ORACLE®

PEOPLESOFT

PeopleTools 8.55: Fluid User Interface Developer's Guide

November 2017

ORACLE®

PeopleTools 8.55: Fluid User Interface Developer's Guide
Copyright © 1988, 2017, Oracle and/or its affiliates. All rights reserved.

Documentation Accessibility

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

Access to Oracle Support

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

Contents

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

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

PeopleSoft Hosted Online Help

You access the PeopleSoft Online Help on Oracle's PeopleSoft Hosted Online Help website, which enables you to access the full help website and context-sensitive help directly from an Oracle hosted server. The hosted 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, because that documentation is now incorporated into the hosted website content. The Hosted Online Help website is available in English only.

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 After a new release is posted, previous releases remain available but are no longer updated.

Locally Installed Help

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

In addition, if you locally install the PeopleSoft Online Help, you can use any search engine for full-text searching. Your installation documentation includes instructions about how to set up Oracle Secure Enterprise Search for full-text searching. See *PeopleTools Installation* for your database platform, "Installing PeopleSoft Online Help." If you do not use Secure Enterprise Search, see the documentation for your chosen search engine.

Note: Beginning with PeopleSoft PeopleTools 8.55.11, you can use Elasticsearch as the search engine for locally installed PeopleSoft Online Help.
See Oracle Support Document 2189508.1 (PeopleTools 8.55.11 Elasticsearch Documentation Home Page) for more information on using Elasticsearch with PeopleSoft.

Note: Before users can access the search engine on a locally installed help website, you must enable the Search field. For instructions, select About This Help, Managing Locally Installed PeopleSoft Online Help, Enabling the Search Button and Field in the Contents sidebar of the online help site.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has

a different structure and it does not include the interactive navigation features that are available in the online help.

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.

Typographical Convention	*Description*
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.
⇒	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

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

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

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

Region and Industry Identifiers

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

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

Region Identifiers

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

• Asia Pacific

• Europe

• Latin America

• North America

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

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

- What's new in the PeopleSoft Online Help.

- PeopleSoft Online Help accessibility.

- Accessing, navigating, and searching the PeopleSoft Online Help.

- Managing a locally installed PeopleSoft Online Help website.

PeopleTools Related Links

Hosted Online Help Home

PeopleSoft Information Portal

My Oracle Support

Oracle University

Oracle's PeopleSoft PeopleTools 8.55 Documentation Home Page (Doc ID 2052626.1)

"PeopleTools Product/Feature PeopleBook Index" (PeopleTools 8.55: Getting Started with PeopleTools)

Contact Us

Send your suggestions to PSOFT-INFODEV_US@ORACLE.COM. Please include release numbers for the PeopleTools and applications that you are using.

Follow Us

Facebook.

YouTube

Twitter@PeopleSoft_Info.

PeopleSoft Blogs

LinkedIn

Chapter 1

Getting Started with Fluid User Interface Development

Fluid User Interface Overview

The PeopleSoft Fluid User Interface is designed to be a significant enhancement to the PeopleSoft's "classic" user interface, which has been the interface display on browsers for PeopleSoft end users for well over a decade. The PeopleSoft Fluid User Interface moves away from pixel-perfect page layout and provides greater flexibility with the enhanced use of cascading style sheets (CSS3), HTML5, and JavaScript (if needed).

PeopleSoft application fluid pages scale gracefully from large screen devices such as laptops and desktops down to the reduced viewing space of tablets and smartphones. Many commercial websites use a similar design model where the presentation and layout of information is adjusted dynamically to conform to the dimensions of the user's device. The Fluid User Interface design approach gives developers just this type of control over the user experience. When a larger screen size is detected (a screen with more "real estate") the application content will adjust and conform accordingly to fill the space. Similarly, if a smaller screen size is detected, non-essential information can be removed and the presentation of content will adjust to "flow" in a very usable fashion.

This documentation focuses on the tools, the approaches, and the techniques that application implementation teams will use to create, configure, and customize fluid application pages.

Note: The PeopleSoft Fluid User Interface is an evolving technology. As information becomes available or is requested, PeopleTools will post updates as needed on the following My Oracle Support page: PeopleSoft Fluid User Interface Supplemental Documentation(Doc ID 1909955.1).

Implementing PeopleSoft Fluid User Interface

Developing or customizing PeopleSoft Fluid User Interface applications involves many of the same tools and techniques used for developing or customizing classic PeopleSoft applications. A working knowledge and some expertise in the most current, industry-standard internet development tools and techniques is also required. The following list contains the main tools and technologies Oracle recommends developers are familiar with prior to beginning a fluid application implementation.

- PeopleSoft Application Designer

- PeopleCode

- PeopleTools Portal Technology

- PeopleSoft Pure Internet Architecture

- HTML 5.0

- Cascading Style Sheets 3.0

- JavaScript

Chapter 2

Understanding PeopleSoft Fluid User Interface

Understanding the Need for a Fluid User Interface

A traditional, or classic, PeopleSoft application displayed in a browser assumes the screen size of the device running the application is of a standard, relatively predictable size, such as a browser running on a desktop or laptop computer. The classic PeopleSoft application *will run* on a mobile device, such as a tablet or cell phone, but it *is not* tailored or optimized for the mobile device.

With the traditional PeopleSoft application, the interface display is defined in Application Designer using a fixed layout, based on and bound to a pixel system. In the fixed layout, a developer can meticulously align fields, labels, group boxes, and so on, and be reasonably assured the page will appear in the browser running on a desktop or laptop just as it appears in the design-time layout. The traditional PeopleSoft application development design-time layout provides a WYSIWYG (what you see is what you get) scenario.

However, the fixed layout does not adapt -- expand or contract-- based on the device or the size of the screen on which it renders. A classic PeopleSoft application does not display dynamically. While an end user can access and use a traditional PeopleSoft application with a cellular phone, for example, most likely the user will need to shrink or expand the display in order to interact with the fields, with maybe only a small percentage of the page being visible at a time, losing the context of the information on the page.

With the emergence and popularity of mobile devices, namely tablets and smart cellular phones, PeopleSoft applications need to display and run on a wide range of mobile devices. For example, a small, light-weight self-service application must be equally usable on a browser running on a desktop, laptop, iPhone, Android, iPad, and so on.

To address this business demand, PeopleTools enables you to develop and customize components and pages for a *fluid display* when running on supported mobile devices. PeopleSoft applications that have been developed and enabled for fluid display will appear dynamically, depending on the device and screen size used for access, all based on a single page layout definition. That is, you create a *single* page definition, using the flexible fluid design approach, and it will display as needed on all the different supported devices, be it an Andriod phone or iPhone or tablet.

Terms used to describe similar design approaches in the industry include, *fluid design*, *responsive design*, *relative* layout, and *elastic* layout, to name a few. For PeopleTools, we refer to this approach as *fluid*, as in *fluid design, fluid applications, fluid mode* and so on. Regardless of the term used, the goal remains the same, which is to enable organizations to deploy applications to all the devices being used to conduct business such that the content of the PeopleSoft page resizes and adapts to the size of the browser window of the device

Understanding the Characteristics of Fluid Applications

A PeopleSoft application enabled for fluid mode recognizes the device used to access it, and displays so that the application appears naturally, as expected by the user of a certain device type. For example, the following shows a PeopleSoft application page enabled for fluid display appearing on a browser on a laptop or a tablet device.

Image: Fluid page

This example shows a fluid page displayed on a laptop or tablet.
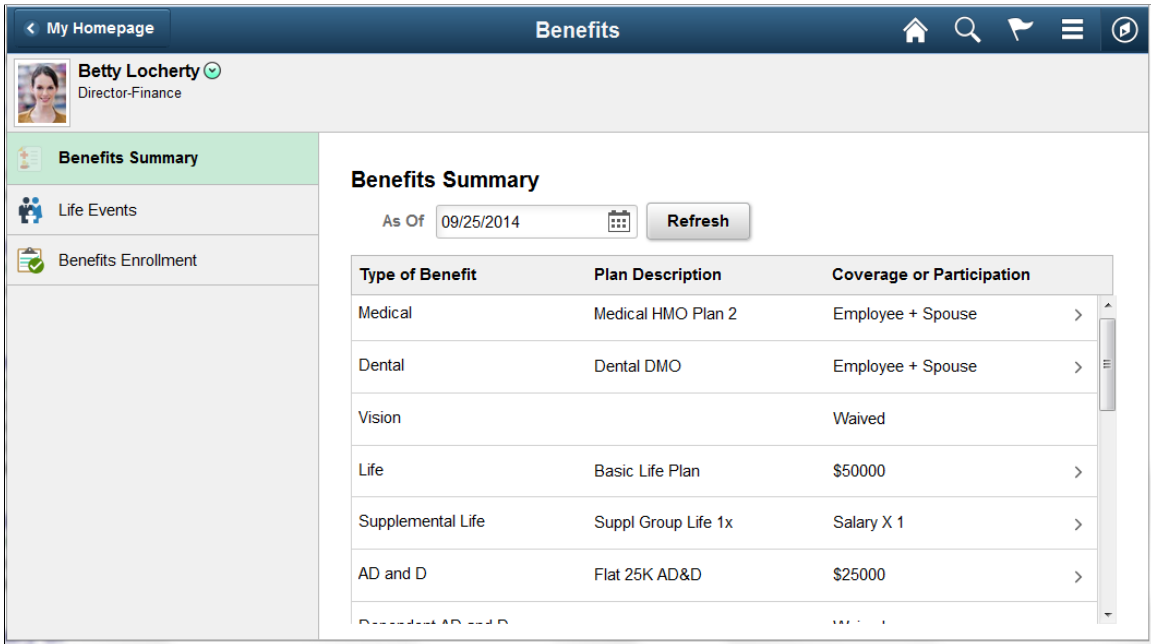
If you access that same page using a cell phone and view it in landscape mode, it would appear similar to the following example.

Image: Fluid page in landscape mode

The following example shows how a fluid page may appear on a smartphone in landscape mode.
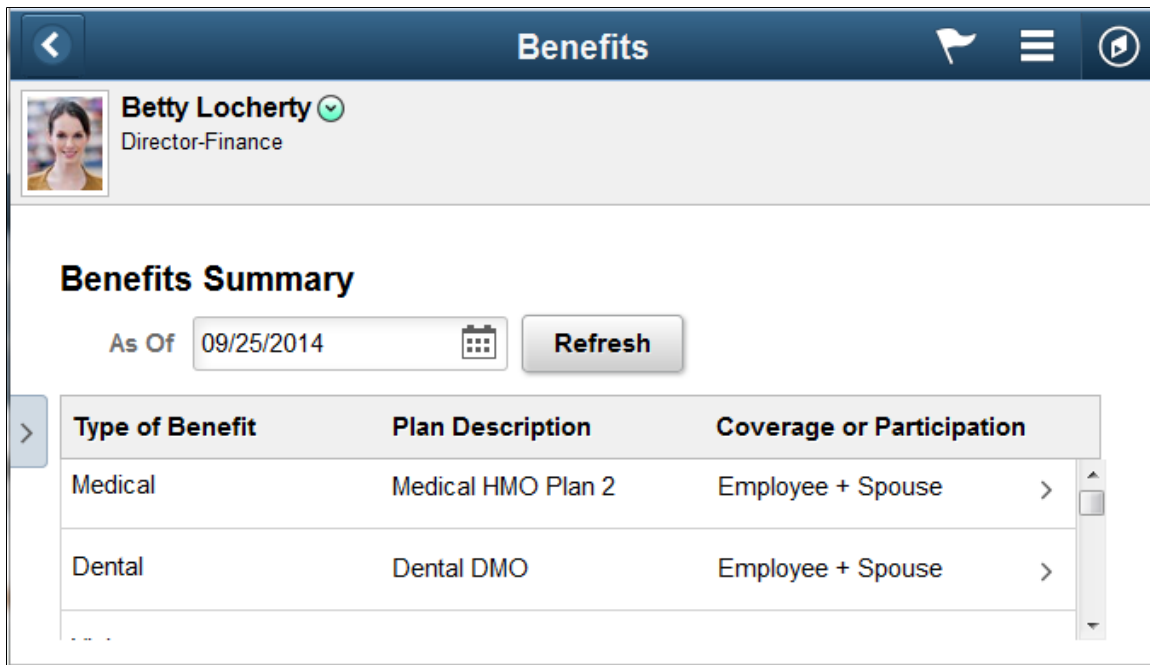
Notice that the page shrinks to fit the screen size of the device. The elements in the header are the same, only closer together, and the grid displays only the most critical columns. The user can use typical gestures, such as swiping (on homepages only), to scroll through the fluid page.

If you access the same page using a cell phone and view it in portrait mode, you will notice additional changes as the page further adapts and responds to the adjusted screen display. The PeopleSoft Fluid User Interface is *adaptive* in that the system detects the device and generates the UI appropriate to that device, and it is *responsive* in that the CSS defined for the application responds to user actions on the client device, such as changing from portrait to landscape mode.

Note: The PeopleSoft Fluid User Interface adaptive qualities require a server trip, while the responsive qualities do not.

Image: Fluid page in portrait mode

The following example shows a fluid page displayed on a smartphone in portrait mode.
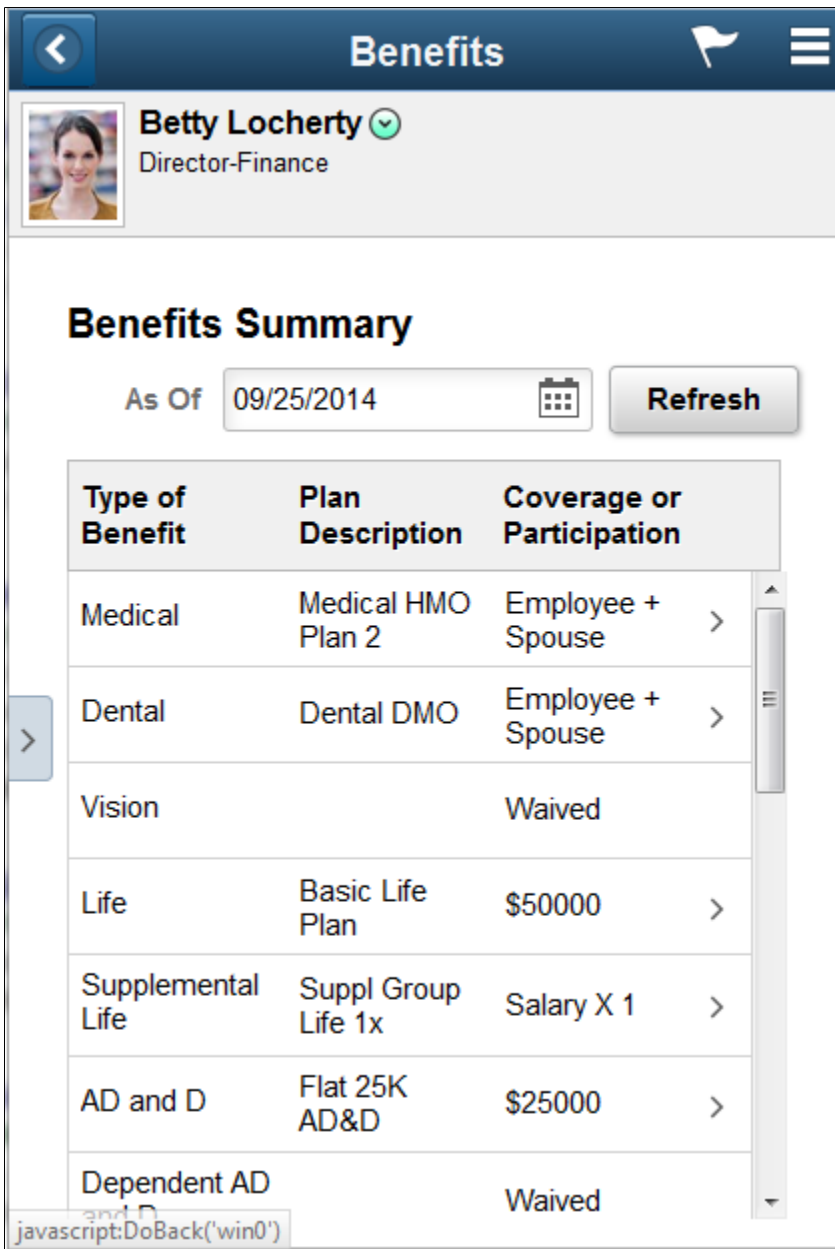
Notice that the system further shrinks the page, while maintaining the fundamental structure.

Related Links

"Working with Fluid Homepages and Dashboards" (PeopleTools 8.55: Applications User's Guide)
"Working With Fluid Pages and Controls" (PeopleTools 8.55: Applications User's Guide)

Understanding How to Build Fluid Applications

PeopleSoft Fluid applications are built in large part the same as you build a traditional or classic PeopleSoft application, using the same general application development steps in Application Designer. You create fields, records, and build SQL tables just as you would a traditional PeopleSoft application, but when it comes to designing pages and configuring components, you depart somewhat from the steps and techniques used for building traditional PeopleSoft applications.

The following table contains the traditional PeopleSoft application development steps as they pertain to developing fluid applications.

Development Step	*Fluid Consideration*
1. Design application.	Review your in-house development cycle and documentation.
2. Create fields.	Create fields just as you would for a traditional PeopleSoft application. In most cases, the same fields that already exist would be used for any fluid development projects.
3. Create record definitions.	Create records just as you would for a traditional PeopleSoft application. In most cases, the same records that already exist would be used for any fluid development projects.
4. Build SQL tables.	Build SQL tables using the same techniques as for traditional PeopleSoft applications.
5. Create pages.	While you still use Application Designer to create the fluid page definitions, the approach, the layout and controls used are completely different than what is used for traditional PeopleSoft page development. CSS is used heavily for appearance and layout. JavaScript can be used, if required.
6. Create components.	Components used for fluid pages must be configured specifically for that purpose. The fluid component options must be selected and set for fluid pages within that component to display in fluid mode.
7. Register components.	Classic and fluid applications both need to have the components registered using the registration wizard.
8. Test the application.	Use emulators, cellular phones, tablets, and laptops.

Note: The majority of this documentation focuses on creating pages and creating components for fluid display.

