE report into the PeopleSoft application, this step fails.

   See Step 2: Data Source Information - OBIEE.
5. Specify target page information.

   See Step 3: Target Page Information.
6. Specify tile display options such as size, interactivity, and so on.

   See Step 4: Tile Layout Properties.
7. Publish the tile definition.

   See Step 5: Review and Submit.

Chapter 19

Configuring Interwindow Communication

Understanding Interwindow Communication

This topic provides an overview of interwindow communication and discusses:

- Interwindow communication process flow.
- Interwindow communication events and messages.
- Interwindow communication considerations and limitations.

Related Links

Special Uses of Interwindow Communication

Interwindow Communication

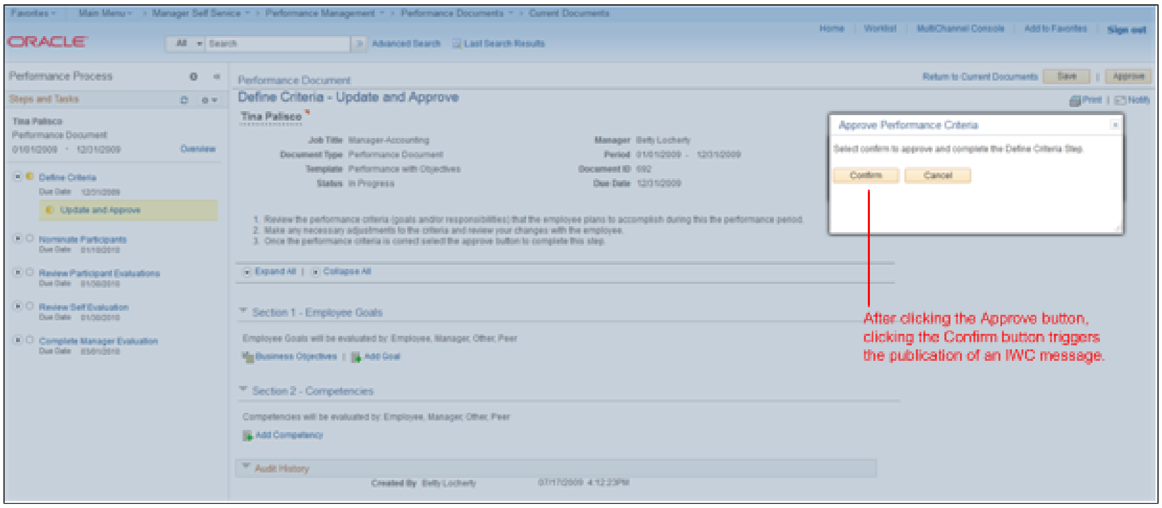
Interwindow communication (IWC) is a frame-to-frame messaging mechanism that enables browser window objects—such as pages or pagelets—to be aware of and react to field-level data changes that occur in other browser window objects that share the same browser session. Interwindow communication enables you to publish field-level changes as they occur so that any subscribing page or pagelet becomes instantly aware of the change and then, can act on it in real time.

Interwindow communication does not depend on any specific portal template, frame, or iframe and works throughout PeopleSoft applications in many scenarios for which you require interaction between browser window objects. You can configure interwindow communication to operate between pagelets, between the page and a pagelet, and also within the same pagelet, if necessary. Although interwindow communication is ideal for use with related content and WorkCenter pages, interwindow communication is independent of both technologies. In addition, interwindow communication does not depend on Integration Broker.

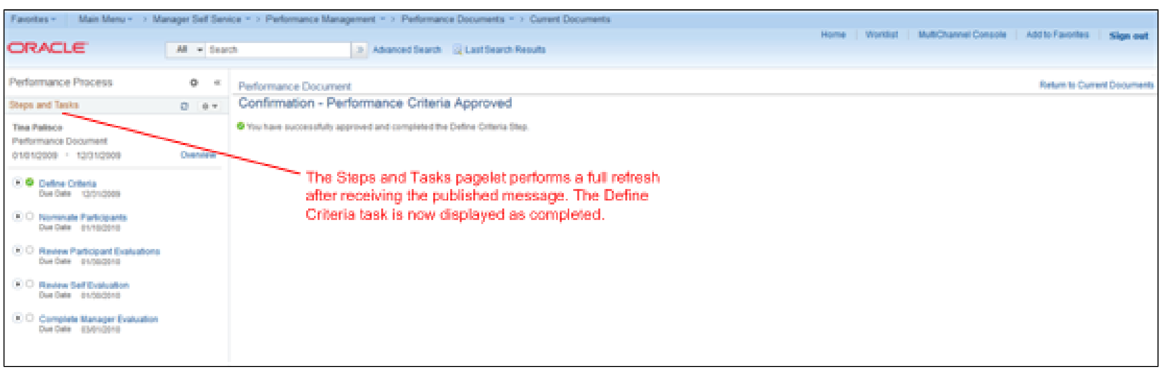
In a simple example, a dashboard includes the Employees pagelet, which contains a list of employee names. The dashboard also includes an Employee Projects pagelet, which contains the project listing for a specific employee. Interwindow communication enables you to click a name in the Employees pagelet and *simultaneously* have the Employee Projects pagelet retrieve the project listing for the selected employee.

In a different example, interwindow communication can be used to refresh a pagelet with data that has been updated in the database. For example, the target content area of a WorkCenter displays transactions related to the benefits enrollment process. The pagelet area of the WorkCenter includes a Benefits Summary pagelet that itemizes the cost to the employee of each benefit selection. When the employee saves a benefit selection on a transaction page in the WorkCenter, standard component processing saves the data to the database. When the server processing returns, the interwindow communication publish event fires indicating that a user action has occurred. The Benefits Summary pagelet subscribing to this event performs a full refresh to retrieve the revised data from the database.

In the following example from a performance management WorkCenter, a publication event on the transaction page causes a full pagelet refresh in the pagelet area. The WorkCenter page displays a document for approval. After clicking the Approve button, the action of clicking the Confirm button triggers the publication of an interwindow communication message.



The Steps and Tasks pagelet on the left has subscribed to this interwindow communication message and performs a full refresh after receiving the published message. The Define Criteria task in the pagelet now shows that it has been completed:

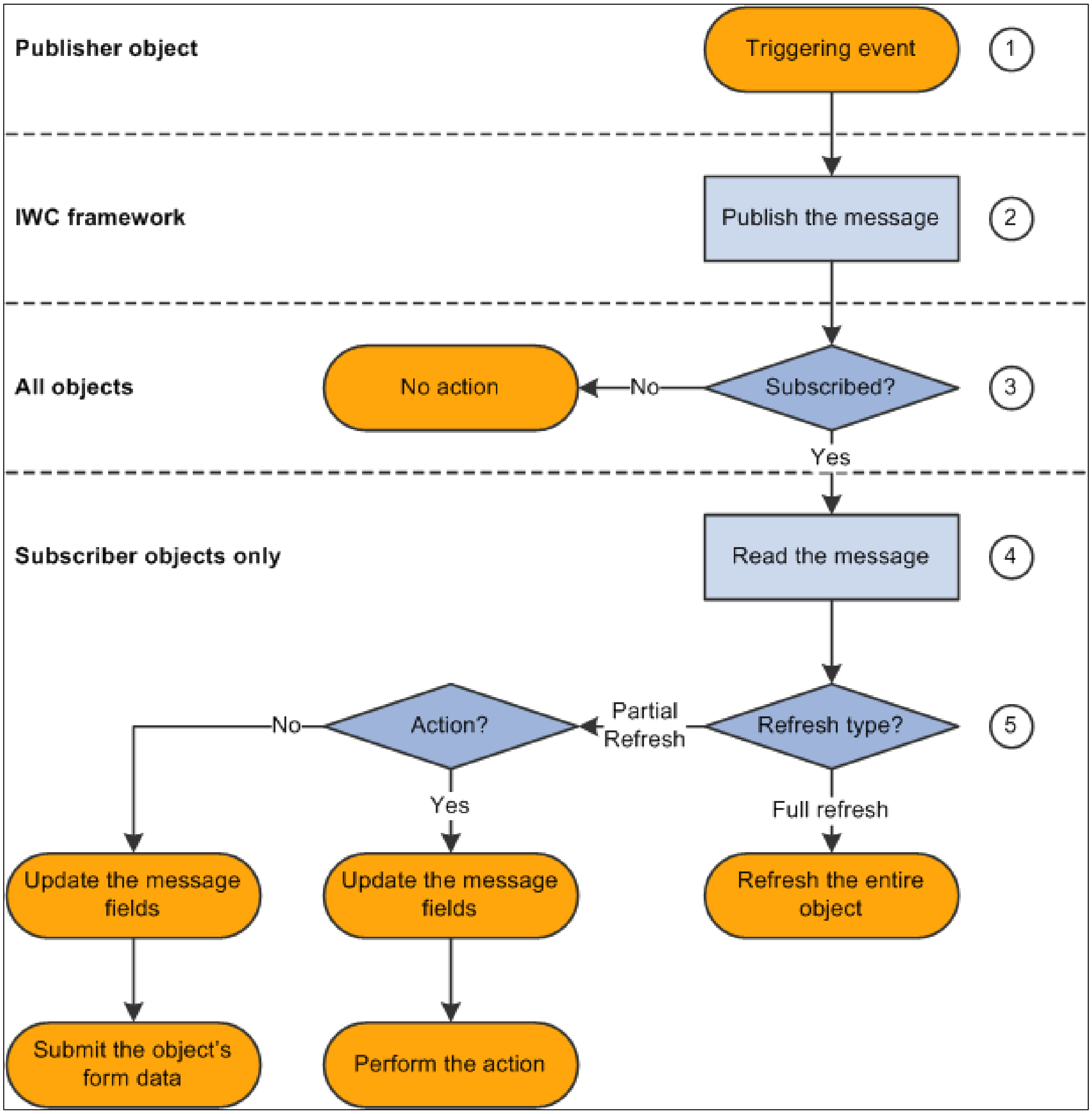


Interwindow Communication Process Flow

Interwindow communication interactions occur according to this simplified process flow:

1. The triggering event occurs (click or change).
2. The interwindow communication message is published.
3. Each object (pagelet or page) determines whether it is subscribed to the message.
4. Subscriber objects only read the message.
5. Subscriber objects only act according to the subscription definition.

The following diagram presents the flow of these actions and events, which comprise the publish and subscribe process of interwindow communication:



The following table presents the process flow in more detail:

Actor	***Step***	***Description***
Publisher object	**1.** Triggering event occurs.	User clicks a button or link or changes the value of a field that is configured for publication.
Interwindow communication framework	**2.** Publish the message.	A simple string formatted message is sent to the top window in the browser session—for example, that would be the homepage or dashboard window.

Actor	***Step***	***Description***
All objects	**3.** Determine whether subscribed.	All objects determine whether they are subscribed to the event.
Subscriber objects only	**4.** Read the message.	All objects in the window that are subscribed to the event read the message.

Subscriber objects only	**5.** Act according to the subscription definition.	Objects in the window that are subscribed to the event do one of the following:
		• If a full refresh is set, refresh the entire object (with URL parameters if specified). Any message data is essentially lost.
		• If a partial refresh is set:
		  • If an event action is defined, refresh only the field(s) specified in the subscribing message definition with the published message data *and then* perform that action.
		  • Otherwise, refresh only the field(s) specified in the subscribing message definition with the published message data *and then* submit the form data.
		Important! Submitting the form data forces a trip to the application server regardless of whether the field or component is in deferred processing mode.

Note: Interwindow processes occur *after* any application server interactions invoked by the user's action. For example, a publication event could be defined for clicking a link, which merely retrieves data from the database, or it could be defined for clicking a Save button, which invokes component save processing to save updated data to the database. Interwindow communication fires after the server interaction returns to the browser.

Interwindow Communication Events and Messages

Interwindow communication involves publishing and subscribing to messages.

Triggering Events

The first half of interwindow communication is a JavaScript event that triggers the publication of a message:

- **OnChange**

  The user changes a value in a field on a transaction page or pagelet.
- **OnClick**

  The user clicks a button or link on a transaction page or pagelet.

Message Subscriptions

The second half of interwindow communication is the consumption of the published message by subscribing objects only:

- **Full refresh**

  If full refresh is set for the object, the entire object is refreshed from the database (with URL parameters if specified). Therefore, any message data is essentially lost.
- **Partial refresh**

  If partial refresh is set for the object, only the data as specified in the message definition is updated, not the entire object.

Important! In addition, by requiring that some HTML field be identified as the acted upon field, interwindow communication ensures that a server trip is invoked regardless of whether the component or field is in deferred processing mode.

Interwindow Communication Considerations and Limitations

Page and Pagelet Types

This table describes the types of pages and pagelets that interwindow communication supports and the type of refresh that is available to each page and pagelet type:

Page or Pagelet Type	***Description***	***Refresh Support***
Transaction pages	The standard PeopleSoft page that appears in the target content area.	• Partial refresh. • Full refresh.
Component-based pagelets	A pagelet built by using Application Designer.	• Partial refresh • Full refresh.
iScript or Pagelet Wizard pagelets	The PeopleCode-based pagelets built by using Pagelet Wizard or by registering an iScript.	Full refresh.

Considerations and Limitations

Note the following considerations when implementing interwindow communication:

- The domain for interwindow communication is the entire, *visible* HTML page.

  This domain includes all pagelets and active transaction pages visibly displayed, but excludes any tabbed pages of the component definition that are currently not displayed (not active). Therefore, the visible HTML page can encompass multiple component and page definitions.

- A subscribing page or pagelet must be completely loaded or it will fail to consume a published message.

- A field that will participate in interwindow communication must have a unique HTML field ID within this domain.

  Use developer tools in the browser to determine HTML field IDs. See Identifying HTML Field IDs.

- Component definitions for objects participating in interwindow communication must remain in deferred processing mode.

  Page definitions and field definitions can be set to either deferred or interactive mode as needed.

- A publishing field or a subscribing field can include FieldChange PeopleCode.

- However, a publishing field *should not* include FieldEdit PeopleCode, which typically includes errors and warnings.

  An interwindow communication event will not fire on a publishing field that includes FieldEdit PeopleCode.

- If a non-visible field is required to either publish or subscribe to an interwindow communication event, change the style of the page field properties to PTTRANSPARENT (on the Record tab in Application Designer).

  Do not set the field to invisible (on the Use tab); fields set to invisible in this manner do not get generated as part of the HTML.

- In a modal secondary window, you cannot configure a publication event on the close icon (the "X" with id="ptModClose_0") in the upper right corner of the window.

  Instead, you must configure the publication event on a button such as a Save, OK, or Cancel button within the window.

Publishing and Subscribing to IWC Messages

This topic discusses:

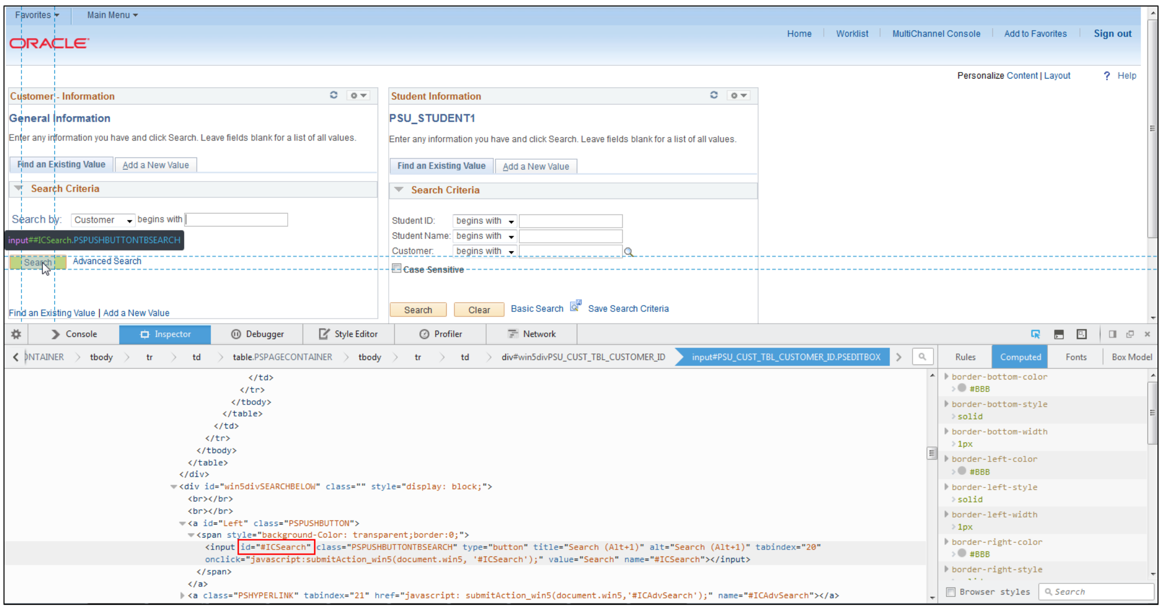
- Identifying HTML field IDs.
- Publishing interwindow communication messages.
- Subscribing to interwindow communication messages.

- Special uses of interwindow communication.

Identifying HTML Field IDs

To participate in interwindow communication as a triggering field, as the acted upon field, or as a message data field, the HTML field must have a value for the id attribute, and this value must be unique within the HTML page domain. You can use your browser's built-in developer tools to examine the HTML code for a page and to extract the value of the id attribute for a field. Developer tools are available as part of Chrome, Firefox, and Safari. In addition, there are third-party add-ons (for example, Firebug) that provide an alternative toolset for web developers.

In the following example showing the web developer tools integrated into Firefox, the Search button in the Customer Information pagelet has been selected.



In developer tools pane at the bottom of the browser window, the corresponding line of code has been highlighted by the web developer tool:

```
<input id="#ICSearch" class="PSPUSHBUTTONTBSEARCH" type="button" title="Search (Alt⇒

+1)" alt="Search (Alt+1)" tabindex="20"
onclick="javascript:submitAction_win5(document.win5, '#ICSearch');" value="Search" ⇒

name="#ICSearch"></input>
```

Within that line of HTML code, the id attribute and its value (`#ICSearch`) has been enclosed in a red rectangle:

```
id="#ICSearch"
```

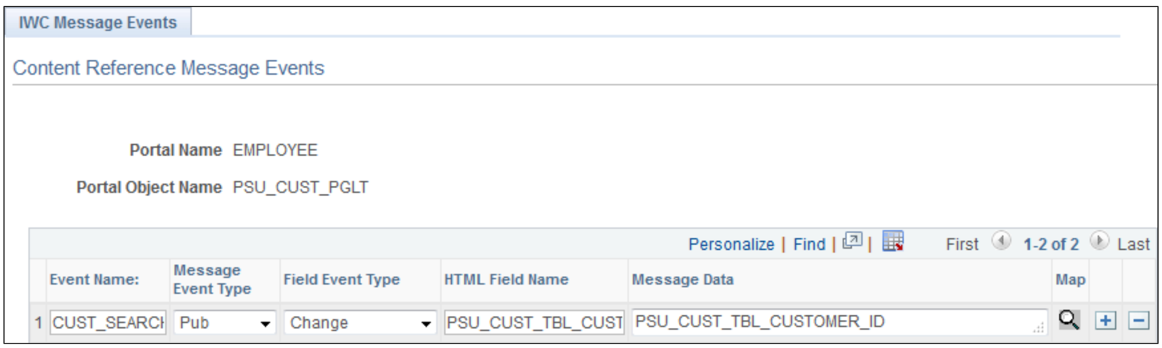
Publishing Interwindow Communication Messages

Use the IWC Message Events page to define interwindow communication publication and subscription events.

Navigation:

- PeopleTools, Portal, Inter Window Communication.
- Navigate to the content reference definition:
  1. PeopleTools, Portal, Structure and Content.
  2. Browse through the folder hierarchy in the portal registry.
  3. Click the Edit link for a content reference definition.
  4. Click the IWC Message Events link.

This example illustrates the fields and controls on the IWC Message Events page for a publication event.



To create an interwindow communication publication message:

1. In the Event Name field, enter a name for this message. Because messages are subscribed to by event name only, this value should be unique for all messages defined within this portal.
2. Select *Pub* as the message event type.
3. Select either *Change* or *Click* as the field event type.
4. In the HTML Field Name field, enter the HTML field ID of the *triggering field* as it appears in the HTML source.

   Important! To participate in interwindow communication as the triggering field, as the acted upon field, or as a message data field, the HTML field must have a value for the id attribute, and this value must be unique within the HTML page domain. See Identifying HTML Field IDs for more information.

5. Enter the HTML field ID of the field (or fields) that you want to publish as the message data. Separate multiple field IDs with a comma.

   Note: The triggering field and the message data field are not necessarily the same field.

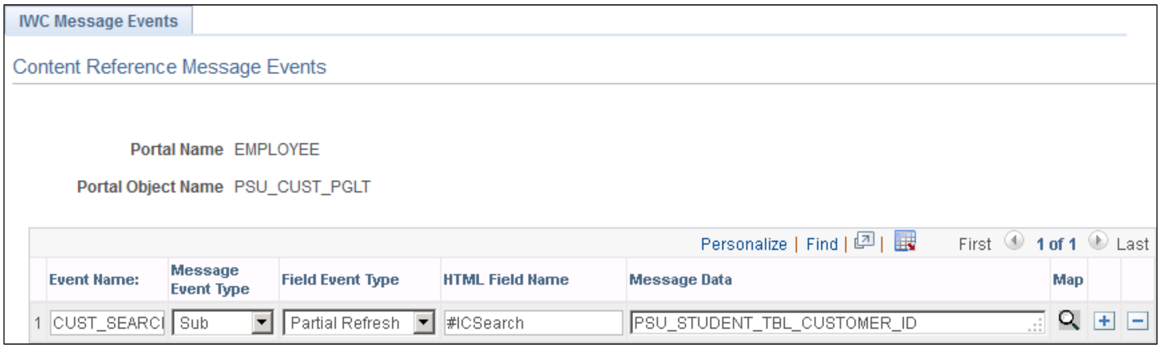
6. Save your changes.

Note: While the Map button is displayed for all interwindow event definitions, the Map button is active only for push notification event subscriptions of type *Server Sub*. See Special Uses of Interwindow Communication.

Subscribing to Interwindow Communication Messages

Use the IWC Message Events page to define interwindow communication publication and subscription events.

This example illustrates the fields and controls on the IWC Message Events page for a subscription event.



To subscribe to an interwindow communication message:

1. In the Event Name field, enter the name of the message as it was defined in the publication event.

2. Select *Sub* as the message event type.

3. Select either *Full Refresh* or *Partial Refresh* as the field event type.

   Important! For iScript and Pagelet Wizard pagelets, a full refresh is the only valid event type.

4. In the HTML Field Name field for a partial refresh subscription only, enter the HTML field ID of the *acted upon field* as it appears in the HTML source.

   Important! For a full refresh subscription, the *field to be acted upon* is required to be blank.

   Important! To participate in interwindow communication as the triggering field, as the acted upon field, or as a message data field, the HTML field must have a value for the id attribute, and this value must be unique within the HTML page domain. See Identifying HTML Field IDs for more information.

5. In the Message Data field, enter the HTML field ID of the field (or fields) that you want to receive the message data. Specify receiving fields in the same order as the message data is defined in the publishing message; separate multiple fields IDs by a comma.

   Note: For a full refresh subscription, the *message data field* is optional.

6. Save your changes.

Note: While the Map button is displayed for all interwindow event definitions, the Map button is active only for push notification event subscriptions of type *Server Sub*. See Special Uses of Interwindow Communication.

Special Uses of Interwindow Communication

Several PeopleTools features use interwindow communication in specific and unique manners:

- Interwindow communication can be used to synchronize changes between pivot grid pagelets. The publishing field name is always *PVGPUB*; the subscribing field name is always *PVGSUB*. In addition, there is a specific nomenclature for specifying a filter or a prompt as either the published message data or as the subscribing field.

  See “Pivot Grid Pagelet Overview” (Pivot Grid) for more information.

- Interwindow communication can be used to subscribe to push notification events defined on the server. Always select *Server Sub* as the event type; then use the Map Event Data To Page Elements page to complete configuration of the remaining items by mapping event data to page elements.

  See “Subscribing to Events” (Fluid User Interface Developer’s Guide)for more information.

Chapter 20

Using the PeopleSoft Related Content Framework

Understanding the PeopleSoft Related Content Framework

This section lists prerequisites and limitations, and discusses:

- PeopleSoft Related Content Framework flexibility.
- Explicit definition versus anonymous services.
- Fluid related content.
- The related content development process.
- Delivered related content services for PeopleSoft Interaction Hub.
- Delivered related content services for Oracle WebCenter.

The PeopleSoft Related Content Framework enables developers and subject matter experts to link application pages with contextually relevant collaborative content, which creates opportunities for the user to view or quickly access information that supports, extends, and augments any PeopleSoft transaction. The PeopleSoft Related Content Framework ties together all types of content such as queries, discussion threads, PeopleSoft component data, Oracle Business Intelligence Enterprise Edition (OBIEE) analytics, queries, discussion threads, and any relevant non-PeopleSoft data on PeopleSoft application pages.

Related content services can be any collaborative, analytical, or informational content that is useful for completing business tasks. Related content services include any content such as PeopleSoft components; pagelets or tiles that are offered as services to be consumed by applications; or related discussion forums, wiki pages, or links. Related content services can be configured within the same database, among multiple PeopleSoft databases, and between a PeopleSoft database and external sources.

Prerequisites

This table lists the necessary and optional prerequisites to defining and configuring related content services:

Task	***Reference***
Enter an authentication domain in the web profile.*	See Configuring General Portal Properties.
Define the content URI text and portal URI text for each active node in your system.*	See Setting Portal Node Characteristics

Task	***Reference***
Configure single signon between your system and any systems providing remote content.	See "Understanding Single Signon" (Security Administration).
Configure the security authorization service, if required, as required by each related content service.	See "Understanding Developing and Invoking the Security Authorization Service" (Security Administration).

* Indicates the prerequisite is required.

Limitations

Note the following limitations on the display and use of related content:

- A component must be registered in order to select it to assign related content.
- The related content menu and the Related Content frame will not be displayed on pages that have the pagebar option set to disabled.
- The related content menu and the Related Content frame will not be displayed for modal or secondary pages.
- If the modal or secondary page has a related actions menu or glyph on a field that has a service targeted to open in *Target Content,* the frame will be opened in a new window instead.
- The Related Content frame configured for a classic page will not be displayed when that page is rendered inside any fluid wrapper, such as in a fluid activity guide or in the fluid master/detail container.

PeopleSoft Related Content Framework Flexibility

The PeopleSoft Related Content Framework provides flexibility in how related content is defined, assigned, and accessed. You can assign, configure, and maintain certain types of related content services without access to developer tools such as Application Designer. The related content service pages provide a variety of configuration options, which enable you to define related content services, associate them with pages or pivot grids and grant access to these services based on your business rules. This flexible framework allows you to:

- Determine whether information is displayed as related content or related actions.
- Specify the placement of the Related Content frame at the bottom or on the right of a classic page.

  Note: On fluid pages, the Related Information panel displays on the right only.

- Configure the layout of related content and actions menus.
- Assign explicitly defined services or anonymous (non-explicit) services.
- Define local or remote resources as related content services.

Related Content Versus Related Actions

The PeopleSoft Related Content Framework allows related content services to be deployed in two ways—as related content and as related actions—based on where the content appears when the user accesses the service.

- *Related content* appears in the Related Content frame only, which can be selected to be displayed at the bottom or to the right of the transaction content.
- *Related actions* appear outside of the Related Content frame in a modal window, in a new browser window, in the existing browser window replacing the current transaction content, or in the existing browser window replacing the entire page.

Configure the related content service as *related content* by selecting the Configure Related Content page:

- If you want the related content to appear in a separate frame beside or below the transaction content.
- If you want the related content to be dynamically updated when the user moves to a different transaction instance, or changes a field value, and so on.

Configure the related content service as a *related action* by selecting the Configure Related Actions page:

- If you want the related content to appear in a modal window, in a new browser window, or to replace the current page.
- When you are configuring related content for pivot grids or keyword search results.

The Classic Related Content Frame

The Related Content frame can be configured to display at the bottom or to the right of the classic transaction content.

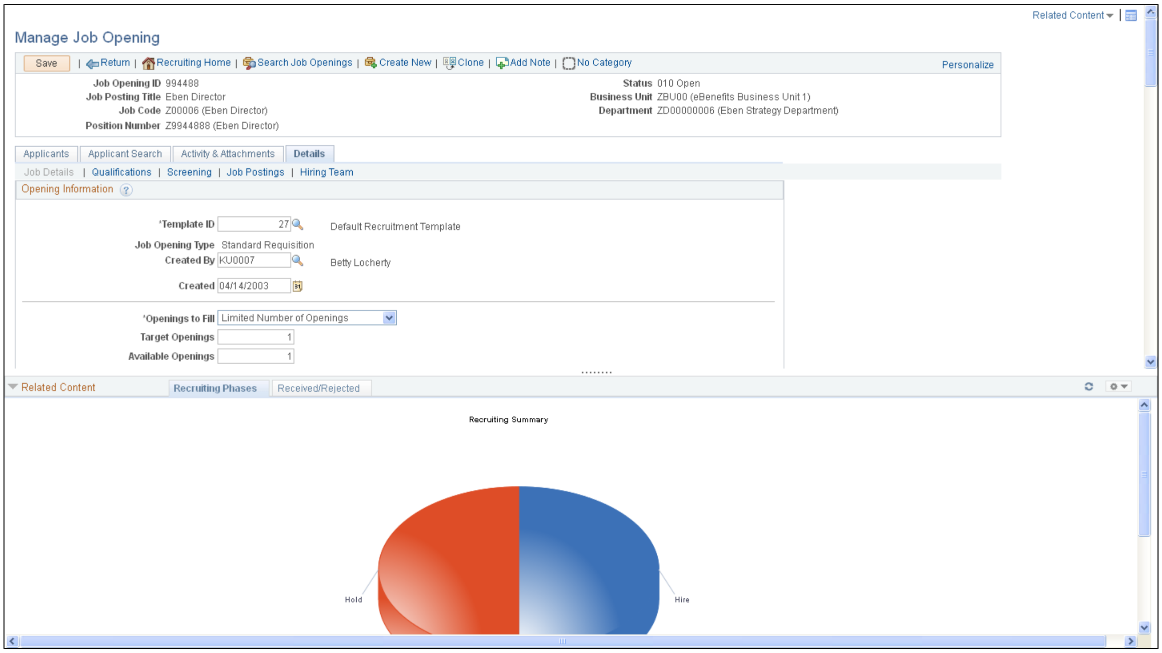
Note these points about the Related Content frame when it appears at the bottom of a classic page:

- The frame extends across the width of the transaction page.
- The visible area increases and decreases when you drag the handle on the upper edge of the frame in a vertical direction.
- The frame appears and disappears when you click the **Expand/Collapse** button on the left side of the frame header.
- The frame reloads when you click the **Refresh** button in the frame header .
- The frame loads the current active service in a new window when you click the **Zoom** icon.
- The frame closes when you click the **Close** button.

Note these points about the *services* that appear in the bottom frame:

- Only one service can be active and have focus at a time.
- Multiple services appear as tabs across the top of the frame.
- Users click the tab to bring the service into focus.

The following example shows a classic page with related content in the Related Content frame at the bottom of the page:



Note these points about the Related Content frame when it appears at the right of a classic page:

- The frame extends down the length of the transaction page.
- The visible area increases and decreases when you drag the left edge of the frame in a horizontal direction.
- The frame appears and disappears when you click the **Maximize** and **Minimize** buttons on the left border of the frame.
- The frame divides the visible area equally among all open pagelets.
- The bottom of the frame displays a **More** menu, which you use to access minimized pagelets.

Note these points about the *services* that appear in the right frame:

- Each service exists as a self-contained pagelet inside the Related Content frame.
- Any number of services can open be active and in the expanded state simultaneously.
- All maximized service pagelets divide the available space equally and always share the same vertical and horizontal dimensions. As users maximize and minimize other service pagelets in the frame, any service pagelets that remain open resize automatically so that the space appears equally divided.

  Note: Users cannot resize individual service pagelets.

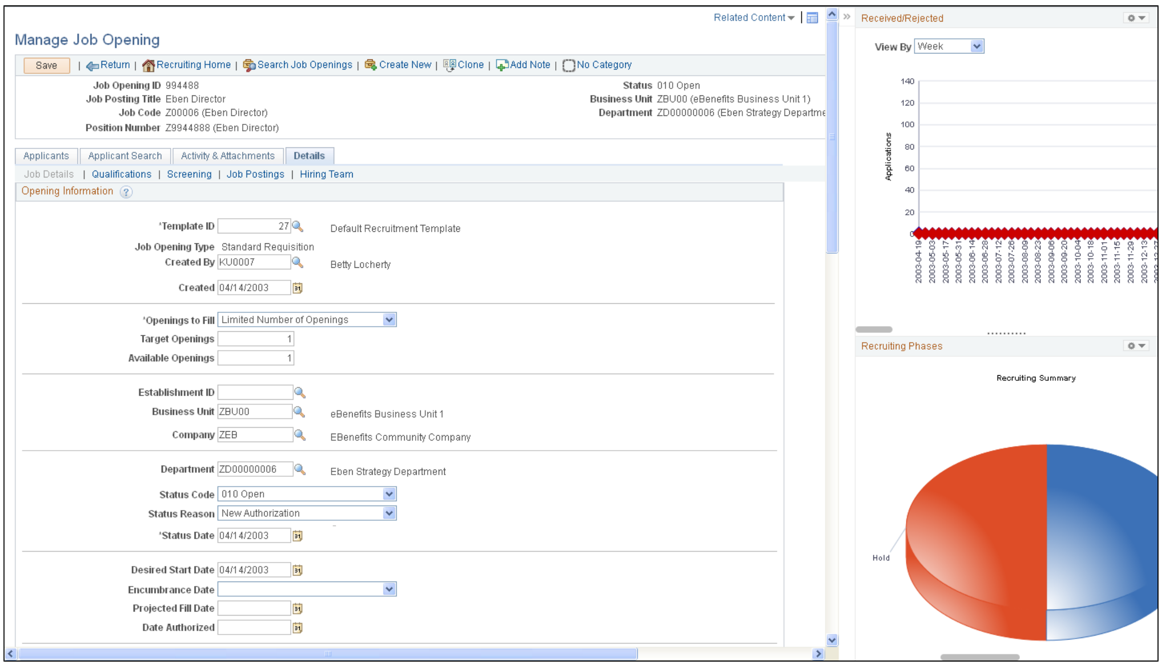
- Each pagelet header can contain:
  - A **Zoom** button that users click to display the service in a new window.
  - A **Refresh** button that users click to reload the service.

- A **Maximize/Minimize** button that users click to toggle the service between the expanded and collapsed states.

  Note: The expanded or collapsed state of each pagelet persists between sessions.

- Scroll bars appear on the right and bottom of a service pagelet if it is too small to fully display the content.

The following example shows a classic page with related content in the Related Content frame at the right of the page:

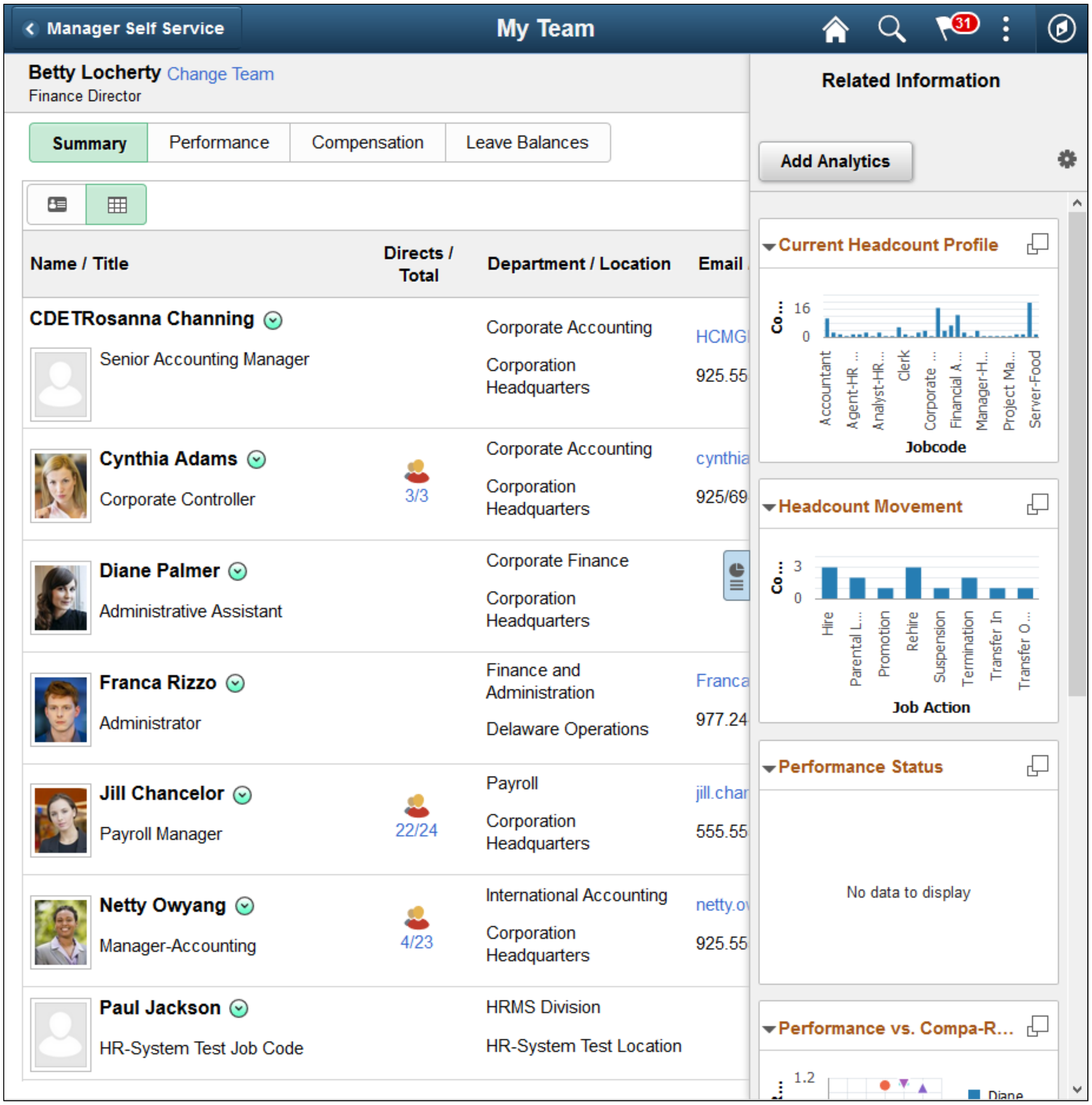


The Fluid Related Information Panel

Note these points about the Related Information panel on a fluid page:

- The frame, labeled Related Information, always displays on right.
- Each service exists as a self-contained grouplet inside the Related Information panel.
- Grouplets are always initially expanded. However, users can collapse or expand grouplets.
- Each grouplet header includes a zoom button to display the content in a modal window
- The frame can also include simplified analytics.

The following example shows a fluid page with related content in the Related Information panel at the right of the page:



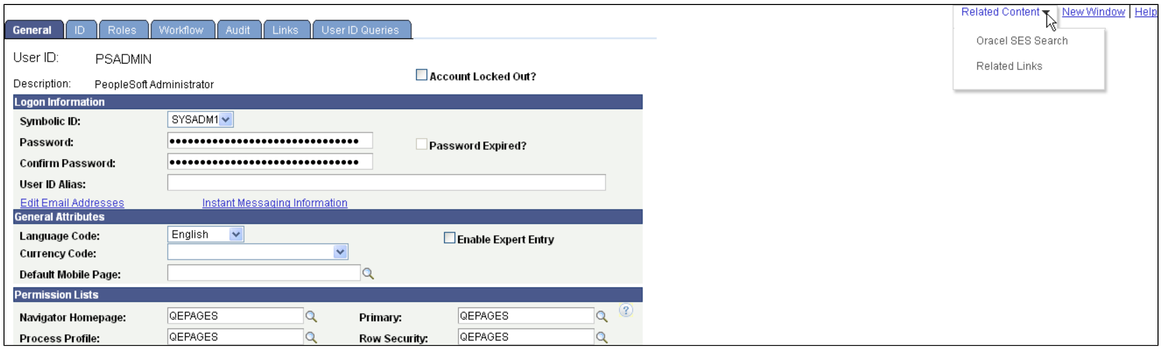
Related Content Menus

Three types of menus provide access to related content services:

- The Related Content menu, which appears in the page bar at the top of a classic page. The Related Content menu displays services configured as *related content* only.
- The related actions menus, which can appear on any page that includes the action widget, on pivot grids, in keyword search results, in SmartNavigation folders, and on chart nodes in a SmartNavigation chart. The related actions menus displays services configured as *related actions* only.
- Page field contextual menus on classic pages or fluid pages, and which appear when you click the page-field glyph. On a classic page, a page field contextual menu can combine both *related content* and *related actions*.

The Related Content menu, which appears in the page bar at the top of a classic page, displays services configured as *related content* only.

The following example shows two services in the Related Content menu:

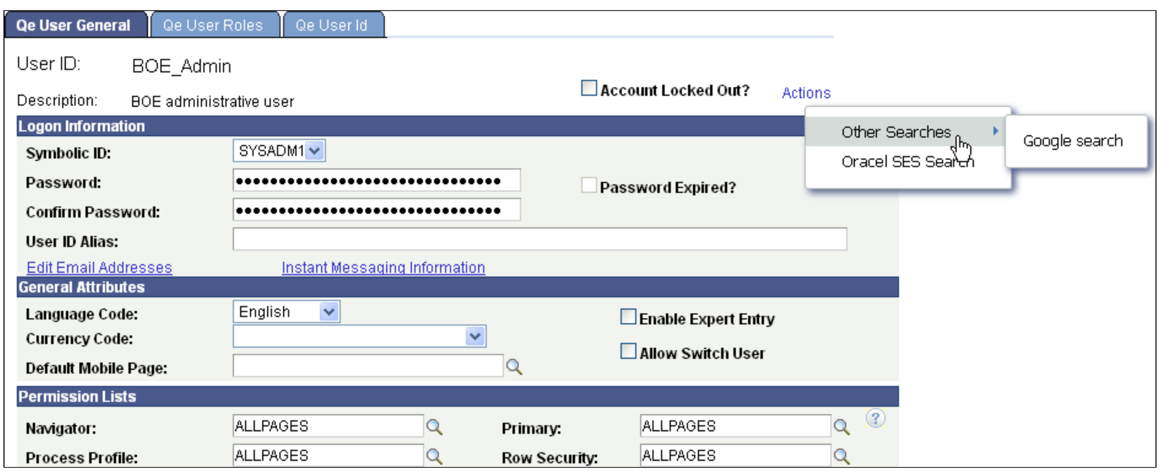


The related actions menus, which can appear on any page that includes the action widget, on pivot grids, in keyword search results, in SmartNavigation folders, and on chart nodes in a SmartNavigation chart. The related actions menus displays services configured as *related actions* only. A related action menu can appear as:

- The action widget on any page.

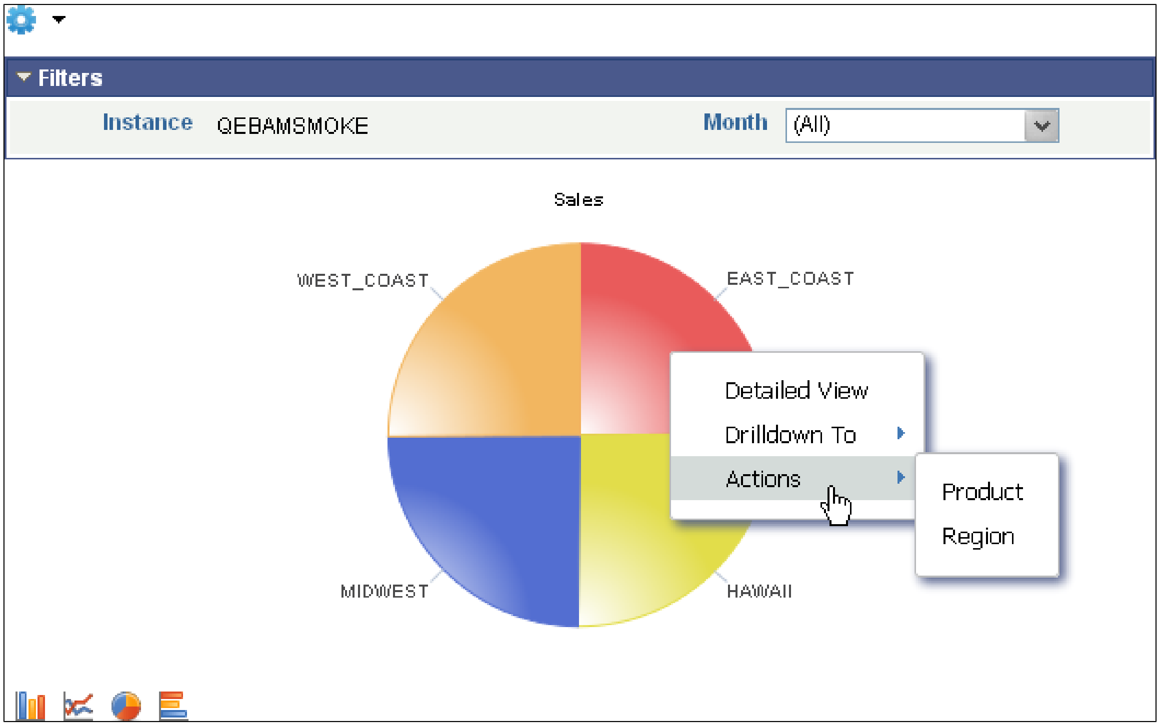
  Note: The label on the action widget (Actions) is configurable within Application Designer.

  This example shows the action widget (the Actions link) that has been added to the page to display a list of related actions:

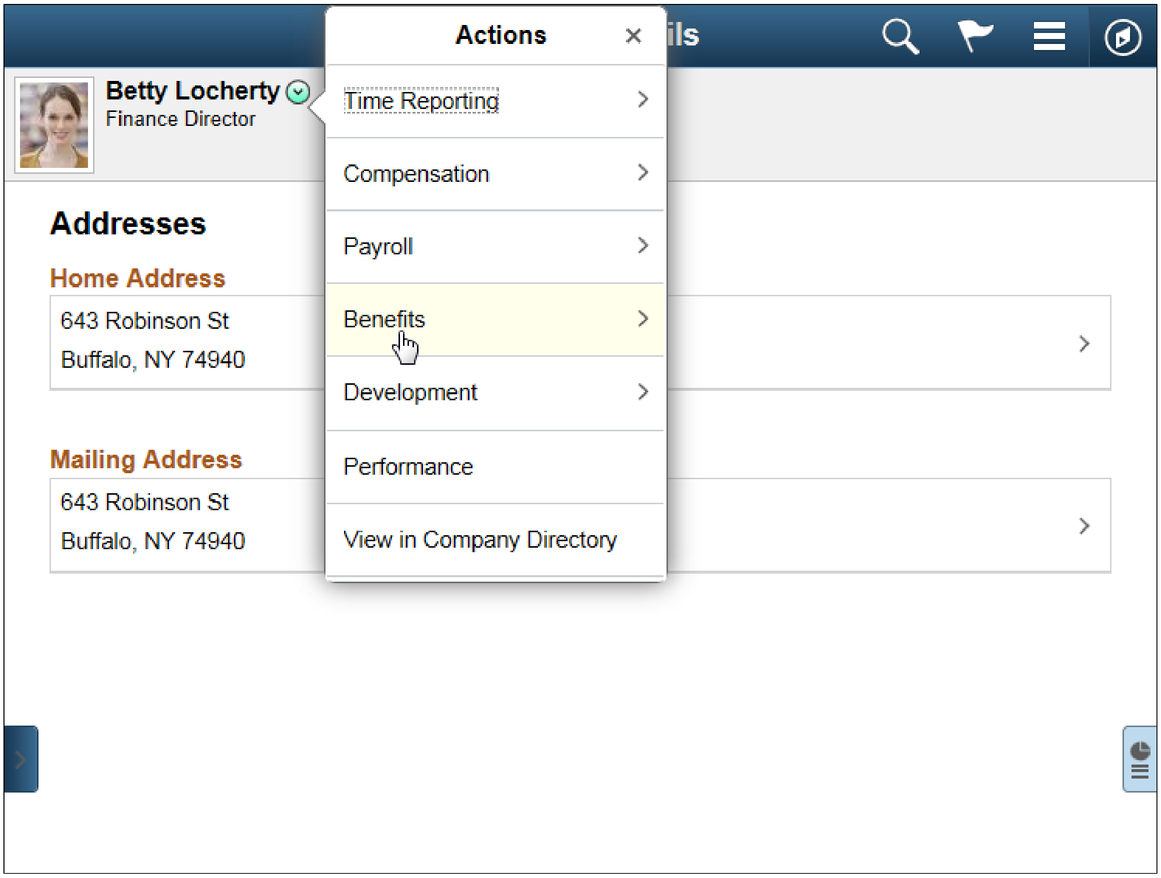
- An Actions menu on a pivot grid chart.

This example shows an Actions menu on a pivot grid chart:



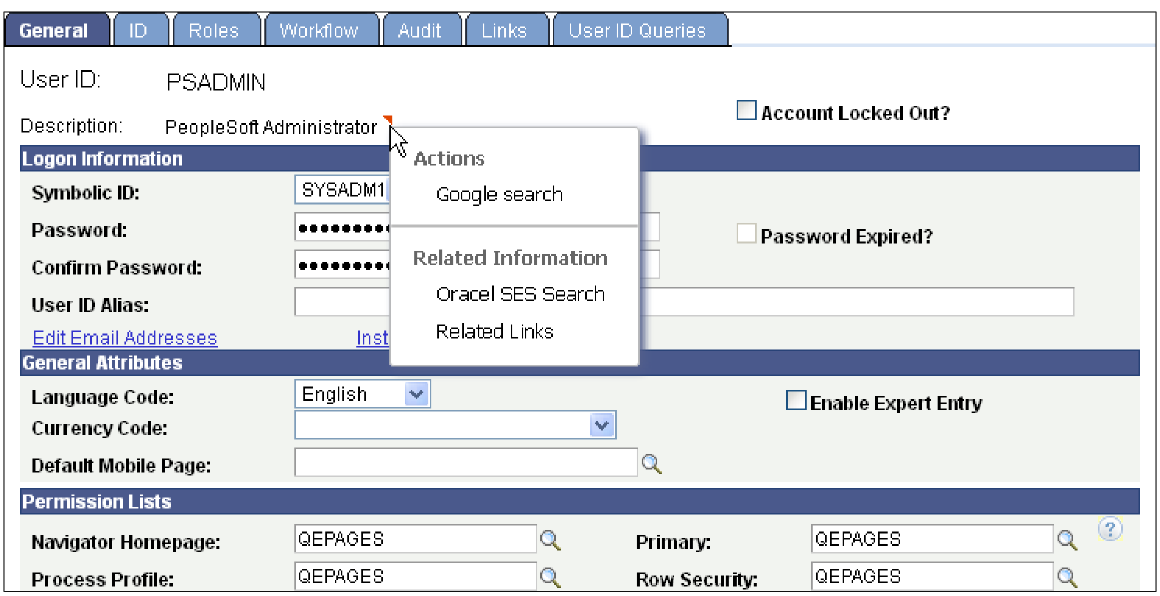
- A page field contextual menu that appears when you click the page field glyph. Page field contextual menus can appear on classic pages or fluid pages.

The following example shows a page field contextual menu on a fluid page:



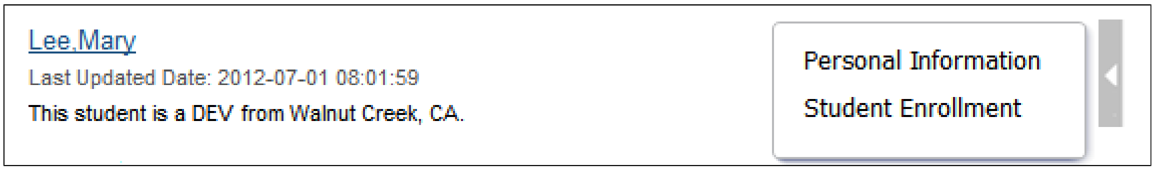
On classic pages, page field contextual menus appear when you click the red page-field glyph. On classic pages only, a page field contextual menu can combine both *related content* and *related actions*.

The following example shows a page field contextual menu with a related action (under the Actions folder) plus related content (under the Related Information folder):



- As pop-up related actions in the search results of a keyword search.

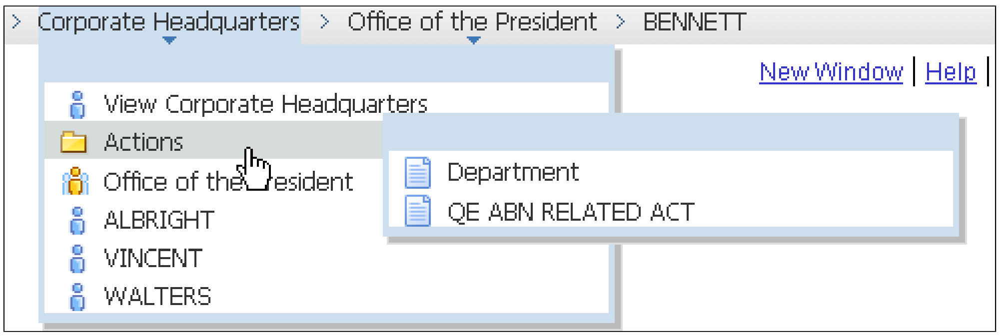
  The following example shows pop-up related actions links in the search results of a keyword search:

- An Actions folder in drop-down SmartNavigation menus.

  Note: While existing SmartNavigation configurations may be available in your PeopleSoft system, the SmartNavigation feature has been desupported.

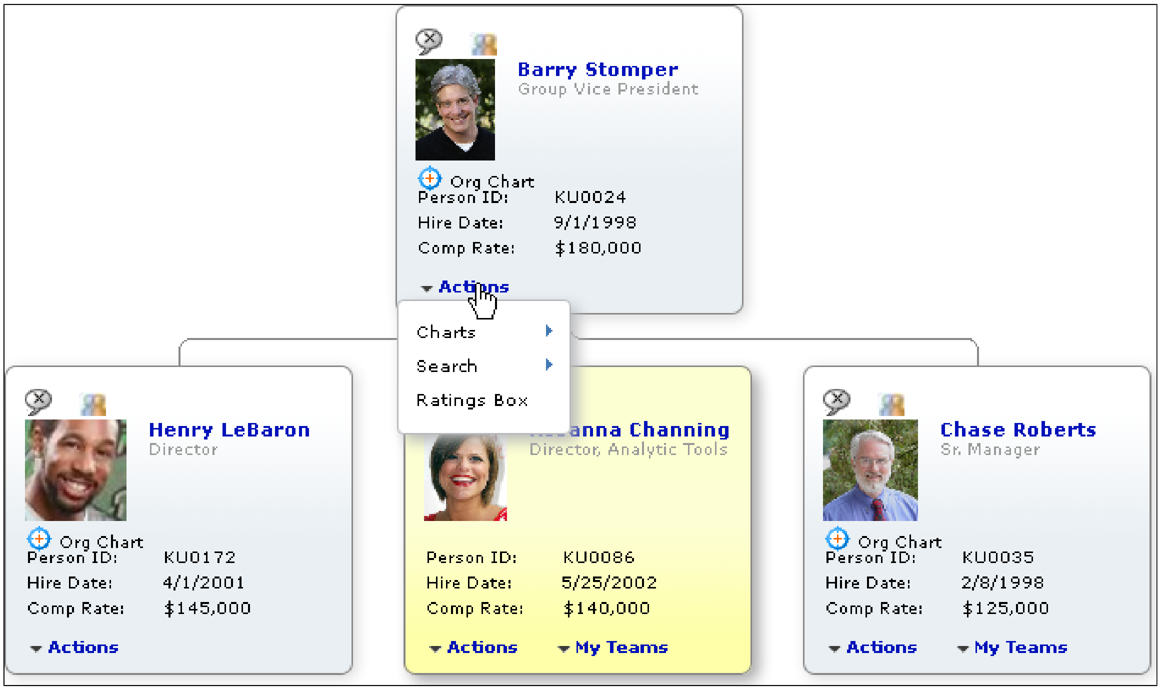
  This example shows the Actions folder in a drop-down SmartNavigation menu:

- A drop-down **Actions** menu on chart nodes of SmartNavigation charts.

  Note: While existing SmartNavigation configurations may be available in your PeopleSoft system, the SmartNavigation feature has been desupported.

This example shows a drop-down Actions menu on a chart node of a SmartNavigation chart:



Related Content from Remote Systems

When you configure single signon between PeopleSoft systems, you can use these remote resources as related content services:

- Embeddable pagelets
- OBIEE pagelets
- PeopleSoft components
- PeopleSoft iScripts
- PeopleSoft queries
- PeopleSoft tiles

Explicit Definition Versus Anonymous Services

Some PeopleSoft content such as components, locally registered pagelets, and tiles can be used directly as *anonymous services* and do not need to be manually defined as a related content service. Content references, homepage pagelets, template pagelets, and tiles that are currently defined in the portal registry can be assigned as anonymous services and as such require no explicit related content service definition.

While the framework enables you to assign most content references as anonymous services without first creating an explicit service definition, you must create an explicit service definition when:

- You want to specify the page within the component that opens when the service is accessed.

- You want to pass parameters to the component other than the key structure.

Fluid Related Content

Related content on fluid pages can be based on either an explicit related content service definition (using the PeopleSoft Tile URL type) or on a content reference definition. As with other anonymous services, the latter does not allow for additional URL parameters while the explicit service definition does. Fluid related content can be configured at:

- The component level as *related information* displayed in the right frame.
- The page level as *embedded related content* within a specific page.

Note: In addition, simplified analytics are configured at the page level.

Fluid Attributes Fields Utilized by Fluid Related Content

Fluid related content relies on a content reference definition regardless of whether the service type is specified as a *Service* (an explicit service definition) or as a *Grouplet* (an anonymous service). The Fluid Attributes page of the content reference definition determines key characteristics of fluid related content. The following fields are used by either embedded related content, related information, or both. Fields marked with an N and all other fields on the Fluid Attributes page are ignored.

Important! Tile definitions also use the fields of the Fluid Attributes page to determine important characteristics of the tile. In some cases, the fields and values used by a tile definition can be compatible with reuse of that definition for embedded related content. In other cases, these fields and values will be incompatible with embedded related content. In such cases, you will need to create a separate content reference definition to use the same component as embedded related content.

Field	***Embedded Related Content***	***Related Information***
Image Name	Y	Y
Interactive	Y[1]	Y
Height	N	Y
Autosize	N	Y
Dynamic Tile Content	Y	Y
URL Type: *PeopleSoft Component*	Y	Y
Node Name	Y	Y

Field	***Embedded Related Content***	***Related Information***
Menu Name	Y	Y
Market	Y	Y
Component Name	Y	Y
Page Name	N	N
Grouplet ID[2]	Y	Y
Additional Parameters	Y	Y
URL Type: *PeopleSoft Generic URL*	Y	Y
Node Name[3]	Y	Y
Portal URL[3]	Y	Y
URL Type: *PeopleSoft Script*	Y	Y
Node Name	Y	Y
Record (Table) Name	Y	Y
Field Name	Y	Y
PeopleCode Event Name	Y	Y
PeopleCode Function Name	Y	Y
Additional Parameters	Y	Y
URL Type: *Non-PeopleSoft URL*	Y	Y
Node Name[3]	Y	Y
Portal URL[3]	Y	Y

Footnotes:

1. While embedded related content can be defined as interactive, you should be aware of the limitations. See the following section, "Limitations with Interactive Grouplets," for more information.

2. The grouplet ID is ignored when the definition is defined to be interactive; the entire component is rendered instead.

3. The node name is optional. If the node is not specified, then the Portal URL field must contain the full URL; otherwise, the Portal URL field can contain a partial URL.

Sizing Fluid Related Content

Note the following for sizing embedded related content:

- Embedded related content ignores the size and alternate size settings on the Fluid Attributes page of the content reference definition.

- Embedded related content is sized by the style specified in the group box definition in Application Designer. If no style is specified, then the embedded related content consumes the default size, which depends on the container or page. See "Working with Embedded Related Content" (Fluid User Interface Developer's Guide) for more information.

Note the following for sizing related content in the Related Information panel:

- The pst_rc-general style is delivered for styling items in the Related Information panel. Create custom styles based on this delivered style rather than modifying the delivered style itself.

- The width of items in the Related Information panel is fixed by the width of the frame itself.

- Items in the Related Information panel have a minimum height of 1 tile unit. When autosize is selected in the content reference definition, there is no maximum height for the item.

Limitations with Interactive Grouplets

If you choose to create an interactive grouplet—for example, by defining interactive embedded related content or programmatically with the SetGroupletType method—you need to be aware of the following limitations:

- Most significantly, interactive grouplets can lose data changes when:

  - The component used is a multi-page component. For example, if a change is made on the first page of the component, the user navigates to another page within the component, and then returns to the first page, that page is refreshed and any changes made there are lost.

  - The interactive grouplet is located within an expandable group box. When the group box is expanded or collapsed, a server trip is triggered, the interactive grouplet is refreshed, and any changes are lost.

  - The interactive grouplet is located within a scroll area or grid that is modifiable (for example, rows can be added, deleted, or hidden). When the scroll area or grid is modified, a server trip is triggered, the interactive grouplet is refreshed, and any changes are lost.

  - A FieldChange event is triggered within the interactive grouplet *and* then a modal window is opened. When the modal window is closed, the page is refreshed and any changes are lost.

- In addition, FieldChange events do not fire within an interactive grouplet in the same manner that they fire elsewhere in PeopleSoft applications. Specifically, a FieldChange event will fire automatically within an interactive grouplet *only if* the user changes focus to another field within the interactive grouplet or the user presses the Enter key within the field with the associated FieldChange PeopleCode program. If the user moves to another field outside the interactive grouplet, no FieldChange event is fired. (The converse is now also true. If a FieldChange PeopleCode program is associated with a field outside of the interactive grouplet, no FieldChange event is fired when the user changes focus to another field inside the interactive grouplet.
- Finally, to prevent data integrity errors in the component buffer during a user's session, you *must not* provide access to the same fluid component as a grouplet or in a "modeless" manner and then again in a "standalone" manner.

Due to the stated limitations, there are a few scenarios that are appropriate for interactive grouplets:

- When the interactive behavior is completely self-contained within the interactive grouplet *and* there are no field changes or interactivity required on the main page containing the interactive grouplet.
- When the interactive behavior is limited to a search that is performed on the value entered in an edit box and the search is invoked by the user pressing the Enter key or clicking a Go-like button.

Related Links

"Limitations on Accessing a Component as a Grouplet and Then in a Standalone Manner" (Fluid User Interface Developer's Guide)

Setting Fluid Attributes for Content References

"SetGroupletType" (PeopleCode API Reference)

The Related Content Development Process

This table lists the steps in the process flow, the roles involved during each step, and provides cross-references to supporting PeopleBooks documentation:

Step	***Role***	***Reference***
Define related content services (including entering help text).	Developer	See Creating and Managing Related Content Service Definitions. See Writing Help Text for Related Content Service Definitions.
Test related content services.	Developer	See Testing Related Content Service Definitions.
Assign related content services to content references or pivot grids.	Subject matter expert	See Managing Related Content for Content References or Managing Related Actions for Pivot Grids.

Step	*Role*	*Reference*
Map service parameters and configure service layout.	Subject matter expert	See Configuring Service Parameters, Filters, and Security. See Configuring Related Content Menus. See Configuring the Related Content Frame.
(Optional) Develop integration of related content—for example, inter-service communication.	Subject matter expert	See Understanding Interwindow Communication.
Access the transaction and test the related content display.	Subject matter expert Application user	See “Accessing Related Content and Actions” (Applications User’s Guide).

Delivered Related Content Services for PeopleSoft Interaction Hub

PeopleTools provides the service definitions for three related content services for PeopleSoft Interaction Hub:

- Related Discussion service
- Related Links service
- Related Tags service

These related content services provide features of Oracle’s PeopleSoft Interaction Hub that can be added to PeopleSoft application pages through the PeopleSoft Related Content Framework. While the service definitions are provided in PeopleTools, an active PeopleSoft Interaction Hub database is required to provide the services and to store the related data.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, “Setting Up Related Content Services.”

Related Discussion Service

Using the Related Discussion service, you can collaborate to share relevant information about a transaction or a transaction instance. Discussion forums in the Related Discussion service provide a platform that groups can use to discuss topics of interest in a manner similar to standalone forums in PeopleSoft Interaction Hub. Discussion forums enable multiple relevant individuals to contribute to the review and resolution of a question. Your can post discussion topics, such as issues, suggestions, or questions, and receive replies and feedback.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, “Working With the Related Discussion Service.”

Related Links Service

The Related Links service enables you to create links to relevant content for the current transaction instance or for all instances in a transaction. You can link to content residing in PeopleSoft Interaction Hub—specifically, blogs, discussions, or collaborative workspaces—add file attachments, or create links to external URLs. In addition, the Related Links service enables you to create a new workspace and to link that workspace to the transaction.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, "Working With the Related Links Service."

Related Tags Service

The Related Tags service enables you to create user-specified tags for the current transaction instance. You can characterize and bookmark these transactional or business objects with terminology of your choosing, providing ease of access as well as collaborative classification of the transaction.

Setup procedures for these services is covered in the product documentation for PeopleSoft Interaction Hub.

See *PeopleSoft Interaction Hub: Using Portal Features*, "Working With the Related Tags Service."

Delivered Related Content Services for Oracle WebCenter

PeopleTools provides these service definitions for use with Oracle WebCenter:

- WebCenter Blogs service
- WebCenter Discussions service
- WebCenter Documents service
- WebCenter Links service
- WebCenter Wikis service

Note: As of PeopleTools 8.54, integration with Oracle WebCenter through the delivered related content services has been desupported. These service definitions may be removed from a future release of PeopleTools.

These related content services provide features of Oracle WebCenter that can be added to PeopleSoft application pages through the PeopleSoft Related Content Framework. While the service definitions are provided in PeopleTools, an active Oracle WebCenter database is required to provide the services and to store the related data.

See Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter.

Note: Oracle recommends that you have a WebCenter that you dedicate solely to your PeopleSoft applications integration.

WebCenter Blogs Service

Blogs provide a useful tool for discussing or evangelizing any type of idea, strategy, or point of view. Blogs may be projected out to a select group of people or to a wider audience. Typically, blogs invite readers to comment on the overall concepts.

The Blogs service enables users to create instance specific or shared WebCenter blogs, which appear for all transaction instances of the transaction page.

WebCenter Discussion Service

The Discussions service lets you expose discussion forums on your application pages, so users can create forums, post questions, and search for answers. For example, customers can share product reviews, or a customer service department can answer questions online. Discussion forums additionally provide the means to preserve and revisit discussions.

The Discussion service enables users to create instance specific or shared WebCenter discussions, which appears for all transaction instances of the transaction page.

WebCenter Documents Service

The Documents service provides features for accessing, adding, and managing folders and files; configuring file and folder properties; and searching file and folder content in the connected content repositories. When you set up a content repository connection to integrate the Documents service, you can add content to the transaction pages of the PeopleSoft application.

The Documents service enables users to create instance specific or shared WebCenter documents, which appears for all transaction instances of the transaction page.

WebCenter Links Service

The Links service provides a way to view, access, and associate related information. For example, in a list of project assignments, you can link to the specifications relevant to each assignment. In a discussion thread about a problem with a particular task, you can link to a document that provides a detailed description of how to perform that task.

The Links service enables users to create instance specific links or shared links, which appears for all transaction instances of the transaction page. When the transaction page or instance also uses other WebCenter related content services, related links to the other WebCenter group spaces appear by default and cannot be removed.

Unlike other WebCenter related content services, the Links service is a PeopleSoft component-based page. When you click the links in the service, the target opens in a new browser window.

WebCenter Wiki Service

A wiki is a collection of useful content or information that users can browse, update, and remove, sometimes without the need for registration. This ease of interaction and the variety of operations make wiki an effective tool for collaborative authoring, where multiple people create written content together.

The wiki service enables users to create instance specific wiki pages or shared WebCenter wiki pages, which appears for all transaction instances of the transaction page.

Related Links

Understanding WebCenter Services

Creating and Managing Related Content Service Definitions

This section provides overviews of service types, service URL parameters, and service security, and discusses how to create, test, and copy related content service definitions.

Understanding Service Types

The PeopleSoft Related Content Framework supports nine explicitly definable service types. Each type is defined by the characteristics of its data source, which determines the information to form the complete URL that locates and requests the actual data.

When you select a service URL type, various service parameters appear for you to configure. The service parameters differ based on the service type that you select. For example, when you select *Embeddable Pagelet,* the **Node** and **Pagelet Name** fields appear as service parameters because the framework must know the name of the node where the pagelet resides and the name of the pagelet to construct the URL that retrieves the pagelet and its data.

In addition to system-defined service parameters, you can enter user-defined parameters, which offers maximum flexibility in configuring the service to meet business requirements.

Definable Service Types

You can explicitly define the following related content service types:

Term	***Definition***
Application Class	Use this service type to define a PeopleCode application class as a related content service. Application class services can: • Access data in the component buffer. • Run any PeopleCode program that is valid in the FieldChange event, for example, transferring the user to another page, opening a modal window, or modifying data in the component buffer. • Be assigned as related actions only. With the exception of the delivered Analytic Service (for use on fluid pages only), application class services cannot be targeted to the Related Content frame because the target frame and the Related Content frame maintain separate sessions and separate buffer structures.

Term	***Definition***
Embeddable Pagelet	Use this service type to define a Pagelet Wizard pagelet that has been published as an *embeddable* pagelet as a related content service.
Generic PeopleSoft URL	Use this service type to define any PeopleSoft object (iScript, component, and so on) as a related content service. Choose to use this option rather than the specific iScript or component type when you are not required to inherit the security specified on that object.
Non-PeopleSoft URL	Use this service type to define any non-PeopleSoft pages, external content, or services as a related content service. **Note:** Use this service type to deploy Oracle User Productivity Kit (UPK) modules and topics as related content. See Creating UPK-Based Related Content Service Definitions.
OBIEE Pagelet	Use this service type to define a published OBIEE pagelet as a related content service. You use Pagelet Wizard to create and publish OBIEE pagelets that use OBIEE reports as the data source. These OBIEE analytics reports are then consumable in the PeopleSoft system.
PeopleSoft Component	Use this service type to define a PeopleSoft component as a related content service. The component can be from the local PeopleSoft system or from a remote PeopleSoft system. The node name specifies whether the service resides in the local or remote PeopleSoft system. The framework also enables you to assign most PeopleSoft components as *anonymous services* without first creating a service definition. However, in certain circumstances, you must create a PeopleSoft component service definition. See Explicit Definition Versus Anonymous Services.
PeopleSoft Query	Use this service type to define a PeopleSoft query as a related content service. The query results can also be filtered based on the query definition and the parameters that it requires.

Term	***Definition***
PeopleSoft Script	Use this service type to define an iScript as a related content service. Use this service type in situations in which an iScript has been used to develop the application functionality based on PeopleSoft data or non-PeopleSoft data, such as stock quotes or weather information.
PeopleSoft Tile	Use this service type to use a tile definition as a related content service. See Understanding Tile Definitions and the Tile Repository

The information that the system uses to construct the nine service types differs for each type and is described in the following sections.

Related Content Services as Action Items

Action items in activity guides can be explicitly defined as related content services. When creating the service definition for an action item, you can ignore the display options and the security options. The basic PeopleTools transaction security model (user-role-permission list) is used to provide user access to the service. The following service types can be used as action items:

- Application class
- Non-PeopleSoft URL
- PeopleSoft component
- PeopleSoft query
- PeopleSoft script

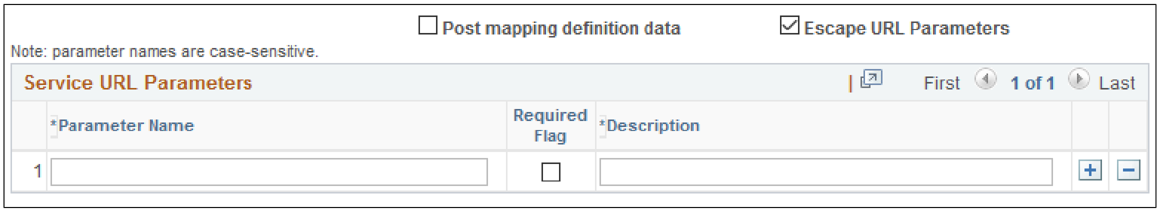
See Creating and Maintaining Action Items.

Understanding Service URL Parameters

A service might need additional parameters to obtain the context for the data that the service provides. You define these parameters in the Service URL Parameter grid. When the subject matter expert assigns the service to a content reference, she or he maps values to the parameters. Then, when the service is invoked from the transaction page, contextual values from fields in the buffer, system variables, or even fixed values that were mapped to the parameters are passed as arguments to the service.

Service URL Parameter Grid

This example illustrates the fields and controls related to the Service URL Parameter grid. You can find definitions for the fields and controls later on this page.



Term	*Definition*
Post mapping definition data	Select to have the mapping definition data, such as the parameter name and its mapping page field or key field name, and the parameter's actual mapped value posted with the service request. This option is deselected by default.
Escape URL Parameters	Select to escape, or encode, the **service URL** parameters and values when the application forms the complete service URL. The default value is selected. Use when either service parameters or parameter values include special characters, such as the single quote (') and the percent sign (%). **Note:** The PeopleSoft system encodes only the portion of the URL that follows the equal (=) sign and excludes the sign itself.
Parameter Name	Enter any additional service parameters for the service. You can map the parameter to a contextual field on an application page when you assign the service to the page. See "Service Parameter Mapping" in Understanding Service Parameter, Filter, and Security Configuration.

Term	***Definition***
Required	Select this check box when the additional parameter is necessary for a service to return a valid output. This check box is deselected by default. If the service is designed with no default value and a blank value is passed to the parameter, an error or blank output could occur, depending on how the service is designed to handle this scenario. To avoid this situation, you can mark one or more additional service parameters as required so that the service will be invoked with a valid value for its service parameters. **Important!** When this required check box is selected, the subject matter expert or administrator must map a field or value to the parameter on the Related Content - Map Fields page or the system does not allow the service assignment to be saved. See "Service Parameter Mapping" in Understanding Service Parameter, Filter, and Security Configuration.
Description	Enter a parameter description that is meaningful for the subject matter expert or administrator who configures the related content assignment and mapping.

Content Reference Attributes as Service URL Parameters

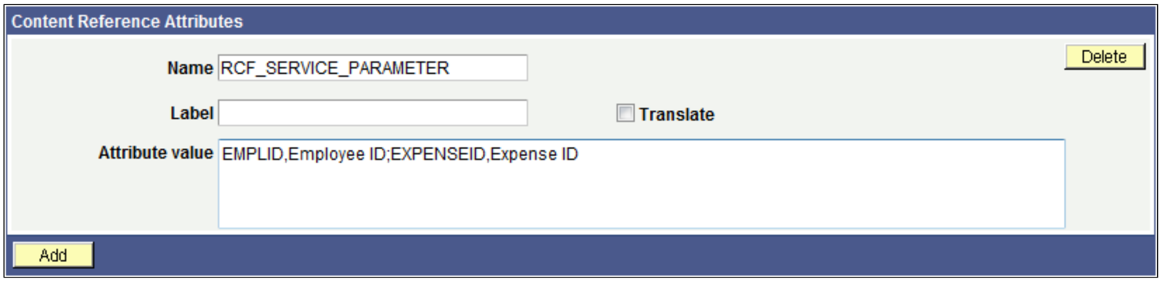
When you assign a content reference as an anonymous service, the system automatically defines service parameters for the components that provide either a valid search record or contextual record. You can manually define the RCF_SERVICE_PARAMETER attribute on the Content Ref Administration page for any content reference, such as for pagelets, iScripts, or for other content references that do not have a valid search record. If this RCF_SERVICE_PARAMETER attribute exists, the system automatically uses the attribute values as the service parameters for the content reference.

Note these points about the RCF_SERVICE_PARAMETER attribute.

- The RCF_SERVICE_PARAMETER attribute overrides the default behavior of component-based URLs. The system will not retrieve the service parameters from a valid search record or contextual record, rather the system overrides any search record parameters with the parameters included in the **Attribute Value** field.
- The attribute value format should follow this pattern:

  NAME1,Label1;NAME2,Label2 with NAME identifying the service parameter and Label identifying the label for the service parameter. For example, *EMPLID,Employee ID;EXPENSEID,Expense ID.*
- The NAME attribute value must *not* include spaces.

The following example shows content reference definition that uses the RCF_SERVICE_PARAMETER attribute:



Related Links

Defining Content Reference Links

Understanding Service Security

PeopleSoft Related Content Framework uses the basic PeopleTools security model, which secures local and remote PeopleSoft transaction pages and the application data that underlies these pages based on the user's assigned permission lists and roles.

For services that provide access to non-PeopleSoft content, the framework enables you to use the security model of the content provider. You should consult the content provider and your security administrator to determine the best course of action.

PeopleSoft Security Authorization Web Service

The PeopleSoft security authorization web service runs on PeopleSoft systems and certain service consumer systems such as Oracle WebCenter. The service examines the basic PeopleTools security on the providing system and can respond to a web service call from another PeopleSoft system.

The PeopleSoft Related Content Framework uses the authorization service in these two ways:

- To show and hide related content links that access PeopleSoft content.
- To grant access to third-party systems, such as Oracle WebCenter that contain PeopleSoft content.

The authorization service can check for basic PeopleTools security at the definition level and at the data, or row, level.

At the definition level, the authorization service response determines whether related content or related actions links to PeopleSoft content appear for the user. This level of authorization is automatically engaged for any PeopleSoft content, and no other configuration is required beyond basic PeopleTools security setup.

However, row-level, or data permission, security is *not* automatic. Use the Authorization Configuration component to configure the authorization service to implement row-level security on remote systems. When the user has the appropriate permissions to view the content, then the authorization service determines whether data, or row-level security has been configured. If any such configurations exist, then the appropriate data is returned when the user clicks the related content or related actions link to access the data. Procedures for configuration of the security authorization service are covered elsewhere.

See “Understanding Developing and Invoking the Security Authorization Service” (Security Administration).

Service Definition Security Options

At design time, you select which security options will be available when the service is assigned and configured for an application page. You can select one or more options to make available to the subject matter expert who assigns the service to a content reference. During the service configuration process, he or she selects the security option to be in effect when the service is accessed from a specific application page; therefore, a different security could be selected for use with a different application page. Before you allow a security option for a service, you should understand the service’s security model.

Important! If you allow a security option for a service definition that isn't applicable or hasn’t actually been implemented by the service itself, unexpected results can occur.

The framework offers four security options.

- Public access.
- Related content provider security.
- Related content consumer security.
- App class required.

Important! The security authorization web service is called for all PeopleSoft content regardless of the option that you select; however, the service is not called for non-PeopleSoft content.

Public Access

The Public Access security option provides access to the service based on the permissions of the public user that you specify in the web profile. The Public Access option is intended for use with services that require few parameters, such as a generic help policy or an internet search engine.

Components that use basic PeopleTools security, which provides authorization and access through permission lists and roles, are not public. If you select the Public Access option for a PeopleSoft component service definition, users must have access to the PeopleSoft component associated with the service.

See “Public Users” in Configuring Portal Security.

See “Setting Page Permissions” (Security Administration).

Related Content Provider Security

Some content providers offer complex and industry-specific security models and mechanisms that are native to the service. The Related Content Provider Security option is intended for use with services that implement security that is specific to their data model or the content that they provide.

As an example, consider the Compensation Details component in Payroll as the producer and the Employee component in an HCM environment as the consumer. If you select Related Content Provider Security, then the provider system (Compensation Details/Payroll) must provide the security.

As another example, the Related Discussion service of PeopleSoft Interaction Hub implements both provider security and consumer security. When provider security is selected, then the security model of the PeopleSoft Interaction Hub discussion forums is used, which means that the forum moderator is responsible for adding members and privileges for access to the forum.

Related Content Consumer Security

PeopleSoft applications use basic PeopleTools security, which provides authorization and access to the many elements of your PeopleSoft application through the assignment of permission lists and roles to users. The Related Content Consumer Security option is intended for use with services that rely on the basic PeopleTools for authorizing access to the service content, for example:

- PeopleSoft component, embeddable pagelet, OBIEE pagelet, PeopleSoft query, and PeopleSoft iScript services that access remote PeopleSoft nodes.
- Oracle WebCenter services that create group spaces linked to PeopleSoft transaction pages. The WebCenter maintains the group space security, but authorization and access are based on the permissions of the PeopleSoft transaction page.

For example, consider the Compensation Details component in Payroll as the producer and the Employee component in HCM as the consumer. If you select Related Content Consumer Security, then the consumer system (Employee/HCM) must provide the security. Therefore, the provider system (Compensation/Payroll) creates an authorization request to the consumer (Employee/HCM), which responds with the authorization details for that user.

App Class Required

Important! Service filters are desupported and this option is retained for backward compatibility only. Do not select this option for new service definitions.

When selected, this option indicates that a service filter is required when this service is assigned to an application page.

Related Links

“Understanding Permission Lists” (Security Administration)
“Understanding Roles” (Security Administration)
“Understanding User Profiles” (Security Administration)
“Understanding Developing and Invoking the Security Authorization Service” (Security Administration)

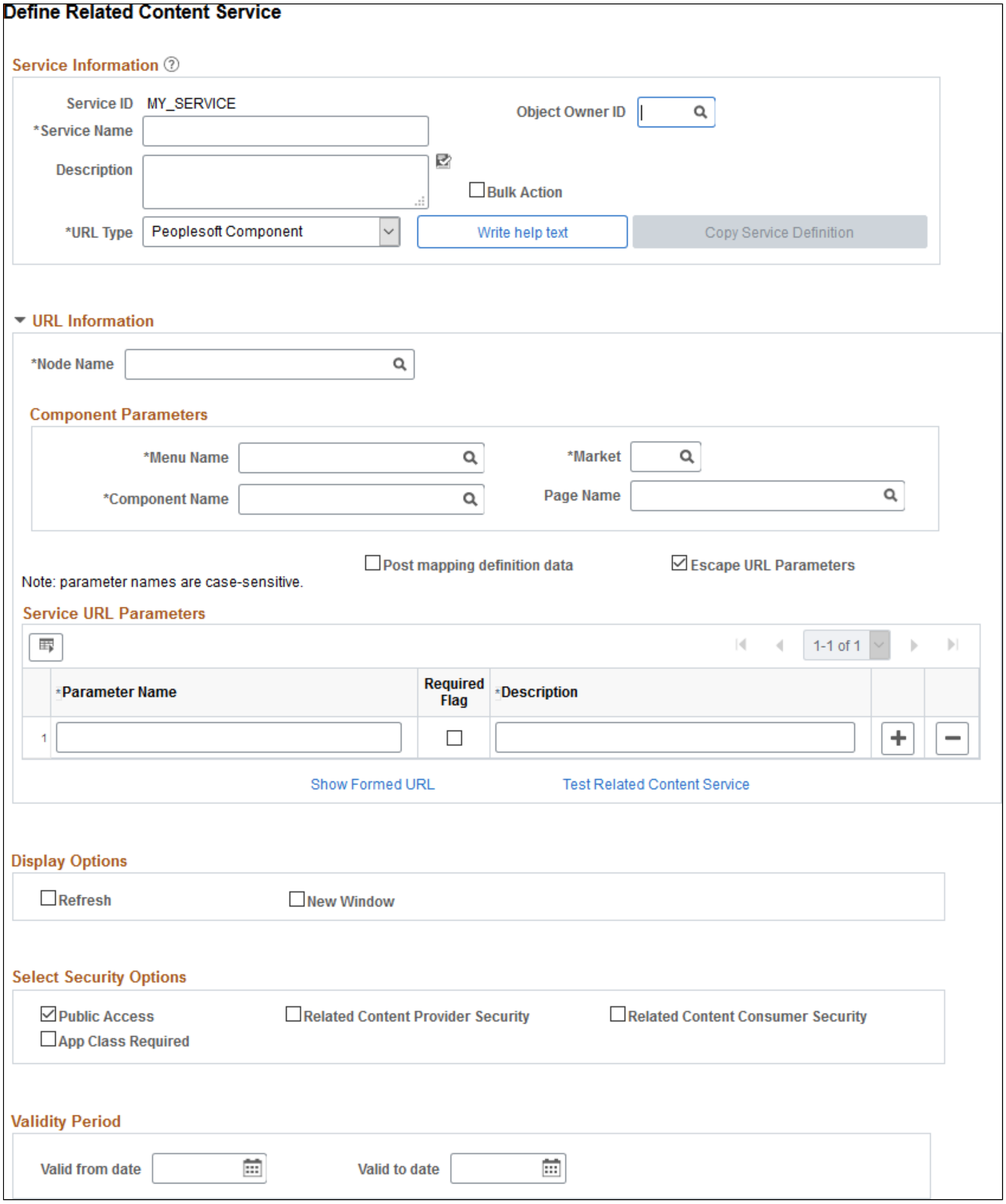
Define Related Content Service Page

Use the Define Related Content Service page to create and manage service definitions.

Navigation:

PeopleTools > **Portal** > **Related Content Services** > **Define Related Content Service**

This example illustrates the fields and controls on the Define Related Content Service page. You can find definitions for the fields and controls later on this page.



This section describes the common elements used in defining related content services. The elements appear in the lists in the order that they appear on the page.

Related Content Service

This table describes fields that are common to all service URL types.

Field or Control	***Description***
Service ID	Specifies the unique identifier of the service definition.
Service Name	Enter a short name for the service, which will be used as the default label for the service when it is assigned as related content.
Object Owner ID	Select which product, feature, or application owns this service definition.
Description	Enter a description for the related content service.
Bulk Action	Select this option to enable the service to execute a related action on multiple rows of data. If the Bulk Action option is not selected, users execute the related action on a single row of data at a time. The Bulk Action option is available for the following URL types: • Application Class • PeopleSoft Script • PeopleSoft Component See Selecting Bulk Action for Related Content Service Definitions.
Template	Select this option to create a master template for template-based services. **Note:** This option is available only when the service URL type is *Application Class*.

Field or Control	***Description***
URL Type	Select from nine service types that build the URL that defines the location and parameters of the service: • *Application Class* • *Embeddable Pagelet* • *Generic PeopleSoft URL* • *Non-PeopleSoft URL* • *OBIEE Pagelet* • *PeopleSoft Component* • *PeopleSoft Query* • *PeopleSoft Script* • *PeopleSoft Tile* See Understanding Service Types.
Write help text	Click to open a help text page where you can create instructions that are specific to assigning this related content service and mapping its service parameters. A subject matter expert can access the help text for assistance when configuring a related content assignment. See Writing Help Text for Related Content Service Definitions.
Copy Service Definition	Click to clone the service definition and its parameters. See Copying Related Content Service Definitions.

URL Information

This table describes URL information fields. These fields are common to several or all service URL types.

Field or Control	***Description***
Node Name	Enter the node name of the PeopleSoft system where the service resides. If the service resides in the same PeopleSoft system, the node name will be either *LOCAL_NODE* or any default local node of the PeopleSoft system.
Populate parameters	Supplies the service parameters in the Service URL parameter grid based on the URL type. This button appears for embeddable pagelet, OBIEE pagelet, and PeopleSoft Query service types only. **Note:** For embeddable and OBIEE pagelets, only context-sensitive parameters are supplied. For query services, only prompt fields are populated. See Step 3: Specifying Pagelet Data Source Parameters.
Show Formed URL	Click to display the entire formed URL with test parameter values in a display-only edit box beneath the **Service URL Parameter** grid. **Note:** Application class URLs cannot be formed by means of this link because the URL is dynamically generated by PeopleCode using data in the buffer.
Formed URL	This field appears only after you click the **Show Formed URL** link. The formed URL appears within a display-only edit box.
Test Related Content Service	Click to test the related content service in a new window. The Test Related Content Service page appears enabling you to enter a parameter value and test the functionality of the service.

Display Options

This table describes display option fields. Display options apply only when the subject matter expert configures the service to appear in the Related Content frame. These fields are common to all service URL types.

Field or Control	Description
Refresh	Select to enable the administrator to include the refresh icon in the related content frame; this icon enables the user to force a server trip to retrieve and refresh the data.
New Window	Select to enable the administrator to include the new window icon in the related content frame; this icon enables the user to send the service to a new browser window. When the service opens a new browser window, the service data in the new window is no longer actively linked to the parent transaction page. If the user searches for a different key value on the parent page, the content of the new window is *not* updated; however, you can continue to view and use it. This is important in situations in which you are viewing content and might want to continue using it after completing work in the transaction page.

Note: Any browser setting that enables the browser to determine whether to open pages in a new window or a tab can affect the expected functionality of the **New Window** property. Configure the *BNEWWIN* navigation personalization option to override any user-specific browser settings so that the browser never uses tabbed browsing for PeopleSoft applications. See "Understanding Navigation Personalizations" (Security Administration).

Select Security Options

This table describes security options.

Field or Control	Description
Public Security	Select this option to provide access to the service based on the permissions of the public user.
Related Content Provider Security	Select this option to use the security implemented by the content provider system.
Related Content Consumer Security	Select this option to use the security implemented by the consuming system, which is the basic PeopleTools transaction security model (user-role-permission list).

Field or Control	*Description*
App Class Required	**Important!** Service filters are desupported and this option is retained for backward compatibility only. Do not select this option for new service definitions. Select this option to indicate that a service filter is required when this service is assigned to an application page.

Before you allow a security option for a service, you should understand the service's security model.

Important! If you allow a security option for a service definition that isn't applicable or hasn't actually been implemented by the service itself, unexpected results can occur.

See Understanding Service Security

Validity Period

Use the validity period to specify whether a service is displayed as related content or a related action at runtime. Regardless, a service can still be configured or displayed within an existing configuration outside of its validity period.

Note: Even if defined, valid from and to dates are ignored for service definitions used in event mapping and activity guides.

Field or Control	*Description*
Valid from date	Select or enter the start date on which the service is available.
Valid to date	Select or enter the end date to which the service is available.

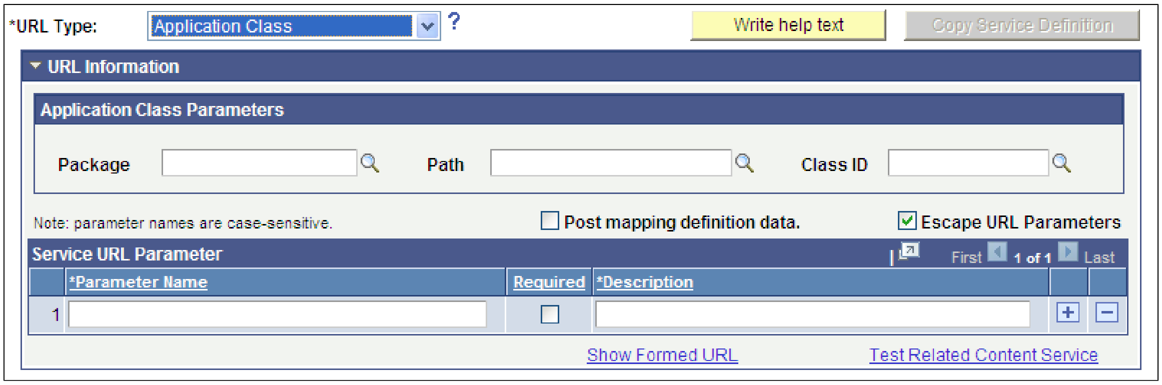
Add to Project

Click the Add to Project link to access the Add To Project page to insert this related content service definition into an ADS project. See "Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide) for more information.

Creating Application Class Service Definitions

On the Define Related Content Service page, select the *Application Class* URL type.

This example illustrates some of the fields and controls on the Define Related Content Service page for an application class. You can find definitions for some of the fields and controls later on this page.



Application Class Parameters

Field or Control	*Description*
Package	Enter the name of the application class package that you want to use to define the related content service.
Path	Enter the names of each subpackage in the application class hierarchy that define the location of the application class that defines the service. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class ID	Enter the name of the application class that defines the related content service. The class must exist in the application package and subpackage that you specify.

These fields are validated when you save the definition; the application does not allow you to save the service if you enter an invalid application class.

Service URL Parameters

Application class service definitions do not require service URL parameters except when you want to map field-level data to the service and display the service in a field-level contextual menu.

Application class URLs cannot be formed by clicking the Show Formed URL link because the URL is dynamically generated by PeopleCode using data in the buffer. For the same reason, application class URLs cannot be tested by clicking the **Test Related Content Service** link.

Simple Application Class PeopleCode Example

An application class that will be used to define a related content service must import and extend the PT_RCF:ServiceInterface class. In addition, it must define a single method named execute.

This is an example of application class PeopleCode that could be used to create an application class service for use as a related action:

```
import PT_RCF:ServiceInterface;


class res_det implements PT_RCF:ServiceInterface
   method execute();
end-class;

Global number &PRL_Schedule_Current_Row;

method execute
   /* Extends/implements PT_RCF:ServiceInterface.execute */

   Evaluate %Component
      When = Component.PGM_RESOURCE_LIST

/*********************************************************************
  Transfer to the Resource detail page.
*********************************************************************/
         If (%Page = Page.PC_PRL) Then
            GetPage(Page.PC_PRL_SCHED).Visible = True;
            &PRL_Schedule_Current_Row = CurrentRowNumber();

          /*AN: Hide the grids if the resource is not a resource class */
            If (GetRow().PROJECT_TEAM.TEAM_MEMBER_CLASS.Value <> "L") Then
               GetRow().GetRowset(Scroll.PROJ_TEAM_SCHED).HideAllRows();
               GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
            Else
               Local Rowset &ipRowSet = GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD);
               If (&ipRowSet.ActiveRowCount > 1 Or
                  (&ipRowSet.ActiveRowCount = 1 And
                     All(&ipRowSet(1).PC_TEAM_IP_SCHD.START_DT.Value))) Then
                  GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD).ShowAllRows();
               Else
                  GetRow().GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
               End-If;
            End-If;

            Local Rowset &arlResDtlVw = GetRow().GetRowset(Scroll.PC_ARL_RESDTLVW);
            &arlResDtlVw.Flush();
            &arlResDtlVw.Select(Record.PC_ARL_RESDTLVW, "where BUSINESS_UNIT = :1 A⇒

ND PROJECT_ID = :2 AND TEAM_MEMBER = :3 and TEAM_MEMBER_CLASS = :4 and TEAM_MEMBER_⇒

IND = :5 ", PROJECT.BUSINESS_UNIT, PROJECT.PROJECT_ID, GetRow().PROJECT_TEAM.TEAM_M⇒

EMBER.Value, GetRow().PROJECT_TEAM.TEAM_MEMBER_CLASS.Value, GetRow().PROJECT_TEAM.T⇒

EAM_MEMBER_IND.Value);

            TransferPage(Page.PC_PRL_SCHED);

         End-If;

            /*AN: Hide the grids if the resource is not a resource class */

         If (&teamRow.PROJECT_TEAM.TEAM_MEMBER_CLASS.Value <> "L") Then
            &teamRow.GetRowset(Scroll.PROJ_TEAM_SCHED).HideAllRows();
            &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
         Else
            Local Rowset &ipSet = &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD);
            If (&ipSet.ActiveRowCount > 1 Or
               (&ipSet.ActiveRowCount = 1 And
                  All(&ipSet(1).PC_TEAM_IP_SCHD.START_DT.Value))) Then
                  &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD).ShowAllRows();
            Else
               &teamRow.GetRowset(Scroll.PC_TEAM_IP_SCHD).HideAllRows();
            End-If;
```



```
            End-If;

               TransferPage(Page.PC_PRL_SCHED);

            End-If;

      End-Evaluate;

end-method;
```

Creating Application Class Service Definitions for Event Mapping

To create and configure application class-based service definitions specifically for event mapping, use the event mapping pages instead.

Related Links

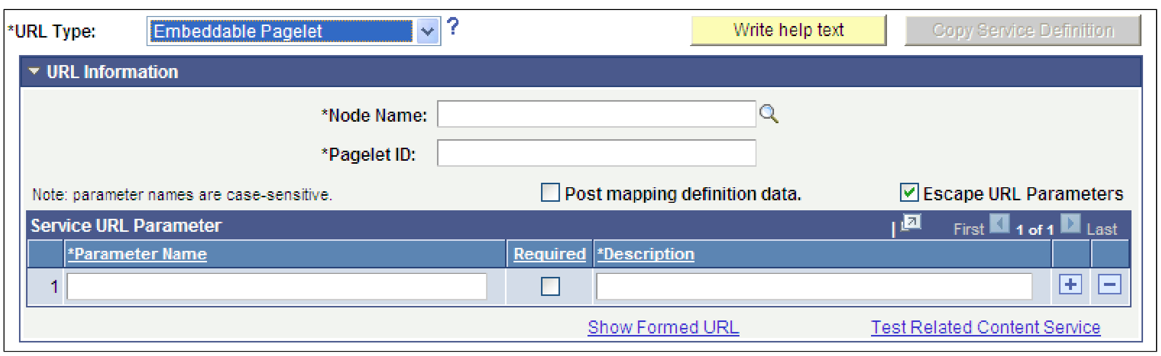
Creating Service Definitions for Event Mapping

Mapping Application Class PeopleCode to Component Events

Creating Embeddable Pagelet Service Definitions

On the Define Related Content Service page, select the *Embeddable Pagelet* URL type.

This example illustrates some of the fields and controls on the Define Related Content Service page for an embeddable pagelet. You can find definitions for some of the fields and controls later on this page.



URL Information

Field or Control	Description
Pagelet ID	Enter the unique identifier of the embeddable pagelet created in Pagelet Wizard. See Step 6: Specifying Pagelet Publishing Options.

Service URL Parameters

Click the **Populate parameters** button to populate the **Service URL Parameter** grid with parameters that you defined as *context sensitive* in step three of the pagelet construction in Pagelet Wizard.

If the pagelet resides in the same system, as soon as you select the node, a lookup appears next to the pagelet ID field. When clicked, the lookup lists all of the available embeddable pagelets in the system.

See Step 3: Specifying Pagelet Data Source Parameters.

Embeddable Pagelet URL Example

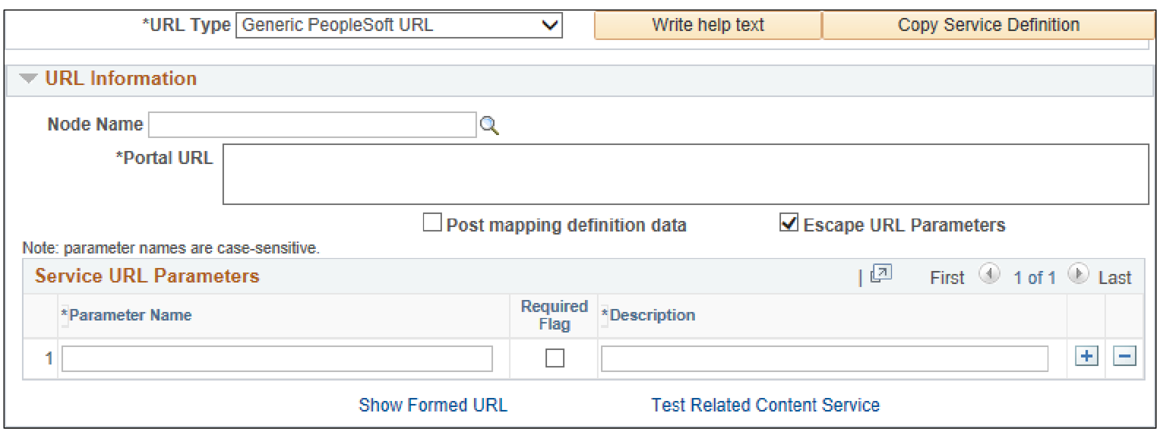
This is an example of a fully formed embeddable pagelet URL:

```
http://myserver.example.com/psc/ps/EMPLOYEE/PT_LOCAL/s/WEBLIB_PTEMBED.ISCRIPT1.Fiel⇒
dFormula.IScript_EmbeddedPageletBuilder?PAGELETID=USER_ROLES&OPRID=PTDMO&.MAXROWS=1⇒
0
```

Creating Generic PeopleSoft URL Service Definitions

On the Define Related Content Service page, select the *Generic PeopleSoft URL* URL type.

This example illustrates the fields and controls on the Define Related Content Service page for a generic PeopleSoft URL. You can find definitions for the fields and controls later on this page.



URL Information

Field or Control	***Description***
Node Name	Optionally, select a defined node name.
Portal URL	Enter the complete or partial URL in this field. If you selected a defined node, then enter the partial URL here. The result is the concatenation of the selected node's URI text and the portal URL that you entered.

Service URL Parameter

Enter the names of all parameters that the component, iScript, or other PeopleSoft definition must have to render the service in the related content service frame.

PeopleSoft Generic URL Example

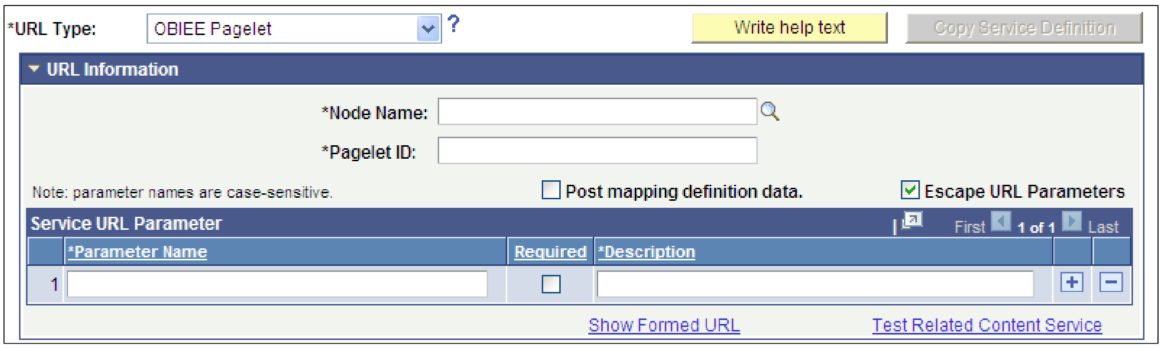
This is an example of a fully formed generic PeopleSoft URL:

```
http://myserver.example.com/psc/ps/EMPLOYEE/PT_LOCAL/c/NUI_FRAMEWORK.PT_AGSTARTPAGE⇒
_NUI.GBL?CONTEXTIDPARAMS=TEMPLATE_ID:NAVCOLL_COMP&scname=PTGP_NAV_COLL
```

Creating OBIEE Pagelet Service Definitions

On the Define Related Content Service page, select the *OBIEE Pagelet* URL type.

This example illustrates some of the fields and controls on the Define Related Content Service page for an OBIEE pagelet. You can find definitions for some of the fields and controls later on this page.



URL Information

Field or Control	***Description***
Pagelet ID	Enter the unique identifier of the OBIEE pagelet created in Pagelet Wizard.

Service URL Parameters

Click the **Populate parameters** button to populate the **Service URL Parameter** grid with parameters that you defined as *context sensitive* in step three of the pagelet construction in Pagelet Wizard.

If the pagelet resides in the same system, then as soon as you select the node, a lookup appears next to the pagelet ID field. When clicked, the lookup lists all of the available OBIEE pagelets in the system.

The **Populate parameters** button supplies the parameters only for the locally available pagelets. If a pagelet resides in a remote PeopleSoft system, you must manually enter the pagelet ID and pagelet parameters.

See Step 3: Specifying Pagelet Data Source Parameters.

OBIEE Pagelet URL Example

This is an example of a fully formed OBIEE pagelet URL:

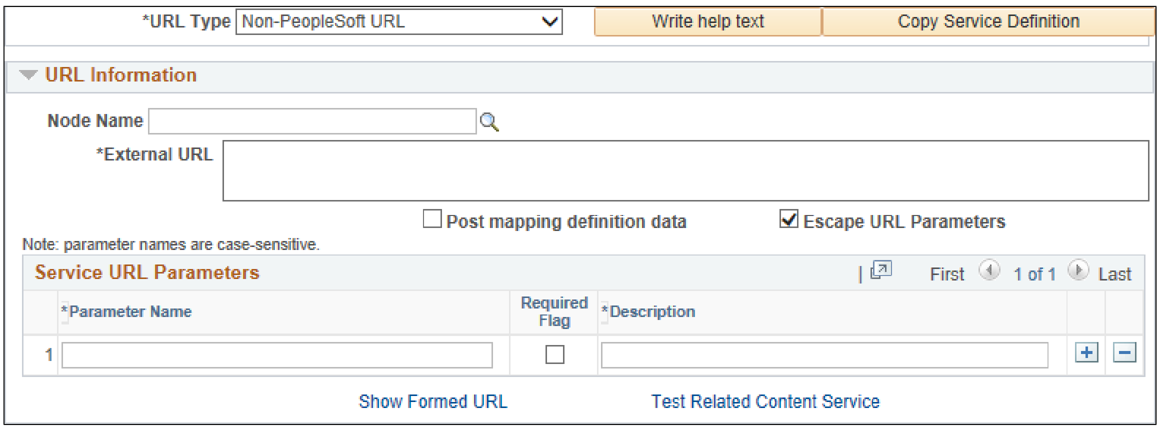
```
http://myserver.example.com/psc/ps/EMPLOYEE/PT_LOCAL/s/WEBLIB_PTEMBED.ISCRIPT1.Fiel⇒
```

```
dFormula.IScript_EmbeddedPageletBuilder?PAGELETID=OBIEE_REPORT&1) Markets.Region=EA⇒
STERN REGION
```

Creating Non-PeopleSoft URL Service Definitions

On the Define Related Content Service page, select the *Non-PeopleSoft URL* URL type.

This example illustrates the fields and controls on the Define Related Content Service page for a non-PeopleSoft URL. You can find definitions for the fields and controls later on this page.



URL Information

Field or Control	*Description*
Node Name	Optionally, select a defined node name.
External URL	Enter a full or partial URL external to the PeopleSoft application. For example: `http://search.oracle.com/search/search`

Service URL Parameters

Enter any parameters that the external service requires. For example: q

Non-PeopleSoft URL Example

This is an example of a fully formed non-PeopleSoft URL:

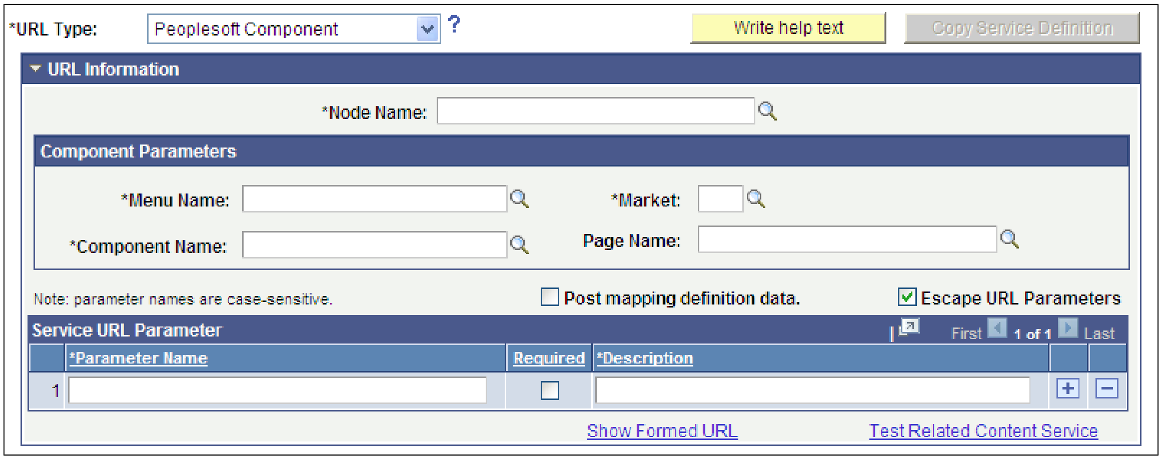
```
http://search.oracle.com/search/search?q=PeopleTools
```

Creating PeopleSoft Component Service Definitions

On the Define Related Content Service page, select the *PeopleSoft Component* URL type.

Note: PeopleSoft components can implicitly act as anonymous services (select *Content Reference* as the service type). You must define this type of service explicitly in certain circumstances. See Explicit Definition Versus Anonymous Services.

This example illustrates some of the fields and controls on the Define Related Content Service page for a PeopleSoft component. You can find definitions for some of the fields and controls later on this page.



Component Parameters

Field or Control	***Description***
Menu Name	Enter the definition name of the PeopleSoft menu on which the component is located as configured in PeopleSoft Application Designer.
Market	Enter the market, such as *GBL,* of the component as defined in PeopleSoft Application Designer.
Component Name	Enter the component definition as defined in PeopleSoft Application Designer.
Page Name	Enter the page definition as defined in PeopleSoft Application Designer. Use this field to open a specific page in the component when it appears in the related content target area.

PeopleSoft Component URL Example

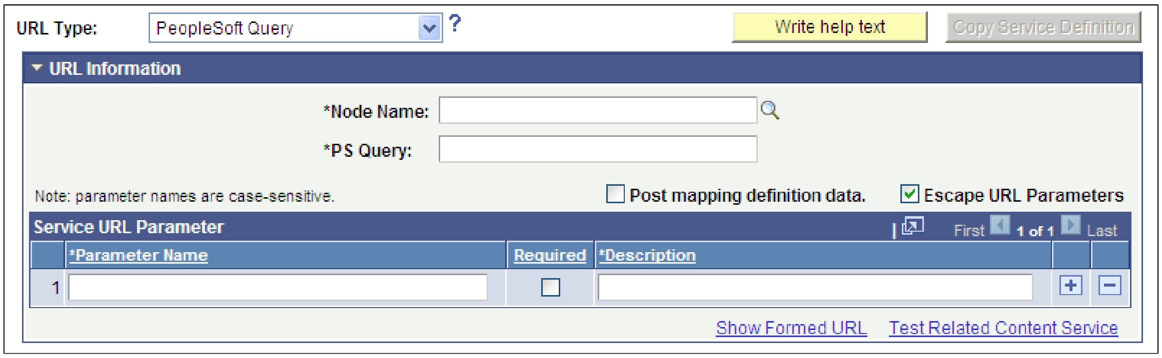
This is an example of a fully formed PeopleSoft component URL:

```
http://myserver.example.com/psp/ps/EMPLOYEE/PT_LOCAL/c/MAINTAIN_SECURITY.USERMAINT.⇒
GBL?OPRID=PTPORTAL
```

Creating PeopleSoft Query Service Definitions

On the Define Related Content Service page, select the *PeopleSoft Query* URL type.

This example illustrates some of the fields and controls on the Define Related Content Service page for a PeopleSoft Query. You can find definitions for some of the fields and controls later on this page.



URL Information

Field or Control	***Description***
PS Query	Enter the query definition name as defined in Query Manager.

Service URL Parameters

Click the **Populate parameters** button to populate the **Service URL Parameter** grid with fields that you defined as *prompts* when creating the query definition.

If the query resides in the same system, as soon as you select the node, a lookup appears next to the **PS Query** field. When clicked, the lookup lists all of the available queries in the system.

The **Populate parameters** button supplies the parameters only for the locally available queries. If a query resides in a remote PeopleSoft system, you must manually enter the query name and query parameters.

See “Defining Prompts” (Query).

PeopleSoft Query URL Example

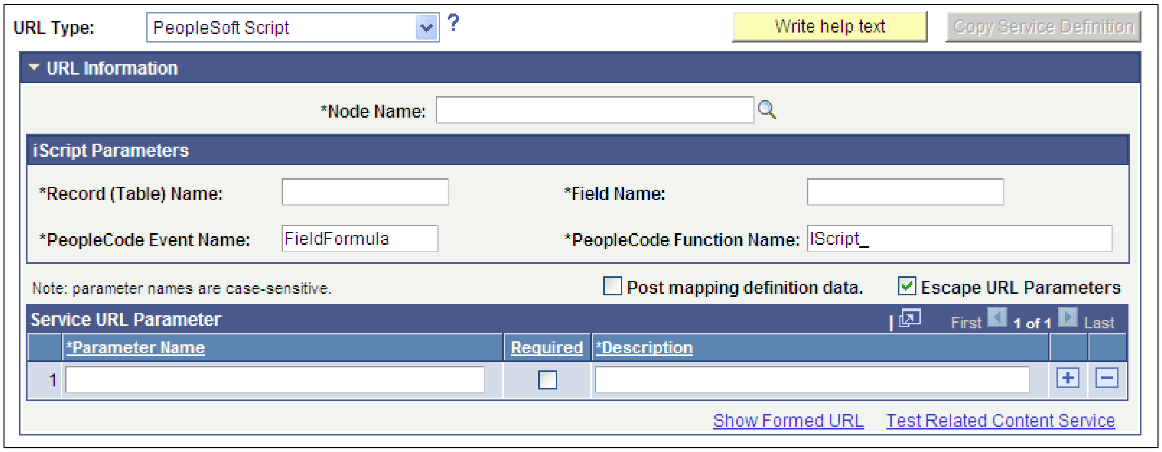
This is an example of a fully formed PeopleSoft query URL:

```
http://myserver.example.com/psc/ps_newwin/EMPLOYEE/PT_LOCAL/q/?ICAction=ICQryNameUR⇒
L=PUBLIC.MESSAGES_FOR_MSGSET&MESSAGE_SET_NBR=240
```

Creating PeopleSoft Script Service Definitions

On the Define Related Content Service page, select the *PeopleSoft Script* URL type.

This example illustrates some of the fields and controls on the Define Related Content Service page for a PeopleSoft Script (iScript). You can find definitions for some of the fields and controls later on this page.



iScript Parameters

Field or Control	*Description*
Record (Table) Name	Enter the name of the WEBLIB_ record where the iScript is stored. See “Web Libraries” (PeopleCode API Reference).
Field Name	Enter the name of the field where the iScript is stored.
PeopleCode Event Name	Enter the name of the PeopleCode event under which the iScript is stored. The default event is FieldFormula.
PeopleCode Function Name	Enter the name of the function that performs the logic of the service.

Service URL Parameter

Enter the names of all parameters that the iScript must have to render the service in the related content service frame.

PeopleSoft Script URL Example

This is an example of a fully formed PeopleSoft script URL:

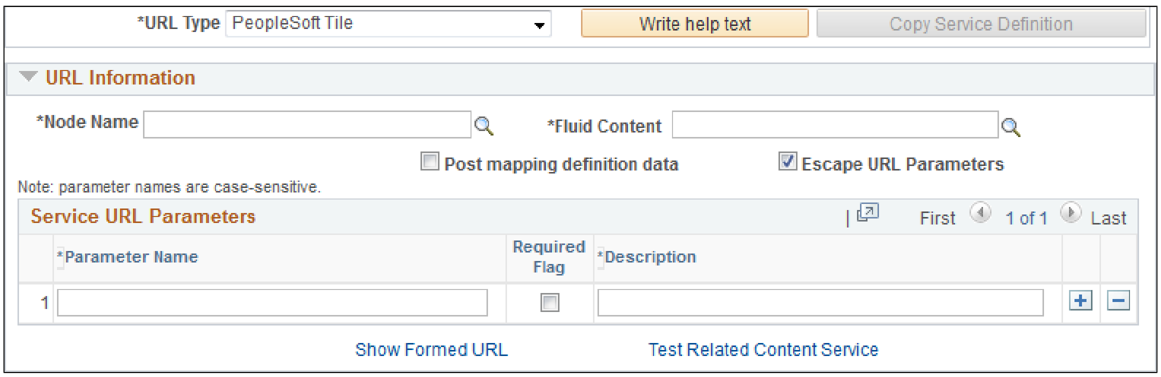
```
http://myserver.example.com/psc/ps/EMPLOYEE/PT_LOCAL/s/WEBLIB_STOCK.ISCRIPT1.FieldF⇒
ormula.IScript_StockQuote?stockquote=ORCL
```

Creating PeopleSoft Tile Service Definitions

On the Define Related Content Service page, select the *PeopleSoft Tile* URL type.

Note: PeopleSoft tiles can implicitly act as anonymous services (select *Grouplet* as the service type). You must define this type of service explicitly in certain circumstances. See Explicit Definition Versus Anonymous Services.

This example illustrates the fields and controls on the Define Related Content Service page for a PeopleSoft tile. You can find definitions for the fields and controls later on this page.



URL Information

Field or Control	***Description***
Fluid Content	Use the Tile Search page to select the tile definition on which this service is based. See Tile Search Page for more information.

PeopleSoft Tile URL Example

The fully formed URL is not available for a PeopleSoft tile.

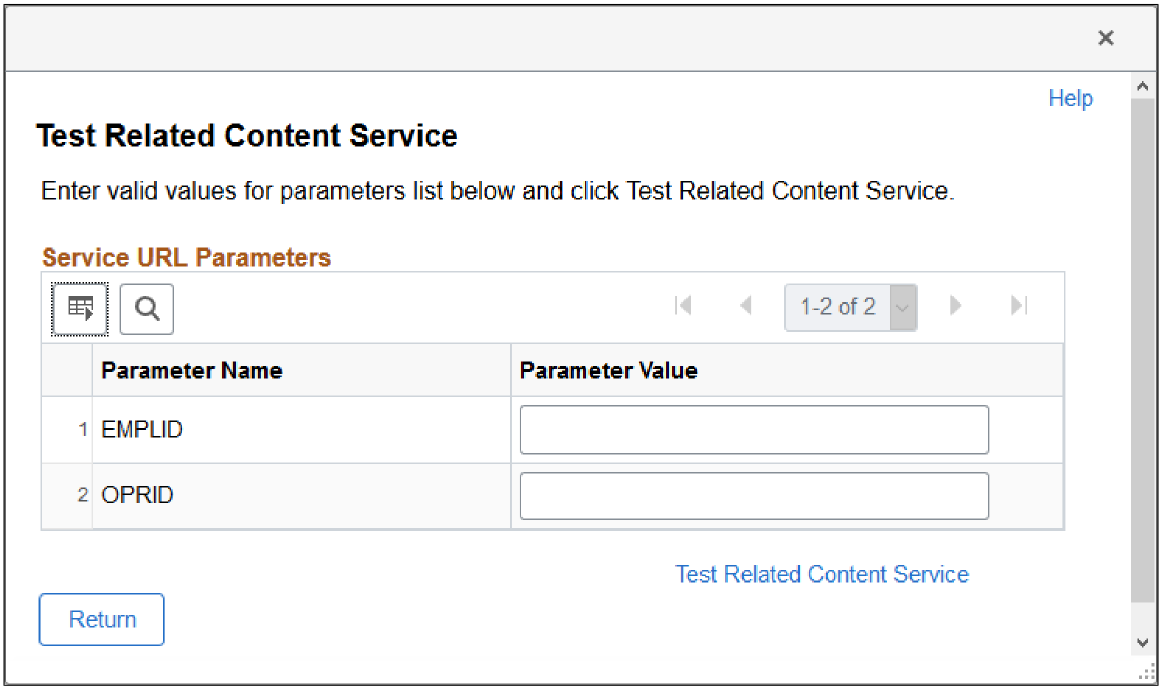
Testing Related Content Service Definitions

Use the Test Related Content Service page to enter parameter values and then test the related content service.

Navigation:

On the Define Related Content Service page, click the **Test Related Content Service** link.

This example illustrates the fields and controls on the Test Related Content Service page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Parameter Name	Displays the parameters that you enter on the Define Related Content Service page. This field is display-only.
Parameter Value	Enter a test value for the parameter.
Test Related Content Service	Click to test the related content service with the parameter values that you enter.
Return	Click to return to the Define Related Content Service page after testing the service.

Testing Related Content Service Parameter Values

To test parameter values:

1. Enter the value in the **Parameter Value** field.
2. Click the **Test Related Content Service** link.

   The service appears in a separate browser window using the specified parameter values.
3. Click the **Return** button to return to the Define Related Content Service page.

Selecting Bulk Action for Related Content Service Definitions

When you define certain types of related content services, you can select whether they should be performed as a *bulk action* on multiple rows of data when configured as a related action.

The bulk action option is available for the following service types:

- Application class
- PeopleSoft script
- PeopleSoft component

When you select the Bulk Action check box, only the supported service types are displayed in the URL Type drop down.

Note: If you select a service type that is not supported by the Bulk Action option, and then select the Bulk Action check box, a message displays that the selected service type is not supported. When you click OK, the Bulk Action check box is no longer selected.

Component-Level Related Actions

At the component level, you can associate a bulk action with a service. The Look Up Service ID page and the Configure Service page reflect the selection of a bulk action.

Page-Level Related Actions

At the page level, a bulk action is available for a service that is linked to the related action widget. A bulk action is not available for a glyph. The Related Action Look Up page and the Configure Service page reflect the selection of bulk action.

Note: If a bulk action is mapped to a glyph field, the related action is applicable for a single row of data; not multiple rows.

Limitations

Bulk action functionality is not supported:

- In the Related Content frame.
- For the following service types:
  - Embeddable pagelet
  - Generic PeopleSoft URL
  - Non-PeopleSoft URL
  - OBIEE pagelet
  - PeopleSoft query
  - PeopleSoft tile

Bulk Action Availability for Authorized/Unauthorized Rows

The bulk action functionality is available or not available in the following scenarios:

Scenario	***Bulk Action Availability***
User is authorized for all rows of a transaction.	Yes
User is not authorized for any of the rows of a transaction.	No
User is authorized for partial number of rows of a transaction.	Yes Only the authorized rows are processed and a message displays the number of processed rows.

Writing Help Text for Related Content Service Definitions

When you define related content services, you should document the service and how the service parameters should be mapped to transaction content, especially if your service uses multiple, complex service parameters. The components and pages can include numerous fields, some of which are named similarly, which can make mapping the service parameters difficult.

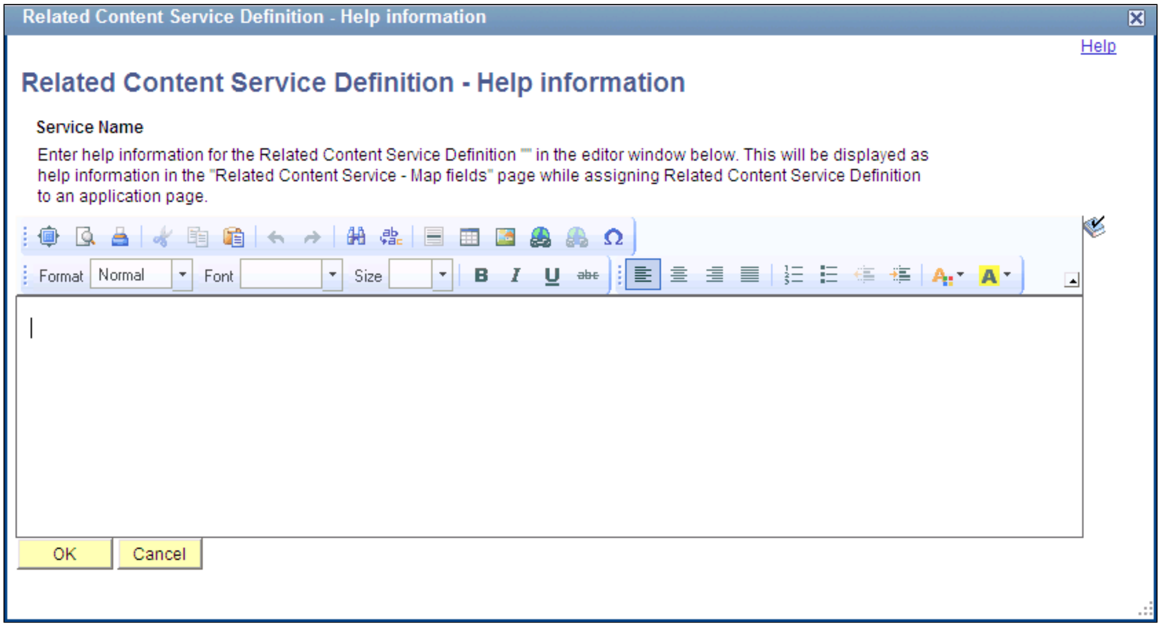
Use the help text to clearly communicate the values that must be assigned to the service parameters. The text that you enter is accessible to the subject matter expert when she clicks the **Read help text** link on the Related Content Service - Configure Service page.

Important! Use explicit, concise, clear, understandable language in your instructions.

Navigation:

On the Define Related Content Service page, click the **Write help text** button.

This example illustrates the fields and controls on the Related Content Service Definition - Help information page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Service Name	Displays the name of the related content service. This field is display-only.
OK	Click to save the text and return to the Related Content Service Definition page. **Note:** You must save the service definition to save the help text.
Cancel	Click to discard any changes and return to the Related Content Service Definition page.

To write help text for a related content service definition:

1. Enter help information for the related content service definition in the editor window.
2. Click the **OK** button.

   This returns you to the Related Content Service Definition page.
3. Save the definition.

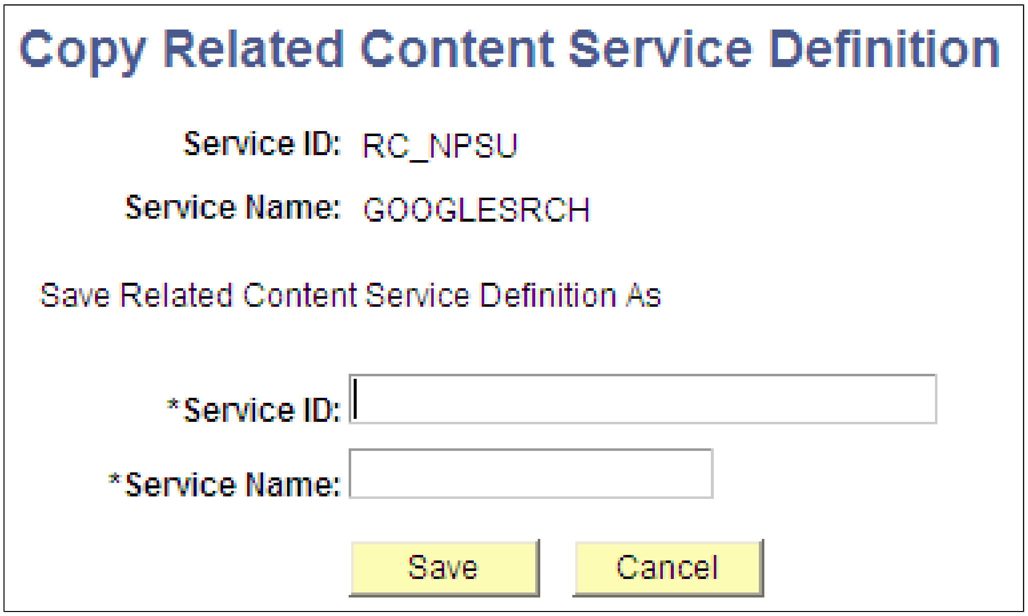
Copying Related Content Service Definitions

Use the Copy Related Content Service Definition page to save the current service definition as a new service definition.

Navigation:

On the Define Related Content Service page, click the **Copy Service Definition** button.

This example illustrates the fields and controls on the Copy Related Content Service Definition page. You can find definitions for the fields and controls later on this page.



To copy a related content service definition:

1. Enter a new value in the **Service ID** field.
2. Enter a new value in the **Service Name** field.
3. Click the **Save** button.

   A message appears indicating a successful save.
4. Click the **OK** button to dismiss the message box.
5. Click the **Go to Related Content Service Definition** link that appears to access the new related content service definition or click the **Cancel** button to close the dialog box.

Managing Related Content Configurations and Data

This section provides an overview of related content assignment, and discusses how to manage related content for content references and pivot grids.

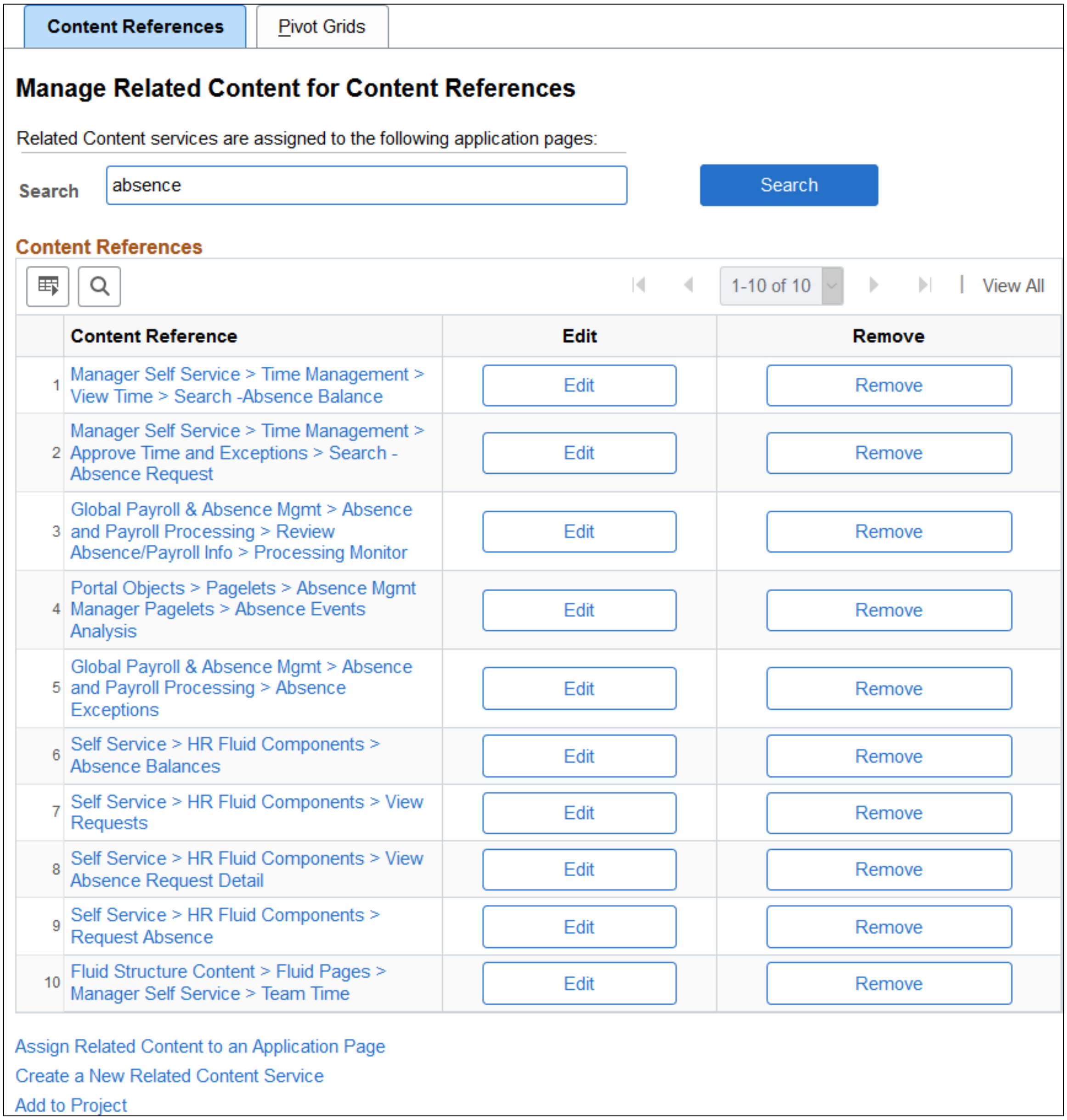
Managing Related Content for Content References

Use the Manage Related Content for Content References page to manage related content and related action configurations for content references.

Navigation:

PeopleTools > **Portal** > **Related Content Services** > **Manage Related Content Service**

This example illustrates the fields and controls on the Manage Related Content for Content References page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Search	Enter a search term or phrase and click the **Search** button to filter the list of content references using the search terms. Search terms are case insensitive and results are returned for partial word matches. For example, both *dep* and *dept* will match the Department (DEPT_TBL.GBL) content reference. When you click the **Search** button, a database search is performed on the following values: • Content reference name • Content reference label • Values stored in the KEYWORD content reference attribute. See “Content Reference Attributes” in Defining Content References.
Content Reference	Click a content reference link to access the actual transaction page, which allows you to test any configured related content.
Edit	Click an **Edit** button to edit an existing related content configuration. See Assigning Related Content to Classic Components and Pages, Assigning Related Content to Fluid Components and Pages, or Assigning Related Actions to Classic Pages, Fluid Pages, and Pivot Grids.
Remove	Click a **Remove** button to remove an existing related content configuration.
Assign Related Content to an Application Page	Click this link to access the Select a Content Reference page to create a new related content assignment.
Create a New Related Content Service	Click the link to access the Define Related Content Service page. See Creating and Managing Related Content Service Definitions.
Add to Project	Click the Add to Project link to access the Add To Project page to insert one or more related content configurations by component into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) for more information.

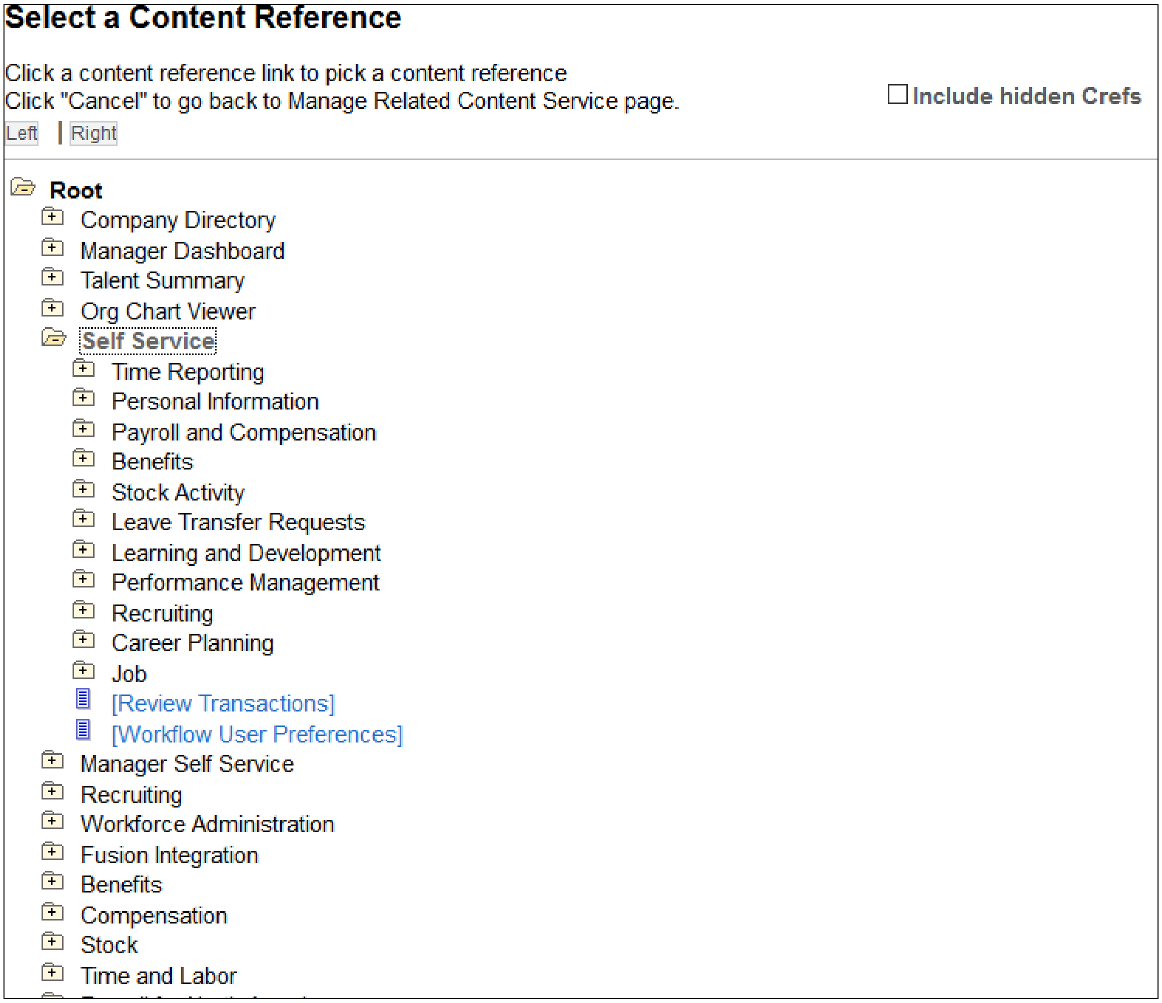
Selecting an Application Page (Content Reference)

Use the Select a Content Reference page to select a content reference for configuration.

Navigation:

On the Manage Related Content for Content References page, click the **Assign Related Content to an Application Page** link.

This example illustrates the fields and controls on the Select a Content Reference page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Include hidden Crefs	Select to make hidden content references available for assignment. **Note:** The tree shows both PeopleSoft component URLs and PeopleSoft component-based generic URLs.
	Click to expand the folder and display child folders and links.
	Click to collapse the folder and hide all children.

Field or Control	Description
[link]	Click a content reference link to select that content reference to which related content will be assigned.

After selecting a content reference, continue by assigning and configuring related content or related actions.

Related Links

Assigning Related Content to Classic Components and Pages
Assigning Related Content to Fluid Components and Pages
Assigning Related Actions to Classic Pages, Fluid Pages, and Pivot Grids

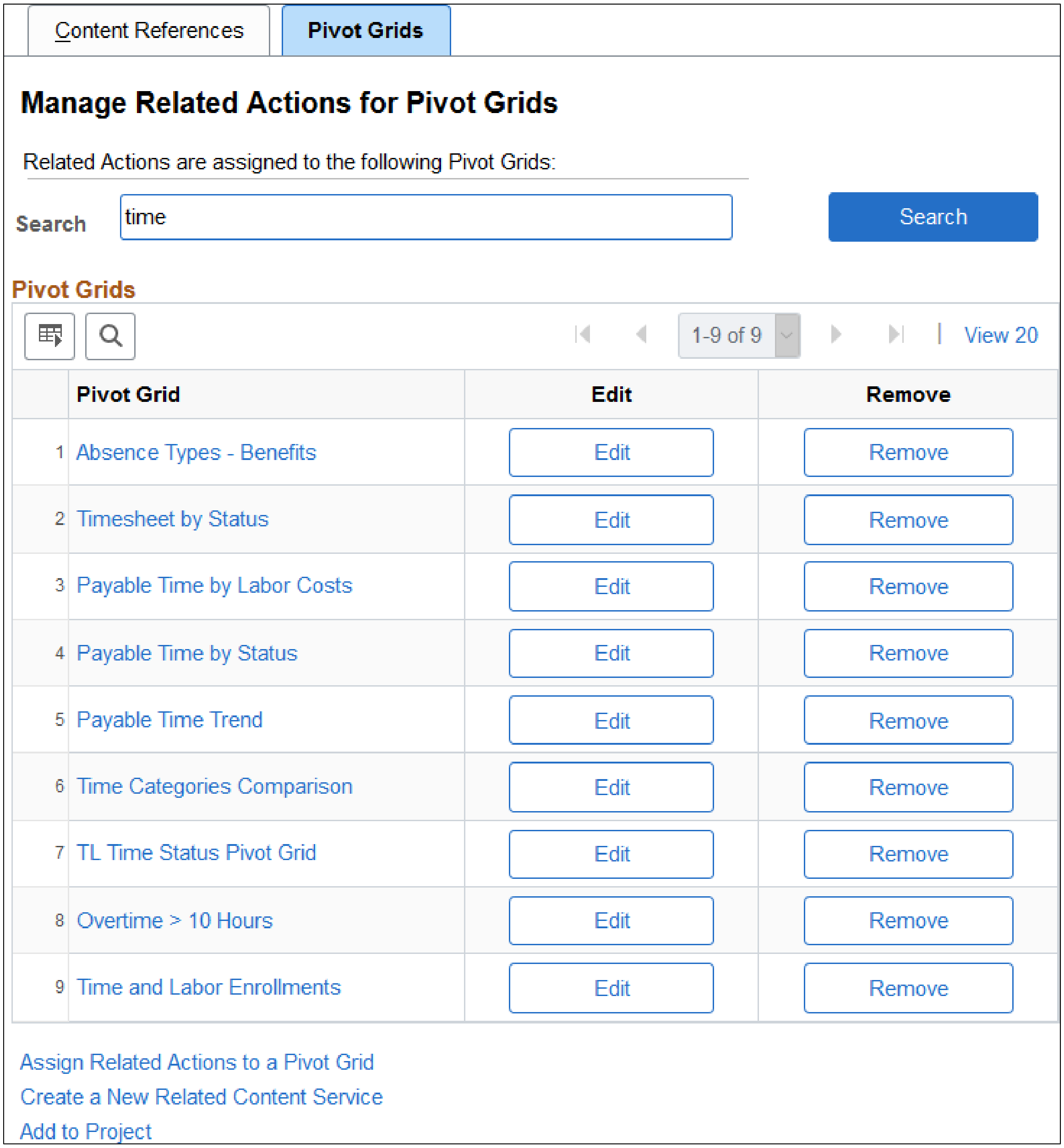
Managing Related Actions for Pivot Grids

Use the Manage Related Actions for Pivot Grids to manage related action configurations for pivot grids.

Navigation:

1. **PeopleTools** > **Portal** > **Related Content Services** > **Manage Related Content Service**
2. Select the **Pivot Grids** tab.

This example illustrates the fields and controls on the Manage Related Actions for Pivot Grids page. You can find definitions for the fields and controls later on this page.



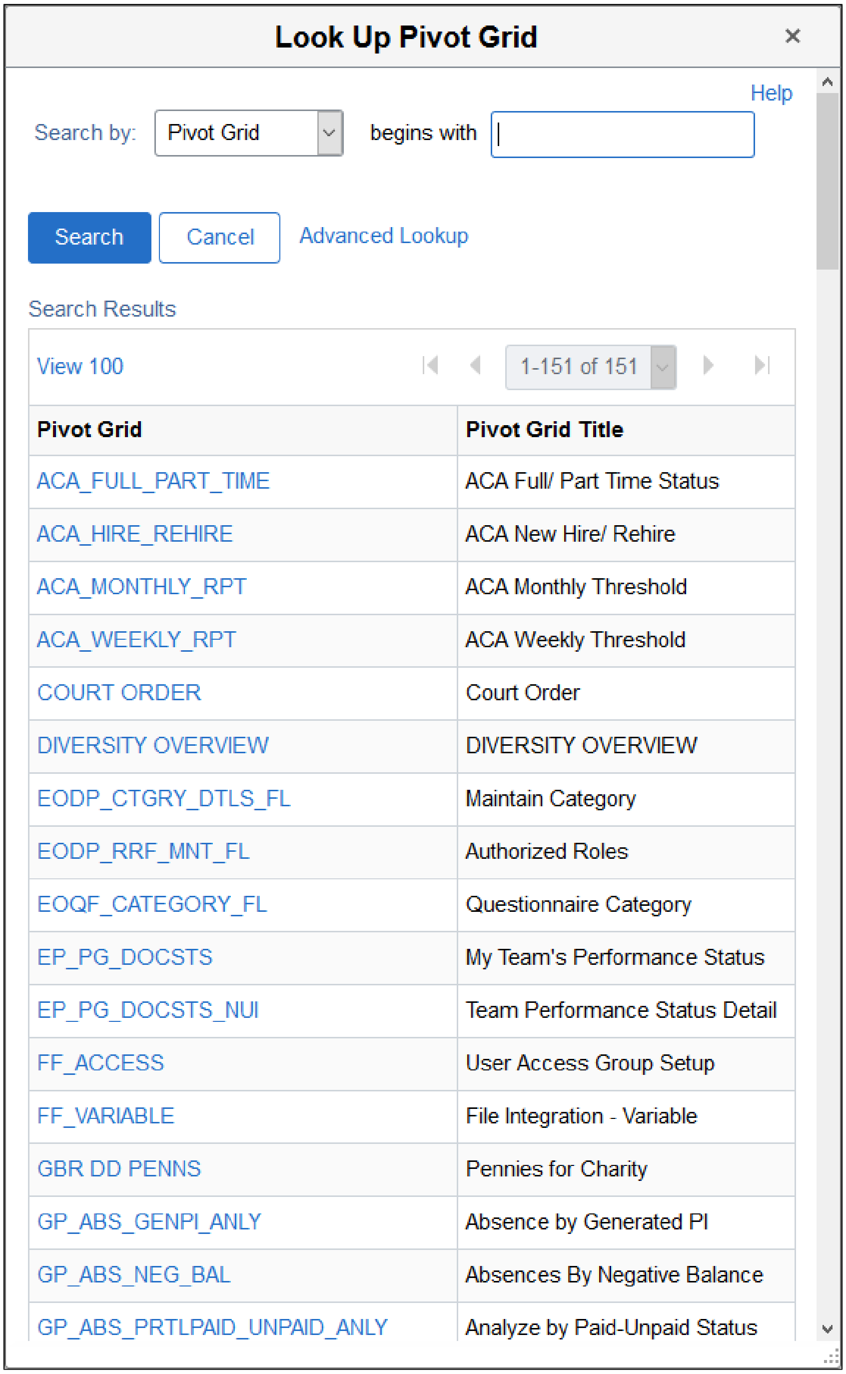
Field or Control	*Description*
Search	Enter a search term or phrase and click the **Search** button to filter the list of pivot grids using the search terms. Search terms are case insensitive and results are returned for partial word matches. For example, both *gri* and *pv_gr* will match the Sales Grid (PV_GRID) pivot grid. When you click the **Search** button, a database search is performed on the following values: • Pivot grid name • Pivot grid title
Pivot Grid	Click a pivot grid link to open this pivot grid in the Pivot Grid Viewer, which allows you to test any configured related actions.
Edit	Click an **Edit** button to edit an existing related actions configuration. See Assigning Related Actions to Classic Pages, Fluid Pages, and Pivot Grids.
Remove	Click a **Remove** button to remove an existing related actions configuration.
Assign Related Actions to a Pivot Grid	Click this link to select a pivot grid for a new related actions assignment.
Create a New Related Content Service	Click the link to access the Define Related Content Service page. See Creating and Managing Related Content Service Definitions.
Add to Project	Click the Add to Project link to access the Add To Project page to insert this related content service definition into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) for more information.

Selecting a Pivot Grid for a New Configuration

To select a pivot grid for a new related actions assignment:

1. Click the **Assign Related Actions to a Pivot Grid** link on the Manage Related Actions for Pivot Grids page.
2. On the Add a New Value tab, click the Look up Pivot Grid button.
3. Select a pivot grid on the Look Up Pivot Grid page:

Search and select a pivot grid:



4. Click the **Add** button to begin assigning and configuring related actions to the selected pivot grid.

5. See Assigning Related Actions to Classic Pages, Fluid Pages, and Pivot Grids.

Upgrading or Copying Related Content Data Using ADS Definitions

Using ADS definitions for upgrading or migrating Related Content Framework data, you can preserve affected configurations and personalizations on a target database. The ADS relationship framework and ADS merge framework preserve configurations and personalizations on a target database.

You can use Data Migration Workbench to migrate related content data. Data Migration Workbench identifies configurable changes that are preserved during upgrade and non-reconcilable changes that are not preserved during upgrade. The configurable and non-reconcilable changes are associated with database tables and its fields that contain related content service definitions and related content services data.

Oracle recommends that you:

- Do not delete any PeopleSoft delivered objects.
- Restrict your configurations to configurable fields specified for each table because these configurations are preserved.
- Avoid modification to any field that is in the non-reconcilable section because any modification to these fields is considered to be a customization and will not be restored.

The following table lists the configurable and non-reconcilable changes associated with related content service definitions. The database tables and their configurable and non-reconcilable fields associated with related content services are described later.

Configurable Changes	***Non-reconcilable Changes***
Modification to service name of a definition.	Changing the URL type of a service definition
Customer modifies the node name to match the environment	Deleting a required parameter of a service definition
Modification to service definition’s description and help text	Adding a required parameter of a service definition
Modification to the display options: refresh and/or new window	Modifying the following options: • URL escape parameters • Post mapping definition • Security options
Modifying the parameter’s description	Modifying a parameter name

The following table lists the configurable and non-reconcilable changes associated with related content and related action configuration.

Configurable Changes	***Non-reconcilable Changes***
Modification of ‘layout options’ (open in expanded view)	Deleting a CREF association.
Modification of the ‘related content location’ of related content frame	Deleting an existing configuration from a CREF.

Configurable Changes	***Non-reconcilable Changes***
Modification of the service label of related content/action	Mapping type of parameter of a configuration (key field, fixed etc).
Making/ removing the default service of an related content	Changing the page field menu option.
Modifying the service target of an related action	Deleting a PeopleSoft-delivered folder.
Enabling or disabling a related content / action	
Modifying the values of service filter (package, path, class ID) app package	
Modifying the options of the related content frame to be minimized when it is displayed.	
Modifying the ‘display in field menu’ option for related action	
Modifying ‘refresh service on change’ option	
Modifying ‘is value required?’ option	
Reordering the sequence of folders.	

The following sections list the database tables and its configurable and non-reconcilable fields.

PSPTCSSRVDEFN (Service Definition)

This table holds the service definition data.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	***Comment***
PTCS_SERVICENAME	Service Name
DESCR254	Description
MSGNODENAME	Node Name
PTCS_SRVHELPTEXT	Help Text

Field	**Comment**
USEEDIT	Display Options
PORTAL_URL_ADM	URL (for URL Type == Non PeopleSoft URL)

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

Field	**Comment**
PTCS_SERVICEURLTYP	URL Type
PTCS_ESCAPEPARAM	Escape URL parameters option
PTCS_POST_MAPDATA	Post Mapping Definition Data option
PTCS_SECUSEEDIT	Security options

PSPTCS_PARAMS (Parameters table)

This is the child table of the service definition table, which holds the parameters details defined for the service definition. This is a metadata table and hence the scope for secondary merge is very limited. PeopleSoft recommends that users do not delete any of the delivered parameters, update the parameter name and modify the required flag options.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	**Comment**
PTCS_DESCR128	Description
SEQNUM	Parameter’s order of definition

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

Field	**Comment**
PTCS_PARAMETERNAME	Parameter Name

Field	Comment
REQUIRED_FLG	Required Flag

PSPTCS_SRVCFG

This table stores the top-level related content metadata for all the content references that have related content associated with them. It also stores the metadata of how the related content frame is displayed (Right Frame or Bottom Frame).

Any changes, other than the ones specified in the following table are non-reconcilable.

Field	Comment
PTCS_EXPANDFRAME	Open in Expanded View Options
PTCS_RCFRAMELOC	Related Content Frame Location

PSPTCSSRVCONF (Service Configurations)

This is the actual table that stores the relationship between instances of service definitions and content references.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	Comment
IS_DEFAULT	Default service
PTCS_INSTANCELBL	Service Label
PTCS_URLTARGET	Service Target
PACKAGEROOT	Service Filter – Package
QUALIFYPATH	Service Filter – Path
APPCLASSID	Service Filter – Class ID
PTCS_INITMIN	RC Frame to be open minimized
PTCS_ENABLE	Enable/Disable Related Content/Actions

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

Field	***Comment***
SEQNUM	Customer cannot recreate this configuration as the instance ID will be different.
PTCS_MAPFIELDTYPE	Mapping Type of the parameter
PTCS_FIELDMENUOPT	Applicable for Related Actions as this is the field to which the glyph is associated.

PSPTCS_MAPFLDS

This is a child table of PSPTCSSRVCONF and stores the field mappings.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	***Comment***
PTCS_FIELDMENUOPT	Display In Field Menu option. Applicable for related Actions
PTCSREFRESHSERVICE	Refresh Service on Change option. Applicable for Related Content.
PTCS_MAPNULLVALUE	Is Value Required Option

The following table lists the non-reconcilable fields. These fields contain metadata and might be required for the functionality to work; any modification is considered as customization.

Field	***Comment***
PTCS_MAPFIELDTYPE	Mapping Type
PTCS_PNLFLDNAME	Page Field Menu Option

PSPTCS_MNUFLDRS (Folders)

This table stores the data of the folders and their sequencing. A maximum of one child folder for a non root folder is allowed. From the user interface, a user can add or delete folders, but cannot rename a folder. Also, a user can reorder the folder sequence by using the Move Up and Move Down options.

Note: Deleting a delivered folder is a non-reconcilable change.

The following table lists the configurable fields. Any modification to these fields is preserved.

Field	***Comment***
PTCS_FOLDERSEQNUM	Sequence/ ordering of folder

ADS Definitions for Related Content Data

PeopleSoft delivers the following ADS definitions for related content data:

ADS Definition	***Description***
RCF_SERVICE_DEFINITIONS (RC Service Definition)	Defines the related content services that can be any collaborative, analytical, or informational content, which is useful for completing business tasks.
RCF_SERVICES (RC Services)	Encapsulates the content reference for which the services are assigned along with the configuration data (Related Content Configuration) and how they are displayed in the related content frame (Related Content Layout). This ADS data set should be used for shipping Related Content Framework service changes from PeopleTools 8.54 onwards. In PeopleTools 8.53, this ADS data set was recommended for delivering new services only. In PeopleTools 8.54, this restriction is not valid, and this data set can be used for any update or new service.

You can also use Add To Project links available on Related Content Framework pages to add specific definitions to existing ADS projects.

Related Links

"Defining ADS Project" (Lifecycle Management Guide)
"Defining Data Set Content" (Lifecycle Management Guide)
"Inserting Data Content" (Lifecycle Management Guide)
"Copy Compare Overview" (Lifecycle Management Guide)
"Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide)

Upgrading or Copying Related Content Data Using Prior Methodologies

Rather than using Add to Project links and ADS definitions, you can use prior methodologies when your situation warrants their use.

Copying Related Content Data

You can use PeopleSoft Application Designer to create upgrade projects that contain these types of related content data:

- Related Content Definitions (that is, services defined on the Define Related Content Service page)

  Also select the Application Package related definition when the service type of the selected related content service is *Application Class*.

- Related Content Services (that is, service assignments made through the Manage Related Content Service menu item).

  - *Always* select the Related Content Configurations related definitions.
  - Also select the Related Content Layouts related definitions when the service was added to or deleted from a menu, or when the menu order was changed.

You can save these projects to files or copy them directly to other PeopleSoft databases.

See “Inserting Definitions Into Projects” (Application Designer Developer’s Guide).

Upgrading Related Content Data from PeopleTools 8.51 or PeopleTools 8.50

Use the Upgrade Related Content page to complete the import and upgrade of related content data from PeopleTools 8.51 or PeopleTools 8.50.

To upgrade related content data from PeopleTools 8.51 or PeopleTools 8.50:

1. On the system on the prior release, use PeopleSoft Data Mover to export the related content data.
2. On your system on the current release, use PeopleSoft Data Mover to import the related content data.
3. Access the Upgrade Related Content page (**PeopleTools** > **Portal** > **Related Content Service** > **Upgrade Related Content**).
4. Click the **Run** button to upgrade the imported related content data.

Configuring Related Content, Related Actions, and Menu and Frame Layouts

This section provides overviews of:

- Related content versus related actions.
- Component level versus page level assignment
- Service parameter, filter, and security configuration.
- Menu configuration.
- Related Content frame configuration.

- PeopleCode FieldChange event related actions configuration.

Then, this section discusses how to:

- Assign related content to classic components and pages.
- Assign related content to fluid components and pages.
- Assign related actions to classic pages, fluid pages, and pivot grids.
- Configure service parameters, filters, and security.
- Configure menus.
- Copying menu layouts.
- Configure the Related Content frame.
- Configure PeopleCode FieldChange event related actions.

Prerequisites

This table discusses the prerequisites for assigning related content.

Prerequisite	***Cross Reference***
For the Service service type, explicit service definitions must exist before you can assign these related content services to a component, page field, or pivot grid as related content or as related actions.	See Creating and Managing Related Content Service Definitions.
For the Content Reference, Grouplet, and Pagelet service types, which are accessed as anonymous services, an explicit service definition is not required. However, in certain circumstances, you might require an explicit service definition.	See Explicit Definition Versus Anonymous Services.
For pagelets that require parameters, you must enter the RCF_ SERVICE_PARAMETER attribute.	See Understanding Service URL Parameters.

Understanding Assignment of Related Content and Related Actions

The Related Content Framework is designed so that subject matter experts or business analysts with an understanding of PeopleSoft applications and your business processes can assign related content services to components with little or no assistance from the developer after the service is defined. Almost every aspect of assigning and configuring related content services is done in the browser; no access to developer tools, such as PeopleSoft Application Designer, is necessary.

The Manage Related Content Configuration page is the starting point for creating and assigning related content services. The first time you access this page, you will see that no application pages have been assigned related content. After you assign related content services to application pages, a grid appears listing the related content assignments.

When you design and configure related information correctly, the application user is presented with seamless access to related information, which can increase productivity and improve job performance.

Process Flow for Assigning Related Content and Related Actions

Related content assignment follows this basic process flow:

To assign related content or related actions to an application page or to assign related actions to a pivot grid, you will:

1. Select the application page (content reference) or pivot grid.
2. Determine whether the related content service is to be assigned as related content or as a related action. For pivot grids, related content can be assigned as related actions only.

   See Related Content Versus Related Actions.
3. Determine whether the related content service is to be assigned at the component or page field level.

   See Component Level Versus Page Level Assignment.
4. Select the specific service, content reference, or pagelet to use as the related content service.

   See Assigning Related Content to Classic Components and Pages, Assigning Related Content to Fluid Components and Pages, or Assigning Related Actions to Classic Pages, Fluid Pages, and Pivot Grids.
5. Configure the related content service.

   See Configuring Service Parameters, Filters, and Security.
6. Configure the menu and layout of the related content service.

   See Configuring Related Content Menus and Copying Menu Layouts.

Related Content Versus Related Actions

The PeopleSoft Related Content Framework allows related content services to be deployed in two ways—as related content and as related actions—based on where the content appears when the user accesses the service:

- *Related content* appears in the Related Content frame only, which can be selected to be displayed at the bottom or to the right of the transaction content.
- *Related actions* appear outside of the Related Content frame in a modal window, in a new browser window, in the existing browser window replacing the current transaction content, or in the existing browser window replacing the entire page.

Configure the related content service as *related content* by selecting the Configure Related Content page:

- If you want the related content to appear in a separate frame beside or below the transaction content.
- If you want the related content to be dynamically updated when the user moves to a different transaction instance, or changes a field value, and so on.

Configure the related content service as a *related action* by selecting the Configure Related Actions page:

- If you want the related content to appear in a modal window, in a new browser window, or to replace the current page.
- When you are configuring related content for pivot grids or keyword search results.

Component Level Versus Page Level Assignment

Assign the related content service or related action at the component level:

- If you want the related content to appear on each page of the component.
- If you want data to appear or change in the related content based on key values from the component.
- If you are configuring related actions for component search results

Assign the related content service or related action at the page field level:

- If you want the related content to appear on one, but not all, pages in a component.
- If you want data to appear or change in the related content based on non-key, page field values.
- If you are assigning the service as a *related action* and you want to add this action to an application page.
- If you are assigning embedded related content to a fluid page.
- If you adding the Analytic Service to a fluid page.

Understanding Service Parameter, Filter, and Security Configuration

When you assign a related content or related actions, you typically must configure the service parameters. Configuration can include mapping the field data to the service parameters, specifying which type of menu the service link is listed in, adding filters to the service, and specifying security options. The service parameter list and other configuration options vary based the service type, the service definition, and the service target that you choose.

Related content service configuration establishes the relationships between the service and the transaction data so that the service appears in context with the current transaction. To establish contextually relevant relationships, service configuration includes working with:

- Service parameter mapping.
- Security options.

Service Parameter Mapping

Many services include parameters that you must map to key fields or page fields, system variables, and fixed values. You map the parameters to these elements so that they can pass the data to the service and create the relevant context for the related content.

This table discusses the source of the parameters for each of the service types.

Service Type	***Parameter Source***
Content reference	Search key fields from the component search record **Note:** If the search record has no key fields, the system retrieves search key fields from the component context search record.
Grouplet	Values in the Name field if the RCF_SERVICE_PARAMETER content reference attribute is defined on the Content Reference Administration page.
Pagelet	Values in the Name field if the RCF_SERVICE_PARAMETER content reference attribute is defined on the Content Reference Administration page.
Service	Values in the URL Parameters grid for the service definition.

When you map parameters, usually you must first select the mapping type and then select the mapping data. Some services might already have the mapping defined.

This table discusses the mapping types.

Mapping Type	***Description***	***Example***
Fixed value	A hard-coded value that does not change respective to context.	Service that includes the service ID as a parameter for example, ANALYTICSERVICE.
Key field	Primary or composite keys from the component search record.	DEPT_ID from the department table.
Page field	Any level 0, 1, 2, or 3 field in the component buffer.	SCHOOL_NAME on the education page.
System variable	Any variable that the PeopleSoft system resolves at run time.	For example, %UserId (the current user).

After you select a mapping type, click the **Select** button to see the mapping type prompt page.

Note these points about the prompt pages:

- If the mapping type is *Key Field,* then the prompt page displays a list of search key fields from the component search record.

- If the mapping type is *Page Field,* then the prompt page constructs a grid and dynamically populates it with a row for each page field that is available to the service. For each page field row, the grid also includes a hidden field indicator, the occurs or scroll level, the record (table) name, the field definition name, and the page field ID number.

  Use the page field ID number to identify a specific occurrence of a page field when the field appears multiple times in the same occurs level on the page. Each instance of a page field has a unique ID number that you find on the Order tab of the page definition in Application Designer.

- If the mapping type is *System Variable,* then the prompt page displays valid system variables.

  See “Understanding System Variables” (PeopleCode Language Reference).

- If the mapping type is *Fixed Value,* then the **Select** button is hidden.

Note: To reduce mapping effort, the PeopleSoft Related Content Framework defines a set of reserved words and service parameters.

See Reserved Words Default Mapping.

Service Filters

Important! Service filters are desupported and are retained for backward compatibility only.

Select this option to indicate that a service filter is required when this service is assigned to an application page.

A service filter hides or displays a service link in a field-level menu or the Related Content menu based on the value in the field at runtime. The application package PeopleCode that you specify can access the component buffer, using field values to trigger data-specific logic that causes the related content service link to be visible or hidden based on the value of the field. The filter enables the context of the data to determine whether the service link appears.

Security Options

The service definition determines the choices of security options that appear.

See Understanding Service Security.

Understanding Menu Configuration

Menus provide access to related content services. Three menu layout options exist:

- Related Actions menus can contain up to three levels of folders. Folders appear first in the menu followed by links.
- The Related Content menus display a flat hierarchy and can contain no folders.
- Page field contextual menus can contain two sections: related actions and related content.
  - Related actions appear first; this part of the menu can be hierarchical displaying folders and links.

- Related content appears last; this part of the menu is flat displaying links only.

Understanding the Configuration of the Related Content Frame

The PeopleSoft Related Content Framework enables you to configure classic related content to appear in two related content frames. The Related Content frames appear beneath or to the right of the transaction page so that users can view the transaction page and the related content simultaneously. When you assign related content to a content reference, you configure the services to appear either in the bottom frame or the right frame; the two frames cannot appear simultaneously.

Understanding PeopleCode FieldChange Event Related Actions

The PeopleSoft Related Content Framework provides a mechanism to associate a FieldChange event of a particular field to an existing action widget. The association of a FieldChange event to an existing action widget is valid only when the following conditions are met.

- The field type must be Push Button or Hyperlink.
- The destination must be PeopleCode Command.

In addition, when an action widget includes only a single action for a user, the PeopleSoft Related Content Framework provides three options to display the single action:

- Action widget
- Link
- Button

Common Elements Used in This Section

Field or Control	***Description***
Configure	Click to access the Configure Service page, where you map field values to the service parameters.
Content Reference	This is the content reference to which you are assigning the related content. You select the content reference on the Select a Content Reference page. This field is display-only.

Field or Control	***Description***
Display in field menu	Select to enable access to a related content field-level contextual pop-up menu in the field to which you are mapping the service parameter. This field is enabled only when the mapping type is *Page Field.* Selecting this check box causes the red related content glyph to appear in the field at runtime, even if the field is display-only. **Important!** When mapping a service parameter to a page field, always examine the page to determine whether the selected page field lies within a collapsible page control such as a group box, grid, and scroll area. Services that access page fields that lie within a collapsible control that is set to the collapsed state by default must be configured to use the **Display in field menu** contextual menu option only. **Note:** If you select a hidden field, the menu does not appear at runtime.
Enable	Select to activate the configured related content or related action. **Note:** Deselecting an item allows you to deactivate the related content or related action while retaining its configuration.
Is Value Required?	Some service might accept valid null values being passed to its parameters. If you select the **Refresh Service On Change** and **Is Value Required?** check boxes for this parameter, then when the mapped field value of that parameter on the transaction page changes to a null value, the system does not refresh the related content frame.
Mapping Data	Enter the constant if the mapping type is *Fixed Value.* For all other mapping types, this field is disabled and displays the name of the field or variable that you map to the service parameter.

Field or Control	***Description***
Mapping Type	Select from four possible mapping types to map the service parameter to: • *Fixed Value:* Select to map the service parameter to a constant. Selecting this option enables the **Mapping Data** field. • *Key Field:* Select to map the service parameter to one of the level 0 search keys of the component assigned to the content service. **Note:** For related actions, the key name used for the related actions component is the field name that you enter here. The key value comes from the corresponding key field from the detail component key list. • *Page Field:* Select to map the service parameter to any level 0, level 1, or level 2 page field, including hidden fields. This option is available for page-level services only. • *System Variable:* Select to map the service parameter to one of the available system variables.
Package	Enter the application class package name to use for the service filter.
Path	Enter the names of each subpackage in the application class hierarchy that define the location of the application class that defines the service filter. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon. **Note:** The application class parameters are validated when you save the definition; the service does not allow you to save an invalid application class.
ClassID	Enter the class name that defines the data-specific logic.
Page	Select the specific page in the component to which you are assigning related content. This field appears in the grid when you select the **Services at Page level** option.
Page Name	Displays the page label. This field is display-only. This field appears in the grid when you select the **Services at Page level** option.
Portal Name	This is the name of the current portal where the related content will be assigned. This field is display-only.

Field or Control	***Description***
Read help text	Click to display the help text as defined on the Related Content Service Definition page. See Writing Help Text for Related Content Service Definitions.
Refresh Service On Change	Select to have the service data refresh automatically when the key values on the transaction page changes. The default value is deselected. **Important!** Use this option judiciously. If the service is linked to a level 1 or level 2 key field and the application user scrolls through data rapidly, performance can be affected. **Note:** If the page or the page field to which a service is mapped is in *deferred processing mode,* a trip to the server is required to refresh related content.
Required	Specifies whether the service parameter is defined as *Required* in the service definition on the Related Content Service Definition page in the **Service URL Parameter** grid. See Understanding Service URL Parameters.
Select	Click to access the lookup page associated with the corresponding service type. The lookup pages enable you to choose the content reference, pagelet, or service that you want to assign as related content. This button is enabled only after you select the service type.
Service Parameter	Displays the list of parameters that the service requires to form the contextual relationship to the primary transaction page. See “Understanding Search Records” (Application Designer Developer’s Guide). See Understanding Service URL Parameters.
Service ID	Specifies the unique identifier of the service definition. For content references and pagelets, the service ID corresponds to the Name field on the Content Reference Administration page.
Service Label	Enter the label for this service. The content reference name (for Content Reference, Grouplet, and Pagelet service types) or the service definition name appears as the service label by default when you select the service; you can change the default value.

<table>
<tr><th>Field or Control</th><th>Description</th></tr>
<tr><td>Service Target</td><td>Select the target location where the related action appears. Select from these options:
Modal Window - select to have the related content appear in a modal window that hovers in front of the transaction page.
New Window - select to have the related content appear in a new browser window.
Replace Window - select to have the related content refresh the entire browser window, including navigation.
Target Content - select to have the related content appear in the target area, replacing the current page.</td></tr>
<tr><td>Service Type</td><td>The service types available depend on the item being configured and the type of related content or related actions:
Content Reference – Select this option to assign an existing content reference as related content. Because content references can be used as anonymous services, you do not have to create an explicit service definition. However, in certain circumstances, you might require an explicit service definition. See Explicit Definition Versus Anonymous Services.
Grouplet – Select this option to assign an existing local content reference definition as related content. Because locally registered content references can be used as anonymous services, you do not have to create an explicit service definition.
Pagelet – Select this option to assign an existing locally registered pagelet as related content. Because locally registered pagelets can be used as anonymous services, you do not have to create an explicit service definition.
Service – Select this option to assign an existing service that you configured using the Define Related Content Service page.
Note: After you select a service type, the system enables the Select button.</td></tr>
<tr><td>Add to Project</td><td>Click the Add to Project link to access the Add Project page to insert this related content configuration into an ADS project. See “Searching for, Adding and Deleting Projects” (Lifecycle Management Guide) for more information.</td></tr>
</table>

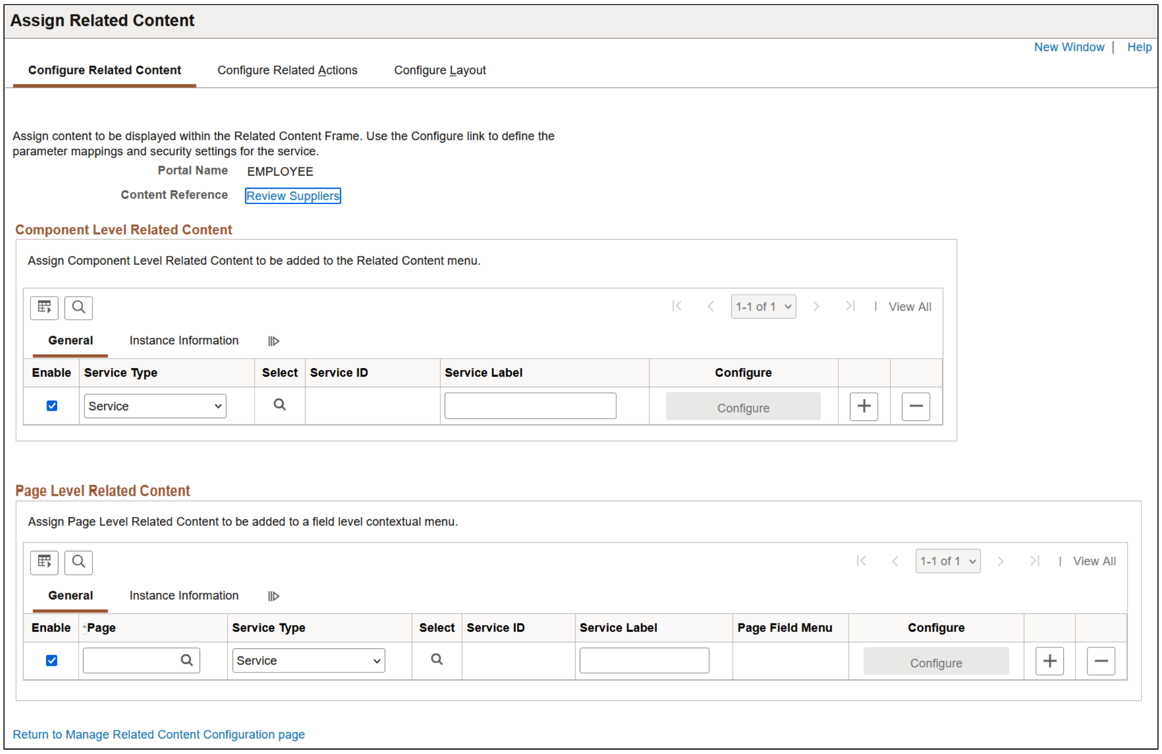
Assigning Related Content to Classic Components and Pages

To assign related content to a classic component or page:

1. Access the Configure Related Content page:

- Click an Edit button on the Manage Related Content for Content References page.
- Select a content reference for a classic page on the Select a Content Reference page.

This example illustrates the fields and controls on the Configure Related Content page for a classic content reference.



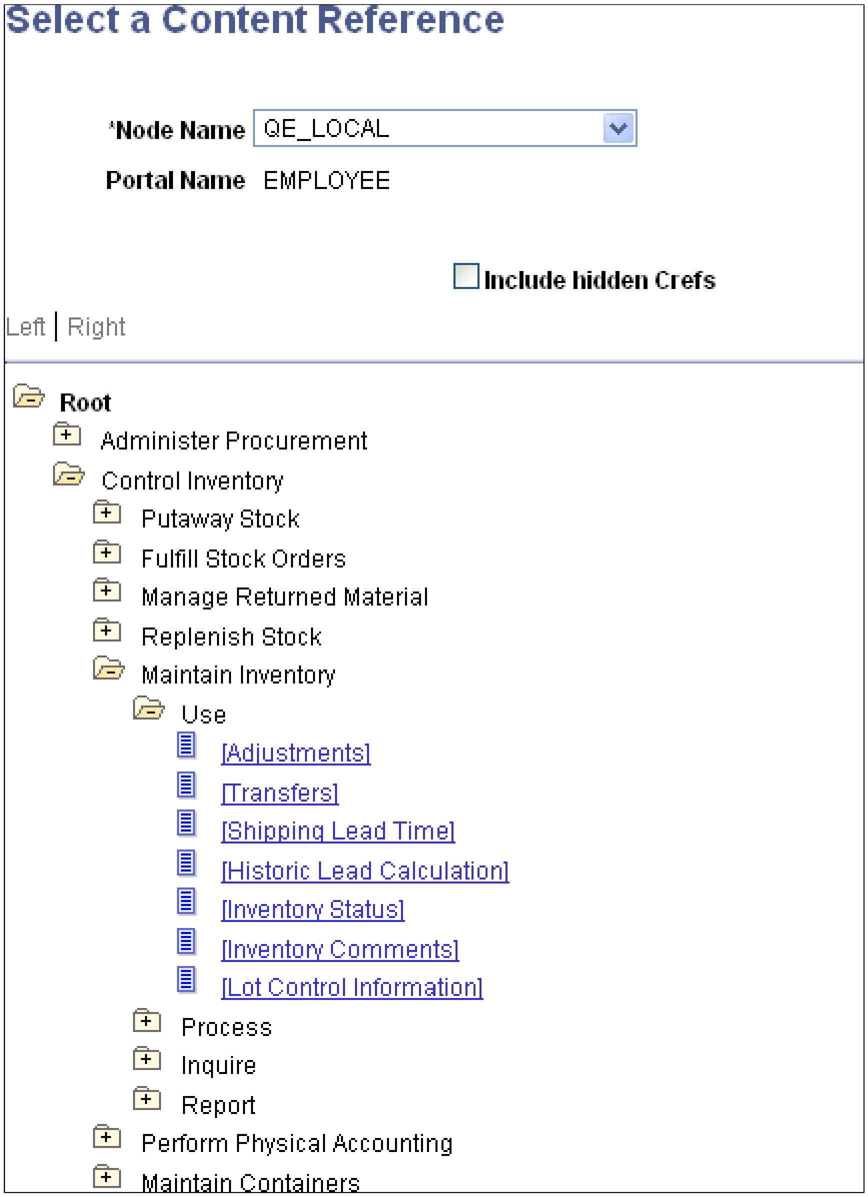
2. Determine whether the assignment is to be made at the component level or at the page level. See Component Level Versus Page Level Assignment for more information.