Application Designer is the main development tool used for building both traditional and fluid PeopleSoft applications, and PeopleCode is the main programming language used to interact with component processing events and implement business logic. However, with fluid applications, the application implementation teams that will develop and\or customize PeopleSoft fluid applications will need expertise in these additional areas (listed in order of importance and recommended experience):

• CSS 3

• HTML 5

- JavaScript

Important! With the flexibility provided by supporting creating and customizing HTML, CSS style sheets, and JavaScript, it is up to the developer to determine the amount of development work performed outside of the realm of PeopleTools. For example, PeopleTools cannot implement restrictions on what a developer adds to a freeform style sheet or to a JavaScript program. Developers customizing such elements need to be aware of performance and security implications.

Note: After reading this document, your application documentation, and any related information related to PeopleSoft Fluid User Interface, one very effective way to learn about fluid applications is to use the fluid applications delivered with your PeopleSoft application as examples. Consider and explore the intended audience, the capabilities of the pages, the page design, the page controls used, the component contents and settings, and so on. Especially, pay close attention to how CSS style classes are applied.

Considerations for PeopleSoft Fluid Application Implementation

Not all PeopleSoft application pages will be delivered for the PeopleSoft Fluid User Interface. The majority of PeopleSoft applications, components, and pages will, for the foreseeable future, continue to be delivered and accessed in classic mode, which is the way PeopleSoft applications have been accessed using desktops and laptops for over a decade. PeopleSoft application teams will be offering selected applications for fluid deployment that your organization can choose to enable and deploy.

In most cases, there is no quick and simple method for converting an existing custom classic application to a fluid application. Plenty of time should be spent analyzing requirements and discussing design options and considerations. In some cases, adjusting a classic page following the fluid approach to page design may be all that's required, while in other cases, you may consider building a new application from the ground up, perhaps reducing the number of fields and records required, and streamlining the entire data model.

Converting an application to fluid and bringing with it *all* the classic application techniques and elements may result in poor performance and usability on a mobile device. While some of the structural elements of the application must be analyzed and adjusted for fluid application deployment, such as page layout, some elements, such as PeopleCode business logic, can be used in either classic or fluid mode with little, if any, modification.

Before you consider deploying any PeopleSoft applications in fluid mode, it is important to consider the following items, which may help you to determine if fluid applications are appropriate for your user base at this time.

Consideration	Discussion
Not all applications (currently) are candidates for fluid deployment.	Currently, the applications that are most suited for fluid deployment are those that are already considered streamlined, self-service type applications. For example, accessing a page to check a new hire's job status would be a more appropriate fluid application as opposed to accessing a component with a cell phone to complete a new hire's "add employee" business process.

Consideration	Discussion
Are mobile devices such as cell phones and tablets used widely to access PeopleSoft applications at your site?	If your user base already accesses or has expressed a demand for accessing your PeopleSoft applications using mobile devices, fluid deployment would be a good option for you. However, if mobile devices are not part of your user base or application delivery strategy, you can elect to disable fluid deployment at the system level.
Application usage and workload?	Pages where there is heavy data entry required would not be suitable for phones or even tablets. The functional use of the page must be paramount.
Does an application *require* functionality that does not exist in the fluid user interface?	For example, some features, like spell check, rich text editor, Find In for grids, Download to Excel, modal movement or re-sizing, are not available in fluid.

Chapter 3

Creating Pages for PeopleSoft Fluid Applications

Selecting Page Types

When developing applications for fluid mode, you can create multiple page types for simultaneous display, each page serving a different role in accordance with HTML 5 specifications. The main page is referred to as the fluid page, which is a standard page that has been designated a Fluid Page on the Use tab of the Page Properties. The fluid page definition contains the bulk of the transaction fields and controls. For more information on setting page properties for fluid pages, see Setting Properties for Fluid Pages.

When creating a new fluid page or opening an existing fluid page, you can select *Page (Fluid)* from the definition list in Application Designer.

Image: New Definition dialog box

The following example shows the Page and Page (Fluid) definition types in the definition list in Application Designer.
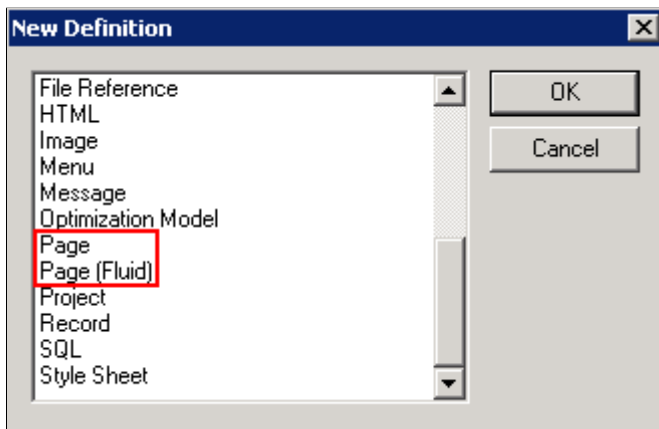
You select the remaining page types for fluid mode, from the Page Type drop-down list on the Use tab of the Page Properties dialog.

Image: Page Properties dialog box

This example shows the fluid page types listed in the Page Type drop-down list on the Page Properties, Use tab.
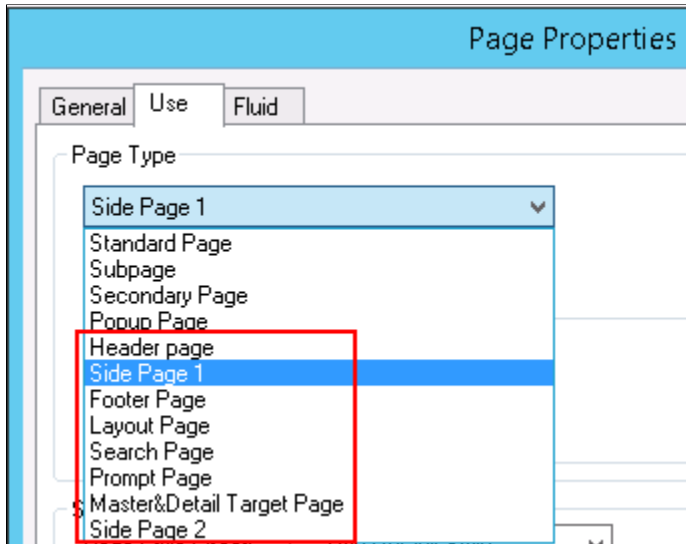
The following table provides a brief description of each fluid page type.

Page Type	Description
Header Page	Page displayed in <header> section of the HTML acting as the banner area fixed at the top of every page. It does not scroll with the page). The header page appears at the top of the browser interface over the main page, containing elements used for basic navigation and orientation, like a back button, page title, home button, search button, and so on. PeopleTools supplies a default header page, PT_HEADERPAGE, but this can be overridden for a custom header page to be used by selecting No System Header Page in the Component Properties dialog box.
Side Page 1	Page displayed on the left side off the screen, often containing left-side navigational constructs. Users can slide it open and shut.
Footer Page	Page displayed in the <footer> section of the HTML at the bottom of the main page, containing elements related to the end of a transaction, such as a Save button. The footer page is fixed at the bottom of the content page and does not scroll.
Search Page	Page generated in <aside> section containing search pages, which include Find an Existing Value and Keyword Search. **Note:** An Add a New Value page is not provided by the search page definitions provided by PeopleTools.

Page Type	Description
Layout Page	You can designate a page definition as a layout page for use as a template for development reuse or for testing purposes, such as finalizing CSS work. **Note:** This is a base layout page, which can be used when selecting a new Page (Fluid). At that time you can select a layout page where the structure will be copied into your new page. You cannot apply a layout page after you have created a page or if you create a new a new page (regular).
Master&Detail Target page	Displays the transactional fluid page for use within a master/detail component, in which the master provides the navigation and the detail displays the target content. For more information on the master/detail approach, see Working with Master/Detail Components.
Side Page 2	A side page that appears from the right side of the screen. Users can slide it open and shut. Typically used to contain related content.

Image: Fluid Screen Layout

The following diagram illustrates the *general* layout of the various fluid page types.
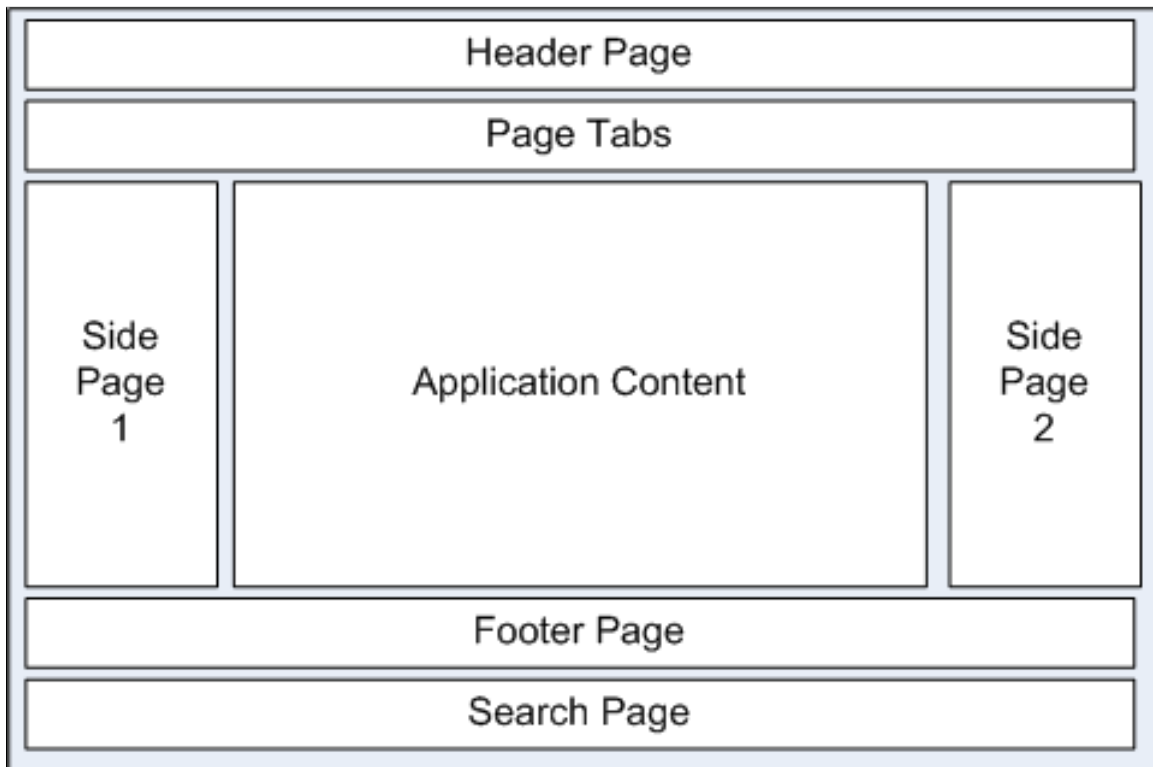
Note: Not all page types are required to be used. Page Tabs is not a page type, however, in this diagram it represents the area where page tabs will display by default, if used. The Search Page, if used, will consume the entire screen, initially, when it is being used to display search results. The Application Content area displays the content presented by the primary fluid page.

Creating the Primary Fluid Page

The fluid primary page is the main content page, or target page. The primary page is used directly within the component and is not modal (unless used in the context of DoModalComponentPopup).

Creating a New Primary Fluid Page

To create a new primary fluid page based on an existing layout template:

1. In Application Designer, select File, New.

2. Select Page (Fluid) on the New Definition dialog box, and click OK.

3. Select an initial layout page (template) from the Choose Layout Page and click Choose.

    See <u>Choosing a Layout Page</u>.

4. Enter the page name on the Save As dialog box.

5. Indicate whether you'd like to save the PeopleCode associated with the layout page for your new page (Yes/No).

6. Define layout and add page controls according to the fluid techniques and guidelines described in this documentation.

Cloning and Existing Page for Fluid Deployment

To create a primary fluid page from an existing standard page:

1. Open the existing standard page in Application Designer.

2. (Recommended) Select File Save As, and rename the page to distinguish it from the standard, classic PIA page.

    Note: Generally, for any fluid page, there would be an original counterpart, unless the page or application is new, and intended only for a mobile, fluid deployment.

3. Define layout and add or remove page controls according to the fluid techniques and guidelines described in this documentation.

4. Access the Page Properties dialog box, select the Use tab, and enable the Fluid Page check box.

    Note: For a classic page we define a visual layout with Application Designer. With fluid pages, you design HTML structure, where styles are applied to implement visual changes to that structure.

5. Apply CSS style classes as needed to arrive at the appropriate look and feel for your page.

Choosing a Layout Page

This section describes the layout pages you can use as the basis for any new fluid pages. Layout pages can be compared to templates. They provide a basic structure to get you started with a fluid page.

Image: Choose Layout Page Dialog Box

This example illustrates the layout page options provided by the Choose Layout Page dialog box.
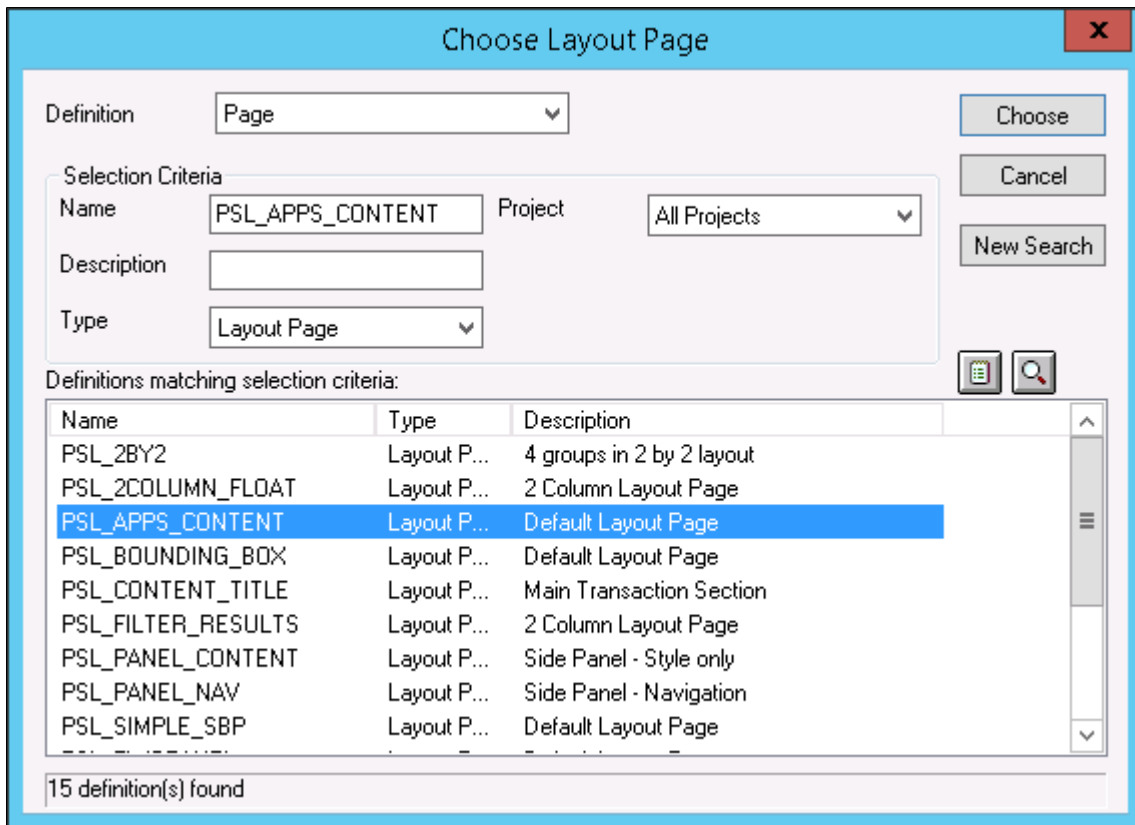
Select a template, and save the page with a name other than the template name.

You may add or remove elements as needed for your purposes. Make sure to examine the template pages so that you understand how any existing PeopleCode or style application affects the interface.

Important! Not all the layout pages displayed are intended for external usage.

Recommended Layout Pages

This table describes the layout pages suitable for custom fluid development projects.

Layout Page	Description
PSL_APPS_CONTENT	The basis for any fluid page. Contains the outermost group box to act as the overall container for the fluid page content. This group box has the *ps_apps_content* style applied, which provides the overall flexbox model approach for fluid applications.

Layout Page	Description
PSL_2COLUMN_FLOAT	Contains two columns inside of the ps_apps_content group box. Sets up basic container for a two-column container, having psc_columnitem* as children. If using, you need to supply the styling for the child containers based on your needs.
PSL_CONTENT_TITLE	Page used for the content area of a component which also has a panel (or content) title. This is typically needed for two panel usage or for optimization for use within an activity guide.
PSL_PANEL_CONTENT	Single container having the default styling for the side panel (left or right panel) usage. If using this template, you also need to specify your own record/field and make sure the appropriate role (ARIA) is applied to the outer container for accessibility.
PSL_PANEL_NAV	Left (typically) panel usage to contain navigational elements, which control the main region content. Role (ARIA) automatically applied to indicate that the contents will be navigational in nature.
PSL_TWOPANEL	Used in PeopleTools 8.54 to provide a two-panel layout, typically with navigation on the left side and the transaction page on the right.
PSL_USER_PREF	Use for creating user preferences, such as with My Preferences.

Layout Pages Reserved for Internal Usage

These layout page types *are reserved for internal usage* by Oracle development teams:

- PSL_2BY2

- PSL_BOUNDING_BOX

- PSL_FILTER_RESULTS

- PSL_SIMPLE_SBP

- PTPG_GRIDVIEWERNUI

- PTPG_NUI_VWR

- PTPN_SEEALL

- PT_LANDINGPAGE

Setting Properties for Fluid Pages

You create fluid pages using Application Designer and use the same Page Properties dialog box to set the properties for a fluid page. This topic describes the settings that are specific to fluid pages. For general page development information refer to the product documentation for Application Designer, the *Application Designer Developer's Guide.*

Related Links

"Understanding Page Design and Development" (PeopleTools 8.55: Application Designer Developer's Guide)

"Creating New Page Definitions" (PeopleTools 8.55: Application Designer Developer's Guide)

Setting Use Properties

On the Use tab in the Page Properties dialog box, some properties need to be described separately in the context of the fluid application (not classic PIA).