   If the assignment is to be made at the page level, in the **Page Level Related Content** group box, click the Page lookup button and then, select the page to which you want to assign the service.

3. Select the service type:

   - If you select *Content Reference,* click the Select button and select the content reference to use as an anonymous service on the Select a Content Reference page:

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



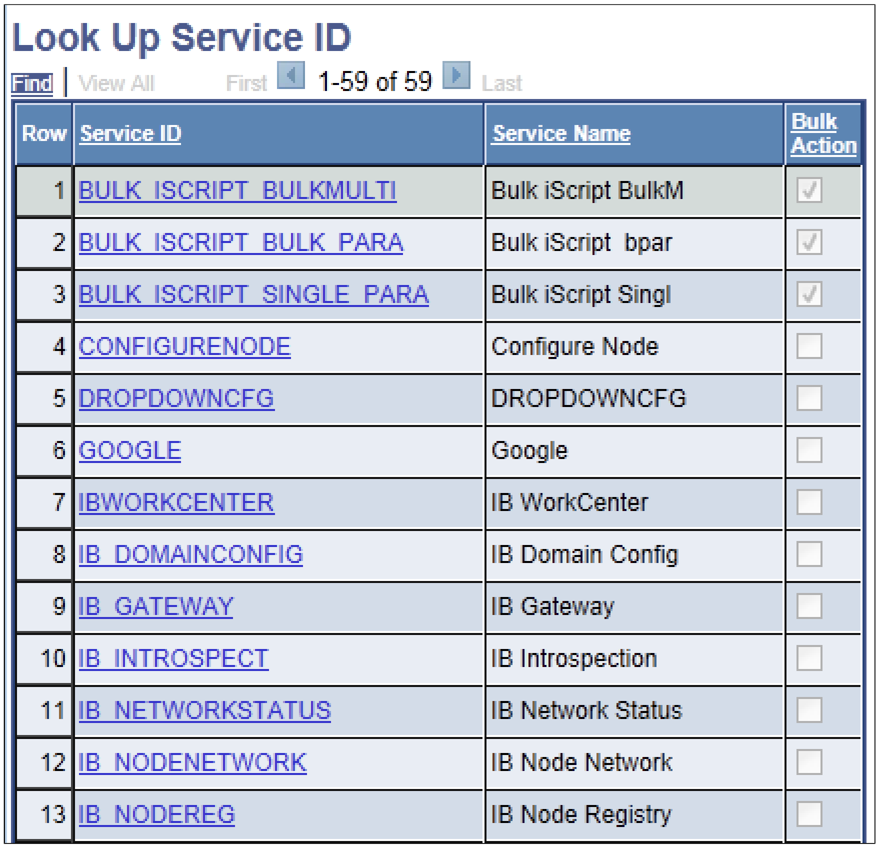
Select the local node or a remote node to select a local or remote content reference.

- If you select *Pagelet,* click the Select button and select the pagelet from the Pagelet Search page to use as an anonymous service.

  See Searching for and Importing Pagelets and Tiles or more information on searching for pagelets on local and remote nodes.

- If you select *Service,* click the Select button and select the explicitly defined service on the Look Up Service ID page:

This example illustrates the fields and controls on the Look Up Service ID page.



Look Up Service ID

Find | View All First 1-59 of 59 Last

Row	Service ID	Service Name	Bulk Action
1	BULK_ISCRIPT_BULKMULTI	Bulk iScript BulkM	✓
2	BULK_ISCRIPT_BULK_PARA	Bulk iScript bpar	✓
3	BULK_ISCRIPT_SINGLE_PARA	Bulk iScript Singl	✓
4	CONFIGURENODE	Configure Node	
5	DROPDOWNCFG	DROPDOWNCFG	
6	GOOGLE	Google	
7	IBWORKCENTER	IB WorkCenter	
8	IB_DOMAINCONFIG	IB Domain Config	
9	IB_GATEWAY	IB Gateway	
10	IB_INTROSPECT	IB Introspection	
11	IB_NETWORKSTATUS	IB Network Status	
12	IB_NODENETWORK	IB Node Network	
13	IB_NODEREG	IB Node Registry	

4. Click the Configure button to configure the service parameters, filters, and security. See Configuring Service Parameters, Filters, and Security.

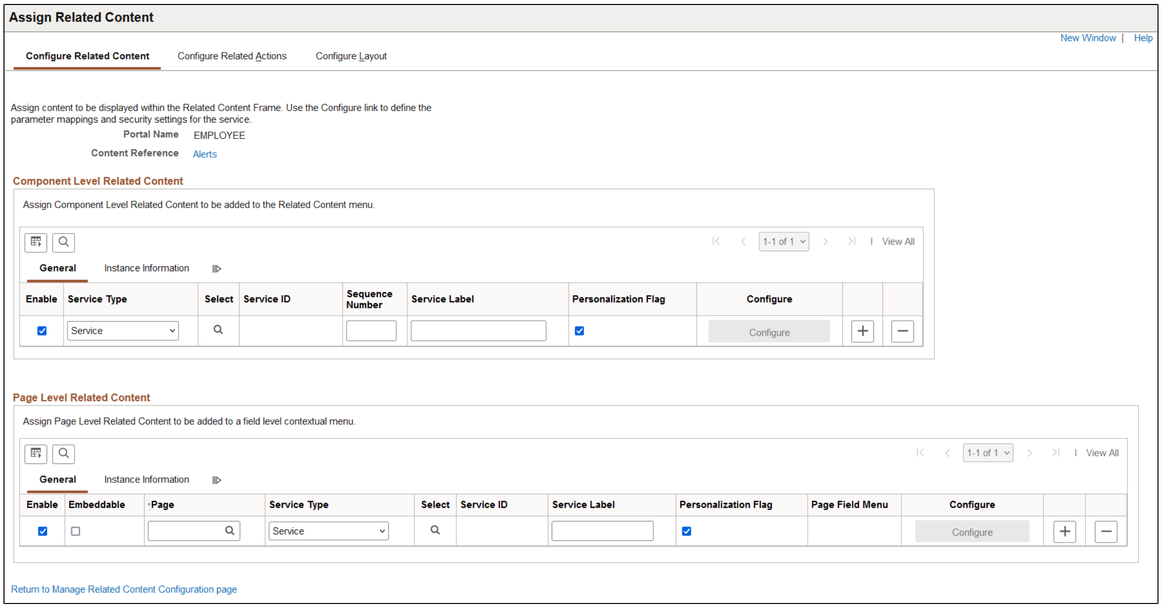
Assigning Related Content to Fluid Components and Pages

To assign related content to a fluid component or page:

1. Access the Configure Related Content page:

   - Click an Edit button on the Manage Related Content for Content References page.
   - Select a fluid content reference on the Select a Content Reference page.

This example illustrates the fields and controls on the Configure Related Content page for a fluid content reference.



2. Determine the type of assignment to be made:

   - Related content at the component level to be displayed in the Related Information panel at the right of the page.
   - Embedded related content at the page level.

3. For related content at the component level:

   a. Select the service type: *Service* or *Grouplet*.

   b. Then select a PeopleSoft Tile related content service definition (*Service*) or an existing content reference definition (*Grouplet*).

   c. Specify the sequence number for the item within the Related Content frame.

   Note: Unlike other items of related content, the sequence is not specified on the Configure Layout page tab.

   d. Select the User Personalization option if you want to give users the ability to show or hide this related content item.

   e. Click the Configure button to configure the service parameters, filters, and security. See Configuring Service Parameters, Filters, and Security.

   f. Save the configuration without configuring the layout.

   Note: The Configure Layout page tab is not used to configure fluid related content.

4. For embedded related content at the page level:

a. Click the Page lookup button and then, select the page to which you want to assign the service.

Important! Select only a page that appears in the actual component definition in Application Designer. Do not select any of the PeopleTools-delivered pages such as PTGP_USERPRE_S1_FL, PTIA_MDSIDEPAGE, and so on.

---

b. Always select the Embeddable option.

c. Select the service type: *Service* or *Grouplet*.

d. Then, select a PeopleSoft Tile related content service definition (*Service*) or an existing content reference definition (*Grouplet*).

e. Select the User Personalization option if you want to give users the ability to show or hide this related content item.

f. Click the Configure button to configure the service parameters, page field menu, and filters. (See Configuring Service Parameters, Filters, and Security for additional information.)

g. For the Page Field Menu field, select the related content group box (drop zone) defined in Application Designer. (See "Working with Embedded Related Content" (Fluid User Interface Developer's Guide) for more information on creating related content group boxes.)

h. Save the configuration.

Note: At the page level, the Analytic Service (ANALYTICSERVICE) can also be selected as the service type to display simplified analytics in the Related Information panel at the right of the page. See "Simplified Analytics Overview" (Pivot Grid) for more information on configuring the Analytic Service.

Related Links

Fluid Related Content

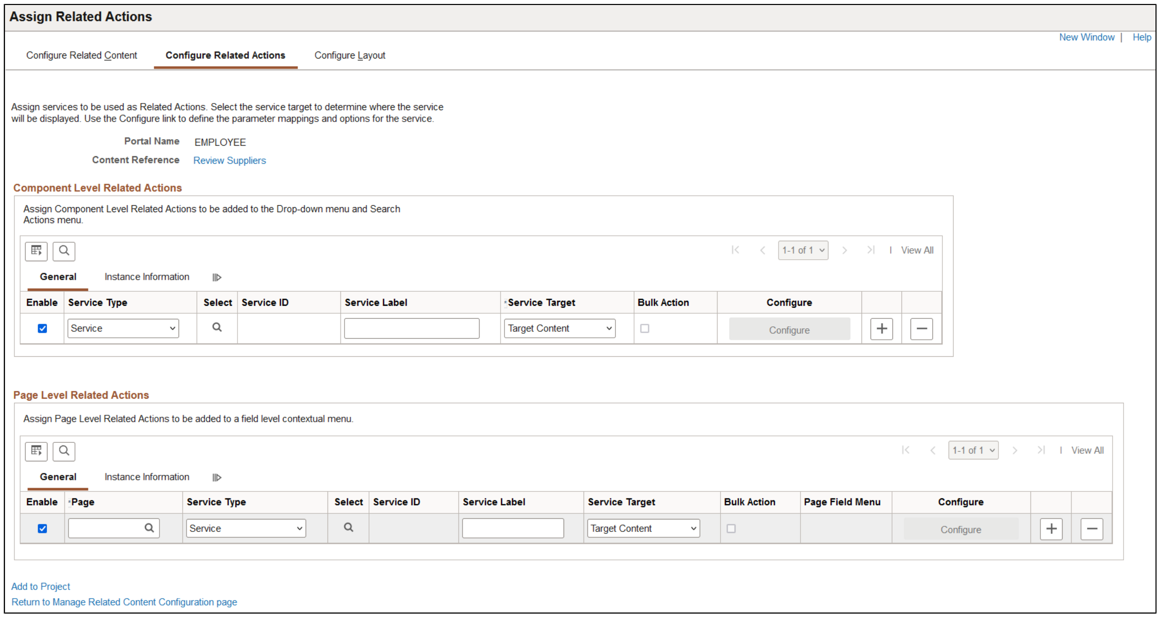
Assigning Related Actions to Classic Pages, Fluid Pages, and Pivot Grids

To assign related actions to classic pages, fluid pages, or pivot grids:

1. Access one of the following pages:

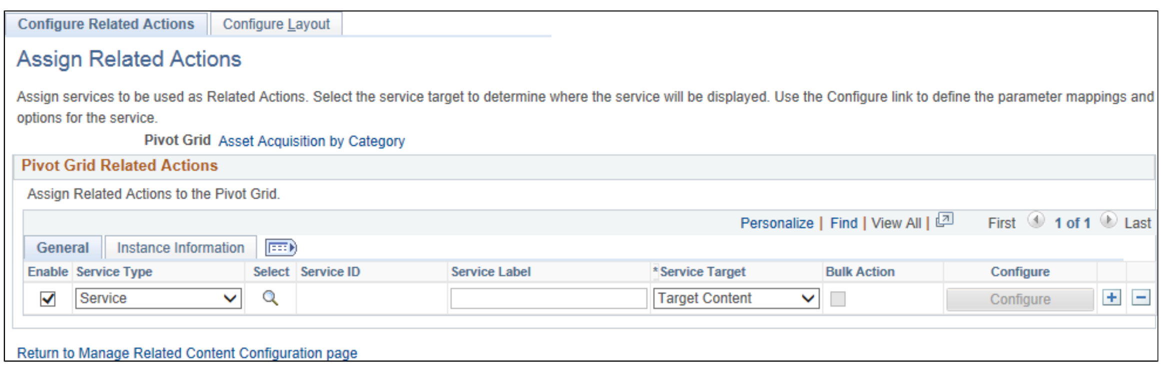
    - Configure Related Actions page (for content references, select the Configure Related Actions page tab on the Configure Related Content page).

This example illustrates the fields and controls on the Configure Related Actions page for a classic content reference.



- Configure Related Actions page (for pivot grids, click an Edit button on the Manage Related Actions for Pivot Grids page or select a pivot grid on the Look Up Pivot Grid page).

  This example illustrates the fields and controls on the Configure Related Actions page for a pivot grid.



2. Determine the type of assignment to be made:

   - For classic and fluid pages, related action assignments can be made at the component and the page level.
   - For pivot grids, related action assignments can be made at the pivot grid level only.

   See Component Level Versus Page Level Assignment for more information.

3. Select the service type:

   - If you select *Content Reference,* click the Select button and select the content reference to use as an anonymous service on the Select a Content Reference page.

Select the local node or a remote node to select a local or remote content reference.

- If you select *Pagelet,* click the Select button and select the pagelet from the Pagelet Search page to use as an anonymous service.

  See Searching for and Importing Pagelets and Tiles or more information on searching for pagelets on local and remote nodes.

- If you select *Service,* click the Select button and select the explicitly defined service on the Look Up Service ID page.

4. Click the Configure button to configure the service parameters, filters, and security. See Configuring Service Parameters, Filters, and Security.

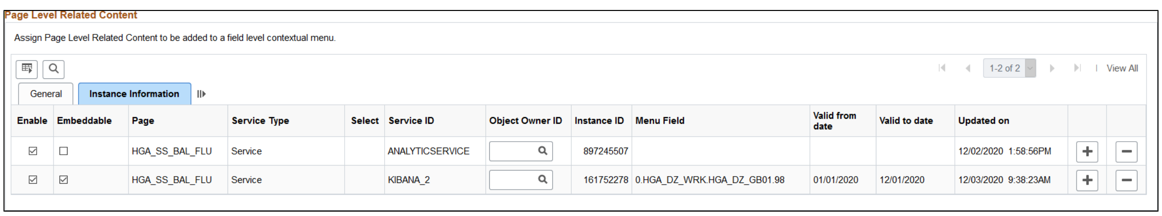
Related Links

PeopleSoft Fluid UX Standards: Related Actions

Viewing Instance Information

After a service configuration is saved, instance information is generated by the system. In either the Component Level Related Content, Page Level Related Content, Component Level Related Actions, or Page Level Related Actions group box, select the Instance Information tab to view the generated system data.

This example illustrates the fields and controls on the Instance Information tab. You can find definitions for the fields and controls unique to this tab later on this page.



Field or Control	***Description***
Object Owner ID	Select an optional object owner for the configuration of this service instance.
Instance ID	Displays the system-generated instance ID.
Menu Field	Displays the page level field identifier in the following format: *occurs_lvl.RECORD_NAME.FIELD_NAME.Field_ID*

Field or Control	***Description***
Valid from dateValid to date	Displays the validity period specified on the related content service definition. **Note:** The validity period specifies whether a service is displayed as related content or a related action at runtime. Regardless, a service can still be configured or displayed within an existing configuration outside of its validity period.
Updated on	The last date and time the service instance configuration was updated.

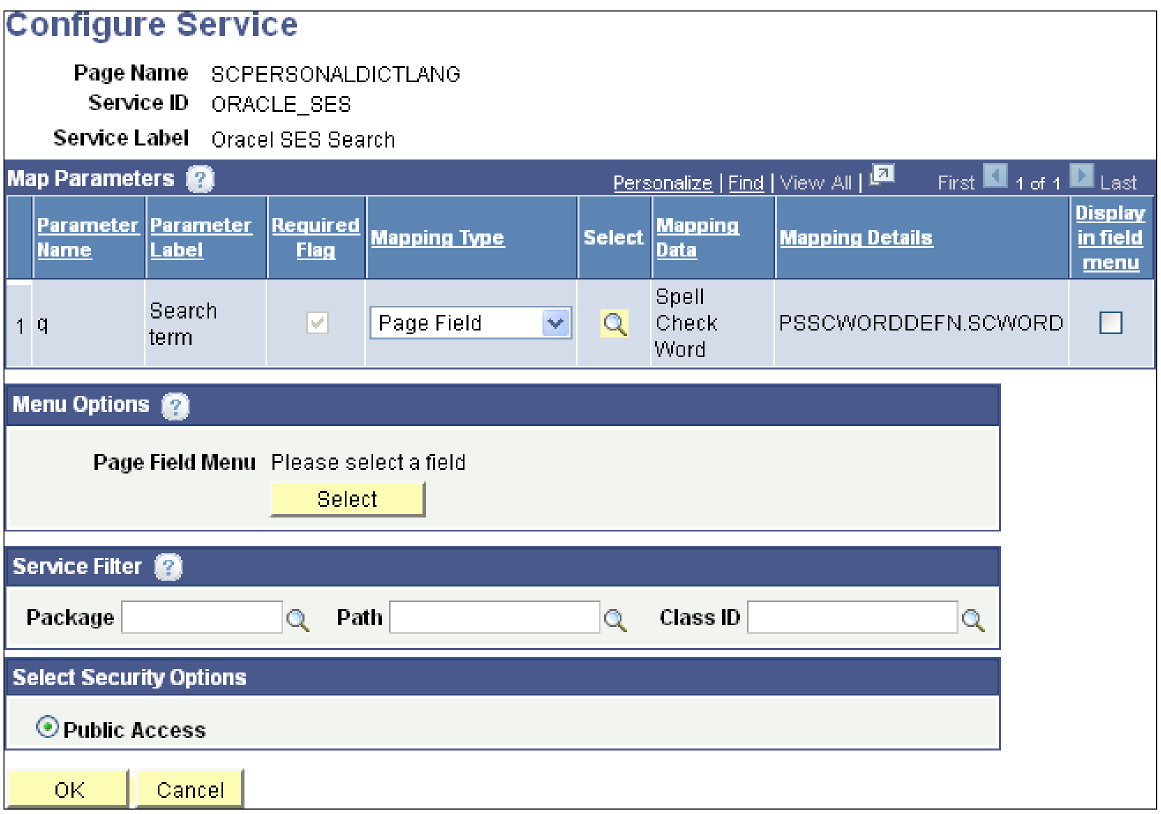
Configuring Service Parameters, Filters, and Security

Use the Configure Service page to configure the service instance.

Navigation:

- On the Configure Related Content page, click the **Configure** button for a service instance.
- On the Configure Related Actions page, click the **Configure** button for a service instance.

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



To configure the service parameters, filters, and security:

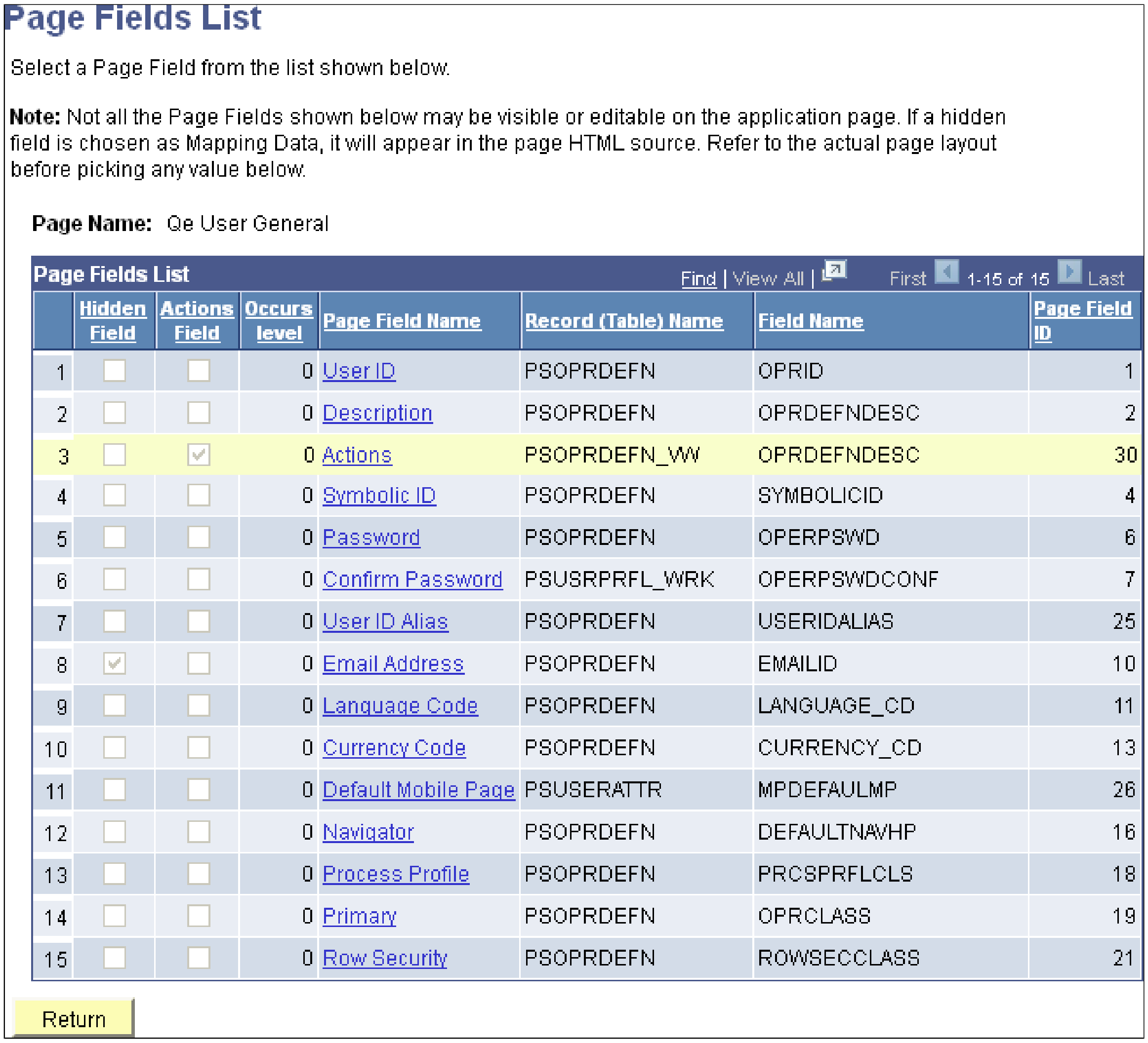
1. For each service parameter, select the mapping type.
2. For each service parameter, do one of the following:
   - For a fixed value mapping type, enter the value in the **Mapping Data** field.
   - For a key field, page field, or system variable mapping type, click the **Select** button.

     Click a link to select the field or variable on the prompt page.
3. For page-level services, do one of the following:
   - Select the **Display in field menu** option for the field that provides access to the contextual pop-up menu.

     Note: When the **Display in field menu** option is selected, then the **Select** button in the Menu Options group box is disabled.

   - Click the **Select** button in the Menu Options group box to select a page field different from the one that is mapped to the service parameter; the Page Fields List page is displayed:

This example illustrates the fields and controls on the Page Fields List page:



Page Fields List

Select a Page Field from the list shown below.

Note: Not all the Page Fields shown below may be visible or editable on the application page. If a hidden field is chosen as Mapping Data, it will appear in the page HTML source. Refer to the actual page layout before picking any value below.

Page Name: Qe User General

Page Fields List — Find | View All | First 1-15 of 15 Last

	Hidden Field	Actions Field	Occurs level	Page Field Name	Record (Table) Name	Field Name	Page Field ID
1	☐	☐	0	User ID	PSOPRDEFN	OPRID	1
2	☐	☐	0	Description	PSOPRDEFN	OPRDEFNDESC	2
3	☐	☑	0	Actions	PSOPRDEFN_VW	OPRDEFNDESC	30
4	☐	☐	0	Symbolic ID	PSOPRDEFN	SYMBOLICID	4
5	☐	☐	0	Password	PSOPRDEFN	OPERPSWD	6
6	☐	☐	0	Confirm Password	PSUSRPRFL_WRK	OPERPSWDCONF	7
7	☐	☐	0	User ID Alias	PSOPRDEFN	USERIDALIAS	25
8	☑	☐	0	Email Address	PSOPRDEFN	EMAILID	10
9	☐	☐	0	Language Code	PSOPRDEFN	LANGUAGE_CD	11
10	☐	☐	0	Currency Code	PSOPRDEFN	CURRENCY_CD	13
11	☐	☐	0	Default Mobile Page	PSUSERATTR	MPDEFAULMP	26
12	☐	☐	0	Navigator	PSOPRDEFN	DEFAULTNAVHP	16
13	☐	☐	0	Process Profile	PSOPRDEFN	PRCSPRFLCLS	18
14	☐	☐	0	Primary	PSOPRDEFN	OPRCLASS	19
15	☐	☐	0	Row Security	PSOPRDEFN	ROWSECCLASS	21

Return

Use this methodology when you want to place the page field contextual menu on a page field that is different from the one mapped to the service parameter. For example, the service parameter could be mapped to the OPRID value, but this methodology would allow you to place the page field contextual menu on the OPRDEFNDESC field.

In addition, use this methodology to add the related action to the menu on the action widget on a page.

4. (Optional) Enter the package, path, and class ID of the service filter.

Important! Service filters are desupported and are retained for backward compatibility only.

Note: If the *App Class Required* option was selected on the service definition, then a service filter is *not* optional and must be identified to be able to save the service configuration.

5. Select the security option.

6. Click the **OK** button to accept the configuration and return to the configuration page.

Related Links

"Specifying Type Properties for Push Buttons or Links" (Application Designer Developer's Guide)

Creating and Managing Related Content Service Definitions

Configuring Related Content Menus

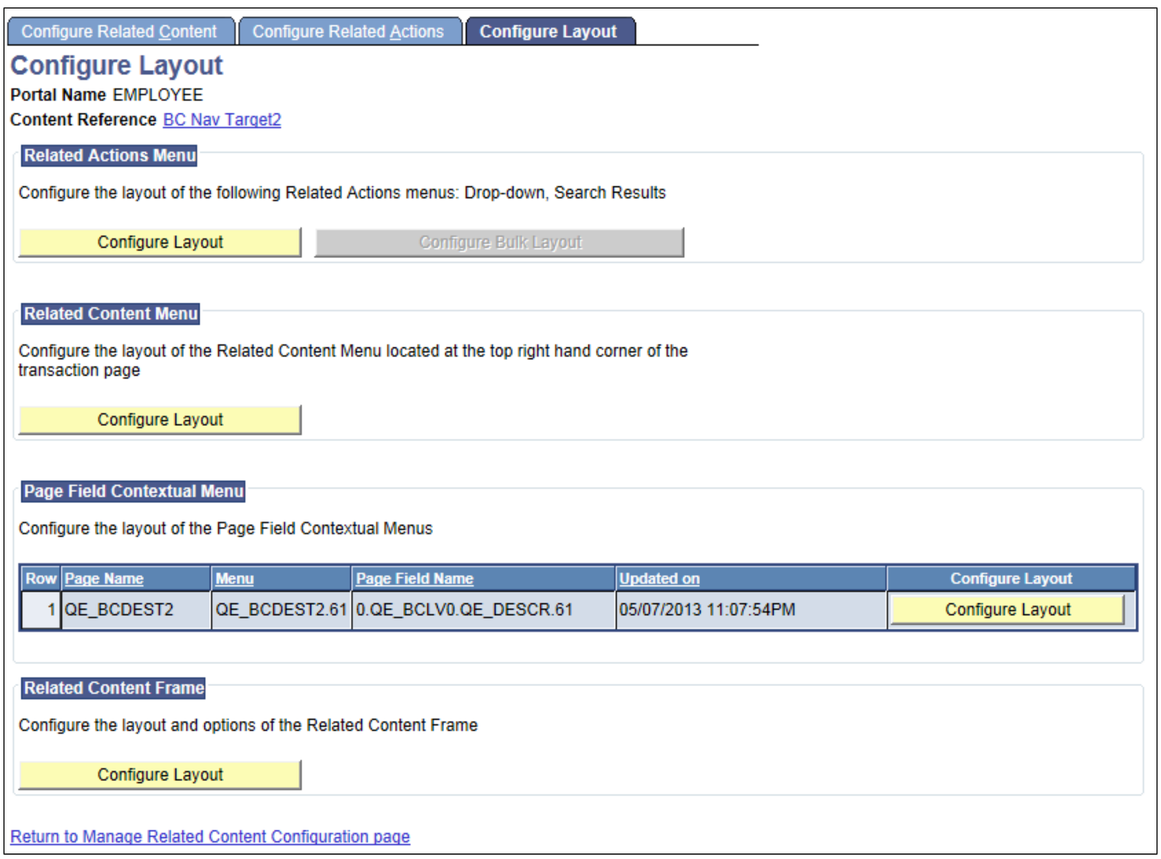
Use the Configure Layout page to configure the layout of the related actions menu, the related content menu, a page field contextual menu, or the related content frame.

Navigation:

- On the Configure Related Content page, select the Configure Layout tab.
- On the Configure Related Actions page, select the Configure Layout tab.

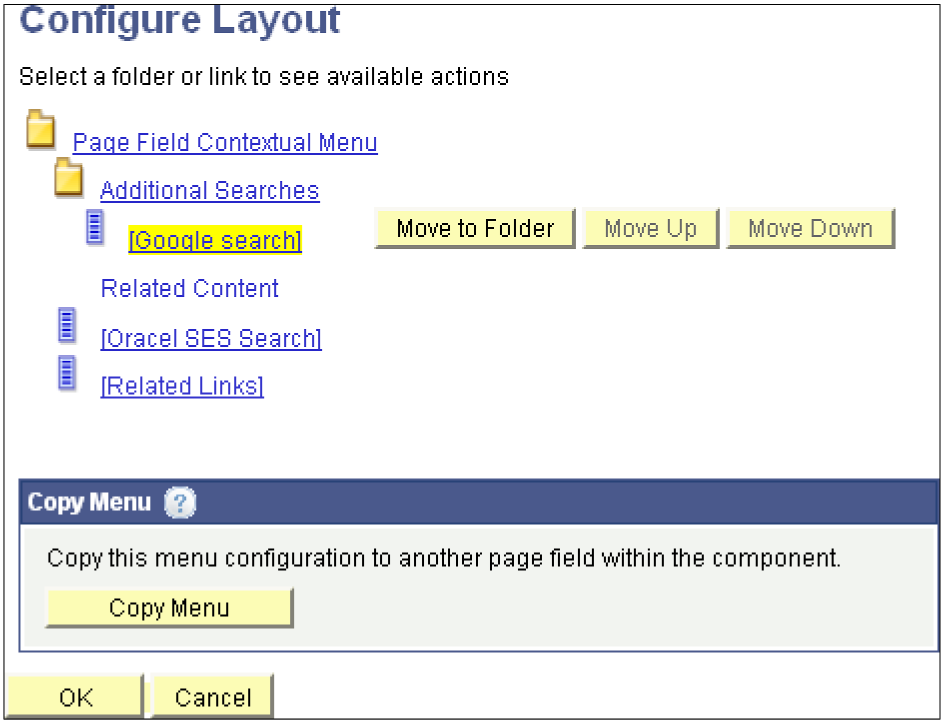
1. Access the Configure Layout page.

   This example illustrates the fields and controls on the Configure Layout page.



2. On the Configure Layout page, click an active **Configure Layout** button for related actions, related content, or a page field to configure a menu layout.

This example illustrates the fields and controls on the Configure Layout secondary page. You can find definitions for the fields and controls later on this page.



3. For related actions menus and page field contextual menus, you can add and move folders in the menu.

Important! You cannot add or move folders in a related content menu.

When you select a folder, these options appear:

Note: Not all options appear for all folders.

Field or Control	***Description***
Add Folder	Click to add a folder one level below the selected folder. If this option does not appear, you cannot add a subfolder.
Delete Folder	Click to delete the folder If this option does not appear, you cannot delete the folder.
Move Up	Click to move the folder up within the list of folders in the same level.
Move Down	Click to move the folder down within the list of folders in the same level.

4. For all menu types, you can move service links within the menu structure.

When you select a service link, these options appear:

Note: Not all option appear enabled for all services.

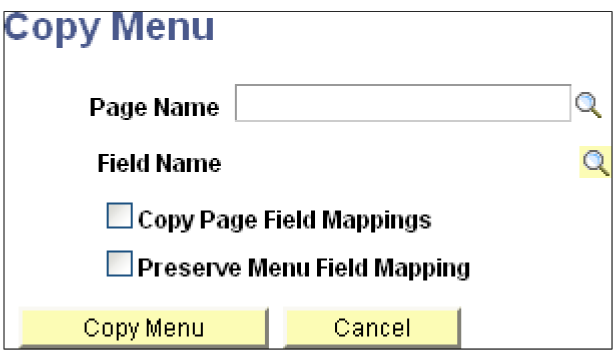
Field or Control	*Description*
Move to Folder	Click to access the Select a Folder page, which enables you to move the service to another folder. If this option does not appear, you cannot move the service link to a different folder.
Move Up	Click to move the service link up within the list of services in the same folder.
Move Down	Click to move the service link down within the list of services in the same folder.

5. Click the **OK** button.
6. Click the **Save** button.

In addition, you can use the Copy Menu page to copy a page field contextual field menu layout from one page field to a different page field on the same page or on a different page within the same component. See Copying Menu Layouts.

Copying Menu Layouts

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



Use the Copy Menu page to copy a page field contextual field menu layout and services configuration from one page field to a different page field on the same page or on a different page within the same component.

The following values are copied to the new page field:

- Service target.
- Service label.

- Key field mappings.
- Fixed value mappings.
- System variable mappings.
- Page field mappings (see note).
- Menu layout.
- Menu to which the copied service is attached.

Important! The **Copy Page Field Mappings** option must be selected to copy page field mappings. However, no error checking is performed to ensure that the page field mappings are still correct. Therefore, select this option only in circumstances that provide equivalent mapping of page fields.

The **Preserve Menu Field Mapping** option is valid only when the **Copy Page Field Mappings** option is also checked. Use the **Preserve Menu Field Mapping** option only when the menu location is moved to the new field. For example, a service parameter is mapped to the User ID field, which is marked as the menu display field as well. When this menu is copied to a new page field, Description, we want to keep the original parameter mapping to the User ID field. In this instance, we would also select the **Preserve Menu Field Mapping** option.

After the menu layout and service configuration is copied to the new page field, additional manual configuration might be required depending on the original configuration and how it was copied:

- If the services copied do not contain any page field mappings or if the **Copy Page Field Mappings** option was selected, the services will automatically be marked active and will be assumed as completely configured.
- If the services copied contain page field mappings *that were not copied,* the services will automatically be marked inactive by default. You must manually map these service parameters to page fields. See Configuring Service Parameters, Filters, and Security.
- After you manually map unmapped service parameters, the PeopleSoft Related Content Framework will check other services in the same menu to see if they contain a service parameter by the same name. If so, you will be prompted to copy this mapping to the other services as well. Choosing yes will apply the same mapping to any other inactive services in the same menu that also contain the same service parameter. The framework will not update any active service configurations that also contain the same service parameter.
- After the framework applies this service parameter mapping to other inactive services, an additional check will be made to determine if all service parameters are now mapped. If they are, you will be prompted to mark all inactive services as active in the new page field menu.

Configuring the Related Content Frame

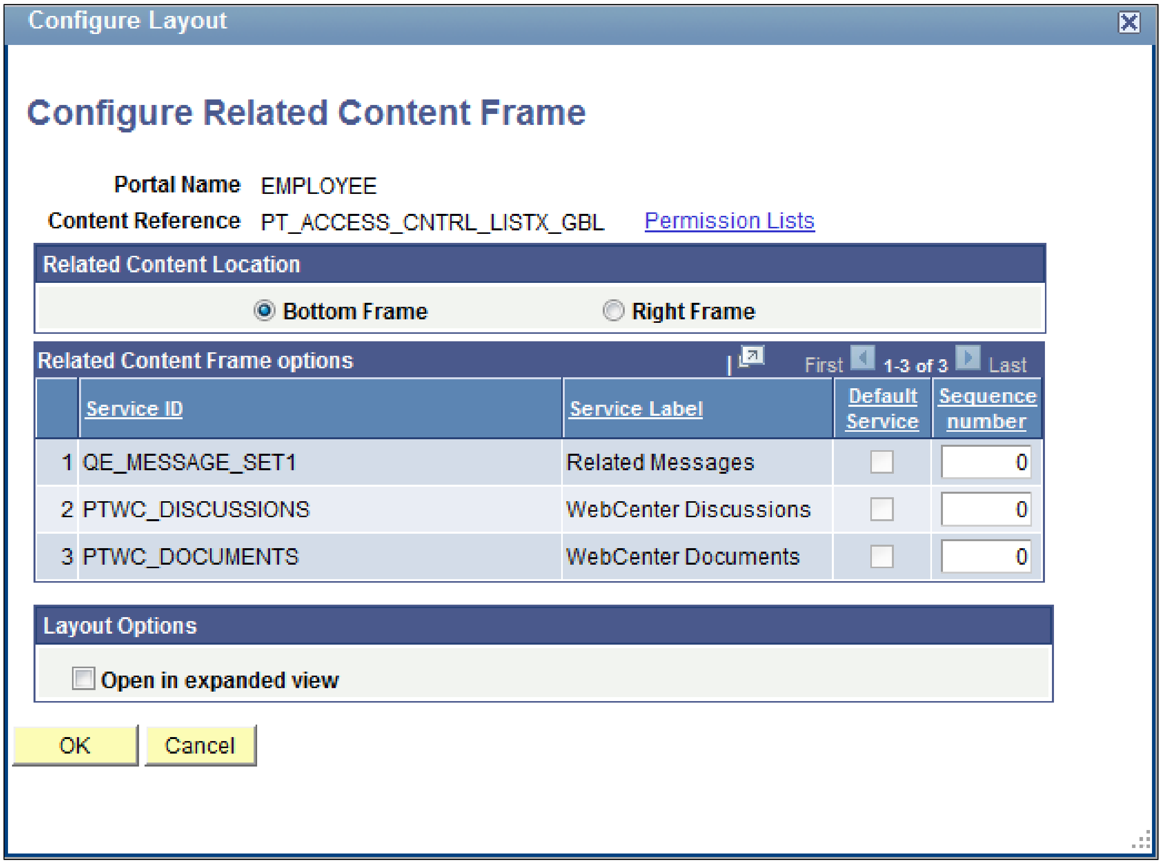
Use the Configure Related Content Frame page to configure the layout of related content frame on classic pages only.

Note: You cannot configure this frame for fluid pages.

Navigation:

On the Configure Layout page, click the Configure Layout button in the **Related Content Frame** group box.

This example illustrates the fields and controls on the Configure Related Content Frame page. You can find definitions for the fields and controls later on this page.



1. Select the frame location:

   - *Bottom Frame* – Select to have related content services appear as tabs in the frame beneath the transaction page.
   - *Right Frame* – Select to have related content services appear as pagelets in the frame to the right of the transaction page.

2. Configure the order and initial state of the services that appear in the Related Content frame:

   - *Default Service* – Select the service to be displayed by default when the Related Content frame is opened automatically at the bottom of the page.

     Note: You must select the **Open in expanded view** option first in order to select a default service.

   - *Initially Minimized* – Select so that the service appears in the closed state when the user initially accesses the Related Content frame at the right of the page.
   - *Sequence number* – Specify the sequence for the services.

3. Select the frame layout options:

   - *Open in expanded view* – Select to open the Related Content frame automatically when the user accesses the target application page rather than after the user selects a service.
   - *Max no of minimized pagelets* – Enter the number pagelets that can appear minimized when the Related Content frame appears at the right of the page. If a user minimizes more than this number, the pagelets move to the **More** drop-down list at the bottom of the frame.

Configuring Display of PeopleCode FieldChange Events as Related Actions

PeopleCode FieldChange Event Actions

You cannot configure the display of PeopleCode FieldChange events as related actions using the browser interface. You must use Application Designer. The following table describes the two options:

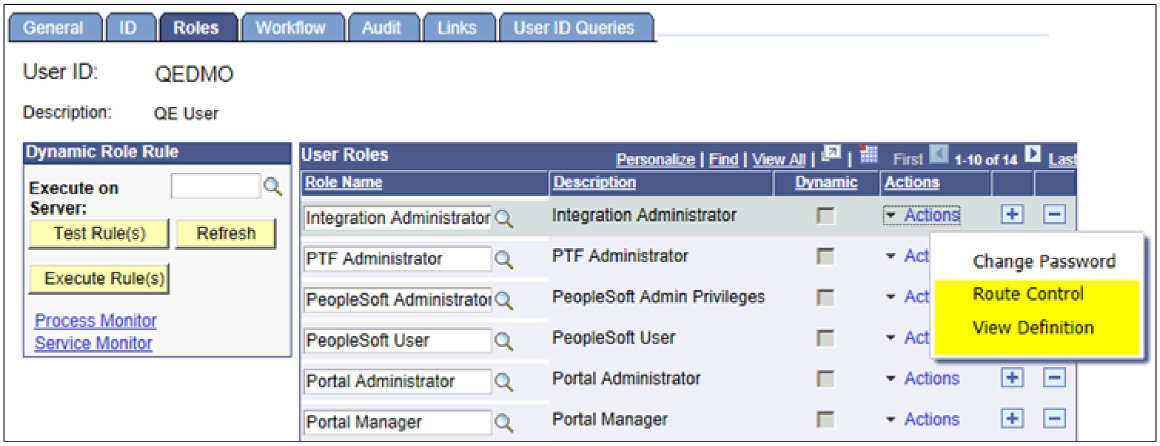
Display Option	***Configuration***
Show FieldChange event actions under the Actions menu only.	In Application Designer, access the Push Button/Hyperlink Properties dialog box, and in the Related Content Event group box, select Display in Related Action Menu. **Note:** The Display in Related Action Menu check box is available only if the field is of type Push Button or Hyperlink and the destination is PeopleCode Command. The Action Widget drop-down list displays all the action widgets configured for the page. When you select an action widget, the FieldChange event of this field is associated as a menu item in the widget. Related action widgets in the drop-down list will only display the level at which the field is selected.
Show FieldChange event actions under the Actions menu and on a page.	In Application Designer, access the Push Button/Hyperlink Properties dialog box, and in the Related Content Event group box, select Display in Related Action Menu and Show in Page.

See “Using Push Buttons and Links” (Application Designer Developer’s Guide).

FieldChange Event Actions in the Actions Menu

In this example, the Display in Related Action Menu check box is selected.

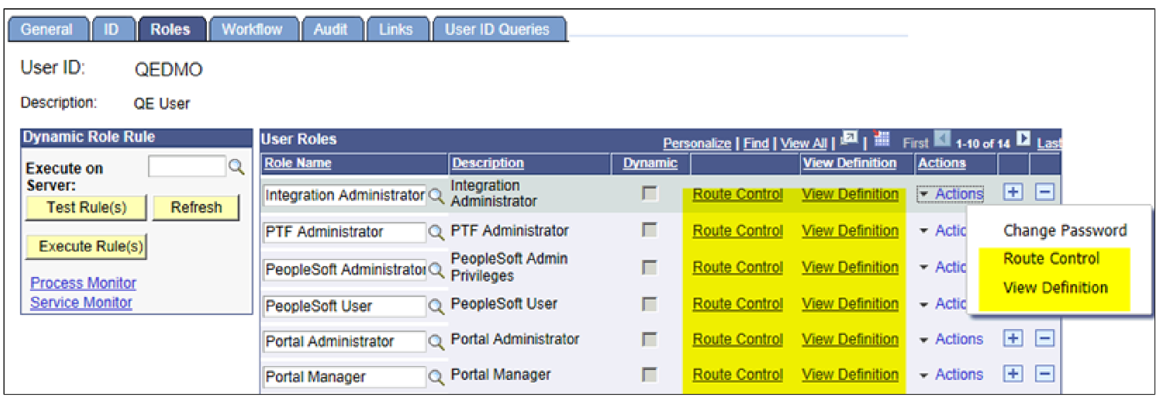
This example illustrates the display of FieldChange event actions in the Actions menu.



FieldChange Event Action Shown in Page and Related Action Menu

In this example, the Display in Related Action Menu and the Show in Page check boxes are selected.

This example illustrates the display of FieldChange event actions on a page and in the Actions menu.



Configuring Display of a Single Action on a Page

When a related action widget includes only a single action for a user, you can configure the display of that single action by selecting either action widget, link, or button.

To configure the display of single action on a page, in Application Designer, access the Push Button/ Hyperlink Properties dialog box, and in the Action Widget Settings group box, select an option (Action Widget, Link, or Button) from the Display Single Action Menu as drop-down list.

Display of a Single Action on a Page

In this example, the single action of the action widget is shown as the View User Profile link on the page.

Chapter 21

Configuring Event Mapping

Mapping Application Class PeopleCode to Component Events

You can map application class PeopleCode programs to component level, page level, component record level, and component record field level events. This allows you to add custom PeopleCode programs to components without customizing the component definition in Application Designer. These custom PeopleCode programs can be configured to run before, run after, or override any PeopleCode program defined for the same event on the component definition.

The PeopleCode Editor provides an Event Mapping button in the Application Designer tool bar. The Event Mapping button is a visual indicator of custom PeopleCode programs mapped to events of a component, page, component record, or component record field.

Related Links

"Viewing Event Mapping References" (PeopleCode Developer's Guide)

Prerequisites

The Menu search definition (PTPORTALREGISTRY) must be deployed and indexed first before you can search for and select a content reference for configuration.

Event Mapping Configurations from PeopleTools 8.54 or 8.55

If you have directly migrated PeopleTools 8.54 or 8.55 event mapping configurations via an ADS project, these configurations have an instance ID of 0, which is invalid in the current release. You will receive an error message if you attempt to open and resave such configurations. To fix this issue, manually run the UPGPT859EVM Application Engine program, which is automatically run in a standard upgrade. After running the UPGPT859EVM program, review the log file to identify the new service IDs. With this information, you can run queries to determine which configurations have been updated.

For example, this log file indicates that one instance ID was updated to 998188404:

```
'UPGPT859EVM' Program Started (226,2)
.
Current Max Instance ID:998188403 (240,4272)
Message Set Number: 240
Message Number: 4272
Message Reason: Current Max Instance ID:998188403 (240,4272)
.
Updated Row with Instance ID 0 to 998188404 (240,4291)
Message Set Number: 240
Message Number: 4291
Message Reason: Updated Row with Instance ID 0 to 998188404 (240,4291)
.
Updated respective map fields rows if Any, of Table:PSPTCS_MAPFLDS.
(240,4275)
Message Set Number: 240
Message Number: 4275
```



```
Message Reason: Updated respective map fields rows if Any, of
Table:PSPTCS_MAPFLDS. (240,4275)
.
Section: MAIN; Step: PTUPDINS; Updated 1 Rows of Tables:PSPTCSSRVCONF ,
PSPTCSSRVCONFLG. (240,4276)
Message Set Number: 240
Message Number: 4276
Message Reason: Section: MAIN; Step: PTUPDINS; Updated 1 Rows of
Tables:PSPTCSSRVCONF , PSPTCSSRVCONFLG. (240,4276)
.
'UPGPT859EVM' Program Completed (226,5)
Application Engine program UPGPT859EVM ended normally
```

Using this information, you could query PSPTCSSRVCONF, PSPTCSSRVCONFLG, and PSPTCS_MAPFLDS to determine which content reference (portal object ID) was affected. For example:

```
SELECT * FROM PSPTCSSRVCONF WHERE PORTAL_NAME='_PTCS_PTEVMAP' AND PTCS_INSTANCEID I⇒

N (998188404);
```

Related Links

"Working With the Portal Registry Search" (Search Technology)

Managing PeopleCode Event Mapping

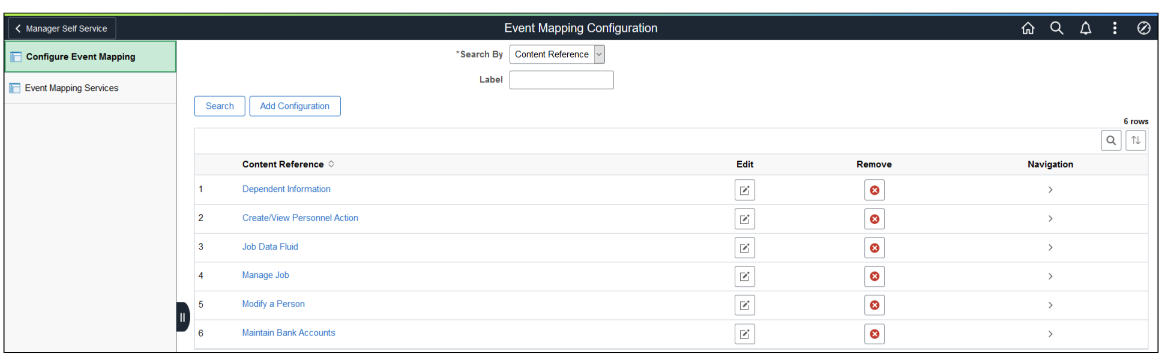
Use the Event Mapping Configuration search page to manage which content references have associated PeopleCode application package programs.

Note: Event mapping is restricted to content references for registered components only.

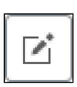
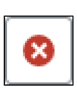
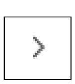
Navigation:

PeopleTools > **Portal** > **Event Mapping** > **Event Mapping Configuration**.

This example illustrates the fields and controls on the Event Mapping Configuration search page. You can find definitions for the fields and controls later on this page.



You can search for existing configurations by content reference label or by application package or class configured on the content reference.

Field or Control	***Description***
Search By	Select either: • *Content Reference* to search by content reference label. • *Application Class* to search by application package or class.
Label	When *Content Reference* is selected as the search type, enter a string or partial string to search on. The search results include matches on the content reference label or description.
Root Package IDPathClass ID	When *Application Class* is selected as the search type, enter a portion of or the fully qualified path to the application class: • **Root Package ID**: Select or enter the name of the application package that contains the application class. • **Path**: Select or enter the names of each sub-package in the application class hierarchy that define the location of the application class. Separate sub-package names by a colon. If the class is defined in the top-level application package, enter or select the colon. • **Class ID**: Select or enter the name of the application class.
Search	Click the Search button to execute the search.
Add Configuration	Click the Add Configuration button to access the Event Mapping Configuration page to search for a content reference and then add a new event mapping configuration. See Configuring Event Mapping.
Content Reference	Click a content reference link to access the actual transaction page, which allows you to test any configured PeopleCode programs.
	Click an **Edit Configuration** button to modify an existing event mapping configuration. See Configuring Event Mapping.
	Click a **Remove Configuration** button to remove an existing event mapping configuration.
	Click a **Show Navigation Path** button to display the full menu navigation path to the content reference.

Configuring Event Mapping

Use the Event Mapping Configuration page to map application class PeopleCode to component level, page level, component record level, and component record field level events.

Navigation:

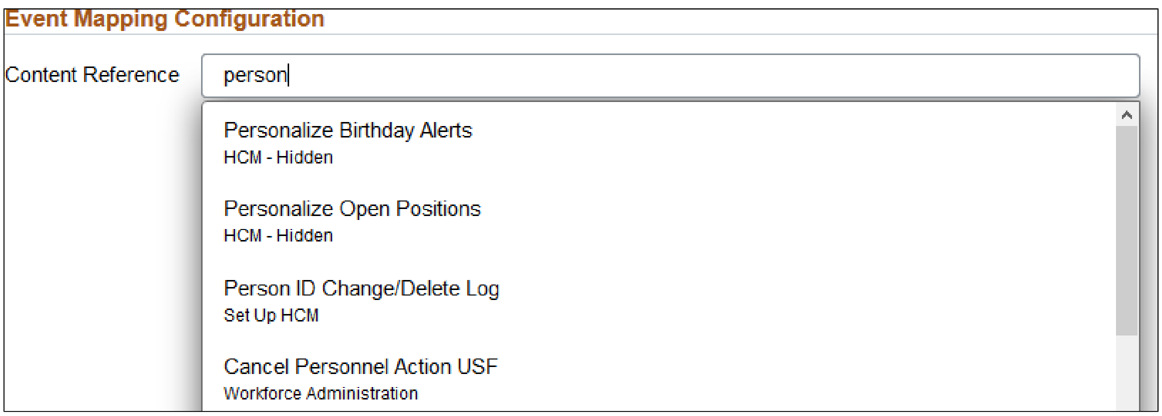
On the Event Mapping Configuration search page:

- Click the Add Configuration button to search for a content reference and then add a new event mapping configuration.
- Click an **Edit Configuration** button to modify an existing event mapping configuration.

Searching for a Content Reference

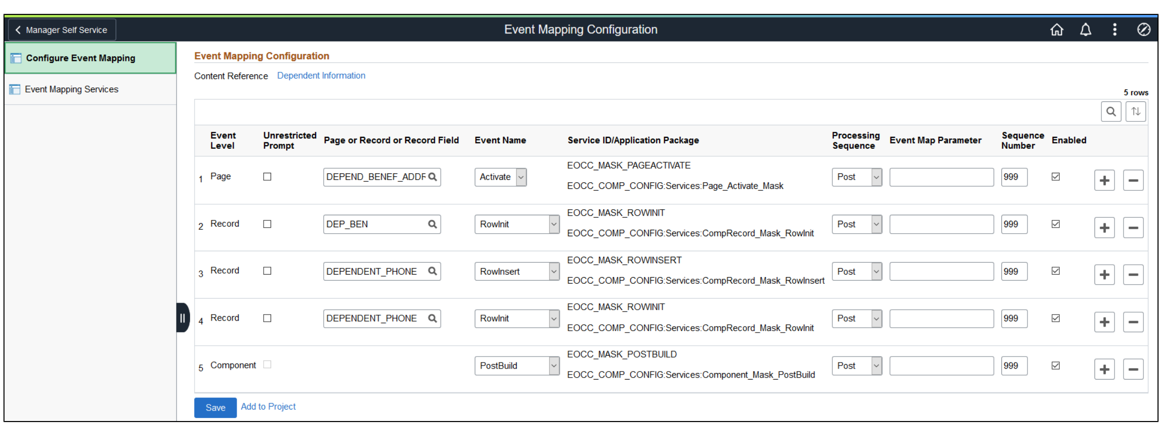
Note: Event mapping is restricted to content references for registered components only. Therefore, search results are filtered to include registered components only.

After clicking the Add Configuration button, enter a search term or phrase. The search results are updated as you type. Select a content reference from the search results to add a new event mapping configuration.



Completing the Configuration

This example illustrates the fields and controls on the Event Mapping Configuration page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Event Level	Select an option to specify the event level: • *Component* • *Page* • *Rec Field* • *Record*
Unrestricted Prompt	Select the unrestricted prompt *only when* the item for which you are searching is not found on a page definition that is explicitly included in the component definition. This allows an unrestricted search of all matching definitions within the system. Use the unrestricted prompt only when: • Searching for secondary pages invoked by the main page or by a subpage at any level within the component. • Searching for records, derived/work records, or record fields defined on any subpage or secondary page at any level within the component. **Note:** If you return to this configuration, the unrestricted prompt check box is no longer selected. However, the actual configurations are retained. This allows you select the unrestricted prompt again if you need to reconfigure these items in an unrestricted manner.
Page or Record or Record Field	Select the page name, record name, or combination of record name and field name corresponding to the event level selected previously. If *Component* is selected as the event level, this field is not displayed and no selection is required.

Field or Control	*Description*
Event Name	• At the component level, an application class-based service can be associated with one of the following PeopleCode events: ◦ PostBuild ◦ PreBuild ◦ SavePostChange ◦ SavePreChange ◦ Workflow • At the page level, an application class-based service can be associated with the Activate event only. • At the component record level, an application class-based service can be associated with one of the following PeopleCode events: ◦ RowDelete ◦ RowInit ◦ RowInsert ◦ RowSelect ◦ SaveEdit ◦ SavePostChange ◦ SavePreChange ◦ SearchInit ◦ SearchSave • At the component record field level, an application class-based service can be associated with one of the following PeopleCode events: ◦ FieldChange ◦ FieldDefault ◦ FieldEdit ◦ SearchInit

Field or Control	*Description*
	• SearchSave
Service ID/Application Package	Select an application class-based related content service definition. **Note:** The list of application class-based services is filtered to include only those that have implemented the PT_RCF:ServiceInterface base class.
Processing Sequence	Select the processing sequence with respect to any PeopleCode program defined for the same event on the component definition: • *Override* – Select to *override* any PeopleCode program defined for the same event on the component definition. • *Post* – Select to execute the mapped application class *after* any PeopleCode program defined for the same event on the component definition. • *Pre* – Select to execute the mapped application class *before* any PeopleCode program defined for the same event on the component definition.
Event Map Parameter	Optionally, specify a string value to selectively invoke a specific block of code within the application class. This allows you to use a single application class in multiple configurations separating each bit of functionality into separate blocks. This application class must implement the eve_execute method in addition to an empty execute method. Within the eve_execute body, conditional logic must evaluate the event map parameter and execute the appropriate block of code for each valid parameter value. See the following PeopleCode example.
Sequence Number	Specify a processing sequence when more than one custom application class program is mapped to the same event.
Enable	Deselect this check box to disable execution of the application class, but retain the event mapping configuration.
Save	Click the Save button to save the configuration.

Field or Control	*Description*
Add to Project	Click the Add to Project link to access the Add To Project page to insert this event mapping configuration into an ADS project. See "Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide) for more information.

PeopleCode Example for Multiple Events in a Single Application Class

The following application class presents an example of how eve_execute could be implemented:

```
import PT_RCF:ServiceInterface;

class App_Class1 implements PT_RCF:ServiceInterface
   method execute();
   method eve_execute(&str As string);
end-class;

method execute
   /+ Extends/implements PT_RCF:ServiceInterface.execute +/
end-method;

method eve_execute
   /+ &str as String +/
   /+ Extends/implements PT_RCF:ServiceInterface.eve_execute +/
   /* &str variable will have the Event Map Parameter that was */
   /* set in the configuration page for that particular event  */
   /* that is being triggered.                                 */
   Evaluate &str
   When = "1"
      /* Code block for PreBuild event */
   When = "2"
      /* Code block for PostBuild event */
   When-Other
      /* Code block to handle an invalid event map parameter  */
   End-Evaluate;
end-method;
```

Configuring Global Event Mapping

Global event mapping enables you to map application class PeopleCode programs to the PostBuild event of all components at a global level.

Global events are triggered after any existing PRE/OVERRIDE/POST event mapping configurations associated with the PostBuild event of the target components are completed.

Currently, PeopleTools does not deliver a configuration page to define global events. Global events can be configured using the delivered methods of the PTCS_GLOBALEVENTMAPPING application package, such as CreateGlobalEventMappingConfig and DeleteGlobalEventMappingConfig.

For more information on the delivered methods, see "GlobalEventMapping Class Methods" (PeopleCode API Reference).

Considerations for Using Global Event Mapping

When you plan to perform event mapping on a global level, you should consider the following:

- You must ensure that global event mapping does not impact the loading of components from performance and error handling perspectives.
- You should ensure that errors are handled gracefully.
- You should be aware that if a page has multiple components, component-based grouplets, component-based pagelets, or component-based modals and so on, the configured global events run on each of the loaded components. Therefore, your PeopleCode logic should conditionally identify or filter the components, component-based grouplets, component-based pagelets, or component-based modals to ensure that the required logic is run only for the required components.

  For example, if a page contains a target component and a component-based grouplet, and you want to run it only for the target component, then your PeopleCode logic should perform a conditional check to exclude the component-based grouplet.

Prerequisites

Before you create a global event mapping configuration using the CreateGlobalEventMappingConfig method, you must create a related content service definition.

See Creating Service Definitions for Event Mapping for more information.

Verifying the Global Event Mapping Configuration

The following database tables – PSPTCS_SRVCFG and PSPTCSSRVCONF – store the event mapping configuration data. After you create a global event mapping configuration using the CreateGlobalEventMappingConfig method, you can verify the configuration using the following SQL statements:

- `SELECT * FROM PSPTCS_SRVCFG WHERE PORTAL_OBJNAME LIKE 'PTCS_EVMAP_GBL_%'`
- `SELECT * FROM PSPTCSSRVCONF WHERE PORTAL_OBJNAME LIKE 'PTCS_EVMAP_GBL_%'`

Warning! Oracle recommends that you do not directly modify these database tables containing configuration data using SQL.

You can identify the global event mapping configuration by the data contained in the PORTAL_OBJNAME and the PTCS_PROCSEQ columns of these tables.

- In the PORTAL_OBJNAME column, identify the configuration name specified in the CreateGlobalEventMappingConfig method. The global event mapping configuration name is prefixed with PTCS_EVMAP_GBL.

  See “CreateGlobalEventMappingConfig” (PeopleCode API Reference).
- In the PTCS_PROCSEQ column, the global event mapping configuration is identified by the term GBL.

Creating Service Definitions for Event Mapping

To use an application class for event mapping, you must create a related content service definition based on the application class. Always use the Service Definitions page to create service definitions for event mapping.

Note: While you can still create generic application class service definitions using the Define Related Content Service page, additional options—for example, valid from and to dates—are presented there that are ignored when the service definition is used for event mapping. However, you can use the Define Related Content Service page to add a specific service definition to an ADS project. See Define Related Content Service Page for more information.

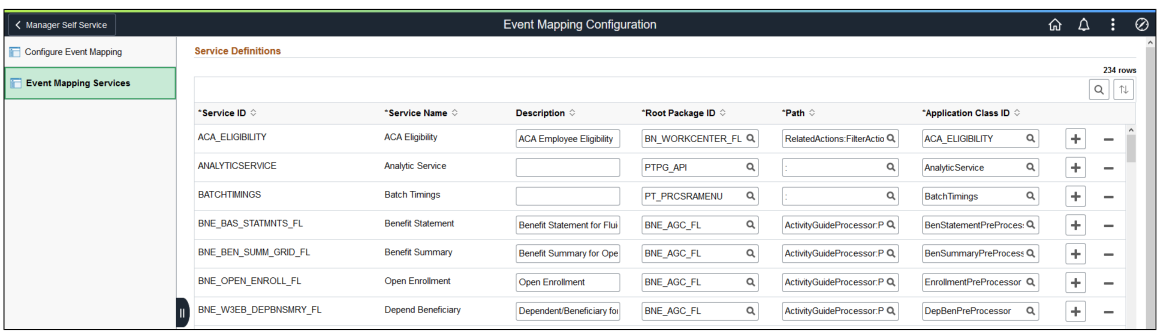
Navigation:

1. **PeopleTools** > **Portal** > **Event Mapping** > **Event Mapping Configuration**.

2. Select the Event Mapping Services link in the left panel.

Use the Service Definitions page to create a simple service definitions based on an application class.

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



Field or Control	***Description***
Service ID	Enter a unique identifier for the service definition.
Service Name	Enter a short name for the service, which will be also be displayed as search criteria when selecting service definitions for event mapping.
Description	Enter a description for the related content service.
Root Package ID	Select or enter the name of the application class package that you want to use to define the related content service.

Field or Control	***Description***
Path	Select or enter the names of each subpackage in the application class hierarchy that define the location of the application class that defines the service. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Application Class ID	Select or enter the name of the application class that defines the related content service. **Note:** The application class must implement the PT_ RCF:ServiceInterface base class.

Chapter 22

Creating and Configuring WorkCenter Pages

Understanding WorkCenter Pages

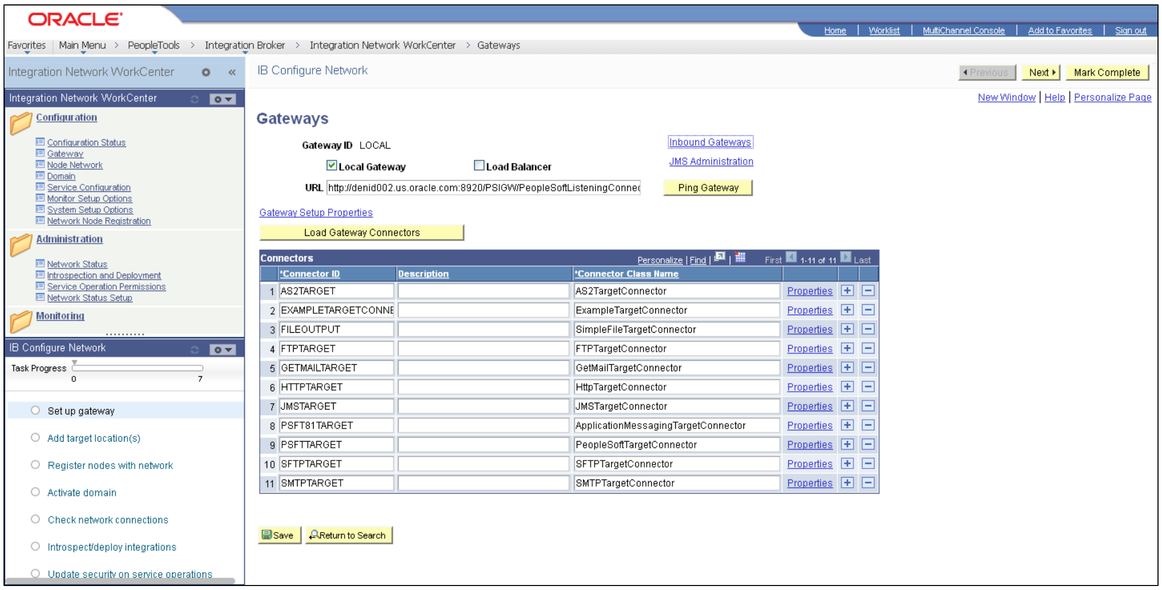
WorkCenter pages (or WorkCenters for short) are a type of configurable PeopleSoft classic page that enable you to offer access to related transactions, analytics data, query results, and other PeopleSoft resources from one central location. You can organize the elements on WorkCenter pages based on processes, procedures, roles, or other business requirements that are specific to your organization. When carefully designed, WorkCenter pages bring together the PeopleSoft elements that your users access most when performing the tasks that comprise their primary responsibilities. As a result, WorkCenter pages can promote productivity by minimizing the time that users spend navigating between tasks.

WorkCenter pages can bring these different elements together because WorkCenter pages use a classic portal template that provides flexibility in the way that you access classic components and pages and also in the way the application displays these elements. The WorkCenter portal template differs from the standard portal template in two primary ways:

- The WorkCenter portal template partitions the browser window into two frames: an adjustable pagelet area and target a content area.
- The two areas of the WorkCenter portal template are designed to interact by passing contextual information back and forth by using an event-driven model. The actions in the pagelet area determine the content in the other.

When users access WorkCenter pages, they are presented with pagelets that contain links to components, analytics, query results, and other PeopleSoft resources. The WorkCenter pagelet area is designed to extract and leverage values that are necessary to complete the transaction in the target area. When users click a link in the pagelet area, it triggers the appropriate transaction, and passes values from the pagelet area so that the application can display the appropriate content in the target content area or in a new browser window.

The following example illustrates the main parts of a WorkCenter page. On the left, the pagelet area displays two pagelets configured to display in this WorkCenter. On the right, the target area displays the current transaction page as selected in a WorkCenter pagelet. The standard portal header, including the drop-down menu and navigation breadcrumbs, is displayed at the top of the WorkCenter page.



WorkCenter-Like Functionality in Fluid Applications

WorkCenters are a classic-only feature delivered in PeopleTools. However, several PeopleTools features allow you to create WorkCenter-like functionality in fluid applications.

Fluid master/detail components provides an approach for providing access to pages spanning multiple components, aggregating the navigation to the various pages all within a single master/detail component. Similar to classic WorkCenters, the pages accessible from within master/detail components are intended for a particular role, providing quick access to the components and pages used most frequently.

Fluid navigation collections allow you to display navigation collection links in a fluid wrapper. Fluid navigation collections can consist entirely of classic content references, entirely fluid content references, or a combination of both.

Related Links

“Working with Master/Detail Components” (Fluid User Interface Developer’s Guide)
Fluid Navigation Collections

The Pagelet Area

The WorkCenter pagelet area consists of three parts:

- The header.
- The pagelet area.
- The footer.

In the header, the title, the Home button, and the pagelet area controls drop-down menu are visible. You click the Home button to return to the WorkCenter starting page. You click the pagelet area controls drop-down menu to access up to three links that enable you to reload, personalize, and configure the WorkCenter page.

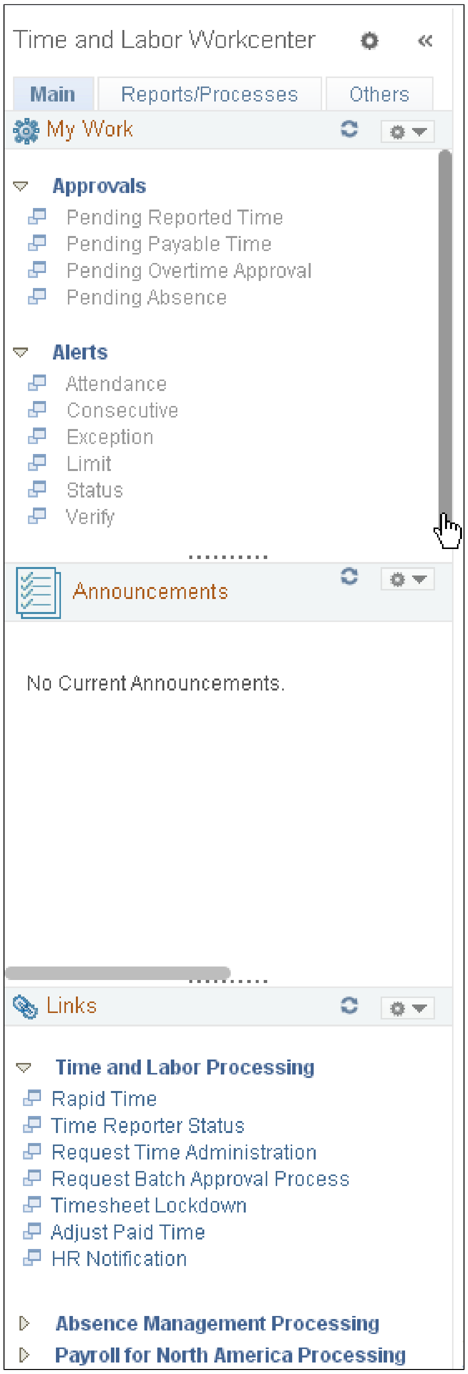
Note: When the WorkCenter is created from an existing menu item, the default title of the pagelet area is the same as the label for the menu item. However, when the WorkCenter is created from a link to a menu item, the title of the pagelet area is Task Panel.

The pagelets that you assign to the WorkCenter page appear in the main body of the pagelet area. Each pagelet you assign to the area is a self contained unit that you control independently of other pagelets. Pagelets appear one above the other, either in an expanded or collapsed state based on your configuration. Expanded pagelets divide the available space evenly until the user resizes a pagelet by dragging the pagelet handle. The system automatically adjusts the size of remaining pagelets to fill the available space.

At the bottom of the pagelet area, the **More** drop-down menu appears in the footer. You can configure the WorkCenter page to *hide* any pagelet that is minimized instead of collapsing the pagelet into its header. The hidden pagelet is minimized to the **More** menu. When you click the drop-down button, the titles of the minimized pagelets appear in a list. Select a pagelet from the list and the systems restores the pagelet to its original size and positions it at the bottom of any other expanded pagelets in the pagelet area.

The pagelet area is adjustable in width and its configuration is flexible. You can organize the pagelet area by creating pagelet groups. Each group appears as a separate tab. If you configure multiple pagelet groups, a tab appears at the top of the each page to identify the pagelet group and provide a familiar vehicle for accessing the pagelets. You can define options such as the initial size of the pagelet area, the pagelet group order, and the pagelet group that appears by default when you access a WorkCenter page.

The following example of a WorkCenter pagelet area shows three, tabbed pagelet groups:



After creating pagelet groups, you assign pagelets to the groups. You specify options such as the order of the pagelets, which pagelet receives focus by default, whether the pagelet is collapsed or expanded as its initial state, and the maximum number of pagelets that remain visible when in a minimized state. The pagelet area also supports user personalizations to the extent that you enable them.

Generally, pagelets that you assign to WorkCenter pages are narrow format pagelets that are specifically designed to appear in the resizable pagelet area. The pagelet area supports homepage and template pagelets that are registered in either the Pagelets or the Template Pagelets folder in the portal registry.

Note: You access these folders by selecting **PeopleTools** > **Portal** > **Structure & Content** and then clicking the **Portal Objects** link.

The pagelet area supports iScript-based pagelets, component-based pagelets, and even external URL-based pagelets. You can use Pagelet Wizard to create and publish iScript-based template and homepage pagelets such as those that use navigation collections and queries as data sources. You can also manually create and register component-based or other pagelets.

Related Links

Understanding WorkCenter Pagelets

The Target Content Area

The primary purpose of the target content area is to display the transaction page, query results, analytics, or other resources that the application user requests as the result of clicking a link in the pagelet area. You use target content area to search, view, edit, and complete your transactions. The target content area can display any type of content that a standard PeopleSoft transaction page can display. In addition, the target content area can display a special type of page known as a WorkCenter dashboard, which allows you to incorporate dashboard functionality into your WorkCenters.

You can also set up and configure pages and pagelets to use interwindow communication, which enables pagelets to communicate in any direction: from pagelet area to target content area, from target content area to pagelet area, and even between elements that occupy the same section.

You can target resources to appear not only in the target content area, but to a new browser window or to a current browser replacing the entire WorkCenter page with the resource.

Example of Setting the Target to Open in a New Window

You can set the target to open in a new window when you need to view both the current page and the new page simultaneously. For example, an accounts payable representative is completing an invoice and needs to view customer history at the same time. You could design a pagelet so that the link opens the Customer History page in a new window.

Example of Setting the Target to Replace the Current Window

You can set the target to replace the current window entirely. For example, an account representative is reviewing her largest account and then wants to view a dashboard that shows all accounts in relation to one another. You could design a pagelet with a link that opens the Accounts Dashboard and replaces the current window.

WorkCenter Page Layout Templates

WorkCenter pages use an iframe-based portal template that supports four sections:

- The portal header and drop-down style menu.
- The pagelet area.
- The target content area.

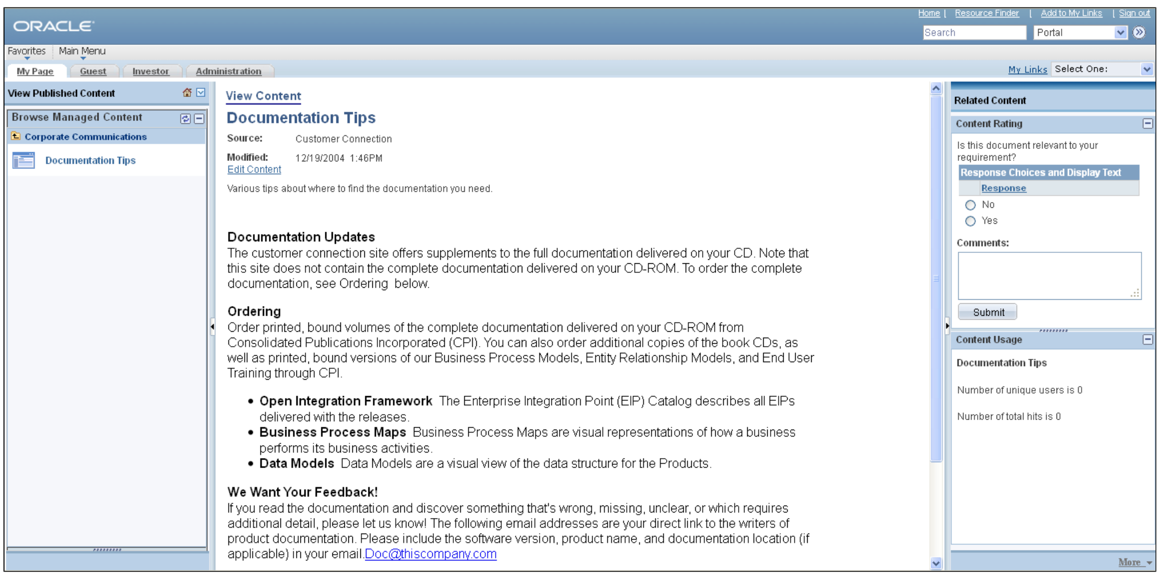
- The related content frames (optional).

The portal header and drop-down style menu, when enabled, appear at the top of the page above the pagelet and target content areas. The adjustable pagelet and target content areas occupy the middle of the page. The related content frames appear either below or beside the target content area when the current transaction page has related content configured.

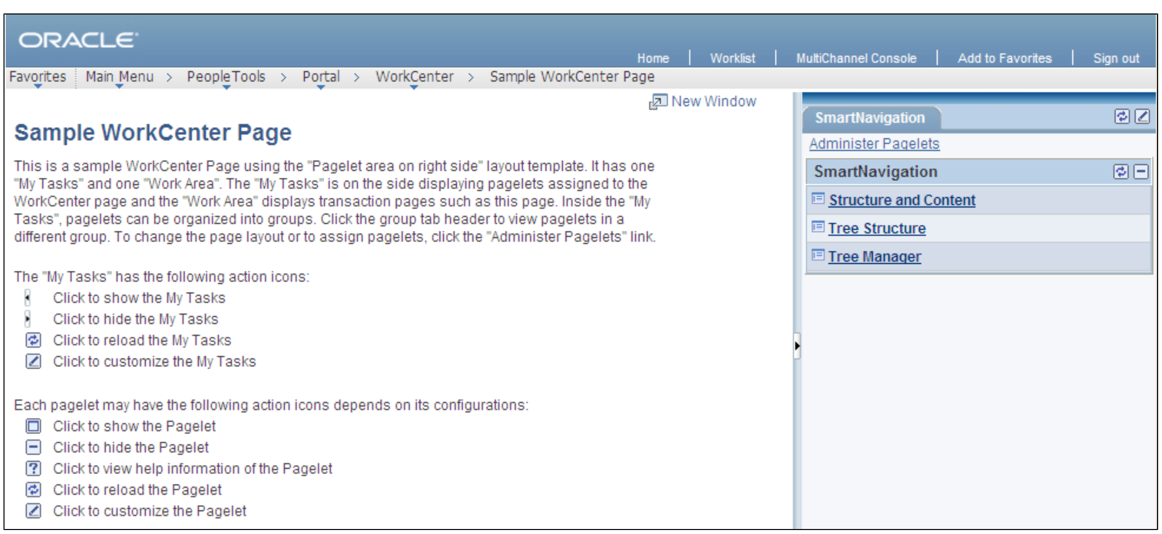
You can change the layout of WorkCenter pages by using page layout templates. Page layout templates use Application class PeopleCode to determine the location of the pagelet area and target content area. WorkCenter pages deliver application class PeopleCode that renders the pagelet area on either the left side or the right side of the page.

WorkCenter Page Layout Template Examples

In this example, which uses the default WorkCenter page layout template, you see the drop-down menu, the pagelet area on the left, and the Related Content frame to the right of the target content area:



In this example, you see the pagelet area on the right side of the WorkCenter page as specified by a custom page layout template:



Related Links

Defining WorkCenter Page Layout Templates

Additional Considerations When Customizing Pages

Page Design and Processing Considerations for WorkCenter Pages

WorkCenters have different characteristics from standard PeopleSoft components and pages. This section discusses design and processing differences that you should consider when you work with WorkCenters, WorkCenter starting pages, and pagelets.

The areas of the WorkCenter are organized in a left-to-right, or top-to-bottom fashion. The pagelet area on the left presents the navigation links for the WorkCenter. The target content area in the middle provides the work area of the WorkCenter. The optional Related Content frame, on the right or at the bottom, presents additional information related to the current transaction.

Starting Page Considerations

When you create a WorkCenter starting page, consider the following differences and page design recommendations, which accommodate those differences:

- WorkCenter starting pages are determined by the menu item associated with the content reference that you select when you create a new WorkCenter.

  While the page that appears initially in the target content area can be any page, you might want to start with a summary page showing the status of the tasks to be completed in the WorkCenter. The Unified Navigation WorkCenter employs this approach. Alternatively, you might want to display the first task to be completed. WorkCenters that have designated an activity guide pagelet as the in-focus pagelet employ this approach. In addition, you could choose a WorkCenter dashboard as a starting page to display homepage pagelets on a special dashboard within the target content area.

- Typically, the component should consist of only one page, but no tabs or page links should appear on the starting page.

  Deselect the **Multi-Page Navigation** check boxes on the Internet tab.

- The elements that make up the tool bar and page bar are not applicable and should not appear on the WorkCenter starting page.

  Select the **Disable Toolbar** and **Disable Pagebar** check boxes on the Internet tab.

- WorkCenter starting pages generally require no save functionality.

  Select the **Disable Saving Page** check box on the Use tab.

- Do not use a search page as a WorkCenter starting page.

  If necessary, select a “dummy” record—one that has no keys or contains only one row of data—as the component search record on the Use tab.

  The INSTALLATION record is an example of a “dummy” search record.

Processing Considerations

Note these points about WorkCenter page processing:

- WorkCenter pages, like homepages and dashboards in your PeopleSoft applications, use deferred loading to load pagelets by default. When a WorkCenter page uses deferred loading, each pagelet appears independently from one another. When the data for the pagelet has been loaded, the pagelet appears. (This default behavior can be overridden be setting the PTPGLTDEFERLOAD content reference attribute to false for a specific pagelet.)
- WorkCenter pages and assigned pagelets support interwindow communication. Use the Interwindow Communication page to configure interactivity among pagelets that share a WorkCenter page.

  See Understanding Interwindow Communication.
- WorkCenter pages can process contextual parameters that you include in the WorkCenter page URL. The WorkCenter template accesses the parameters and passes them to the pagelets, which use the values to determine the records to retrieve and display. For example, you can create a WorkCenter page with pagelets that display customer information. You can pass the customer ID to the WorkCenter page URL as a context parameter so that all pagelets retrieve and display information about the same customer.

  The context parameter, CONTEXTPARAMS, must use the NAME:Value format and be passed as a query parameter string. Use this format when you build the WorkCenter page URL:

  ```
  http://myserver.example.com:8920/psp/ps/EMPLOYEE/EMPL/c/PTAL_ADMIN.PTAL_SAMPLE⇒
  _PAGE.GBL?CONTEXTPARAMS=USERPROFILE:VP1& CONTEXTPARAMS=EMPLOYEEID:1242.
  ```

  Using the previous example URL, the WorkCenter template retrieves the query string parameters, which it then sends to each pagelet in the following query string format ?USERPROFILE=VP1& EMPLOYEEID=1242.

  When you develop WorkCenter pagelets, use the **%Request.getParameter** PeopleCode method to retrieve and make use of the context parameter values.

Page Loading Sequence

The parts of a WorkCenter page are loaded in the following order:

1. Pagelet area.
2. Pagelets by group starting with the default pagelet group. Within each pagelet group, the pagelets are loaded in the personalized sequence order.
3. Target content area.

Other Considerations

When implementing and accessing WorkCenters, be aware of these additional considerations:

- Users who access WorkCenters must have permissions that grant access to all functions in the WEBLIB_PTAL web library. Either:
  - Add the PTPT1000 permission list to the user roles.
  - Add full access to all functions in the WEBLIB_PTAL web library to the user permission list.

- Users can create a favorite to the WorkCenter when on the WorkCenter starting page. If users navigate to other links within the WorkCenter and then attempt to create a favorite, an error message will be displayed:

```
A failure has occurred saving this favorite. Please contact your portal admini⇒
strator.
```

Additional Considerations When Customizing Pages

When you customize the look and feel of your PeopleSoft system, there are some additional considerations with respect to WorkCenters and WorkCenter pages.

If you make style customizations to the background color for pages that appear in the WorkCenter target content area, or if you make style customizations to the background color for WorkCenter pagelet area, or if you make style customizations to the background color for the portal homepage, you will want to make similar style changes to the following additional style classes:

- ptalPgltAreaResizeBar
- ptalPgltAreaHorizontalResize

In PeopleSoft Application Designer, make these changes in the active style sheet or to each of these style sheets:

- PTAL_SUBSTYLE_VL
- PTAL_SUBSTYLE_VL_SWAN
- PTAL_SUBSTYLE_VL_TANGERINE

Understanding WorkCenter Pagelets

You can create iScript-based and component-based homepage pagelets and template pagelets for use with WorkCenter pages. You use Pagelet Wizard to create iScript-based pagelets. Pagelet Wizard pagelets can be based on many possible data sources, such as activity guides, navigation collections, search records, HTML, and queries. You use PeopleSoft Application Designer to create component-based pagelets that are based on PeopleTools definitions. This section presents overviews of pagelets based on these PeopleSoft data sources:

- Activity guides
- Navigation collections
- PeopleSoft query
- Components

Related Links

Understanding Pagelet Wizard Data Sources
Developing iScript-Based Pagelets

Pagelet Development Overview

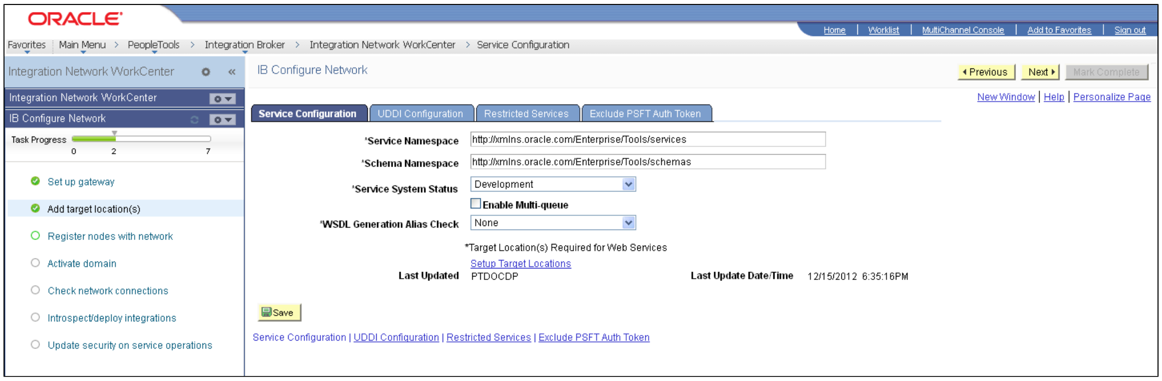
This table summarizes the steps to be performed and references the corresponding documentation that you should review for developing WorkCenter pagelets.