Image: Page Properties > Use tab

This example shows the Use tab in the Page Properties dialog box in Application Designer. The following descriptions cover the properties that are specific to fluid pages.
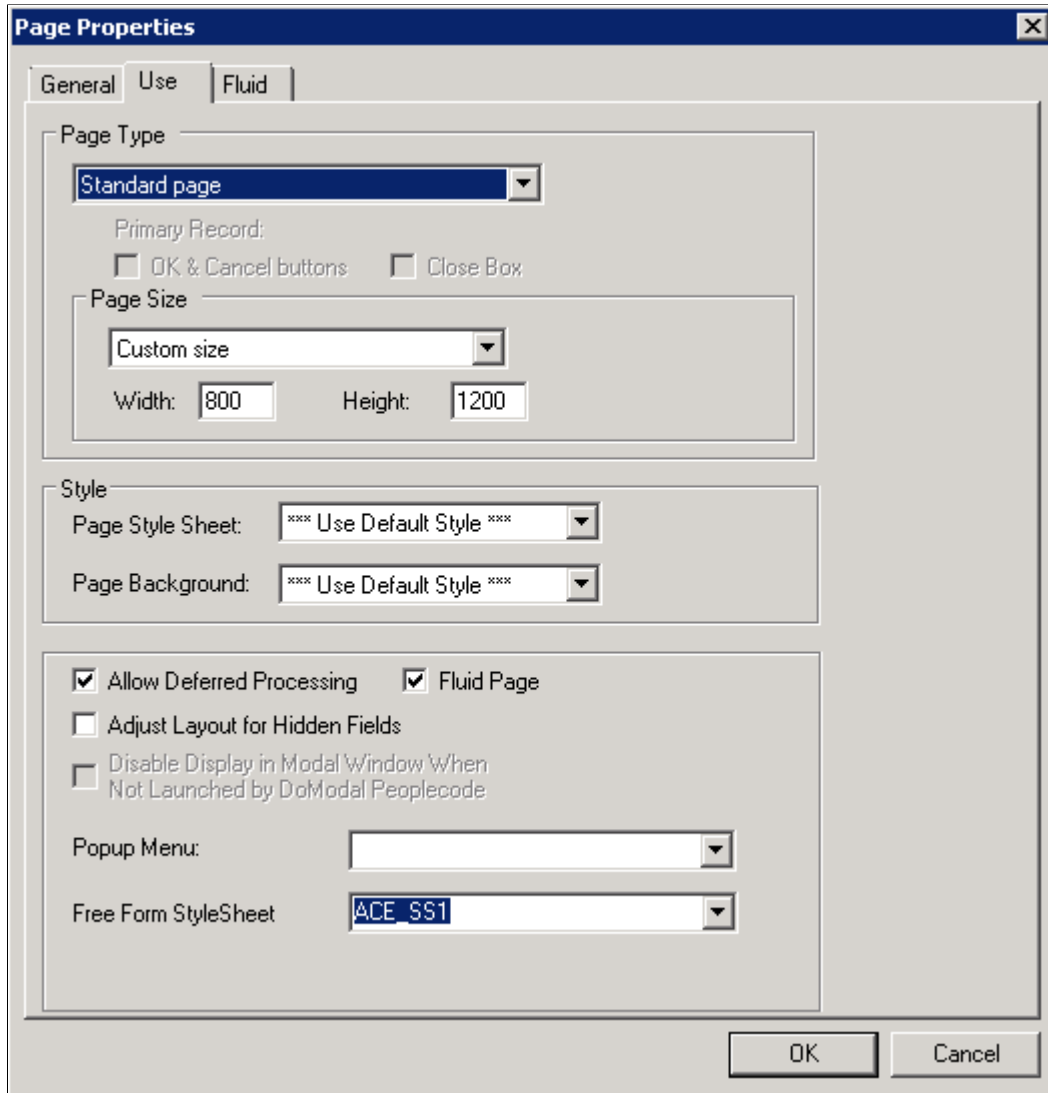
Page Type	Select the type of page for use in the fluid application, such as header, footer, search, and so on.
	For more information on page types used for fluid development, see Selecting Page Types .
Fluid Page	For a page to be displayed and rendered in fluid mode, the Fluid Page check box *must* be selected. This is required.

Setting Fluid Properties

The Fluid tab contains a collection of properties that apply only to fluid pages.

Image: Page Properties > Fluid tab

This example shows the Fluid tab in the Page Properties dialog box in Application Designer. The following descriptions cover the properties that are specific to fluid pages.
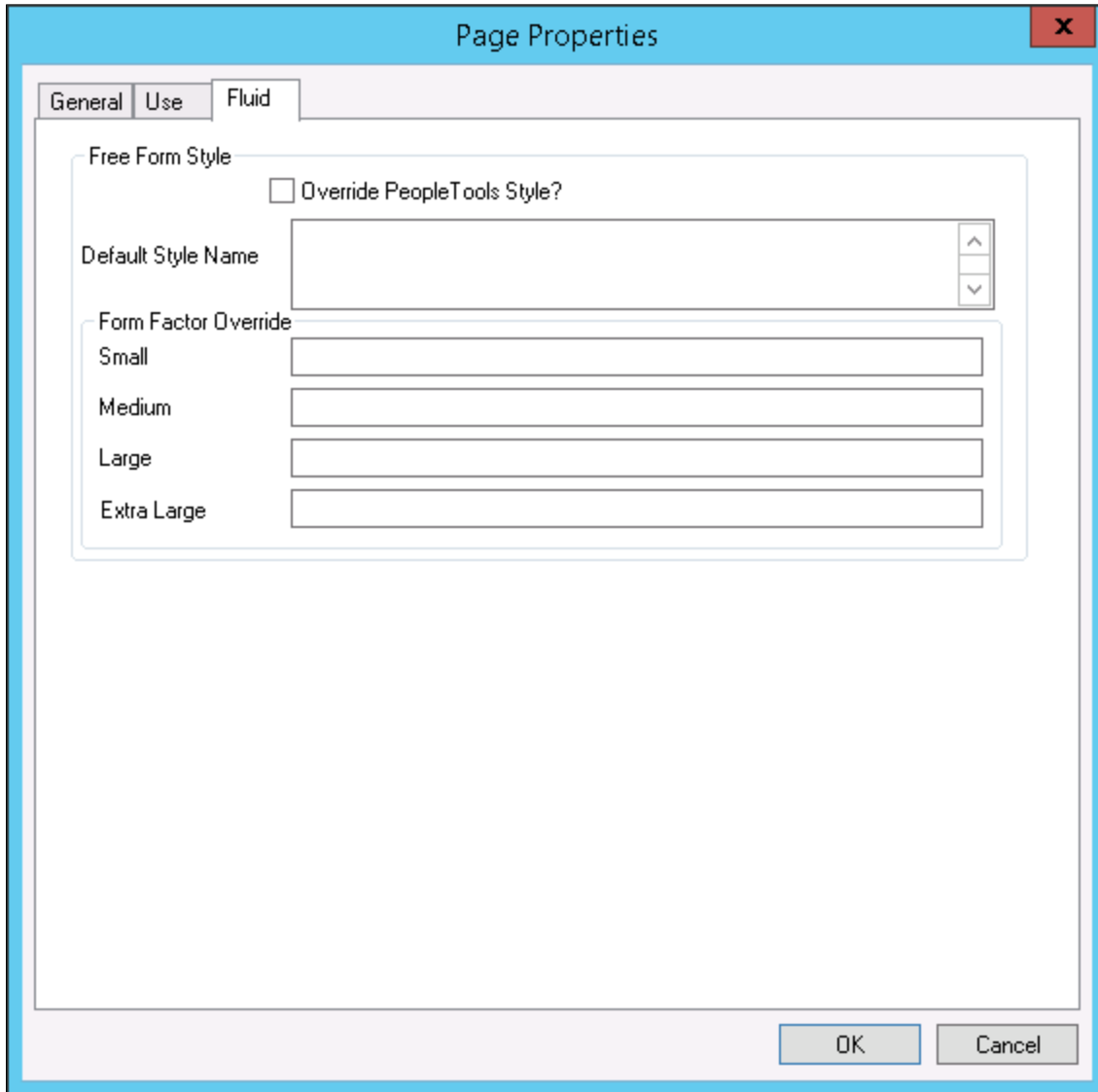
Free Form Style

Override Tools Style	Select to bypass the system default style completely.
	If not selected, the custom free form style specified will be appended to the default system style, such that both style sheets are used.
Default Style Name	Enter the custom default style name to include.
Form Factor Override	Use to override the style based on the form factor (device size) of the device accessing the page. Form factors are represented in these ranges:

- Small: Applies to most smartphones.

- Medium: Applies to 7-inch tablets.

- Large: Applies to 10-inch tablets and laptops.

- Extra Large: Applies to large monitors.

Defining the Layout of the Main Fluid Page

Designing and defining the layout of a fluid page is one of the major differences between developing fluid applications and developing classic PIA applications. In a classic PIA application, you perform layout of a page in a WYSIWYG format, where page elements and controls are placed exactly where you want them within a pixilated invisible table.

Image: Classic page definition at design time

This example illustrates a classic PeopleSoft page definition in Application Designer, where each page element is meticulously placed on the Layout tab.
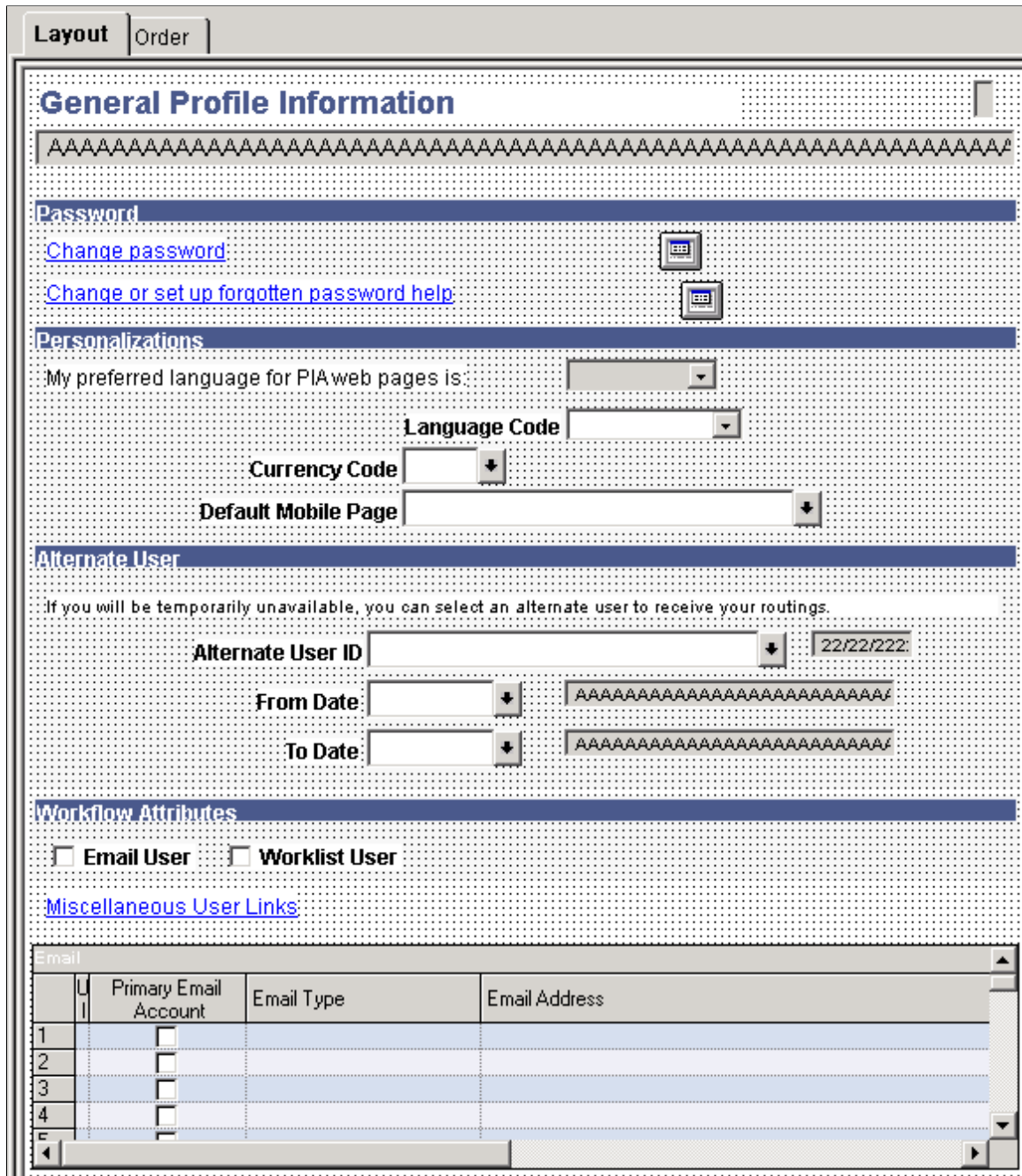
And, despite the obviously more polished presentation in the generated HTML page that displays in the browser interface for the end user, the layout, spacing, order, color, and so on is for the most part identical between the design time definition and the runtime presentation.

Image: Classic page at runtime

This example illustrates the same page as depicted in the previous example at runtime. The design time and runtime appearance is practically identical in terms of layout, positioning, sizing, color, font, and so on.

When performing layout for a fluid page, you will notice that design representation in no way resembles the design representation for the classic PIA page.

Image: Fluid page at design time

This example shows a fluid page layout at design time in Application Designer, with various elements contained in nested group boxes and little if any indication regarding how the page might appear at run time to the end user.
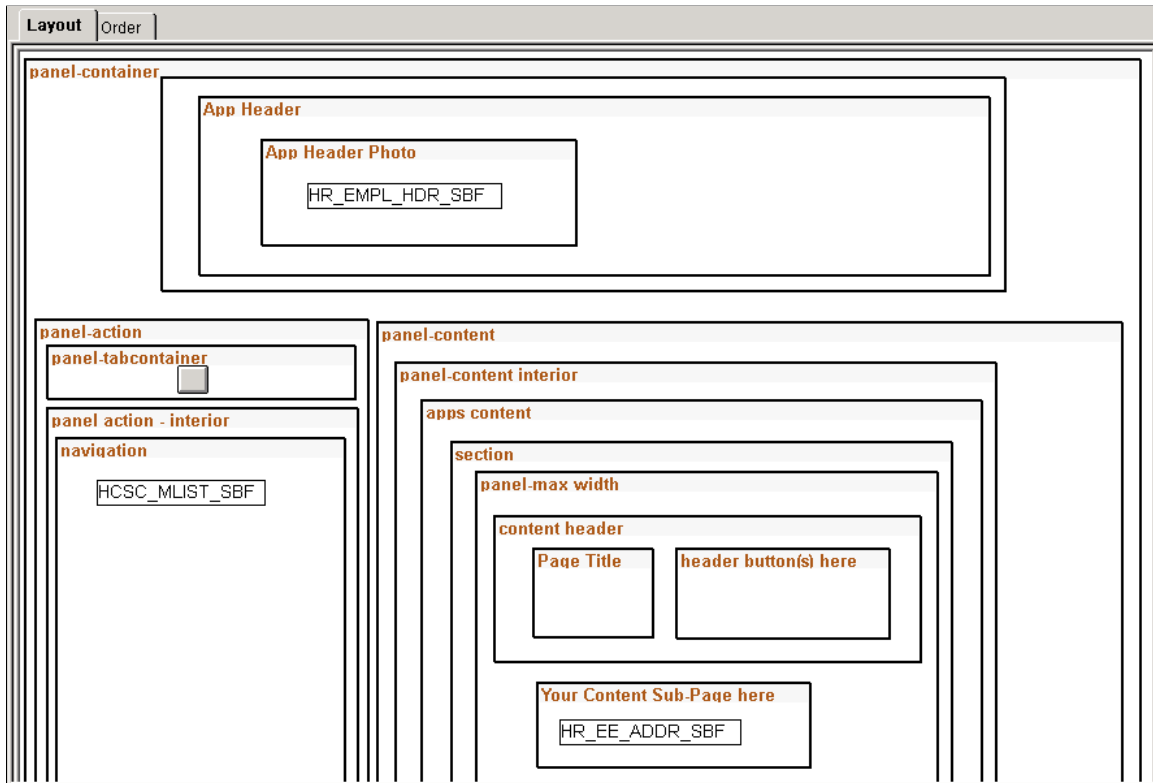
The layout of the fluid application is not what you would call a WYSIWYG approach. If an ender user were to access a page in this state, it would be impossible to complete a transaction. Notice the boxes that are used to indicate levels of containment and to provide visual cues, making each element from high-level to low-level a self contained and discrete. Group boxes are used to achieve this effect. The rectangular boxes within some of the group boxes represent subpages.

Image: Fluid page at run time

This example illustrates the fluid rendering of this very same page definition (above) in a browser.
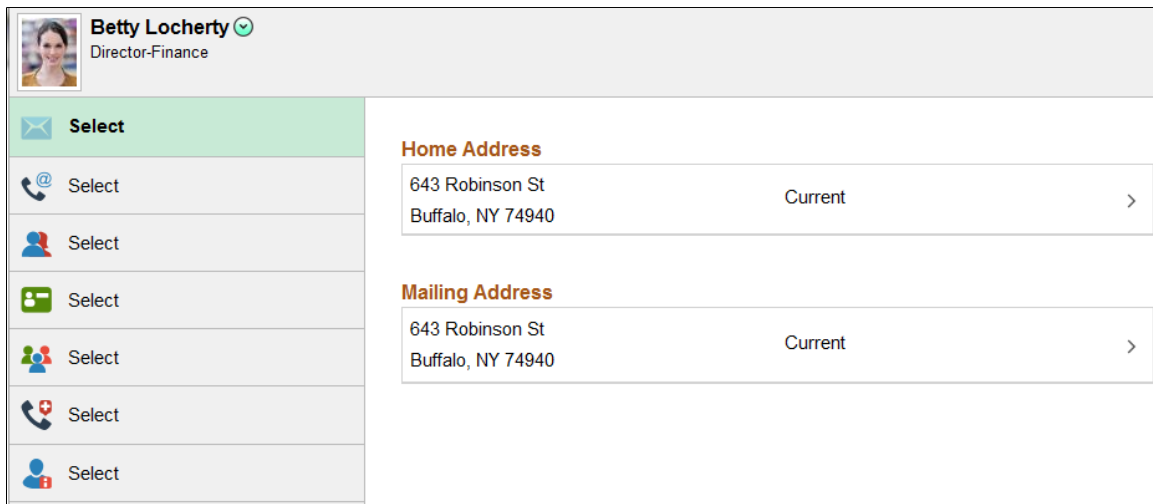
The strip running across the top displays the default PeopleTools, or system, header page definition. Because it is a separate page definition entirely, it is not depicted in the layout view of the main page (the content below it) in Application Designer.

At a high-level, you can map the rendering in the browser to the layout as:

• The area showing the employee photo is represented in the layout as the App Header Photo group box.

• The area showing the navigation on the left is represented in the layout as the Left Side Navigation group box.

• The area showing the Name and Addresses fields is represented in the layout as the Content group box. When Names and Addresses is selected from the left hand navigation, it appears in the Content container. When Contact Details is selected, the contact information is displayed in the Content container, and so on.

While there are many factors contributing to the appearance of a fluid page, which will be discussed in this document, the main elements of fluid page layout are:

PeopleSoft Design Elements	*Description*
group boxes	Group boxes are the primary control, providing not only visual cues, but, more importantly, it acts as the container for *all* controls and page elements. For example, subpages must be enclosed in a group box, grids and scroll areas need to be enclosed in a group box, and so on. Each control is comprised of two parts: the label and the control – wrapped in the container of a group box. In fact, the entire page itself *must* be contained in one super group box: a requirement for fluid rendering.

PeopleSoft Design Elements	Description
style sheets	While group boxes take on the role of discretely containing each and every element on a page, the physical placement of the group boxes within the Layout tab in Application Designer is largely irrelevant. How and where a page element or control renders on the end user device is determined by style sheets. PeopleTools delivers the system-wide PSSTYLEDEF_ FMODE style sheet. You can make custom, free form style sheets to extend or override the system-wide style sheet.

Both group boxes and free form style sheets are discussed in detail in other sections of this document.

Note: Tab order index is determined only by the order in which the fields and controls appear on the page from top-to-bottom and left-to-right.

Chapter 4

Defining Components for Fluid Applications

Adding Fluid Pages to a Fluid Component

Like classic PIA applications, the fluid application uses components to contain sets of pages. With all the different page types that are included within a single fluid transaction (header pages, footer pages, primary pages, and so on) they must all be included within the same component so they run in the same panel buffer. Having these pages all in the same panel buffer means you can write PeopleCode to work on all page types at the same time.

Note: When using the default system pages, as in the default header page provided by PeopleTools, the system automatically loads those pages into the component. If you intend to use those pages, they do not need to be added manually to the component. However, if you have built custom header pages, search pages, and so on, those pages need to be inserted manually into the component to associate them with that component so the system loads them at runtime.

Setting Component Properties for Fluid Components

This section describes component properties from a fluid perspective.

Setting Fluid Component Properties

Use the Component Properties, Fluid tab to set fluid component properties.

Image: Component Properties dialog box: Fluid tab

This example illustrates the options on the Fluid tab. You can find descriptions for the options after the example.
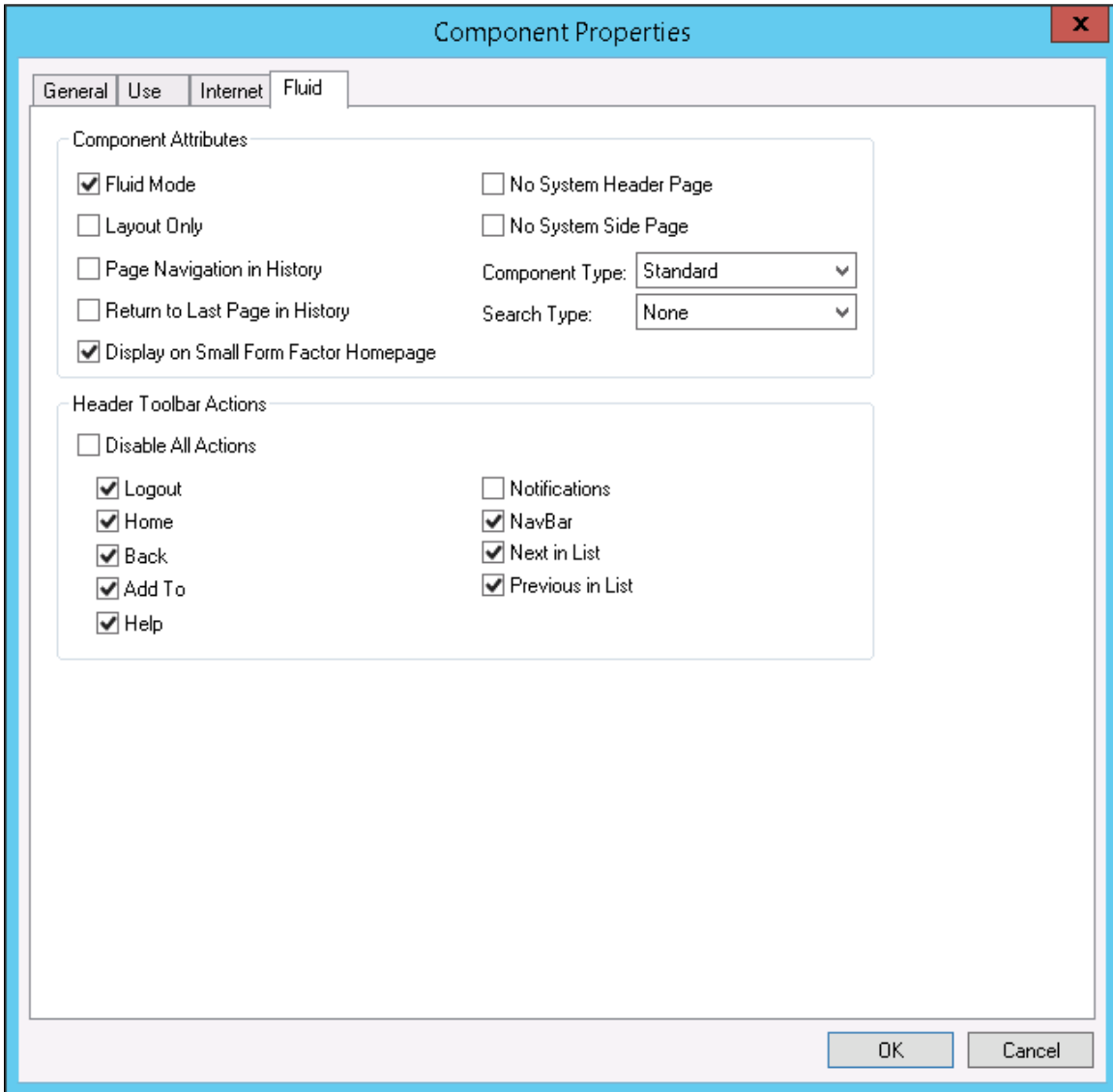
Component Attributes

Fluid Mode

Indicates that the pages in this component are to be accessed in fluid mode. This is *required* for fluid rendering.

Layout Only

Select when testing the layout of the pages within the component. When selected the borders of all the group boxes on the page appear in different colors to illustrate the scope of each group box container.

Page Navigation in History

Enables the system to store the visited pages in the Back button history. This setting applies to the pages within the current component.

This can be enabled also using the SetTransferAttributes PeopleCode built-in function.

For more information, see "SetTransferAttributes" (PeopleTools 8.55: PeopleCode Language Reference).

Return to Last Page in History   Select to return the user to the last viewed page within a component after being transferred to a different component. This setting applies in the event of a transfer between components.

The default behavior when using the Back button to return from a transfer between components is to display the initial page of the component, regardless if it was the last viewed page.

For example, with Return to Last Page in History selected, if the user was on the third page within the component prior to the transfer to another component, the system displays the third page in the original component after using the Back button to return from the transfer.

This can be enabled also using the SetTransferAttributes PeopleCode built-in function.

For more information, see "SetTransferAttributes" (PeopleTools 8.55: PeopleCode Language Reference).

Display on Small Form Factor Homepage   Select to enable the tile for this component to be displayed on fluid homepages and fluid dashboards on small form factor devices such as smart phones.

If Display on Small Form Factor Homepage *is not* selected, the tile for that component *will not appear* on fluid homepages and fluid dashboards when accessed by a small form factor device.

If a fluid homepage does not contain *any* tiles for components with Display on Small Form Factor selected, that fluid homepage *will display as empty* on the small form factor device, if your PeopleTools version is greater than or equal to 8.55.09.

If your PeopleTools version is less than 8.55.09, the system *does not display empty fluid homepages* on the small form factor device. If *none* of the fluid homepages available to a user contain tiles for components with Display on Small Form Factor selected, the system routes the small form factor device for that user to the classic portal homepage.

In general, if you intend to provide access to your PeopleSoft system for small form factor devices, at least one of the fluid homepages to which an end user has access should contain at least one tile for a component with Display on Small Form Factor Homepage selected.

No System Header Page	Select if you have a custom header page that you want to display instead of the default, system header.
No System Side Page	Select if you have a custom side page that you want to display instead of the default, system side page.
Component Type	Select the type of search the type of component you are building:

- *Standard.* Applies to most fluid components used for serving fluid application pages.

- *Activity Guide.* A special type of component aggregating a procedure or activity that spans multiple components. See "Understanding Activity Guides" (PeopleTools 8.55: Portal Technology).

- *Master/Detail.* A special type of component, similar to an activity guide, that is used to provide access to multiple components, aggregating the navigation to the various components in a master list. A master/detail component enables you to create a means of navigating easily amongst a collection of components sharing a common theme, such as those most likely to be accessed regularly by a particular role. See Working with Master/Detail Components for more information.

Search Type	Select the type of search the user will interact with when accessing the component:

- *None.* Disables search pages for the component.

- *Standard.* Displays a "standard" search page, which provides end users with search results that they click, and drill into component pages and data.

- *Master/Detail.* Displays another type of search page, which provides end users with search results that they click, and drill into component pages and data, however, the search results remain active and visible in the left side panel, while the application page displays in the right panel.

For more detailed information on these options and how they are configured, see Working with Search Pages.

Header Toolbar Actions

Disable All Actions	Select to deselect all header toolbar actions. At runtime the system displays a header page toolbar containing no buttons.
Logout	Select for the Logout button to appear on the header toolbar.

	Use to logout, or sign off, of the application, returning the user to the fluid login page.
Home	Select for the Logout button to appear on the header toolbar.
	Returns the user to the default fluid homepage, as if they've just signed on.
Back	Select for the Back button to appear on the header toolbar.
	Returns the user to the previous location.
	Note: If you switch languages during the middle of a PeopleSoft session (such as during development or testing), the Back button will continue to display the original language. The current language for a session can be switched using PeopleTools, Utilities, International, Preferences.
Add To	Select for the Add To button to appear on the header toolbar.
	Enables a user to add a link to the component to the homepage or the Navbar or Favorites.
Help	Select for the Help option to appear in the Actions List for the fluid pages in this component. The system displays the PeopleSoft Online Help in a new browser page for all form factors. The Help URL value must be added to the General tab of the web profile to enable online help (same as for classic components and pages).
	Note: In fluid mode, you cannot launch online help from modal pages.
	Note: As with classic mode, you cannot launch online help from workcenter or dashboard pages.
	Note: For the Safari browser, running on iOS in full screen mode, the PeopleSoft Online Help is displayed in a a full modal page.
	For more information on configuring online help URL for your PeopleSoft system, see "Configuring General Portal Properties" (PeopleTools 8.55: Portal Technology).
Notifications	Select for the Notifications button to appear on the header toolbar. Enabled by default.
	The Notification button shows the list of notifications for the logged in user, such as events and notifications from AWE.
	Use this option to disable notifications while within a specific component (exception to rule and also allows debugging).

	The banner uses this option to determine whether to hide or show the notification button and to enable the appropriate JavaScript.
NavBar	Select for the NavBar button to appear on the header toolbar.
	Note: For small form factors, personalizing the NavBar is not supported.
Next in List	Select for the Next in List button to appear on the header toolbar.
	Takes the user to the next search result on a search page.
Previous in List	Select for the Previous in List button to appear on the header toolbar.
	Takes the user to the next search result on a search page.

Setting Component Internet Properties for Fluid Components

Use the Component Properties, Fluid tab to set *some* fluid component properties.

If a component is a fluid component, then most of the properties on the Internet tab of the Component Properties dialog box *do not* apply. For example, none of the Toolbar actions apply.

However, these properties on the Internet tab *do apply* to Fluid components:

• Primary Action (Search Pages).

• Display Folder Tabs.

• Process Mode options (except for Allow Expert Entry).

For more information on the Internet tab off the Component Properties dialog box, see "Setting Internet Properties" (PeopleTools 8.55: Application Designer Developer's Guide).

Note: In the current release, spell check is not supported for fluid mode. Do not enable spell check at the component level or at the field level for fluid applications.

Providing Left Navigation Using PT_SIDE_PAGETABS

Some fluid pages are the only page within a fluid component. In which case, they are standalone, and once an end user arrives at them, there is obviously no need to provide additional navigation devices within the component. However, other fluid components may have multiple pages within a component. You then need to consider how you would like users to navigate through the component.

By default, similar to classic PIA, the page tabs appear across the top of the interface, always visible to the end user.

Image: Fluid Component: Page Tabs on Top

This example illustrates the default appearance of page tabs in a fluid component. They appear across the top of the main transaction page, similar to the classic PIA interface.
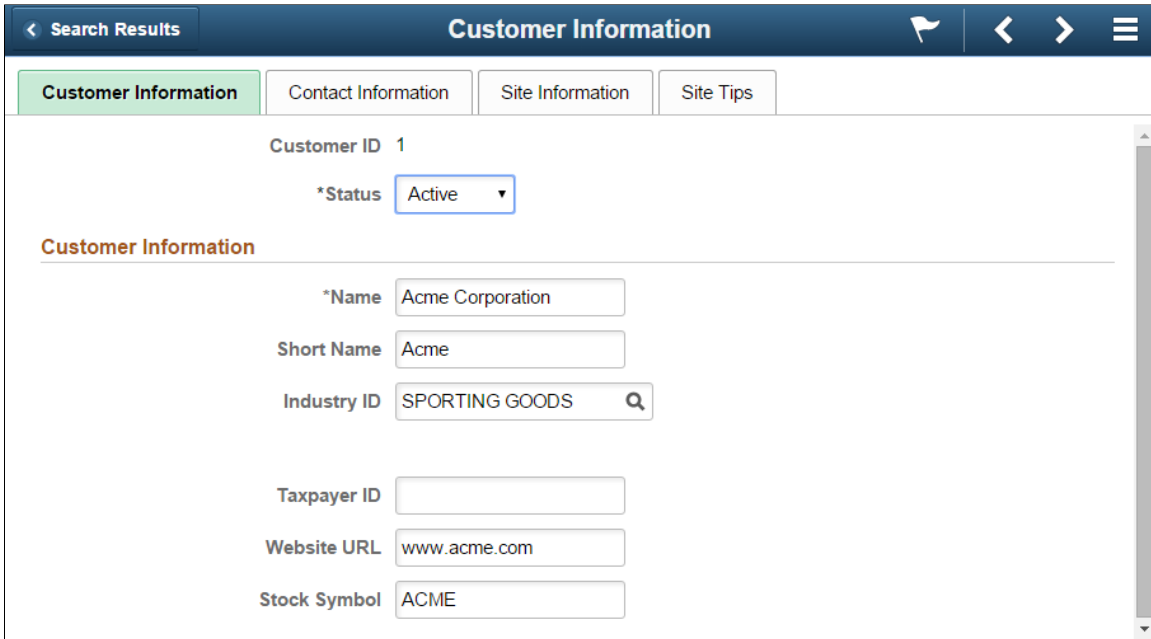
Image: PT_SIDE_PAGETABS Added to Fluid Component

This example illustrates adding the delivered PT_SIDE_PAGETABS page definition to a fluid component. No other configuration, other than inserting this page definition into the component, is required to achieve left navigation.
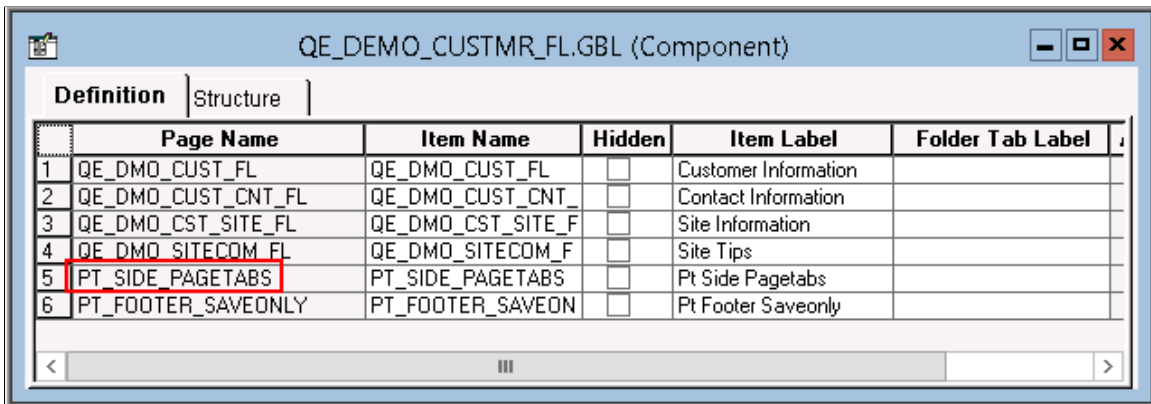
Image: Fluid Component: Page Tabs on Left Panel

This example illustrates the effect of inserting PT_SIDE_PAGETABS into the fluid component. The page tab navigation appears on the left side panel, which end users can slide in and out when needed.
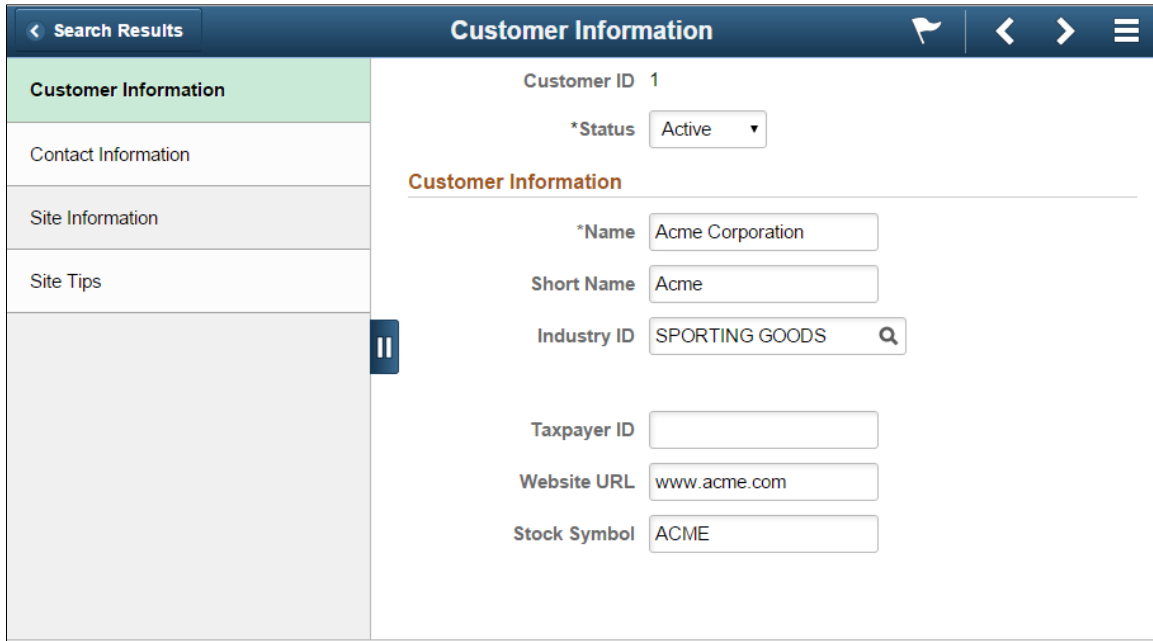
Working with Search Pages

This topic discusses setting up search pages for fluid components.