Prerequisite	***Cross-Reference***
Create an activity guide template and generate an activity guide pagelet.	See Creating and Maintaining Activity Guide Templates and Deploying Classic-Only Activity Guides.
Create navigation collections that you want to assign to the pagelet area.	See Creating and Maintaining Navigation Collections.
Create PeopleSoft query definitions that you want to use in the pagelet area.	See "Creating New Queries" (Query).
Create PeopleSoft page definitions, add and configure hyperlink page controls, and then define and register the components that you want to use as pagelets.	See "Understanding Page Design and Development" (Application Designer Developer's Guide). See Understanding Pagelet Development. See "Using Push Buttons and Links" (Application Designer Developer's Guide). See "Defining Components" (Application Designer Developer's Guide).
Use Pagelet Wizard to create and publish pagelets that use your navigation collections and query definitions as data sources.	See Understanding Pagelet Wizard. See Selecting a PeopleSoft Query Data Source. See Selecting a Navigation Collection Data Source.
Manually add component-based template pagelets to the template pagelets portal objects folder.	See Defining Content References.

Activity Guide Pagelets

Activity guides are typically deployed on a WorkCenter page, which is where they can be accessed by users. Prior to creating a WorkCenter page where the activity guide will be deployed, you must first use Pagelet Wizard to generate a pagelet for the pagelet area of the WorkCenter page. This pagelet will present the action items to the user in list form.

In this example, the IB Configure Network activity guide pagelet is displayed in the pagelet area of the WorkCenter page with the Service Configuration page in the target content area:



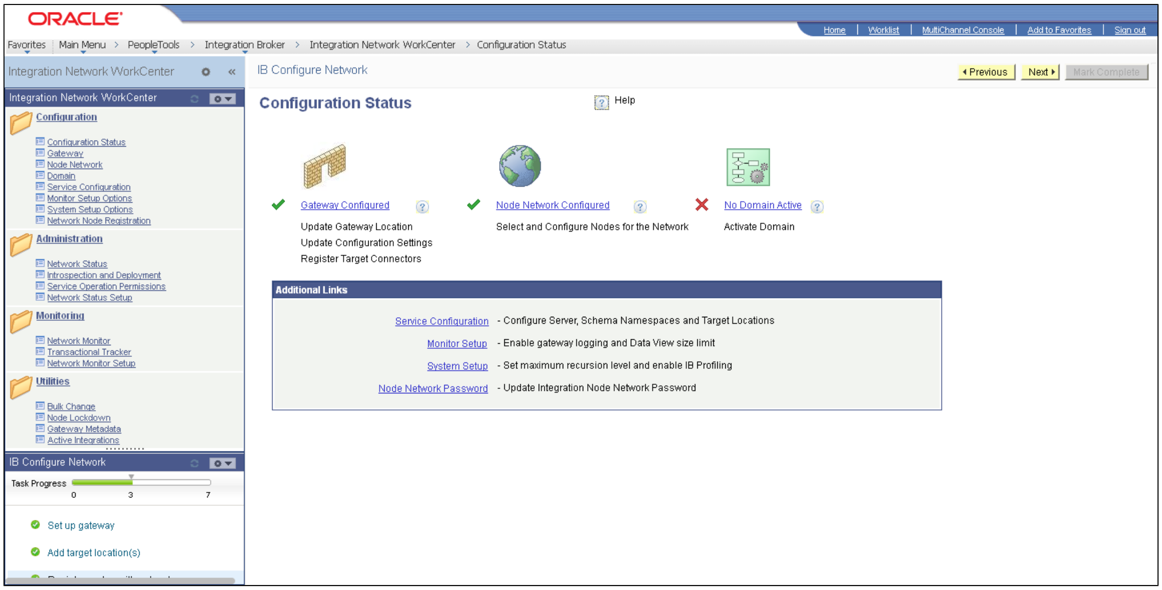
Depending on whether you want to tie the activity guide pagelet to a specific activity guide instance or to dynamically create the activity guide instance, there are three methodologies for generating the activity guide pagelet in Pagelet Wizard. For more information, see Deploying Classic-Only Activity Guides.

Navigation Collection-Based Pagelets

Navigation collections are portal objects that you use to create your own groupings of existing folders and content references, which you can then deploy or publish to different users or groups of users. Navigation collections enable you to create links to content that you already have in the portal registry and to organize the links in ways that specifically address your users' business needs.

Navigation collection-based pagelets appear in the pagelet area as folders and links. When users click a folder, the folder contents appear in the pagelet. When users click a link, the resource opens in the target content area by default. Because navigation collections are designed as alternate menu structures, you navigate through folders to find links, which generally reference components. When users click the link, the component search page appears in the target content area unless the search record is configured so that the search page is bypassed. In such cases, the component requires no search parameters and immediately opens in the target content area.

In this example, the Integration Network WorkCenter navigation collection pagelet is displayed in the pagelet area of the WorkCenter page with the Configuration Status page in the target content area:



Publishing Navigation Collection-Based Pagelets for Use With WorkCenter Pages

You create and maintain navigation collections using the navigation collections component. Using the pages in this component, you set the properties that determine the order in which the folders and links appear, and also the location where the target appears. If you decide to create pagelets based on navigation collections, you should review the link properties to ensure that neither replace the current page or open in a new browser window is selected. This will allow the content to open in the target content area, which is typically the desired behavior in a WorkCenter context.

You can use Pagelet Wizard to create pagelets based on any *published* navigation collection. At step two of the pagelet creation process in Pagelet Wizard, you must specify your navigation collection as the data source. Choose from any of your custom navigation collections or the PeopleSoft-delivered navigation collections. Continue through the process to step 6, where you specify *Homepage Pagelet* or *Template Pagelet* as a publishing option. When you save the pagelet, Pagelet Wizard creates the pagelet and you can then assign it to the pagelet area of a WorkCenter page.

Specific details for creating and publishing navigation collections and using Pagelet Wizard to publish them as pagelets are located in this documentation.

See Creating or Editing a Navigation Collection.

See Selecting a Navigation Collection Data Source.

See Step 6: Specifying Pagelet Publishing Options.

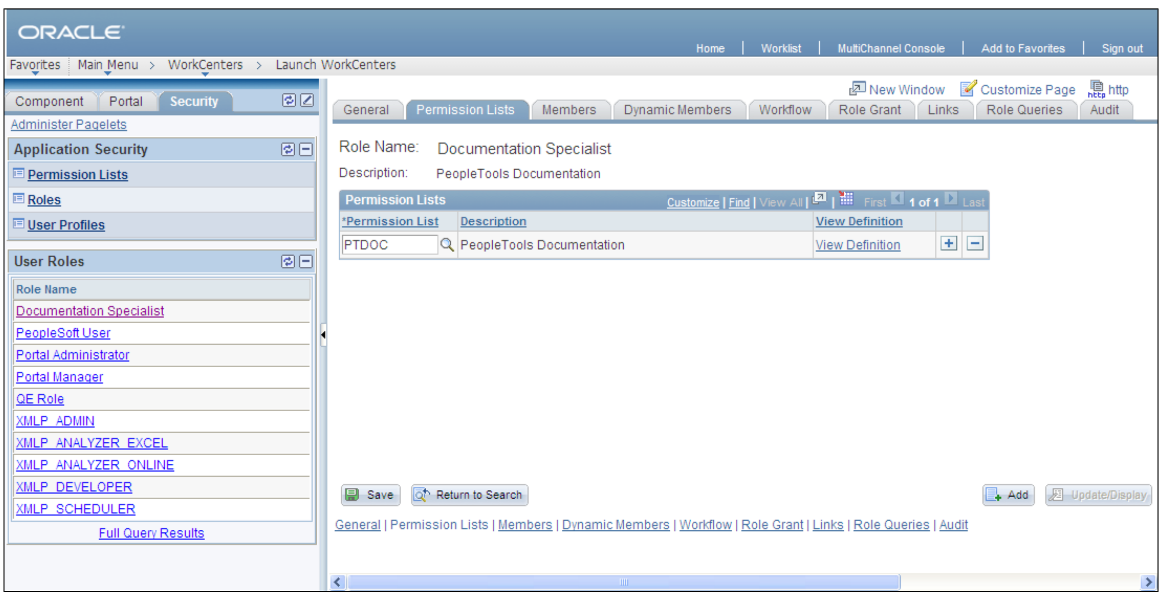
PeopleSoft Query-Based Pagelets

PeopleSoft Query is a reporting tool that enables you to extract your data by using visual representations of your PeopleSoft database, without writing SQL statements. The queries that you create can be as simple or as complex as necessary. You can design queries that return basic lists of static text. You can also use more advanced features, such as *drilling URLs*. By configuring drilling URLs and associating them with query fields, the values that the query retrieves appear as links, which you click to access both internal and external content.

Query-based pagelets can take advantage of queries, whether they are simple or complex. You might base a pagelet on a query that retrieves informational text to display to the user. However, query-based pagelets also have access to the more advanced query metadata, such as the drilling URLs. If you link fields in query-based template pagelets to these drilling URLs, the values that the pagelet retrieves appear as links as they do in PeopleSoft Query. Unlike PeopleSoft Query, however, you can configure the links to open the resource in the target content area of the WorkCenter page.

Because you can configure the links so that the associated resources open in the WorkCenter page target content area, query-based pagelets that are configured to display the retrieved values as active links are ideal for using with WorkCenter pages.

In the pagelet area of this WorkCenter page, the User Roles pagelet is based on the *PT_SEC_USER_ROLES* query definition, which returns the roles that are assigned to the current user. Each role name in the list is a link to the Roles component. When users click a role name such as *Documentation Specialist,* that role name value passes to the component as a search parameter. Using the value, the Roles component opens to the *Documentation Specialist* role.



Publishing Query-Based Pagelets for Use With WorkCenter Pages

You create and maintain queries using the Query Manager component.

You can use Pagelet Wizard to create pagelets based on any query. At step two of the template pagelet creation process in Pagelet Wizard, you must specify your query as the data source. At step 5, you use the link icon to access the Link Details page, where you can configure fields to use the drilling URLs and appear as links. At this step, you also specify whether a link appears in the target content area or a new window. Continue through the process to step 6, where you specify *Homepage Pagelet* or *Template Pagelet* as a publishing option. When you save the pagelet, Pagelet Wizard creates the pagelet and you can then assign it to the pagelet area of a WorkCenter page.

Specific details for creating and maintaining queries are located in *PeopleTools: Query*.

Specific details for using Pagelet Wizard to publish queries as template pagelets are located in this PeopleBook.

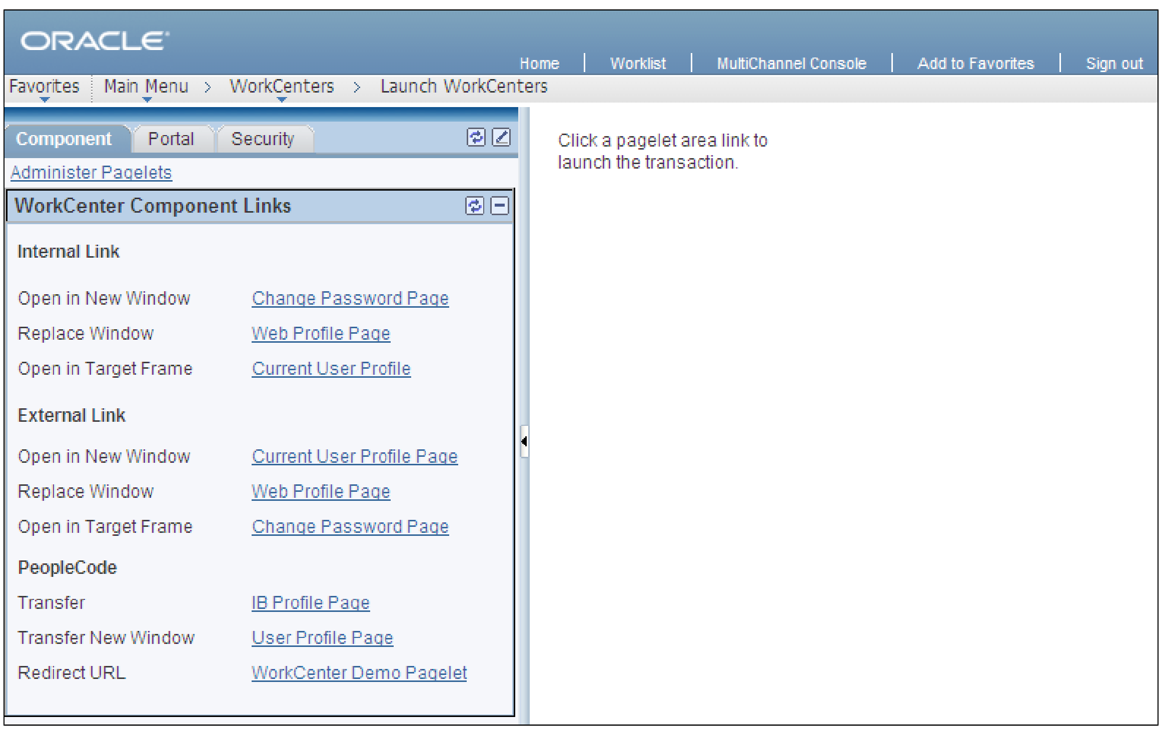
Related Links

"Defining Query Drilling URLs" (Query)
Selecting a PeopleSoft Query Data Source
Specifying Data Link Details
Step 6: Specifying Pagelet Publishing Options

Component-Based Pagelets

Component-based pagelets are PeopleSoft component definitions that you create and configure in PeopleSoft Application Designer and then register in the PeopleTools portal as pagelets. With minor differences, the process that you use to create, configure, and assemble the definitions that compose this type of component definition follows closely the application development process. While the *process* is similar to that which you use to create standard component definitions, some definition *properties* are different. In addition to these definition properties, other portal-related differences can exist depending on whether you use an existing component or create a new one.

This section discusses the important differences in settings between standard component definitions and components that you use in component-based pagelets. All examples in this section use an example component-based pagelet, titled *WorkCenter Component Links,* which you see in the pagelet area of this WorkCenter page:



Record Definition Settings

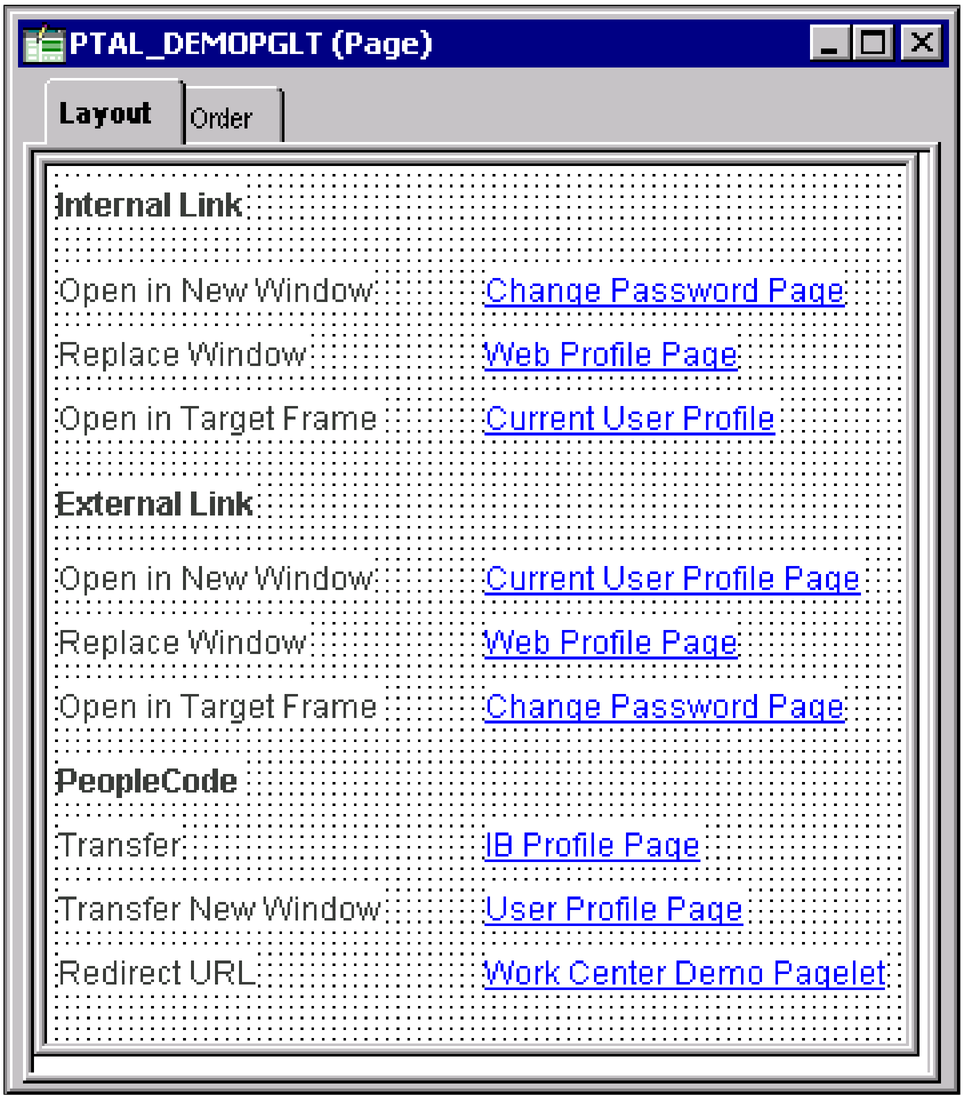
Record definitions that support component-based pagelets typically are not set to the SQL table record definition type because the pagelet area is not intended to write data to tables. Create or use record definitions with a record type of SQL View or Derived/Work.

See "Setting Record Properties" (Application Designer Developer's Guide).

Page Field Definition Settings

All pagelets should be designed to produce interaction between the pagelet area and the target content area. When defining pages that are part of component-based pagelets, the page developer can most easily facilitate this interaction by inserting push button/hyperlink page controls and then configuring properties that are inherent to that type of control. The developer can also write PeopleCode programs that accomplish the same pagelet area to content area interaction.

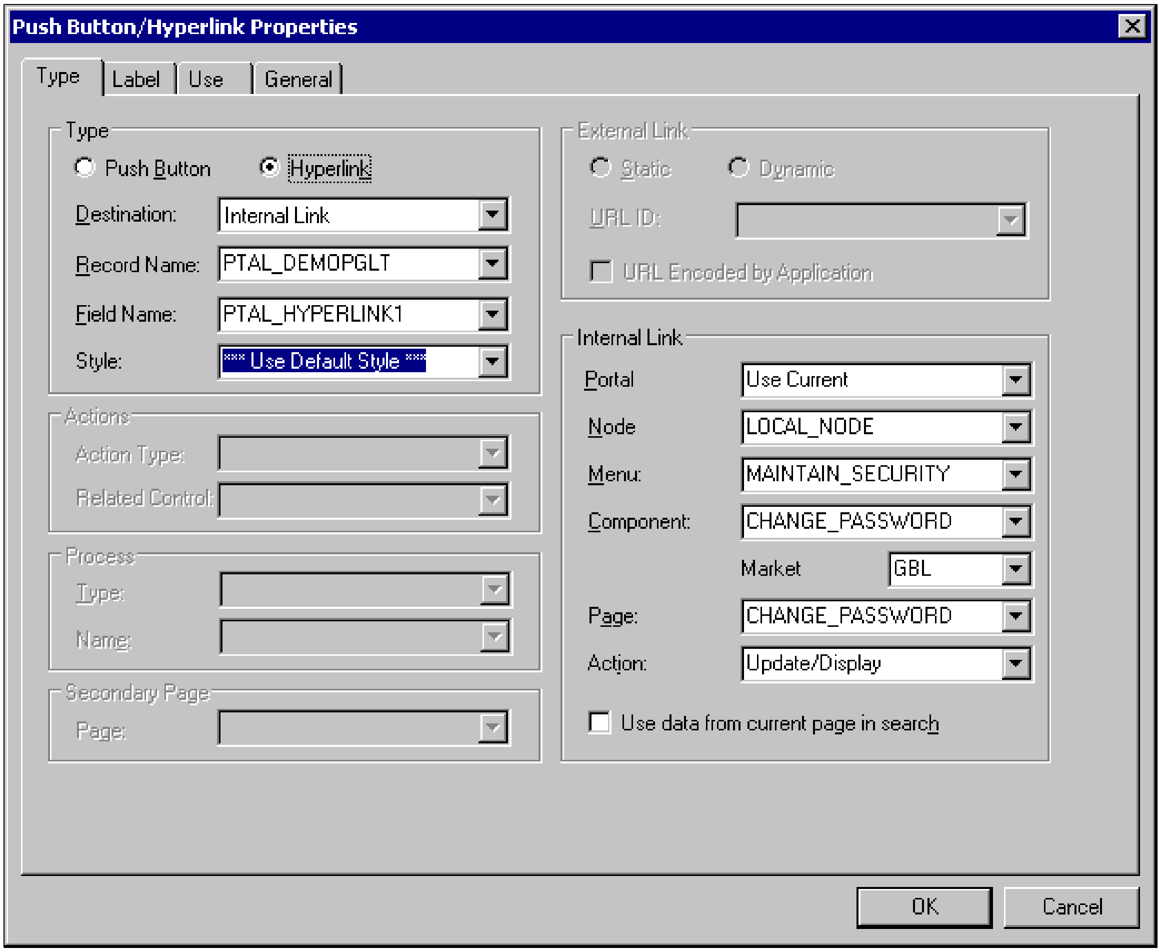
The WorkCenter Component Links pagelet is based on the PTAL_DEMOPGLT page definition, which includes nine push button/hyperlink controls:



See "Understanding Page Design and Development" (Application Designer Developer's Guide).

By setting the push button/hyperlink page control Type and Field Use properties at design time, the developer determines the resource that appears when the user clicks the link and also where that the resource appears, such as in the target content area, a modal window, or a new window. Secondary pages, PeopleCode, internal links and external links only are valid destination types for push button/hyperlink controls when you use them on component-based template pagelets.

The Push Button/Hyperlink **Type** properties specify an internal link type. The internal link properties specify the location of the CHANGE_PASSWORD page:



See “Using Push Buttons and Links” (Application Designer Developer’s Guide).

Component Definition Settings

Standard components and the components that you use as component-based pagelets have different purposes, which require that the developer configure the types of components differently. The following list describes the recommended property settings for components that you use as component-based pagelets on WorkCenter pages:

- WorkCenter pages should not generate a search page or start the search process when the user clicks the content reference to access the page. The user should immediately see the initial page.

  Select a “dummy” record—one that has no keys or contains only one row of data—as the component search record on the Use tab.

  The INSTALLATION record is an example of a “dummy” search record.

- Component-based pagelets generally require no save functionality.

  Select the **Disable Saving Page** check box on the Use tab.

- Typically, the component should consist of only one page, but no tabs or page links should appear on the pagelet.

  Deselect the **Multi-Page Navigation** check boxes on the Internet tab.

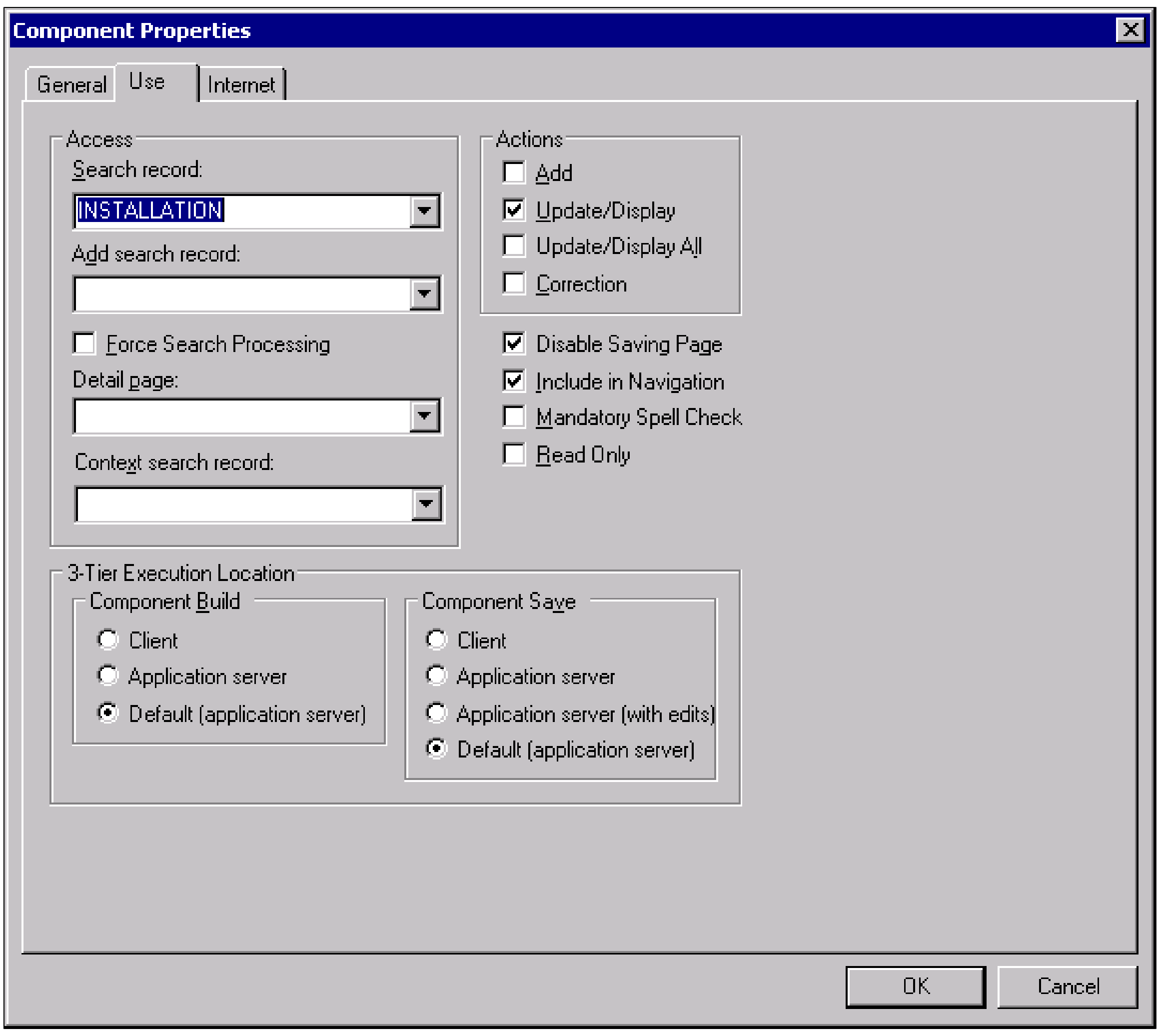
- The elements that make up the tool bar and page bar are not applicable and should not appear on the page.

  Select the **Disable Toolbar** and **Disable Pagebar** check boxes on the Internet tab.

Notice the settings of the PTAL_DEMOPGLT component in this example:

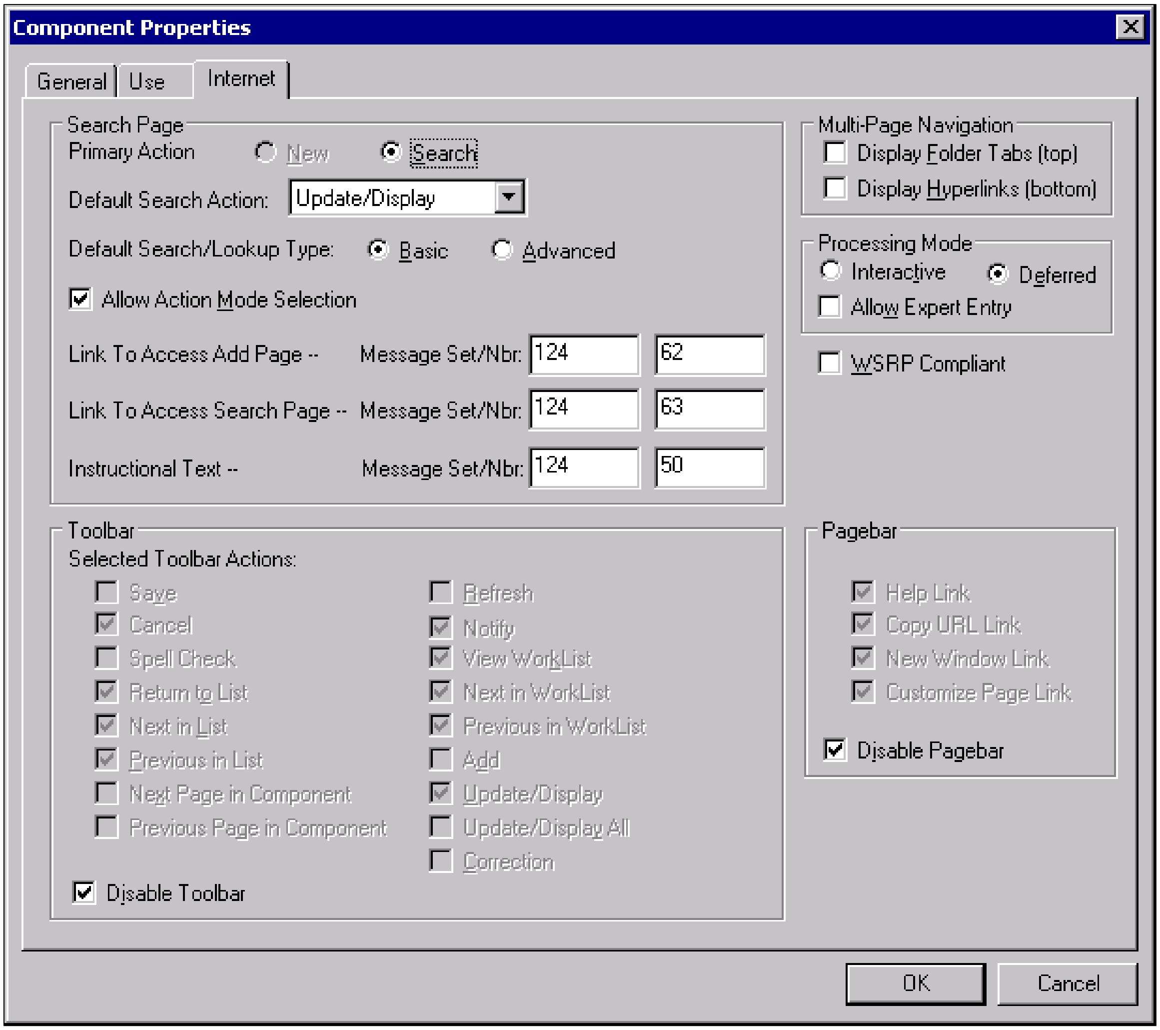
- The search record is *INSTALLATION* and the **Disable Saving Page** check box is selected:

  The following example shows the Use tab settings for the Component Properties dialog box for a component-based template pagelet:



- The **Multi-Page Navigation** check boxes are deselected and the Pagebar and the Toolbar are disabled:

The following example shows the Internet tab settings for the Component Properties dialog box for a component-based template pagelet:



See "Setting Component Properties" (Application Designer Developer's Guide).

Registering Components as Pagelets for Use as WorkCenter Pagelets

After creating the component definition, you must register the component as a homepage pagelet or template pagelet in the PeopleTools portal. The PeopleTools portal stores content references based on their URL information. As a result, the combination of elements that make up the URL of every content reference in a portal must be unique.

If a component is already registered as a content reference in the portal, then you must enter a “dummy” value in the **Additional Parameters** field when you register that same component as a template pagelet. Adding this field value ensures that the two content references are unique. A sample value that works well is *name1=value1,* as shown in this example:

See Defining Content References.

Related Links

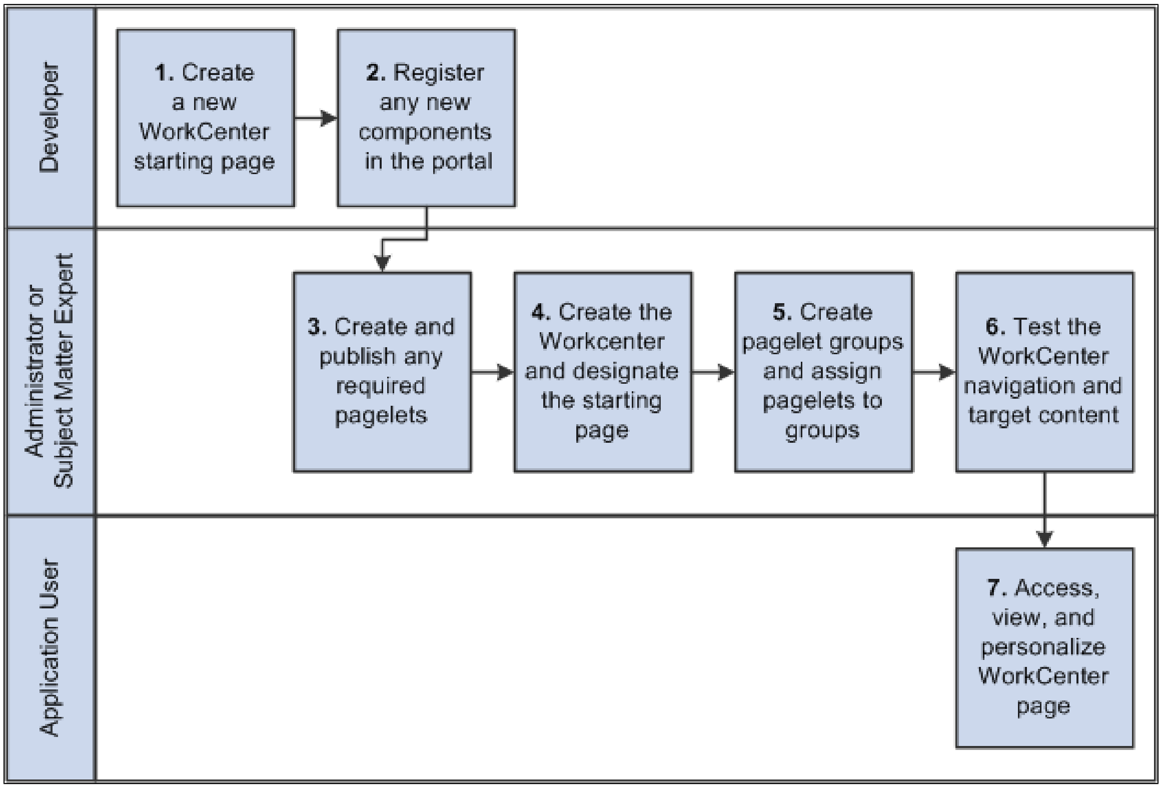
“Main Application Development Steps” (Application Designer Developer’s Guide)

Understanding the WorkCenter Development and Implementation Process

The WorkCenter development and implementation process consists of these steps:

1. Create a new WorkCenter starting page, which can be based on a:
   - Content reference
   - Content reference link
   - WorkCenter dashboard
2. If necessary, register any new components (and pages) in the portal.
3. Create and publish any required pagelets.
4. Create the WorkCenter and designate the starting page.
5. Create pagelet groups and assign pagelets to groups.
6. Test the WorkCenter page including navigation and generation of target content.
7. Once testing is complete, users can access and personalize the WorkCenter page for their use.

This diagram presents an overview of the process for developing, implementing, and using WorkCenter pages:



Managing WorkCenter Pages

This section discusses how to:

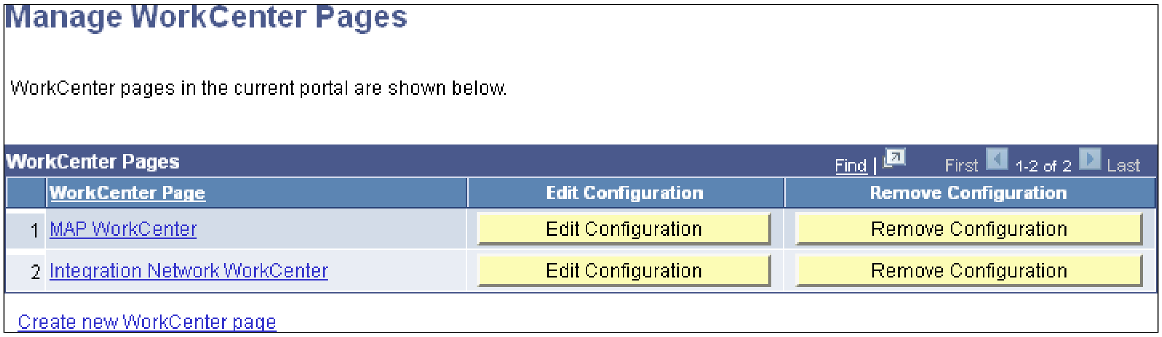
- Administer WorkCenter pages.
- Select the WorkCenter starting page.
- Administer WorkCenter starting pages.
- Configure WorkCenter pages.

Administering WorkCenter Pages

Access the Manage WorkCenter Pages page. (Select **PeopleTools** > **Portal** > **WorkCenters** > **Manage WorkCenter Pages**.)

Field or Control	***Description***
Create New WorkCenter Page	Click to access the **Select the WorkCenter Starting Page** page and begin the process of creating a WorkCenter page.

This example illustrates the fields and controls on the Manage WorkCenter Pages page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
WorkCenter Page	Displays the content reference name as an active link. Click to access the content reference and view the WorkCenter.
Edit Configuration	Click to access the **Configure WorkCenter Page** page.
Remove Configuration	Click to delete the WorkCenter page, all pagelet groups, and all pagelet configurations associated with the content reference. **Note:** Removing WorkCenter page configurations neither deletes the content reference upon which it is based nor does it delete the pagelets.
Create New WorkCenter Page	Click to access the **Create WorkCenter Page** page to begin the process of creating a WorkCenter by selecting a WorkCenter starting page.

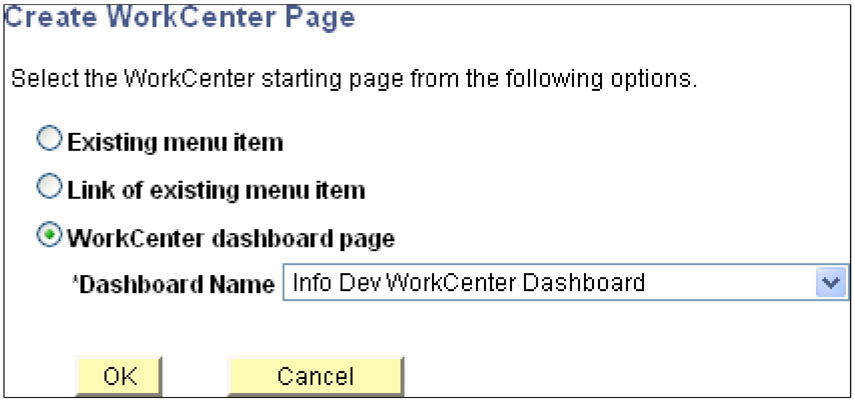
Selecting the WorkCenter Starting Page

To select the WorkCenter starting page:

1. Select **PeopleTools** > **Portal** > **WorkCenters** > **Manage WorkCenter Pages.**
2. Click the **Create new WorkCenter Page** link.

   The **Create WorkCenter Page** page is displayed:

This example illustrates the fields and controls on the Create WorkCenter Page page:



3. On the **Create WorkCenter Page** page, select the type of starting page:

   - **Existing menu item** – Select this option to *convert* an existing content reference into a WorkCenter.

     Important! When users navigate to the menu item for this content reference, they will now open the WorkCenter once it is converted in this manner.

   - **Link of existing menu item** – Select this option to *create a link* to an existing content reference. The link will be used as the WorkCenter starting page.

     Note: When users navigate to the menu item for this content reference, they will continue to open the content, and not the WorkCenter.

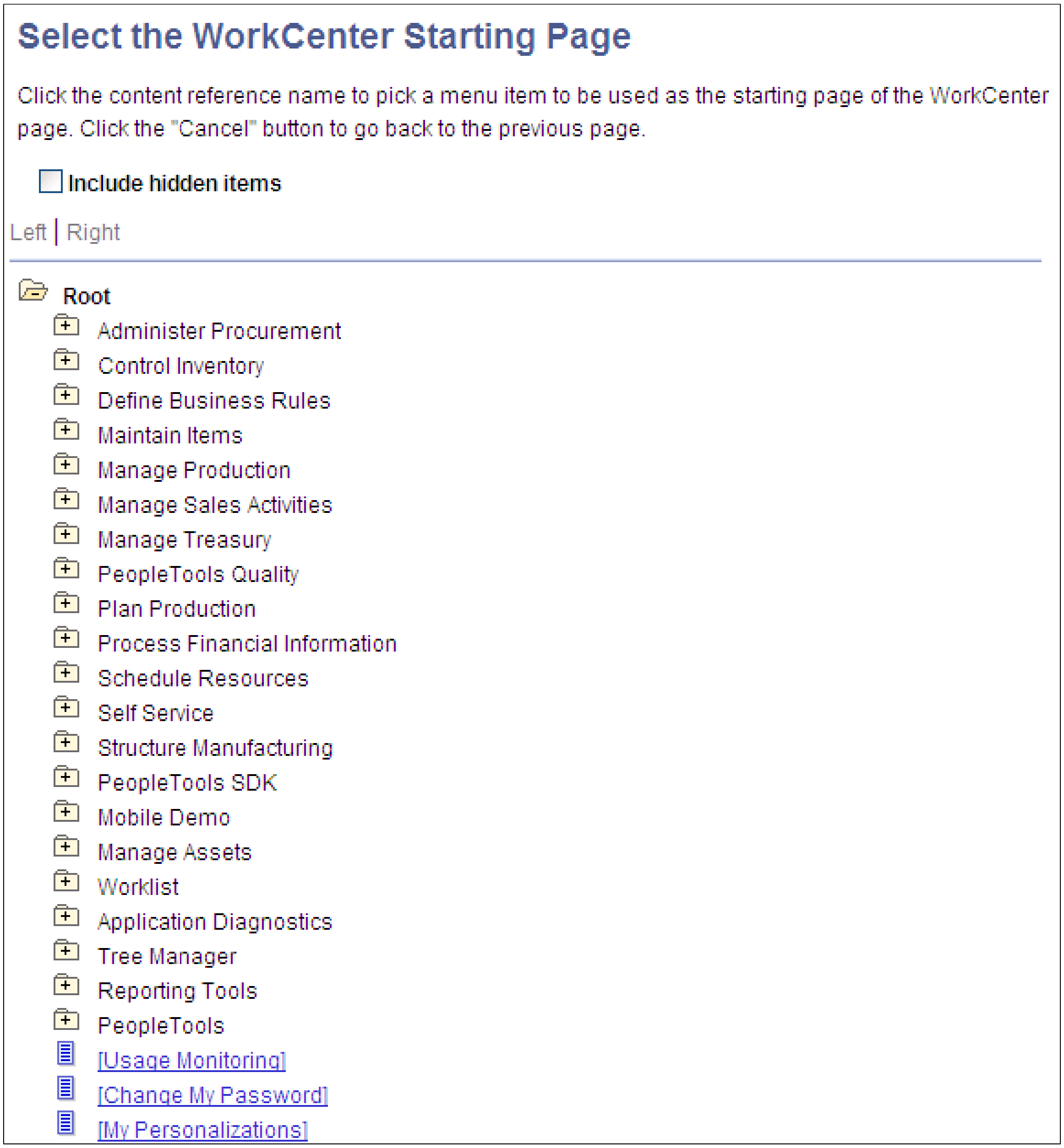
   - **WorkCenter dashboard page** – Select this option to use an existing WorkCenter dashboard page as the WorkCenter's starting page.

     Note: When no WorkCenter dashboards exist in the system, this option is not displayed. In addition, the **Dashboard Name** field is displayed only after you select this option.

     After selecting the WorkCenter dashboard page, continue by configuring the WorkCenter. See Configuring WorkCenter Pages

4. If you selected either **Existing menu item** or **Link of existing menu item**, continue by selecting the content reference that represents that menu item:

This example illustrates the fields and controls on the Select the WorkCenter Starting Page page:



5. Continue by:

   - If you selected **Existing menu item,** then configure the WorkCenter. See Configuring WorkCenter Pages.

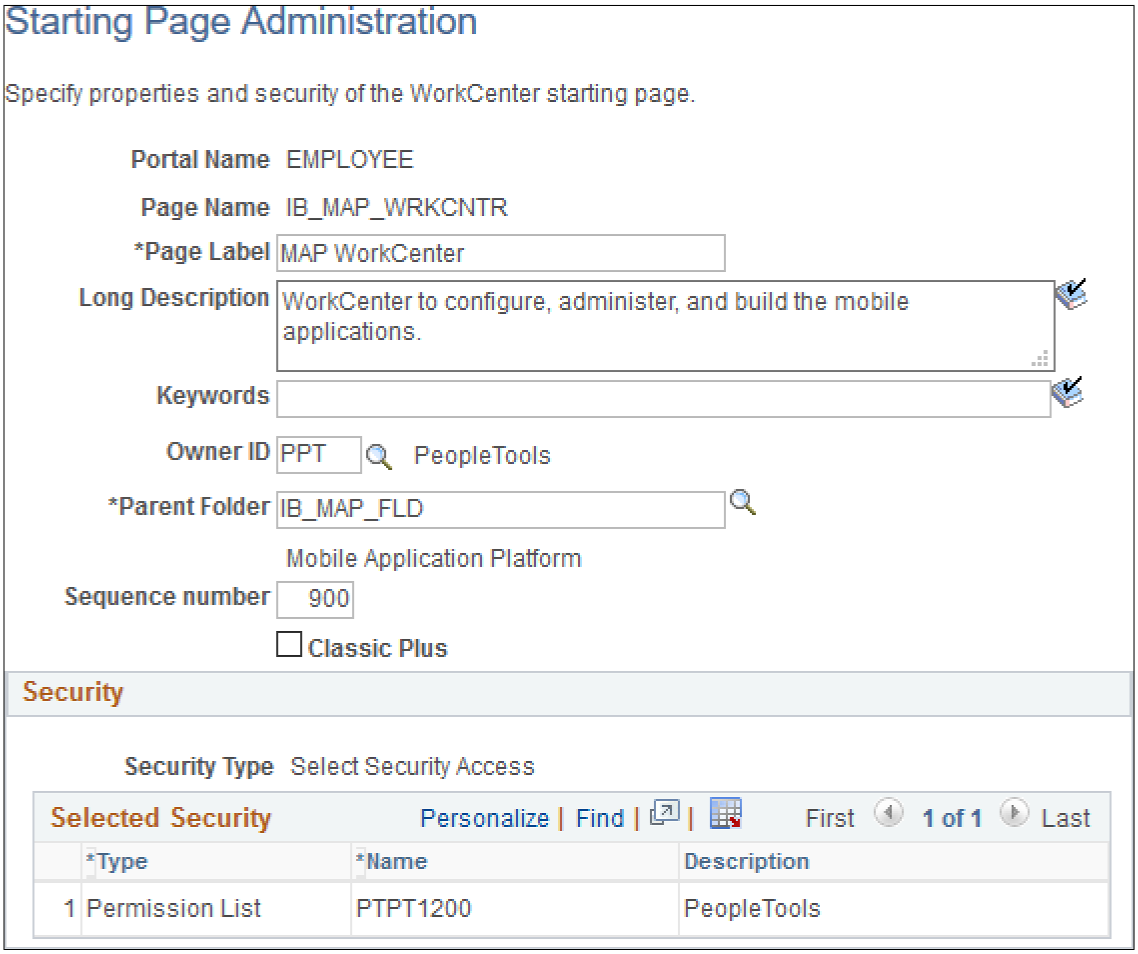
   - If you selected **Link of existing menu item,** then you must administer the starting page properties first. See Administering WorkCenter Starting Pages. Then, configure the WorkCenter. See Configuring WorkCenter Pages.

Administering WorkCenter Starting Pages

Access the Starting Page Administration page:

- On the Configure WorkCenter Page page, click the Properties button.
- When linking to an existing menu item as the WorkCenter starting page, click the content reference that represents the menu item. This page is displayed automatically

This example illustrates the fields and controls on the Starting Page Administration page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Portal Name	Displays the portal in which the content reference is registered.
Page Name	Enter the name for the new page when *linking* to an existing content reference only. **Note:** Otherwise, for existing starting pages, this field is display only.
Page Label	Enter the menu item title for the WorkCenter as it will appear in the portal menu and breadcrumbs.

Field or Control	***Description***
Long Description	Enter a description for the WorkCenter starting page. **Note:** This value also appears in tool tip text.
Keywords	Enter a keyword or keyword phrase to be stored as a content reference attribute. **Note:** Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language). See Defining Content References for more information on content reference attributes.
Owner ID	Select the object owner ID.
Parent Folder	Specify the parent folder for the menu item representing the WorkCenter.
Sequence number	Specify a sequence number for this menu item within its parent folder.
Classic Plus	Select this check box to display the pagelet area of this WorkCenter with fluid-like styling (which is also referred to as "classic plus"). Only the pagelet area of the WorkCenter is affected by this setting. The components displayed in the target content area of the WorkCenter are enabled for classic plus on a component by component basis. For more information on classic plus, see PeopleTools Components Delivered as Classic Plus. **Note:** If this WorkCenter includes an activity guide pagelet, then this setting is ignored and the pagelet area is displayed with standard, classic styling.

Security

The Security group box displays the selected security settings for this content reference. When the WorkCenter starting page is a WorkCenter dashboard, the fields in the Security group box are editable. Otherwise, when the starting page is either a link or an existing content reference, these fields are display only.

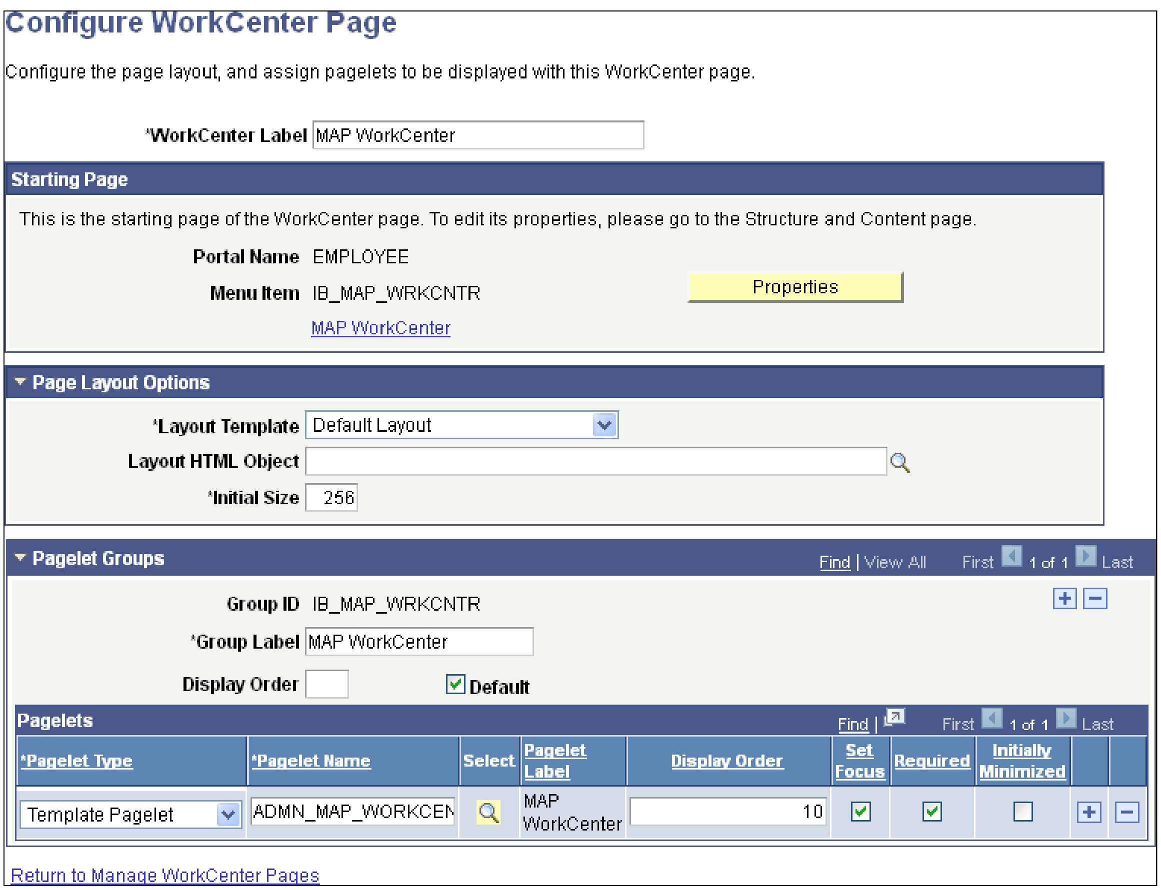
Field or Control	*Description*
Security Type	Select one of the following security types: • *Public* – Select this option to allow all users access to the content reference for this WorkCenter. • *Select Security Access* – Select this option to specify which permission lists have access to the content reference for this WorkCenter.
Selected Security	Select which permission lists have access to the content reference for this WorkCenter.

Configuring WorkCenter Pages

Access the Configure WorkCenter Page page by using one of these navigation paths:

- On the Manage WorkCenter Pages page, click the **Edit Configuration** button for a WorkCenter.
- In the title bar of the WorkCenter pagelet area, click the Configure link.

This example illustrates the fields and controls on the Configure WorkCenter Page page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
WorkCenter Label	Enter the name that appears on the Manage WorkCenter Pages page and in the header of the WorkCenter pagelet area. This field can contain up to 30 characters. **Note:** The default value is the name of the content reference, content reference link, or WorkCenter dashboard that you select as the starting page. This value also appears on the User Personalization page if one exists.
Return to Manage WorkCenter Pages	Click to access the Manage WorkCenter Pages page.

Starting Page

Field or Control	***Description***
Portal Name	Displays the current portal definition name.
Menu Item	Displays the menu item name that is associated with the selected content reference or WorkCenter dashboard or the page name that you specified for a content reference link on the Starting Page Administration page.
Properties	Click to administer properties of the WorkCenter starting page on the Starting Page Administration page. See Administering WorkCenter Starting Pages.
<Starting Page Label>	Click to access the WorkCenter.

Page Layout Options

Use these options to change the appearance and character of the WorkCenter page.

Changes that you make to the page layout options are visible after you save the page and click the Refresh button on your *browser*.

Field or Control	***Description***
Layout Template	Select a layout template from the list of all active page layout templates. The name of the default page layout template appears here when you create a new WorkCenter page.

Field or Control	*Description*
Layout HTML Object	Select a custom layout, which you define in an HTML definition in PeopleSoft Application Designer. Only HTML definitions that begin with *PTAL_LAYOUT_* appear in this prompt field.
Initial Size	Enter an integer between 200 and 9999 that determines the width of the pagelet area when the user initially accesses the WorkCenter page. The default value is *256*.

Pagelet Groups

Use the fields in this scroll area to organize the pagelets in the pagelet area into tabbed groups.

Field or Control	*Description*
Group ID	Enter a value that identifies the group. Each value must be unique to the WorkCenter page and can contain no more the 20 characters. After you initially add this value, the field becomes display-only. This field is required.
Group Label	Enter a meaningful name of up to 20 mixed-case characters for the pagelet group. The label appears on a tab in the pagelet area and is visible to the user. This field is required.
Display Order	Enter a number between 0 and 999 to manually sequence the order in which the pagelet groups appear. This field is empty by default and pagelet groups appear in the same order as the row sort order in the **Pagelet Groups** scroll area. If you omit the **Display Order** value then the system considers it to be *0* and appear before groups for which the field contains a value.
Default	Select to display the group by default when the WorkCenter page is initially accessed. Only one group can be the default.

Note these points about pagelet groups:

- If you create only one pagelet group, no tab appears to the user.
- If you delete a pagelet group, the application deletes all subordinate pagelet data and personalizations.

Pagelets

Use the fields in this grid to assign the pagelets to the group.

Field or Control	***Description***
Pagelet Type	Select *Template Pagelet* or *Homepage Pagelet.* **Note:** Remote template pagelets cannot be imported and used with a WorkCenter.
Pagelet Name	Select the pagelet from a list of registered pagelets. Only registered pagelets appear in the list. A pagelet can be assigned to *only one* group on a WorkCenter page.
Select	Click the Select button and select the pagelet from the Pagelet Search page to add to the WorkCenter definition. See Searching for and Importing Pagelets and Tiles for more information on searching for pagelets on local and remote nodes.
Pagelet Label	Displays the label that appears in the pagelet header.
Display Order	Enter a number between 0 and 999 to specifically determine the position of the pagelets in the group. This field is empty by default and pagelet groups appear in the same order as the row sort order in the **Pagelets** grid. Pagelets for which you leave the **Display Order** field blank are considered to have a value of *0* and appear before groups for which the field contains a value.
Set Focus	Select to move the cursor to the pagelet when the user initially accesses the WorkCenter page. You can select this check box for only *one* pagelet in the same group.
Required	Select to prevent the application user from hiding the pagelet in the group. The user can still minimize the pagelet.
Initially Minimized	Select to display the pagelet in the minimized state when the user initially accesses the WorkCenter page.

Add to Project

Click the Add to Project link to access the Add To Project page to insert this WorkCenter definition into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) for more information.

Creating and Managing WorkCenter Dashboards

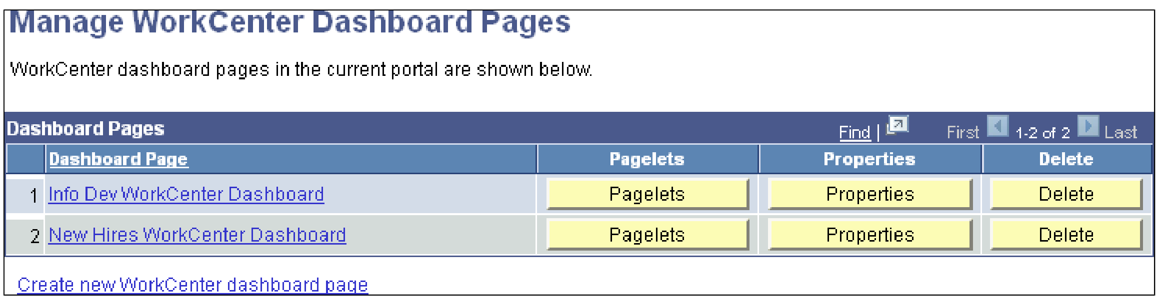
This section discusses how to:

- Manage WorkCenter dashboards
- Create WorkCenter dashboards
- Configure the layout of WorkCenter dashboards
- Select a WorkCenter dashboard as a starting page

Managing WorkCenter Dashboards

Access the Manage WorkCenter Dashboard Pages page. (Select **PeopleTools** > **Portal** > **WorkCenters** > **Manage WorkCenter Dashboards**.)

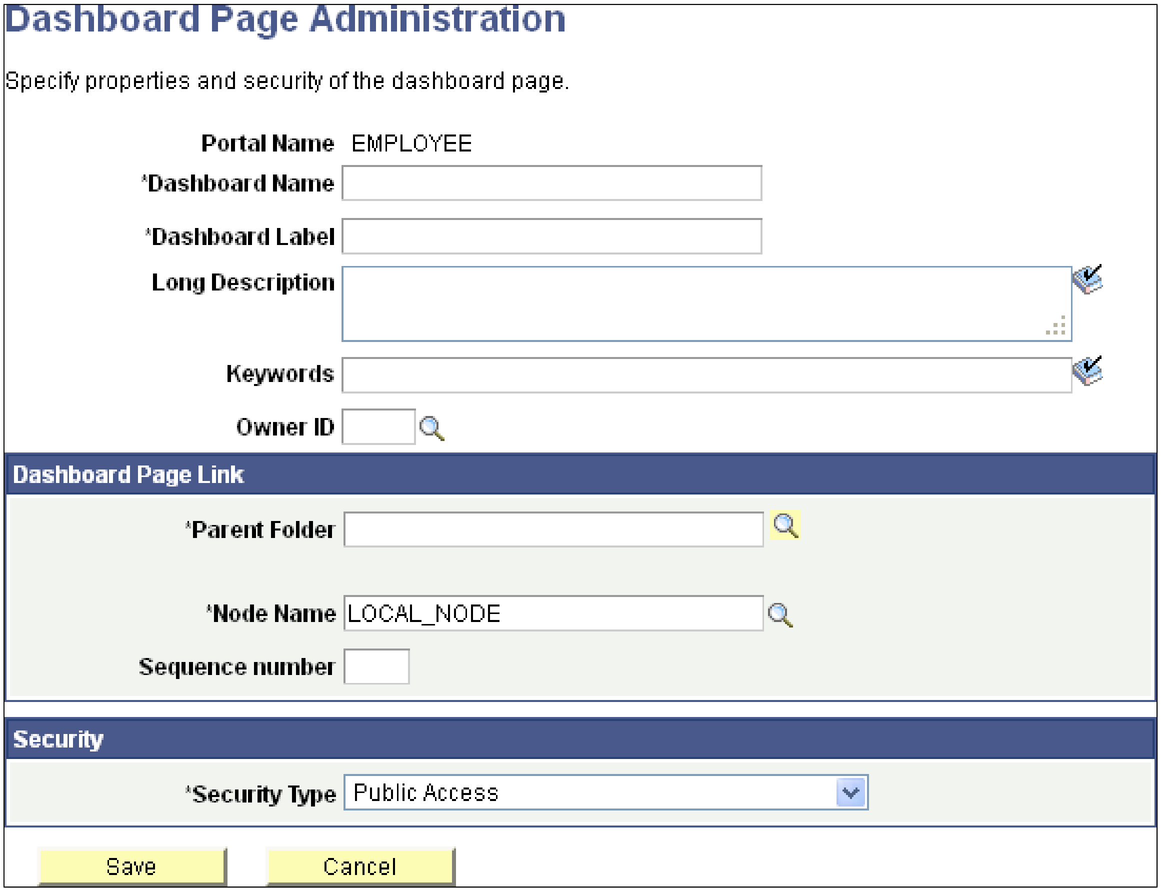
This example illustrates the fields and controls on the Manage WorkCenter Dashboard Pages page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Dashboard Page	Displays the content reference name as an active link. Click to access the content reference and view the WorkCenter dashboard page.
Pagelets	Click the **Pagelets** button to configure the layout of the dashboard. See Configuring the Layout of WorkCenter Dashboards.
Properties	Click the **Properties** button to administer the properties for the dashboard page. See Creating WorkCenter Dashboards.
Delete	Click the **Delete** button to delete the dashboard page.
Create new WorkCenter dashboard page	Click the link to create a new WorkCenter dashboard. See Creating WorkCenter Dashboards.

Creating WorkCenter Dashboards

This example illustrates the fields and controls on the Dashboard Page Administration page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Portal Name	Displays the portal in which the content reference is registered.
Dashboard Name	Enter the name for the new WorkCenter dashboard page.
Dashboard Label	Enter the menu item title for the WorkCenter dashboard as it will appear in the portal menu and breadcrumbs.
Long Description	Enter a description for the WorkCenter dashboard page. **Note:** This value also appears in tool tip text.

Field or Control	Description
Keywords	Enter a keyword or keyword phrase to be stored as a content reference attribute. **Note:** Two content reference attributes are created: KEYWORD (to store the keywords in the base language) and KEYWORD_*LNG* (to store the keywords in the current session language). See Defining Content References for more information on content reference attributes.
Owner ID	Select the object owner ID.

Dashboard Page Link

Field or Control	Description
Parent Folder	Specify the parent folder for the menu item representing the WorkCenter dashboard.
Node Name	Specify the portal host node for the WorkCenter dashboard.
Sequence number	Specify a sequence number for this menu item within its parent folder.

Security

The Security group box displays the selected security settings for this content reference.

Field or Control	Description
Security Type	Select one of the following security types: • *Public* – Select this option to allow all users access to the content reference for this WorkCenter dashboard. • *Select Security Access* – Select this option to specify which permission lists have access to the content reference for this WorkCenter.
Selected Security	Select which permission lists have access to the content reference for this WorkCenter dashboard.

Configuring the Layout of WorkCenter Dashboards

To specify the layout of WorkCenter dashboards:

1. On the Manage WorkCenter Dashboard Pages page, click the Pagelets button for a WorkCenter dashboard.
2. On the Tab Content page, select the pagelets that you want to appear on the WorkCenter dashboard. See Specifying the Pagelets or Tiles That Can Appear on a Dashboard.
3. On the Tab Layout page, arrange the layout of the pagelets. See Arranging Pagelets on Classic Dashboards.

Note: For WorkCenter dashboards, only the two-column layout is available.

Selecting a WorkCenter Dashboard as a Starting Page

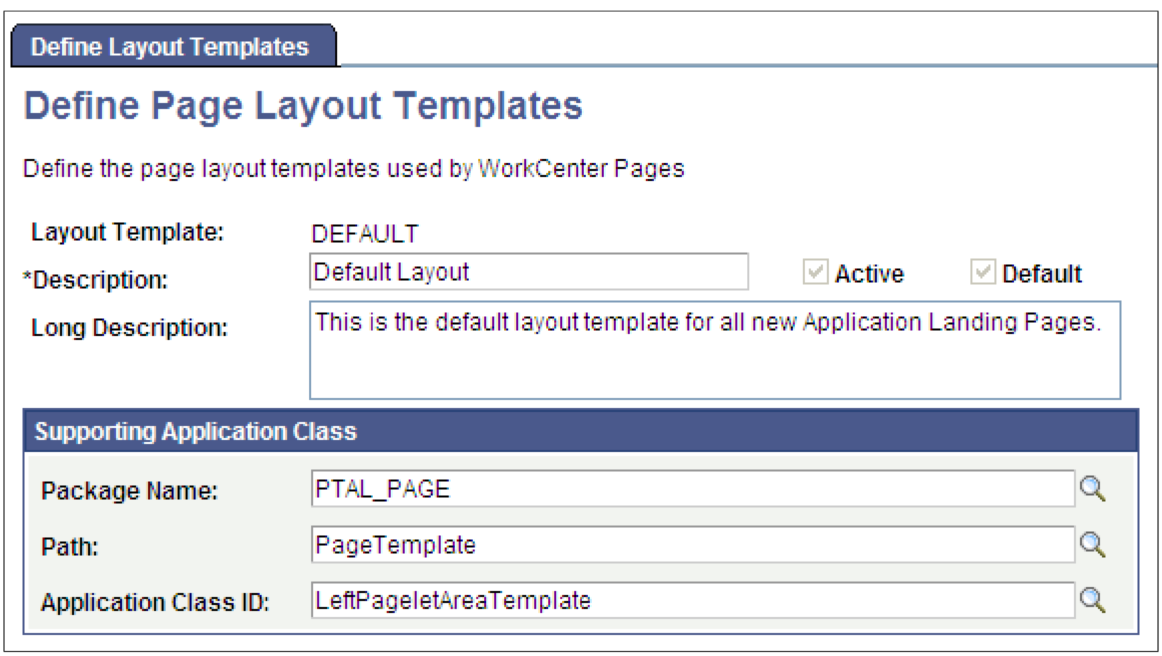
When a WorkCenter dashboard page is created, it is effectively already the starting page for its own WorkCenter, although it cannot yet be managed on the Manage WorkCenter Pages page.

From the Manage WorkCenter Pages page, you can create a new WorkCenter and designate a WorkCenter dashboard as the starting page for its own WorkCenter. Additionally, you can designate a WorkCenter dashboard as the starting page for a different WorkCenter. See Selecting the WorkCenter Starting Page.

Defining WorkCenter Page Layout Templates

Access the Define Page Layout Templates page (Select **PeopleTools** > **Portal** > **WorkCenters** > **WorkCenter Layout Templates**.)

This example illustrates the fields and controls on the Define Page Layout Templates page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Layout Template	View the template ID, which is the name as entered on the **Add New Value** page. This field is display-only.
Description	Enter text that briefly describes the purpose of the layout template. You can enter up to 30 characters.
Active	Select to enable the template. If you select this check box, you must enter the supporting application class. This check box is disabled if the template is designated as the default template because the default template must be active.
Default	Select to specify the template as the default template for all new WorkCenter pages. If you select this check box, you must designate the template as active. The application allows only one default template. To replace the current default template, you must create a new template and designate it as the active, default template. When you save the page, the application makes the **Default** and **Active** check boxes display-only.

Supporting Application Class

After you define the application class that creates the WorkCenter template in PeopleSoft Application Designer, you enter the class-identifying information in these fields:

Field or Control	***Description***
Package Name	Enter the application class package name that contains the PeopleCode that defines the template.
Path	Enter the names—delimited by colons—of each subpackage in the hierarchy that drills down to the class that contains the method that defines the template. If the class is defined in the top-level package, the Path lookup search results display a colon. Select the colon.
Application Class ID	Enter the name of the application class that determines the layout of the page. The class must exist in the application package and subpackage (path) that you specify. *LeftPageletAreaTemplate* and *RightPageletAreaTemplate* application classes come with all PeopleSoft applications.

Note: If the combination of the preceding fields creates an invalid application class path when you attempt to save the component, then the system displays an error message and prevents you from saving an invalid template. The error message specifies alternatives that you can take to eliminate the condition that causes the error.

Chapter 23

Developing and Deploying Activity Guides

Understanding Activity Guides

This section provides an overview of activity guides.

What Is an Activity Guide?

Activity guides are a PeopleTools feature that allow you to define guided procedures for a user or group of users to complete. Users are presented with a list of tasks, or steps, that need to be completed to finish the procedure. Activity guides can be useful in a number of procedural scenarios. For example:

- The on-boarding of a new employee.
- Benefits enrollment.
- A life change event, such as a marriage or the birth of a child.
- Period end closing.
- System or feature configuration.

An activity guide lists the tasks (referred to as action items) that need to be completed by a user or group of users. These action items are linked to specific local or remote transactions or external URLs. The action items can be assigned to a specific user by user ID or to a group of users by PeopleTools role. Activity guides provide a hierarchical organization of sequential or non-sequential tasks.

Typically, activity guides are generated from reusable definitions known as activity guide templates. These activity guide templates and action items are defined and managed through pages available under the PeopleTools menu. At run time, an activity guide instance can be instantiated from a template by providing context data. The state of the entire procedure as well as that of individual tasks is maintained between sessions, allowing a user to return later to complete the activity.

Important! PeopleTools activity guides are very different from workflow activity guides that are created from workflow activities within PeopleSoft Application Designer. See “Understanding Workflow Activity Guides” (Workflow Technology) for more information on workflow activity guides.

Activity Guide Templates and Instances

Typically, activity guides are defined by the properties and characteristics of activity guide templates. An *activity guide template* provides a reusable definition of the important aspects of an activity guide: its properties, participants, contextual data, list of action items and to whom they are assigned, and so on. Then, at run time, an *activity guide instance* can be created from the definitions stored in the template. The instance will be generated with the contextual data that uniquely differentiates one instance from another. For example, for a benefits enrollment activity guide, the contextual data would likely include the employee ID, the country and locale (state, province, and so on) where the employee works, and any other key data required to initiate the benefits process.

While activity guide instances are typically created dynamically through user interaction with the system, they can also be created manually from a template.

When an activity guide template is defined, you select one of seven process types for the template and its instances:

- Horizontal optimized guided sequential.
- Optimized guided sequential.
- Optimized guided non-sequential.
- Optimized non-guided non-sequential.
- Non-optimized guided sequential.
- Non-optimized guided non-sequential.
- Non-optimized non-guided non-sequential.

For more information on these template types and related terminology, see Understanding Activity Guide Templates and Template Types.

Action Items

Action items are the tasks that are assigned for completion within an activity guide. As an application developer, you will refer to these as action items. To users, they are presented as tasks or steps.

These action items can be linked to specific local or remote components, PeopleSoft queries, PeopleCode programs (an application class or an iScript), self-service process definitions, or external URLs. Within an activity guide, action items are organized into an ordered list; however, in many simple activity guides, action items can be completed in a non-sequential order. Action items can be organized hierarchically with a parent (summary) action item composed of child (detail) action items. Furthermore, an action item can be set to be dependent on the status of another action item.

Action items can be created and maintained within activity guide templates or within activity guide instances. However, typically you will define action items and their details at the template level only. Once an activity guide instance is created, updates to the action items themselves occur programmatically or through actions of the user.

Accessing Activity Guides

There are several mechanisms for users to access and display an activity guide:

- Classic activity guides require an activity guide pagelet, which is the interface that presents the activity guide and its action items to a user. The pagelet lists tasks in order and can optionally include a status bar indicating completion progress. Activity guide pagelets must be accessed through a classic WorkCenter.
- Fluid activity guides can be accessed in a number of ways:

- Through a tile that opens the specified activity guide instance.
- Through a tile that opens the My Activity Guide Instances page listing all activity guide instances for that template that the user is contributor to.
- Through a menu item.
- Through a button or link that programmatically generates the URL for the activity guide.

Procedure for Developing and Deploying Activity Guides

To develop and deploy an activity guide, follow this high-level procedure:

1. Determine the tasks to be completed by the users and the order of the tasks.
2. Determine whether task groupings exist—that is, summary tasks serve as parents to detailed tasks.
3. Determine which PeopleSoft component (transaction) corresponds to each task. Alternatively, a task can be associated with a PeopleSoft query, a non-PeopleSoft URL or a process.
4. Determine which user or role is to be assigned to each task.
5. Determine whether each task requires any pre-processing or post-processing logic.

   For example, post-processing logic is needed to change the status of an action item to complete.
6. Determine whether the activity guide instance is to be created dynamically or manually.
7. Determine the activity guide template type.

   See Selecting the Activity Guide Template Type.
8. Use the Setup Activity Guide WorkCenter to:

   a. Create the activity guide template.

      See Creating and Maintaining Activity Guide Templates.

   b. Create a definition for each action item in the template.

      See Creating and Maintaining Action Items.
9. For a fluid activity guide, deploy the activity guide in one or more ways:

   - Through a tile that opens the specified activity guide instance.
   - Through a tile that opens the My Activity Guide Instances page listing all activity guide instances for that template that the user is contributor to.
   - Through a menu item.
   - Through a button or link that programmatically generates the URL for the activity guide.

10. For a classic activity guide:

    a. Generate the activity guide pagelet.

       See Deploying Classic-Only Activity Guides.

    b. Configure a WorkCenter for the activity guide.

       See Configuring a WorkCenter Page for a Classic-Only Activity Guide.

11. (Optional) Generate an activity guide instance from the template.

    See Creating New Activity Guide Instances Manually.

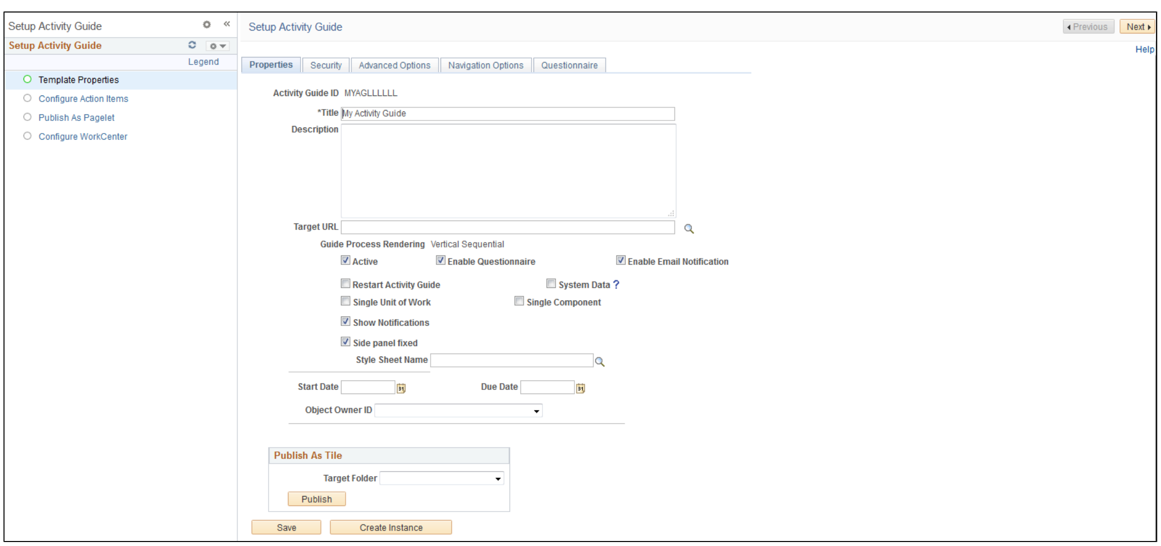
Using the Setup Activity Guide WorkCenter

Use the Setup Activity Guide WorkCenter to maintain an activity guide template definitions, maintain the action item definitions in the template , and to configure the deployment of the activity guide.

Navigation:

1. **PeopleTools** > **Activity Guides** > **Activity Guide Templates**
2. On the Manage Activity Guide Templates page do one of the following:
    - Click the Create Template button to create a new activity guide template and open the WorkCenter.
    - Click the Properties link for an existing template to open the WorkCenter on the Template Properties task.
    - Click the title link for an existing template to open the WorkCenter on the Configure Action Items task.

This example illustrates the Setup Activity Guide WorkCenter and the tasks in the Setup Activity Guide pagelet.



The Setup Activity Guide pagelet (itself an activity guide) presents the four tasks you must complete to define and deploy an activity guide:

1. Template Properties –

   Use the Template Properties task to define the activity guide template including its properties, members and privileges, advanced options, and pagelet options. See Creating and Maintaining Activity Guide Templates.

2. Configure Action Items –

   Use the Configure Action Items task to create a definition for each action item in the template. See Creating and Maintaining Action Items.

3. Publish as Pagelet –

   For classic activity guides only, use the Publish as Pagelet task to generate the activity guide pagelet. See Deploying Classic-Only Activity Guides.

4. Configure WorkCenter –

   For classic activity guides only, use the Configure WorkCenter task to configure a WorkCenter for the activity guide. See Configuring a WorkCenter Page for a Classic-Only Activity Guide.

Prerequisites

Action items can be linked to specific local or remote components, PeopleSoft queries, PeopleCode programs (an application class or an iScript), self-service process definitions, or external URLs. Except for self-service process definitions, these links are made through the PeopleSoft Related Content Framework. Moreover, except for component-based related content service definitions, which can be created by clicking the Create Service link in the Action Item Details grid, you *must* explicitly use the Define Related Content Service page to create any other related content service definition prior to associating that service definition with an action item.

Related Links

Creating and Managing Related Content Service Definitions

Limitations

Please note the following limitations on developing and using activity guides:

- If the user accesses and updates a transaction directly (for example, through standard menu navigation) rather than through the action item link in the activity guide pagelet, then any status change effected there will not be reflected in the activity guide pagelet.
- If an action item is linked to an external URL (that is, a non-PeopleSoft URL), then you must provide a custom button to allow the user to mark the action item as complete.
- Do not link an action item to a multistep transaction such as a transaction that represents a wizard (for example, Pagelet Wizard).
- Activity guides support a two-level hierarchy only—that is, only one level of detail is supported beneath a summary action item.

- In custom action buttons, do not attempt component buffer access, use think-time functions such as MessageBox, WinMessage, or others, or use system variables. See Developing Custom Action Buttons.
- Do not configure a remote activity guide pagelet for use in a WorkCenter defined on the local node. Instead, define both the WorkCenter and activity guide pagelet on the remote node. Then, using unified navigation or another mechanism, access the content reference for the remote WorkCenter from the local node.
- The Related Content frame configured for a classic page will not be displayed when that page is rendered inside any fluid wrapper, such as in a fluid activity guide or in the fluid master/detail container.
- Activity guides and fluid navigation collections do not support conditional navigation.

Related Links

“Think-Time Functions” (PeopleCode Developer’s Guide)

Creating and Maintaining Activity Guide Templates

This section discusses how to create and maintain activity guide templates.

Guidelines and Limitations for Using Fluid Components in Activity Guides

Note: See the “Terminology” section for definitions of terms including *optimized, non-optimized, activity guide-optimized fluid components,* and others.

Guidelines for All Activity Guide Types

All fluid components must adhere to the following guidelines to be used in an activity guide template of any type (*optimized* or *non-optimized*):

- The fluid component must be defined as a *Standard* fluid component.

  Important! A fluid wrapper component (that is, another fluid activity guide, a master/detail component, or fluid navigation collection) can never be included as target content inside a fluid activity guide.

- Typically, for each page defined within the fluid component, the outermost group box must include the ps_apps_content style or another styling appropriate to the page design.
- The fluid component must have the search page type set to *None* to be included in a fluid activity guide. Otherwise, if the search page type is set to *Standard* or *Master/Detail,* the fluid component cannot be included in a fluid activity guide.
- Unless AJAX transfers are enabled on the activity guide, the fluid component should not perform a transfer to another component. This includes transfers via a hyperlink, a push button, a PeopleCode program, or transfers from the left panel. Such transfers will result in the user exiting that activity guide instance in an invalid manner. See “AJAX Transfers” (Fluid User Interface Developer’s Guide) for more information.

- The activity guide instance maintains the fluid banner and the visibility of items in the banner.
  - *Direct* manipulation of the fluid banner via PeopleCode (to set the page title, for example) will result in a PeopleCode error. Therefore, use the PT_PAGE_UTILS:Banner application class to *indirectly* attempt to manipulate the fluid banner. When used, the methods and properties of PT_PAGE_UTILS:Banner are ignored and have no effect if the component is rendered in a context that does not allow access to the fluid banner and Actions menu such as in an activity guide instance. See "Banner Class Methods" (PeopleCode API Reference) for more information.
  - In addition, pages in the fluid component cannot include an "application header"—that is, the use of any of the Custom Header group box types (Left, Right, Middle, or Bottom) is not allowed.
  - Nevertheless, custom Actions menu items can be defined using the Custom Action Menu group box type.
- If the fluid component includes a page definition for a Header page type, this page definition will be ignored and not displayed within an activity guide instance.

Additional Guidelines for Optimized Activity Guide Types

The following additional guidelines and limitations apply to "activity guide-optimized fluid components":

- If the target content is a fluid component that employs an internal two-panel construct (that is, it either includes a Side Page 1 page, was built from the PeopleTools 8.54 PSL_TWOPANEL layout page, or was designed with an internal left panel similar to PSL_TWOPANEL), you must enable left panel collision handling with the slide-out left panel on the activity guide. See "Left Panel Collision Handling with Slide-out Left Panels" (Fluid User Interface Developer's Guide) for more information.
- For each page defined within the component, include a group box of type Page Title that is defined to be visible when the page is rendered within an activity guide context. Depending on the implementation, the activity guide itself might control the title displayed by this group box. To hide this page title if the page is rendered outside of an activity guide instance, use the psc_hideinmode-notag style.
- To maintain a clear and consistent user experience, Oracle recommends that the fluid component not include a page definition for a Footer Page type.
- To use a multi-page fluid component in an optimized activity guide, you can do one of the following to maintain a clear and consistent user experience:
  - Re-parent the page tabs to a group box of type Page Tab that is included on *each* page definition.
  - Include PT_SIDE_PAGETABS in the component definition to re-parent the page tabs to the left panel. You must also enable left panel collision handling with the slide-out left panel on the activity guide. See "Left Panel Collision Handling with Slide-out Left Panels" (Fluid User Interface Developer's Guide) for more information.
  - Hide the page tabs. You can do so in the component definition by deselecting the Display Folder Tabs (top) on the Internet tab. Alternatively, you can use delivered styles to hide the page tabs.

Related Links

PeopleSoft Fluid UX Standards: Activity Guides

Understanding Activity Guide Templates and Template Types

Activity guide templates provide a mechanism for defining the properties, security, action items, contextual data, and other options for activity guides once. Then, individual activity guide instances can be generated from these reusable templates. Once an activity guide has been instantiated from a template, the instance cannot be associated with a different template. In addition, while the activity guide template can be modified, any active or inactive instances already generated from that template will not be altered by the modifications.

Typically, activity guide properties, security, advanced options, and other attributes are created and maintained at the activity guide template level only. However, you can also maintain these attributes within activity guide instances; the same pages with the same fields are used in both cases. While the examples in this section illustrate creating and maintaining these attributes within an activity guide template, these examples could apply equally to activity guide instances.

Terminology

The following terms are used to differentiate the types of activity guides that can be created, and the types of processes these activity guides support:

Term	***Definition***
Optimized	*Optimized* activity guides support only activity guide-optimized fluid components as well as classic components. This allows an optimized activity guide to control the use and display of the left panel. *Non-optimized* activity guides support any fluid components that adhere to the guidelines noted previously as well as classic components. Instead of presenting the action items in the left panel, a non-optimized activity guide presents a vertical drop-down list in the fluid banner (or header) or uses an activity guide pagelet for classic activity guides.
Activity guide-optimized fluid components	An *activity guide-optimized fluid component* adheres to the guidelines for all fluid components as well as the additional guidelines for optimized activity guide types.
Sequential	In a *sequential* activity guide, action items are numbered and must be completed in the specified order. With *non-sequential* activity guides, action items are not numbered and can be completed in any order unless dependencies are defined. **Note:** Auto-save is enabled for sequential activity guides automatically. However, each fluid component used on a step should implement a standard PeopleTools save processing mechanism—for example, a Save button with the toolbar action set to Save on the page field properties.

Term	***Definition***
Guided	*Guided* activity guides include Next and Previous in the activity guide subbanner. *Non-guided* activity guides do not include any navigation buttons; in addition, non-guided activity guides display the standard fluid banner.
Activity guide subbanner	The activity guide subbanner displays Next and Previous navigation buttons, as well as any custom action buttons defined for the activity guide.
Simple	*Simple* activity guides support a simple hierarchy with only one level of action items. Otherwise, activity guides can support up to two levels of action items: summary and detail.
Single unit of work (SUOW)	Certain activity guide types can be defined as a *single unit of work* spanning multiple components in the same system. No data is committed until a submit is performed at the final step; if the user exits the activity guide before the submit, all changes are rolled back. All SUOW activity guides are sequential; moreover, data dependencies can be configured between action items using input/output parameters.
Single component	Certain activity guide types can be defined as *single component,* which allows different pages from the same component to be displayed as separate steps within the activity guide. Similar to a SUOW activity guide, no data is committed until a submit is performed at the final step; if the user exits the activity guide before the submit, all changes are lost. However, unlike a SUOW activity guide, a Save button can be configured to perform a save at any step.

Note the following differences between single unit of work and single component activity guides along with important limitations:

Single Unit of Work	***Single Component***
Each action item (step) is a separate component.	Each action item (step) is a page from the same component.
Supported only in optimized guided sequential activity guide types. **Important!** Although optimized activity guide types also support classic components, classic components are not supported in SUOW activity guides.	Supported in optimized guided sequential or non-sequential activity guide types. **Important!** Although optimized activity guide types also support classic components, classic components are not supported in single component activity guides.

Single Unit of Work	***Single Component***
Executes component PeopleCode events such as PreBuild and PostBuild for each component.	Executes component PeopleCode events such as PreBuild and PostBuild once only.
Auto-save is enabled for SUOW activity guides automatically. SaveEdit PeopleCode is triggered for every action item after a user clicks the Next button or navigates to any other action item. **Important!** Components included in this activity guide must never invoke the DoSave or DoSaveNow built-in PeopleCode functions. **Note:** Any push button configured with the Save action type is automatically hidden.	Auto-save is not enabled for single component activity guides. SaveEdit PeopleCode is triggered only once after a submit configured on the last step of the component. To perform validation at a specific action item, override the behavior of the default Next or Previous buttons.
Does not support components from multiple PeopleSoft applications—that is, all the components have to be defined on the same local PeopleSoft application in which the activity guide is defined.	NA

Template Process Types

When an activity guide template is defined, you select one of seven process types for the template and its instances. The following table summarizes the features available with each activity guide type. Note the following regarding the columns in this table:

- *Opt Fl*: Activity guide-optimized fluid components (that comply with the guidelines listed previously).
- *Fluid*: Any standard fluid components (that comply with the guidelines listed previously).
- *Classic*: Any classic components.
- *S/D*: Summary-detail action items.
- *SUOW*: Single unit of work.
- *1 Comp*: Single component.

Note: Except for the one type labeled horizontal, all other activity guides present the list of action items in one type of vertical format or another.

Activity Guide Template Type	***Opt Fl***	***Fluid***	***Classic***	***S/D***	***SUOW***	***1 Comp***
Horizontal optimized guided sequential • Presents the numbered steps horizontally. • Supports only activity guide-optimized fluid components in a *simple* hierarchy with only one level of action items. • Uses the activity guide subbanner to present navigation buttons.	Y	N	Y	N	Y[1]	Y[1]
Optimized guided sequential • Presents the numbered steps vertically in the left panel. • Supports only activity guide-optimized fluid components in up to a two-level hierarchy with summary and detail action items. • Uses the activity guide subbanner to present navigation buttons.	Y	N	Y	Y	Y[1]	Y[1]
Optimized guided non-sequential • Presents unnumbered steps vertically in the left panel. • Supports only activity guide-optimized fluid components in up to a two-level hierarchy with summary and detail action items. • Uses the activity guide subbanner to present navigation buttons.	Y	N	Y	Y	N	Y[1]
Optimized non-guided non-sequential • Presents unnumbered steps vertically in the left panel. • Supports only activity guide-optimized fluid components in up to a two-level hierarchy with summary and detail action items. • Displays the standard fluid banner without navigation buttons.	Y	N	Y	Y	N	N

Activity Guide Template Type	***Opt Fl***	***Fluid***	***Classic***	***S/D***	***SUOW***	***1 Comp***
Non-optimized guided sequential • Presents the numbered steps vertically in a drop-down list in the activity guide subbanner. • Supports any component type in up to a two-level hierarchy with summary and detail action items. • Uses the activity guide subbanner to present navigation buttons.	Y	Y	Y	Y	N	N
Non-optimized guided non-sequential • Presents unnumbered steps vertically in a drop-down list in the activity guide subbanner. • Supports any component type in up to a two-level hierarchy with summary and detail action items. • Uses the activity guide subbanner to present navigation buttons.	Y	Y	Y	Y	N	N
Non-optimized non-guided non-sequential • Presents unnumbered steps vertically in a drop-down list in the fluid banner. • Supports any component type in up to a two-level hierarchy with summary and detail action items. • Displays the standard fluid banner without navigation buttons.	Y	Y	Y	Y	N	N

Note:
[1] Classic components are not supported in SUOW or single component activity guides.

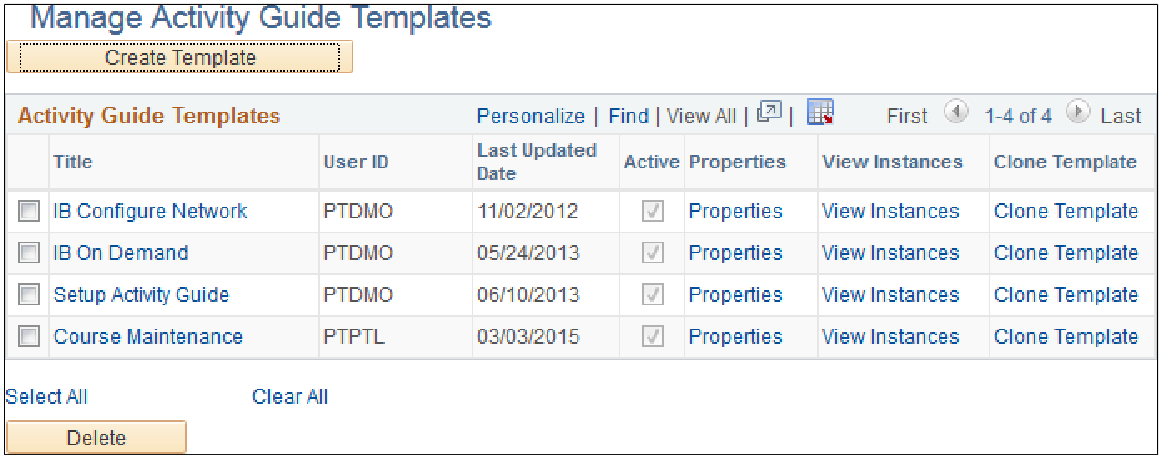
Managing Activity Guide Templates

Use the Manage Activity Guide Templates page to manage the activity guide templates.

Navigation:

PeopleTools > **Activity Guides** > **Activity Guide Templates**

This example illustrates the fields and controls on the Manage Activity Guide Templates page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Create Template	Click the **Create Template** button to create a new activity guide template. After you specify the activity guide template type (and data type), the new activity guide template is displayed in the Setup Activity Guide WorkCenter.
Title	Click a title link to manage the action items for this template. Clicking this button opens Setup Activity Guide WorkCenter with the Configure Action Items step selected.
Properties	Click the **Properties** link to manage the properties for this template. Clicking this button opens Setup Activity Guide WorkCenter with the Template Properties step selected.
View Instances	Click the **View Instances** link to view the Manage Activity Guide Instances page displaying all instances created from this template.
Clone Template	Click the **Clone Template** link to make a copy of this activity guide template.
Select All	Click to select all templates.
Clear All	Click to clear the selection of any templates.
Delete	Click to delete the selected templates.

Related Links

Using the Setup Activity Guide WorkCenter
Defining Activity Guide Template Properties
Creating and Maintaining Action Items
Managing Activity Guide Instances
Copying Activity Guide Templates

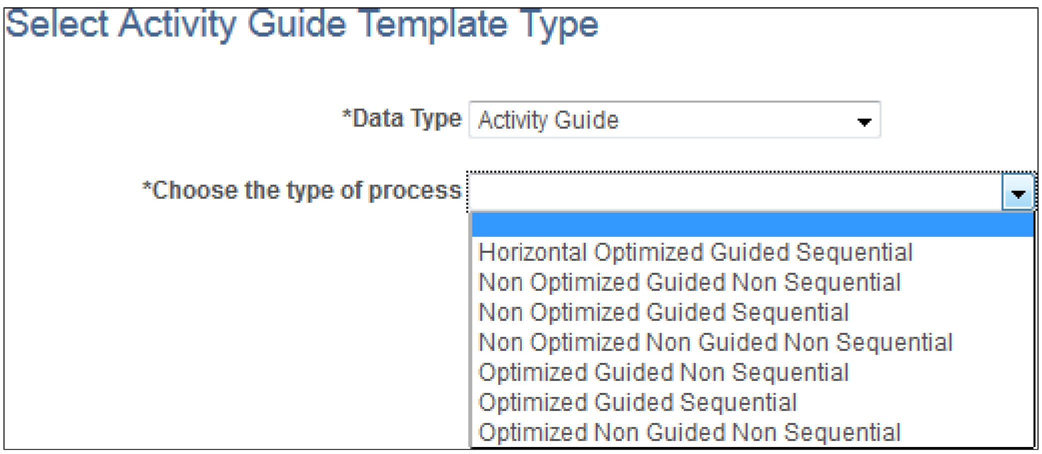
Selecting the Activity Guide Template Type

Use the Select Activity Guide Template Type page to select the activity guide data type and process type.

Navigation:

On the Manage Activity Guide Templates page, click the Create Template button to create a new template definition.

This example illustrates the fields and controls on the Select Activity Guide Template Type page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Data Type	Select *Activity Guide* to use the PeopleTools-delivered activity guide data source. **Note:** If you have created a custom activity guide data type, you can select it instead of *Activity Guide.* See Creating a Custom Activity Guide Data Type for more information.
Choose the type of process	Select one of the seven activity guide template types. See the preceding section, “Template Process Types,” for a table describing the features of each of the template types. **Important!** Once a template type has been selected, it cannot be changed for that template definition.

Related Links

Understanding Activity Guide Templates and Template Types

Defining Activity Guide Template Properties

Use the Template Properties: Properties page to define the properties for an activity guide template.

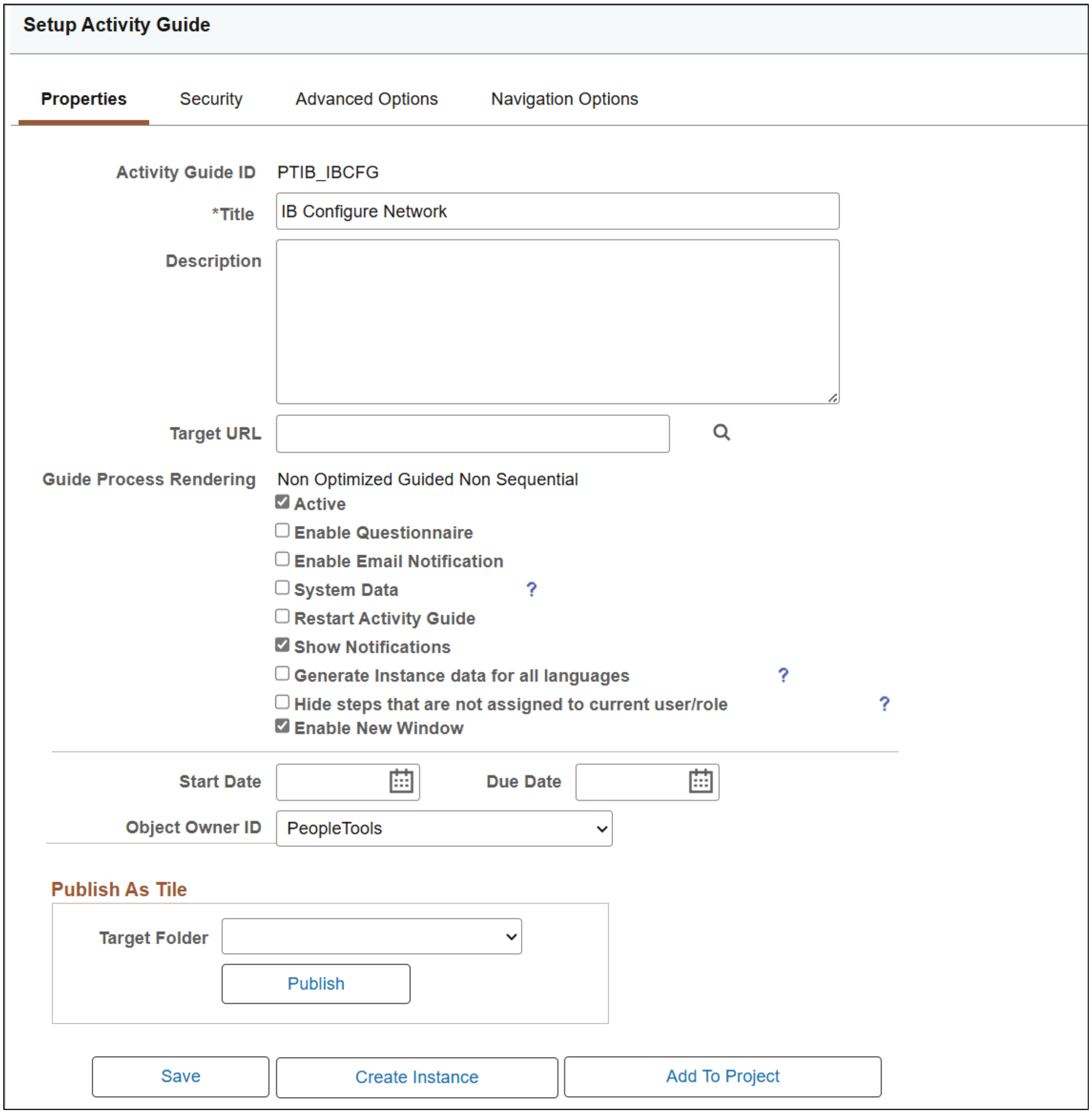
Navigation:

- On the Manage Activity Guide Templates page, click the Properties link to modify an existing template.

  Note: In addition, on the Manage Activity Guide Instances page, click the Properties link for an existing instance.

- In the Setup Activity Guide WorkCenter, click the Template Properties task.

This example illustrates the fields and controls on the Template Properties: Properties page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Activity Guide ID	Enter up to 10 characters as an identifier for the template. When you save the template, the system will fill characters up to the 10-character maximum. **Important!** You must ensure that each template ID is unique.
Title	Enter a name for the template.
Description	Enter a description for the template.

Field or Control	***Description***
Target URL	For classic activity guides only, select the URL for the WorkCenter that provides access to this activity guide. Defining the URL here provides access from the Activity Guides - In Progress homepage pagelet to the WorkCenter. **Important!** To ensure that the URL is valid, select the URL from the lookup list, rather than typing or pasting a URL.
Active	Select this option to make the template active. If this option is cleared, the template definition remains in the system, but is not available to be used to create new activity guide instances.
Enable Questionnaire	Select this option to enable a questionnaire for this template. **Note:** The Questionnaire page tab will be displayed after you save the template definition.
Enable Email Notification	Select this option to enable email notifications to the users assigned to action items of type *Process*. **Note:** Email messages are sent when the process has completed successfully or if it has terminated with an error.
Restart Activity Guide	Select this option to have the activity guide instance deleted and restarted if the user exits without completing the activity guide. If this option is not selected, the default is to return to and display the current step whenever the user exits and returns to the activity guide. **Note:** SUOW and single component activity guides that have not been submitted as complete always restart at the first step, similar to selecting this option. This is because the old instance is deleted and a new instance is generated whenever the user re-enters the activity guide. **Important!** Do not set restart on multi-user activity guides. Note the following important considerations for single user and multi-user activity guides: • If a single-user activity guide is set with either restart, single component, or SUOW, then the context keys must be set in a way that only allows access by that user only —for example, by using OPRID as one of the context keys. • If an activity guide is intended to be worked by multiple users, it cannot have any of restart, single component, or SUOW set.

Field or Control	*Description*
System Data	Select this option to lock the activity guide definition including all template properties and action item definitions. **Important!** Once you select this option, you will be unable to make any modifications to the template definition. Do not select this option before you have completed specified all template properties, defined all action items, and generated and tested activity guide instances.
Single Unit of Work	Select this option to enable this activity guide as a single unit of work. This option is available for two activity guide template types: • Horizontal optimized guided sequential. • Optimized guided sequential. **Note:** SUOW and single component activity guides that have not been submitted as complete always restart at the first step, similar to selecting the restart option. This is because the old instance is deleted and a new instance is generated whenever the user re-enters the activity guide. **Important!** Do not set SUOW on multi-user activity guides. Note the following important considerations for single user and multi-user activity guides: • If a single-user activity guide is set with either restart, single component, or SUOW, then the context keys must be set in a way that only allows access by that user only—for example, by using OPRID as one of the context keys. • If an activity guide is intended to be worked by multiple users, it cannot have any of restart, single component, or SUOW set.

Field or Control	*Description*
Single Component	Select this option to enable this activity guide as a single component. This option is available for three activity guide template types: • Horizontal optimized guided sequential. • Optimized guided sequential. • Optimized guided non-sequential. **Note:** SUOW and single component activity guides that have not been submitted as complete always restart at the first step, similar to selecting the restart option. This is because the old instance is deleted and a new instance is generated whenever the user re-enters the activity guide. **Important!** Do not set single component on multi-user activity guides. Note the following important considerations for single user and multi-user activity guides: • If a single-user activity guide is set with either restart, single component, or SUOW, then the context keys must be set in a way that only allows access by that user only—for example, by using OPRID as one of the context keys. • If an activity guide is intended to be worked by multiple users, it cannot have any of restart, single component, or SUOW set.
Show Notifications	Select this option to display the Notification icon in the fluid banner. This option is available for the five *guided* activity guide template types: • Horizontal optimized guided sequential. • Optimized guided sequential. • Optimized guided non-sequential. • Non-optimized guided sequential. • Non-optimized guided non-sequential.

Field or Control	*Description*
Side panel fixed	Select this option to fix the left panel, which disables the users' ability to hide the left panel on any device except for small form factor devices. This option is available for the three *vertical optimized* activity guide template types: • Optimized guided sequential. • Optimized guided non-sequential. • Optimized non-guided non-sequential. **Note:** On small form factor devices or when the browser width is too narrow, this option is ignored and the left panel becomes collapsible and expandable.
Generate Instance data for all languages	Select this option to indicate that multi-language data present in the template is to be copied when creating a new instance from the template. Changing this option in an already created instance has no effect since the translation data can only be copied from the template at instance creation. However, any instance-specific action items added programmatically using the InstanceCreation method will not have translations in the generated instance unless the application code also adds the translation data. In addition, other programmatic updates handled during instance creation may have translation impacts. For example, if the title of the activity guide is changed to reflect the specific instance, then the application code should also update the title in all languages supported by the template. **Important!** For performance reasons during instance creation, enable this setting only on templates that require it.
Hide steps that are not assigned to current user/role	Select this option to indicate that steps that are not assigned to the current user are hidden and not displayed to that user. This setting can be changed dynamically or manually in instances created from any activity guide template. This option can be used on non-sequential activity guides without dependencies among steps. Use care if the non-sequential activity guide has dependencies among steps to ensure that each contributor user is able to access at least one step in the activity guide when they access the instance. Otherwise, an error will occur. **Note:** This option is not available for sequential activity guides, single component activity guides, or single unit of work activity guides, or for activity guides that have restart enabled.

Field or Control	*Description*
Enable New Window	Select this option to configure the New Window option for Activity Guides. By default, this option is selected. The New Window option is available as an item in the Actions Menu. When you select the New Window item, the system either • opens the last visited homepage in a new window if the Back button stack contains a last visited homepage, or • opens the default page (in a new window) if the Back button stack does not contain a last visited homepage. The default page is the page that displays when you select the Home button. For more information, see “Opening New Windows” (Applications User’s Guide).
Start Date	Select a start date.
Due Date	Select a due date. **Note:** The start and due dates specified at the template level do not have any effect on start and due dates at the action item level.
Object Owner ID	Select the object owner ID.
Create Instance	Use the Create Instance button to create an activity guide instance from this template.
Add To Project	Click the Add To Project button to access the Add To Project page to insert this activity guide template definition into an ADS project. See “Adding Data Set Instances from Object Designer Pages” (Lifecycle Management Guide) for more information.

Publish As Tile

Use the Publish As Tile group box to publish a tile definition that opens the My Activity Guide Instances page for this activity guide template. This tile definition will include the template ID only, and no context data or instance IDs. Therefore, users who use this tile will be redirected to the My Activity Guide Instances page, on which they can select from a list of *existing* activity guides instances generated from the template for which they have contributor privileges.

Important! The published tile cannot be used as a general deployment mechanism or to generate new activity guide instances. See Deploying Optimized and Non-Optimized Fluid Activity Guides for information on how to deploy fluid activity guides.

Field or Control	*Description*
Target Folder	Select the tile repository folder in which you want to create the tile definition.
Publish	Click the Publish button to publish the tile definition.
Edit Tile Properties	Click this link to open and edit the content reference definition for the tile within the portal registry.

Related Links

Creating New Activity Guide Instances Manually
Accessing Instances from the My Activity Guide Instances Page
Accessing Instances from the Activity Guides - In Progress Pagelet

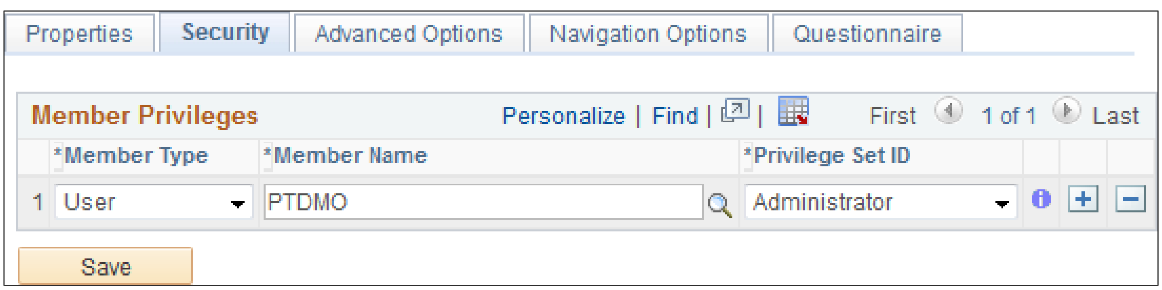
Defining Activity Guide Template Security

Use the Template Properties: Security page to manage participants and privileges for the template.

Navigation:

1. In the Setup Activity Guide WorkCenter, click the Template Properties task.
2. Select the Security page.

This example illustrates the fields and controls on the Template Properties: Security page. You can find definitions for the fields and controls later on this page.



Use the *Administrator* privilege set for the roles (or users) who will be responsible for maintaining this template and any activity guide instances generated from it. Use the *Contributor* privilege set for the users (or roles) who will be assigned to complete the action items within an active instance.

Field or Control	***Description***
Member Type	Select the type of participant you want to add to the activity guide template. Available values include: • *Role* — Select to be able to select a role in the **Member Name** field. Available names are derived from PeopleSoft roles defined in the Roles component. See “Defining Role Options” (Security Administration). • *User* — Select to be able to select a user in the **Member Name** field. Available users are derived from PeopleSoft users defined in the User Profiles component. See “Specifying User Profile Attributes” (Security Administration).
Member Name	Specify the user ID or role name for the participant.
Privilege Set ID	Select a privilege level you want to assign to the participant: • *Administrator* — Administrators can make modifications to the template and any instances generated from the template including modifying properties, modifying participants, and maintaining action item definitions. Oracle recommends that you assign the administrator privilege to a system-defined administrative role such as Portal Administrator, PTAI_ADMINISTRATOR, and so on. • *Contributor* — Contributors are the only participants who can complete assigned action items within an active instance. Oracle recommends that you assign the contributor privilege to user ID such as an employee, a customer, a partner, and so on. In addition, an activity guide tile generated from this template will display only those activity guide instances generated from this template for which they have contributor privileges. Plus, the Activity Guides - In Progress homepage pagelet will display activity guide instances only for users who have been assigned the contributor privilege. • *Viewer* — Viewers have read-only access to the activity guide instances and their action items. **Important!** The *Viewer* privilege does not have any practical usage in the current implementation.

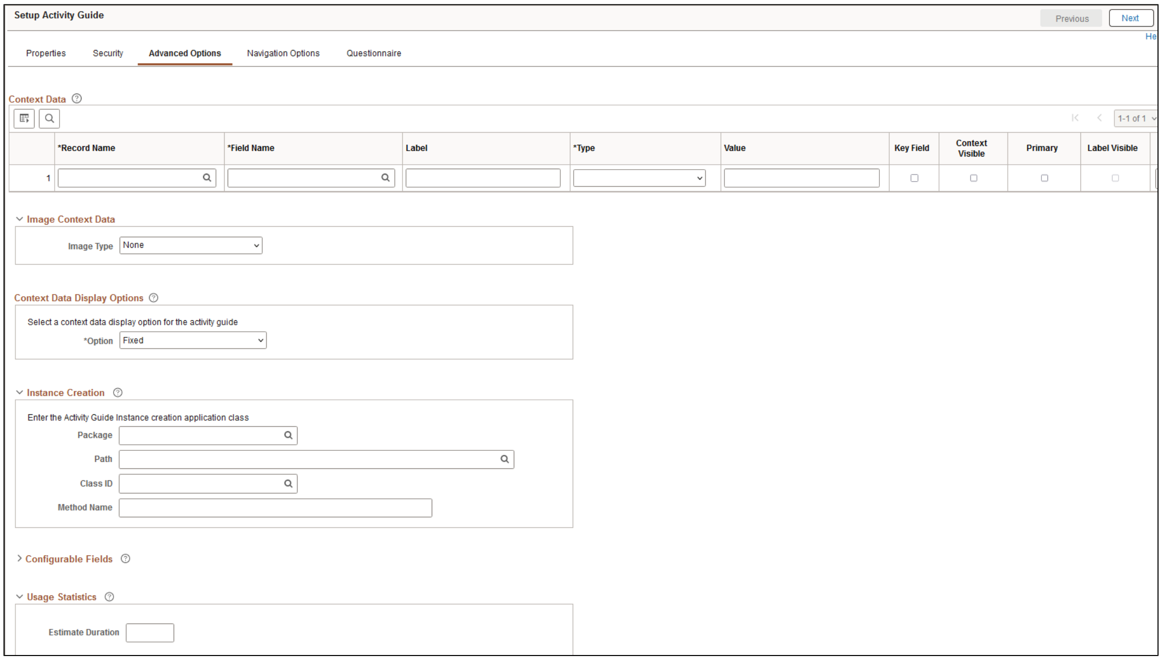
Defining Advanced Options

Use the Template Properties: Advanced Options page to define advanced options for this template.

Navigation:

1. In the Setup Activity Guide WorkCenter, click the Template Properties task.
2. Select the Advanced Options page.

This example illustrates the fields and controls on the Template Properties: Advanced Options page. You can find definitions for the fields and controls later on this page.



Context Data

Important! If multiple activity guide instances will be generated from the same template (for example, a life events activity guide template), then you must define at least one key context field in order to identify and differentiate these activity guide instances.

You can define contextual data that will be passed to the activity guide instance when it is created. Contextual data defined as key fields will be used to uniquely identify each activity guide instance. For example, if the employee ID were defined as a key contextual field, then each activity guide instance will be created for a specific employee ID. Contextual data defined as non-key fields is not used to differentiate an instance, but can be passed to the instance nevertheless. For example, non-key data can be displayed in the activity guide pagelet by specifying the Context Visible option for the field. In this way, instead of displaying the employee ID, you could display the employee name in the pagelet.

Field or Control	***Description***
Record Name	Specify the record name that is the source for the contextual field.
Field Name	Specify the name of the source contextual field.

<table>
<tr><th>Field or Control</th><th>Description</th></tr>
<tr><td>Label</td><td>Enter a label for the contextual field. This label can be displayed along with the data value in the activity guide pagelet on the WorkCenter page.</td></tr>
<tr><td>Type</td><td>Select one of the following context types:
Fixed Value: Select to enter a fixed value as the context data.
Runtime: Select to have the context data value passed on the activity guide URL.<hr>Note: The following special characters are not recommended in context parameter values as they have special meaning in URLs: # & ? + = : % /
Additional restrictions may apply if other special characters are not acceptable in the components configured as action items—for example, the space character is not acceptable in key values for certain components. You may see errors in the target component when such restricted characters are used in context parameter values.<hr>
System Variable: Select to specify a system variable as the context data value. The value of the system variable is determined at runtime.</td></tr>
<tr><td>Value</td><td>Enter the value when the context type is fixed value. Alternatively, select the system variable from the look up list when the context type is system variable.<hr>Note: No value is entered when the context type is runtime.<hr></td></tr>
<tr><td>Key Field</td><td>Specify whether the contextual field is a key field.<hr>Note: Keys fields are used to uniquely identify an activity guide instance.<hr></td></tr>
<tr><td>Context Visible</td><td>Specify whether to display the contextual data and labels in the activity guide.</td></tr>
</table>

Field or Control	Description
Primary	For fluid activity guides only, specify whether this contextual data field is to be displayed as the primary field (that is, larger and in bold). **Note:** If the Label Visible option is selected for any field, then this selection is ignored.
Label Visible	Specify whether the labels are displayed in addition to the contextual data. **Note:** If this option is selected for any field, then labels are displayed for all visible contextual data fields.

Image Context Data

You can specify a static or dynamic image to be displayed in each activity guide instance. Depending on the selected value of the Image Type field, additional fields are displayed.

Field or Control	Description
Image Type	Select an image type: • *App Class* – Select an application package, class path, and method that will dynamically return the image URL at runtime based on the instance's context data. See Dynamically Returning an Image URL as Context Data for more information. • *Image Catalog* – Select a static image from the database for all activity guide instances. • *None* – No image is associated with the activity guide instances. • *Static* – Specify the fully qualified URL to a static image for all activity guide instances.
App Class: **Package**	Enter the name of the application package that contains the method to be invoked to dynamically return the image URL.
App Class: **Path**	Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Field or Control	*Description*
App Class: **Class ID**	Enter the name of the application class that contains the method to be invoked to dynamically return the image URL.
App Class: **Method Name**	Enter the name of the method to be invoked to dynamically return the image URL.
Image Catalog: **Image Name**	Select the static image from the database.
Static:**Static URL**	Specify the fully qualified URL to a static image.

Context Data Display Options

You can select an option for how the context data is displayed.

Field or Control	*Description*
Option	Select a context data display option: • *Fixed* – Displays up to three fixed rows of context data. • *Popup* – Displays one row of primary context data as a clickable link. When this link is clicked, up to three rows of context data are displayed in a modeless pop-up window. • *Modal* – Displays one row of primary context data as a clickable link. When this link is clicked, the URL returned by the associated method is opened in a modal window. See Dynamically Returning a Content URL to Be Displayed in a Modal Window for more information.
Modal: **Package**	Enter the name of the application package that contains the method to be invoked to dynamically return the URL for the content of the modal window.
Modal: **Path**	Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Modal: **Class ID**	Enter the name of the application class that contains the method to be invoked to dynamically return the URL for the content of the modal window.

Field or Control	Description
Modal: **Method Name**	Enter the name of the method to be invoked to dynamically return the URL for the content of the modal window.

Instance Creation

You can identify a method that will be executed whenever an activity guide instance is created from this template. When an instance is created from the template, the instance properties and action item details are initially set based on the template definitions. Then, use this method to initialize instance properties, participants, and action item properties (such as status, due date, assignment, and so on), or even to hide or show action items.

Field or Control	Description
Package	Enter the name of the application package that contains the method to be invoked at instance creation.
Path	Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class ID	Enter the name of the application class that contains the method to be invoked at instance creation.
Method Name	Enter the name of the method to be invoked to initialize the activity guide instance.

Configurable Fields

You can also define labels for fields that will appear on the Related Data page for an action item. In the action item definition, use these additional fields to associate data that will be used by the related content service at run time.

For example, a configurable field could be used to pass a run control ID to the related content service (and its underlying component, a run control page). A different action item definition—in the same template even—could be defined to supply a different run control ID to the same related content service.

Field or Control	Description
Text Fields	Enter the label for up to five configurable text fields.
Yes/No Fields	Enter the label for up to five configurable yes/no fields.

Usage Statistics

The Activity Guide framework enables you to compute the duration of time in which a user completes a step in the guided process and the duration of a guided process in its entirety. This data can be indexed and can be used for analytics, if required.

At the template level and at the action item or step level, the **Estimate Duration** fields are configurable. However, at the instance level, the framework computes the actual duration of a step or instance based on the completion of the step or instance.

Field or Control	*Description*
Estimate Duration	Specify the approximate duration in days that an user will take to complete the guided process.

For more information, see Defining Action Item Details and Reviewing Usage Statistics.

Related Links

Initializing an Activity Guide Instance

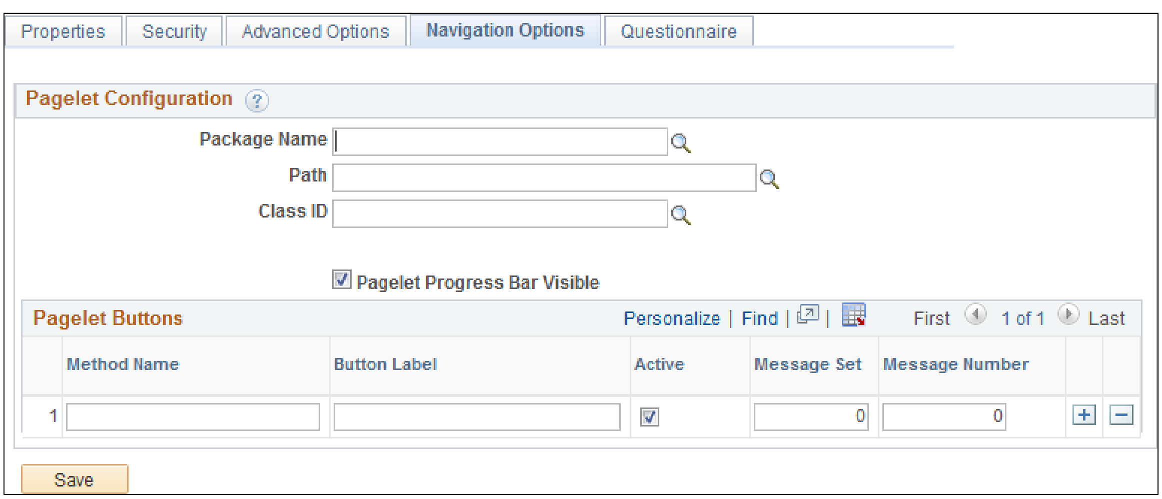
Defining Navigation and Pagelet Options

Use the Setup Activity Guide - Navigation Options page to add custom action elements and to define pagelet properties including pagelet pre-processing.

Navigation:

1. In the Setup Activity Guide WorkCenter, click the Template Properties task.
2. Select the Navigation Options page.

This example illustrates the fields and controls on the Template Properties: Navigation Options page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Package Name	Enter the name of the application package that contains the method (or methods) to be invoked to create custom action buttons.
Path	Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class ID	Enter the name of the application class that contains the method (or methods) to be invoked to create custom action buttons. In addition to or in lieu of navigation buttons, this application class can define a pagelet pre-processing method that will be executed every time the pagelet is loaded or reloaded in the WorkCenter pagelet area. This method *must* be named PageletPreProcess. See Performing Pre-Processing for the Activity Guide Pagelet for an example.
Pagelet Progress Bar Visible	Indicates whether the progress bar will be displayed in the activity guide pagelet on the WorkCenter page. **Note:** This option applies to classic activity guides only when accessed through a WorkCenter.

Pagelet Buttons

In fluid activity guides, these buttons are rendered in the activity guide subbanner. In classic activity guides, these buttons are rendered in the navigation frame of the classic WorkCenter.

Note: For classic activity guides, Next and Previous buttons are displayed in the navigation frame of the classic WorkCenter. Oracle recommends that you add no more than two additional custom action buttons to classic activity guides.

Field or Control	***Description***
Method Name	Enter the name of the method to be invoked to create the additional navigation button. **Important!** The method named here *must* exist in the selected class. No validation is performed to ensure this is the case when the template definition is saved. You can implement certain methods with reserved names (MarkComplete, for example). In addition, you can use certain reserved words (for example, PTAI_NAV_SAVE, PTAI_NAV_SAVE_NEXT, and others) as method names to invoke other delivered functionality. See Developing Custom Action Buttons for detailed information.
Button Label	Enter the label for the custom action button as it will be displayed to users. **Note:** Button labels are ignored for the following reserved word methods: PTAI_NAV_AUTO_SAVE, PTAI_NAV_SAVE_NEXT, PTAI_NAV_SAVE_PREV, and ExitAGProcess.
Active	Select this option to retain the button definition as active.
Message Set	Specify the message set containing an optional message to be displayed after the user clicks the button.
Message Number	Specify the message number containing an optional message to be displayed after the user clicks the button and before invoking the specified method. For example, if the user clicks a Cancel button to cancel an activity guide instance, this could display a confirmation message.

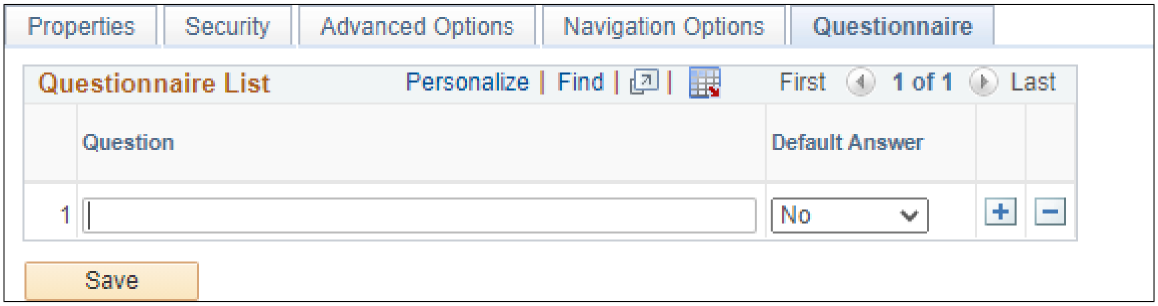
Defining an Activity Guide Questionnaire

Use the Template Properties: Questionnaire page to define the questions that can be associated with action items.

Navigation:

1. In the Setup Activity Guide WorkCenter, click the Template Properties task.
2. Select the Questionnaire page.

This example illustrates the fields and controls on the Template Properties: Questionnaire page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Question	Enter the complete text for the question. **Important!** Only questions with a yes or no answer are supported.
Default Answer	Select a default answer for the question.

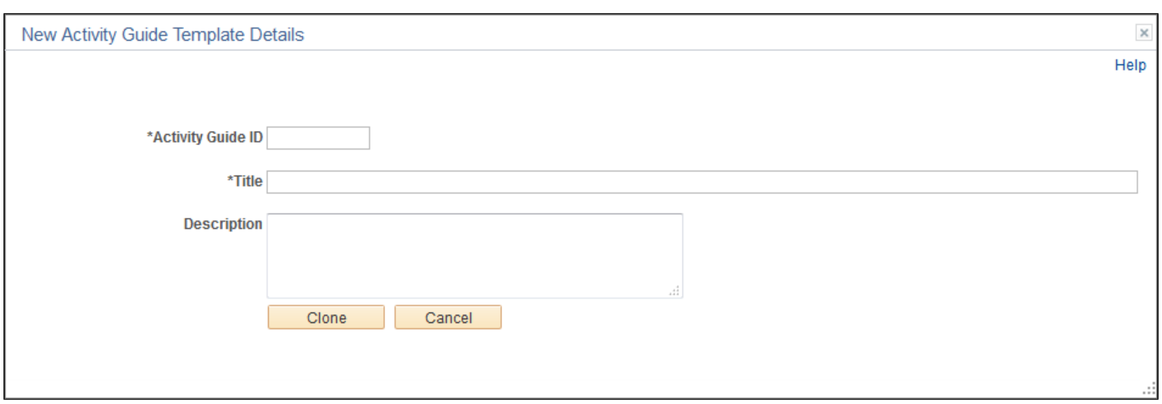
Copying Activity Guide Templates

Use the New Activity Guide Template Details page to create a copy of the selected activity guide template.

Navigation:

1. **PeopleTools** > **Activity Guides** > **Activity Guide Templates**
2. Click the Clone Template link for an activity guide template.

This example illustrates the fields and controls on the New Activity Guide Template Details page. You can find definitions for the fields and controls later on this page.



Note: When you copy an activity guide template, the IDs for all action items will be updated to new and unique values in the new template.

Field or Control	***Description***
Activity Guide ID	Enter an unique identifier for the template. When you save the template, the system will fill characters up to the 10-character maximum.
Title	Enter a name for the template.
Description	Enter a description for the template.
Clone	Click the Clone button to create a new activity guide template.

Creating and Maintaining Action Items

This section describes how to create and maintain action items.

Understanding Creation and Maintenance of Action Items

Action items are the tasks that are assigned for completion within an activity guide. As an application developer, you will refer to these as action items. To users, they are presented as tasks or steps.

Note: The horizontal optimized guided sequential template type supports a simple structure with one level of action items only; summary/detail action items are not supported by this type.

Action items can be linked to specific local or remote components, PeopleSoft queries, PeopleCode programs (an application class or an iScript), self-service process definitions, or external URLs. Within an activity guide, action items are organized into an ordered list; however, in many simple activity guides, action items can be completed in a non-sequential order. Action items can be organized hierarchically with a parent (summary) action item composed of child (detail) action items. Furthermore, an action item can be set to be dependent on the status of another action item.

Typically, action items are created and maintained at the activity guide template level only. However, you can also create action items within activity guide instances; the same pages with the same fields are used in both cases. Once an activity guide instance is created, updates to the action items themselves would occur programmatically or through actions of the user rather than through these administrative pages. While the examples in this section illustrate creating and maintaining action items within an activity guide template, these examples could apply equally to activity guide instances.

Sequence Numbers

You must designate a sequence number for each action item, which determines its order within its designated level—that is, at the root level or as a detail action item beneath a summary item. In addition, if you want to make an action dependent on another action item, that action item must be a predecessor —that is, it must have a lower sequence number than the dependent action item. Each sequence number must be unique within an activity guide template or instance.

While sequence numbers define the order in which action items are presented to a user, they do not in and of themselves dictate that tasks be completed in sequence. Additional attributes, such as dependencies, must be defined to ensure that tasks be completed in a specific sequence when that is required. Therefore, the simplest activity guide definitions allow for non-sequential completion of tasks.

Note: Activity guides support a two-level hierarchy only—that is, only one level of detail is supported beneath a summary action item.

With a little planning, you can specify action item sequence numbers that clearly specify the order of tasks and that allow modifications to the list at a latter date—for example, to add or delete action items. Oracle recommends that you specify initial sequence numbers in units of 10 to permit insertion of new action items into an existing sequence. For example, for a simple activity guide with no summary action items, the initial sequence would be:

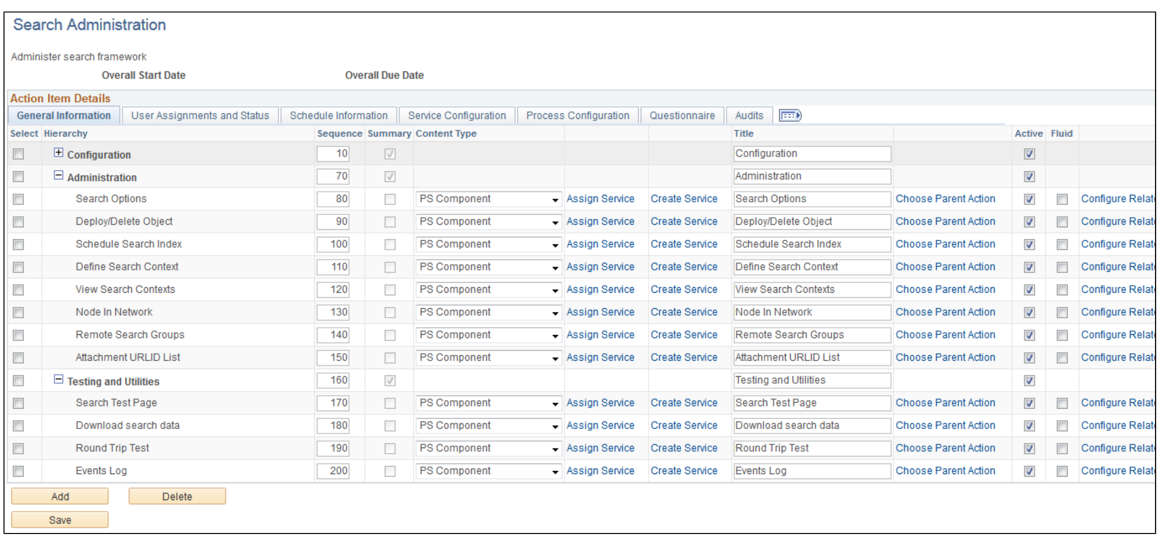
```
10, 20, 30, 40 ...
```

If your activity guide contains summary as well as detail items, then increment the root-level items by 100 instead of by 10. For example:

```
100 - Summary item 1
  110, 120, 130, 140, ...
200
300
400 - Summary item 2
  410, 420, 430, 440, ...
500
```

Designating the initial sequence numbers in this manner will allow you to easily convert an existing root-level action item into a summary item without having to renumber the rest of the sequence.

The following example shows sequence numbers for summary and detail action items. In this list, Configuration (10, also collapsed), Administration (70), and Testing and Utilities (160) are parent, or summary, items:



Maintaining the List of Action Items

Use the Action Item Details grid on the action item list page to maintain the list of action items.

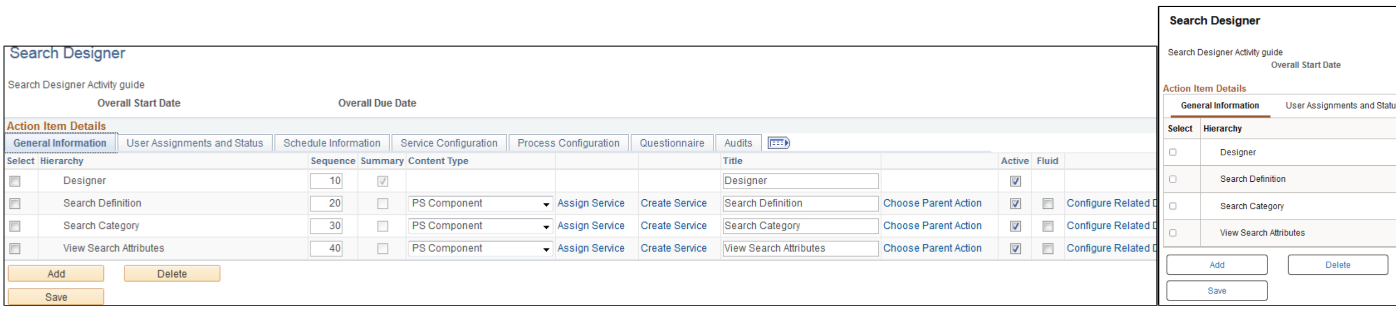
Navigation:

- On the Manage Activity Guide Templates page, click the title link for an existing template.

  Note: In addition, on the Manage Activity Guide Instances page, click the title link for an existing instance.

- In the Setup Activity Guide WorkCenter, click the Configure Action Items task.

This example illustrates the fields and controls on the action item list page for an activity guide template. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Overall Start Date	Displays the start date of the activity guide template (as defined on the Properties page).
Overall Due Date	Displays the due date of the activity guide template (as defined on the Properties page).
Select	Select an action item to delete. Selecting a summary action item also selects any detail action items associated with it.
Add	Click the Add button to access the Add an Action Item page to create a new action item.
Delete	Click the Delete button to delete the selected action items. **Note:** There are no checks to prevent you from deleting an action item from an activity guide instance no matter its current status or relationship to other action items in the instance.
Save	Click the Save button to save updates to any action item details.

Adding Action Items

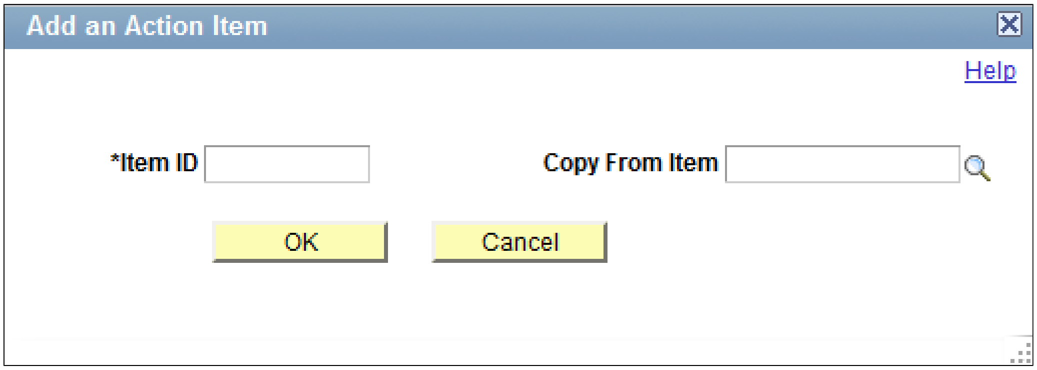
Use the Add an Action Item page to add a new action item.

Navigation:

1. In the Setup Activity Guide WorkCenter, click the Configure Action Items task.

2. Click the Add button.

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



Field or Control	***Description***
Item ID	Enter up to 10 characters as an identifier prefix. When you save the action item, the system will append digits to ensure that this identifier is unique within the template. **Important!** If you specify an identifier prefix that is already in use in another template, an error is generated.
Copy From Item	Select the ID for an existing action item within this template. This allows you to select an existing action item from this template to clone as a new action item. You can then modify the properties of the new action item.

Defining Action Item Details

Use the following tabs of the Action Item Details grid to define the details for each action item:

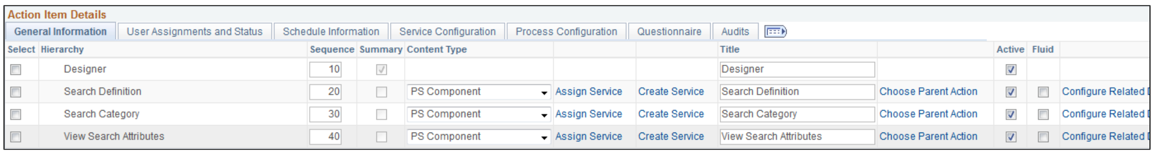
- Use the **General Information tab** to specify the title of the action item, assign a related content service (which can also be assigned on the Service Configuration tab), choose a parent item, and configure related data.
- Use the **User Assignments and Status tab** to maintain the action item's status and to whom it is assigned.
- Use the **Schedule Information tab** to define the start date and time and due date and time for each action item.
- Use the **Service Configuration tab** to associate related content service definitions with action items.
- Use the **Process Configuration tab** to define the self-service process details for action items defined as a *Process* type.

- Use the **Usage Statistics tab** to specify an approximate duration (in days) for each action item.
- Use the **Questionnaire tab** to associate questions defined on the template to one or more action items.
- Use the **Audits tab** to review audit information for each action item including a change log.

General Information Tab

Use the General Information tab to specify the title of the action item, assign a related content service (which can also be assigned on the Service Configuration tab), choose a parent item, and configure related data.

This example illustrates the fields and controls on the Action Item Details - General Information tab for an activity guide template. You can find definitions for the fields and controls later on this page.



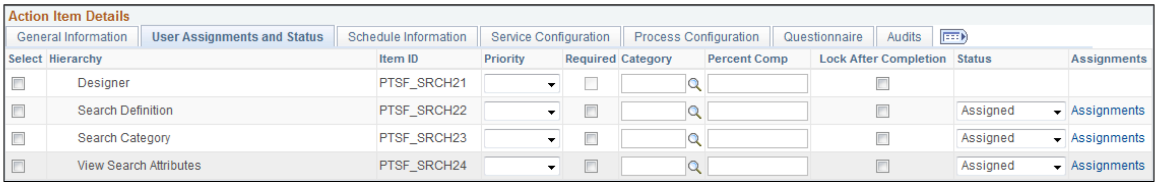
Field or Control	*Description*
Hierarchy	Displays the summary-detail (parent-child) relationship of the action items in the template or instance.
Sequence	Enter up to a four-digit sequence number to indicate this item's position in the ordered list of action items. A sequence number must be unique within an activity guide template or instance. See "Sequence Numbers" in Understanding Creation and Maintenance of Action Items.
Summary	Select this option to indicate that this is a summary action item. **Important!** Important! You must select this option first and then save the action item. If you save the action item without selecting this option it will be created as a child action. Once an action item has been saved, it cannot be changed from a summary item to a child item, or vice versa.
Content Type	Select the content type for the action item. This column is duplicated on the Service Configuration tab and the content types are documented there. **Note:** The column is also editable on the Process Configuration tab.

Field or Control	*Description*
Assign Service	Click this link to select an existing related content service definition of the specified content type. **Note:** This function is duplicated by the lookup button for the Service ID field on the Service Configuration tab
Create Service	Click the this link to create a new related content service definition for a component-based related content service only. See Creating a Component-Based Service Definition for more information. **Note:** This link is also displayed on the Service Configuration tab.
Title	Enter the title for the action item. **Note:** The default value is the name of the selected related content service.
Choose Parent Action	Click this to access the Look Up Parent Action Item page to change the parent for an action item within the hierarchy.
Active	Select this option to indicate that the action item is active. Clear this option to make an action item inactive. The action item definition remains in the template, and is copied to any instances generated from this template. However, an inactive action item will not be displayed to users.
Fluid	Select this option to indicate that the action item is associated with a fluid component. **Note:** For optimized activity guide types, this option is selected automatically for each action item. Deselect this option for an action item associated with a classic component.
Configure Related Data	Click this link to configure dependencies, related data, and pagelet button overrides for this action item. See Configuring Related Data for more information.

User Assignments and Status Tab

Use the User Assignments and Status tab to maintain the action item's status and to whom it is assigned.

This example illustrates the fields and controls on the Action Item Details - User Assignments and Status tab. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Item ID	Displays the ID for the action item.
Priority	Select a priority for the action item: *High*, *Low*, or *Medium*.
Required	Select this option to indicate that an action item is required. For the status of an activity guide instance to be designated as complete, all required action items must have a complete status. Similarly, for a summary action item to be designated as complete, all required detail action items under it must have a complete status.
Category	Enter an optional category for this action item. **Note:** If the category does not exist already, it will be created for you when you save the action item.
Percent Comp	Specify a percent completion for this action item.
Lock After Completion	Select this option to prevent the action item from being updated after it has been marked complete.
Status	Select a value for the status of the action item. Typically, in an activity guide template, the initial value should be *Assigned*. **Note:** When viewing the action item list in an activity guide instance, the status for action items will be updated dynamically through user actions.
Assignments	Click the Assignments link to assign the action item on the Action Item Assignments page.

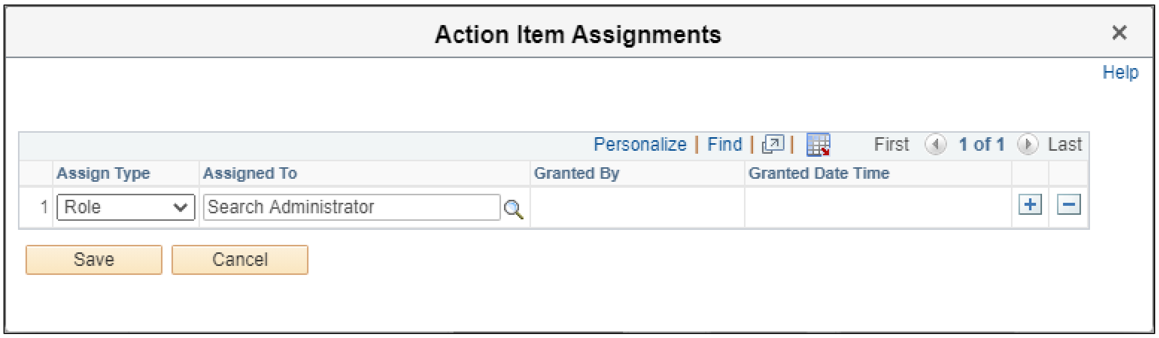
Action Item Assignments Page

Use the Action Item Assignments page to assign one or more users, roles, or a combination to the action item.

Navigation:

Click the Assignments link on the User Assignments and Status tab of the Action Item Details grid.

This example illustrates the fields and controls on the Action Item Assignments page. You can find definitions for the fields and controls later on this page.



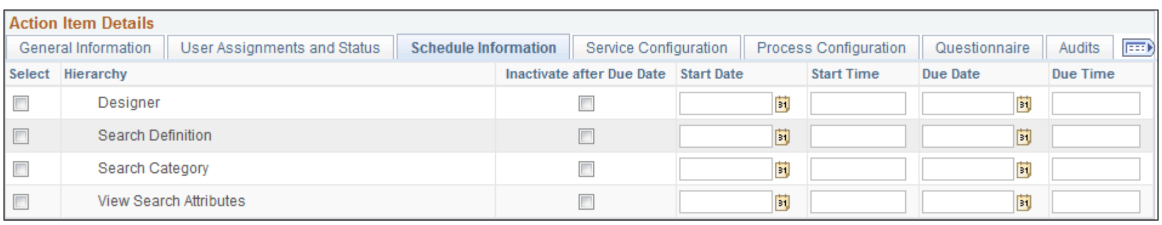
Field or Control	***Description***
Assign Type	Select the type of participant that you want to assign the action item to. Available values include: • *Role* — Select to be able to select a role in the Assigned To field. • *User* — Select to be able to select a user in the Assigned To field.
Assigned To	Select an assignee from the list of all users or roles in the system. **Important!** While you can select any user or role, only a user or role that has also been assigned *Contributor* privileges on the template or instance Security page will be allowed to complete the action item. The system will not validate your selection against the users or roles defined on the Security page.

Field or Control	***Description***
Granted By	If the custom Grant button has been enabled, this field displays the ID of the user who granted access to this action item. Only instance administrators can revoke a grant by removing rows from this table. **Note:** This field is populated in activity guide instances only. This field should be populated by the system only, and never programmatically.
Granted Date Time	If the custom Grant button has been enabled, this field displays the date and time the grant was made. **Note:** This field is populated in activity guide instances only.

Schedule Information Tab

Use the Schedule Information tab to define the start date and time and due date and time for each action item.

This example illustrates the fields and controls on the Schedule Information tab. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Inactivate after Due Date	Select this option to make the action item unavailable after the specified due date has passed. **Note:** An administrator can manually reset the action item in the activity guide instance.
Start Date	Select a start date from the calender, or enter a start date in the format specified by your user personalizations.
Start Time	Enter a start time.

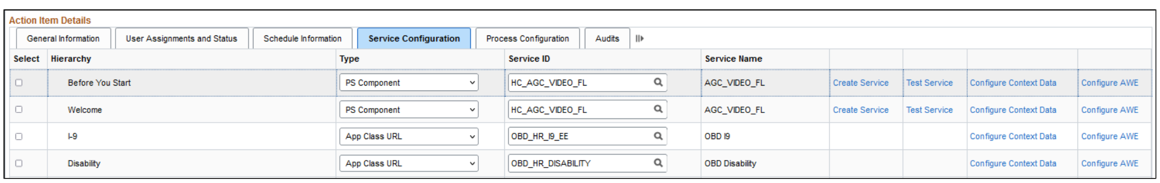
Field or Control	***Description***
Due Date	Select a due date from the calender, or enter a due date in the format specified by your user personalizations.
Due Time	Enter a due time.

Service Configuration Tab

Use the Service Configuration tab to associate related content service definitions with action items.

Important! Action items can be linked to specific local or remote components, PeopleSoft queries, PeopleCode programs (an application class or an iScript), self-service process definitions, or external URLs. Except for self-service process definitions, these links are made through the PeopleSoft Related Content Framework. Moreover, except for component-based related content service definitions, which can be created by clicking the Create Service link in the Action Item Details grid, you *must* explicitly use the Define Related Content Service page to create any other related content service definition prior to associating that service definition with an action item.
See Creating and Managing Related Content Service Definitions for more information.

This example illustrates the fields and controls on the Action Item Details - Service Configuration tab. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Type	**Note:** This column is also editable on the General Information tab and the Process Configuration tab. Select one of the following types: • *App Class URL –* Use this type when you wish to perform any pre-processing prior to displaying the transaction to the user. Also, use this type when the transaction associated with the action item is not known at design time and needs to be determined at runtime. In both cases, the method specified in the service definition must redirect to the URL of the target transaction. See Performing Pre-Processing for Action Items for coding guidelines and an example method. • *Non PeopleSoft URL –* Use this type when the user's task must be performed on a non-PeopleSoft system. In this case, because there is no callback mechanism from the non-PeopleSoft system, the user will have to manually mark the action item as complete. **Note:** The External URL in the Define Related Content Service page for a non-PeopleSoft URL must contain the additional parameter, *PTAI=T*. For example, `http://example.com/2014/09/15/technology/mobile/moto-x/index.html?iid=TL_Popular&PTAI=T`. **Important!** You will need to define a custom button that the user can click to manually mark this action item complete. • *PS Component –* Use this type when the action needs to be performed on a PeopleSoft transaction page. The page can reside on either a local or remote node. • *PS Query –* Use this type when the action to be performed is running a PeopleSoft query. For example, a manager may be assigned an action item to run a query to determine which issues are open on a

Field or Control	***Description***
	project. In this case, the manager will have to manually mark the action item as complete. **Important!** You will need to define a custom button that the user can click to manually mark this action item complete. • *PS Script –* Use this type when the transaction associated with the action item is not known at design time and needs to be determined at run time. As well, you can use this type when you wish to perform any pre-processing prior to displaying the transaction to the user. In both cases, the iScript specified in the service definition must specify and transfer the user to a target transaction. • *Process –* Use this type to execute a self-service process automatically. Self-service processes can be local or remote. See the following "Process Configuration Tab" section for more information on configuring process type action items.
Service ID	Specify the ID for an *existing* related content service that will be executed for this action item. **Note:** The related content service definitions available for selection are filtered by the type selected. **Note:** This field is equivalent to the Assign Service link on the General Information tab.
Service Name	Displays the label of the service.
Create Service	Click the this link to create a new related content service definition for a component-based related content service only. See Creating a Component-Based Service Definition for more information. **Note:** This link is also displayed on the General Information tab.

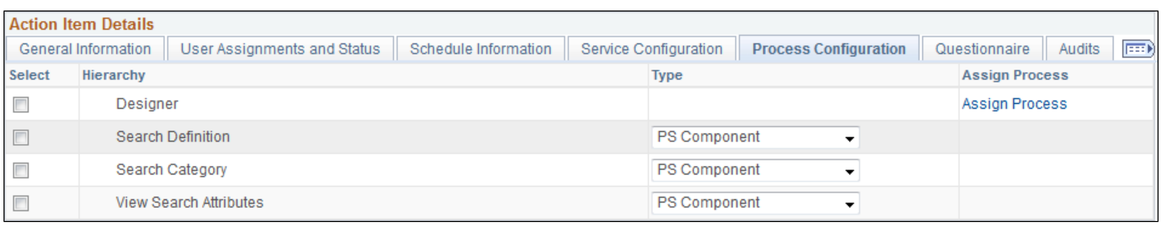
Field or Control	***Description***
Test Service	Click this link to test the related content service. See Testing Related Content Service Definitions for more information.
Configure Context Data	Click this link to define additional contextual data that will be passed to the related content service at runtime. See Configuring Additional Context Data.
Configure AWE	Click this link to associate approvals with action item. See Associating Approvals with Action Item.

Process Configuration Tab

Use the Process Configuration tab to define the self-service process details for action items defined as a *Process* type. Self-service processes are executed automatically either at instance creation or when all predecessor action items configured with a finish-to-start dependency are marked complete.

For more information about creating, selecting, and configuring self-service processes, see Selecting and Configuring Self-Service Processes.

This example illustrates the fields and controls on the Action Item Details - Process Configuration tab. You can find definitions for the fields and controls later on this page.



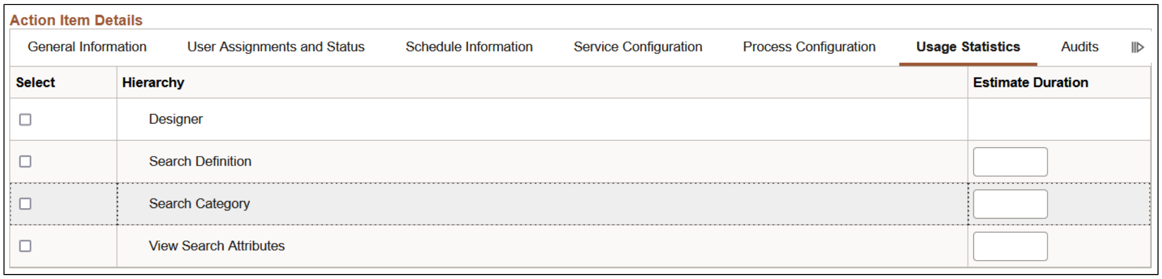
Field or Control	***Description***
Type	Select *Process* to configure a self-service process as an action item. **Note:** The other values for this column are documented on the Service Configuration tab. This column is also editable on the General Information tab.
Assign Process	Click this link to select a process. See Selecting and Configuring Self-Service Processes for more information.

Field or Control	Description
Target Node	Displays the source node for the process.
Schedule Name	Displays the schedule name associated with the action item.
Process Run Status	Displays the status of the process.
Pause Before	Select this option to pause the process before scheduling the process. For example, you may pause the process to validate or modify the input parameters before scheduling the process.
Pause After	Select this option to pause the process after the completion of process. For example, You may validate the process output before proceeding to the next.
Skip on Error	Select this option to enable display of the process status as Complete even if a process item runs into Error or No Success.
Resume on Error	Select this option to allow the users to reschedule process if it had stopped due to an error.
Configure Input Parameters	Select this option to enable the end-users to modify the run control parameters. This activates the Configure CI Parameters link that enable you to access the Configure CI Parameters page. **Note:** This option is valid only if the **Pause Before** option is selected.
Configure CI Parameters	Click this link to configure the component interface parameters. See Selecting and Configuring Self-Service Processes for more information.

Usage Statistics Tab

Use the Usage Statistics tab to specify an approximate duration (in days) for each action item.

This example illustrates the fields and controls on the Action Item Details - Usage Statistics tab. You can find definitions for the fields and controls later on this page.



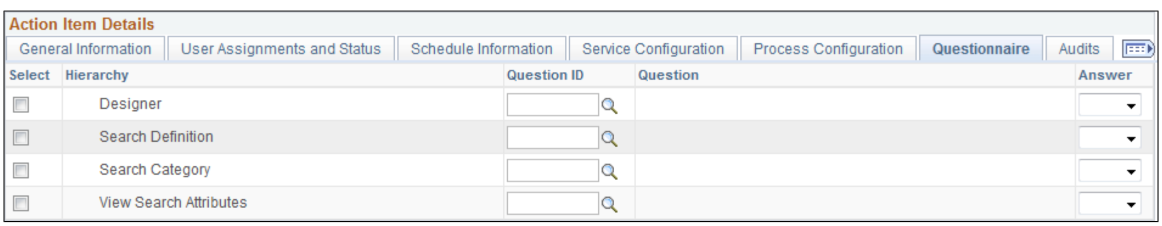
Field or Control	Description
Estimate Duration	Specify an approximate duration (in days) required to complete the action item or step.

For more information, see Defining Advanced Options and Reviewing Usage Statistics.

Questionnaire Tab

Use the Questionnaire tab to associate questions defined on the template to one or more action items.

This example illustrates the fields and controls on the Action Item Details - Questionnaire tab. You can find definitions for the fields and controls later on this page.

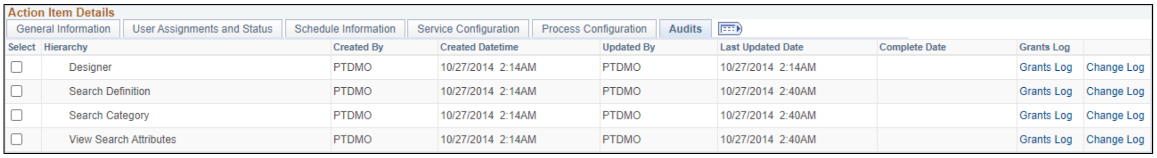


Field or Control	Description
Question ID	Select a defined question to associate with this action item. **Note:** If a question is associated with a summary item, then all the detail actions under that summary are made active or inactive based on a single answer.
Answer	Select the answer (*Yes* or *No*) that if provided by the user will set this action item to active in the activity guide instance.

Audits Tab

Use the Audits tab to review audit information for each action item including a change log.

This example illustrates the fields and controls on the Action Item Details - Audits tab. You can find definitions for the fields and controls later on this page.



Action Item Details

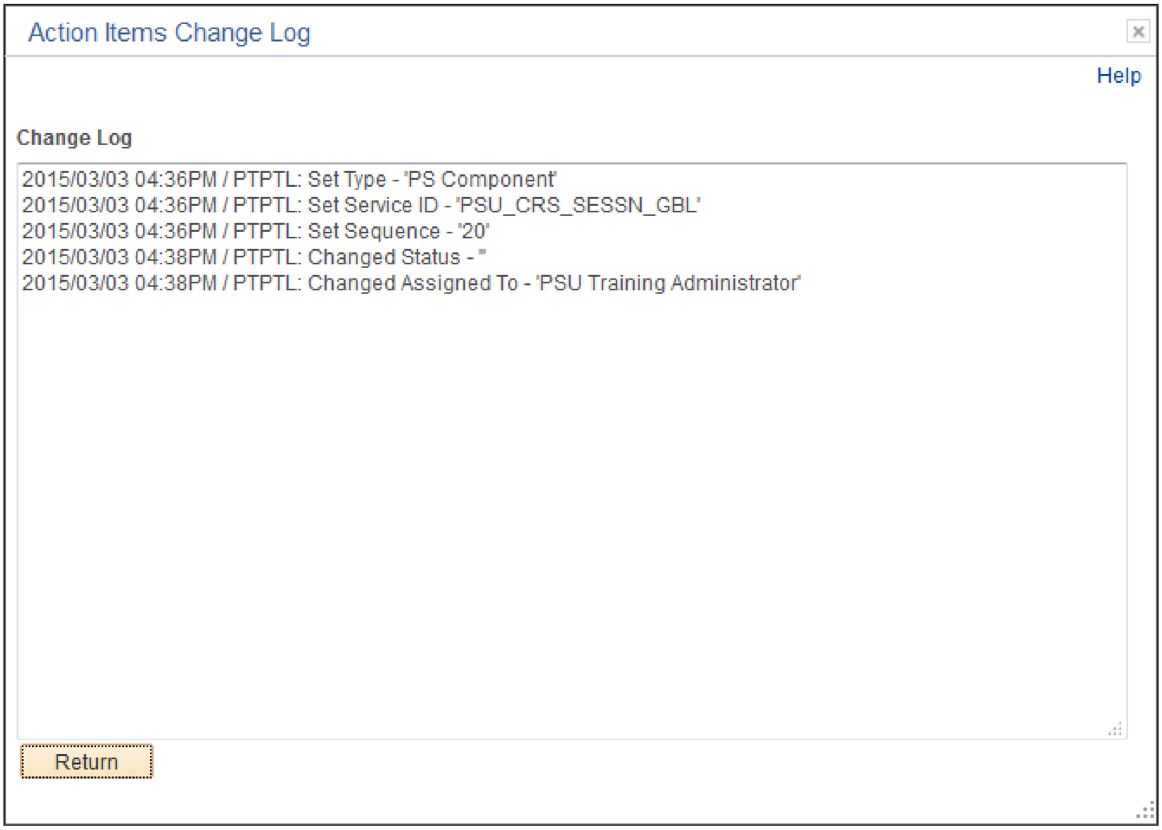
General Information | User Assignments and Status | Schedule Information | Service Configuration | Process Configuration | Audits

Select	Hierarchy	Created By	Created Datetime	Updated By	Last Updated Date	Complete Date	Grants Log	
☐	Designer	PTDMO	10/27/2014 2:14AM	PTDMO	10/27/2014 2:14AM		Grants Log	Change Log
☐	Search Definition	PTDMO	10/27/2014 2:14AM	PTDMO	10/27/2014 2:40AM		Grants Log	Change Log
☐	Search Category	PTDMO	10/27/2014 2:14AM	PTDMO	10/27/2014 2:40AM		Grants Log	Change Log
☐	View Search Attributes	PTDMO	10/27/2014 2:14AM	PTDMO	10/27/2014 2:40AM		Grants Log	Change Log

Field or Control	***Description***
Created By	Displays the user who created the action item definition.
Created Datetime	Displays the date and time when this action item was created.
Updated By	Displays the user who last updated the action item definition.
Last Updated Date	Displays the date and time when the action item was last updated.
Grants Log	Click this link to view the action item’s grants log. **Note:** Grant data is runtime data and is therefore only available in action item instances, and not in templates.
Change Log	Click this link to view the action item’s change log.

Use the Action Items Change Log page to view the changes to the action item definition.

This example illustrates the fields and controls on the Action Items Change Log page.



Use the Action Items Grants Log page to view grants and revocation of grants for the action item.

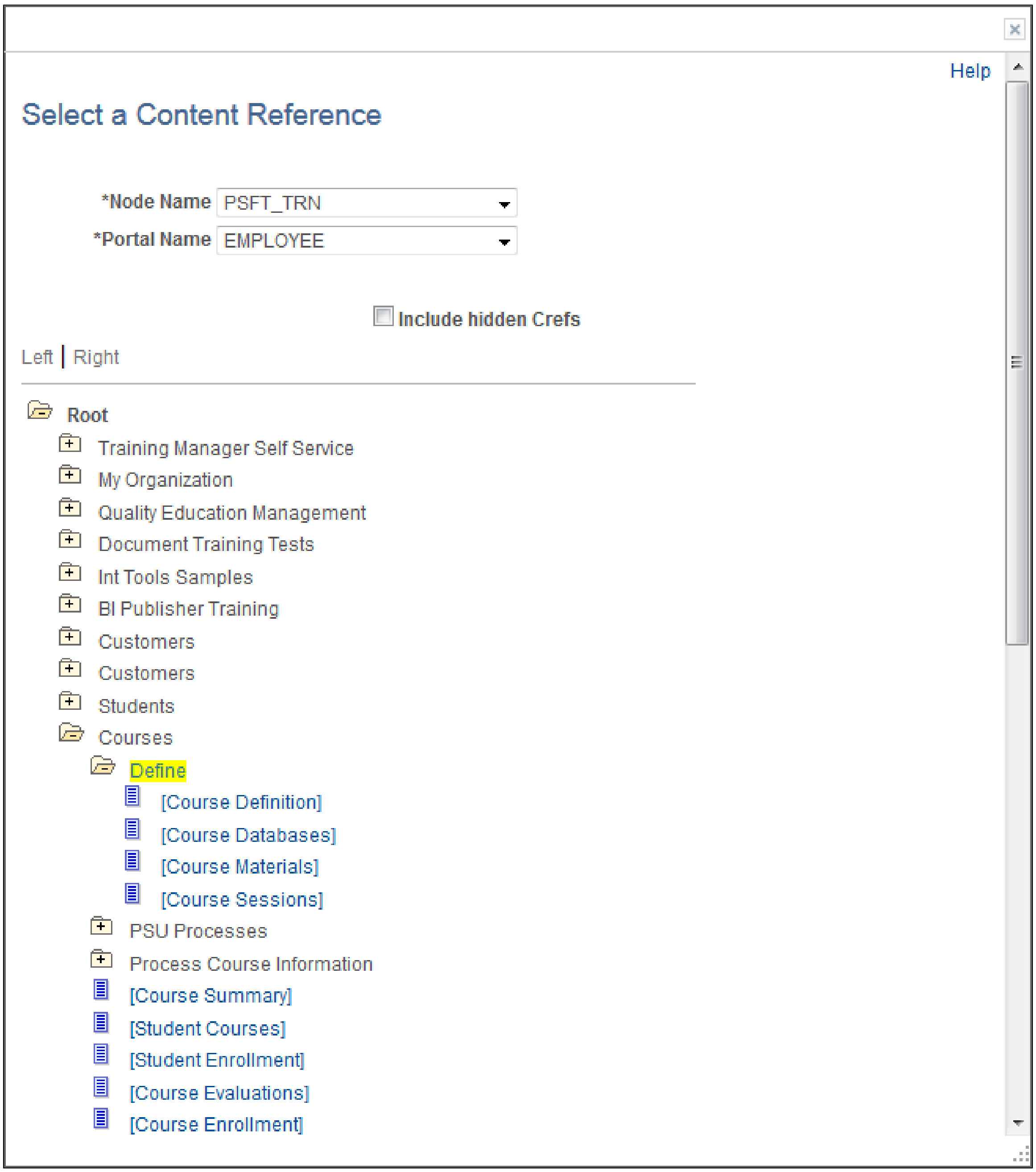
This example illustrates the fields and controls on the Action Items Grants Log page.

Creating a Component-Based Service Definition

To create a component-based service definition:

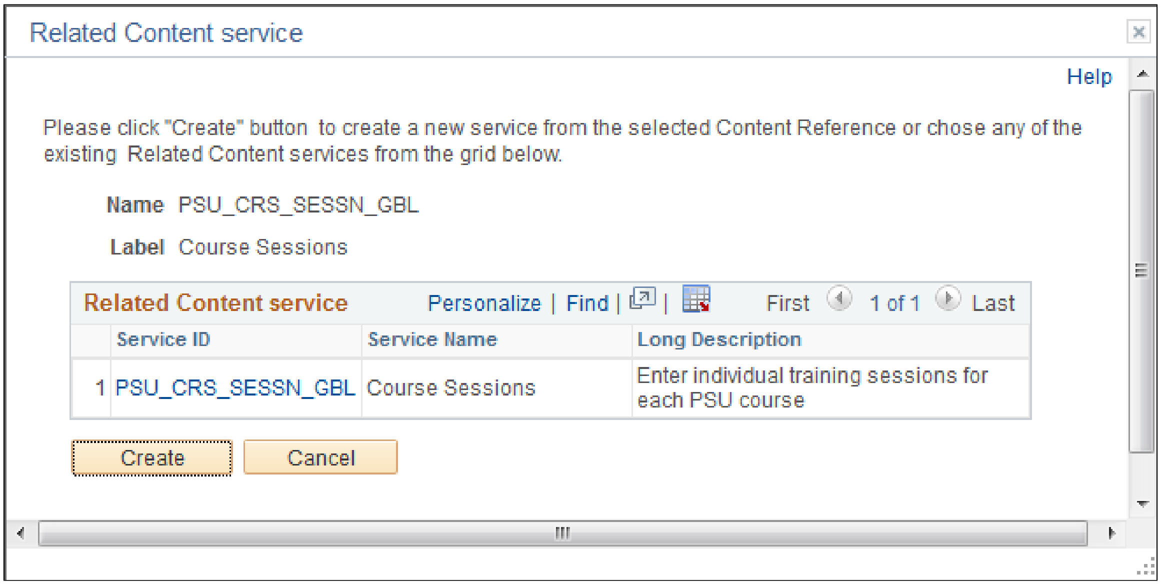
1. In the Action Item Details grid, select *PS Component* as the content type.
2. Click the Create Service link.
3. Use the Select a Content Reference page to first select the component.

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



4. Then, use the Related Content service page to confirm your selection and create the service definition.

This example illustrates the fields and controls on the Related Content service page.



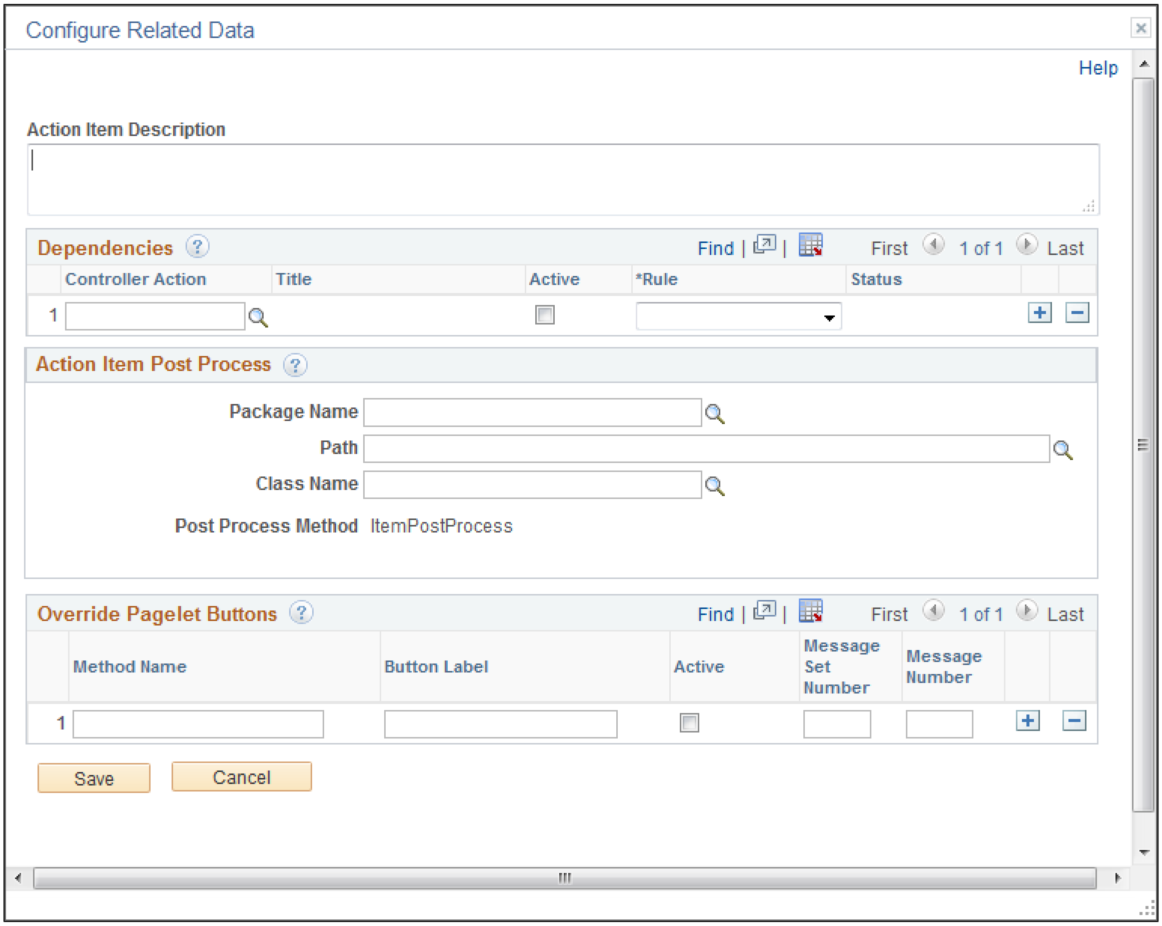
Configuring Related Data

Use the Configure Related Data page to specify dependencies to other action items, to specify post-processing, to create (or override) navigation buttons for this action item, and to specify values for related data fields.

Navigation:

Click the Configure Related Data link on the General Information tab of the Action Item Details grid.

This example illustrates the fields and controls on the Configure Related Data page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Action Item Description	Enter any additional description about the action item.

Dependencies

Use the Dependencies grid to designate controller action items upon which this action item is dependent. The finish-to-start dependency rule means that the controller action item must be completed before this (the dependent) action item is allowed to start.

Field or Control	*Description*
Controller Action	Select a controller action item. The list of action items available for selection is filtered to include only those action items with a lower sequence number than the current action item and to exclude completed or cancelled action items.
Title	Displays the title for the controller action item.

Field or Control	***Description***
Active	Select this option to retain the dependency as active.
Rule	Select the from the following rule types: • *Finish to Start* – Specifies that the controller action item must finish before this action item is allowed to start. In addition, in sequential activity guides, use this rule to specify that the controller action item is required. • *Start to Start* – Specifies that the controller action item must start before this action item is allowed to start. In addition, in sequential activity guides, use this rule to specify that the controller action item is not required.
Status	Displays the current status of the controller action item.

Action Item Post Process

Use the Action Item Post Process group box to specify an application package and class that contains the required method to be executed as post-processing when a user completes the transaction—for example, to update the status of the action item to complete. In addition, this application class can optionally define custom action buttons on a per action item basis.

Post-processing can occur on the local system or the remote system if that's where the transaction is hosted. Post-processing is executed through component save processing only. If you have implemented a custom save button on the page to save the component, the button's page field properties must be configured to invoke the toolbar save action or the PeopleCode executed by these buttons must include either the DoSave or DoSaveNow built-in functions to trigger the component save events; otherwise, the post-processing specified here will not be executed.

Field or Control	***Description***
Package Name	Enter the name of the application package that contains the method to be invoked for post-processing and the methods to be invoked for any optional, custom action buttons.
Path	Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.

Field or Control	*Description*
Class Name	Enter the name of the application class that contains the method to be invoked for post-processing and the methods to be invoked for any optional, custom action buttons. **Important!** This class should extend the PT_ RCF:ServiceAGInterface interface class and must implement a method named ItemPostProcess, which is the method that will be executed for post-processing. If you do not require any post-processing but you want to implement custom action buttons for the action item, you must still implement a stub method named ItemPostProcess. Otherwise, an error will occur and save processing will be blocked for this action item.

Related Data

The Related Data group box and fields are displayed only when configurable field labels have been defined for the activity guide template or instance. The displayed field labels allow you to associate application data with an action item at either the template level or the instance level.

In the following example, a configurable field labeled Session was defined for the activity guide template. Here, at the action item level, the Session field is displayed along with instance-specific data (the value *801*, providing one methodology for passing data from one action item to the next in the instance:

Override Pagelet Buttons

The Override Pagelet Buttons group box allows you to create custom action buttons on a per action item basis. If the name of any of these buttons is the same as a button defined at the template level, then the effect is to override the definition with the method defined here at the action item level. If the name of any of these buttons differs from buttons defined at the template level, then the button is created in addition to those defined at the template level.

In fluid activity guides, these buttons are rendered in the activity guide subbanner. In classic activity guides, these buttons are rendered in the navigation frame of the classic WorkCenter.

Note: For classic activity guides, Next and Previous buttons are displayed in the navigation frame of the classic WorkCenter. Oracle recommends that you add no more than two additional custom action buttons to classic activity guides.

Field or Control	***Description***
Method Name	Enter the name of the method to be invoked to create the additional navigation button. **Important!** The method named here *must* exist in the selected class. No validation is performed to ensure this is the case when the template definition is saved. You can implement certain methods with reserved names (MarkComplete, for example). In addition, you can use certain reserved words (for example, PTAI_NAV_SAVE, PTAI_NAV_SAVE_NEXT, and others) as method names to invoke other delivered functionality. See Developing Custom Action Buttons for detailed information.
Button Label	Enter the label for the custom action button as it will be displayed to users. **Note:** Button labels are ignored for the following reserved word methods: PTAI_NAV_AUTO_SAVE, PTAI_NAV_SAVE_NEXT, PTAI_NAV_SAVE_PREV, and ExitAGProcess.
Active	Select this option to retain the button definition as active.
Message Set	Specify the message set containing an optional message to be displayed after the user clicks the button.
Message Number	Specify the message number containing an optional message to be displayed after the user clicks the button and before invoking the specified method. For example, if the user clicks a Cancel button to cancel an activity guide instance, this could display a confirmation message.

Configuring Additional Context Data

Use the Action Item Context Data page to define service parameters for the related content service.