Understanding Component Search Options for Fluid Applications

For a fluid component, you have these options for providing search when a user accesses the component:

Fluid Component Search Option	Description
Real-time	Based on a pivot grid that builds a query against the component search record. See Setting up Real-Time Search for Fluid Components.
Keyword	Based on a search index created using the Search Framework and mapped to the fluid component. See Setting up Keyword Search for Fluid Components.

For any custom fluid component that you create, Oracle recommends that as part of the normal page and component development, you also create:

• A component-based pivot grid (for real-time component search).

- A search definition in the Search Framework for indexing the underlying application data (for keyword search).

A custom fluid component must employ *at least one* of the search options. You can employ both options. If neither search option is configured, you will need to bypass search completely using some other mechanism (for example, using SearchInit PeopleCode). Otherwise, the system displays an error, preventing access to the component.

When comparing one search option over the other, consider that the real-time search runs a search on all the data that is currently stored in the database while the keyword search may contain less current data. How current the data provided by the keyword search depends on how frequently the search indexes are updated. If you intend to employ only one of these search options, to ensure the best user experience, Oracle recommends employing the real-time search so the system reflects real-time data for end users.

Setting up Real-Time Search for Fluid Components

In classic PIA, the system automatically builds the component search page displayed to a user, with no additional effort required by the application developer. The system uses built-in logic to create the component search page on the fly based on the search keys defined on the search record for the component.

In the fluid user interface, the real-time component search page is *not populated* automatically by the system. For any custom fluid component that requires real-time, component search capabilities, the application developer must create a pivot grid, using the component as the data source. The pivot grid generates the underlying query using the search keys defined on the records in the component and populates the search page. See the product documentation for *PeopleTools: Pivot Grid* for creating a component-based pivot grid for use with implementing real-time search for a fluid component.

Creating a Pivot Grid for Real-Time Search

When creating a pivot grid for real-time fluid component search, keep these items in mind while using the Pivot Grid Wizard:

- Make sure the pivot grid Data Source Type is set to *Component* and the Component Name under Data Source reflects your fluid component on Step 1 of the wizard.

- Make sure the Default View is set to *Chart Only* under View Options on Step 4 of the wizard.

- Under Fluid Mode, Viewer Options on Step 4 of the wizard, select either *Grid and List* or *List* for Result View if you intend to use the Master - Detail search page type or you intend for the search results to be displayed on a small form factor. Selecting *Grid* is a valid option for the Standard search page type and if the fluid component would not be accessed by small form factor devices.

Setting Search Defaults for Real-Time Search

In a fluid component when search is enabled and an end user accesses the component, the system displays search results on the search page using the default search criteria provided by the application developer. If you do not provide default search criteria, the system displays an empty page, which may detract from the user experience.

When creating a pivot grid for real-time component search, you can provide default search criteria using these options:

- In the Pivot Grid Wizard, enter the desired values under Select Query Prompt Values in Step 3.

- In Application Designer on the Record Field > Use tab, set a Default Value for a search key in the search record.

- Use SearchInit PeopleCode to set a search record value to a particular value (record_name.field_name=value;).

Note: If you are using multiple options for setting default search criteria, the system imposes this order of precedence: SearchInit PeopleCode takes precedence over Record Field Default Value settings, and Record Field Default Value settings take precedence over any Query Prompt Values set in the pivot grid definition.

Related Links

"Understanding Component Real Time Search Using Pivot Grid in the Fluid Mode" (PeopleTools 8.55: Pivot Grid)
"Creating a Component Pivot Grid Model Using the Pivot Grid Wizard" (PeopleTools 8.55: Pivot Grid)

Setting up Keyword Search for Fluid Components

The keyword search for a fluid component is similar to the keyword search for a classic component. The keyword search runs against a search index built using the PeopleSoft Search Framework and deployed to the search engine, such as Oracle SES. The keyword search is more free-form and more "Google-like." You implement keyword search for a fluid component using the same steps as you would complete for a classic PIA component.

Task	*Documentation Cross Reference*
Create a query for the search definition.	See "Understanding Search Definition Queries" (PeopleTools 8.55: Search Technology).
Create a search definition.	See "Creating Search Definitions" (PeopleTools 8.55: Search Technology).
Map the search definition to your fluid component.	See "Mapping Components to Search Definitions" (PeopleTools 8.55: Search Technology).
Create a search category.	See "Understanding Search Categories" (PeopleTools 8.55: Search Technology).
Deploy and build your search index.	See "Administering Search Definitions and Search Categories" (PeopleTools 8.55: Search Technology).

When accessing a component with keyword search enabled, the search category mapped to that component appears selected in the Global Search box. Running a search on that category returns results only from the indexes in that category and displays them in PTS_NUI_SEARCH. If the user selects All or another search category and runs a search, the system treats the search as a global search, and the global search page (PTS_NUI_GBLSRCH) is used to display results.

Setting Component Properties for Search Options

This section discusses setting component properties as they relate to setting up search for a fluid component.

Setting Internet Properties

You can set the default search mode for the fluid component using the Primary Action property in the Search Page group box on the Component Properties, Internet tab:

- New: Add mode.

- Search: Real time component search.

- Keyword Search: Search Framework-based search on the search category associated with the component.

Note: If a fluid component is designed to provide both real-time component search and keyword search, then real-time component search should be used as the default search option. That is, the Primary Action setting on Component Properties, Internet tab: should set to *Search.*

Setting Fluid Properties

When setting properties on the Component Properties > Fluid tab, the properties in the following table relate to implementing search for a fluid component:

Property	*Description*
Search Type	- *None:* Disables a search page from appearing for a fluid component. If selected, you need to ensure you have bypassed the search requirement sufficiently, such as by using SearchInit PeopleCode. - *Standard:* Select to display the "standard" search page, derived from the search page definition: PTS_NUI_ SEARCH. Using the page, the end users view the search results, and drill into the component. - *Master/Detail:* Select to display the "standard" search page, derived from the search page definition: PTS_NUI _SEARCH_S. Using the page, the end users view the search results, and drill into the component. With the Master/Detail option, the search results remain displayed in the left side panel while the component displays in the right panel. **Note:** Standard or Master/Detail search types may be used for both real-time or keyword search options.
Next/Previous in List	Select to display features for end users to use for navigating through the search results. The Next/Previous icons will appear in the fluid banner after drilling into a component search result.

Related Links
Setting Component Properties for Fluid Components

"Setting Component Properties" (PeopleTools 8.55: Application Designer Developer's Guide)

Working with Search Pages

The search pages the system displays for classic PIA applications are not displayed for fluid applications. For each type of fluid search, component search or Search Framework-based keyword search, PeopleTools delivers a default search page. The delivered search page definitions must be the base for any custom or system search page as they contain the basic framework and structures to use as a reference. If you decide you need a custom search page for a specific fluid component, you can clone the delivered search page and add or remove elements as needed. The following table lists the delivered default search pages:

Definition Name	Description
PTS_NUI_SEARCH	Standard search page.
PTS_NUI_SEARCH_S	Master - Detail search page.
PTS_NUI_GBLSRCH	Fluid global search page. Global search is available if at least one search group has been deployed and indexed and the current user has access to that search group. Global search is not dependent on a component and can be accessed throughout the system. Users access this search type using the Global Search icon in the banner. For more information on global search, see "Working with Global Search" (PeopleTools 8.55: Search Technology).

The default system-wide search, PTS_NUI_SEARCH, is set on the PeopleTools Options page in the Fluid Mode Properties group box. It is not recommended to override the system-wide search page on PeopleTools options, as this would affect every fluid component. If you need a custom search page for your fluid application, set the Search Type to None on the Fluid tab of the Component Properties dialog box, and insert your custom search page within your fluid component so that it loads at runtime when the component is accessed.

For more information on the Fluid Mode Properties, see Configuring PeopleTools Options for Fluid Applications.

The page design of PTS_NUI_SEARCH uses the two-panel layout template, and contains numerous group box containers and subpages. If creating a custom seach page based on this template, you can remove the subpage for the feature you don't want to include on your search page. For example, if you don't want to include charts or facets, remove the PTS_NUI_FACETS and PTS_NUI_CHARTAREA subpages.

Comparing the Availability of Fluid Search Features to Classic Search Features

Due to the differences between classic PIA applications and fluid user interface applications, not every search feature that is available for classic PIA is available for a fluid application, and vice versa. Likewise, within a fluid application, not every search feature is available on every device type. For example, a search option used on a tablet or laptop, may not be available for use on a smartphone.

The following table lists the feature differences between classic PIA and the fluid user interface.

Feature	Classic PIA	Fluid User Interface
Persistent Search	Yes	No
Images in results	No	Yes
Related Actions in component search results	No	Yes
Related Actions in keyword search results	Only in List View	Yes
Bulk Related Actions	No	Yes
Conditional Global Search	Yes	Yes
Facets in component search results	No	Yes
List view in component search results	No	Yes
Chart view in component search results	No	Yes
Add a New Value	Yes	Yes

The following table lists the feature differences between various devices when accessing the fluid user interface.

Feature	Desktop/Laptop	Tablet	Phone
Related Actions	Yes	Yes	No
Bulk Related Actions	Yes	Yes	No
Chart interaction in component search	Yes	Yes	No

Note: The features by device table contains only those features where differences occur on one or more devices. For any feature not appearing, it can be assumed that feature is available on all supported devices.

Working with Master/Detail Components

This topic provides an overview and describes how to implement a fluid application using the master/detail model.

Understanding Master/Detail Components

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. Typically, the components and pages accessed from within master/detail components are not necessarily presented sequentially, as in there is no implied procedure or sequence for accessing the pages. Without the use of the master/detail construct, users would navigate to these components and pages separately as needed.

Example of a Master/Detail Application

Image: Master/detail example: Accounts Payable WorkCenter

This example illustrates a master/detail application, the Accounts Payable WorkCenter. The left panel provides navigation while the target content is displayed on the right.

Understanding the Elements of Master/Detail Components

This section describes the essential elements of master/detail components.

Image: Master/detail elements

This graphic illustrates the main elements of master/detail components.

- The master/detail component definition - PT_MDSTARTPAGE_NUI in the example.

- A side page definition, which will display the navigation elements - PT_MDSIDEPAGE in the example.

- A target page definition, which will display the target content - PT_MDSTARTPAGE_NUI in the example.
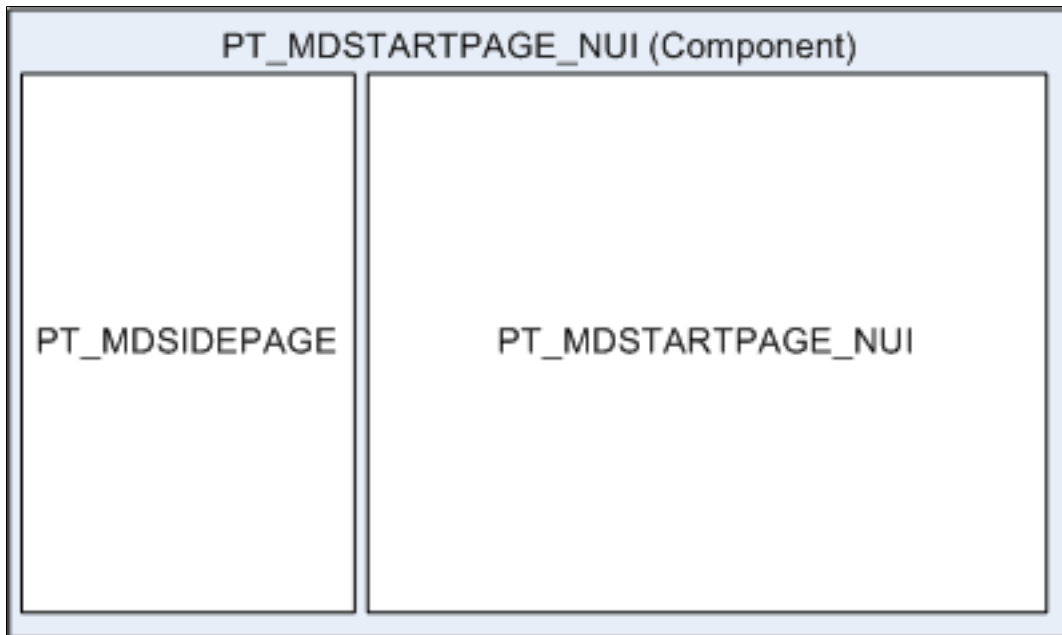
Master/Detail Component

The master/detail component has its component type set to *Master/Detail* on the Component Properties: Fluid tab. This component contains two page definitions: a target page for the details and a side page for the master list.

PeopleTools provides a default master/detail component: PT_MDSTARTPAGE_NUI that you can clone and modify or use as is.

Image: PT_MDSTARTPAGE_NUI component

This example illustrates the PT_MDSTARTPAGE_NUI component definition.

	Page Name	Item Name	Hidden	Item Label	Folder Tab Label	Allow Deferred Processing
1	PT_MDSTARTPAGE_NUI	PT_MDSTARTPAGE_NUI	☐	Detail		☐
2	PT_MDSIDEPAGE	PT_MDSIDEPAGE	☐	Master		
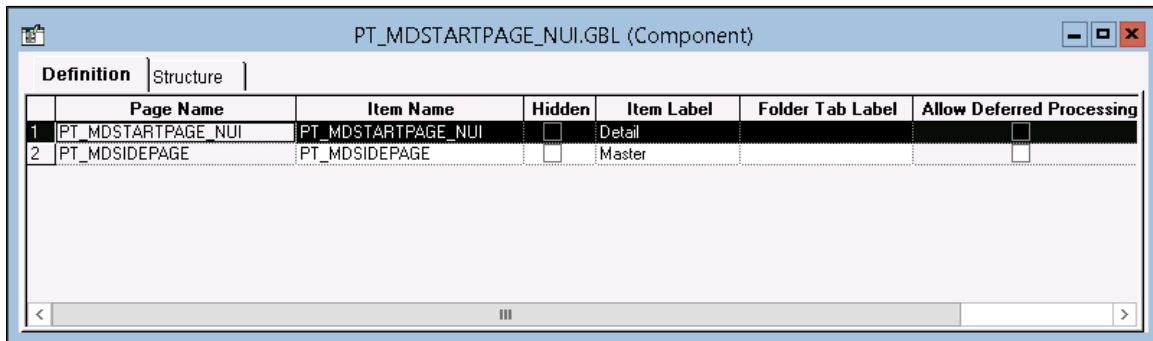
Master/Detail Side Page

The master/detail side page provides the left navigation in the form of a side panel. The side page acts as the *master,* controlling the navigation, whereas the target page acts as the *details,* displaying when called by the master. The master/detail side page has page type set to *Side Page 1* on the Page Properties: Use tab.

PeopleTools provides a default master/detail side page: PT_MDSIDEPAGE that you can clone and modify or use as is.

Image: Master/detail side page

This example illustrates the PT_MDSIDEPAGE definition.

The inner group box (selected in this example) will display the master list to the user. The group box type is Layout Only with this style applied: *ps_masterlist-group*.
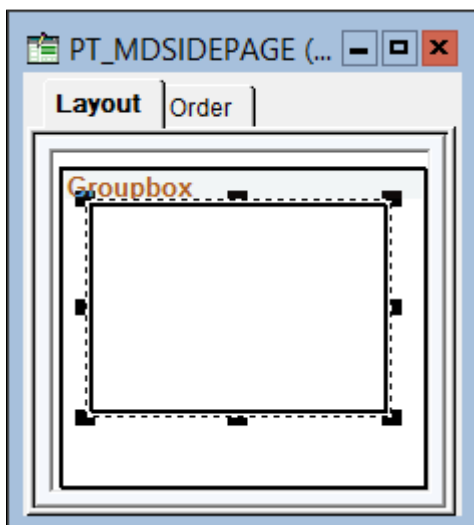
Master/Detail Target Page

The target page displays the content accessed from the side page navigation. The target page has page type set to *Master&Detail Target Page* on the Page Properties: Use tab.

PeopleTools provides a default master/detail target page: PT_MDSTARTPAGE_NUI that you can clone and modify or use as is.

Image: Master/detail target page

This example illustrates the PT_MDSTARTPAGE_NUI definition.

The Target Content group box is placed on the target page and displays the target content at runtime. The group box type is Master/Detail Target with this style applied: *ps_ag-targetcontent.*
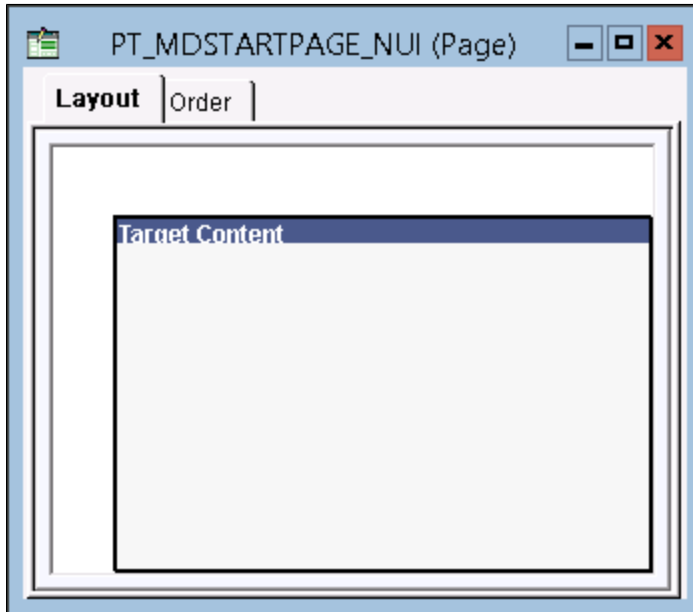
Implementing Master/Detail Components

When implementing a custom master/detail application, you have two approaches to consider:

• Cloning the delivered PT_MDSTARTPAGE_NUI component.

• Loading the master list from a different component.

Cloning the PT_MDSTARTPAGE_NUI Component

When creating your own master/detail component, you can clone the delivered PT_MDSTARTPAGE_NUI component and each of the delivered pages within the component. First, clone each of the delivered pages. Then, clone the delivered component. Finally, replace each of the delivered pages within your cloned component with your cloned page definitions. Use the Registration Wizard to register your component in the portal registry.

Important! For each definition that you clone, copy all PeopleCode associated with the definition.

Then, determine how you want to define the master list:

• By modifying the side page definition directly. Continue with "Creating the Master List."

• By loading the master list from a different component. See the next section, "Loading a Master List from a Different Component."

Loading a Master List from a Different Component

You can use either the PeopleTools-delivered PT_MDSTARTPAGE_NUI component or your custom master/detail component as the base URL when loading the master list from a different component. Then, you will append query string parameters to the URL to specify the menu, component, and page definitions that contain the master list details.

The following example shows an excerpt of the delivered PT_MDSTARTPAGE_NUI component's URL demonstrating how you would do this:

```
c/NUI_FRAMEWORK.PT_MDSTARTPAGE_NUI.GBL?GMenu=QE_NUI&GComp=QE_MD_GROUPLET&GPage=QE_M⇒
D_GROUPLET2
```

Three query string parameters are required:

- `GMenu` specifies the ID of the menu definition.

- `GComp` specifies the ID of the component definition.

- `Gpage` specifies the ID of the page definition.

Creating the Master List

You can create your master list directly in your side page definition. Alternatively, you can create your master list in a separate component and page definition, which is then loaded into your master/detail component via query string parameters on the URL.

Typically, you can create your master list in a grid, or a scroll:

- A row for each component navigation.

- A hyperlink in each row pointing to the desired target destination.

Image: Sample Navigation Grid

The following example shows how to populate the navigation grid dynamically.
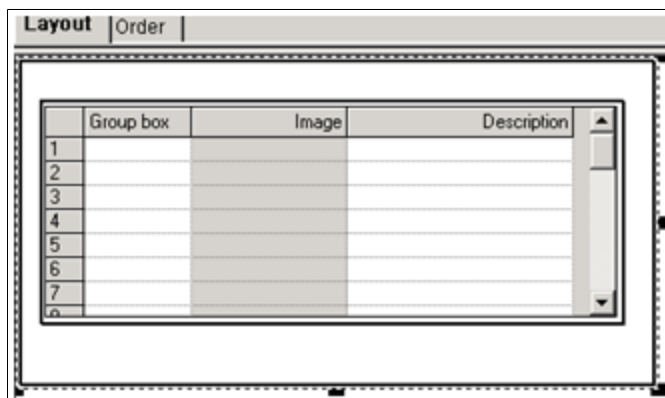
Using Activate PeopleCode you can initialize the grid. For example:

```
&rs(1).PTLAYOUT.GROUPBOX1.SetGroupletActionUrl(&sNewUrl);
&rs(1).PTLAYOUT.GROUPBOX1.SetGroupletDisplayIn(7);
&rs(1).PTLAYOUT.GROUPBOX1.IsGroupletLabelOnly = True;
```

The page Activate PeopleCode must call DoRowAction(nRow,gid) to launch the default target with nRow when loading. nRow sets the grid row, which points to the current target content.

If nRow is not provided, row 0 is the default.

```
AddOnLoadScript("DoRowAction();");
AddOnLoadScript("DoRowAction(" | &nRow | ");");
```

If gid (grid ID) is not provided the system locates a single grid in the grouplet. If there is more than one grid in the grouplet, the gid will be required.

```
AddOnLoadScript("DoRowAction(" | &nRow | ");");
AddOnLoadScript("DoRowAction(" | &nRow",'QE_WORK_REC2$0');");
```

Related Links

Working with Two-Panel Implementations

Working with Two-Panel Implementations

Several fluid application design patterns make use of the two-panel paradigm in which navigation links are displayed in the left panel and the content is displayed in a target content area on the right. These fluid application design patterns include:

- Master/detail components.

- Fluid activity guides.

- Fluid navigation collections.

AJAX Transfers

AJAX transfers, or asynchronous in-place transfers, use AJAX (asynchronous JavaScript and XML) technology to allow component-to-component transfers to occur without exiting from the fluid wrapper (fluid activity guide or master/detail component) that contains the source component from which the transfer was invoked. When AJAX transfers are enabled for the fluid wrapper, then the new content replaces the target content area within the fluid wrapper. When AJAX transfers have not been enabled (the default configuration), the new content replaces the entire browser window. AJAX transfers apply to four PeopleCode functions only: Transfer, TransferExact, ViewContentURLContent, and ViewContentURLFluid.

Image: Elements within a fluid wrapper component

The following diagram illustrates the elements that are presented within a fluid wrapper component:

- Fluid banner — With fluid activity guides, the content of the fluid banner is maintained by the activity guide. With master/detail components, the content is maintained by the individual components displayed in the target content area.

- Optional sub-banner — With fluid activity guides, the content of the sub-banner is maintained by the activity guide.

- Navigation panel — The items in the navigation panel are presented by the fluid wrapper component.

- Target content area — The component selected in the navigation panel is displayed in the target content area.
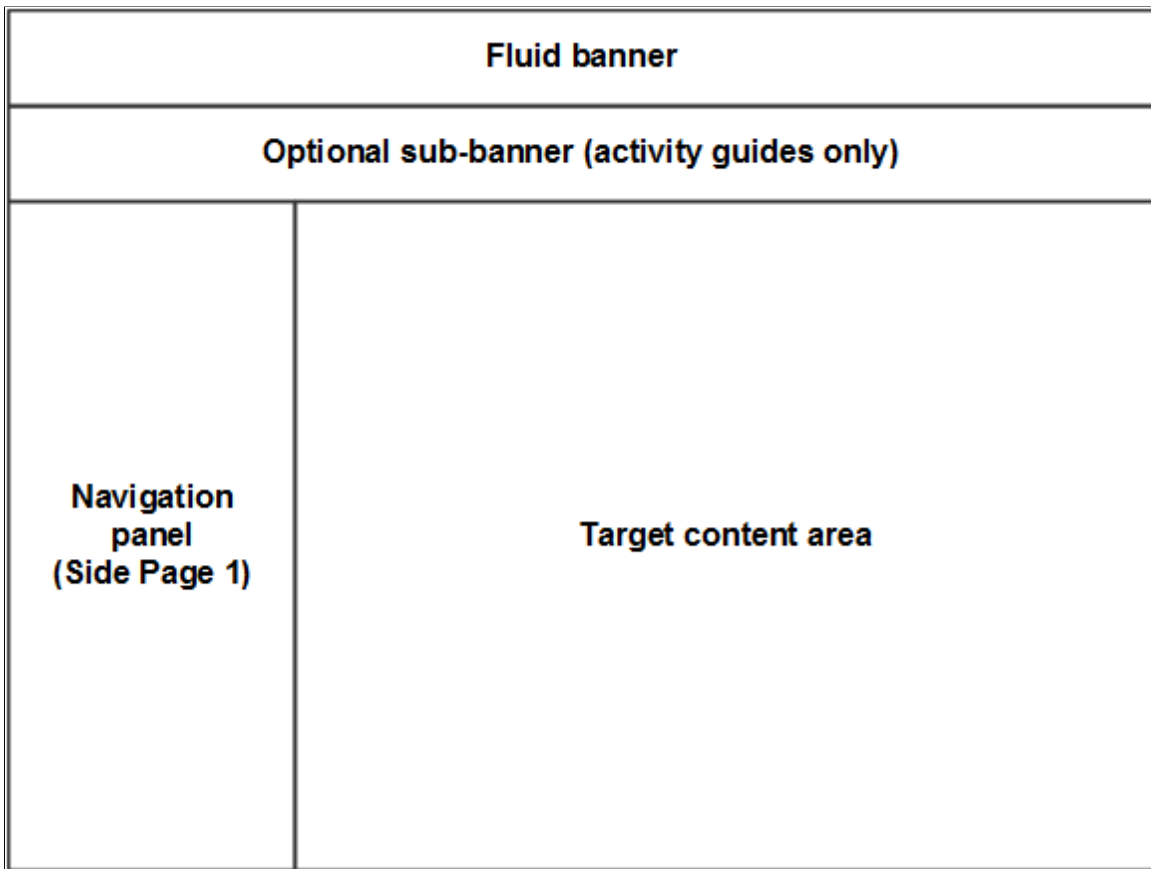
Since AJAX transfers are disabled by default, you must explicitly enable AJAX transfers on each fluid wrapper where the behavior is desired:

- For fluid activity guides including fluid navigation collections, add the following query string parameter to the URL for the content reference definition that launches the activity guide: &AJAXTransfer=y

- For master/detail components, add the following query string parameter to the URL for the content reference definition that launches the master/detail component: &ICAJAXTrf=true

- If you are creating a custom fluid wrapper definition in Application Designer (that is, a custom master/detail component or a custom activity guide component), use the SetMDAJAXTrf function.

In addition, when you enable AJAX transfers, then you might also need enable a slide-out left panel to prevent inadvertent left panel collisions:

•   For master/detail components, fluid activity guides, and fluid navigation collections, add the following query string parameter to the URL for the content reference definition that launches the fluid wrapper: &ICMDListSlideout=true

•   If you are creating a custom fluid wrapper component definition, use the SetMDListSlideout function.

Related Links
"SetMDAJAXTrf" (PeopleTools 8.55: PeopleCode Language Reference)
"SetMDListSlideout" (PeopleTools 8.55: PeopleCode Language Reference)
"Transfer" (PeopleTools 8.55: PeopleCode Language Reference)
"TransferExact" (PeopleTools 8.55: PeopleCode Language Reference)
"ViewContentURLClassic" (PeopleTools 8.55: PeopleCode Language Reference)
"ViewContentURLFluid" (PeopleTools 8.55: PeopleCode Language Reference)
Left Panel Collision Handling with Slide-out Left Panels

Left Panel Collision Handling with Slide-out Left Panels

The "slide-out" left panel is a feature that enables you to include a component that itself has a left panel within fluid wrappers (master/detail components, fluid activity guides, fluid navigation collections, or custom fluid wrapper components) that also use the left panel to present navigation links. If collision handling with the slide-out left panel is not enabled, contention will occur as both the outer and inner components attempt to manage the contents of the left panel. When this occurs, users lose the ability to navigate via the navigation links of the outer, wrapper component, via the navigation links of the inner component, or both.

Two types of contention (or collisions) can occur depending on the implementation of the inner component that causes the collisions:

•   *Intra-page collisions* occur when the inner component has included its own left panel (that is, a page type of Side Page 1). Because the fluid wrapper also implements the left panel using Side Page 1, the collisions occur *within* this side page. Fluid components developed on PeopleTools 8.55 and later releases use Side Page 1 to implement the left panel.

•   *Inter-page collisions* occur when the inner component has included the left panel as part of the page content—for example, in PeopleTools 8.54, the PSL_TWOPANEL layout page provided the mechanism for creating a left panel. In this case, the collisions occur *between* the Side Page 1 of the fluid wrapper and the content of the inner component.

Image: No collisions detected (slide-out left panel not in effect)

When the slide-out left panel is enabled and an inner component *without* its own left panel is displayed, the left panel of the outer, fluid wrapper is displayed normally; there is no need for a slide-out left panel.
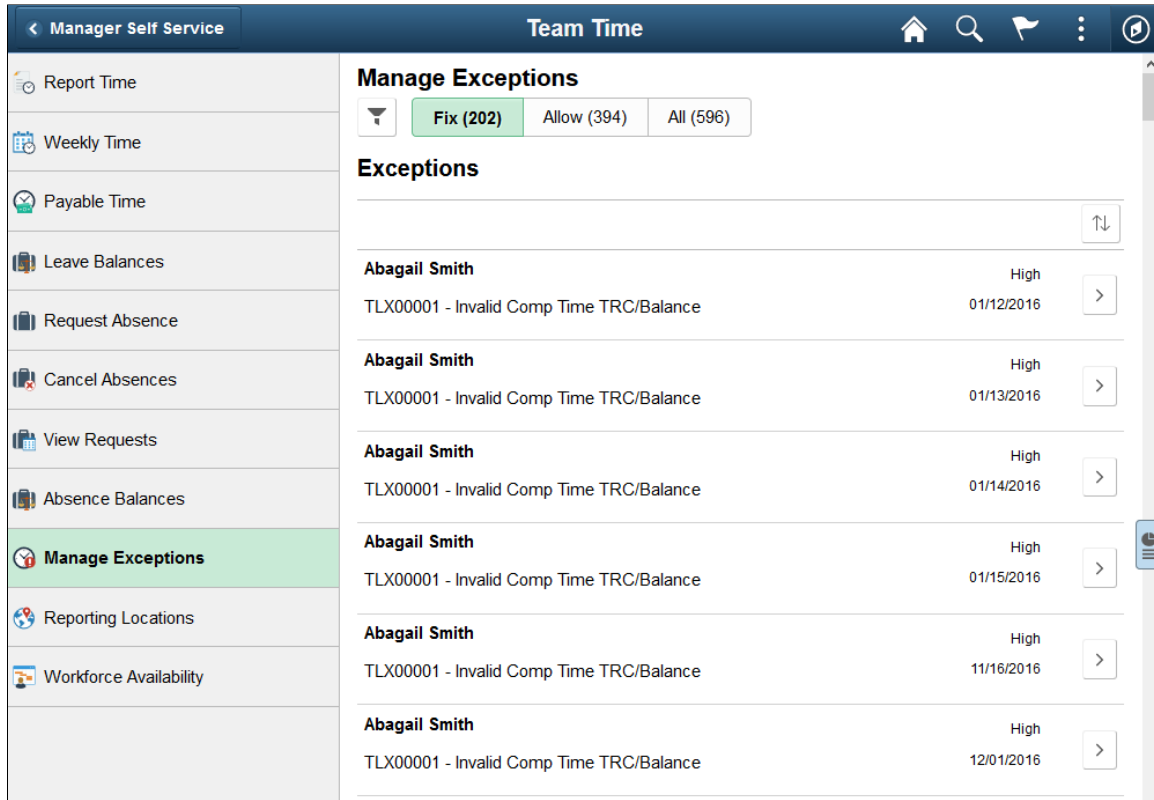
Image: Slide-out left panel in a closed state

When the slide-out left panel is enabled and an inner component *with* its own left panel is displayed, the left panel of the outer, fluid wrapper is initially minimized as shown in the following example. The navigation links displayed are those from the inner component:

Image: Slide-out left panel in an open state

Clicking anywhere on the tab divider (labeled Detail, in this example) will slide open the left panel of the outer, fluid wrapper. The navigation links displayed are those from the outer, fluid wrapper:
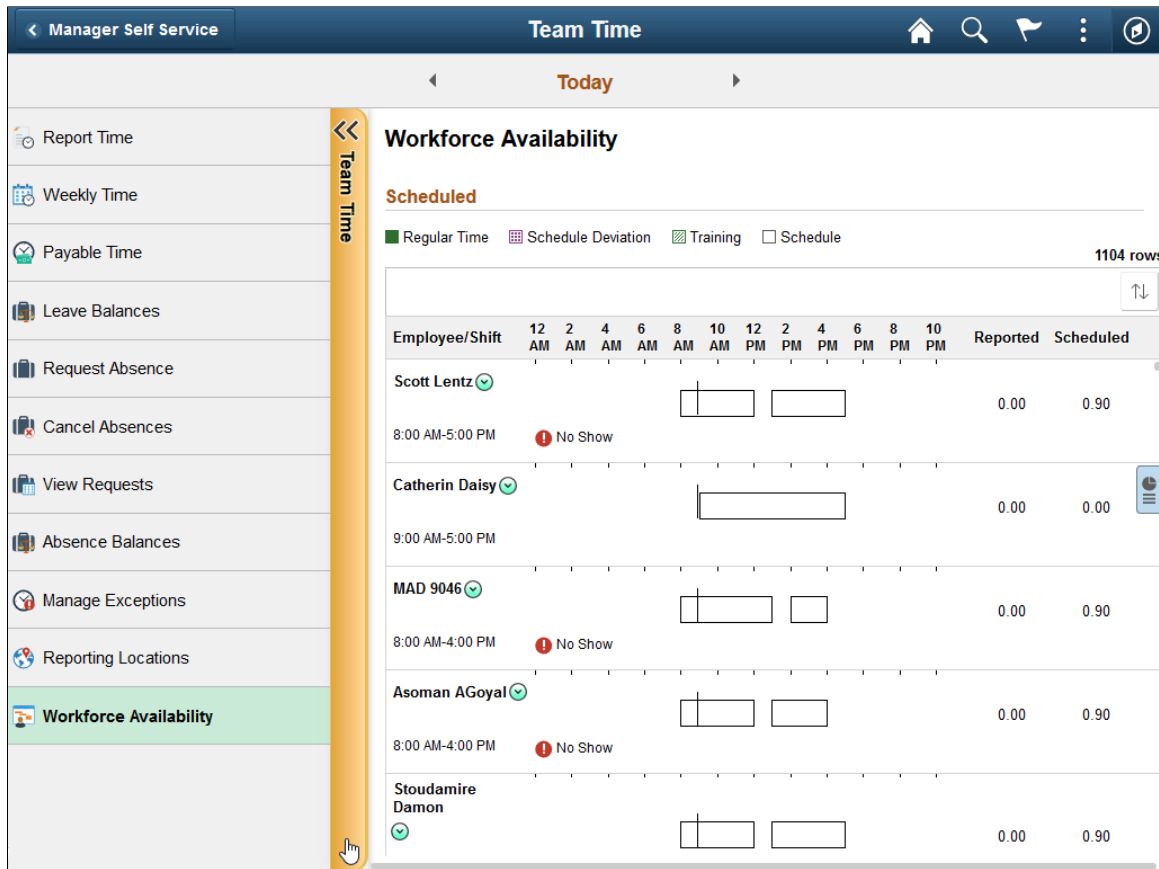
The slide-out left panel is disabled by default. You must enable the slide-out left panel on each fluid wrapper definition which requires its use:

- For master/detail components, fluid activity guides, and fluid navigation collections, add the following query string parameter to the URL for the content reference definition that launches the fluid wrapper: &ICMDListSlideout=true

- If you are creating a custom fluid wrapper, use the SetMDListSlideout function.