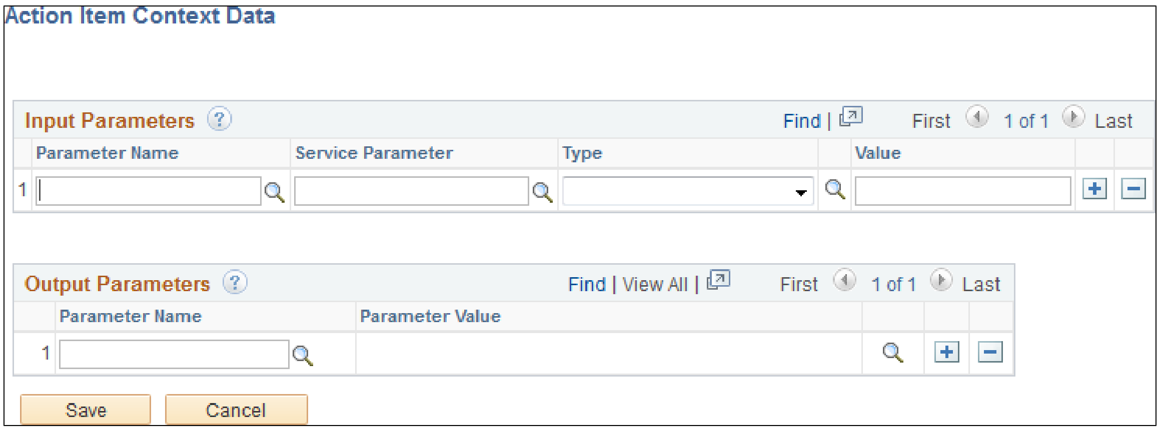
Navigation:

Click the Configure Context Data link on the Service Configuration tab of the Action Item Details grid.

Context data that is defined as a key field at the template (or instance) level is automatically passed to each related content service. However, an activity guide definition can include additional non-key context data. Use the Action Item Context Data page to map these additional context data fields to related content service parameters as needed on a per action item basis.

In addition, for activity guides defined as a single unit of work, one or more output parameters defined for one action item can serve as the input parameters to subsequent action items.

This example illustrates the fields and controls on the Action Item Context Data page. You can find definitions for the fields and controls later on this page.



Input Parameters

Map additional non-key, context data fields to related content service parameters as needed on a per action item basis. In addition to non-key context data fields, you can select from output parameters defined on predecessor steps. At runtime, the values of the action item input parameters are passed to the associated related content service.

Field or Control	***Description***
Parameter Name	Specify the parameter name in one of two ways: • Select an existing context parameter name (as defined on the Template Properties: Advanced Options page). • Enter a free-form parameter name. In this case, you must explicitly add the parameter to the URL used to access the activity guide.
Service Parameter	Select the service parameter (typically, a component key) for the related content service.

<table>
<tr><th>Field or Control</th><th>Description</th></tr>
<tr><td>Type</td><td>Select one of the following context types:
Fixed Value: Select to enter a fixed value as the context data.
Runtime: Select to have the system determine the context data value at runtime.

Note: The following special characters are not recommended in context parameter values as they have special meaning in URLs: # & ? + = : % /
Additional restrictions may apply if other special characters are not acceptable in the components configured as action items—for example, the space character is not acceptable in key values for certain components. You may see errors in the target component when such restricted characters are used in context parameter values.
System Variable: Select to specify a system variable as the context data value. The value of the system variable is determined at runtime.</td></tr>
<tr><td>Value</td><td>Enter the value when the context type is fixed value. Alternatively, select the system variable from the look up list when the context type is system variable.

Note: No value is entered when the context type is runtime.</td></tr>
</table>

Output Parameters

Note: The Output Parameters section is displayed only when the activity guide is defined as a single unit of work.

Select one or more page field to serve as output parameters for this action item. The values of the selected page fields are available to subsequent action items as input parameters.

Field or Control	***Description***
Parameter Name	Select the page field name to serve as an output parameter.

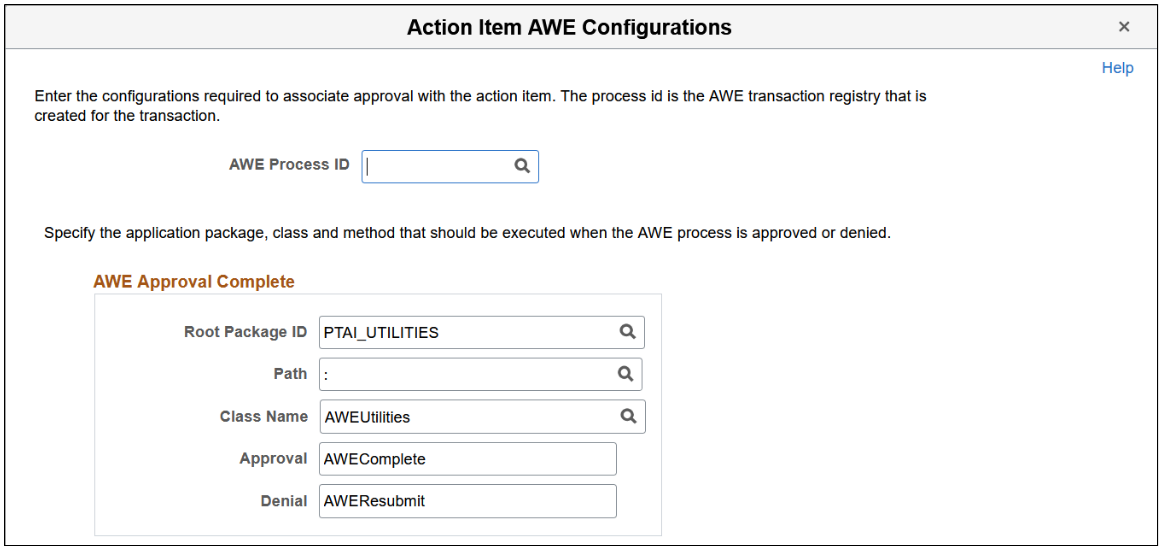
Associating Approvals with Action Item

Use the Action Item AWE Configurations page to associate approvals with action item and specify application package class and methods that should run when the AWE process is approved or denied.

Navigation:

Click the Configure AWE link on the Service Configuration tab of the Action Item Details grid.

This example illustrates the fields and controls on the Action Item AWE Configurations page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
AWE Process ID	Select an existing process ID from the AWE transaction registry that is created for this transaction.
Root Package ID	Select or enter the name of the application package that you want to use for the AWE process.
Path	Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class Name	Select or enter the name of the application class that you want to use for the AWE process.
Approval	Select or enter the name of the method that you want to use when workflow is approved.
Denial	Select or enter the name of the method that you want to use when workflow is *not* approved.

Selecting and Configuring Self-Service Processes

Self-service scheduling allows users to execute processes and generate reports using context-specific information. You can configure execution of self-service processes from any transaction page where self-service scheduling is enabled. Scheduling in this manner retrieves mapped application data from the transaction page and uses it for the process execution. Once the process is completed, the authorized user can access the report from the dynamically generated notification from any PeopleSoft page.

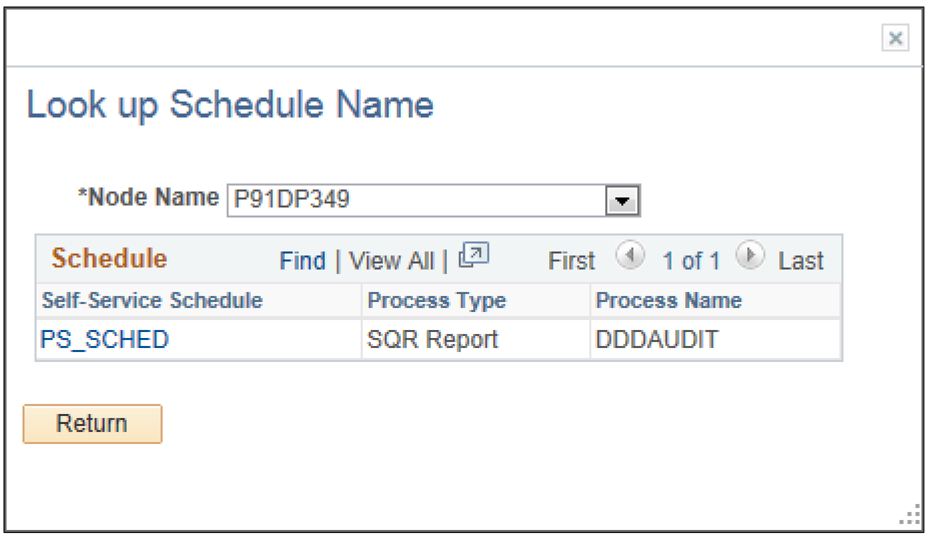
Within an activity guide, an action item can be defined as a *Process* type, which allows you to associate that action item with an already defined self-service schedule. Use the Process Configuration tab of the Action Item Details grid to complete configuring an action item as a process type:

1. Click the Assign Service link to Look up Schedule Name page.
2. If modifiable component interface parameters have been defined for the self-service schedule, click the Configure CI Parameters link to map activity guide data to the component interface.

Look up Schedule Name Page

Use the Look up Schedule Name page to select the already defined self-service schedule.

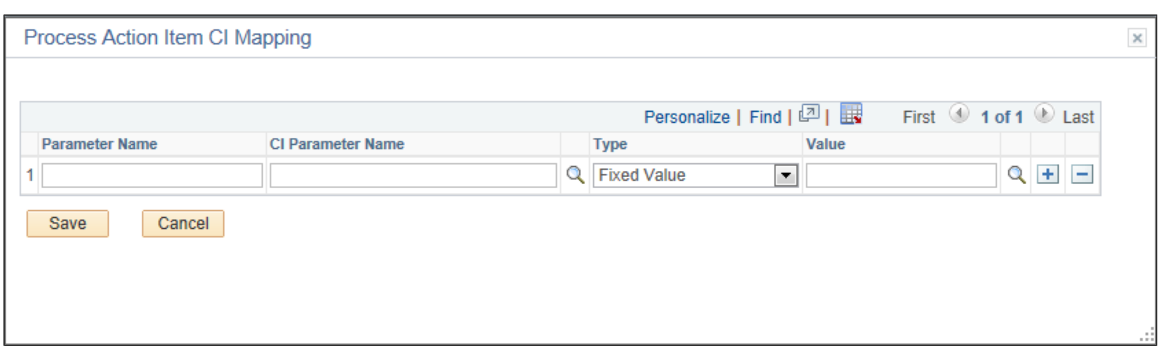
This example illustrates the fields and controls on the Look up Schedule Name page.



Process Action Item CI Mapping Page

Use the Process Action Item CI Mapping page to map activity guide context data to any modifiable parameters defined for the component interface for the self-service schedule.

This example illustrates the fields and controls on the Process Action Item CI Mapping page. You can find definitions for the fields and controls later on this page.



<table>
<tr><th>Field or Control</th><th>Description</th></tr>
<tr><td>Parameter Name</td><td>Enter the field name (as specified on the Template Properties: Advanced Options page) for a context data item.</td></tr>
<tr><td>CI Parameter Name</td><td>Select the modifiable parameter defined for the component interface.</td></tr>
<tr><td>Type</td><td>Select one of the following context types:
Fixed Value: Select to enter a fixed value as the context data.
Runtime: Select to have the context data value passed on the activity guide URL.

Note: The following special characters are not recommended in context parameter values as they have special meaning in URLs: # & ? + = : % /
Additional restrictions may apply if other special characters are not acceptable in the components configured as action items—for example, the space character is not acceptable in key values for certain components. You may see errors in the target component when such restricted characters are used in context parameter values.
System Variable: Select to specify a system variable as the context data value. The value of the system variable is determined at runtime.</td></tr>
<tr><td>Value</td><td>Enter the value when the context type is fixed value. Alternatively, select the system variable from the look up list when the context type is system variable.

Note: No value is entered when the context type is runtime.</td></tr>
</table>

Related Links

“Procedure for Developing and Deploying Self-Service Schedule” (Process Scheduler)

Deploying Activity Guides

Different activity guide types are deployed in different manners:

- Optimized (fluid optimized components only) – You must generate a tile definition or menu link to deploy an optimized activity guide.

- Non-optimized (any combination of fluid optimized, fluid non-optimized, and classic components) – You must generate a tile definition or menu link to deploy this type of non-optimized activity guide.

  Important! While the Setup Activity Guide WorkCenter will present the option to generate a pagelet and configure a WorkCenter for this type of an activity guide, any fluid components that are part of this activity guide cannot be accessed and rendered properly through a WorkCenter and pagelet.

- Non-optimized (classic components only) – You must generate a pagelet definition and configure a WorkCenter to deploy a classic-only, non-optimized activity guide.

Deploying Optimized and Non-Optimized Fluid Activity Guides

To deploy optimized and non-optimized fluid activity guides, you must manually create an access mechanism that invokes the fluid activity guide start page component (PT_AGSTARTPAGE_NUI). This access mechanism must:

- Pass the activity guide template ID as a query string parameter.
- Pass any required context parameters defined as runtime parameters as query string parameters.

  Note: Any required context parameters defined as system variables are supplied automatically and do not have to be passed as query string parameters.

Methodologies for creating access mechanisms include:

- Manually creating a content reference definition for a tile in the tile repository.
- Manually creating a content reference definition for a menu item in the portal menu structure.
- Creating a “jump” (or launch) page to collect required context data from the user. Then, pass the data as query string parameters when invoking the fluid activity guide start page component via a link or button.

Regardless of the methodology you choose, the following special characters are not recommended in context parameter values as they have special meaning in URLs: # & ? + = : % /

Additional restrictions may apply if other special characters are not acceptable in the components configured as action items—for example, the space character is not acceptable in key values for certain components. You may see errors in the target component when such restricted characters are used in context parameter values.

If you create a launch page, use the EncodeUrlForQuerystring built-in PeopleCode function to encode the context parameter values. For example:

```
&ctxtParams = &ctxtParams | "&CONTEXTIDPARAMS=" | &ctxtObj.fieldname | ":" | Encode⇒

URLForQueryString(&ctxtObj.keyValue);
```

Manually Creating a Content Reference Definition

When creating a content reference definition to invoke the fluid activity guide start page component:

1. Select the following options on the General page:

- **No Template**
- **Fluid Mode**
- **Display on Small Form Factor**

2. Set the node name to *LOCAL_NODE*.
3. Select the *PeopleSoft Generic URL* URL type.
4. Enter the following as the beginning of the Portal URL field:

   ```
   c/NUI_FRAMEWORK.PT_AGSTARTPAGE_NUI.GBL
   ```

5. Append the template ID as a query string parameter as follows:

   ```
   ?CONTEXTIDPARAMS=TEMPLATE_ID:template_ID
   ```

6. Append any additional required runtime parameters as query string parameters similar to the following:

   ```
   &CONTEXTIDPARAMS=PARAM_NAME:Value
   ```

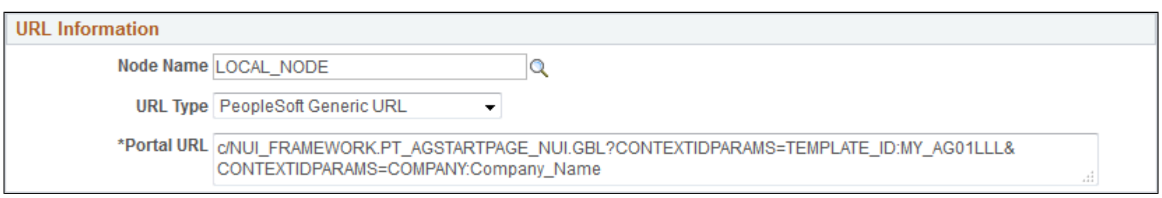
7. If the activity guide should not be accessible via the Back button—for example, after the user has completed the activity guide instance and exited—then append the following URL parameter:

   ```
   &AgInHistory=N
   ```

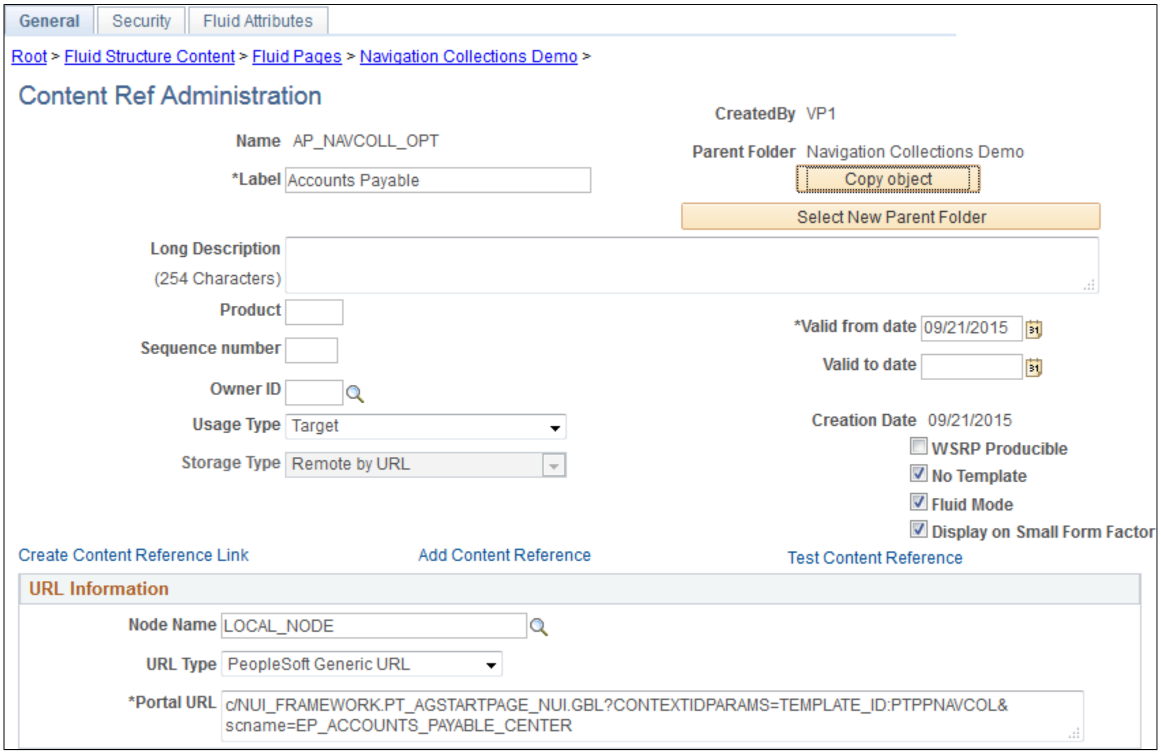
   Note: This URL parameter is also valid for fluid navigation collections.

In the following example, the Portal URL field includes the template ID (MY_AG01LLL) and a required runtime parameter (COMPANY) with its runtime value (Company_Name). The complete Portal URL field is defined as:

```
c/NUI_FRAMEWORK.PT_AGSTARTPAGE_NUI.GBL?
CONTEXTIDPARAMS=TEMPLATE_ID:MY_AG01LLL&CONTEXTIDPARAMS=COMPANY:Company_Name
```



The following example displays a tile definition to display a navigation collection and links as an optimized non-guided non-sequential activity guide.



Note: While the Template Properties: Properties page includes a group box to publish the template as a tile, this tile definition will include the template ID only, and no context data or instance IDs. Therefore, users who use this tile will be redirected to the My Activity Guide Instances page, on which they can select from a list of *existing* activity guides instances generated from the template for which they have contributor privileges. Therefore, this group box cannot be used as a general deployment mechanism or to generate new activity guide instances. See Defining Activity Guide Template Properties for more information.

Related Links

Creating and Maintaining Tile Definitions Manually

"Working with Two-Panel Implementations" (Fluid User Interface Developer's Guide)

Deploying Classic-Only Activity Guides

Classic activity guides are deployed on a WorkCenter page, which is where they can be accessed by users. Prior to creating a WorkCenter page where the activity guide will be deployed, you must first generate a pagelet for the pagelet area of the WorkCenter page. This pagelet will present the action items to the user in list form.

Depending on whether you want to tie the activity guide pagelet to a specific activity guide instance or to dynamically create the activity guide instance, there are three approaches for generating the activity guide pagelet:

- Provide the template ID, but not the instance ID, which allows the instance to be obtained or generated at runtime. At runtime, the required context data is passed as *query string parameters* in the

WorkCenter URL as name-value pairs. If an activity guide instance exists for this context data, then it is retrieved. Otherwise, a new activity guide instance will be created dynamically.

You can use the Publish as Pagelet task in the Setup Activity Guide WorkCenter to generate an activity guide pagelet of this type, which is typically all that is required in most circumstances. Alternatively, you can use Pagelet Wizard to generate this type of activity guide pagelet.

- Provide both the template ID and the activity guide instance ID, which creates a pagelet tied to an existing activity guide instance.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. The template ID is specified as a data source identifier in step 2 of Pagelet Wizard; the instance ID is specified as a data source parameter in step 3.

- Do not provide either the template ID or the activity guide instance ID, which generates an empty, generic activity guide pagelet that can be used to generate any instance from any activity guide template at run time.

  You must use Pagelet Wizard to generate an activity guide pagelet of this type. At run time, the template ID is passed as a data source parameter to the pagelet and the context data is provided as query string parameters in the WorkCenter URL as name-value pairs. This will create the activity guide instance dynamically.

Using the Setup Activity Guide WorkCenter to Generate an Activity Guide Pagelet

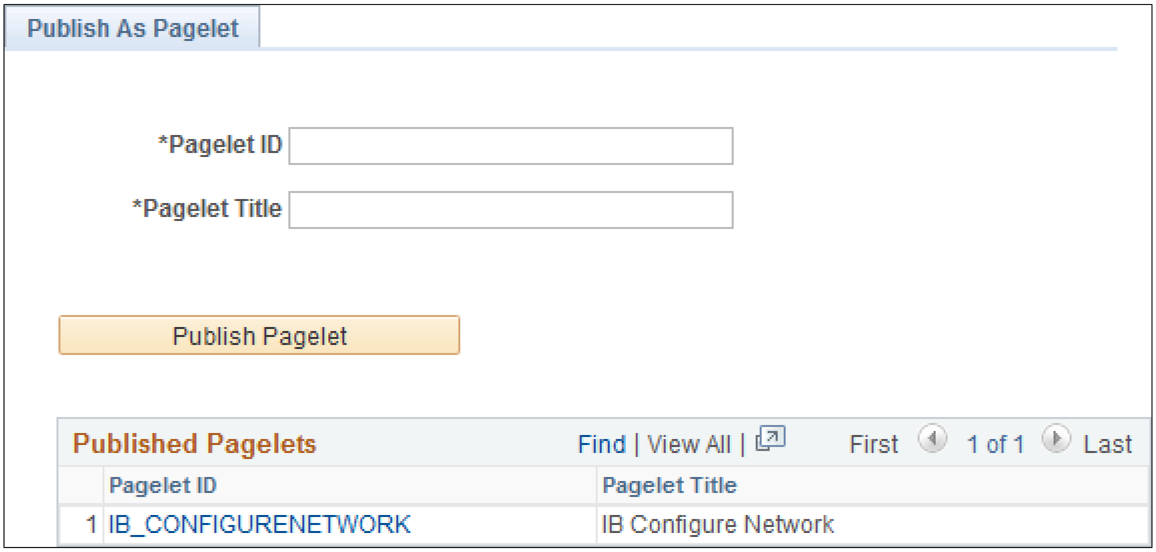
Use the Publish as Pagelet page to publish an activity guide pagelet with a template ID, but without an instance ID.

Important! The Publish as Pagelet page is available for non-optimized activity guides only. However, while the Setup Activity Guide WorkCenter will present the option to generate a pagelet and configure a WorkCenter for all non-optimized activity guide templates, it should not be used for activity guide templates that include fluid components.

Navigation:

1. **PeopleTools** > **Activity Guides** > **Activity Guide Templates**
2. Open a non-optimized template in the Setup Activity Guide WorkCenter:
   - Click the title link for the template.
   - Click the Properties link for the template.
3. Click the Publish as Pagelet task.

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



Field or Control	***Description***
Pagelet ID	Enter an ID for the pagelet definition.
Pagelet Title	Enter a title for the pagelet, which will appear in the pagelet's title bar.
Publish Pagelet	Click the Publish Pagelet button to create the pagelet definition.
Pagelet ID (link)	Click the Pagelet ID link to open the pagelet definition in Pagelet Wizard in a new browser window (or tab).

Using Pagelet Wizard to Generate an Activity Guide Pagelet

You must use Pagelet Wizard to generate an activity guide pagelet for these two scenarios:

- When you need to provide both the template ID and the activity guide instance ID, which creates a pagelet tied to an existing activity guide instance.
- When you do not want to provide either the template ID or the activity guide instance ID, which generates an empty, generic activity guide pagelet that can be used to generate any instance from any activity guide template at runtime.

When specifying a display format in step 4 of Pagelet Wizard, select the Passthru format. When specifying publication options in step 6 of Pagelet Wizard, always publish an activity guide pagelet as a *template pagelet.* Activity guide pagelets should not be made available or used as homepage pagelets. In addition, activity guide pagelets cannot be consumed as WSRP content.

Related Links

Step 2: Selecting a Pagelet Data Source

Step 3: Specifying Pagelet Data Source Parameters

Step 4: Selecting a Pagelet Display Format

Step 6: Specifying Pagelet Publishing Options

Configuring a WorkCenter Page for a Classic-Only Activity Guide

Use the Configure WorkCenter task in the Setup Activity Guide WorkCenter to configure a WorkCenter page definition for a classic-only activity guide. You must use the Activity Guide Layout (PTAILAYOUT) WorkCenter template, which displays the pagelet area is on the left, and the target content area is on the right. In addition, the template displays an activity guide navigation frame above the target content area providing Next and Previous navigation buttons by default. You must also add the activity guide pagelet to the WorkCenter definition.

Important! The Configure WorkCenter task is available for non-optimized activity guides only. However, while the Setup Activity Guide WorkCenter will present the option to generate a pagelet and configure a WorkCenter for all non-optimized activity guide templates, it should not be used for activity guide templates that include fluid components.

To configure a WorkCenter page for a classic-only activity guide, follow the standard WorkCenter development process:

1. Create a starting page for the WorkCenter that will display the activity guide pagelet.
2. If necessary, register any new components (and pages) in the portal.
3. Create the WorkCenter and designate its starting page. You can use any of the three options as the starting page:
   - Existing menu item.
   - Link of existing menu item
   - WorkCenter dashboard page.
4. Specify the Activity Guide Layout template as the layout template for this WorkCenter page.

   Important! If the Activity Guide Layout template is not used, then the activity guide navigation frame will not be displayed above the target content area and action item statuses will not be updated in the activity guide pagelet.

   ---

5. Add the activity guide pagelet to the WorkCenter pagelet area.

Important! Do not configure a remote activity guide pagelet for use in a WorkCenter defined on the local node. Instead, define both the WorkCenter and activity guide pagelet on the remote node. Then, using unified navigation or another mechanism, access the content reference for the remote WorkCenter from the local node.

After you have configured the WorkCenter for the activity guide, you can update the template's Target URL property. See Defining Activity Guide Template Properties for more information.

Related Links

Understanding the WorkCenter Development and Implementation Process

Selecting the WorkCenter Starting Page

Configuring WorkCenter Pages

Modifying Access to the WorkCenter

You must ensure that users can only access the WorkCenter in the proper manner:

- If the activity guide pagelet was defined with a specific instance ID, then standard menu navigation to the WorkCenter can be allowed.
- If the activity guide requires unique context data to identify each instance, you must ensure that context data is defined prior to accessing the WorkCenter:
  - Use a landing/page questionnaire approach to pass query string parameters to the WorkCenter.
  - Use a query pagelet to pass query string parameters to the WorkCenter.
  - In both cases, hide the WorkCenter's content reference from menu navigation.

If the activity guide requires context data to retrieve or generate the correct activity guide instance, then you must ensure that the WorkCenter is accessed in a manner that provides the context data as query string parameters on the WorkCenter URL. (Additional parameters at the end of a URL are typically referred to as *query string parameters* and sometimes simply as *URL parameters.*) Some PeopleSoft applications achieve this by creating a landing page with a questionnaire that sets the contextual data prior to launching the WorkCenter page. In other cases, you might use a query and query pagelet to set the context data on the WorkCenter URL. In either case, you must also ensure that users cannot access the WorkCenter directly, because doing so would not set required context data on the WorkCenter URL. In some cases, this means hiding the WorkCenter content reference from direct menu navigation.

Regardless of the methodology you choose, the following special characters are not recommended in context parameter values as they have special meaning in URLs: # & ? + = : % /

Additional restrictions may apply if other special characters are not acceptable in the components configured as action items—for example, the space character is not acceptable in key values for certain components. You may see errors in the target component when such restricted characters are used in context parameter values.

If you create a launch page, use the EncodeUrlForQuerystring built-in PeopleCode function to encode the context parameter values. For example:

```
&ctxtParams = &ctxtParams | "&CONTEXTIDPARAMS=" | &ctxtObj.fieldname | ":" | Encode⇒

URLForQueryString(&ctxtObj.keyValue);
```

Managing Activity Guide Instances

This section discusses how to create, manage, and access activity guide instances.

Understanding Creation and Management of Activity Guide Instances

Activity guide instances can be created dynamically or manually in the following manners:

- Dynamically by accessing a tile or menu link that has been configured to use the fluid start page component.

  See Deploying Optimized and Non-Optimized Fluid Activity Guides for more information.

- Dynamically by accessing a WorkCenter displaying an activity guide pagelet.

  See Deploying Classic-Only Activity Guides for more information.

- Manually by clicking the Create Instance button on the Template Properties: Properties page for an activity guide template.

  Typically, activity guide instances are created manually for testing purposes or when only one instance of the activity guide is required—for example, for a guided administrative process. See Creating New Activity Guide Instances Manually for more information.

The fields and properties of activity guide instances match those of activity guide templates. Therefore, this data can be viewed or maintained in much the same way as activity guide templates. In fact, the underlying pages to maintain this data are exactly the same in most cases. However, because activity guide instances can and often will be created dynamically, there is not the same need to maintain activity guide instance data manually.

For users to be able to generate and access their own activity guide instances, they must have the following permissions:

- The PTAI_ACCESS role.
- At a minimum, viewer privileges on the activity guide template used to generate the instance.
- At a minimum, contributor privileges on the activity guide instance itself.

Important! Once an activity guide instance has been created and saved, which template is associated with the instance cannot be changed. Moreover, changes to the template are not propagated to any active instances.

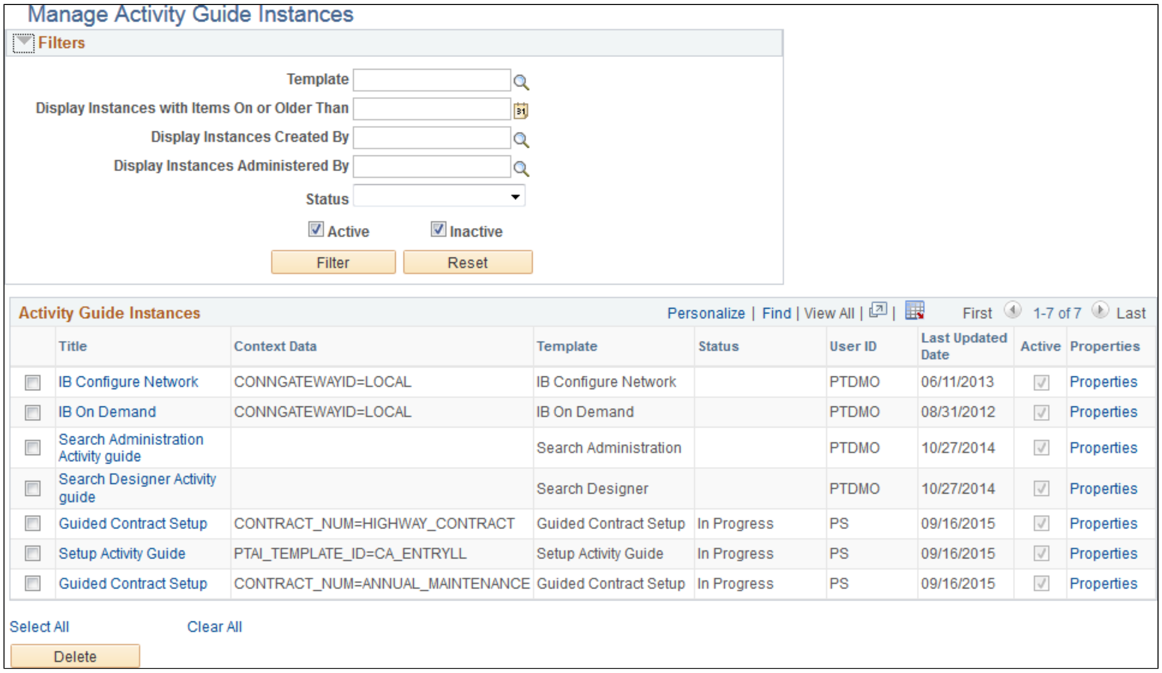
Administering Activity Guide Instances

Use the Manage Activity Guide Instances page to manage activity guide instances.

Navigation:

PeopleTools > **Activity Guides** > **Activity Guide Instances**

This example illustrates the fields and controls on the Manage Activity Guide Instances page. You can find definitions for the fields and controls later on this page.



Filters

Field or Control	*Description*
Template	Select an activity guide template as a filter criteria.
Display Instances with Items On or Older Than	Select a date to filter by.
Display Instances Created By	Select a user ID as a filter criteria.
Display Instances Administered By	Select a user ID as filter criteria.
Status	Select a status as a filter criteria.
Active	Select whether to include active instances.
Inactive	Select whether to include inactive instances.
Filter	Click to filter the list by the selected criteria.
Reset	Click to clear all filter criteria.

Activity Guide Instances

Field or Control	***Description***
Title	Click a title link to manage the action items for that instance.
Properties	Click the **Properties** link to manage the properties for that instance.
Select All	Click to select all instances.
Clear All	Click to clear the selection of any instances.
Delete	Click to delete the selected instances.

Creating New Activity Guide Instances Manually

Use the Create Activity Guide Instance page to specify required context data in order to generate an instance manually.

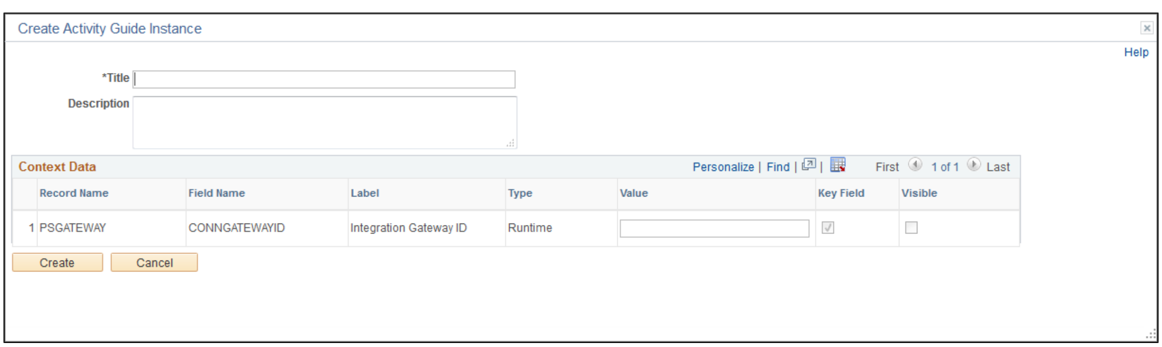
Navigation:

1. **PeopleTools** > **Activity Guides** > **Activity Guide Templates**
2. Click the Properties link to modify an existing template.

   Note: Alternatively, in the Setup Activity Guide WorkCenter, click the Template Properties task.

3. Click the Create Instance button.

This example illustrates the fields and controls on the Create Activity Guide Instance page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Title	Specify the title for this activity guide instance.

Field or Control	*Description*
Description	Specify an optional description.
Value	If contextual data has been defined for the activity guide template, then you must specify a value for each contextual field.
Create	Click to create the instance.

When you create an activity guide instance in this manner, action items along with template characteristics such as participants, contextual data fields, and pagelet and navigation options are copied from the template to the instance.

Accessing Instances from the My Activity Guide Instances Page

Use the My Activity Guide Instances page to access fluid activity guide instances that have been generated from a specific template and to which you have access (that is, the contributor role).

Navigation:

Click an activity guide tile that has been configured to open the My Activity Guide Instances page.

This example illustrates the fields and controls on the My Activity Guide Instances page. You can find definitions for the fields and controls later on this page.

Click a title link to open the activity guide instance.

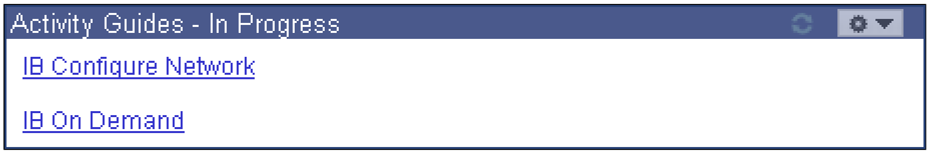
Accessing Instances from the Activity Guides - In Progress Pagelet

Use the Activity Guides - In Progress pagelet to view the list of *all* activity guide instances for which you have contributor privileges.

Navigation:

1. On a classic homepage or dashboard, click the Content link.
2. Select the *Activity Guides - In Progress* option to display the pagelet on your homepage or dashboard.

This example illustrates the fields and controls on the Activity Guides - In Progress pagelet. You can find definitions for the fields and controls later on this page.



The Activity Guides - In Progress homepage pagelet can be used to access active activity guide instances to which the user is a contributor.

Field or Control	***Description***
link	Click a link to go to the WorkCenter page for that activity guide instance. **Note:** While a hyperlink is always displayed, the target URL property must be defined on the instance's Properties page in order for the link to work. (Typically, the URL property is defined and inherited from the activity guide template.)

Related Links

Defining Activity Guide Template Properties

Reviewing Usage Statistics

The Activity Guide framework enables you to compute the elapsed duration of a step or instance in the guided process.

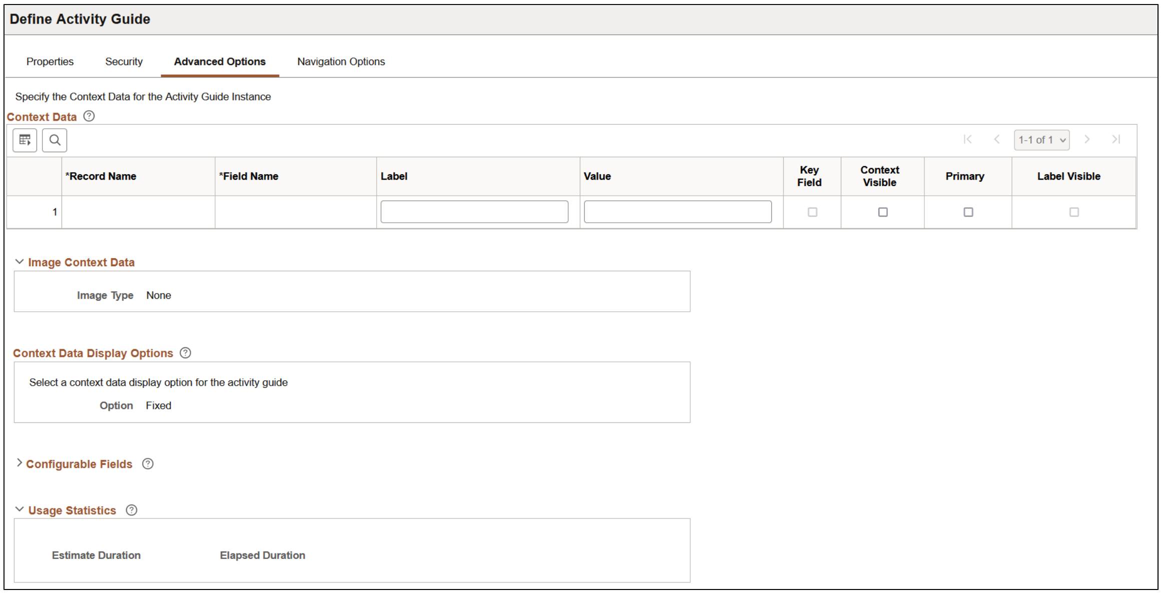
The Advanced Options page and the Usage Statistics tab provide the required information, which are described in this topic.

Advanced Options Page

Use the Advanced Options tab to view the Estimate Duration and Elapsed Duration values, which are populated by the Activity Guide framework.

PeopleTools > **Activity Guides** > **Activity Guide Instances** and on the Manage Activity Guide Instances, select the Properties link of the activity guide.

This example illustrates the fields and controls on the Advanced Options tab. You can find definitions for the fields and controls later on this page.



This section attempts to describe the Usage Statistics group box. All other fields on the page are similar to the Template Properties: Advanced Options page.

Field or Control	***Description***
Estimate Duration	It denotes the approximate duration required to complete an instance. The value entered in the **Estimate Duration** field on the template is copied to this field.
Elapsed Duration	Elapsed Duration is the time (in days) elapsed between the start and completion of the instance. The value is computed by the framework when the end date value is set at completion of the instance.

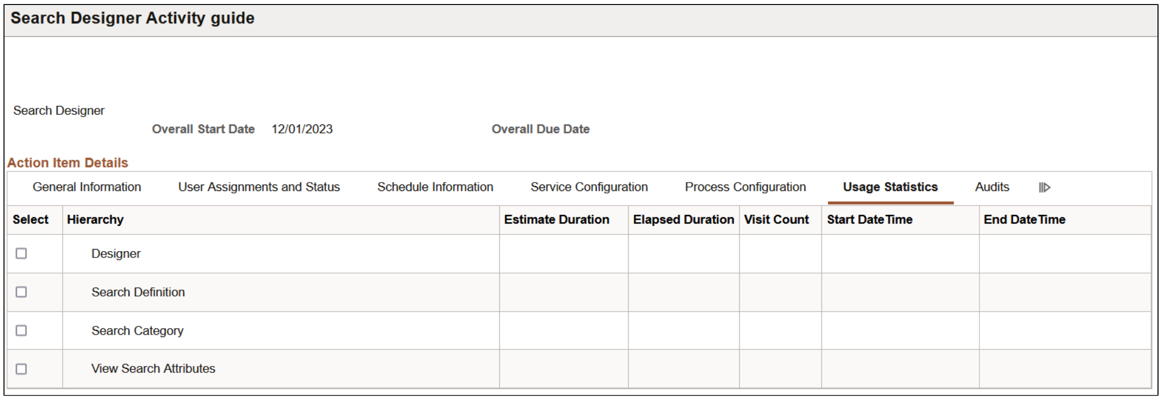
Usage Statistics Tab

The Usage Statistics tab displays information that is computed by the framework.

Navigation:

PeopleTools > **Activity Guides** > **Activity Guide Instances** and on the Manage Activity Guide Instances, select the activity guide.

This example illustrates the fields and controls on the Usage Statistics tab. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Estimate Duration	It denotes the approximate duration required to complete a step. The value is populated by the framework.
Elapsed Duration	Elapsed Duration is the time (in days) elapsed between the start and completion of the step. The value is computed by the framework when the end date value is set at completion of the step.
Visit Count	The activity guide framework computes this value based on the number of times a user visits a step.
Start Date Time	The activity guide framework sets the start date/time for the step when a user first lands on the step.
End Date Time	It denotes the date/time on which the step was set to completed.

The Estimate Duration, Elapsed Duration, Visit Count, Start Date Time, and End Date Time fields can be selected for indexing when you define an Activity Guide type search definition.

For more information, see Defining Advanced Options and Defining Action Item Details.

Enhancing Activity Guides

This section discusses how to enhance the usage and implementation of activity guides including:

- Using PeopleCode to enhance activity guides.
- Creating a custom activity guide data type.

- Defining activity guide system options.

Understanding Where PeopleCode Can Be Used

The activity guide framework provides multiple methodologies for using PeopleCode to implement and customize activity guide usage in your system including:

- Initializing activity guide instances.

  Whether an activity guide is created dynamically or manually, each instance created from a template can be initialized through a PeopleCode method that you define specifically for this purpose.

- Performing pre-processing for action items.

  You can define a PeopleCode application class and method to perform pre-processing on an action item before the target transaction is displayed to the user. In addition to performing this pre-processing, the method *must* also redirect to the target transaction. In addition, Oracle recommends that you develop this application class and method by implementing and extending the PT_RCF:ServiceAGInterface interface class.

  Alternatively, pre-processing can be performed in an iScript that must also specify and transfer the user to the target transaction.

- Performing post-processing for action items.

  You can define another PeopleCode method to perform post-processing on an action item after the user has completed the task. Post-processing can occur on the local system or the remote system if that's where the transaction is hosted.

- Developing custom action buttons.

  The navigation frame provides default navigation buttons to allow a user to move to the next or previous action item. You can create additional navigation buttons—for example, to allow a user to manually mark an action item as complete or to cancel the activity guide instance altogether.

- Dynamically returning an image URL as context data.

  For fluid activity guides only, you can define a method to use defined context data to dynamically return an image URL as additional context data.

- Dynamically returning a content URL to be displayed in a modal window.

  For fluid activity guides only, you can define a method to dynamically return a content URL to display additional information. Typically, this additional content is static text; however, it can include context data values.

- Performing pre-processing for the activity guide pagelet.

  For classic activity guides only, you can define a PeopleCode method to perform pre-processing that will be executed every time the activity guide pagelet is loaded or reloaded in the WorkCenter pagelet area.

Initializing an Activity Guide Instance

Whether an activity guide is created dynamically or manually, each instance created from a template can be initialized through a PeopleCode method that you define specifically for this purpose. This method is identified on the Advanced Options page for the activity guide template.

Limit initialization to operations such as inactivating steps based on the context, initializing the step statuses, updating step assignments, and so on. Also, perform any updates, additions, or deletions that impact translations. Oracle recommends to avoid complex operations such as accessing or updating application tables.

Note: To create your own application class to initialize an activity guide instance, you can clone the PTAI_UTILITIES:InstanceIDCreation class and then modify your copy to suit the needs of your specific activity guide.

In the following example, the PTAI_UTILITIES:InstanceIDCreation class implements and extends the delivered PTAI_ACTION_ITEMS:AGInterface class. The constructor method implements the delivered abstract method by copying the action item assignments from the template *and* adding the user ID of the current user who instantiates the activity guide.

```
import PTAI_ACTION_ITEMS:*;
import PTAI_ACTION_ITEMS:AGInterface;
import PTAI_COLLECTION:*;
import PTAI_ACTION_ITEMS:Member:*;


class InstanceIDCreation implements PTAI_ACTION_ITEMS:AGInterface
   method InstanceCreation(&list As PTAI_ACTION_ITEMS:List);
end-class;


/**
* Sets the Instance items status to Assigned and Assigned to Operator ID to %User i⇒

d.
**/
method InstanceCreation
   /+ &list as PTAI_ACTION_ITEMS:List +/
   /+ Extends/implements PTAI_ACTION_ITEMS:AGInterface.InstanceCreation +/

   Local array of string &childItemIds;
   Local integer &i;
   Local boolean &return;
   Local string &ptai_mbr_name, &ptai_mbr_type, &ptai_privset_ID, &ListID, &templat⇒

eID, &templatelabel, &templatelabel_lng;

   Local PTAI_ACTION_ITEMS:ActionItem &item;
   Local PTAI_COLLECTION:Collection &members, &buttons, &contextfields;
   Local PTAI_ACTION_ITEMS:Member &member;
   Local PTAI_ACTION_ITEMS:Privilege &privileges;
   Local PTAI_ACTION_ITEMS:PgltButtons &button;

   &list.Status = "IP";
   SQLExec("Select PTAI_PARENT_TMPL From PS_PTAI_LIST Where PTAI_LIST_ID = :1 ", &l⇒

ist.ListId, &templateID);

   rem get the template label;
   SQLExec("Select PTAI_LABEL From PS_PTAI_LIST Where PTAI_LIST_ID = :1 ", &templat⇒

eID, &templatelabel);
   If %Language <> %Language_Base Then
```



```
      SQLExec("Select PTAI_LABEL From PS_PTAI_LIST_LNG Where PTAI_LIST_ID =:1 AND L⇒

ANGUAGE_CD=:2", &templateID, %Language, &templatelabel_lng);
      If All(&templatelabel_lng) Then
         &templatelabel = &templatelabel_lng;
      End-If;
   End-If;

   &list.Label = &templatelabel;
   &return = &list.save();

   <*
   rem the button label and method;
   &buttons = create PTAI_COLLECTION:Collection();

   &button = create PTAI_ACTION_ITEMS:PgltButtons("Cancel", "CancelAG");
   &buttons.InsertItem(&button);

   &button = create PTAI_ACTION_ITEMS:PgltButtons("Continue Later", "ExitAG");
   &buttons.InsertItem(&button);

   rem &button = create PTAI_ACTION_ITEMS:PgltButtons("Mark Task Completed", "MarkC⇒

omplete");
   rem &buttons.InsertItem(&button);

   &list.SavePgltBtn(&buttons);
   *>

   Rem change the status of the action items;
   &childItemIds = &list.getActionItems();

   For &i = 1 To &childItemIds.Len
      &item = create PTAI_ACTION_ITEMS:ActionItem();
      &item.open(&childItemIds [&i]);
      &item.Status = "1";
      /*
         &item.PackageRoot = "W3EB_LIFE_EVENT";
         &item.QualifyPath = ":";
         &item.AppClassID = "ItemStatus";
 */

      Local PTAI_COLLECTION:Collection &oAssignmentColl = create PTAI_COLLECTION:Co⇒

llection();
      Local PTAI_ACTION_ITEMS:ActionItemAssignments &oAssignment = create PTAI_ACTI⇒

ON_ITEMS:ActionItemAssignments();
      &oAssignment.AssignType = "USER";
      &oAssignment.AssignValue = %OperatorId;
      /* if you want to copy the assignments from template's action items to instan⇒

ce action items, call the getAssignments method and add your new user/role
   assignments in the returned collection and then call saveAssignments with the co⇒

mbined collection object as shown below. Or if you just want to copy
   the template assigments to instance and not adding any new assignments, you don'⇒

t need the following 3 lines of code */
      &oAssignmentColl = &item.getAssignments();
      &oAssignmentColl.InsertItem(&oAssignment As PTAI_COLLECTION:Collectable);
      &item.saveAssignments(&oAssignmentColl);

      &return = &item.save();

   End-For;

   Rem Save the logged in UserID for security and privileges;

   &members = create PTAI_COLLECTION:Collection();
```

```
   &members = &list.getMembers();
   &ptai_mbr_name = %UserId;
   &ptai_mbr_type = "USER";

   SQLExec("Select PTAI_PRIVSET_ID From PS_PTAI_PRIVSET Where DESCR = 'Contributor'⇒

 and PTAI_FEATURE = 'AI' AND ACTIVE_FLAG = 'A' AND PTAI_SYSTEM_ITEM = 'Y' ", &ptai_⇒

privset_ID);


   &member = create PTAI_ACTION_ITEMS:Member(&ptai_mbr_name, &ptai_mbr_type);
   &member.PrivilegeSetID = &ptai_privset_ID;
   &members.InsertItem(&member);
   &list.saveMembers(&members);

end-method;
```

Related Links

Defining Advanced Options

Performing Pre-Processing for Action Items

You can define a PeopleCode application class and method to perform pre-processing on an action item before the target transaction is displayed to the user. In addition to performing this pre-processing, the application class and method *must* redirect to the target transaction. Alternatively, you can implement this functionality in an iScript instead of in an application class.

Important! During pre-processing, do not attempt component buffer access, use think-time functions such as MessageBox, WinMessage, or others, or use system variables.

To implement pre-processing for an action item in an application class and method:

1. Define an application class that extends the PT_RCF:ServiceAGInterface.

2. Define the execute method, which implements and extends the execute method in the base class.

3. Set the %This.StrRedirectUrl property to redirect to the target transaction.

4. Append the %This.StrAGQueryParameter property to the end of the generated component URL.

5. Create a related content service definition for this application class.

6. In the Action Item Details grid, select *App Class URL* as the link type and select the related content service definition.

The following example shows the PTAI_AGPKG_UNINAV:TestPreProcessing, which implements and extends the PT_RCF:ServiceAGInterface base class:

```
import PT_RCF:ServiceAGInterface;
import PTAI_ACTION_ITEMS:*;

class TestPreProcessing extends PT_RCF:ServiceAGInterface
   method execute();
end-class;

method execute
   /+ Extends/implements PT_RCF:ServiceInterface.execute +/
```



```
   Local PTAI_ACTION_ITEMS:ActionItem &ObjAItem;
   Local boolean &boolSaved;
   Local string &market;

   &ObjAItem = create PTAI_ACTION_ITEMS:ActionItem();
   &ObjAItem.open(%This.StrActionItemId);
   If &ObjAItem.Status <> "4" Then
      &ObjAItem.Status = "2";
   End-If;
   &boolSaved = &ObjAItem.save();

   &market = %Market;
   If None(&market) Then
      &market = "GBL";
   End-If;

   %This.StrRedirectUrl = GenerateComponentContentURL(%Portal, %Node, MenuName.PTUN⇒

_MENU, &market, Component.PTUN_REMOTENODECFG, Page.PTUN_REMOTENODECFG, "") | %This.⇒

StrAGQueryParameter;
end-method;
```

Performing Post-Processing for Action Items

You can define a PeopleCode application class and method to perform post-processing on an action item after the user has completed the transaction—for example, to update the status of the action item to complete. For post-processing, the method name must be called ItemPostProcess. Post-processing can occur on the local system or the remote system if that's where the transaction is hosted.

Important! Post-processing can occur on the local system or the remote system if that's where the transaction is hosted. Post-processing is executed through component save processing only. If you have implemented a custom save button on the page to save the component, the button's page field properties must be configured to invoke the toolbar save action or the PeopleCode executed by these buttons must include either the DoSave or DoSaveNow built-in functions to trigger the component save events; otherwise, the post-processing specified here will not be executed.

Example 1

In the first example, the class has been developed by implementing and extending the PTAI_ACTION_ITEMS:AGInterface interface class. The ItemPostProcess method defines the post-processing that will occur for this action item.

```
import PTAI_ACTION_ITEMS:AGInterface;
import PTAI_ACTION_ITEMS:ActionItem;
import PTAI_ACTION_ITEMS:Constants;

class IB_GATEWAY implements PTAI_ACTION_ITEMS:AGInterface
   method IB_GATEWAY();
   rem method InstanceCreation(&list As PTAI_ACTION_ITEMS:List);
   rem method PageletPreProcess(&list As PTAI_ACTION_ITEMS:List);
   rem method ItemPreProcess(&item As PTAI_ACTION_ITEMS:ActionItem);
   method ItemPostProcess(&list_id As string, &item_id As string, &Nodename As stri⇒

ng);
end-class;

/* constructor */
method IB_GATEWAY
end-method;
```

```
method ItemPostProcess
   /+ &list_id as String, +/
   /+ &item_id as String, +/
   /+ &Nodename as String +/
   /+ Extends/implements PTAI_ACTION_ITEMS:AGInterface.ItemPostProcess +/
   Local PTAI_ACTION_ITEMS:Constants &PTAI_CONSTANTS = create PTAI_ACTION_ITEMS:Con⇒

stants();
   Local PTAI_ACTION_ITEMS:ActionItem &item = create PTAI_ACTION_ITEMS:ActionItem()⇒

;
   Local boolean &ret;
   Local string &connurl;
   Local string &defaulturl = "http://machinename:port/PSIGW/PeopleSoftListeningCon⇒

nector";
   Local number &count;

   /*** Debug Test ***/
   rem MessageBox(0, "", 0, 0, "I'm being triggered");

   &item.open(&item_id);
   /*** Add logic to check for step completion ***/
   SQLExec("select connurl from psgateway where local_flag='Y'", &connurl);
   SQLExec("select count(*) from psconn", &count);

   If All(&connurl) And
         (&connurl <> &defaulturl) And
         &count > 0 Then
      &item.Status = &PTAI_CONSTANTS.STATUS_COMPLETED;
   Else
      &item.Status = &PTAI_CONSTANTS.STATUS_IN_PROGRESS;
   End-If;

   /*** Save the Action Item State ***/
   &ret = &item.save();
end-method;
```

Example 2

In this second example, the ItemPostProcess method includes conditional logic that handles execution differently depending on whether the transaction is on the local node or a remote node. If the ItemPostProcess method is executing on the remote node for the remote transaction, then it uses the PTAI_UPDATEITEM service operation to update status on the local node.

```
import PTAI_ACTION_ITEMS:*;

Declare Function SecuritySyncMessage PeopleCode PTUN_NODECFGDVW.PTUN_BTN FieldFormu⇒

la;
Declare Function IsLocalNode PeopleCode PSPTCSSRVDEFN.PTCS_SERVICEURLTYP FieldFormu⇒

la;

method ItemPostProcess
   /+ &list_id as String, +/
   /+ &item_id as String, +/
   /+ &Nodename as String +/
   Local boolean &boolSaved;
   Local PTAI_ACTION_ITEMS:ActionItem &ObjAItem;
   Local PTAI_ACTION_ITEMS:Constants &AIConstants;
   Local string &Status, &Priority;
   Local string &Required;
   Local number &PercCompl, &Seqno;

   &AIConstants = create PTAI_ACTION_ITEMS:Constants();
   /* Test values */
   &Status = &AIConstants.STATUS_COMPLETED;
```



```
   &Priority = "1";
   &Required = "Y";
   &PercCompl = 100;
   rem &Seqno = 35;
   If Not IsLocalNode(&Nodename) Then
      Local Message &Req_Msg, &Response;
      Local XmlDoc &requestDoc;
      Local number &i1;
      Local XmlNode &currentParamsNode;
      Local array of XmlNode &SuccessNode, &errorNode;
      Local string &ResponseXml;
      Local SOAPDoc &TransformedDoc;

      &Req_Msg = CreateMessage(@("Operation." | "PTAI_UPDATEITEM"), %IntBroker_Requ⇒
est);
      &requestDoc = CreateXmlDoc(GetHTMLText(HTML.PTAI_UPDATEITEM_REQ, %UserId, Get⇒
HTMLText(HTML.PTAI_UPDATEITEM_REQ_PARAMS, &item_id, &list_id, &Nodename, &Priority,⇒
 &Status, &Required, &PercCompl, &Seqno)));
      &Req_Msg.SetXmlDoc(&requestDoc);

      &Response = SecuritySyncMessage(&Req_Msg, &Nodename, "PTAI_UPDATEITEM");
      If &Response <> Null Then
         &ResponseXml = &Response.GenXMLString();
         &TransformedDoc = CreateSOAPDoc(&ResponseXml);
         &SuccessNode = &TransformedDoc.DocumentElement.GetElementsByTagName("SAVES⇒
UCCESS");
         &errorNode = &TransformedDoc.DocumentElement.GetElementsByTagName("ERRORMS⇒
G");

         If &SuccessNode <> Null And
               &SuccessNode.Len = 1 Then
            rem do nothing;
         End-If;
         If &errorNode <> Null And
               &errorNode.Len = 1 Then
            Error "Error from Application hosting Activity Guide : " | &errorNode [⇒
1].NodeValue;
         End-If;
      Else
         Error "No Response from IB";
      End-If;
   Else
      &ObjAItem = create PTAI_ACTION_ITEMS:ActionItem();
      &ObjAItem.open(&item_id);
      &ObjAItem.Status = &Status;
      &ObjAItem.Priority = &Priority;
      If &Required = "Y" Then
         &ObjAItem.Required = True;
      Else
         &ObjAItem.Required = False;
      End-If;
      &ObjAItem.PercCompl = &PercCompl;
      If All(&Seqno) Then
         &ObjAItem.Sequence = &Seqno;
      End-If;
      &boolSaved = &ObjAItem.save();
   End-If;
end-method;
```


Developing Custom Action Buttons

The activity guide subbanner in fluid activity guides or the navigation frame of the classic WorkCenter provide default navigation buttons to allow a user to move to the next or previous action item. You can create additional navigation buttons to be displayed in these same locations—for example, to allow a user to manually mark an action item as complete or to cancel the activity guide instance altogether. custom action buttons are defined on the Pagelet Options page for a template or instance. Additionally, custom action buttons can be defined per action item on the Configure Related Data page—for example, to add a Submit button to the final step of a sequential activity guide.

Note: For classic activity guides, Oracle recommends that you add no more than two additional custom action buttons.

Important! In custom action buttons, do not attempt component buffer access, use think-time functions such as MessageBox, WinMessage, or others, or use system variables.

Reserved Method Names

Certain reserved method names are recognized for use with guided activity guides. For some reserved method names (for example, MarkComplete and SubmitAGProcess), you must implement custom PeopleCode to perform the designated operation. In other cases (for example, PTAI_NAV_SAVE and PTAI_NAV_AUTO_SAVE), specifying the reserved method name will invoke delivered functions subject to the implementation requirements.

Method Name	***Function***	***How to Implement***
ExitAGProcess	Use this method to render a custom Exit button to run PeopleCode (for example, to display a warning message) prior to exiting the activity guide.	Implement the ExitAGProcess method as custom PeopleCode. See PTAI_UTILITIES:Utilities for an example.
MarkComplete	Use this method to render a custom Mark Complete button to allow a user to manually mark an action item as complete.	Implement the MarkComplete method as custom PeopleCode. See PTAI_UTILITIES:Utilities for an example.
ReturnToQuestionnaire	Use this method to render a custom Return to Questionnaire button to delete the current activity guide instance and return the user to the questionnaire page using the default answers defined in the template properties.	Implement the ReturnToQuestionnaire method as custom PeopleCode. See PTAI_UTILITIES:Utilities for an example.

<table>
<tr><th>Method Name</th><th>Function</th><th>How to Implement</th></tr>
<tr><td>SubmitAGProcess</td><td>Use this method to render a custom Submit button that must be placed on the final step of sequential activity guides. When a user clicks this Submit button, two operations occur in this order:

1. Data added and modified by the user is submitted and then committed to the database.

2. Your custom SubmitAGProcess program is executed. This method must set the activity guide instance status to complete and then save the instance.</td><td>Implement the SubmitAGProcess method as custom PeopleCode. See PTAI_UTILITIES:Utilities for an example.</td></tr>
<tr><td>PTAI_NAV_AUTO_SAVE</td><td>Use this method to introduce auto-save functionality for the built-in Next and Previous buttons for non-sequential, guided fluid activity guides.

Note: Auto-save is enabled for sequential and single unit of work activity guides automatically.</td><td>No additional implementation is required.</td></tr>
<tr><td>PTAI_NAV_SAVE</td><td>For classic activity guides only, use this method to render a custom Save button to invoke the transaction page’s Save button to save the current action item.</td><td>1. At the template level, define PTAI_NAV_SAVE as a configurable field on the Template Properties: Advanced Options page.

2. For each action item that requires this save functionality, you must map the HTML field ID of the Save button on the page to the configurable field.

Use your browser’s built-in web developer tools to determine the HTML field ID. For pages that display the standard PeopleTools Save button, the HTML field ID is #ICSave.

Then, specify this field ID as the value of the PTAI_NAV_SAVE configurable field on the action item’s Configure Related Data page.</td></tr>
</table>

Method Name	*Function*	*How to Implement*
PTAI_NAV_SAVE_ NEXT	Use this method to combine the save action into the Next button, which invokes the transaction page's Save button to save the current action item and then display the next action item. **Note:** Use this method in classic activity guides only.	1. At the template level, define PTAI_NAV _SAVE_NEXT as a configurable field on the Template Properties: Advanced Options page. 2. For each action item that requires this save functionality, you must map the HTML field ID of the Save button on the page to the configurable field. Use your browser's built-in web developer tools to determine the HTML field ID. For pages that display the standard PeopleTools Save button, the HTML field ID is #ICSave. Then, specify this field ID as the value of the PTAI_NAV_SAVE_NEXT configurable field on the action item's Configure Related Data page.
PTAI_NAV_SAVE_ PREV	Use this method to combine the save action into the Previous button, which invokes the transaction page's Save button to save the current action item and then display the previous action item. **Note:** Use this method in classic activity guides only.	1. At the template level, define PTAI_NAV _SAVE_PREV as a configurable field on the Template Properties: Advanced Options page. 2. For each action item that requires this save functionality, you must map the HTML field ID of the Save button on the page to the configurable field. Use your browser's built-in web developer tools to determine the HTML field ID. For pages that display the standard PeopleTools Save button, the HTML field ID is #ICSave. Then, specify this field ID as the value of the PTAI_NAV_SAVE_PREV configurable field on the action item's Configure Related Data page.

Method Name	*Function*	*How to Implement*
PTAI_GRANT	Use this method to render a custom button that allows a user to grant contributor access to the current step to other users and roles who have contributor access to the instance. Once access has been granted in this manner, only an instance administrator can revoke the grant using the Manage Activity Guide Instances page. **Note:** Use this method in fluid activity guides only. **Note:** Do not configure the grant action for activity guides that do not support multiple users such as single unit of work and restartable activity guides. Also, the grant action does not make sense for self-service processes when all of the steps in the process are to be performed only by the one user.	Other than specifying the label for the button, no additional implementation is required. **Important!** In the current release, if an application package and class has not been specified for another custom button, you must specify a placeholder application package and class in order to be able to save the configuration. However, you do not need to implement the PTAI_GRANT method in this class. Therefore, you can use PTAI_UTILITIES:Utilities as a placeholder.
PTAI_NOTIFY	Use this method to render a custom button that allows a user with contributor access to the instance to send ad hoc notifications to other users by user ID, role, or email address. **Note:** Use this method in fluid activity guides only.	Other than specifying the label for the button, no additional implementation is required. **Important!** In the current release, if an application package and class has not been specified for another custom button, you must specify a placeholder application package and class in order to be able to save the configuration. However, you do not need to implement the PTAI_NOTIFY method in this class. Therefore, you can use PTAI_UTILITIES:Utilities as a placeholder.

Method Name	*Function*	*How to Implement*
PTAI_OVERVIEW	Use this method to render a custom button that allows a user to view an overview of the activity guide steps along with their status. **Note:** Use this method in fluid activity guides only.	Other than specifying the label for the button, no additional implementation is required. **Note:** In the current release, if an application package and class has not been specified for another custom button, you must specify a placeholder application package and class in order to be able to save the configuration. However, you do not need to implement the PTAI_OVERVIEW method in this class. Because PTAI_UTILITIES:Utilities has a sample implementation, *do not* use it as a placeholder. Moreover, steps can be hidden from display in the process overview depending on business logic. See PTAI_UTILITIES:Utilities for an example how to do this.

Custom Exit Button

In the following example for a classic activity guide, the Exit button that invokes this custom method exits the activity guide and the WorkCenter and returns the user to his or her default homepage.

Note: Exit is not a reserved method name and is different from ExitAGProcess, which is a reserved name.

```
Declare Function getHomepageURLForPortal PeopleCode FUNCLIB_PTBR.FUNCLIB FieldFormu⇒

la;

method Exit
   /+ &list_id as String +/
   /+ Returns String +/

   Return getHomepageURLForPortal(%Portal);

end-method;
```

Related Links

Defining Navigation and Pagelet Options
Defining Advanced Options
Configuring Related Data
“Think-Time Functions” (PeopleCode Developer’s Guide)
“GenerateHomepagePortalURL” (PeopleCode Language Reference)
“%Market” (PeopleCode Language Reference)

Dynamically Returning an Image URL as Context Data

For fluid activity guides only, you can define a method to use defined context data to dynamically return an image URL as additional context data.

In the following example, the EMPLID is defined as context data, which is used to dynamically generate the image URL to the employee's photo:

```
import PTAI_ACTION_ITEMS:*;
import PTAI_ACTION_ITEMS:AGInterface;
import PTAI_COLLECTION:Collection:*;
...

class AG implements PTAI_ACTION_ITEMS:AGInterface
   method getEmployePhoto(&oCtxt As PTAI_COLLECTION:Collection) Returns string;
   ...

end-class;


method getEmployePhoto
   /+ &oCtxt as PTAI_COLLECTION:Collection +/
   /+ Returns String +/
   Local integer &i = 1;
   While &i <= &oCtxt.Count
      If ((&oCtxt.Item(&i) As PTAI_ACTION_ITEMS:ContextData).fieldname = "EMPLID") ⇒

Then
         Local Rowset &rs = CreateRowset(Record.EMPL_PHOTO);
         Local string &strEmplID = (&oCtxt.Item(&i) As PTAI_ACTION_ITEMS:ContextDat⇒

a).keyValue;
         &rs.Fill("where EMPLID =:1", &strEmplID);
         Return %Response.GetImageURL(&rs.GetRow(1).GetRecord(Record.EMPL_PHOTO));
      End-If;
      &i = &i + 1;
   End-While;
end-method;
```

Dynamically Returning a Content URL to Be Displayed in a Modal Window

For fluid activity guides only, you can define a method to dynamically return a content URL to display additional information. Typically, this additional content is static text; however, it can include context data values.

Follow these guidelines:

- The content URL must point to a local, fluid component. Moreover, this component should contain one main fluid page only, with no footer, header, search, or side pages defined in the component.

  Therefore, the content URL cannot point to a classic component, a remote fluid component, or external, non-PeopleSoft content.

- The page can contain read-only content only. It cannot have any editable fields, links, buttons and other controls that will initiate a server trip or take the user out of activity guide context.

- Set permissions on the component content to be displayed to allow access by all activity guide contributors.

Implement an application class similar to the following. Invoke the GenerateComponentContentURL built-in function to return the URL to the component.

Note: The following code represents a simple implementation. Use the &oCtxt collection to obtain any of the context data values at runtime to include those values within the additional information displayed.

```
import PTAI_COLLECTION:Collection:*;
```

```
class My_Class
   method getContextURL(&oCtxt As PTAI_COLLECTION:Collection) Returns string;
end-class;

method getContextURL
   /+ &oCtxt as PTAI_COLLECTION:Collection +/
   /+ Returns String +/
   Return GenerateComponentContentURL(%Portal, %Node, MenuName.MY_MENU, "GBL", Comp⇒

onent.MY_CONTENT, Page.MY_CONTENT, "U");
end-method;
```

Related Links

"GenerateComponentContentURL" (PeopleCode Language Reference)

Performing Pre-Processing for the Activity Guide Pagelet

For classic activity guides, you can define a pagelet pre-processing method that will be executed every time an activity pagelet is loaded or reloaded in the WorkCenter pagelet area. This method *must* be named PageletPreProcess. The application class containing this method is identified on the Pagelet Options page.

In the following example for the Unified Navigation pagelet, each time the pagelet is reloaded, the PageletPreProcess method creates a link for Manage Related Content for each remote system configured under unified navigation.

```
import PTAI_ACTION_ITEMS:ActionItem;
import PTAI_ACTION_ITEMS:List;
import PTAI_ACTION_ITEMS:AGInterface;
import PTAI_ACTION_ITEMS:Constants;

class UninavPageletPreprocessor implements PTAI_ACTION_ITEMS:AGInterface
   method PageletPreProcess(&oList As PTAI_ACTION_ITEMS:List);
end-class;


<*
 method PageletPreprocess() - This method will be called everytime UniNav pagelet i⇒

s rendered. This function creates a new action item in the UniNav pagelet
for all the remote RC services. If an earlier created remote service is deleted the⇒

n the corresponding action item also will be deleted.
*>
method PageletPreProcess
   /+ &oList as PTAI_ACTION_ITEMS:List +/
   /+ Extends/implements PTAI_ACTION_ITEMS:AGInterface.PageletPreProcess +/

   Local SQL &sqlGetRemoteSvc;
   Local string &svcId, &svcName, &svcDescr, &svcParamname;
   Local boolean &boolStatus;
   Local integer &seq, &i;
   Local array of string &arrChildItems;;
   Local PTAI_ACTION_ITEMS:ActionItem &oItem = CreateObject("PTAI_ACTION_ITEMS:Acti⇒

onItem");
   Local PTAI_ACTION_ITEMS:ActionItem &oNewItem = CreateObject("PTAI_ACTION_ITEMS:A⇒

ctionItem");
   Local PTAI_ACTION_ITEMS:Constants &PTAI_CONSTANTS = CreateObject("PTAI_ACTION_IT⇒

EMS:Constants");

   <* Every time the pagelet is opened child items under the node 'Related Content ⇒
```



```
Setup' will be deleted and recreated. Nodes are deleted and added again
      everytime because, if some remote services are deleted then it will get refle⇒

cted in the tree immediately and to prevent the addition of duplicate nodes *>
   try
      &oItem.open("PAPP_UNITM8_PAPP_UNINA1003");
      &arrChildItems = &oItem.getChildActions();
      If All(&arrChildItems) Then
         For &i = 1 To &arrChildItems.Len
            &oItem.open(&arrChildItems [&i]);
            If Not &oItem.ItemId = "PAPP_UNITM9_PAPP_UNINA1003" Then
               try
                  &boolStatus = &oItem.delete();
               catch Exception &e1
               end-try;
            End-If;
         End-For;
      End-If;

      SQLExec("Select MAX(PTAI_SEQ) from PS_PTAI_ITEM where PTAI_LIST_ID = :1", &oL⇒

ist.ListId, &seq);
      &sqlGetRemoteSvc = GetSQL(SQL.EPPAI_GET_REMOTE_SVC);
      While &sqlGetRemoteSvc.Fetch(&svcId, &svcName, &svcDescr)
         &oNewItem.new(&svcId, &svcName, &oList.ListId);
         &oNewItem.DescrLong = &svcDescr;
         &oNewItem.ServiceId = &svcId;
         &oNewItem.ParentId = "PAPP_UNITM8_PAPP_UNINA1003";
         &oNewItem.Status = "1";
         &oNewItem.CreatedDateTime = %Datetime;
         &oNewItem.CreatedByOprid = %UserId;
         &oNewItem.LastUpdatedDateTime = %Datetime;
         &oNewItem.LastUpdatedByOprid = %UserId;
         &oNewItem.Target = "T";
         &oNewItem.Type = "C";
         Local PTAI_COLLECTION:Collection &oAssignmentColl = create PTAI_COLLECTION⇒

:Collection();
         Local PTAI_ACTION_ITEMS:ActionItemAssignments &oAssignment = create PTAI_A⇒

CTION_ITEMS:ActionItemAssignments();
         &oAssignment.AssignType = &PTAI_CONSTANTS.MEMBER_TYPE_ROLE;
         &oAssignment.AssignValue = "Portal Administrator";

         &oAssignmentColl = &oNewItem.getAssignments();
         &oAssignmentColl.InsertItem(&oAssignment As PTAI_COLLECTION:Collectable);
         &oNewItem.saveAssignments(&oAssignmentColl);
         &seq = &seq + 10;
         &oNewItem.Sequence = &seq;
         &boolStatus = &oNewItem.save();
      End-While;
      &sqlGetRemoteSvc.Close();
   catch Exception &e
   end-try;
end-method;
```

Related Links

Defining Navigation and Pagelet Options

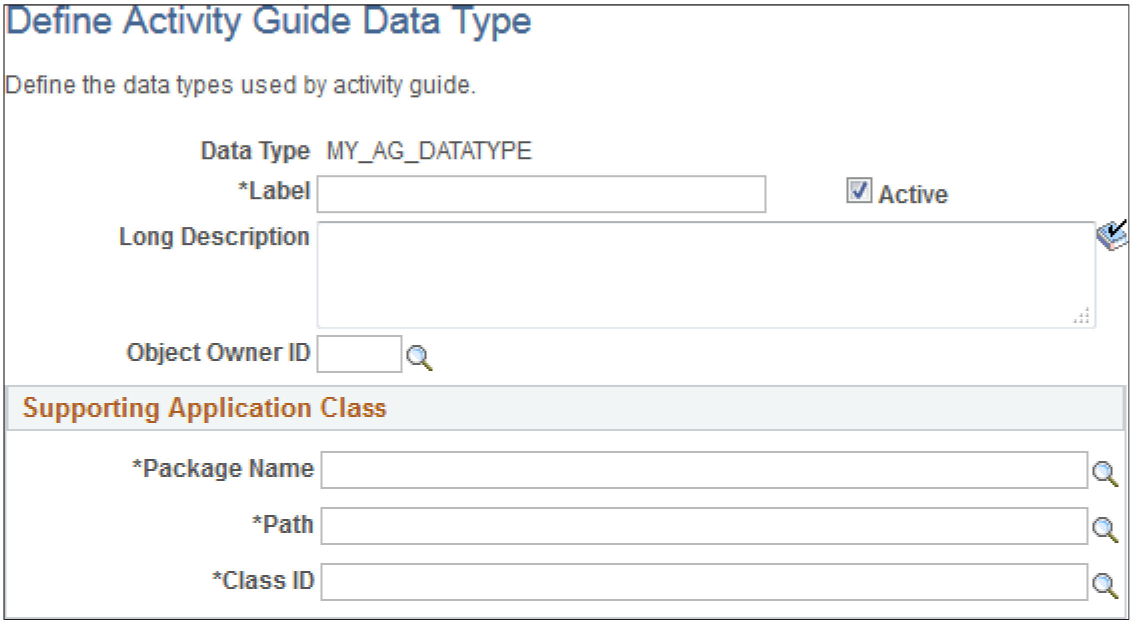
Creating a Custom Activity Guide Data Type

Use the Define Activity Guide Data Type page to define custom activity guide data types.

Navigation:

PeopleTools > **Activity Guides** > **Activity Guide Data Types**

This example illustrates the fields and controls on the Define Activity Guide Data Type page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Data Type	Displays the identifier entered for the custom data type on the Add a New Value page.
Label	Enter the descriptive name for the custom data type.
Long Description	Enter a description for the custom data type.
Object Owner ID	Select the object owner ID.
Package	Enter the name of the application package that contains your custom application class.
Path	Enter the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon.
Class ID	Enter the name of the application class that extends PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDataSource base class.

To implement a custom activity guide data type, create a PeopleCode application class definition:

1. Extend the PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDataSource base class. For example:

   ```
   import PTPP_PORTAL:UTILITY:Collection;
   import PTPP_COLLECTIONS:*;

   import PTGP_GUIDED_PROCESS:DataSources:*;
   import PTGP_GUIDED_PROCESS:GuidedProcesses:*;
   import PTGP_GUIDED_PROCESS:Elements:*;
   import PTGP_GUIDED_PROCESS:Buttons:*;
   import PTGP_GUIDED_PROCESS:Steps:*;

   class MyCustAGDataType extends PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDa⇒

   taSource
   ```

2. In the constructor method for your custom application class, set default values for only the properties necessary to differentiate it from the BaseDataSource class. For example:

   ```
   method MyCustAGDataType
      /+ &pId as String +/

      %Super = create PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDataSource(&pI⇒

   d);
      %This.setDataSourceType("MyCustAGDataType");
      %This.RenderType = %This.RENDER_TYPE_NONGUIDED_OPTIMIZED;

      /* Do not need to defer loading */
      %This.DeferMainGroupletsLoading = False;

   end-method;
   ```

3. Do not implement and thereby override the populateDataSourceObject method of PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDataSource.

   The populateDataSourceObject method loads critical activity guide template metadata.

4. Implement the initializeContainerComponent method manipulate the look of the container component using style sheets, JavaScript, or data source properties. For example:

   ```
   method initializeContainerComponent
      /+ Extends/implements PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDataSour⇒

   ce.initializeContainerComponent +/

      Local string &renderType, &scname;

      /* Call the super */
      %Super.initializeContainerComponent();

      /* Get the collection name */
      &scname = %Request.GetParameter(&cstQUERYPARAMETER_SCNAME);
      If (All(&scname)) Then
         %This.CollName = &scname;
      End-If;

      /* Override render type */
      &PTGP_NAVCOLL_RENDER_TYPE = "";
      &renderType = %Request.GetParameter(&cstQUERYPARAMETER_RENDERTYPE);
      If (All(&renderType)) Then
         try
            %This.RenderType = &renderType;
            &PTGP_NAVCOLL_RENDER_TYPE = &renderType;
         catch Exception &e
         end-try;
      End-If;
   ```

```
   /* Override panel type */
   %This.ShowSide1PanelCollapsible = True;
   %This.ShowSide1PanelOpenWhenCollapsible = ( Not %This.IsSmallFormFactorMode⇒

);

end-method;
```

5. Implement the initializeGroupletComponent method manipulate the look of the grouplet component using style sheets, JavaScript, or data source properties. While it must call the populateGuidedProcessObject to load the process tree, it must not load data source metadata. For example:

```
method initializeGroupletComponent
   /+ Extends/implements PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDataSour⇒

ce.initializeGroupletComponent +/

   Local string &scname;

   /* Override render type */
   If (All(&PTGP_NAVCOLL_RENDER_TYPE)) Then
      %This.RenderType = &PTGP_NAVCOLL_RENDER_TYPE;
   End-If;

   /* Get the collection name */
   &scname = %Request.GetParameter(&cstQUERYPARAMETER_SCNAME);
   If (All(&scname)) Then
      %This.CollName = &scname;
   End-If;

   %Super.initializeGroupletComponent();

end-method;
```

6. Implement the populateGuidedProcessObject method to load the process tree from your custom data source. For example:

```
method populateGuidedProcessObject
   /+ Returns PTGP_GUIDED_PROCESS:GuidedProcesses:BaseGuidedProcess +/
   /+ Extends/implements PTGP_GUIDED_PROCESS:DataSources:ActivityGuideDataSour⇒

ce.populateGuidedProcessObject +/

   Local string &text, &iconUrl, &msgNode;
   Local string &defaultFolderIconUrl, &defaultContentIconUrl;
   Local boolean &succeeded;

   Local Record &rec;

   Local PTGP_GUIDED_PROCESS:GuidedProcesses:BaseGuidedProcess &thisProcess;

   Local PTGP_GUIDED_PROCESS:Elements:StepElement &thisStep;
   Local PTGP_GUIDED_PROCESS:Elements:StepGroupElement &thisStepGroup, &theRoo⇒

tStep;
   Local PTGP_GUIDED_PROCESS:Elements:ContextTextElement &headerText;
   Local PTGP_GUIDED_PROCESS:Elements:ButtonElement &thisButton;

   Local PTPP_COLLECTIONS:NavigationCollection &thisNavColl;
   Local PTPP_COLLECTIONS:Folder &thisFolder;
   Local PTPP_COLLECTIONS:Shortcut &thisShortcut;


   &thisProcess = create PTGP_GUIDED_PROCESS:GuidedProcesses:BaseGuidedProcess⇒

("PTGPTester");
```



```
   &theRootStep = &thisProcess.RootStep;


   Evaluate %This.RenderType
   When %This.RENDER_TYPE_HORIZONTAL
   When %This.RENDER_TYPE_VERTICAL_OPTIMIZED_SEQUENTIAL
   When %This.RENDER_TYPE_VERTICAL_NONOPTIMIZED_SEQUENTIAL
      &thisProcess.IsSequential = False;
      Break;

   When %This.RENDER_TYPE_VERTICAL_OPTIMIZED
   When %This.RENDER_TYPE_VERTICAL_NONOPTIMIZED
   When %This.RENDER_TYPE_NONGUIDED_OPTIMIZED
   When %This.RENDER_TYPE_NONGUIDED_NONOPTIMIZED
      Break;

   End-Evaluate;


   /* Default icons */
   &defaultFolderIconUrl = %Response.GetImageURL(Image.PTPP_FN_LARGE_FOLDER_IC⇒

N);
   &defaultContentIconUrl = %Response.GetImageURL(Image.PTPP_FN_LARGE_CONTENT_⇒

ICN);

   &rec = CreateRecord(Record.PTPP_SITE_OPT);
   &rec.PORTAL_NAME.Value = %Portal;
   If (&rec.SelectByKey()) Then
      /* Site default */
      &defaultFolderIconUrl = %Response.GetImageURL(&rec.PTPP_SCLGFLDICN.Value⇒

);
      &defaultContentIconUrl = %Response.GetImageURL(&rec.PTPP_SCLGCNTICN.Valu⇒

e);
   Else
      SQLExec("SELECT MSGNODENAME FROM PS_PTPP_OPTIONS", &msgNode);
      &rec = CreateRecord(Record.PTPP_OPTIONS);
      &rec.MSGNODENAME.Value = &msgNode;
      If (&rec.SelectByKey()) Then
         /* System default */
         &defaultFolderIconUrl = %Response.GetImageURL(&rec.PTPP_SCLGFLDICN.Va⇒

lue);
         &defaultContentIconUrl = %Response.GetImageURL(&rec.PTPP_SCLGCNTICN.V⇒

alue);
      End-If;
   End-If;


   /* Get the Collection */
   try
      &thisNavColl = create PTPP_COLLECTIONS:NavigationCollection(%This.Portal⇒

Name, %This.CollName);
   catch Exception &ex1
      /* Ignore any error */
   end-try;


   /* Generate the list */
   If ((&thisNavColl <> Null) And
         &thisNavColl.Authorized And
         &thisNavColl.IsValid) Then

      /* Root steps */
      If (&thisNavColl.Shortcuts <> Null) Then
```

```
         &thisShortcut = &thisNavColl.Shortcuts.First();
         While (&thisShortcut <> Null)

            /* Create the step */
            &thisStep = create PTGP_GUIDED_PROCESS:Elements:StepElement(&thisS⇒

hortcut.Name);
            &thisStep.Label = UnEscapeHTML(&thisShortcut.Label);
            &thisStep.URL = &thisShortcut.AbsoluteContentURL;
            &thisStep.SequenceNumber = &thisShortcut.SequenceNumber;
            &thisStep.IsFluid = &thisShortcut.IsFluid;

            If (&thisShortcut.IsNewWindow) Then
               &thisStep.OnClick = "LaunchURL(null, '" | EscapeJavascriptStrin⇒

g(&thisShortcut.AbsolutePortalURLnewWin) | "', 1);";
            End-If;

            If (&thisShortcut.IsTopWindow) Then
               &thisStep.OnClick = "LaunchURL(null, '" | EscapeJavascriptStrin⇒

g(&thisShortcut.AbsolutePortalURL) | "', 4);";
            End-If;

            &iconUrl = &thisShortcut.ImageURL;
            If (All(&iconUrl)) Then
               &thisStep.IconUrl = &iconUrl;
            Else
               &thisStep.IconUrl = &defaultContentIconUrl;
            End-If;

            &theRootStep.addChildStep(&thisStep);

            /* Next shortcut */
            &thisShortcut = &thisNavColl.Shortcuts.Next();

         End-While;

      End-If;


      /* Root folders */
      If (&thisNavColl.Folders <> Null) Then

         &thisFolder = &thisNavColl.Folders.First();
         While (&thisFolder <> Null)

            /* Create the step group */
            &thisStepGroup = create PTGP_GUIDED_PROCESS:Elements:StepGroupElem⇒

ent(&thisFolder.Name);
            &thisStepGroup.Label = UnEscapeHTML(&thisFolder.Label);
            &thisStepGroup.SequenceNumber = &thisFolder.SequenceNumber;

            &iconUrl = &thisFolder.ImageURL;
            If (All(&iconUrl)) Then
               &thisStepGroup.IconUrl = &iconUrl;
            Else
               &thisStepGroup.IconUrl = &defaultFolderIconUrl;
            End-If;

            /* First level child steps */
            If (&thisFolder.Shortcuts <> Null) Then

               &thisShortcut = &thisFolder.Shortcuts.First();
               While (&thisShortcut <> Null)

                  /* Create the step */
                  &thisStep = create PTGP_GUIDED_PROCESS:Elements:StepElement(⇒
```



```
&thisShortcut.Name);
                  &thisStep.Label = UnEscapeHTML(&thisShortcut.Label);
                  &thisStep.URL = &thisShortcut.AbsoluteContentURL;
                  &thisStep.SequenceNumber = &thisShortcut.SequenceNumber;
                  &thisStep.IsFluid = &thisShortcut.IsFluid;
                  &thisStepGroup.addChildStep(&thisStep);

                  If (&thisShortcut.IsNewWindow) Then
                     &thisStep.OnClick = "LaunchURL(null, '" | EscapeJavascrip⇒

tString(&thisShortcut.AbsolutePortalURLnewWin) | "', 1);";
                  End-If;

                  If (&thisShortcut.IsTopWindow) Then
                     &thisStep.OnClick = "LaunchURL(null, '" | EscapeJavascrip⇒

tString(&thisShortcut.AbsolutePortalURL) | "', 4);";
                  End-If;

                  /* Next shortcut */
                  &thisShortcut = &thisFolder.Shortcuts.Next();

               End-While;

            End-If;

            /* Sort and add the step group when it's not empty */
            If (&thisStepGroup.ChildSteps.Count > 0) Then
               &succeeded = &thisStepGroup.ChildSteps.sort(&cstSEQ_NUMBER, Tru⇒

e);
               &theRootStep.addChildStep(&thisStepGroup);
            End-If;

            /* Next folder */
            &thisFolder = &thisNavColl.Folders.Next();

         End-While;

      End-If;


      /* Sort the root list */
      &succeeded = &theRootStep.ChildSteps.sort(&cstSEQ_NUMBER, True);


      /* Set the current step id to the first item */
      &thisStep = &theRootStep.ChildSteps.get(1);
      If (All(&thisStep)) Then
         &thisStepGroup = (&thisStep As PTGP_GUIDED_PROCESS:Elements:StepGroup⇒

Element);
         If (All(&thisStepGroup)) Then
            &thisProcess.CurrentStepId = &thisStepGroup.ChildSteps.get(1).ID;
         Else
            &thisProcess.CurrentStepId = &thisStep.ID;
         End-If;
      End-If;


      /* Header texts */
      &headerText = create PTGP_GUIDED_PROCESS:Elements:ContextTextElement("la⇒

bel");
      &headerText.StyleClass = "ps_ag-header-context-title";
      &headerText.Text = &thisNavColl.Label;
      &thisProcess.addHeaderText(&headerText);

      &text = &thisNavColl.Description;
      If (All(&text)) Then
```

```
         &headerText = create PTGP_GUIDED_PROCESS:Elements:ContextTextElement(⇒

"description");
         &headerText.StyleClass = "ps_ag-header-context-text";
         &headerText.Text = &text;
         &thisProcess.addHeaderText(&headerText);
      End-If;


      /* Header buttons */
      &thisButton = create PTGP_GUIDED_PROCESS:Buttons:ExitButton(&thisProcess⇒

.ID | "-ExitButton");
      &thisProcess.addHeaderButton(&thisButton);

      &thisButton = create PTGP_GUIDED_PROCESS:Buttons:PreviousButton(&thisPro⇒

cess.ID | "-PreviousButton");
      &thisProcess.addHeaderButton(&thisButton);

      &thisButton = create PTGP_GUIDED_PROCESS:Buttons:NextButton(&thisProcess⇒

.ID | "-NextButton");
      &thisProcess.addHeaderButton(&thisButton);

   End-If;


   Return &thisProcess;

end-method;
```

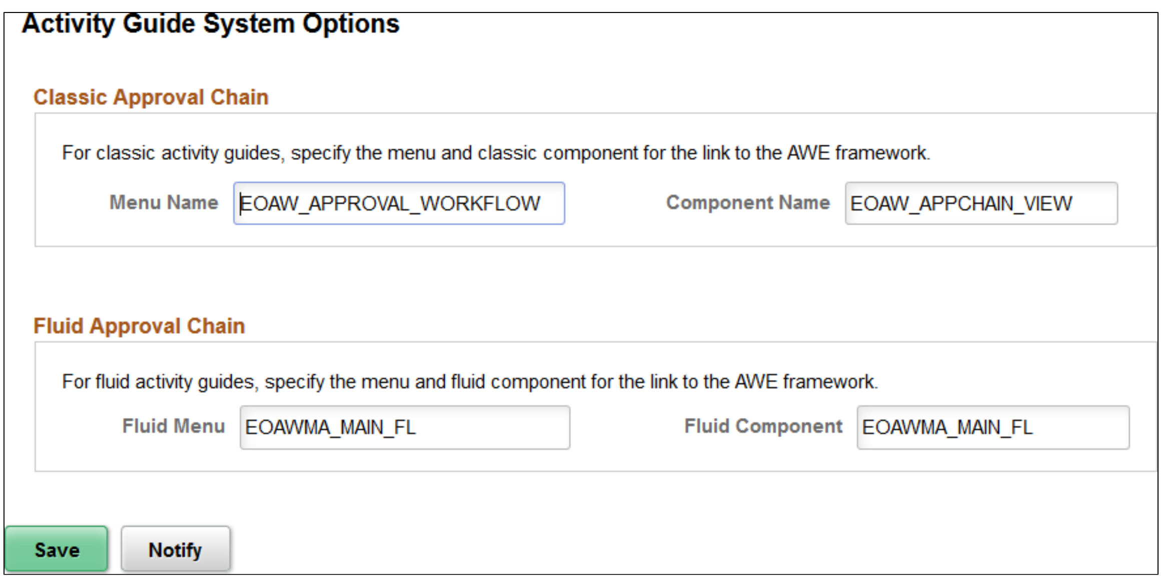
Defining Activity Guide System Options

Use the Activity Guide System Options page to define integration with Approval Workflow Engine (AWE).

Navigation:

PeopleTools > **Activity Guides** > **Activity Guide System Options**

This example illustrates the fields and controls on the Activity Guide System Options page. You can find definitions for the fields and controls later on this page.



Classic Approval Chain

Field or Control	***Description***
Menu Name	Enter the menu name that contains the classic AWE component—for example, EOAW_APPROVAL_WORKFLOW.
Component Name	Enter the name of the classic AWE component—for example, EOAW_APPCHAIN_VIEW.

Fluid Approval Chain

Field or Control	***Description***
Fluid Menu	Enter the menu name that contains the fluid AWE component—for example, EOAWMA_MAIN_FL.
Fluid Component	Enter the name of the fluid AWE component—for example, EOAWMA_MAIN_FL.

Creating Visualizations from Activity Guide Instance Data

You may wish to generate analytics or to report on the progress of activity guide instances, steps, statuses, and so on. For example, your analytics could show instances completed, progress within instances from a specific template, and so on. The Activity Guide search definition type enables you to index activity guide instance data in search engine that then can be used for analytics in PeopleSoft Insights. You can also use an application table as an additional data source in order to provide supplemental information in the index. After creating a search definition and indexing it, you can create visualizations.

Related Links

“Understanding Activity Guide Search Definitions” (Search Technology)

“Understanding Application Data and Insights Dashboards” (Search Technology)

Migrating and Updating Activity Guide Data

Understanding Activity Guide Data

Activity guides can consist of the following related objects, some of which are ADS objects and others of which are managed objects:

ADS Objects	***Managed Objects***
Activity guide templates and instances	Application classes
Action items	Permission lists
Related content services	PeopleSoft queries
WorkCenters definitions	Component definitions
Pagelet Wizard objects	Message catalog entries
Scheduler objects	iScripts
	Process definitions
	Component interfaces

Oracle may deliver new or updated activity guides in a PeopleSoft update image. When such an update is selected, all of the updated objects will be included in the change package.

However, when you migrate your activity guide data between databases, you may need to use both Data Migration Workbench and Application Designer to migrate projects containing related object definitions. With Data Migration Workbench, you can migrate:

- Activity guide template definitions (with or without associated action item definitions).
- Action item definitions.
- Activity guide instance definitions (with or without associated action item definitions).

Important! Oracle recommends that you do not migrate activity guide instances between databases. If you must do so, see “Resetting the Last Instance ID After Migration” at the end of the following section to ensure that new instances are generated with a valid instance ID.

Migrating Activity Guide Data

For example, to migrate an activity guide template definition and its associated action item definitions:

1. Select **PeopleTools** > **Lifecycle Tools** > **Migrate Data** > **Data Migration Workbench**.
2. Click the **Add a New Value** link.
3. Enter a project name and description.
4. On the Project Definition page, select *Activity Guide List* as the Data Set Name.
5. Go to the Data Set Content page.

6. Click the Insert Content button.
7. Search for the activity guide template that you want to migrate.

Note: Search results can display both templates and instances depending on your search criteria. If you don't know the ID for the activity guide template or instance, select the *All* operator.

---

8. Select the activity guide template that you want to migrate.
9. Click the Insert and Return button to insert the selected items.
10. Return to the Project Definition page.
11. Add a row to the Project Content grid and select *Activity Guide Item* as the Data Set Name.
12. Return to the Data Set Content page.
13. Ensure that *ACTIVITY_GUIDE_ITEM* is selected and click the Insert Content button.
14. Search for only the action items that correspond to the template that you want to migrate:

Note: Search results will display action items from both templates and instances depending on your search criteria.

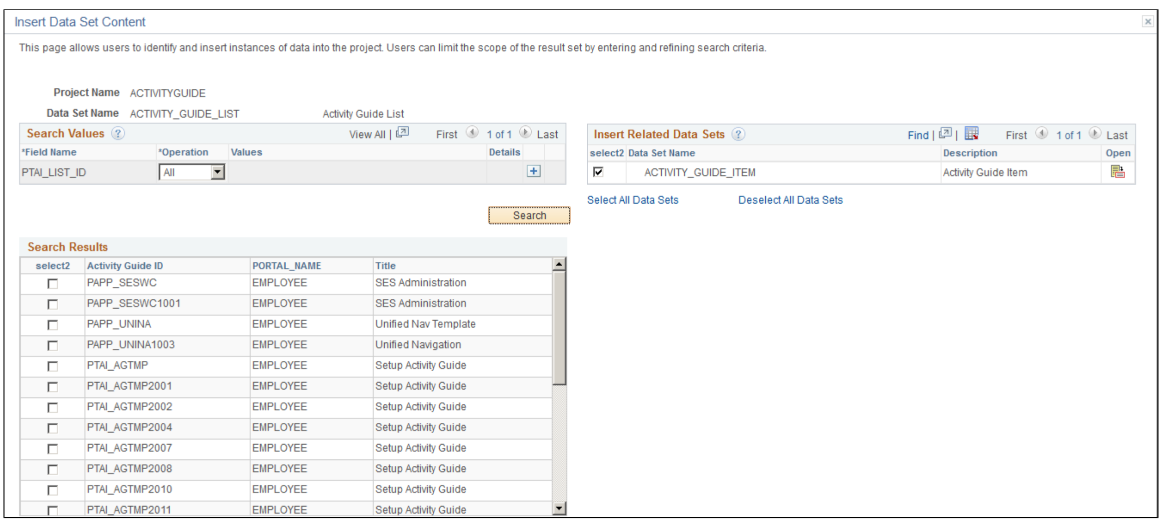
---

a. For the PTAI_ITEM_ID field, select the *All* operator.

b. Add a row to the search values and select PTAI_LIST_ID as the field name.

c. Select the = and specify the template ID.

The follow example illustrates search values on the Insert Data Set Content page.



15. Select the appropriate actions items or click the Select All button.
16. Click the Insert and Return button to insert the selected items.
17. Click the Save button to save the data set.
18. Click the Copy To File button to copy the data set to a file.

19. Select the location for the file.

20. Click Run.

21. On the Process Scheduler Request page, click OK to start the copy process.

22. Once the project has been copied to a file, use Data Migration Workbench on the target database to compare the project data and copy it into the database. See “Copy Compare Overview” (Lifecycle Management Guide).

Managed objects associated with the activity guide template, action items, or instances migrated must be migrated separately via an Application Designer project. Search for and migrate objects identified by each action item’s related content service definition—for example, component definitions. In addition, search for and migrate application class PeopleCode that can be associated with:

- Instance creation.
- Image context data.
- Custom activity guide actions at the instance level.
- Custom activity guide actions at the action item level.
- Dynamically returning a content URL to display additional information.
- Action item post processing.
- Approval Workflow Engine (AWE) approval handlers

Resetting the Last Instance ID After Migration

When you migrate activity guide instances from one database to another, the last list ID may not be synchronized in the two tables where it is used: the PS_PTAI_LIST table and the PS_PTAI_OPTIONS table. When the values are not synchronized, new instances may be created using invalid instance IDs.

Use the following SQL to synchronize the last list ID. Find the maximum list ID in the PS_PTAI_LIST table:

```
SELECT MAX((SUBSTR(PTAI_LIST_ID, 11, LENGTH(PTAI_LIST_ID)))) FROM PS_PTAI_LIST;
```

Then, use this value to update the PS_PTAI_OPTIONS table:

```
UPDATE PS_PTAI_OPTIONS SET PTAI_LAST_LIST_ID=last_ID_from_PS_PTAI_LIST WHERE PTAI_L⇒

AST_ITEM_ID=2000
```

Related Links

“Defining ADS Project” (Lifecycle Management Guide)
“Defining Data Set Content” (Lifecycle Management Guide)
“Inserting Data Content” (Lifecycle Management Guide)
“Copying Project to File” (Lifecycle Management Guide)

Updating Activity Guide Data via PeopleSoft Update Images

Oracle may deliver new or updated activity guides in a PeopleSoft update image. When such an update is selected, all of the updated objects will be included in the change package.

Oracle recommends that you:

- Do not delete any PeopleSoft delivered objects.
- Restrict your changes to configurable fields specified for each definition because these configuration changes are preserved
- Avoid modification to any field that is in the non-reconcilable section because any modification to these fields is considered to be a customization and can be overwritten during an update.

The following table lists some of the configurable and non-reconcilable changes associated with an activity guide template.

Configurable Changes	***Non-reconcilable Changes***
Modifying the template or instance attributes such as the active setting, side panel setting, restart setting, notifications setting, the context data display option, the application package and method used to dynamically return a content URL to display additional information, and so on.	Changing the application package and application class for a navigation button.
Modifying the context field label, context field label visibility, context image application package details, and which context field is the primary field.	Changing the related content service for a delivered action item.
Modifying the label, active setting, or message set and number for a custom action button for a template or instance.	Modifying styles.
Modifying the question text for a template or instance.	Modifying context data.
Modifying an action item's attributes such as required, active, inactivate after due date, lock after completion, status, process type, pause after, pause before, and so on.	
Modifying the context parameter type or parameter value for an action item.	
Modifying the label, active setting, or message set and number for a custom action button for an action item.	
Modifying the AWE approval configuration, question ID and answer association, and configurable field values for an action item.	

The following sections include details on the configurable fields associated with the three activity guide related data sets:

- ACTIVITY_GUIDE_DATATYPE
- ACTIVITY_GUIDE_ITEM
- ACTIVITY_GUIDE_LIST

Any field not listed is considered non-reconcilable.

ACTIVITY_GUIDE_DATATYPE Data Set

The ACTIVITY_GUIDE_DATATYPE data set includes the data that define activity guide data type definitions. This data set does not contain any merge groups.

ACTIVITY_GUIDE_ITEM Data Set

The ACTIVITY_GUIDE_ITEM data set includes the data that define action items. The data set consists of four merge groups:

- PTAI_HIER_GROUP
- PTAI_ITEM_CNTXT_GROUP
- PTAI_ITEM_GROUP
- PTAI_OVRPGT_BTN_GROUP

PTAI_HIER_GROUP Merge Group

The PTAI_HIER_GROUP merge group includes the PTAI_HIER record definition with one configurable field: PTAI_ACTIVE_FLG.

PTAI_ITEM_CNTXT_GROUP Merge Group

The PTAI_ITEM_CNTXT_GROUP merge group includes the PTAI_ITEM_CNTXT record definition with the following configurable fields:

- PTAI_PARAM_TYPE
- PTAI_PARAM_VALUE

PTAI_ITEM_GROUP Merge Group

The PTAI_ITEM_GROUP merge group includes the PTAI_ITEM record definition with the following configurable fields:

- APPCLASSID
- PACKAGEROOT
- PTAI_ACTIVE_FLG
- PTAI_ANSWER
- PTAI_ASSIGN_OPRID

- PTAI_ASSIGN_TYPE
- PTAI_AWE_APPROVAL
- PTAI_AWE_CLASSID
- PTAI_AWE_DENIAL
- PTAI_AWE_PACKAGE
- PTAI_AWE_PATH
- PTAI_AWE_PROC_ID
- PTAI_CATEGORY
- PTAI_DESCRLONG
- PTAI_DUE_DATE
- PTAI_DUE_TIME
- PTAI_FIELD1 *to* PTAI_FIELD10
- PTAI_IMAGE
- PTAI_INACTIVAFTDUE
- PTAI_LABEL
- PTAI_LOCKAFTRCOMPL
- PTAI_PAUSE_AFT_FLG
- PTAI_PAUSE_BEF_FLG
- PTAI_PERC_COMPL
- PTAI_PRIORITY
- PTAI_QUESTION_ID
- PTAI_REQD
- PTAI_RESUM_ERR_FLG
- PTAI_RNTM_INP_FLG
- PTAI_SKIP_ERR_FLG
- PTAI_START_DT
- PTAI_START_TIME
- PTAI_STATUS
- QUALIFYPATH

It also includes the PTAI_ITEM_LNG record definition with the following configurable fields:

- PTAI_DESCRLONG
- PTAI_FIELD1 *to* PTAI_FIELD5
- PTAI_LABEL

PTAI_OVRPGT_BTN_GROUP Merge Group

The PTAI_OVRPGT_BTN_GROUP merge group includes the PTAI_OVRPGTBTLG record definition with one configurable field: PTAI_BTN_LBL.

It also includes the PTAI_OVRPGT_BTN record definition with the following configurable fields:

- MESSAGE_NBR
- MESSAGE_SET_NBR
- PTAI_ACTIVE_FLG
- PTAI_BTN_LBL

ACTIVITY_GUIDE_LIST Data Set

The ACTIVITY_GUIDE_ITEM data set includes the data that define activity guide templates and instances. The data set consists of five merge groups:

- PTAI_CONTEXT_GROUP
- PTAI_LIST_GROUP
- PTAI_LIST_PRIV_GROUP
- PTAI_PGLT_BTN_GROUP
- PTAI_QUES_LIST_GROUP

PTAI_CONTEXT_GROUP Merge Group

The PTAI_CONTEXT_GROUP merge group includes the PTAI_PTAI_CNTXT_LNG record definition with one configurable field: PTAI_KEY_LABEL.

It also includes the PTAI_CONTEXT record definition with the following configurable fields:

- PTAI_IMAGE_TYPE
- PTAI_IMG_CLASSID
- PTAI_IMG_METHOD
- PTAI_IMG_NAME
- PTAI_IMG_PACKAGE
- PTAI_IMG_PATH
- PTAI_IMG_URL

- PTAI_KEY_LABEL
- PTAI_LABEL_CHK
- PTAI_PRIMARY_DATA

PTAI_LIST_GROUP Merge Group

The PTAI_LIST_GROUP merge group includes the PTAI_LIST record definition with the following configurable fields:

- APPCLASSID
- APPCLASSMETHOD
- PACKAGEROOT
- PTAI_ACTIVE_FLG
- PTAI_CTX_CLASSID
- PTAI_CTX_DISP_OPT
- PTAI_CTX_METHOD
- PTAI_CTX_PKGROOT
- PTAI_CTX_QUALPATH
- PTAI_DESCRLONG
- PTAI_FIXEDSIDEPNL
- PTAI_INST_CLASSID
- PTAI_INST_PKGROOT
- PTAI_INST_QUALPATH
- PTAI_LABEL
- PTAI_NOTIFY_TYPE
- PTAI_PBVISIBLE
- PTAI_RESTARTAG
- PTAI_SHOWFLNOTIFY
- QUALIFYPATH
- STYLESHEETNAME

It also includes the PTAI_LIST_LNG record definition with the following configurable fields:

- PTAI_DESCRLONG

- PTAI_LABEL

PTAI_LIST_PRIV_GROUP Merge Group

The PTAI_LIST_PRIV_GROUP merge group includes the PTAI_LIST_PRIV record definition with one configurable field: PTAI_PRIVSET_ID.

PTAI_PGLT_BTN_GROUP Merge Group

The PTAI_PGLT_BTN_GROUP merge group includes the PTAI_PGLTBTNLNG record definition with one configurable field: PTAI_BTN_LBL.

It also includes the PTAI_PGLT_BTN record definition with the following configurable fields:

- MESSAGE_NBR
- MESSAGE_SET_NBR
- PTAI_ACTIVE_FLG
- PTAI_BTN_LBL

PTAI_QUES_LIST_GROUP Merge Group

The PTAI_QUES_LIST_GROUP merge group includes the PTAI_QUES_LIST record definition with the following configurable fields:

- PTAI_QUESTION
- PTAI_ANSWER

It also includes the PTAI_QUES_LNG record definition with one configurable field: PTAI_QUESTION.

Chapter 24

Working with Oracle Guided Learning (OGL)

Understanding OGL in a PeopleSoft Environment

Oracle Guided Learning (OGL) is an enterprise cloud platform that supports the creation and delivery of personalized, guided, and contextual user assistance. OGL is an information and learning interface that can be embedded in PeopleSoft applications.

Prerequisites

An OGL account and appId are required to implement OGL on a PeopleSoft system. Contact your Oracle Account Manager to request an OGL account.

Required Permissions

Setting up OGL requires the following permissions:

- Access to Application Designer to implement OGL JavaScript and PeopleCode.
- PTPT1300 (Portal Administrator) permission list to access the OGL Configuration page.
- If you have custom branding, the Secure Branding Administrator role to access the Branding Headers and Footers page.

For More Information

Additional resources provide you with information on using the OGL Editor to develop OGL guides as well as specific guidelines for developing guides for a PeopleSoft system.

Related Links

Oracle Guided Learning User Guide
Guidelines for Implementing Oracle Guided Learning in a PeopleSoft Environment (Doc ID 2799238.1)

Setting up OGL

Delivered JavaScript and PeopleCode definitions simplify configuration of Oracle Guided Learning on a PeopleSoft system.

To configure OGL on your PeopleSoft system, you will:

1. Implement OGL JavaScript.
2. Implement OGL PeopleCode.

3. (Optional) Add OGL to custom branding.
4. Configure the system and enable OGL.

Important! If you previously implemented OGL on a release prior to PeopleTools 8.62, see Considerations When Upgrading OGL from a Previous Release to disable your current configuration before proceeding.

Implementing Your OGL JavaScript

To implement your OGL JavaScript, copy and modify sample HTML definitions delivered in PeopleTools.

There is one HTML definition for fluid and another for classic. In both definitions, you are required to set three items:

- Your OGL appId.
- The OGL environment type (dev or prod).
- The player URL for your OGL domain.

Other custom JavaScript modifications—such as passing additional parameters—are optional.

To implement your OGL JavaScript:

1. In Application Designer, open the PT_OGL_FLUID_JS_SAMPLE HTML definition.
2. Save the definition with a new name—for example, MY_OGL_FLUID_JS.
3. In Application Designer, open the PT_OGL_CLASSIC_JS_SAMPLE HTML definition.
4. Save the definition with a new name—for example, MY_OGL_CLASSIC_JS.
5. In both versions of the JavaScript:
   - Remove the inline commenting and then the global beginning and end comments as instructed in the file.
   - Change the appId to the value provided to you by your OGL Administrator. The appId is a unique identifier for your provisioned OGL account.

     Note: In a clustered environment, you may have a separate appId for each system in the cluster.

   - Set the env value to dev or prod as is appropriate for this PeopleSoft system.

     This value determines the OGL content that will be displayed on this system. The dev setting displays all guides, including drafts and unpublished revisions; prod displays published content only.

   - Set the player URL to match the domain you use to access the OGL console and editor:
     - NAMER – `https://guidedlearning.oracle.com/player/latest/static/js/iridizeLoader.min.js`

  - EMEA – `https://guidedlearning-emea.oracle.com/player/latest/static/js/iridizeLoader.min.js`
  - APAC – `https://guidedlearning-apac.oracle.com/player/latest/static/js/iridizeLoader.min.js`

- Review the variables being passed to the iridize API for use in guide activation settings and determine whether they are appropriate for your implementation. Match any changes to the bind variables being passed into the JavaScript via PeopleCode with updates to these variables.
- Save your changes.

The following example shows the fluid JavaScript code configured for a production environment in the NAMER region:

```
setTimeout(function () {
top.iridize=top.iridize||function(e,t,n){return top.iridize.api.call(e,t,n);};
top.iridize.api=top.iridize.api||{q:[],call:function(e,t,n){top.iridize.api.q.push(⇒

{method:e,data:t,callback:n});}}};
/* Do not modify the code above this comment. */

/* Use the appId provided to you by your OGL administrator. */
top.iridize.appId="nnnnnnnnn";

/* Set to dev for development environments. Set to prod for published guides only. ⇒

*/
top.iridize.env="prod";

/* Bind variables passed in by PeopleCode */
top.userID = "%Bind(:1)";
top.roles = "%Bind(:2)";
top.show_guide = "%Bind(:3)";

/* Pass PeopleSoft data to the iridize API as required for guide activation conditi⇒

ons.
* user_id used internally by OGL and not available as an activation condition.
* user_roles demonstrates the availability of a subset of a user's PeopleSoft roles⇒

.
* show_welcome_msg demonstrates using a Boolean value as an activation condition. *⇒

/
top.ir_fields = {user_id:userID,user_roles:roles,show_welcome_msg:show_guide};
top.iridize("api.fields.set", top.ir_fields);

/* Set the player URL for your OGL domain here. */
playerURL="https://guidedlearning.oracle.com/player/latest/static/js/iridizeLoader.⇒

min.js";
// playerURL="https://guidedlearning-emea.oracle.com/player/latest/static/js/iridiz⇒

eLoader.min.js";
// playerURL="https://guidedlearning-apac.oracle.com/player/latest/static/js/iridiz⇒

eLoader.min.js";

/* Do not modify the code below this comment. */
var e=top.document.createElement("script");
e.src=playerURL;
e.type="text/javascript";
```



```
e.async=true;
top.document.head.appendChild(e);
}, 0);
```

Implementing Your OGL PeopleCode

To implement your OGL PeopleCode, copy and modify the sample PT_OGL_TEMPLATE application class delivered in PeopleTools.

To implement your OGL PeopleCode:

1. Create and save a new application package—for example, MY_OGL.
2. Add a new application class to the application package—for example, MY_OGL.
3. Open the delivered PT_OGL_HOOK application package. Open the sample PT_OGL_TEMPLATE application class within that package.
4. Copy the sample code and paste it into your new application class.
5. Make the following updates:
    - Change the class and constructor name to your new class name.
    - In the execute method, specify the name of your fluid JavaScript definition.
    - In the get_HTML method, specify the name of your classic JavaScript definition.
    - Remove the beginning and end comments.
    - Save and validate your PeopleCode program.
6. Optionally, review the custom activation conditions. If you require additional PeopleSoft data for activation conditions, match any updates to bind variables that you make in the PeopleCode with similar updates in both JavaScript definitions.

The following example shows an implementation of the MY_OGL:MY_OGL class with no additional bind variables.

```
import PT_RCF:ServiceInterface;

class MY_OGL implements PT_RCF:ServiceInterface
   method MY_OGL();
   method custom_activation_conditions();
   method execute();
   method get_HTML() Returns string;

private
   instance string &user_id;
   instance string &str_roles;
   instance boolean &show_welcome_msg;
end-class;

method MY_OGL
   %Super = create PT_RCF:ServiceInterface();
   &user_id = EscapeJavascriptString(%UserId);
   %This.custom_activation_conditions();
end-method;

method custom_activation_conditions
```



```
   /*
   rem Example code demonstrating passing a subset of the roles;
   rem Create a comma-delimited string of the user roles;
   &str_roles = "";
   Local array of string &roles = %Roles;
   Local integer &i;
   rem Truncate to 3 roles due to the 300-character limit of the iridize API;
   For &i = 1 To 3
      If &i = 1 Then
         &str_roles = &str_roles | EscapeJavascriptString(&roles [&i]);
      Else
         &str_roles = &str_roles | "," | EscapeJavascriptString(&roles [&i]);
      End-If;
   End-For;


   rem Example code demonstrating a boolean condition;
   rem Display the welcome message to specific users only;
   If &user_id = "CADAMS" Then
      &show_welcome_msg = True;
   End-If;
   */
end-method;

method execute
   /+ Extends/implements PT_RCF:ServiceInterface.execute +/

   /* Method to add JavaScript for fluid only. */
   /* When in a CI, do not execute this code. */
   If IsFluidMode(%Component, %Market) Then
      If %CompIntfcName = "" Then
         If %Request <> Null Then
            If Not (IsModeless() Or
                  IsGroupletRequest() Or
                  IsAGRequest() Or
                  IsMDRequest() Or
                  IsModalComponent()) Then

               AddOnLoadScript(GetHTMLText(HTML.MY_OGL_FLUID_JS, &user_id, &str_rol⇒

es, &show_welcome_msg));
            End-If;
         End-If;
      End-If;
   End-If;
end-method;

method get_HTML
   /+ Returns String +/

   rem Method to add JavaScript for classic only;
   If %Request <> Null Then
      Return GetHTMLText(HTML.MY_OGL_CLASSIC_JS, &user_id, &str_roles, &show_welcom⇒

e_msg);
   End-If;
end-method;
```

Adding OGL to Custom Branding

If you have custom branding *and* you are not upgrading OGL from a release prior to PeopleTools 8.62, then you must follow the procedure documented in this topic. If you have custom branding *and* you are upgrading OGL from a release prior to PeopleTools 8.62, see Considerations When Upgrading OGL from a Previous Release.

To operate on classic pages, the OGL JavaScript must be injected into a HTML element in the branding header definition.

To add OGL JavaScript to your custom branding:

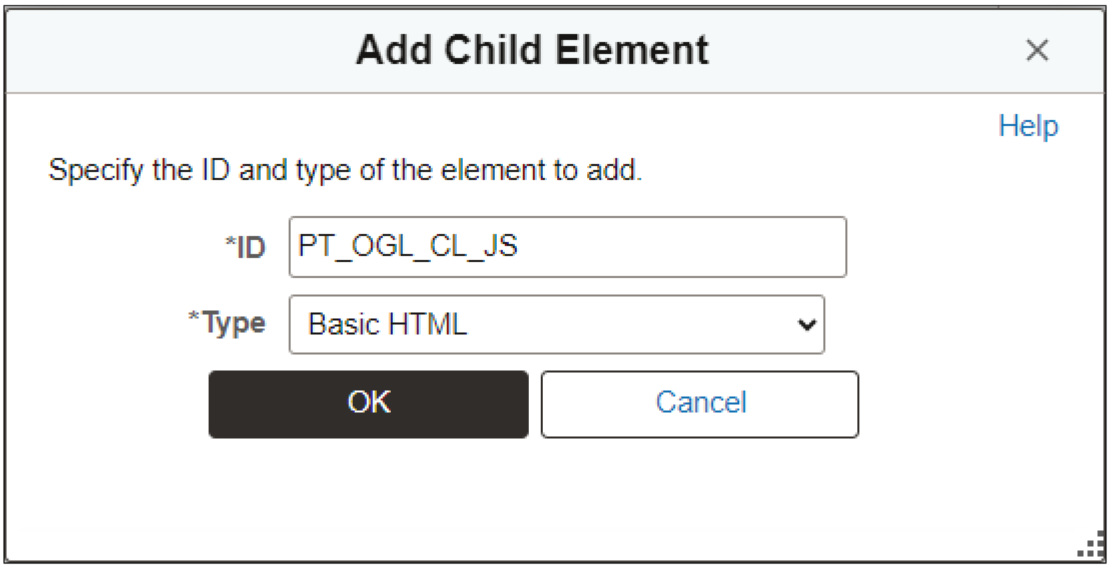
1. Go to PeopleTools, Portal, Branding, Branding Headers and Footers.

Note: Access to the Branding Headers and Footers component requires the Secure Branding Administrator role.

---

2. Open your custom header definition.
3. Add a new child element of type Basic HTML to the pthdr2container element—for example, named PT_OGL_CL_JS.

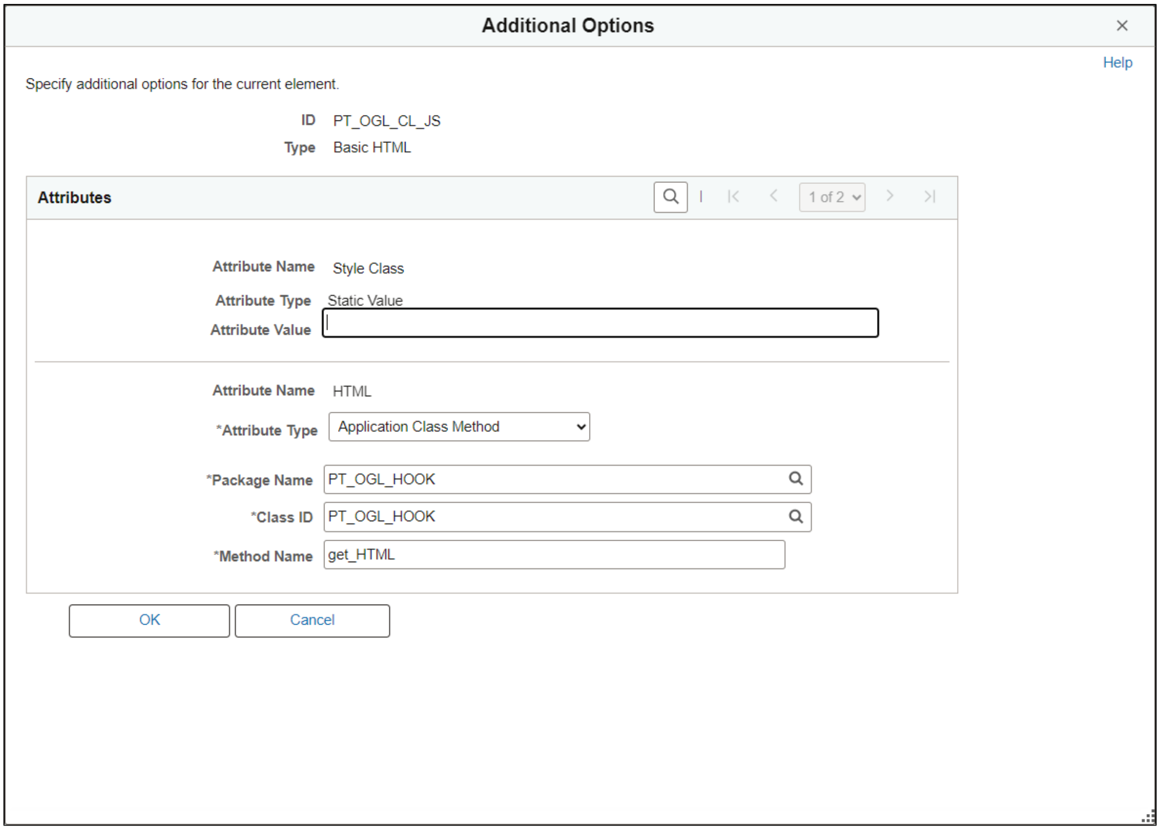
   In this example, the child element is named PT_OGL_CL_JS.



4. Click OK.
5. In the additional options for this HTML element, specify *Application Class Method* and then:

Field	***Value***
Package Name	PT_OGL_HOOK
Class ID	PT_OGL_HOOK
Method Name	get_HTML

In this example of the Additional Options page, the appropriate values have been specified for OGL configuration.



6. Save the changes to your custom header definition.
7. Sign out and sign back in for your changes to take effect. Then, complete the OGL configuration in the next topic.

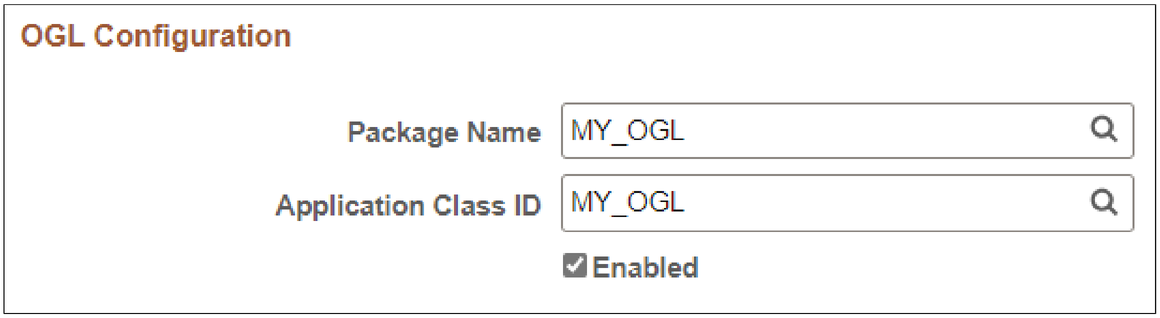
Completing the OGL Configuration

Use the OGL Configuration page to identify your custom OGL application class and to enable OGL system wide.

Navigation:

PeopleTools > **Portal** > **OGL Configuration**

This example illustrates the fields and controls on the OGL Configuration page.



Field or Control	Description
Package Name	Select the name of your OGL application package.
Application Class ID	Select the name of your OGL application class.
Enabled	Select this check box to enable OGL on this system. **Note:** To disable OGL, deselect this check box.

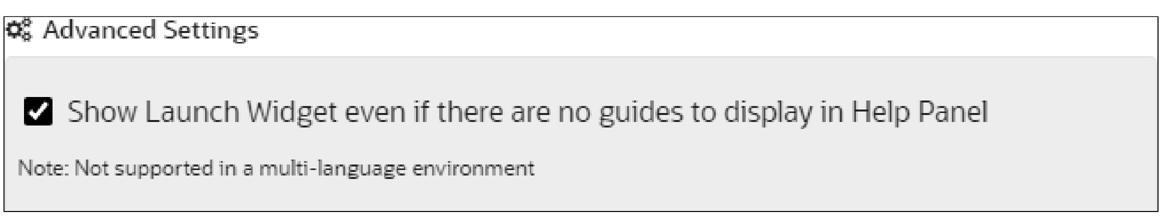
To enable OGL:

1. Specify your custom application package and class.
2. Select the Enabled check box.
3. Click Save.

   A message is displayed confirming that OGL has been enabled.
4. After you have configured and enabled OGL, test to ensure that is working on both classic and fluid pages:
   - Refresh the OGL configuration page to ensure that the OGL widget is now displayed on the page.
   - Navigate to a fluid page such as the homepage to ensure that the OGL widget is now displayed on that page.

Note: By default, the Help Center widget is not displayed when there are no guides to be displayed for the specified OGL appId. You can change this setting in the OGL Editor. Go to Settings, Help Panel.

OGL Editor help widget settings



For more information on moving the OGL widget, see Using OGL.

Adding OGL PeopleCode for Fluid Components with Search

Fluid components with a search page do not complete the PostBuild step of the component processor until after a search result is selected. As a consequence, the event that would trigger injecting the OGL JavaScript has not yet occurred; the OGL widget is not displayed on the search page. This issue occurs for fluid components that automatically insert the standard PTS_NUI_SEARCH page, components that are configured for configurable search and automatically insert the PTS_CFGSRCH_FL page, or those that contain a custom search page in the component definition.

To remedy this, PeopleCode event mapping can be used to manually configure the component with a SearchInit event that invokes the PT_OGL_HOOK PeopleCode.

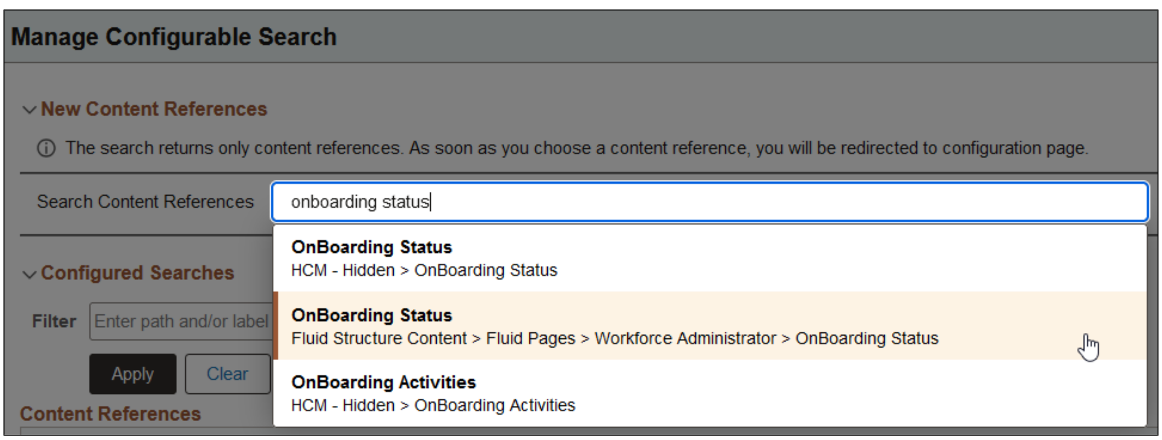
Determining the Component's Search Record

Use the Manage Configurable Search page to determine the component's active search record.

To use the Manage Configurable Search page to determine the active search record:

1. From the menu in the browser, select PeopleTools, Utilities, Manage Configurable Search.
2. In the Content Reference field, enter the content reference name of the component.

   In this example, *onboarding status* was entered.

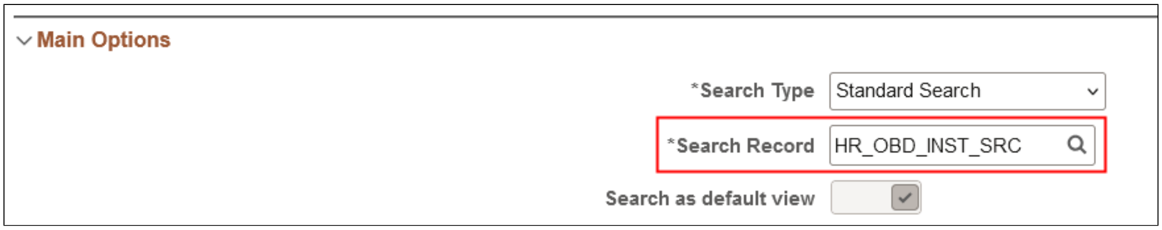
   

3. Select the content reference.

   In this example, the OnBoarding Status content reference definition in the Fluid Structure Content folder was selected.

4. Copy the ID for the search record from the Main Options section.

In this example, HR_OBD_INST_SRC was copied.



Important! Since you are merely obtaining the search record ID, do not save any changes on this page.

Adding Event Mapping to the Component

Use PeopleCode event mapping to manually configure the component with a SearchInit event that invokes the PT_OGL_HOOK PeopleCode.

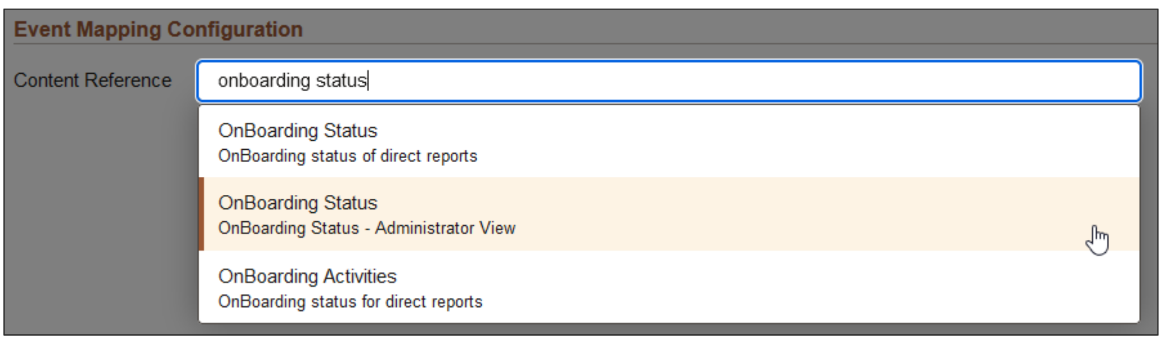
To add event mapping to the component:

1. From the menu in the browser, select PeopleTools, Portal, Event Mapping, Event Mapping Configuration.

2. Click the Add Configuration button.

3. In the Content Reference field, enter the content reference name of the component.

   In this example, *onboarding status* was entered.



4. Select the content reference. In this example, the second OnBoarding Status content reference definition was selected.

5. If you are upgrading OGL from a release prior to PeopleTools 8.62 and you previously configured SearchInit event for this component with the PT_OGL_RC service ID, delete that row from the grid before proceeding. Otherwise, skip to step 6.

6. In the Event Level field, select Record.

7. Select the Unrestricted Prompt check box.

   Important! You must select the unrestricted prompt so that you can find search records that are not defined at level 0 in the component definition.

8. In the Page or Record or Record Field field, enter and then select the exact name of the search record that you identified previously.

   In this example, HR_OBD_INST_SRC was selected.

9. In the Event Name field, select SearchInit.

10. In the Service ID/Application Package field, enter PT_OGL_HOOK_RC.

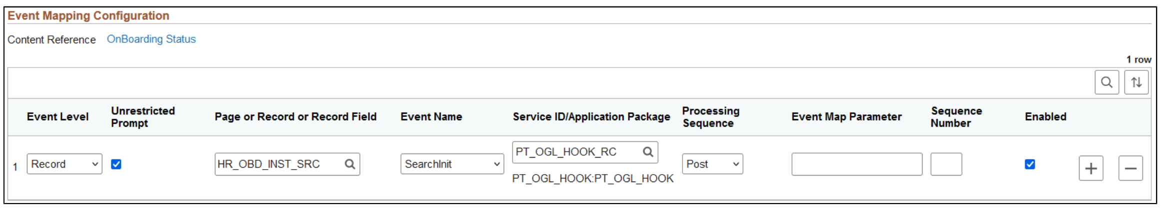
   Important! If you are upgrading OGL from a release prior to PeopleTools 8.62, PT_OGL_RC is also available. Do not select the service ID for prior releases.

   ---

11. In the Processing Sequence field, select Post.

12. Click Save to save the configuration.

   In this example, the SearchInit event has been configured with the PT_OGL_HOOK_RC service ID.



The configuration can be tested directly from this page by clicking the Content Reference link. Otherwise, navigate to the page in the normal manner via a tile or the menu.

Considerations When Upgrading OGL from a Previous Release

In PeopleTools 8.62 and later, new definitions are used to implement and configure OGL. You must disable any prior configuration before using these new definitions.

Disabling a Prior Configuration of OGL

To disable a prior configuration of OGL:

1. In Application Designer, run the PT_OGL_FL_AE Application Engine program to disable OGL on fluid pages. If output to log has been selected, review the log file for this message:

   ```
   Oracle Guided Learning has been successfully disabled. (0,0)
   ```

2. In the browser, go to PeopleTools, Portal, Branding, Branding System Options.

3. If the default branding theme is DEFAULT_THEME_FLUID_OGL, continue with step 3a. Otherwise if you have a custom branding theme and header, skip to step 4.

   a. Change the default theme to DEFAULT_THEME_FLUID.

   b. Check to see if you have any portal-specific overrides. Go to PeopleTools, Portal, Branding, Assign Branding Themes.

c. If you are presented with a standard PeopleSoft search page, no portal-specific overrides are in effect. Skip to step 3h.

d. Otherwise, modify each portal-specific override:

e. Change the default theme from DEFAULT_THEME_FLUID_OGL to DEFAULT_THEME_FLUID.

f. Review and update any user attribute-based theme assignments as appropriate.

g. Save your changes.

h. Sign out and sign back in for your changes to take effect. The prior configuration of OGL is now disabled for all classic pages in your system.

4. Go to PeopleTools, Portal, Branding, Branding Headers and Footers.

Note: Access to the Branding Headers and Footers component requires the Secure Branding Administrator role.

---

5. Open your custom header definition.

6. Modify the definition of the basic HTML element that you added to inject OGL JavaScript. For example, the name suggested in the PeopleSoft implementation guidelines is PT_OGL_CL_JS:

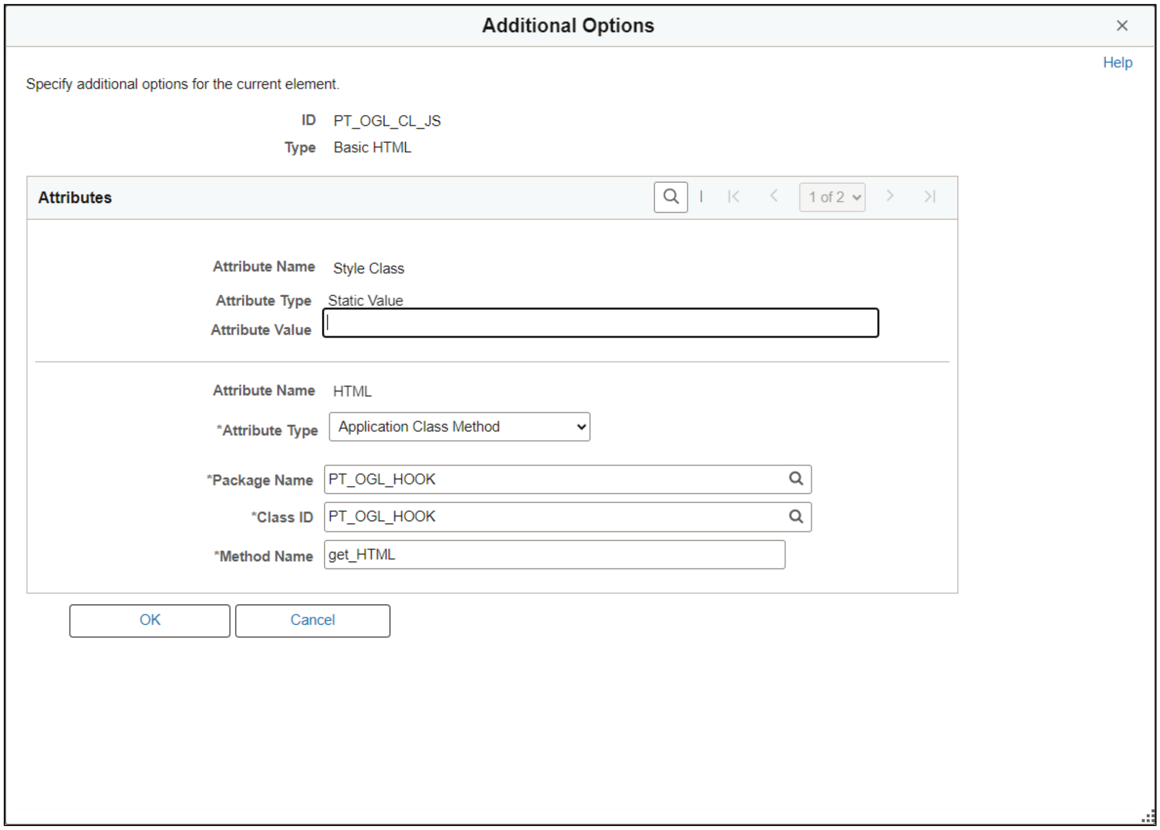
   a. Select the branding header element in the tree on the left—for example, select PT_OGL_CL_JS.

   b. Click the Additional Options link in the Element Properties pane on the right.

   c. Update the package name to PT_OGL_HOOK and the class ID to PT_OGL_HOOK.

   Note: The method name remains get_HTML.

In this example of the Additional Options page, the application class has been updated to PT_OGL_HOOK.



7. Save the changes to your custom header definition.

8. Sign out and sign back in for your changes to take effect. The prior configuration of OGL is now disabled for all classic pages in your system.

OGL PeopleCode for Fluid Components with Search

If you manually configured SearchInit events on fluid components with search, you will need to manually update those configurations to use the PT_OGL_HOOK_RC service ID. See Adding OGL PeopleCode for Fluid Components with Search.

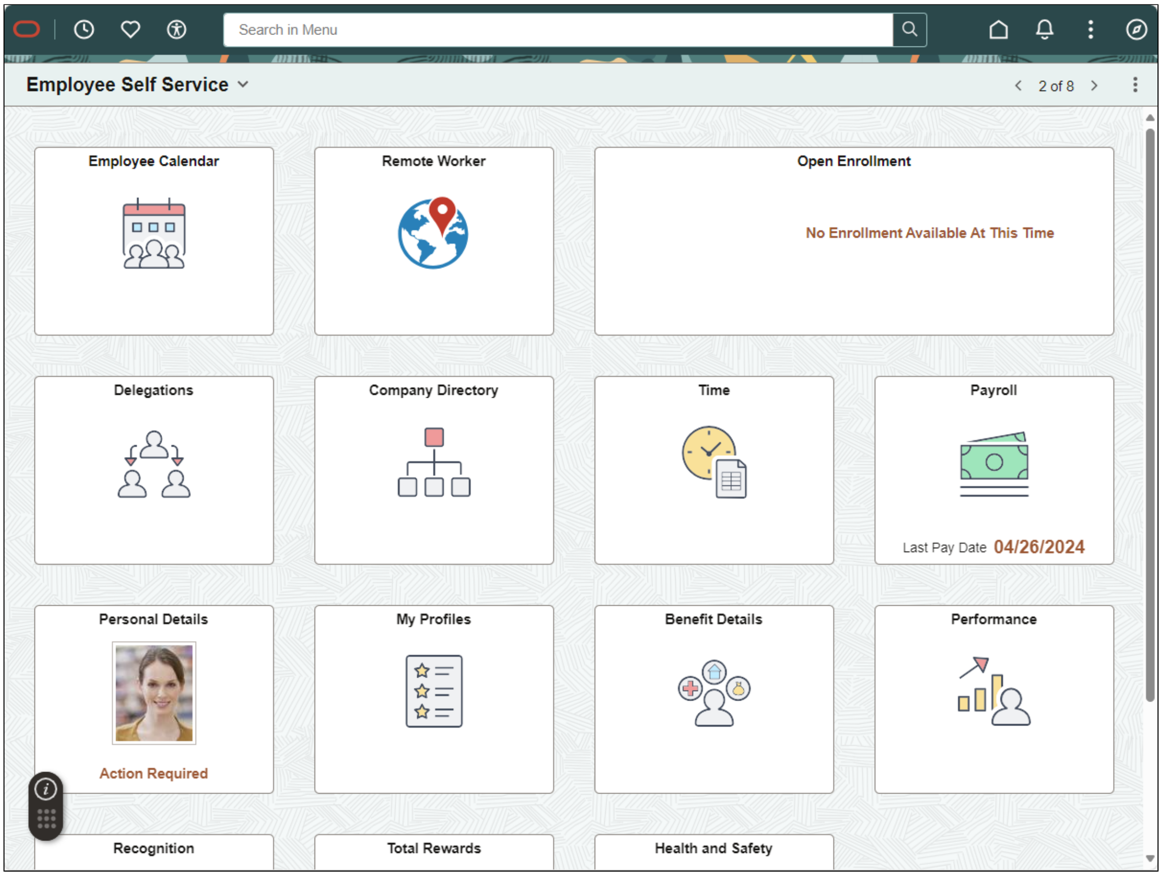
Using OGL

By default, the OGL Help Center widget is displayed on the right side of the page.

You can personalize the location of the OGL widget by dragging and dropping it.

Note: Do not move the OGL widget into the page header. If you move the OGL widget into the header of a classic page, it will become stuck there and it cannot be moved off the header. To remedy this issue, navigate to a fluid page where you will be free to move the OGL widget off the header.

In this example, the default location of the widget has been moved to the lower left corner of the page.



The guides available to you in the OGL Help Center widget are specific to your organization. Contact your OGL administrator for additional information on the guides available for your use.

Chapter 25

Understanding Changes in Portal Configuration Settings

Understanding Changes in Portal Configuration Settings

In the current release, PeopleSoft portal configuration has moved from a file-based environment to a PeopleSoft Pure Internet Architecture interface. This topic provides a complete listing that maps PeopleTools 8.43 and earlier portal settings to the current PeopleTools web profile settings. Within each of several broad functional categories, the release 8.43 properties are listed alphabetically, alongside the corresponding web profile fields that replace them.

The portal configuration files still exist in the current PeopleTools release, but they now contain only a small number of properties, and they're not meant to be edited. Don't modify any of the following files unless directed to do so by a PeopleSoft representative:

- Configuration.properties
- Pstools.properties
- Browserprops.xml
- Cookierules.xml

Related Links

Configuring Web Profiles

General Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - General page:

Release 8.43 Configuration Property	***Corresponding Web Profile General Field***
AuthTokenDomain	**Authentication Domain**
compressCacheFiles	**Compress Response References**
compressResponse	**Compress Responses**

Release 8.43 Configuration Property	*Corresponding Web Profile General Field*
compressMimeTypes	**Compress MIME Types** (compress Multipurpose Internet Mail Extensions types)
CompressReportOutput	**Compress Report Output** field in the **Reports** group box.
CompressReportOutputNetscape	**Compress Report Output** field in the **Reports** group box.
enableNewWindow	**Enable New Window**
enableProcessingWait	**Enable Processing Message**
enableReportRepository	**Enable Report Repository** field in the **Reports** group box.
helpUrl	**Help URL**
physicalpath	**Non-standard Base Path**
ReportRepositoryPath	**Report Repository Path** field in the **Reports** group box.
saveConfirmDisplayTime	**Save Confirmation Display Time**
singleThreadNS	**Single Thread Netscape**
threadDelay	**Single Thread Delay**

Related Links

Configuring General Portal Properties

Security Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Security page:

Release 8.43 Configuration Property	*Corresponding Web Profile Security Fields*
byPassSignOn	**Allow Public Access** field in the **Public Users** group box.
defaultPWD	**Password** field in the **Public Users** group box.

Release 8.43 Configuration Property	***Corresponding Web Profile Security Fields***
defaultUSERID	**User ID** field in the Public Users group box.
defaultXMLLinkUSERID	**User ID** field in the **XML Link** group box.
defaultXMLLinkPWD	**Password** field in the **XML Link** group box.
portalUseHttpForSameServer	**PIA use HTTP Same Server** (PeopleSoft Pure Internet Architecture use HTTP same server).
sessionTimeout	Authenticated Users - **Inactivity Logout** field in the **Authenticated Users** group box.
SSLRequired	**Secured Access Only** field in the **SSL** group box.
timeoutWarningScript	**Timeout Warning Script** and **Override** fields in the **Authenticated Usrs** group box.
userIDCookieAge	**Days to Autofill User ID**
UseSecureCookieWithSSL	**Secure Cookie with SSL** field in the **SSL** group box.
warningTimeout	**Inactivity Warning** field in the **Authenticated Users** group box.
XMLLinkUseHttpForSameServer	**XML Link Use HTTP Same Server** field in the **XML Link** group box.

You can now set these properties from the pstools.properties file by using the Web Profile Configuration - Security page:

Release 8.43 Pstools Property	***Corresponding Web Profile Security Field***
tuxedo_network_disconnect_timeout	**Disconnect Timeout** field in the Web Server Jolt Settings group box.
tuxedo_receive_timeout	**Receive Timeout** field in the Web Server Jolt Settings group box.
tuxedo_send_timeout	**Send Timeout** field in the Web Server Jolt Settings group box.

Related Links

Configuring Portal Security

Virtual Addressing Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Virtual Addressing page:

Release 8.43 Configuration Property	***Corresponding Web Profile Virtual Addressing Fields***
defaultPort	**Port** field in the **Default Addressing** group box.
defaultScheme	**Protocol** field in the Default Addressing group box.
pswebservername	**Name** field in the **Default Addressing** group box.
relativeURL	**Generate Relative URLs**
RPS	**Protocol, Host, HTTP,** and **HTTPS** fields in the **Reverse Proxy Server List** group box.

Related Links

Configuring Virtual Addressing

Cookie Rules

You can now set these elements and attributes from the cookierules.xml file by using the Web Profile Configuration - Cookie Rules page:

Release 8.43 Cookierules.xml Element or Attribute	***Corresponding Web Profile Cookie Rules Field***
block domain	**Cookies Not Passed to Server**
cookie name	**Cookie Pattern**
delete_on_logout	**Delete Cookie on Logout**
forward domain	**Cookies Passed to Server**
proxied	**Proxied**
secure	**Secure**

Related Links

Configuring Cookie Rules

Caching Settings

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Caching page:

Release 8.43 Configuration Property	***Corresponding Web Profile Caching Field***
CachePurgeAllHitCount	**Cache Purge All Hit Count** field in the **On the Web Server** group box.
CacheTargetContent	**Cache Target Content** field in the **On the Web Server** group box.
chartdirphys	**Chart Directory** field in the **On the Web Server - Directories** group box.
chartdirweb	**Chart Web Directory** field in the **On the Web Server - Directories** group box.
cssdirphys	**CSS Directory** (cascading style sheet directory) field in the **On the Web Server - Directories** group box.
cssdirweb	**CSS Web Directory** (cascading style sheet web director) field in the **On the Web Server - Directories** group box.
enableBrowserCache	**Cache Generated HTML** field in the **On the Browser** group box.
enableNoVersion	**Copy Image/CSS (No Versioning)** (copy image/cascading style sheet [no versioning]) field in the On the **Web Server - Directories** group box.
imagedirphys	**Image Directory** field in the **On the Web Server - Directories** group box.
imagedirweb	**Image Web Directory** field in the **On the Web Server - Directories** group box.
jsdirphys	**JavaScript Directory** field in the **On the Web Server - Directories** group box.
jsdirweb	**JavaScript Web Directory** field in the **On the Web Server - Directories** group box.

Release 8.43 Configuration Property	***Corresponding Web Profile Caching Field***
maxSavedState	**Number of States Supported** field in the **On the Browser** group box.
PortalCacheHomepageOnBrowser	**Cache Homepage** field in the On the Browser group box.
PortalCacheObjects	**Cache Portal Objects** field in the **On the Web Server** group box.
PortalCacheStaleInterval	**Cache Stale Interval** field in the **On the Web Server** group box.
PortalHomepageStaleInterval	**Homepage Stale Interval** field in the On the Browser group box.
portalUseCachedProxiedJS	**Cache Proxied JavaScripts** field in the **On the Web Server** group box.

You can now set these elements and attributes from the browserprops.xml file by using the Web Profile Configuration - Caching page:

Release 8.43 Browserprops.xml Element or Attribute	***Corresponding Web Profile Caching Field***
property name="CacheHomePage"	**Cache Home Page** field in the **Browsers** group box.
useragent id	**User Agent ID** field in the **Browsers** group box.

Related Links

Configuring Caching

Trace and Debug Properties

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Debugging page:

Release 8.43 Configuration Property	***Corresponding Web Profile Debugging Field***
connectionInformation	**Show Connection Information**
debug_inlinejavascript	**Show JavaScript Inline HTML**

Release 8.43 Configuration Property	***Corresponding Web Profile Debugging Field***
debug_inlinestylesheet	**Show StyleSheet Inline HTML**
debug_overlap	**Show Overlapping Fields**
debug_savefile	**Create File from PeopleSoft Pure Internet Architecture HTML Page**
debug_showlayout	**Show Layout**
enableDebugDumpFile	**Write Dump File**
enableTrace	**Show Trace Link at Signon**
testing	**Generate HTML for Testing**

Related Links

Configuring Trace and Debug Options

Look and Feel Settings

You can now set these properties from the configuration.properties file by using the Web Profile Configuration - Look and Feel page:

8.43 Configuration Property	***Corresponding Web Profile Look and Feel Fields***
authtokenenabled_page	**Auth Token Enable Page** field in the **Other Pages** group box.
chgPwdOnExpire	**Change Password On Expire** and **Override** fields in the **Password** group box.
chgPwdOnWarn	**Change Password On Warning** and **Override** fields in the **Password** group box.
cookiesrequired_page	**Cookies Required Page** field in the **Other Pages** group box.
exception_page	**Exception Page** field in the Error Pages group box.
expire_page	**Page** field in the **Expire Page** group box.
expirePage_ContentName	**Content Name** field in the Expire Page group box.

8.43 Configuration Property	*Corresponding Web Profile Look and Feel Fields*
logout_page	**Logout Page** field in the **Signon/Logout Pages** group box.
mcfAuthFailure_page	**MCF Auth Failure Page** field in the **Error Pages** group box.
passwordexpired_page	**Password Expired Page** field in the **Password** group box.
passwordwarning_page	**Password Warning Page** field in the **Password** group box.
signon_page	**Signon Page** field in the **Signon/Logout Pages** group box.
signonError_page	**Signon Error Page** field in the **Signon/Logout Pages** group box.
signonresultdoc_page	**Signon Result Doc Page** field in the **Signon/Logout Pages** group box.
signontrace_page	**Enable Trace Page** field in the **Other Pages** group box.
sslrequired_page	**SSL Required Page** field in the **Other Pages** group box.
start_page	**Page** field in the Start Page group box.
startPageScript	**Script** and **Override** fields in the **Start Page** group box.
userprofile_page	**User Profile Page** field in the **Other Pages** group box.

You now specify the locale properties from the pstools.properties file as default language mappings on the Manage Installed Languages page. You can specify additional mappings from International Organization for Standardization (ISO) locale codes to PeopleSoft language codes by using the Web Profile Configuration - Look and Feel page.

Related Links

Configuring Trace and Debug Options

“Adding New Language Codes to the System” (Global Technology)

Chapter 26

Configuring the Application for 8.4x Navigation

Understanding Configuration of PeopleSoft Applications for 8.4x Navigation

The 8.5x navigation is based on iframe templates and features drop-down navigation. The 8.4x navigation is based on frame (HTML) templates and uses menu pagelet navigation. These topics describe the two tasks that you must complete to configure an 8.5x application for 8.4x navigation.

These topics describe how to:

- Reconfigure the default template.
- Reconfigure the navigation pagelet.

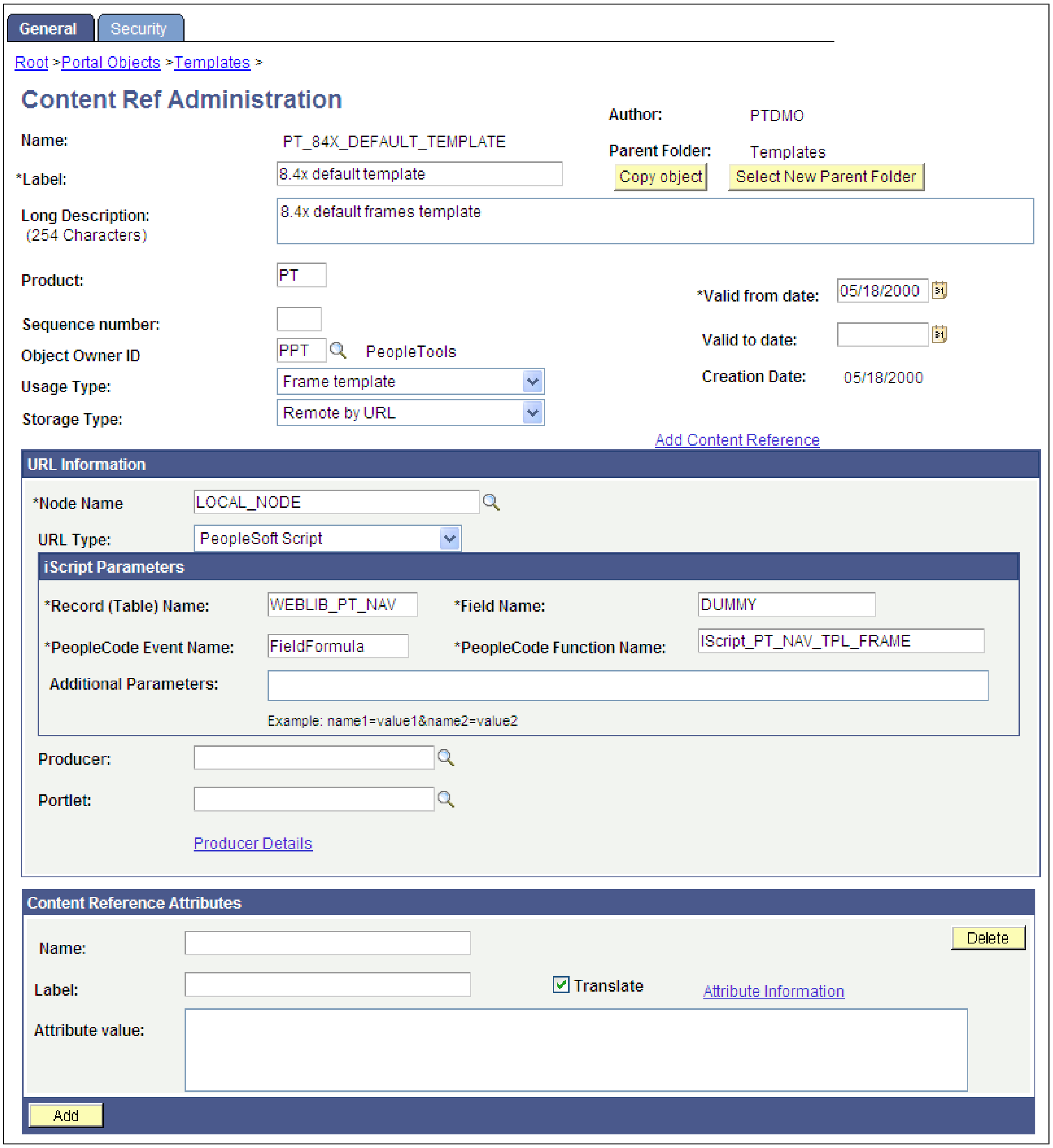
Important! HTML templates are desupported as of PeopleTools 8.50. Many features introduced or enhanced after release 8.50 require iframe templates to function as designed. Features that are not supported in HTML templates include: modal windows and prompts; drop-down navigation; related content; WorkCenter pages; and SmartNavigation.

Reconfiguring the Default Template

To reconfigure the default template:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**
2. In the **Folders** group box, click **Portal Objects,** and then click **Templates.**
3. Click the **Edit** link for the 8.4x default template.
4. In the **Field Name** field, delete the existing value and enter *DUMMY.*
5. Save the component.

After completing the preceding reconfiguration steps, the 8.4X DEFAULT_ TEMPLATE template will appear similar to the following:



6. In the navigation path, click the **Templates** link.

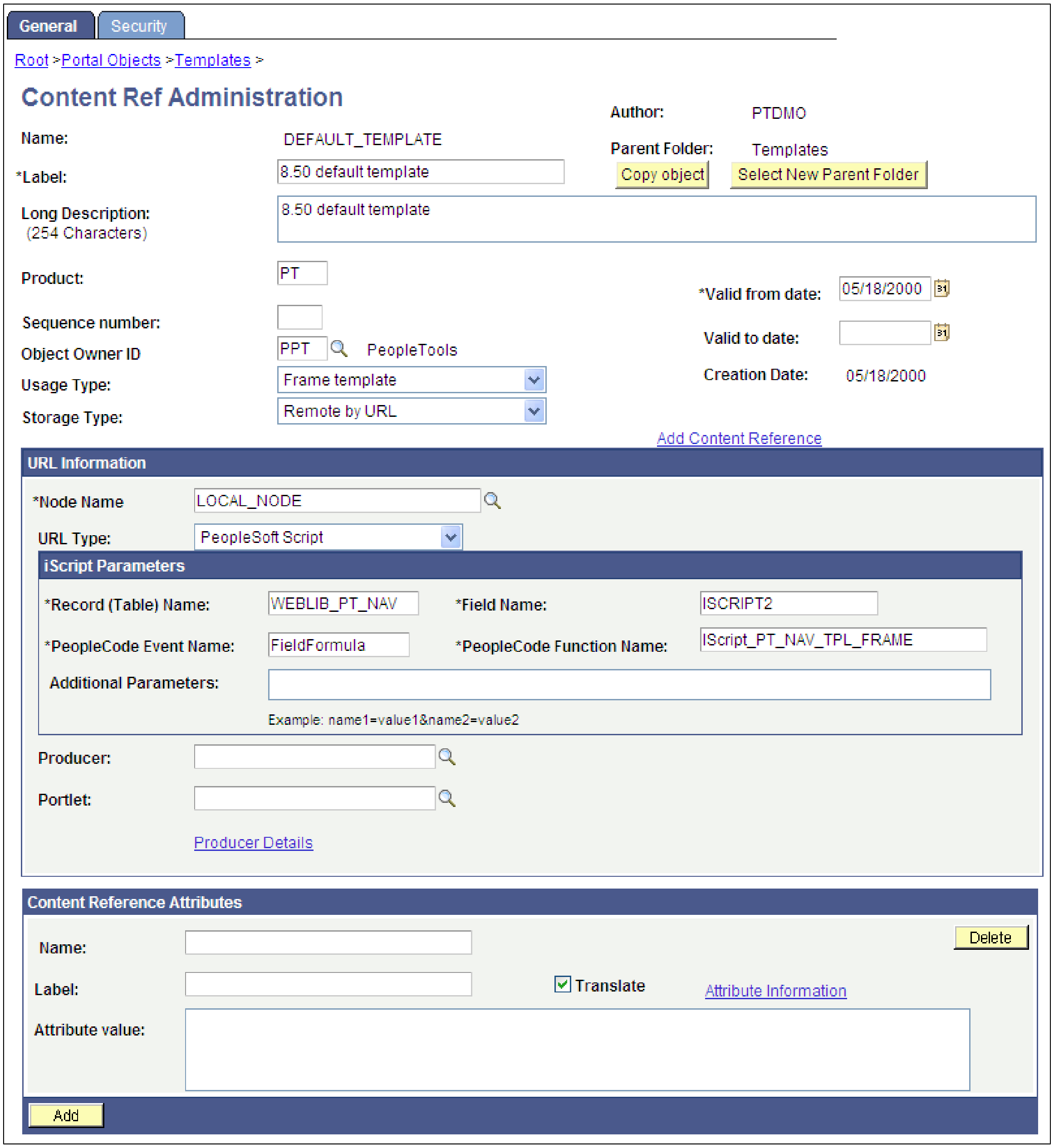
7. Click the **Edit** link for the 8.50 default template.

8. In the **Field Name** field, replace *ISCRIPT1* with *ISCRIPT2*.

9. Change the template Usage Type from *Inline frame template* to *Frame template*.

10. Save the component.

After completing the preceding reconfiguration steps, the 8.50 DEFAULT_ TEMPLATE template will appear similar to the following:



Reconfiguring the Navigation Pagelet

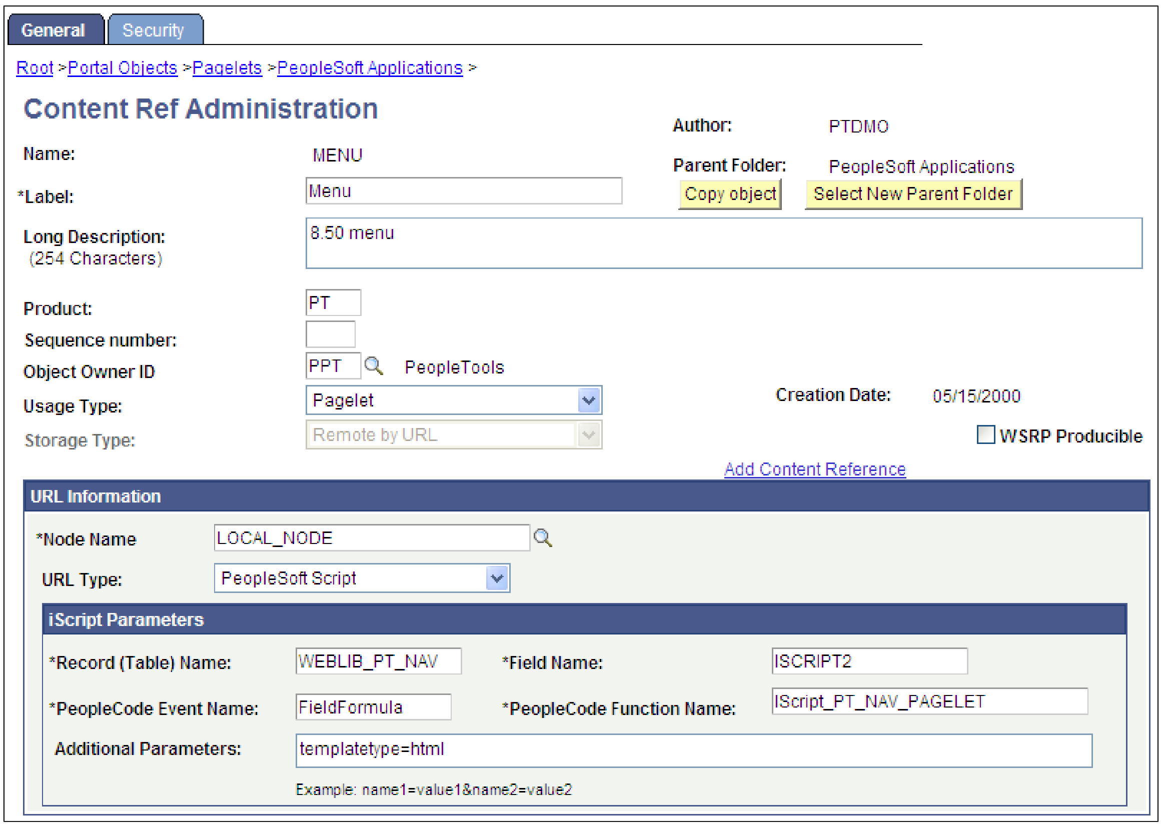
Important! Menu pagelets and other forms of classic navigation are no longer supported. The information in this topic pertains to a prior version of PeopleSoft classic applications only. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see “Working with Fluid Pages and Controls” (Applications User’s Guide).

To reconfigure the Menu navigation pagelet:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**
2. In the **Folders** group box, click **Portal Objects,** and then click **Pagelets.**
3. Click the **PeopleSoft Applications** link, and then click the **Edit** link for the Menu.
4. In the **Field Name** field, delete the existing value and enter *ISCRIPT2.*
5. Save the component.

   The content reference will look like this:

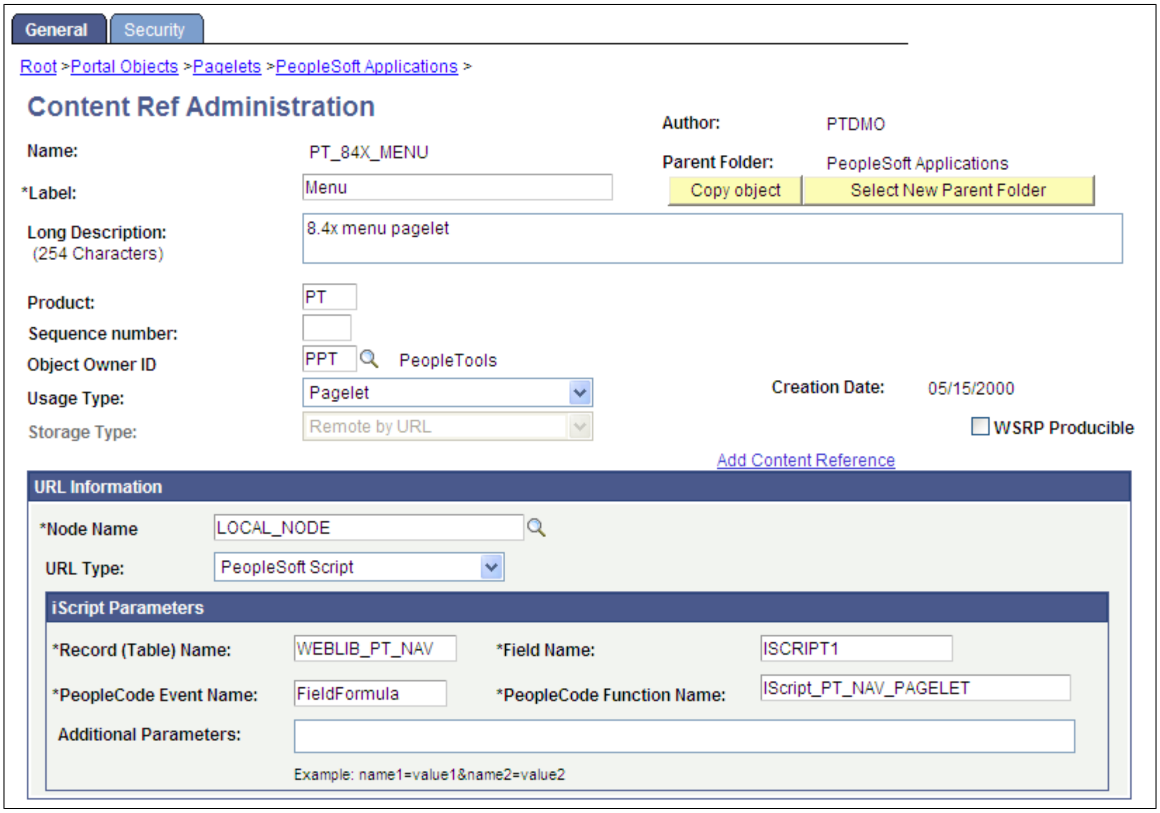
   After completing the preceding reconfiguration steps, the Menu navigation pagelet will appear similar to the following:



6. In the breadcrumbs, click the **PeopleSoft Applications** link.
7. Click the **Edit** link for Menu - Classic.
8. In the **Label** field, replace *Menu - Classic* with *Menu.*
9. In the **Field Name** field, replace *ISCRIPT2* with *ISCRIPT1.*
10. Save the component.

    The content reference will look like this:

After completing the preceding reconfiguration steps, the Menu - Classic navigation pagelet will appear similar to the following:



11. Stop and restart the application and web servers.

Chapter 27

Converting Frames-Based Templates to iFrame-Based Templates

Understanding Conversion of Frames-Based Templates to iFrame-Based Templates

This topic provides a detailed example of converting an existing frames-based template (eProcurement Shopping Cart template) to the 8.5x drop-down navigation template.

Converting a frames-based template to an iframe-based template enables you to take advantage of the new features in the iframe, drop-down navigation template: related content; AJAX-based, drop-down navigation; modal, key-specific favorites improvements; and the recently used menu functionality.

Important! The following conversion example assumes that the current frames-based template you are converting uses a storage type of *Local in HTML (catalog).* If your frames template is a *Remote by URL* template, your steps will be slightly different.

Converting the eProcurement Shopping Cart Template

This section describes how to:

- Gather information about the current template.
- Create the iframe template iScript.
- Create the iframe template HTML definition.
- Set security for the template iScript.
- Convert the template content reference.
- Troubleshoot errors.

Gathering Information About the Current Template

To gather information about the current template:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**
2. Within Structure and Content, click **Portal Objects,** then click **Templates.**
3. Click the **Edit** link next to the ePro Shopping Cart Template.

4. Copy the text in the **HTML Area** field. Save it to a text file. After you have changed the template type, this HTML will be removed.

5. Click the Security tab and note the current security for this content reference.

   This step is important if specific permissions or roles have been set. Later, you apply these permissions to the new iScript that will be created.

Creating the iframe Template iScript

To create the iframe template iScript:

1. Open the WEBLIB_PV.ISCRIPT.FieldFormula PeopleCode in edit mode.

2. At the bottom of the PeopleCode, enter the following code:

```
Declare Function ptIframeTemplateHNav PeopleCode WEBLIB_PT_NAV.ISCRIPT1 FieldF⇒

ormula;

/*
  * builds the inline frame template with the drop-down navigation
*/
Function IScript_PVShopCart_Template

   /* parameter 1: HTML object name that will be used to generate the inline f⇒

rame template */
       %Response.Write(ptIframeTemplateHNav("PV_SHOP_CART_HNAV_TEMPLATE"));

End-Function;
```

Note: For *Remote by URL* templates, use the existing web library iScript and just modify the PeopleCode as shown previously.

Creating the iframe Template HTML Definition

To create the iframe template HTML definition:

1. In Application Designer, open the PT_HNAV_TEMPLATE HTML definition.

2. Clone the definition and save it as the value that was used in parameter 1 of the new iScript.

3. Open the text file that contains the original template HTML and locate the pagelets that you need to move to the new template, for example:

```
<frame name="CART" id="CART" frameborder="no" scrolling="auto" noresize src=""⇒

 style="border-top:ridge #CCCCCC 5px;">
     <IClientComponent Name="Links">
         <Source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT">dummy</Source>
     </IClientComponent>
</frame>
```

4. Change the HTML so that it looks like this:

```
<div id="pvcartpagelet">
     <iframe name="CART" id="CART" frameborder="no" src="">
          <pagelet name="Links">
                <source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT"/>
```



```
          </pagelet>
     </iframe>
 </div>
```

Note: The iframe template does not support the `IClientComponent` template meta-tag. These meta-tags are desupported and are replaced by the `pagelet` meta-tag.

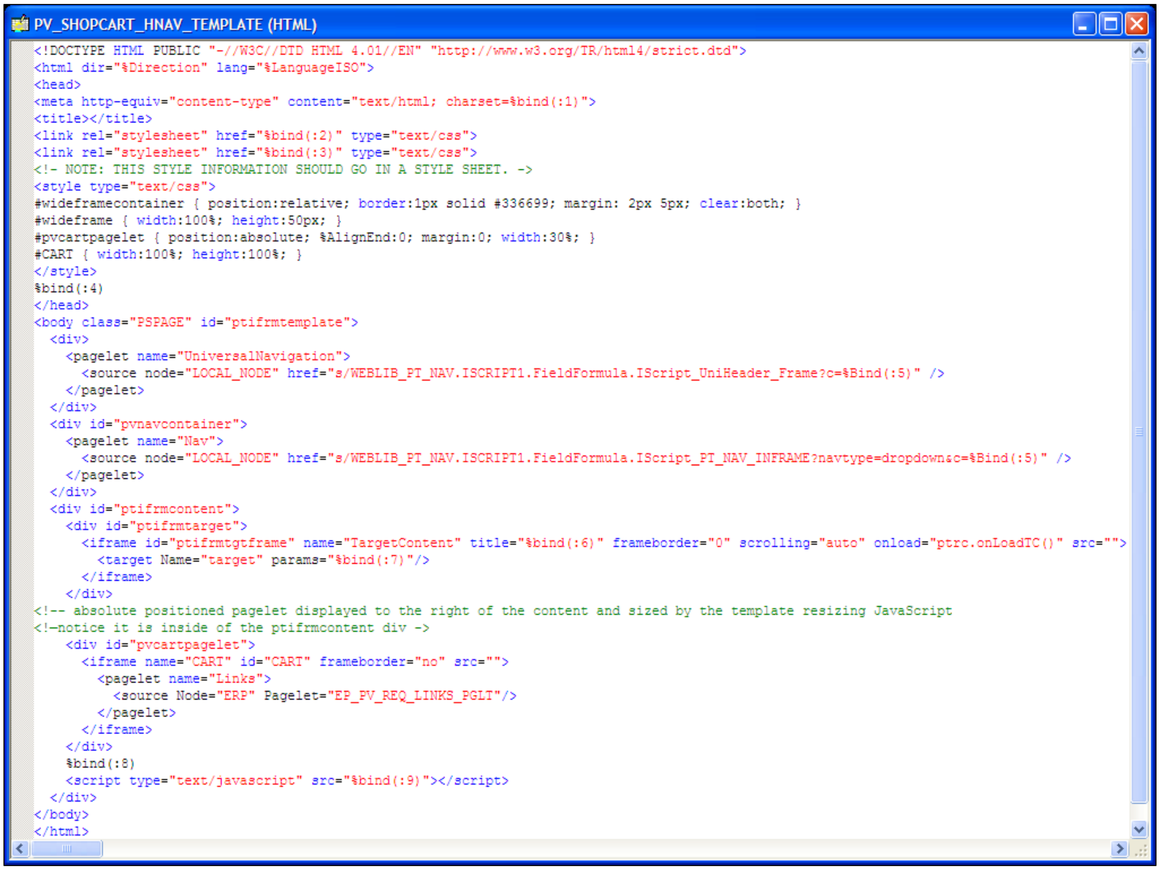
5. Locate where you want to position the Links pagelet in the PV_SHOP_CART_HNAV_TEMPLATE iframe template. If the pagelet is to exist to the right or left of the target content, place the pagelet inside of the iframe content container `<div>`. In this scenario, the pagelet will be located to the right of the target content.

```
                  <div id="ptifrmcontent">
  <div id="ptifrmtarget">
    <iframe id="ptifrmtgtframe" name="TargetContent" title="%bind(:6)" framebo⇒

rder="0" scrolling="auto" onload="ptrc.onLoadTC()" src="">
      <target Name="target" params="%bind(:7)"/>
    </iframe>
  </div><div id="pvcartpagelet">
    <iframe name="CART" id="CART" frameborder="no" src="">
      <pagelet name="Links">
        <source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT"/>
      </pagelet>
    </iframe>
  </div>
  %bind(:8)
  <script type="text/javascript" src="%bind(:9)"></script>
</div>
```

6. Add the style information for the new pagelet. Ideally, these styles should go into a style sheet to take advantage of style sheet caching on the web server and browser. For this example, the embedded style approach places the style in the head section of the template.

```
<style type="text/css">
/* style for the CART pagelet container div */
#pvcartpagelet {
  position:absolute; /* required: the position must be fixed so the iframe can⇒

 be sized properly */

  %AlignEnd:0;       /* meta CSS to position the location of the pagelet */
  margin:0;
  width:30%;         /* required:must be a percentage,value so the correct wid⇒

th is rendered when the browser is resized */
}

/* style for the CART iframe */
#CART { width:100%; height:100%; }
</style>
```


This is an example of the full HTML definition that contains the template with the shopping cart pagelet.