In addition, when you enable AJAX transfers, then you might also need to enable a slide-out left panel to prevent inadvertent left panel collisions.

Related Links

"SetMDListSlideout" (PeopleTools 8.55: PeopleCode Language Reference)
"IsSide1CollisionInterPage" (PeopleTools 8.55: PeopleCode API Reference)
"IsSide1CollisionIntraPage" (PeopleTools 8.55: PeopleCode API Reference)
"IsSide1HasCollision" (PeopleTools 8.55: PeopleCode API Reference)
AJAX Transfers
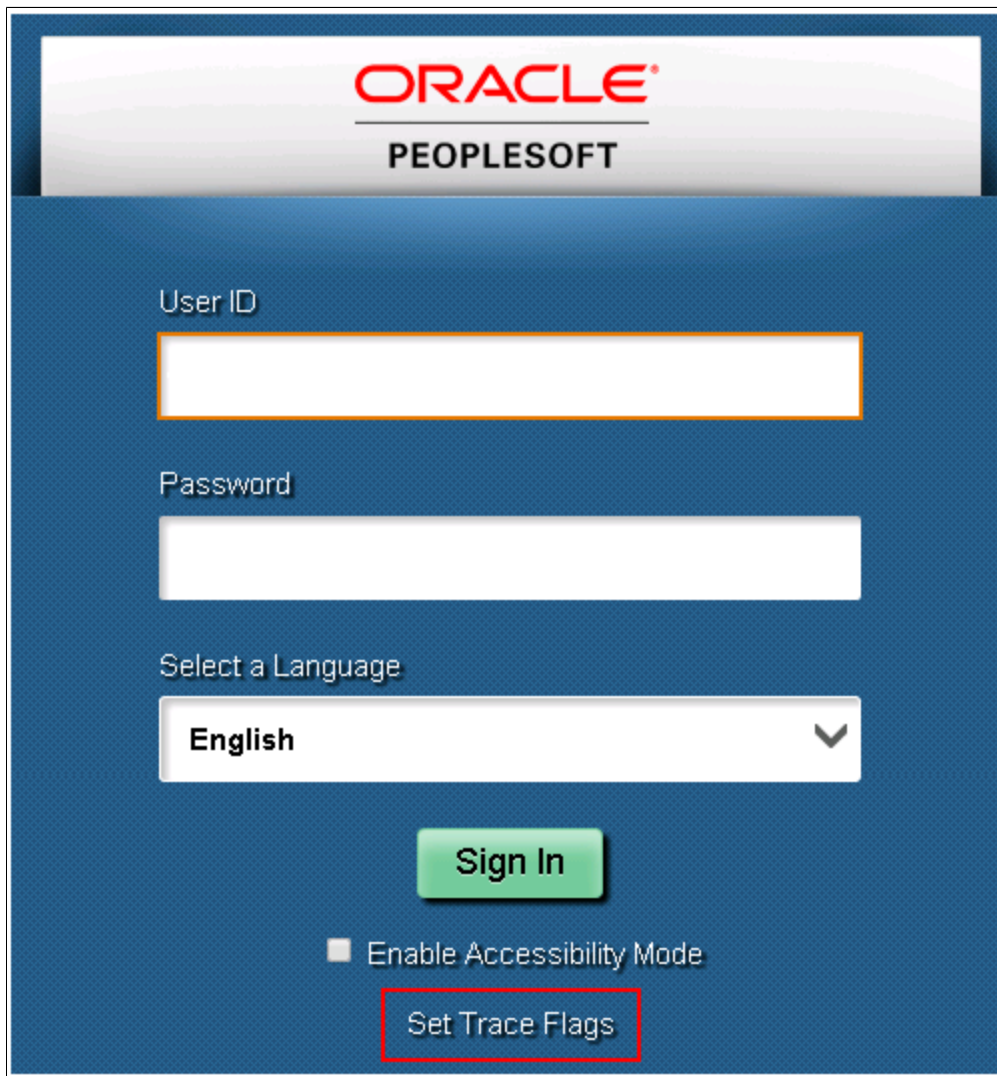
Testing Form Factors

PeopleTools provides a built-in option for testing your fluid components and pages by form factor. You access this option from the PeopleSoft sign in page, by clicking the Set Trace Flags link prior to signing in. You enable the Set Trace Flags link to appear on the sign in page by enabling the Show Trace Link at Signon web profile property on the Debugging tab. Clicking the Set Trace Flags link enables an entire set of tracing options, not just the Form Factor Trace option.

Note: Using the PeopleTools form factor trace option is not a replacement for browser emulation, third party emulation or simulation, or testing on actual devices (smartphones, tablets, laptops, desktops, and so on). It is provided as a tool to give you quick feedback and testing capabilities while developing fluid pages.

Image: Set Trace Flags link on PeopleSoft Sign-in Page
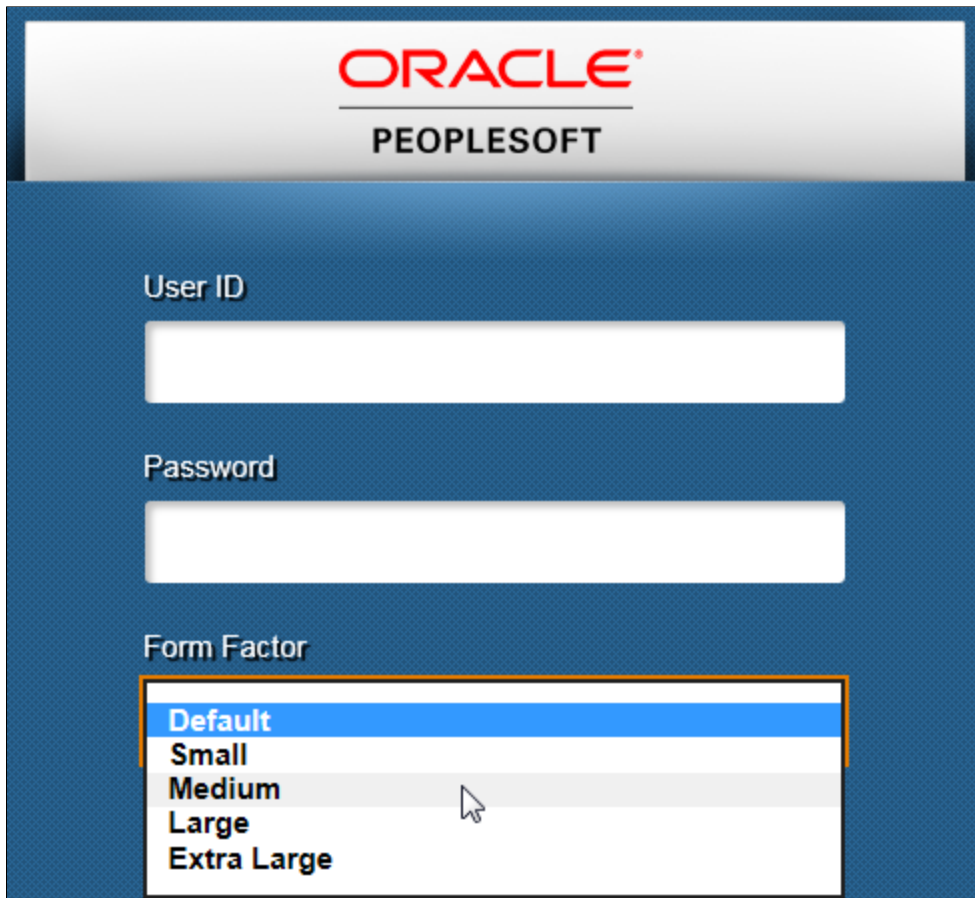
This example illustrates the Set Trace Flag link displayed on the PeopleSoft Sign-in page.

Once you click the Set Trace Flags link, the Select a Language drop-down list becomes the Form Factor drop-down list from which you can select the desired form factor.

Image: Form Factor Drop-Down List Box

This example illustrates the Form Factor drop-down list box displaying the options for selecting a form factor: Small, Medium, Large, and Extra Large.

After you select a particular form factor, the system displays a second browser window rendered in the dimensions of the form factor you selected, displaying the sign-in page.

Image: Form Factor Trace Sign-In Page

This example illustrates the sign-in page assuming the dimensions of the form factor you selected.

After signing in, you can navigate to and view the layout of a fluid page for the form factor you selected.

Image: Form Factor Trace Display

This example illustrates a fluid page rendering in the form factor you selected from the Form Factor trace option.

Chapter 5

Adding Page Controls

Managing Common Control Concepts and Settings for Fluid Pages

This topic provides an overview and discusses common page control properties for fluid pages.

Understanding the Makeup of a Control on a Fluid Page

A basic input page control consists of two items:

- The control.

- The label.

The system considers the control and the label pairing as a single unit and presents them within a single container in the HTML—ps_box-<control_type>.

For example, in the case of an edit box, the system generated this structure in the HTML.

Image: HTML for fluid page controls

This example illustrates the appearance of the HTML used to display a page control.

```
<div id="win0divWM_WR_HDR_CONTACT_DETAIL" class="ps_box-edit" >
  <div id="win0divWM_WR_HDR_CONTACT_DETAILlbl" class="ps_box-label" >
    <label id="WM_WR_HDR_CONTACT_DETAIL_LBL" class="ps-label"
    for="WM_WR_HDR_CONTACT_DETAIL" >
        Alternate E-mail address or Phone number for this request only
    </label>
  </div>
  <div id="win0divWM_WR_HDR_CONTACT_DETAILctrl" class="ps_box-control" >
    <input id="WM_WR_HDR_CONTACT_DETAIL" class="ps-edit" type="text"
    title="Alternate E-mail address or Phone number for this request only"
    onclick="javascript:cancelBubble(event);" maxlength="70" value=""
    ></input>
  </div>
</div>
```

The system generates default styles, which are styled through the style sheets. Applications can override or extend style by adding additional styles or replacing the top-level system default style. In the case of the example above, the application change chose to override the PeopleTools-provided style and it would remove the ps_box-edit class *only* in favor of the specified class from the application definition.

Setting Common Page Control Fluid Properties

For any control that you can add to a page, such as a group box, an image, a grid, and so on, there is a Fluid tab on the properties dialog box for that control. All of the controls share some properties, while others have additional properties that apply only to the function of that control.

Note: The options documented below apply to all fluid controls. Depending on the control selected for the page, additional properties may appear. Properties specific to an individual control type are documented within a section devoted to that control type.

Image: Common Fluid tab Options

This example illustrates the common page control options available for on the Fluid tab of the Properties dialog box for the control. Descriptions of the options follow the example.

Override Tools Style	Indicates to the system *not* to render the default class for that specific control type (as in, the ps_box-edit in the previous example). If this is applied, it applies to *all form factors*.
	No PeopleTools default classes for the control are populated (nor does the system load state information).
Default Style Name	Enter additional class specifications used to adjust the display or behavior of a specific control. This setting applies to all form factors unless specifically overridden by subsequent form factor overrides.
Form Factor Override	Applies classes specifically for adjusting the display (and possibly behavior) of an element on a specific form factor. If no value is specified at the form factor override, the default Style Name will be applied.
Suppress on Form Factor	To prevent a page element from being displayed on specific form factor (device size), select the form factor. If selected, the page element is never rendered on the client device, similar to setting a field as invisible for a classic page. Suppression can improve throughput as the size of the HTML document is reduced.
Label Rendering	The system processes the display of a control as a combined unit of both the display and its label. You can control the position of the label.
	Before Control: Select for the label to appear to the right of the control.
	After Control: Select for the label to appear to the left of the control.
Control Structure	*Basic:* Select to have the control appear on the fluid page just as it would appear on a classic PIA page.
	Advanced: Select to have the control appear on the fluid page having the look and feel of a typical touch-screen application for a tablet or cell phone. For example, a check box would appear as a sliding switch, rather than an empty square box.

Depending on the type of control you are adding to the page, there can be additional options for that control. Use the following links to find more information for each control.

Related Links
Working with Group Boxes
Working with Other Page Controls

Working with Group Boxes

This topic discusses the many uses of group boxes while creating fluid applications.

Using Group Boxes for Fluid Pages

Group boxes are essential containers used for the layout of the fluid page as well as containers for encapsulating and managing control and widget behavior, positioning, and appearance.

Any fluid page must have at least one over-arching group box to contain all other elements of the page. Depending on the number of controls and complexity of the page, additional group boxes can be added. The group box can be used to indicate visual cues but it can also act as a logical grouping of related controls. Using the fluid properties for a group box, you can override or customize the style at each group box layer.

The following example illustrates the use of multiple group boxes in the layout of the fluid page, clearly separating the page elements into distinguishable parts, enabling individual control if needed. For example, the Page Title group box and the Your Content Sub-Page here group box can have individual styles applied if needed.

Every Fluid page should start with a group box. Typically, the outer group box for primary or secondary pages should have the style of ps_apps_content class, though there are exceptions as with 2 panel display. While the ps_apps_content style would not typically be used in subpages, subpages should reside within a containing group box as well.

Image: Group Boxes define fluid page layout

This example illustrates how group boxes are used in the layout of the fluid page, clearly separating the page elements into distinguishable parts.
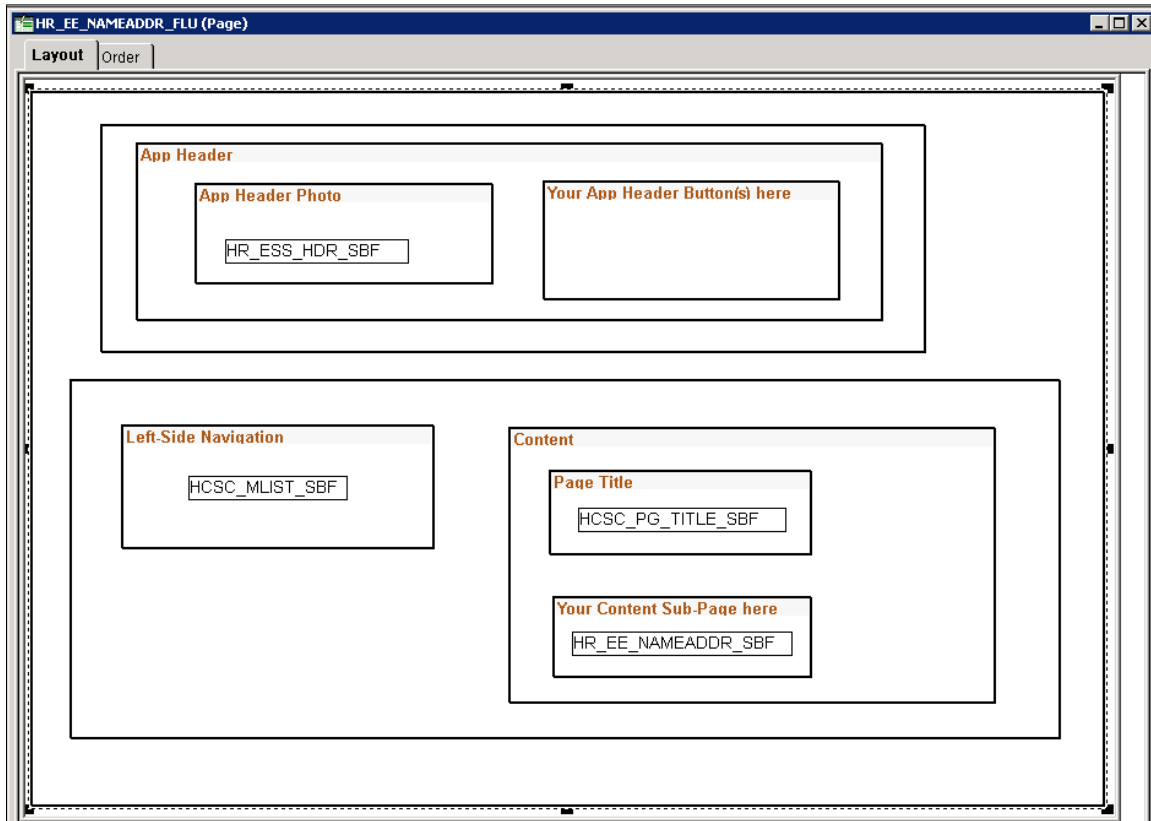
In general, it is recommended to be generous in the use of group boxes to layout and organize your page elements. However, group boxes are not just used for layout purposes. For some controls and widgets, it is required that they be encapsulated within a group box in order to apply behavior characteristics.

In addition to using group boxes as a container with a header (the default type of group box), a group box can have additional functions and behaviors. The most common is the usage of a layout type of group box. Layout group boxes are used to enclose elements for specific layout purposes. Some layouts are defined by styles only.

Some group boxes have additional types, like the Accordion Vertical group box which allows for multiple embedded Accordion group boxes to provide the behavior of a mutually exclusive display of group box, with only one of the Accordion Group boxes expanded and visible at any one time. Group boxes can even be put inside of grids to create new virtual columns containing multiple stacked fields.

Group boxes are the most critical element with which you should become familiar for fluid page development.

Setting Group Box Properties

Image: Group Box Properties Dialog Box: Fluid Tab

This example illustrates to properties to set for group boxes uses on fluid pages. Descriptions of the properties follow the example.

Free Form Style and Suppress On Form Factor are common properties to all fluid controls. See Setting Common Page Control Fluid Properties.

Group Box Type   *Default:* Same as *Container.* It includes both a header structure and a content structure.

Container: Use as a standard container, just as in classic PIA applications, to illustrate groupings of fields.

Accordion Vertical/Horizontal/Group Box: Used to implement expandable sections.