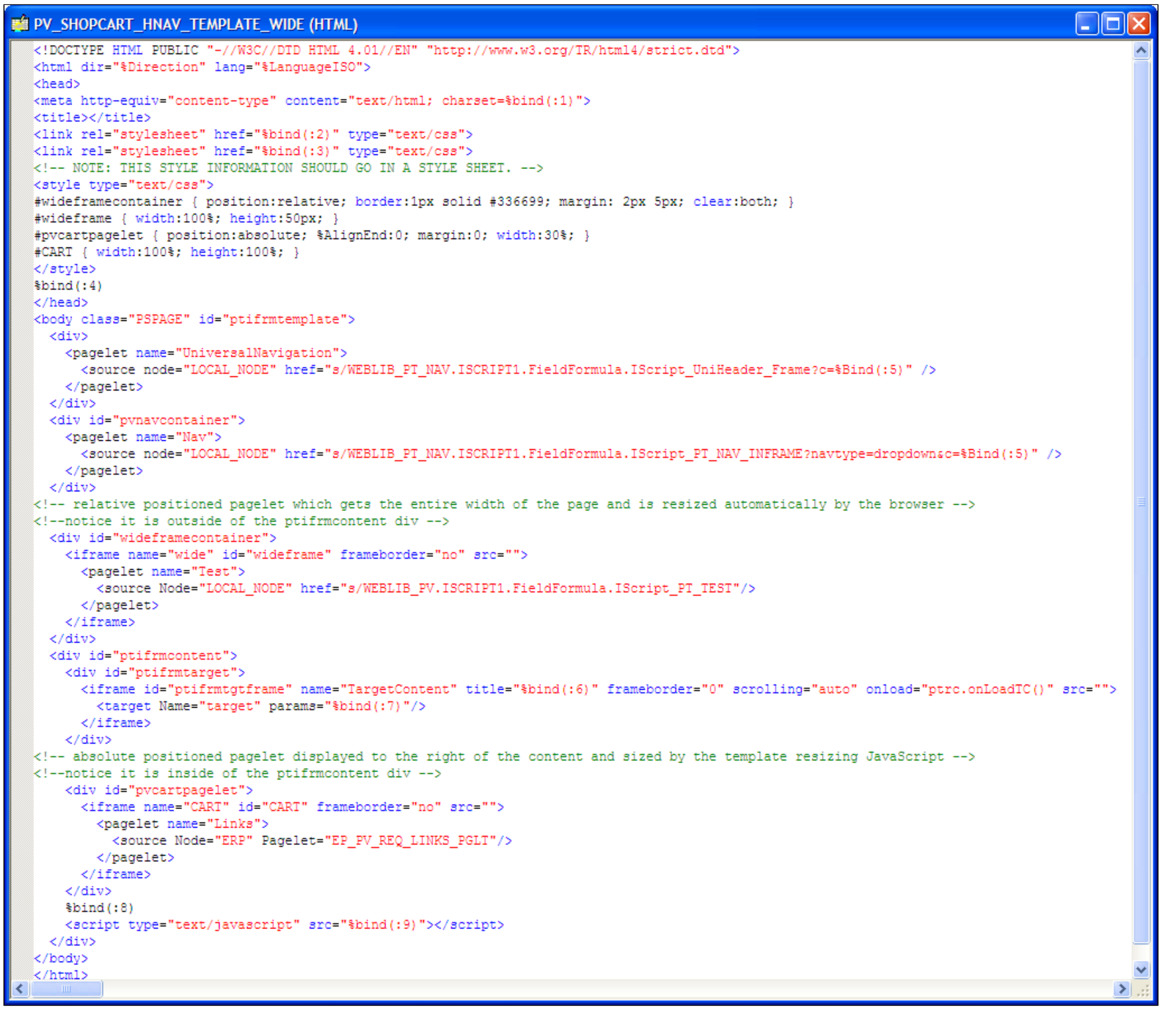

PV_SHOPCART_HNAV_TEMPLATE (HTML)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html dir="%Direction" lang="%LanguageISO">
<head>
<meta http-equiv="content-type" content="text/html; charset=%bind(:1)">
<title></title>
<link rel="stylesheet" href="%bind(:2)" type="text/css">
<link rel="stylesheet" href="%bind(:3)" type="text/css">
<!- NOTE: THIS STYLE INFORMATION SHOULD GO IN A STYLE SHEET. ->
<style type="text/css">
#wideframecontainer { position:relative; border:1px solid #336699; margin: 2px 5px; clear:both; }
#wideframe { width:100%; height:50px; }
#pvcartpagelet { position:absolute; %AlignEnd:0; margin:0; width:30%; }
#CART { width:100%; height:100%; }
</style>
%bind(:4)
</head>
<body class="PSPAGE" id="ptifrmtemplate">
  <div>
    <pagelet name="UniversalNavigation">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_UniHeader_Frame?c=%Bind(:5)" />
    </pagelet>
  </div>
  <div id="pvnavcontainer">
    <pagelet name="Nav">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_PT_NAV_INFRAME?navtype=dropdown&c=%Bind(:5)" />
    </pagelet>
  </div>
  <div id="ptifrmcontent">
    <div id="ptifrmtarget">
      <iframe id="ptifrmtgtframe" name="TargetContent" title="%bind(:6)" frameborder="0" scrolling="auto" onload="ptrc.onLoadTC()" src="">
        <target Name="target" params="%bind(:7)"/>
      </iframe>
    </div>
<!-- absolute positioned pagelet displayed to the right of the content and sized by the template resizing JavaScript
<!—notice it is inside of the ptifrmcontent div ->
    <div id="pvcartpagelet">
      <iframe name="CART" id="CART" frameborder="no" src="">
        <pagelet name="Links">
          <source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT"/>
        </pagelet>
      </iframe>
    </div>
    %bind(:8)
    <script type="text/javascript" src="%bind(:9)"></script>
  </div>
</body>
</html>
```

If a pagelet needs to expand to the width of the page, the following HTML and style information can be used:

```
<style type="text/css">
#wideframecontainer {
  position:relative;                         /* required: */
  border:1px solid #336699;
  margin: 2px 5px;
  clear:both;
 }
#wideframe { width:100%; height:50px; }
</style>
...
<!--must be located outside of the content container div-->
<div id="wideframecontainer">
    <iframe name="wide" id="wideframe" frameborder="no" src="">
      <pagelet name="Test">
        <source Node="LOCAL_NODE" href="s/WEBLIB_PV.ISCRIPT1.FieldFormula.IScr⇒

ipt_PT_TEST"/>
      </pagelet>
    </iframe>
 </div>
```

This is an example of a full HTML definition that contains a wide pagelet:



PV_SHOPCART_HNAV_TEMPLATE_WIDE (HTML)

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html dir="%Direction" lang="%LanguageISO">
<head>
<meta http-equiv="content-type" content="text/html; charset=%bind(:1)">
<title></title>
<link rel="stylesheet" href="%bind(:2)" type="text/css">
<link rel="stylesheet" href="%bind(:3)" type="text/css">
<!-- NOTE: THIS STYLE INFORMATION SHOULD GO IN A STYLE SHEET. -->
<style type="text/css">
#wideframecontainer { position:relative; border:1px solid #336699; margin: 2px 5px; clear:both; }
#wideframe { width:100%; height:50px; }
#pvcartpagelet { position:absolute; %AlignEnd:0; margin:0; width:30%; }
#CART { width:100%; height:100%; }
</style>
%bind(:4)
</head>
<body class="PSPAGE" id="ptifrmtemplate">
  <div>
    <pagelet name="UniversalNavigation">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_UniHeader_Frame?c=%Bind(:5)" />
    </pagelet>
  </div>
  <div id="pvnavcontainer">
    <pagelet name="Nav">
      <source node="LOCAL_NODE" href="s/WEBLIB_PT_NAV.ISCRIPT1.FieldFormula.IScript_PT_NAV_INFRAME?navtype=dropdown&c=%Bind(:5)" />
    </pagelet>
  </div>
<!-- relative positioned pagelet which gets the entire width of the page and is resized automatically by the browser -->
<!--notice it is outside of the ptifrmcontent div -->
  <div id="wideframecontainer">
    <iframe name="wide" id="wideframe" frameborder="no" src="">
      <pagelet name="Test">
        <source Node="LOCAL_NODE" href="s/WEBLIB_PV.ISCRIPT1.FieldFormula.IScript_PT_TEST"/>
      </pagelet>
    </iframe>
  </div>
  <div id="ptifrmcontent">
    <div id="ptifrmtarget">
      <iframe id="ptifrmtgtframe" name="TargetContent" title="%bind(:6)" frameborder="0" scrolling="auto" onload="ptrc.onLoadTC()" src="">
        <target Name="target" params="%bind(:7)"/>
      </iframe>
    </div>
<!-- absolute positioned pagelet displayed to the right of the content and sized by the template resizing JavaScript -->
<!--notice it is inside of the ptifrmcontent div -->
    <div id="pvcartpagelet">
      <iframe name="CART" id="CART" frameborder="no" src="">
        <pagelet name="Links">
          <source Node="ERP" Pagelet="EP_PV_REQ_LINKS_PGLT"/>
        </pagelet>
      </iframe>
    </div>
    %bind(:8)
    <script type="text/javascript" src="%bind(:9)"></script>
  </div>
</body>
</html>
```

Setting Security for the Template iScript

To set security for the template iScript:

1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists.**

2. Enter the permission list that you noted when you gathered information about the template earlier.

3. Enable Web Library security for the permission list.

4. Continue until all permission lists and roles are added.

Note: If the template content reference security is public, only one permission list is needed.

Note: For *Remote by URL* templates, this step is optional because the iScript already exists.

Converting the Template Content Reference

To convert the template content reference:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**
2. Within Structure and Content, click **Portal Objects,** then click **Templates.**
3. Click the **Edit** link next to the ePro Shopping Cart Template.
4. Change the usage type to *inline frame template.*
5. Change the storage type to *Remote By URL.*
6. Enter the iScript record, field, and PeopleCode function names.
7. Click the **Save** button.
8. Stop and restart the application and web servers.
9. Test the template.

Note: For *Remote by URL* templates, omit steps 5, 6, and 7.

Troubleshooting Errors

This table describes some common errors and possible causes:

Error	***Solution***
JavaScript errors	Check existing pagelet JavaScript. Make sure to correct statements like this: `parent.parent.frames["UniversalHeader"] /* no longer valid in an iframe template */ parent.frames[3]. /* use a iframe name or an iframe id */`

Chapter 28

Related Content Reserved Words and Service Parameters

Reserved Words Default Mapping

When you have a service using any of the following reserved words as service parameters, they will have some default mapping done when you first access the Map Fields page. Here is the list of related content-specific reserved words, their service parameters, and the default mapping data.

This table lists the reserved words and their default mapping:

Reserved Word	***Description***	***Default Map Type***
PTCS_CREFNAME	Current content reference name for which the service assignment and mapping is being done.	Fixed value.
PTCS_CREFLABEL	Current related content transaction content reference label.	Fixed value.
PTCS_NODENAME	%Node.	System variable.
PTCS_PORTALNAME	%Portal.	System variable.
PTCS_KEYFIELD1	Component level 0 key1 if available.	Key field.
PTCS_KEYFIELD2	Component level 0 key2 if available.	Key field.
PTCS_KEYFIELD3	Component level 0 key3 if available.	Key field.
PTCS_KEYFIELD4	Component level 0 key4 if available.	Key field.
PTCS_PAGENAME	%Page.	System variable.
PTCS_SERVICEID	Current related content service ID.	Fixed value.
PTCS_RELURL	Relative URL of the content reference.	Fixed value.

Chapter 29

Configuring UPK Topics as Related Content Services

Understanding Oracle UPK and Related Content

You can configure Oracle User Productivity Kit (UPK) modules and topics as component-level and page-level related content services. These sections provide a brief overview of Oracle UPK and related content and describe how to:

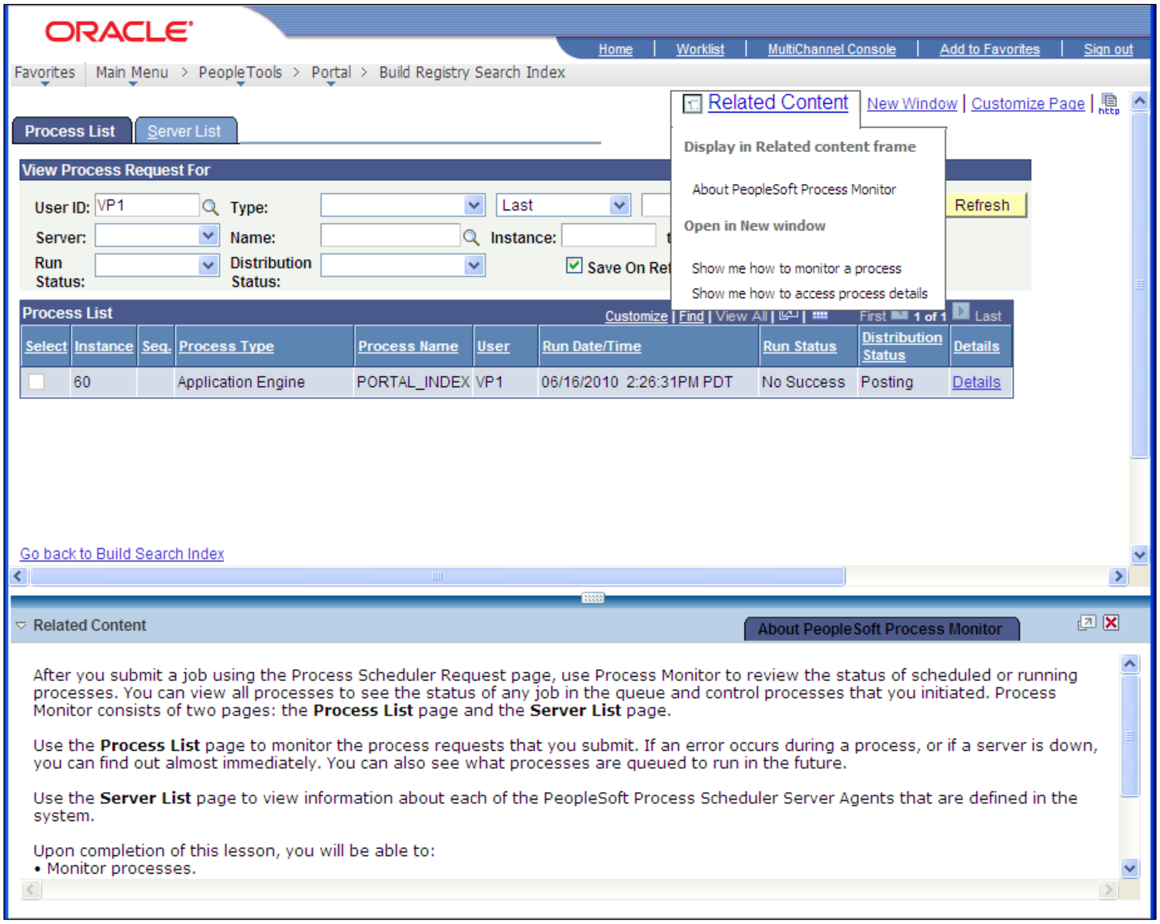
- Create UPK-based related content service definitions.
- Assign UPK-based services as related content.

Note: In these sections, the word *topic* is used to represent both UPK topics and UPK modules unless specified otherwise.

Oracle UPK Developer and content is an end-user training solution that provides a single-source development and customization tool to enable rapid content creation and customization. UPK also can provide system process documentation, web-based training materials, role-based performance support, and other types of documentation.

PeopleSoft Related Content Framework is a set of PeopleTools definitions and processes that enable developers and subject matter experts to link application pages with contextually relevant content. The primary goal of related content services is to provide immediate access to relevant, contextual information for any PeopleSoft transaction without requiring any additional user effort. PeopleSoft Related Content Framework acts as an enterprise mashup, tying together all types of content such as Oracle Business Intelligence Enterprise Edition (OBIEE) analytics, queries, discussion threads, documentation, and relevant non-PeopleSoft data into a single tool—PeopleSoft application pages.

You can use the related content framework to set up access to UPK modules and topics so that any UPK content is available at the page or component levels under the Related Content menu as shown here:



Prerequisites

This table lists the necessary prerequisites for integrating UPK and related content and provides references to information about these prerequisites:

Prerequisite	***Reference***
Perform all related content configurations.	See Understanding Oracle UPK and Related Content.
Create, deploy, and configure UPK player package to your web server.	See Oracle's *PeopleSoft Enterprise Help Menu Integration* See Oracle's User Productivity Kit *In-Application Support Guide*

UPK Player Package Links

The UPK administrator will have access to the player package link (URL). The link should look similar to this:

http://*upkservername*/upk/*content_subject*/toc.html

If you change the player package link ending from *toc.html* to *kp.html* you can access the kp.html file.

This is a sample kp.html page:

PeopleTools 8.52 - Process Scheduler [Concept]

Understanding Process Scheduler [Concept]

- Understanding PeopleSoft Process Scheduler [Try It][Concept]
- Understanding PeopleSoft Process Scheduler Architecture [Try It][Concept]
- Understanding PeopleSoft Process Scheduler Components [Try It][Concept]
- Understanding the Activities to Implement Process Scheduler [Try It][Concept]

Submitting Process Request [Concept]

- Submitting Process Requests [See It][Try It][Know It][Do It][Concept]
- Changing Distribution Options [See It][Try It][Know It][Do It][Concept]
- Submitting a Process from an Application [See It][Try It][Know It][Do It][Concept]
- Scheduling Process Requests [See It][Try It][Know It][Do It][Concept]

Using Process Monitor [Concept]

- Monitoring Processes [See It][Try It][Know It][Do It][Concept]
- Viewing Process Details [See It][Try It][Know It][Do It][Concept]
- Viewing Report Details [See It][Try It][Know It][Do It][Concept]
- Viewing PeopleSoft Process Scheduler Server Agents [See It][Try It][Know It][Do It][Concept]

KP.HTML File Topic Modes

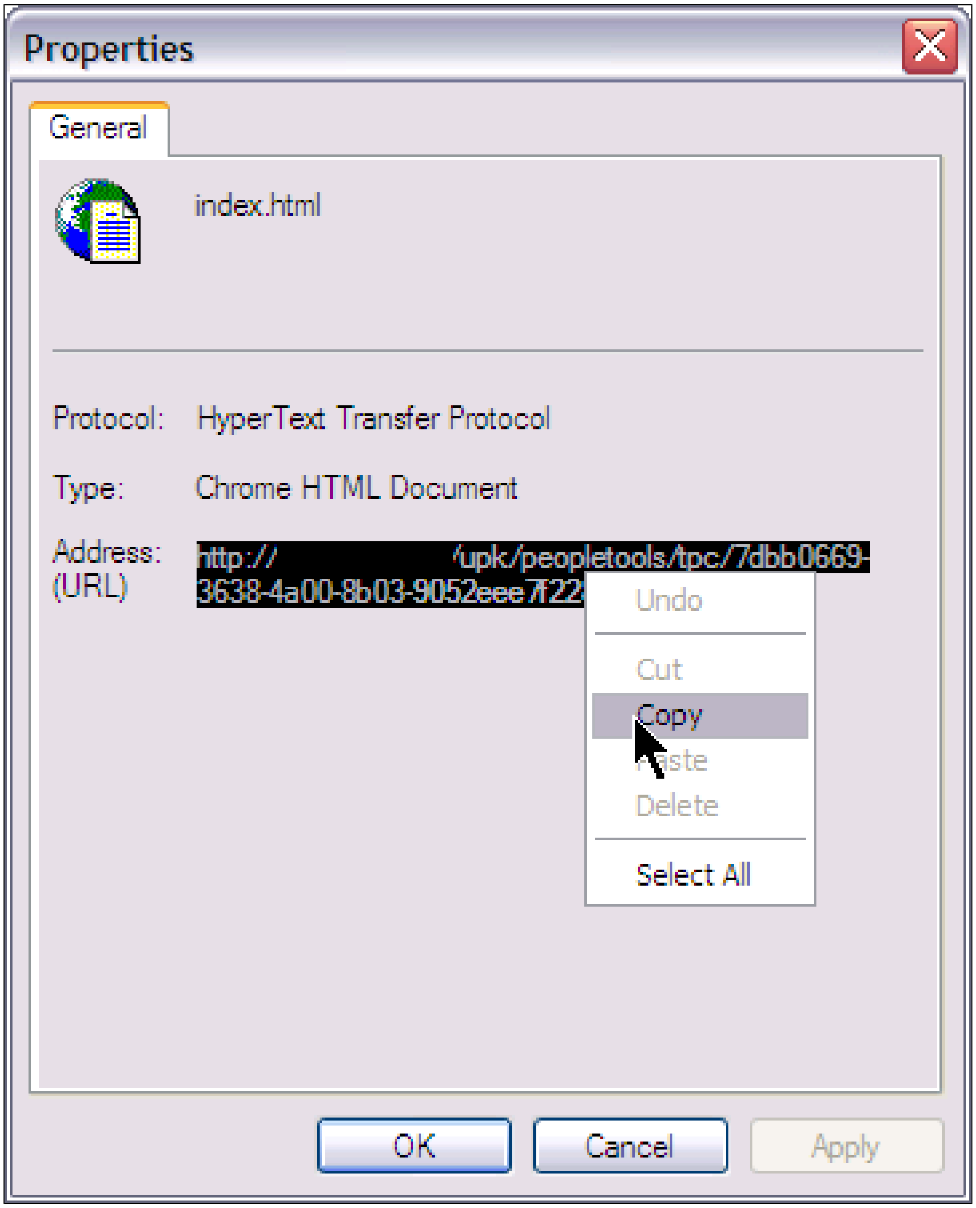
Each UPK installation includes a kp.html file that lists your UPK modules and topics. The kp.html file, which is located in the player package folder, enables you to manually link to a specific mode for a topic.

UPK modules can appear in five modes:

- Do It
- Know It
- Try It
- See It
- Concept

Right-click the topic link and select **Properties** to access the topic-specific URL. If you are creating task-oriented related content, right-click the **Do It** link. If you are creating concept-oriented related content, right-click the **Concept** link.

The Properties dialog box shows the URL in the Address field. Although the fields on the dialog box are display-only, you can highlight and copy the address as this example shows:



Note these points about the format of the URLs:

- For task-oriented information, the URL should look similar to this:

  http://upkserver/upk/peopletools/dhtml_kp.html?Guid=69f0376e-70fc-417c-a43c-ecd6afb72b29&*Mode=D*&Back

- For conceptual information, the URL should look similar to this:

  http://upkserver/upk/peopletools/tpc/0350d9a9-6013-4cc9-9022-016f815029ba/Parts/*index.html*

Use this table to verify that the link accesses the correct mode:

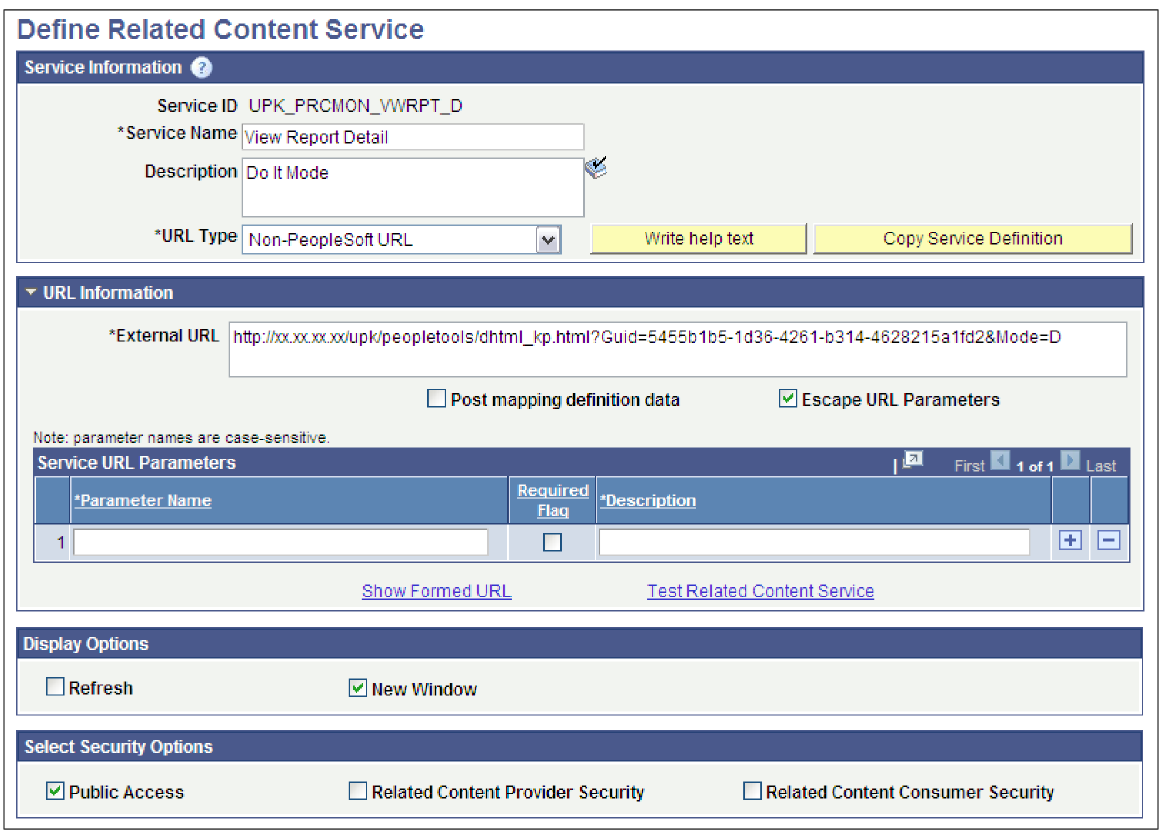
Link Fragment	***Mode***
Mode=D	Do It
Mode=K	Know It
Mode=T	Try It
Mode=S	See It
index.html	Concept

For each URL, the `&Back` fragment is extraneous for related content purposes. You delete this part of the URL when you create the service definition.

Service Definitions

You configure UPK topics as non-PeopleSoft URL service definitions.

This example illustrates the related content service definition for the View Report Detail UPK topic, which has been configured as a non-PeopleSoft URL type:



Related Content Menus and Frames

When you assign UPK services to content references, you assign the services as component-level related content only for these reasons:

- UPK service definitions do not require any parameters; therefore topics cannot be mapped at the field level.
- Users should be able to access UPK content from the **Related Content** drop-down menu. Only services assigned as related content appear in the Related Content drop-down menu.

Menu Item Labels

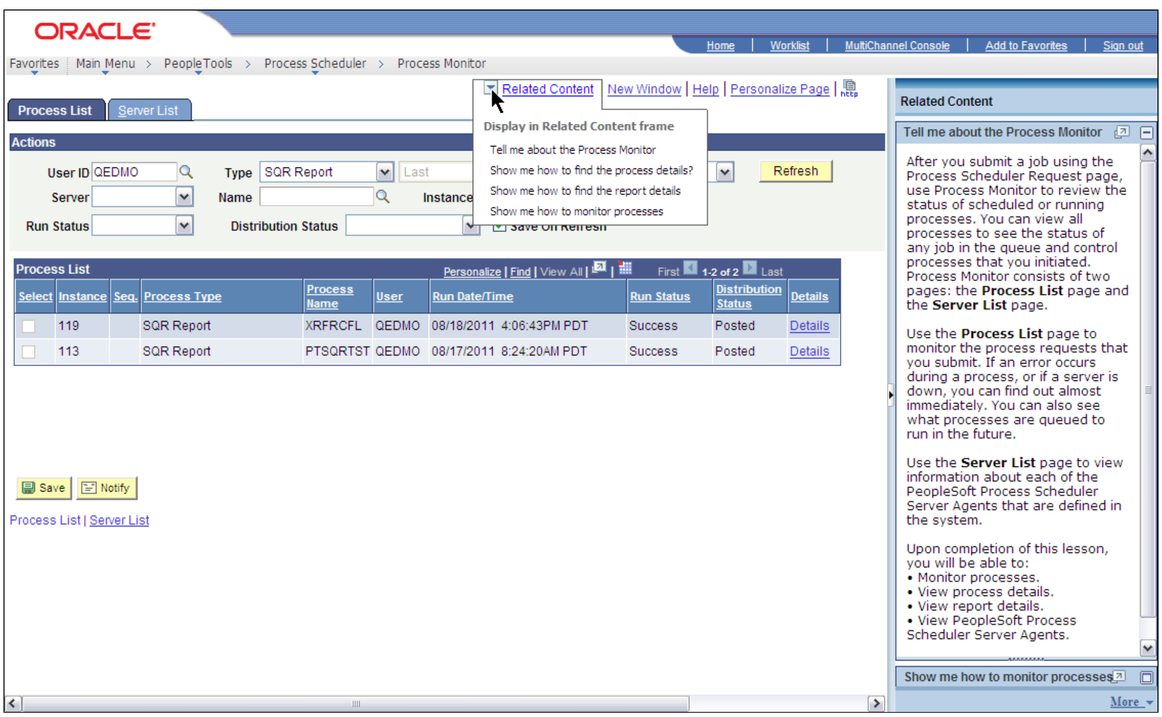
When you create a label for the service, consider the specific context of the service and use direct language. For example, if you want to add a service in which the UPK topic shows users how to submit a process in Do It mode, then *Show me how to submit a process* is an appropriate label.

Related Content Frames

If you assign the UPK topics to appear in the right frame, users can access multiple help topics simultaneously because they appear in individual pagelets. If you assign UPK topics to the bottom frame, one topic only is visible at a time because they appear in tabs.

Example UPK Topic in Related Content Frame

In this example, the UPK topic appears in the right related content frame:



Related Links

Understanding the PeopleSoft Related Content Framework

Configuring Related Content, Related Actions, and Menu and Frame Layouts

Creating UPK-Based Related Content Service Definitions

This section discusses how to:

1. Modify the UPK player package link.
2. Create the UPK-based service definition

Modifying the UPK Player Package Link

To modify the UPK player package link:

1. Contact your UPK system administrator to obtain the player package link (URL).
2. Change the file name *toc.html* to *kp.html* so that the URL looks similar to this:

   http://*upkservername*/upk/*content_subject*/kp.html
3. Paste the link into your browser and access the file.
4. Find the topic. Right-click the **Concept** or **Do It** link associated with the topic. If you are creating task-oriented related content, right-click the **Do It** link; otherwise, click the **Concept** link.
5. From the pop-up menu, select **Copy Shortcut.**

   Alternately, select **Properties** and then copy the URL that you see on the dialog box.
6. If you are creating multiple service definitions, you might consider creating a spreadsheet that contains information similar to that which exists in this table:

Topic	***URL***
Process Scheduler - Concept	http://*upkserver*/upk/peopletools/tpc/0350d9a9-6013-4cc9-9022-016f815029ba/Parts/index.html
Viewing Process Details - Do It	http://*upkserver*/upk/peopletools/dhtml_kp.html?Guid=69f0376e-70fc-417c-a43c-ecd6afb72b29&Mode=D&Back

7. For each task-oriented URL, remove the *&Back* at the end of the address.
8. Save your spreadsheet.

Creating the UPK-Based Service Definition

To create the UPK-based service definition:

1. Create a new related-content service definition.

Choose a naming convention that makes UPK service definitions easily identifiable, for example *UPK_*<service_definition_name>

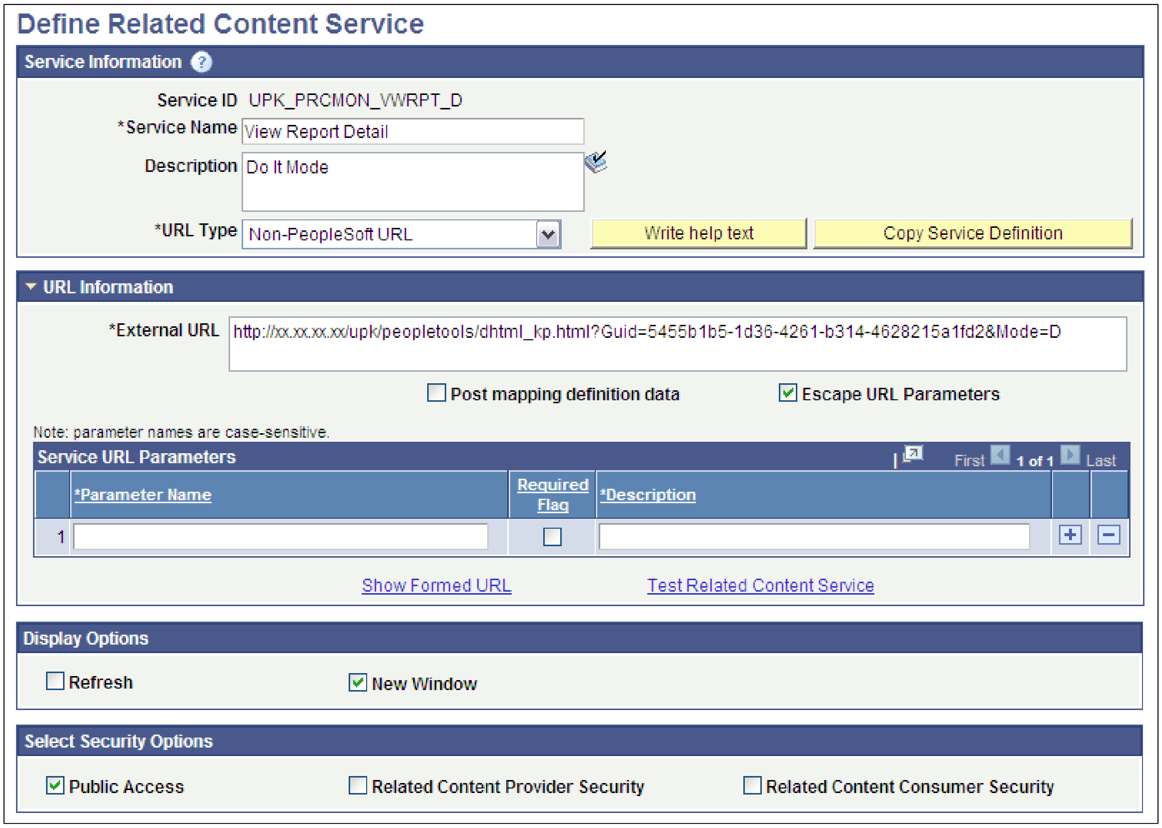
2. Enter the service name and description.
3. Select *Non-PeopleSoft URL* as the URL type.
4. In the **External URL** field, enter the URL to your kp.html file.

   It should look similar to this: *http://myupksite.com/upk/mycontent/dhtml_kp.html?Guid=5cd5ba9f-c3a8-441f-923b-a8aca7e95b6e&Mode=D*
5. Click the **Test Related Content Service** link.

   The **Test Related Content Service** page opens.
6. On the Test Related Content Service page, click the **Test Related Content Service** link.
7. Click the **Return** button to go back to the Define Related Content Service page.
8. For conceptual information, select the **New Window** check box to give the administrator the ability to enable the new window functionality for the user.
9. In the **Select security options** group box, select the **Public Access** check box.
10. Click the **Save** button.

Note: UPK-based service definitions do not require additional parameters.

This example illustrates how a completed service definition shows an external URL that accesses UPK content in Do It mode:



Note: You can use the **Copy Service Definition** button to clone this definition and save time creating new UPK-based service definitions.

Related Links

Creating Non-PeopleSoft URL Service Definitions

Assigning UPK-Based Services as Related Content

Access the Manage Related Content Configuration page. (Select **PeopleTools** > **Portal** > **Related Content Service** > **Manage Related Content Service**.)

Use the Manage Related Content Configuration page to assign UPK-based services as related content:

1. Click the **Assign Related Content to Application Pages** link.
2. Browse the menu tree and select the content reference.
3. On the Related Content tab, add the UPK service as component-level related content.

   See Assigning Related Content to Classic Components and Pages.
4. Configure either:
   - The related content menu. See Configuring Related Content Menus.
   - The related content frame. See Configuring the Related Content Frame.
5. Click the **Save** button.
6. Click the content reference link to test your configuration.

Chapter 30

Configuring OBIEE-Based Related Content Services

OBIEE Configuration Prerequisites

You can configure related content services to display Oracle Business Intelligence Enterprise Edition (OBIEE) analytics reports. The OBIEE servlet must be installed and OBIEE must be configured to enable OBIEE-based related content. In addition to configuring the analytics servlet, you must complete certain prerequisites before you assign OBIEE-based related content services to application pages.

When using OBIEE analytics as related content services, note the following configuration requirements:

- The OBIEE database needs to be on release 10.1.3.3 or later.
- The PeopleSoft application database and the OBIEE database must be configured with the same users to access the reports.
- XML needs to be generated for the OBIEE report definition that you make available as related content. To automatically generate the XML, navigate to the advanced tab on the OBIEE report definition and save the report.
- The sizing of reports needs to be handled on the OBIEE database. The associated grid, width of the page, and centering of the objects cannot be adjusted.
- Set up the Integration Broker Gateway on the PeopleSoft database.
- Encrypt the *secureFileKeystorePasswd.*
    1. Select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Gateway Setup Properties** link.
    2. Enter *administrator* in the **User ID** field and *password* in the **Password** field and click the **OK** button.
    3. Click the **Advanced Properties Page** link and search for *secureFileKeystorePasswd.*
    4. Copy the password and encrypt it using the Password Encryption Utility at the bottom of the page.
    5. Replace the existing plain text password with the encrypted value.
    6. Click the **OK** button.
    7. Save the page and click the **OK** button.
- Test the deployment by entering this URL in your browser: *http://<webservername:port>/analytics*.

- After installing analytics.war, update the OBIEE server setup with the weblogic URI: http://<webservername:port>/analytics and reload the report definitions.

Chapter 31

Configuring PeopleSoft Applications to Use Oracle WebCenter Related Content Services

Understanding the Oracle WebCenter Integration

Oracle WebCenter is a collaborative portal environment that provides an infrastructure for enterprise-wide collaboration. Oracle WebCenter Spaces and Oracle WebCenter Services provide a comprehensive set of standards-based components that enrich your PeopleSoft applications with the most complete and open set of Enterprise 2.0 capabilities in the industry.

Oracle WebCenter Spaces allows users to work more effectively with project teams and work groups, including teams that span multiple geographies and include external members. Within the context of a space, Oracle WebCenter Services provide the capability to include wikis, blogs, communications, content collaboration and social networks. PeopleTools, by leveraging the PeopleSoft Related Content Framework, enables you to integrate WebCenter group spaces and many of the accompanying features with PeopleSoft transaction pages. The following WebCenter related content services are standard with your PeopleTools installation and are available for use with any PeopleSoft transaction page:

- Blogs
- Discussions
- Documents
- Links
- Wikis

Note: As of PeopleTools 8.54, integration with Oracle WebCenter through the delivered related content services has been desupported. These service definitions may be removed from a future release of PeopleTools.

Understanding WebCenter Spaces

Within any company, large groups or departments often must create smaller groups to focus on a particular goal, project, or even topic. To help these groups organize themselves, Oracle WebCenter uses the concept of spaces as the nexus for bringing content together. Spaces provide a meaningful way to manage all project details involving any group of users. The WebCenter Spaces application creates spaces to help PeopleSoft application users share information and interact in a collaborative setting.

Templates

Oracle WebCenter Spaces application provides templates for building spaces that support different types of endeavors. When a user accesses a transaction page for the first time, WebCenter creates a new group space for the page or its transaction instance. The template specifications determine the services that appear by default in the new group space. You can install and use the PeopleSoft template that your PeopleTools installation includes, you can use an out-of-the-box WebCenter template, or create and install your own custom template using an existing group space as the basis for the template.

When you configure the WebCenter options in your PeopleSoft applications, PeopleTools requires that you specify the default template for use with new PeopleSoft transaction group spaces. PeopleTools uses web services to retrieve group space template information from the WebCenter server, and then display the group space services in the bottom related content frame.

Understanding WebCenter Services

Note: As of PeopleTools 8.54, integration with Oracle WebCenter through the delivered related content services has been desupported. These service definitions may be removed from a future release of PeopleTools.

Oracle WebCenter exposes social networking and shared productivity features through a comprehensive set of services. The services are integrated so that you can use them together. For example, you can add a discussion forum to enable shared discussion, upload documents for shared review, and a blog to keep your personal notes regarding a particular PeopleSoft transaction. You can associate these Oracle WebCenter services with PeopleSoft application pages through the Manage Related Content component:

- Oracle WebCenter Blogs service
- Oracle WebCenter Discussions service
- Oracle WebCenter Documents service
- Oracle WebCenter Links service
- Oracle WebCenter Wikis service

While the related content service definitions are provided in PeopleTools, an active Oracle WebCenter database is required to provide the services and to store the related data.

Important! With the exception of the links service, each WebCenter service uses a set of corresponding WebCenter pages for managing the elements of the group space. The WebCenter application, not PeopleTools, controls the behavior of these pages.

See *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter,* Getting WebCenter Portal Applications Up and Running.

Oracle WebCenter Blogs Service

The Blogs service enables users to create instance specific or shared WebCenter blogs. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Blog posts to the group space become shared blogs of that transaction page, which means the blogs appear for all transaction instances of the transaction page. When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Blog posts of the group space appear when the user selects the specific transaction instance on the transaction page.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor privilege to the group spaces. If you specify a default moderator role on the WebCenter Options page, a moderator is also added.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance-level data security.

Oracle WebCenter Discussion Service

The Discussions service lets you expose discussion forums on your application pages, so users can create forums, post questions, and search for answers. For example, customers can share product reviews, or a customer service department can answer questions online. Discussion forums additionally provide the means to preserve and revisit discussions.

The Discussion service enables users to create instance specific or shared WebCenter discussions. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Discussions in the group space become shared discussions of that transaction page, which means the discussions appear for all transaction instances of the transaction page. When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Discussions posts of the group space appear when the user selects the specific transaction instance on the transaction page.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor privilege to the group spaces. If you specify a default moderator role on the WebCenter Options page, a moderator is also added.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance-level data security.

Oracle WebCenter Documents Service

The Documents service provides features for accessing, adding, and managing folders and files; configuring file and folder properties; and searching file and folder content in the connected content repositories. When you set up a content repository connection to integrate the Documents service, you can add content to the transaction pages of the PeopleSoft application.

The Documents service enables users to create instance specific or shared WebCenter discussions. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Documents in the group space become shared documents of that transaction page, which means the documents appear for all transaction instances of the transaction page.

When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Documents in of the group space appear when the user selects the specific transaction instance on the transaction page.

The Documents service uses corresponding WebCenter pages for actions such as, uploading new documents, listing and searching documents in the group space, and managing documents. The WebCenter application, not PeopleTools, controls the behavior of these pages.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor privilege to the group spaces. WebCenter also adds a moderator user group member, if you specify a default moderator role on the WebCenter Options page.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance-level data security.

Oracle WebCenter Links Service

The Links service provides a way to view, access, and associate related information. For example, in a list of project assignments, you can link to the specifications relevant to each assignment. In a discussion thread about a problem with a particular task, you can link to a document that provides a detailed description of how to perform that task.

Unlike other WebCenter related content services, the Links service is a PeopleSoft component-based page. The Links service enables users to search and create links to existing WebCenter group spaces. The service also enables users to add links to external web pages. When you click the links that appear on the service page, the target opens in a new browser window.

The Links service enables users to create instance-specific or shared WebCenter links to resources.

This service supports Related Content Security only. When the transaction page or instance includes other WebCenter related content services such as the WebCenter Document service or the WebCenter Discussion service, the group spaces created for those services appear as related links by default; you cannot remove these links. Conversely, the service does not alter user access to a linked service. When one user does not have access to a group space that is the target of a link added by other users, the group space link does not appear to the unauthorized user.

Note: For this service, related content security governs whether links appear. The service does not control the security of the target content.

Oracle WebCenter Wiki Service

A wiki is a collection of useful content or information that users can browse, update, and remove, sometimes without the need for registration. This ease of interaction and the variety of operations make wiki an effective tool for collaborative authoring, where multiple people create written content together.

The wiki service enables users to create instance specific or shared WebCenter wiki. When users access the service for the first time from a transaction page, WebCenter creates a new group space and associates it with that transaction page. Wikis in the group space become shared wikis of that transaction page, which means the wiki page appear for all transaction instances of the transaction page.

When you specify one or more transaction key fields, WebCenter creates an additional group space and associates it with the transaction instance. Wikis in the group space appear when the user selects the specific transaction instance on the transaction page.

This service supports Related Content Security only. The permissions that the service creates for the group spaces are set equivalent to the content reference for the associated transaction page. All user roles with access to the content reference become user group members and are granted the contributor privilege to the group spaces. WebCenter also adds a moderator user group member, if you specify a default moderator role on the WebCenter Options page.

When the transaction page permissions change, you should run the WebCenter security sync batch process to synchronize content reference permissions to the group spaces.

Important! Currently, the service does not support transaction instance level data security.

Understanding Security Synchronization

To function correctly, the permissions of the group space must be the same as the associated content reference object of the transaction page. Your PeopleSoft applications use the WebCenter *SpaceWebService* to create and maintain group spaces security that is synchronized to PeopleSoft security. You must configure the web service security for the WEBCENTER node before using any of the WebCenter related content services. Currently, SAML token web service security is the only security type that the WebCenter application supports to consume WebCenter web services.

When a WebCenter service creates a groups space for a transaction page or its transaction instances, all user roles with access to the content reference object are added as user group members with the contributor privilege. WebCenter supports transaction data security at the role level only. Any user who has access to a transaction page is granted access to all group spaces that are created for that page or its transaction instances. If permissions change for a transaction page, the same changes must be reflected in the WebCenter permissions. You synchronize security by running the WebCenter Security batch process.

WebCenter Security Batch Process

The WebCenter Security sync batch process maintains security synchronization between PeopleSoft transactions and WebCenter services.

Important! You should run this process when the permissions of any transaction page that uses one of the WebCenter related content services changes.

The WebCenter Security synchronization process:

1. Retrieves a list of expected user group members of the group space.

   This list consists of the roles that have access based on the permission lists that appear on the Security tab of the content reference object for the transaction page.

2. Adds the user roles that do not exist in the list.

3. Adds a moderator user group member, if you specify a default moderator role on the WebCenter Options page.

4. Removes user group members that do not appear in the list.

5. Removes individual user members that do not have one of the listed user roles.

Note these points about the security synchronization process:

- The batch process should be run only once after a transaction page permissions are changed to update permissions of existing group spaces.
- The batch process updates only those group spaces that the WebCenter related content services created.
- The batch process does *not* change the permissions of group spaces that are linked to a transaction page or its transaction instances by the WebCenter Links service.

Note: Links do not appear to users if permission to access the associated target is not granted.

Related Language Considerations

Note these points about the use of related language in the WebCenter integration:

- WebCenter and PeopleSoft applications support related language data differently.
- A related language item that you translate into the nonbase language appears to users of all languages.
- You cannot configure a translated item to appear to base language users only, instead of the base language.
- Both the base language and the nonbase item can coexist in the same system.

Important! WebCenter currently does not support Canadian French or UK English as a local language option.

Example

If you have an application that uses English as the base language and French as a related language:

- If a French-speaking user enters a related language item, then it appears to users of all languages.
- An English-speaking user can not provide an English translation of that item to show to English users only, instead of the French version.
- Both English and French items can appear simultaneously in the same system.

Prerequisites

These tables discuss the prerequisite steps to configuring Oracle WebCenter options in the PeopleSoft application.

Perform these tasks in the WebCenter application.

Prerequisite	***Reference***
Install Oracle WebCenter.	See Oracle® Fusion Middleware Installation Guide for Oracle WebCenter 11g Release 1 (11.1.1.5.0).

Prerequisite	***Reference***
Import the PeopleSoft template into the WebCenter application.	See *Oracle® Fusion Middleware User's Guide for Oracle WebCenter Spaces 11g Release 1 (11.1.1.5.0),* Importing Space Templates.
Enable locale switching.	See Registering Provider Classes.
Set the frame busting context parameter to *never*. **Note:** You set the parameter to *never* because the WebCenter pages appear as related content in the lower Related Content frame, which is an iframe.	See *Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework 11g Release 1 (11.1.1.5.0),* Framebusting.
Modify the configuration of the PeopleSoft template to fit your business needs. (Optional)	See *Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces 11g Release 1 (11.1.1.5.0),* Working with Space Templates.
Create new templates. (Optional) **Important!** When you create your own template, you must enable the Discussions and Documents services and include the default Wiki and Blog pages in your template. WebCenter provides no process or program to synchronize any template modifications; if you make more changes to the default template after your initial modifications, then you must recreate the custom template from the delivered template.	See *Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces 11g Release 1 (11.1.1.5.0),* Working with Space Templates.

Perform these tasks in the PeopleSoft application.

Prerequisite	***Reference***
Configure the Integration Gateway.	See “Administering Integration Gateways” (Integration Broker Administration).
Configure single sign on.	See “Implementing PeopleSoft-Only Single Signon” (Security Administration).
Access the WEBCENTER node and enter the WebCenter server address in the **Content URI Text** field.	See Setting Portal Node Characteristics.

Related Links

“Understanding the Integration Network” (Integration Broker Administration)

“Understanding Single Signon” (Security Administration)

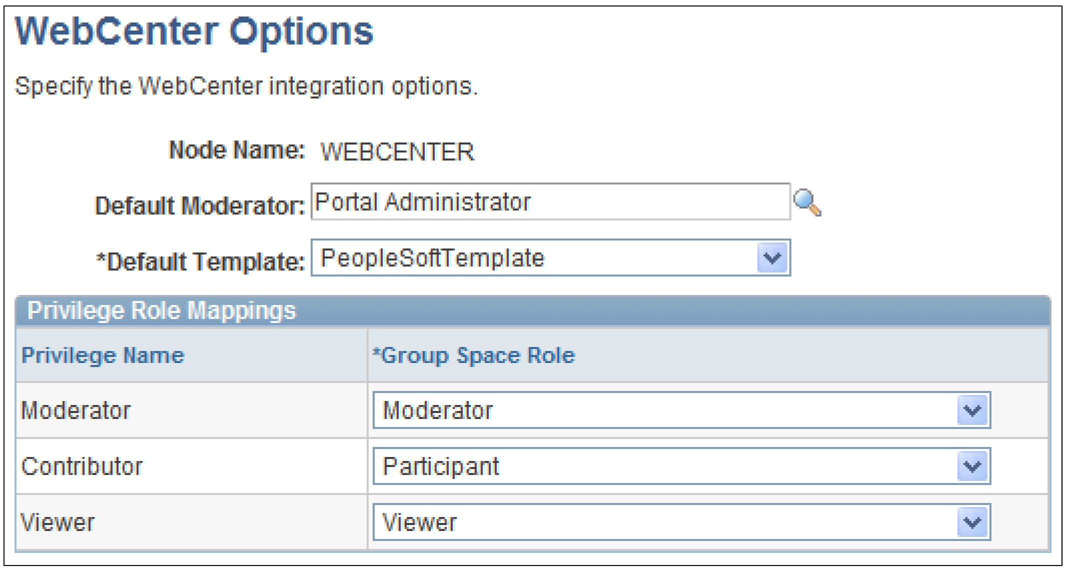
Configuring Oracle WebCenter Options

This section discusses how to configure WebCenter options:

Setting the WebCenter Integration Options

Access the WebCenter Options page (Select **PeopleTools** > **Portal** > **WebCenter Integration** > **WebCenter Options**).

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



WebCenter Options

Field or Control	***Description***
Node Name	Displays the name for the WebCenter application default node, which is *WEBCENTER.*
Default Moderator	Specify the user role to be added as a user group member with the moderator privilege to any group spaces created by WebCenter related content services.
Default Template	Select the template that the WebCenter related content service uses to create new group spaces for PeopleSoft transaction pages. All public group space templates that you have set up in WebCenter appear in the drop-down list. The default value is *PeopleSoftTemplate.* This field requires a value.

Privilege Role Mappings

Use the fields in the Privilege Role Mappings grid to correlate the PeopleSoft privilege to the WebCenter group space role. This grid appears when you select a valid default group space template. All fields in this grid are required when you select a default group space template.

Field or Control	***Description***
Privilege Name	Displays the PeopleSoft privilege.
Group Space Role	Select from the list of privilege roles that exist in the specified default template. If you select a new default template, the values in drop-down list change to reflect the new default. Select from these default values: • *Moderator* • *Participant* • *Viewer* This field requires a value.

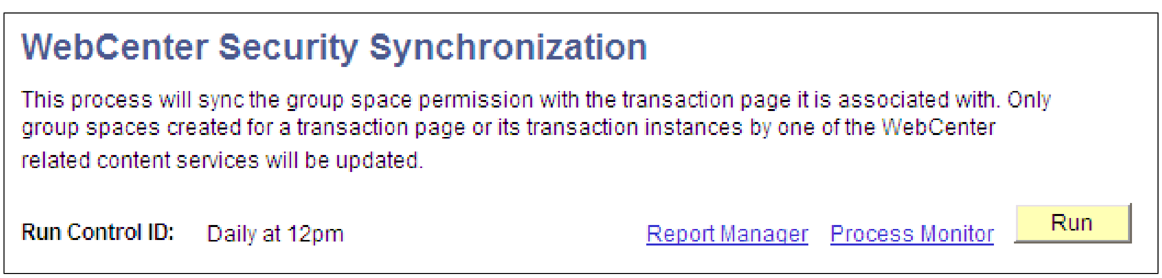
Synchronizing Oracle WebCenter and PeopleSoft Application Security

This section discusses how to run the WebCenter Security Sync batch process.

Running the WebCenter Security Sync Batch Process

Access the WebCenter Security Synchronization page. (Select **PeopleTools** > **Portal** > **WebCenter Integration** > **WebCenter Security Sync**).

This example illustrates the fields and controls on the WebCenter Security Synchronization page.



To run the WebCenter Security Synchronization batch process:

1. Click the **Run** button.

2. Click the **Process Monitor** link to verify success.

Managing WebCenter Links

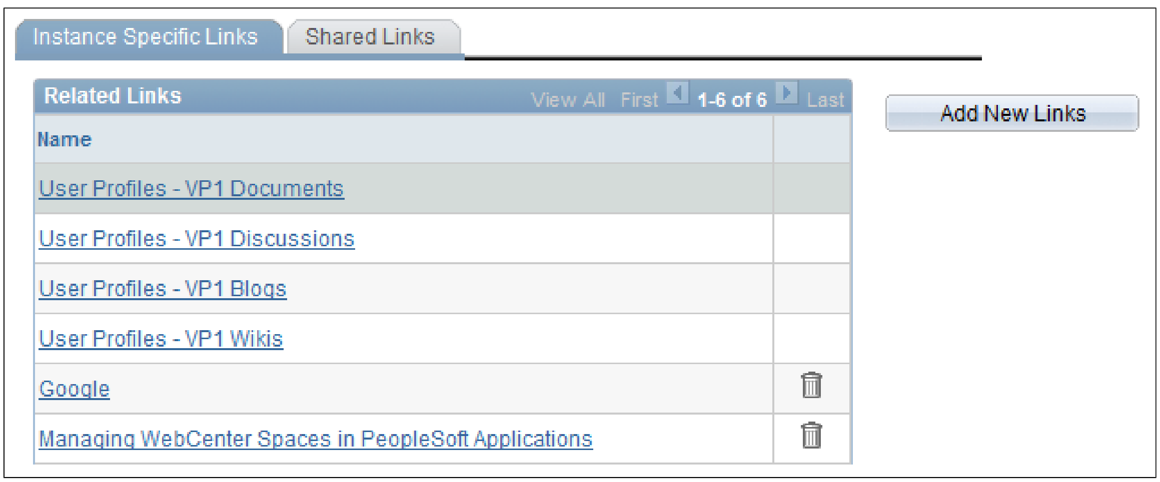
This section describes how to:

- Manage instance-specific links.
- Manage shared links.
- Add group space links.
- Add external URL links.

Managing Instance-Specific Links

Access the Instance Specific Links page **(Select the Instance Specific Links tab)**.

This example illustrates managing instance-specific links on the Instance Specific Links page.



To add new instance-specific links, click the **Add New Links** button. The Add New Links page appears.

See Adding Group Space Links, Adding External URL Links.

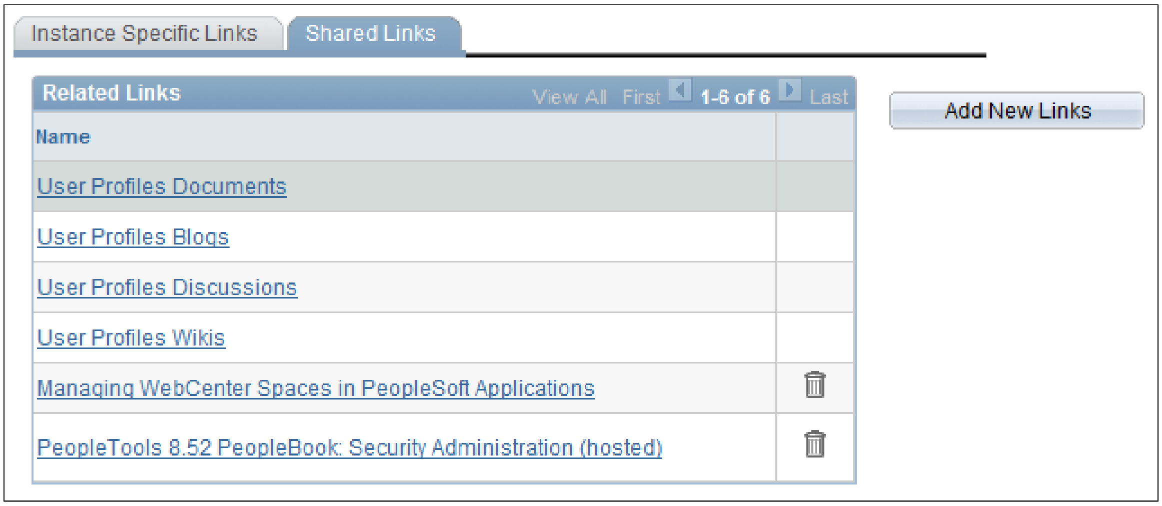
To delete instance-specific links, click the **Delete Link**(trash can) icon.

Note: Links to services within the current group space appear automatically and cannot be removed.

Managing Shared Links

Access the Shared Links page **(Select the Shared Links tab)**.

This example illustrates managing shared links on the Shared Links page.



To add new shared links, click the **Add New Links** button. The Add New Links page appears.

See Adding Group Space Links, Adding External URL Links.

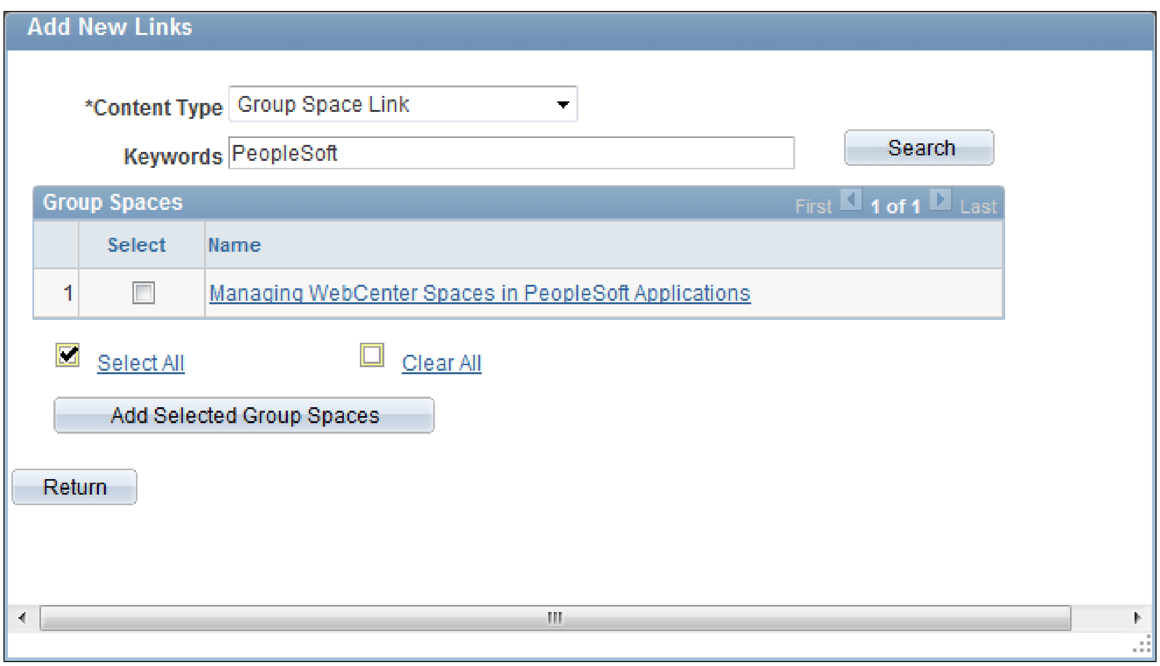
To delete shared links, click the **Delete Link**(trash can) icon.

Note: Links to services within the current group space appear automatically and cannot be removed.

Adding Group Space Links

Access the Add New Links page **(Click the Add New Links button)**.

The following screenshot illustrates adding group space links on the Add New Links page.



To add new group space links:

1. Select *Group Space Link* as the content type.
2. To search the group spaces that are available to you in your WebCenter application, enter keywords and click the **Search** button.

Note: Keywords for each group space are managed by the WebCenter administrator in the WebCenter application

---

3. Select the group spaces for which you want to add links.
4. Click the **Add Selected Group Spaces** button.

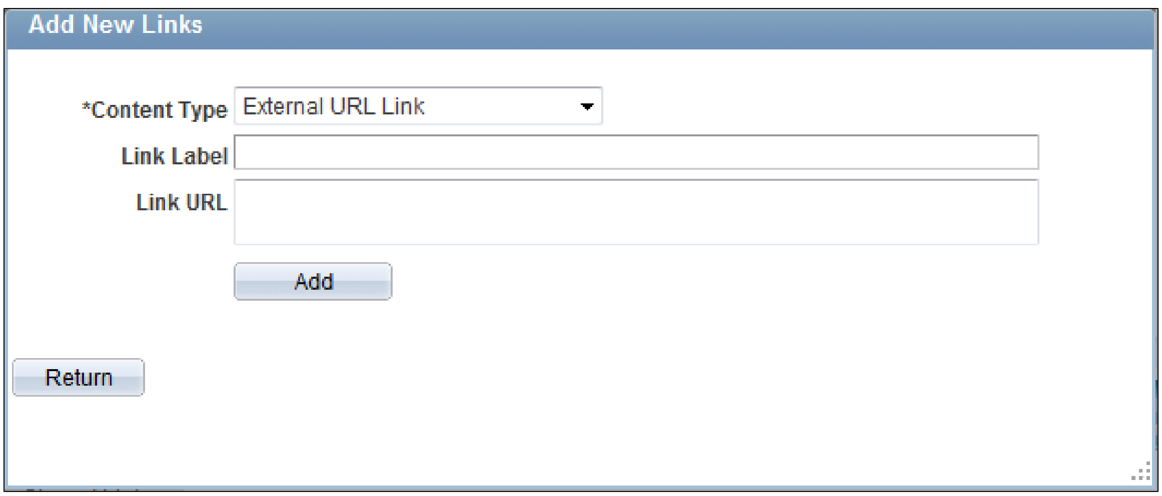
   The application returns you to the WebCenter Links page.

Click the **Return** button to go back to the previous page.

Adding External URL Links

Access the Add New Links page **(Click the Add New Links button)**.

The following screenshot illustrates adding external URL links on the Add New Links page.



To add new external URL links:

1. Select *External URL Link* as the content type.
2. Enter the label that you want to appear in the **Related Links** grid on the WebCenter Links page.
3. Enter the target address of the external link in the **Link URL** field. Use this example as a model: *https://support.oracle.com*.
4. Click the **Add** button.

   The application returns you to the WebCenter Links page.

Click the **Return** button to go back to the previous page without saving your entry.

Chapter 32

Enabling Locale Switching Through Query Parameters in Oracle WebCenter

Registering Provider Classes

Note: As of PeopleTools 8.54, integration with Oracle WebCenter through the delivered related content services has been desupported. These service definitions may be removed from a future release of PeopleTools.

To support locale switching by using the WCGS_PreferredLanguage=*LangCode* query parameter, you must manually register the URLBaseProvider class.

You can do that manually or by using a WebLogic Scripting Tool (WLST) command.

Manual Registration

To register the provider class manually:

1. Using a text editor, open the adf-config.xml file.
2. Find the adf-config element and enter the following xml inside the tag:

```
 <adf-webcenter-generalsettings-config xmlns="http://xmlns.oracle.com/webcente⇒

r/generalsettings/config">
   <provider-class>
     oracle.webcenter.generalsettings.model.provider.URLBaseProviderprovider
   </provider-class>
 </adf-webcenter-generalsettings-config>
```

3. Save the file.

WLST Registration

To use WLST to register the provider class:

1. Open the WebLogic Scripting Tool.
2. Run this command:

```
setWebCenterGeneralSettingsProvider('webcenter','oracle.webcenter.generalsetti⇒

ngs.model.provider.URLBaseProvider')
```

Note: The first parameter is the application name. The second parameter is the providerClass. Sometimes you must supply a third parameter.

or this command:

```
setWebCenterGeneralSettingsProvider('webcenter','oracle.webcenter.generalsetti⇒
ngs.model.provider.URLBaseProvider','WLS_Spaces1')
```

Note: The third parameter is the spaces cluster service name or spaces service name.

3. Restart the server.

See Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework 11g Release 1 (11.1.1.5.0), "Getting Started Using the Oracle WebLogic Scripting Tool."

Chapter 33

Configuring OBIEE Admin User Permissions in Enterprise Manager 11g

Configuring OBIEE Server Admin User Permissions in Enterprise Manager 11g

This section discusses the necessary permission configuration for using OBIEE.

Note: For use with 11g only.

The user that you configure in the Server Setup page must have *impersonate* permissions in Enterprise Manager. In release 10g, the BI Administrator satisfies this requirement. In release 11g, the user must be assigned the BI System application role.

To enable the BI System application role in Enterprise Manager 11g:

1. Sign in to Webserver Enterprise Manager as an administrator.
2. In the left side navigation, click **Business Intelligence** > **coreapplication**.
3. When the new page appears, click **Configure and manage Application Roles**.
4. When the new page appears, click **BI System**.
5. In the users column, click **Add User** and add the same user that you entered as the Admin User on the Server Setup page.
6. Click the **OK** button to save your changes.

Chapter 34

Language Support for Consuming and Producing Remote Portlets

Language Support for Consuming Remote Portlets

Note: The ServiceDescription provides translations for the following strings: title, shortTitle, description, displayName, and keywords. The main language value is placed inside the <offeredPortlets> tag, other language strings are inside <resources> tags at the end of xml file. PeopleSoft applications store the mapping of three-character PeopleSoft language codes such as *ENG, FRA, CFR,* and *DUT,* as well as ISO locales such as *en, en-US, fr, fr-ca,* and *nl*.

When we consume remote portlets, we request service description XML from the remote producer. We store the definition of the remote producer in the PeopleSoft database. We then parse service description XML and store the definitions of the remote portlets. If multi-language data is in the service description, we map ISO locales to PeopleSoft language codes and store the values in our database along with the correct PeopleSoft language codes.

Note: We are interested only in such locales that correspond to installed PeopleSoft language codes. All other locales are ignored.

When parsing the remote ServiceDescription, we look for matches between ISO locales and installed PeopleSoft language codes. We accept the string values when matches are found. Here are the rules:

- Accept if an exact match is found, for example, if *fr-ca* (CFR) is installed and *fr-ca* is provided in ServiceDescription.
- If the child is installed and they provide only the parent, we accept parent for child. For example, if *fr* (FRA) and *fr-ca* (CFR) are installed and only *fr* is provided, it is matched to both *fr* and *fr-ca.*
- If the parent locale is installed, but only children are provided, then accept the child for the parent. For example, if *fr (FRA)* is installed and *fr-ca* (CFR) is provided, then accept it for FRA.

Language Support Rules for Producing Remote Portlets

Note: The ServiceDescription provides translations for the following strings: title, shortTitle, description, displayName, and keywords. The main language value is placed inside the <offeredPortlets> tag, other language strings are inside <resources> tags at the end of xml file. PeopleSoft applications store the mapping of three-character PeopleSoft language codes such as *ENG, FRA, CFR,* and *DUT,* as well as ISO locales such as *en, en-US, fr, fr-ca,* and *nl*.

When we produce remote portlets, we look at the definitions of the appropriate content references and generate the ServiceDescripton. If any multi-language strings (such as cref label, description, and keywords) exist, we may have to include them into ServiceDescripton. The remote producer sends us the list of desired locales he is interested in and in response, we determine whether these desired locales have matching installed PeopleSoft language codes.

When generating the ServiceDescription, the PeopleSoft system tries to find matches for the desired locales to PeopleSoft installed languages using the following rules:

- If the locale is installed, the system returns it.
- If only the parent is installed, the system returns the parent. For example, if the desired locale is *fr-ca* and only *fr* is found, the system returns *fr*.
- If no match is found, the system returns *en* (English).

Chapter 35

Overriding the Default Breadcrumb Behavior

Understanding the Default Breadcrumb Behavior

Important! Breadcrumbs, drop-down menus, and other forms of classic navigation are no longer supported. The information in this topic pertains to a prior version of PeopleSoft classic applications only. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see "Working with Fluid Pages and Controls" (Applications User's Guide).

Breadcrumbs refer to the collective sequence of items that appear above the content area when you navigate to transaction pages or navigation pages. The breadcrumbs display the path you took to get to where you are—that is, the page you are on.

The PeopleSoft application dynamically generates the individual items in the breadcrumb list using the following criteria:

- The first item in the sequence is always the Main Menu.
- The intermediate items in the sequence are the names of each item that you click to access the resource; menu items and component names appear in the order that you click them.
- The last item in the sequence is:
  - The component label when you access a transaction page.
  - The folder label when you access a navigation page.
  - The WorkCenter or dashboard label when you access the WorkCenter or dashboard.
  - The value of the focus node when you access a SmartNavigation page.

Breadcrumbs can reveal both the navigational path to your current location and the hierarchical or parent-child relationships that exist among the items in the breadcrumb sequence.

The following example illustrates how breadcrumbs are constructed for a transaction page, and how those breadcrumbs are updated when you navigate to additional pages:

1. Access the User Profile page (select PeopleTools, Security, User Profiles, User Profiles).

The breadcrumbs display Main Menu > PeopleTools > Security > User Profiles > User Profiles as shown in this example:

2. Select a user profile—for example, psadmin.
3. Go to the Roles page.

Note: No breadcrumb update occurs because the Roles page is within the User Profiles component.

4. Click the View Definition link for a specific role—for example, for PeopleSoft User.

   The breadcrumbs are updated to append the new component at the end of the list. The breadcrumbs display Main Menu > PeopleTools > Security > User Profiles > User Profiles > Roles as shown in this example:

5. Go to the Permission Lists page.

Note: No breadcrumb update occurs because the Permission Lists page is within the Roles component.

6. Click the View Definition link for a specific permission list—for example, for PTPT1000.

   The breadcrumbs are updated to append the new component at the end of the list. The breadcrumbs display Main Menu > PeopleTools > Security > User Profiles > User Profiles > Roles > Permission Lists as shown in this example:

7. Click Roles in the breadcrumbs.

   The breadcrumbs are updated to remove Permission Lists and display the role definition for the selected role.

See “Navigating Breadcrumbs” (Applications User’s Guide) for more information on working with breadcrumbs. This topic includes a list of limitations on when the breadcrumbs are not updated.

However, there might be situations in which you do not want the default breadcrumb behavior—that is, to append the next item to the breadcrumb list. In these situations, you can use either of two methodologies to modify the default, PeopleTools breadcrumb behavior:

- Use the CreateBreadcrumb built-in function to append a breadcrumb element containing application-specific data to the user’s list of breadcrumbs. This function can be included in any PeopleCode event so an application-specific breadcrumb can be added whenever it is needed. See “CreateBreadcrumb” (PeopleCode Language Reference) for more information.

- Add content reference attributes to override the default breadcrumb behavior whenever the content reference is the target—that is, whenever it is navigated to. SeeOverriding Default Breadcrumbs Using Content Reference Attributes for more information.

Overriding Default Breadcrumbs Using Content Reference Attributes

Important! Breadcrumbs, drop-down menus, and other forms of classic navigation are no longer supported. The information in this topic pertains to a prior version of PeopleSoft classic applications only. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications. For information on working with the fluid user interface, see "Working with Fluid Pages and Controls" (Applications User's Guide).

You can use optional content reference attributes to override the default breadcrumb behavior for a target content reference when you want to do any of the following:

- Specify a different label as the text to either append to the breadcrumb list or to replace the last item in the list rather than using the label from the target component.

  Note: The default label is the label for the content reference.

- Skip the breadcrumb update—that is, the breadcrumbs remain unchanged.
- Replace the last item in the breadcrumb list rather than append to the list.
- Preserve the existing breadcrumbs *and* append the WorkCenter page that is the target to the existing list. (Typically, a transfer to a WorkCenter page would regenerate the entire breadcrumb list starting from the Main Menu.)
- Append to the breadcrumb list even if the target is a hidden content reference, in a hidden folder, or when the component is defined such that **Display in Navigation** is disabled.

Note: Alternatively, you can use the CreateBreadcrumb built-in function to append a breadcrumb element containing application-specific data to the user's list of breadcrumbs. This function can be included in any PeopleCode event so an application-specific breadcrumb can be added whenever it is needed. See "CreateBreadcrumb" (PeopleCode Language Reference) in *PeopleTools: PeopleCode Language Reference* for more information.

Define the content reference attributes on the content reference that is being navigated to. You can specify one or both of the following attributes:

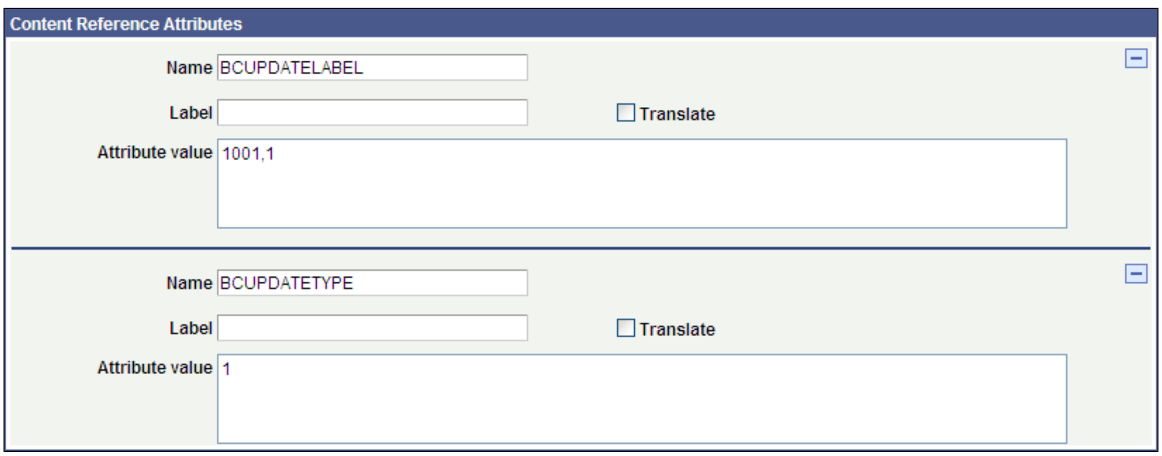
- Use the BCUPDATELABEL attribute to specify a message set and number that defines the label to be displayed in the breadcrumbs.
- Use the BCUPDATETYPE attribute to specify a numeric value that indicates:
  - **0** – Do not update the breadcrumbs.
  - **1** – Replace the last item in the breadcrumbs with the new element.

- **2** – Preserve the existing breadcrumbs and append the WorkCenter page that is the target to the existing list.
- **3** – Append hidden items to the breadcrumb list.

To modify the breadcrumb behavior for a content reference:

1. Go to **PeopleTools** > **Portal** > **Structure and Content.**
2. Navigate the menu folder hierarchy to the content reference that you want to modify.
3. Click the Edit link for the content reference.
4. In the Content Reference Attributes group box, add an attribute for each aspect that you want to specify (BCUPDATELABEL, BCUPDATETYPE, or both).
5. For each attribute, deselect the **Translate** option.
6. Save the content reference definition.

The following example shows both the BCUPDATELABEL attribute and the BCUPDATETYPE attribute defined for a content reference:



Related Links

Defining Content References

Chapter 36

PeopleTools Web Libraries

Understanding Web Libraries

A web library is a derived or work record the name of which starts with WEBLIB_. All PeopleSoft iScripts are embedded in records of this type. An iScript is a specialized PeopleCode function that generates dynamic web content. Administrators should make sure that users have the proper access to web libraries.

For example, the default navigation system for PeopleSoft Pure Internet Architecture users is implemented by means of a web library. If users do not have the proper authorization to the web library and its associated scripts, then they won't have proper access to the system. If users are not authorized for a particular web library or script, then they can't invoke it. After you add a web library, you set the access for each script function individually. Invoking an iScript requires the assembly of a URL. Developers assemble the URL by using PeopleCode.

PeopleTools Web Libraries

This table lists the primary PeopleTools web libraries:

Web Library Name	***Description***
WEBLIB_CTI	Contains iScripts for generating the Computer Telephony Integration (CTI) console and the CTI free-seat extension page.
WEBLIB_FILEATT	Contains HTML for file attachments.
WEBLIB_FILEHTTP	Contains libraries for getting and viewing attachments.
WEBLIB_IB	Contains iScripts for define PeopleSoft Integration Broker functions.
WEBLIB_MCF	Contains iScripts for generating the Multichannel Framework (MCF) console, Agent to Customer chat window, Customer to Agent chat window, Agent to Agent chat window, and MCF tracer window.
WEBLIB_MCFLINK	Generates the HTML page for MultiChannel Framework (MCF) Console Link.

Web Library Name	***Description***
WEBLIB_MDF	Contains libraries for the PeopleSoft Health Center.
WEBLIB_MSGWSDL	This is the record behind the page for the enterprise integration point (EIP) Web Services Description Language (WSDL) generation.
WEBLIB_NAVMAIN	Contains libraries for internet client Query Service (Nav service).
WEBLIB_OPT	Contains iScripts for generating Optimization Framework functions.
WEBLIB_PGAPPRA	Contains libraries for calling ObjectDoMethod for executing any application class.
WEBLIB_PORTAL	Contains the following six fields with FieldFormula iScript PeopleCode, each of which relates to a functional area of the portal: • PORTAL_DYN_TEMP: Support for the dynamic template. • PORTAL_HEADER: Support for the header portion of the page and some generic routines. • PORTAL_HOMEPAGE: Support for homepage runtime interaction, including the homepage version of the menu navigation. • PORTAL_NAV: Main support routines for side navigation. • PORTAL_PGLT_PREV: Support for the pagelet preview functionality. • PT_WSRP: Used to sign in to the PeopleSoft application when it is defined as a producer.
WEBLIB_PPM	Contains iScripts for generating Performance Monitor functions.
WEBLIB_PTAI	Contains libraries for activity guide action items.
WEBLIB_PTAF WEBLIB_PTAFEMC	Contains iScripts for generating Approval functions.
WEBLIB_PTAL	Contains libraries for WorkCenter templates
WEBLIB_PTBR	Contains branding libraries.

Web Library Name	***Description***
WEBLIB_PTCBD	Contains iScripts for displaying common Dashboard components.
WEBLIB_PTCXM	Contains iScripts for displaying Context Manager pagelets.
WEBLIB_PTDIAG	Contains iScripts for displaying PeopleTools Diagnostics pages.
WEBLIB_PTGP	Contains libraries for guided processes (for example, activity guides).
WEBLIB_PTGPLT WEBLIB_PTGPLT1	Contains libraries for grouplets (for example, tiles).
WEBLIB_PTHDR	Contains libraries for generating the portal header.
WEBLIB_PTIA	Contains libraries for PeopleTools impact analysis.
WEBLIB_PTIFRAME	Contains iScripts for portal favorites and popup functionality.
WEBLIB_PTNUI	Contains libraries for fluid features such as fluid homepages, the NavBar, tiles. and so on.
WEBLIB_PTOBIEE	Contains iScripts for displaying Business Intelligence (Analytics) pagelets.
WEBLIB_PTPN	Contains libraries for configuration of user data mapping of the Push Notification Framework.
WEBLIB_PTPP	Contains libraries for portal utility iScripts.
WEBLIB_PTPP_LN	Contains libraries for additional portal utility iScripts.
WEBLIB_PTPP_SC	Contains iScripts for generating navigation pages and the Main Menu navigation pagelet.
WEBLIB_PTPP_PGT	Renders navigation pagelets without Pagelet Wizard.
WEBLIB_PTPPB	Contains iScripts for displaying Pagelet Wizard pagelets as homepage pagelets and template pagelets
WEBLIB_PTPPB1	Contains an iScript for displaying Pagelet Wizard pagelets as WSRP portlets.

Web Library Name	***Description***
WEBLIB_PTRC	Contains iScripts for displaying related content.
WEBLIB_PTRTE	Contains iScripts for uploading inline images in rich text.
WEBLIB_PTSF WEBLIB_PTSF_SB	Contains libraries for the PeopleSoft Search Framework.
WEBLIB_PTSIGNIN	Contains libraries for PeopleTools signon iScripts.
WEBLIB_PTTILE	Contains libraries for tile-related iScripts.
WEBLIB_PTWC	Contains libraries for WebCenter iScripts.
WEBLIB_PT_NAV	Contains iScripts for the menu pagelet, drop-down navigation, and left-hand navigation for transaction pages.
WEBLIB_QUERY	Contains the iScripts for generating a URL for running queries.
WEBLIB_RPT	Contains iScript for the Run report to window output option. Supports access to the new browser window.
WEBLIB_SDK	Contains iScripts for generating software development kit functions.
WEBLIB_SDKAMORT	Contains libraries for the SDK amortization pagelet iScript.
WEBLIB_SDK_BI	Contains iScripts for generating sample code showing a call to the PeopleSoft Business Interlinks PSHttpEnable function to return information from a remote site that is formatted into the HTML display.
WEBLIB_SOAPTOCI	This is the entry point for ExcelToCI and the WSDL Discovery.
WEBLIB_TIMEOUT	Contains iScripts for generating an inactivity timeout warning script.
WEBLIB_UNREMREG	Contains libraries for fetching remote node objects.
WEBLIB_XMLLINK	Contains iScripts for generating PeopleSoft Business Interlinks XML functions.

Web Library Name	***Description***
WEBLIB_XMLP	Contains libraries for Oracle BI Publisher.
WEBLIB_XSLT	Contains libraries for Oracle XSL Mapper launcher.

Chapter 37

Classic Components Delivered as Classic Plus

PeopleTools Components Delivered as Classic Plus

Starting with PeopleTools 8.56, the pages contained within certain PeopleTools classic components have been reworked and are now delivered with fluid-like styling which is also referred to as “classic plus” styling. When the either the *Classic Plus* or the *Classic Plus Pre 858* theme type has been enabled on the Branding System Options page, the pages in these components will be displayed with fluid-like styling rather than as standard classic pages. In addition, the pages in any components delivered with your PeopleSoft applications that have also been reworked will be displayed with fluid-like styling; see your PeopleSoft application documentation for information on which components have been delivered with this fluid-like styling. When any other theme type is selected on the Branding System Options page, these classic pages are rendered like other standard classic pages and components.

Note: The page controls on these “classic plus” pages will resemble the page controls on fluid pages. However, because these components and pages remain as classic definitions, certain aspects of the PeopleSoft fluid user interface experience will be absent—for example, dynamic, responsive rendering across all supported device types.

To prepare a component for classic plus styling, PeopleTools and PeopleSoft application development teams have reworked some of the page controls on each of the pages within the component. Then, the teams have set the Classic Plus check box on the Style tab of the Component Properties dialog box. A component can also be configured with the Classic Plus option on the Component Branding page. Despite this setting, other component and page properties remain set as standard classic components and pages.

In addition, the pagelet area of classic WorkCenters can be configured to display with classic plus styling. Only the pagelet area of the WorkCenter is affected by this setting. The components displayed in the target content area of the WorkCenter are enabled for classic plus on a component by component basis.

Note: Classic plus styling is not supported on classic homepages, classic dashboards, and WorkCenter dashboards.

Related Links

Configuring Branding System Options
“Reworking Components for Classic Plus Display” (Application Designer Developer’s Guide)
Branding Classic Components
Administering WorkCenter Starting Pages

List of PeopleTools Components Delivered as Classic Plus

As of PeopleTools 8.58, all PeopleTools components have been delivered with the Classic Plus check box selected on the Style tab of the Component Properties dialog box. This includes:

- All components under the Worklist menu.
- All components under the Application Diagnostics menu.
- All components under the Tree Manager menu.
- All components under the Reporting Tools menu.
- All components under the PeopleTools menu.
- The following PeopleTools-delivered components under the root menu:
  - My System Profile.
  - My Feeds.