Accordions are a construct where you have a mutually exclusive set of group boxes, only one of which can be expanded at a time (all others collapsed). Accordion Vertical is a container of Accordion Group boxes. The Accordion Group boxes are individual, expandable group boxes within the Accordion Vertical.

Layout Only: Use for a group box only to contain and position child elements.

Panel Control: Use for containing content when creating a side tab.

Custom Header Left/Right/Middle/Bottom: Use to add application controls to header page sections.

Page Title: This is a container for the page title. The header description is generated in a H1 tag (<H1>..<\H1>). It is used by the header page.

Related Field Group: Used to create a FIELDSET collection where the LEGEND (the group box header) is repeated to a screen reader when input type elements are read within the FIELDSET. Typical usage would be when you have a collection of radio buttons based on the same field.

Toggle Container: Use to switch between two group boxes and changes the label of the button controlling the toggle.

Note: Currently, do not use the Toggle Container option. This option is for internal use only until a future release.

Custom Grid Header: Similar to the Custom Header sections, the Custom Grid Header group box type allows you to define a custom area to be available for association with a grid. This is used when defining a grid to look for the available Custom Grid Header defined group boxes to associate with a grid. A custom grid header can be associated only with one grid.

Custom Action Menu: Enables the transaction to add to the Actions List displayed in the banner.

Popup: A type of group box, where the group box label is rendered as a button (a collapsible image can be used to identify an image as well to be displayed with the text). The button is rendered initially and when selected, the content appears as a Popup area associated with the button.

Custom Header Search: Use to create a custom header for component search.

Page Tab: Use to take page tabs running horizontally across the top of the component pages, and insert them into the left sidepage, creating left navigation. For sample usage, refer to PT _SIDE_PAGETABS.

Master/Detail Target: Use to surround the content to be displayed when the master item in the left navigation side page is selected. Add this group box to a master/detail target page. See Working with Master/Detail Components for more information.

Search Action: Use to synchronize component search actions to the actions displayed on the Action List, (other area in the header if preferred). This enables a PeopleCode Event from the selected action, such as Return To List.

Note: When selected, also select *UL* as the HTML Tag Type.

Related Content: Use to display related content on a fluid page. After you add a related content group box to a page, you associate the group box to the related content to the page-level using the Assign Related Content page. (PeopleTools, Portal, Related Content Service, Manage Related Content Service.

HTML Tag Type   Selecting an HTML tag type enables you to create proper HTML structure as defined by HTML specifications.

Default: Same as DIV.

DIV: Set as an HTML 5.0 DIV tag.

Section: Set as an HTML 5.0 Section tag. If creating a section containing an article, you may consider using this group box type.

UL: Set as an HTML 5.0 UL (unordered list).

OL: Set as an HTML 5.0 UL (ordered list).

LI: Set as an HTML 5.0 UL (list item). Required to be structurally correct. The LI should always be within a UL or OL tag, and nothing else should be directly beneath a UL or OL except for LI.

Draggable   Indicates that content can be dragged out of the group box area. Users initiate drag and drop by using touch on touch screen devices and by using the mouse for desktops and laptops.

Droppable   Indicates that content can be dragged into the group box area.

Gestures	Select for group boxes where the drag, drop, and swipe gestures need to be available. Use to limit the scope of the drag, drop and swipe features so that these events are set and invoked efficiently.
	For example, assume you want to drag an element from group box A to group box B, and the parent of these two group boxes is group box C. If you select Gestures for group box C, you do not need to set it individually for group box A and B.
Scroll Bars	*None:* Display group box with no scroll bar.
	Vertical: Display the group box with a scroll bar on the right side, running top-to-bottom.
	Horizontal: Display the group box with a scroll bar on the bottom, running left-to-right.
	Both: Display both vertical and horizontal scroll bars.
	Carousel: Display the scroll bar in carousel mode (slider mode), where records can be viewed in defined sets by spinning or swiping.
Scroll Sizing by Style	Indicates that the size of the scroll will be controlled by style.
	By default, PeopleTools attempts to control the sizing of the group box to accommodate the size of the current page size. If you want to restrict the size of the group box stylistically (using classes and CSS), you can do this so that the system does not attempt to change the size of the group box as needed.

Working with Layout Group Boxes

The layout group box is used only for layout for positioning its child elements. When using layout group boxes, keep these items in mind:

• No default look and feel style is applied, such as no border.

• No header will be generated.

• The label set for the group box will be ignored at runtime, but can be used at design time for reference.

Working with Container Group Boxes

Container group boxes can be used as a standard container, similar to classic PIA application usage. Use group boxes to associate relevant page field controls and to create a page that is intuitive to the user.

When using container group boxes, keep these items in mind:

• A default, system style provides a standard look and feel.

- A header will be generated if a label is defined.

- Custom styles can be used to override default styles.

- The default HTML tag used for group box is "div," which you can override with the HTML Tag Type option.

- You can set the group box as Section and Article elements.

- Can be used as a collapsible data area.

Working with Accordion Group Boxes

The Accordion is a web control that allows you to provide multiple, collapsible panes and display them one at a time. With an accordion group box,

When working with the accordion feature:

- Set the parent group box container as Accordion Vertical for vertical accordion behavior (up and down) or Accordion Horizontal for horizontal accordion behavior (left and right).

- Set the children group boxes (the group boxes that will be the recipient of the expanding and collapsing behavior) as Accordion Group Boxes.

Note: Horizontal Accordion group boxes should not be used in the current release. It is reserved for internal use.

Image: Accordion Group Box Container Structure

This example illustrates that when creating an accordion construct, the outer group box is assigned the Accordion Vertical group box type, and the content within resides in group boxes of type Accordion Group Box.
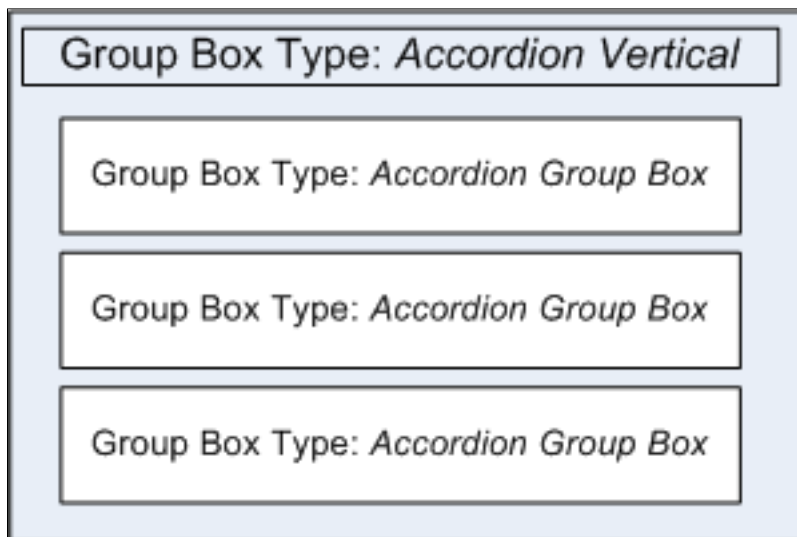
Image: Accordion Group Box Vertical After Page Load

This example illustrates how an accordion group box appears initially, after page load, with all sections collapsed, by default.

Image: Accordion Group Box Vertical After Selecting a Section

This example illustrates how one section can be expanded at a time, showing the page content within, while keeping the other sections collapsed, saving display space.

Creating Custom Headers

Before creating any sort of custom headers, become familiar with the default system header page definition, PT_HEADERPAGE.

You can use the Custom Header Left/Middle/Right/Bottom group box types to reparent custom controls to the default header. You will notice that the default PT_HEADERPAGE header page definition contains a left box, middle box, right box and a bottom box. Each of these group boxes also contains a Custom group box; these correspond directly to the Custom Header group box types and are used to inject custom application controls into the header page sections. On your page definition, create Custom Header Left/Middle/Right/Bottom group box types as needed for your custom controls. At runtime, the content of the Custom Header group boxes will replace the content of the related Custom group box in the header page.

Note: During page rendering, the system identifies the first focusable field based on the organization of the page definition and places the cursor there unless overridden by SetCursorPos PeopleCode. When adding custom buttons to the header of a primary page or a secondary page (for example, when using any of the Custom Header group box types), place the group box with the custom buttons at the bottom of your page definition. This will prevent the custom button from being identified by the system as the first focusable field.

Alternatively, if you do not want to use the default system header page, you can create your own custom header page. See Selecting Page Types.

Working with Related Field Group Boxes

Use related field group boxes to group or associate a set of fields. The fields will be enclosed in

`<fieldSet>…</FieldSet>`

The group header will be within <legend> instead <h..>.

The Collapsible Data Area option should not be used for related field group boxes.

Displaying Embedded Related Content

Using the group box type *Related Content,* you can display embedded related content on a fluid page.

Existing tile definitions can be utilized as related content on fluid pages either as an anonymous service or as an explicitly defined related content service using the PeopleSoft Tile URL type.

Image: Related Content Group Box Type

This example illustrates setting the Group Box Type value to *Related Content,* resulting in the Autosize check box to appear.
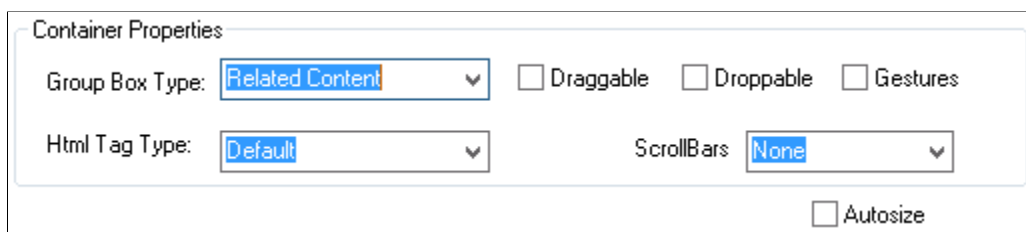
Note: When you set the Group Box Type to *Related Content,* the Autosize check box appears automatically. The Autosize check box on the Fluid Attributes tab for the associated content reference controls the autosize feature for the embedded related content. The Autosize check box in Application Designer applies only to interactive grouplets.

The following procedure provides an example of adding embedded related content to your fluid page. You will complete some of the steps in Application Designer and some in the Manage Related Content Service component in your browser.

Adding a Related Content Group Box to a Fluid Page

To add a related content group box fluid page in Application Designer:

1. Insert a group box on your fluid page in the general position you would like it to display.

2. Double-click the group box, and set the Group Box Type to *Related Content* on the Fluid Tab of the Group Box Properties dialog box.

3. (Optional) Select the Group Box Properties dialog box: Label tab and enter a descriptive label for the displayed related content.

   The label is not rendered at runtime, but it can help in the design time by:

   • Indicating in which field the content will display.

- Differentiating one Related Content group box from another in the case of multiple Related Content group boxes on a page, which is helpful when assigning the appropriate related content to the correct Related Content group box.

Image: Related Content Group Box

This example illustrates a Related Content group box added to a fluid page, with a customized label for identification.

4. (Optional) Assign style, if needed, to this group box using delivered styles. Recall that the system applies default styles, and you would only require additional styling to override or enhance the default styling.

   One of the options for styling is to override the default sizing. If no additional styling is specified, then the embedded related content consumes the default size, which depends on the container or page.

   a. Select the Group Box Properties dialog box: Fluid tab.

   b. Enter a style name in the Default Style Name field:

      To use less than the full width of the container or page, enter the percentage of the width. For example: `psc_width-80pct`.

      To specify a fixed height, specify the height in ems. For example: `psc_height-50em`.

5. Save your changes.

Assigning Page-Level Related Content to the Related Content Group Box

1. Sign in using your browser, and select PeopleTools, Portal, Related Content Service, Manage Related Content Service.

2. Click the Assign Related Content to an Application Page link and select the content reference associated with the component of your fluid page.

   Once selected, the component appears in the Content Reference list on the Manage Related Content for Content References page.

3. On the Manage Related Content for Content References page, click Edit for your content reference.

4. Select the page-level information in the Page Level Related Content group box as described in "Assigning Related Content to Fluid Pages" (PeopleTools 8.55: Portal Technology).

Image: Assigning Page Level Related Content

The following example illustrates sample settings used to assign embedded related content to a fluid page.
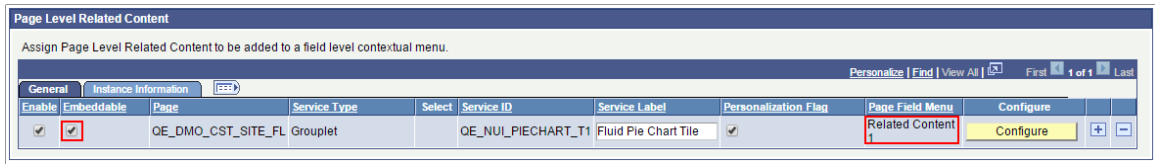
Related Links

"Understanding the PeopleSoft Related Content Framework" (PeopleTools 8.55: Portal Technology)

"Fluid Related Content" (PeopleTools 8.55: Portal Technology)

"Assigning Related Content to Fluid Pages" (PeopleTools 8.55: Portal Technology)

Testing Group Box Layout

Because group boxes are essential to the layout of the fluid page and all of its elements, being able to determine exactly the contents of a group box container is also essential. Select Layout Only on the Fluid tab of the Component Properties dialog box.

Image: Enabling Group Box Layout Testing

This example illustrates where you enable group box layout testing on the Component Properties dialog box, by selecting *Layout Only.*
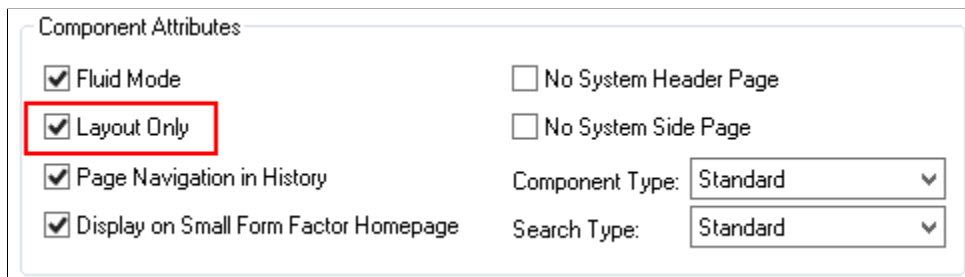
At run time, the system displays the fluid page in the browser using different colored boxes to depict the scope of each group box on the page, providing visual indication for you to verify the scope and contents of each group box container.

Image: Group Box Testing Display

This example illustrates the different colored boxes indicating the scope and coverage of each group box using the My Preferences page (PTGP_USER_PERS_FL).

Working with Other Page Controls

This section contains an overview and discusses aspects specific to fluid page development with regard to adding page controls.

In general, adding page controls to fluid pages is comparable to adding page controls to a classic page, except for using CSS styles to position the control and the obvious look and feel differences at runtime. The content in the this topic call out key differences with regard to fluid development. You should be well versed in the development of classic PeopleSoft applications and all of the various page control options provided by Application Designer prior to beginning a fluid development or customization project.

Working with Grids

Setting Grid Properties

When working with grids, the relevant, common fluid properties apply, but also you can set these properties:

Scroll Sizing by Style	Select to override the system default, automatic scroll container sizing. If you use this option, you *must* control the scroll container sizing through a custom style. If you select this option, be aware of these items:

- If the custom style does not control the scroll container size successfully, the scroll may not appear or work as desired.

- Your custom style must be able to handle window size and orientation change.

- If there is no overflow amount (data rows exceeding occurs limit) lazy scrolling may not be triggered.

Custom Grid Header	Select to specify a custom grid header.
Include Labels in Grid Cells	If using a Tab Separator in your grid, you are allowed to generate field labels within a grid.
	Typically, these are suppressed but because we have the ability to put group boxes within grids, the labels are appropriate to display what the individual fields are (depending on context).

Selecting Grid Layout

There are various grid layout options to select.

For more information on grid layout options, see "Setting Grid Use Properties" (PeopleTools 8.55: Application Designer Developer's Guide).

Enabling Lazy Scrolling for Scrollable Grids and Scroll Areas

Lazy scrolling provides the ability to load grid and scroll content as the end user scrolls downward. Rather than loading all content up front, which would degrade performance, the system loads the content as needed. The system loads the next set of scrollable content automatically as the user reaches the end of the currently displayed content.

The system shows the current set of rows displayed in the grid (row count) on the same line as the grid title. As additional rows display due to scrolling, the row count updates automatically.

To enable lazy scrolling:

1. On the Grid Properties > General tab or Scroll Area Properties > General tab, set the Occurs Count to the number rows to be displayed when the grid appears initially.

   Note: Make sure Unlimited Occurs Count (rows) is not selected.

2. In the Header Navigation Bar Properties for that grid, make sure the Invisible option is clear for these property tabs:

   - Previous

   - Next

• View All

Note: In some situations, such as when displaying a page on a small form factor device, you may want grid scrolling disabled. Apply the style psc_noscroll at the grid level and assign it to Small in the Form Factor Override group box on the Fluid tab in the Grid Properties dialog box to disable scrolling for that grid for small form factors.

Working with Scroll Areas

When working with scroll areas, the relevant, common fluid properties apply, but you can also set these properties:

Layout	Select one of these layout options:
	• *Default* – Displays the rows of the scroll in the standard, default format.
	• *List (Unordered)* – Displays the rows of the scroll as a bulleted (unordered) list.
	• *List (Ordered)* – Displays the rows of the scroll as a numbered (ordered) list.

Working with Edit Boxes and Long Edit Boxes

When working with edit boxes and long edit boxes, the relevant, common fluid properties apply, but also you can set these properties:

Place Holder Text	Display default, place holder, text in the edit box. For example, add a default value or sample format of the field data to assist users.
	None: Disables the display of place holder text.
	Static: Select to enable the Static Text box, and enter the desired text.
	Message Catalog: Select to enable the Message Set/Number boxes. Enter the message set and message number of the message catalog entry that contains the place holder text.
	HTML example:

```
<input type="text" id="PS_APP_END_DATE"
class="ps-edit"
placeholder="Specify the date you expect to
finish the assignment."
value="" maxlength="10" ...>
```

Input Properties	Select the expected data type for users to input. This sets the HTML input type parameter.

Setting input types for forms allow for better input control and validation for various data types, including:

- text

- date

- datetime

- email

- number

- range

- search

- time

- url

Note: The Input Properties setting does not apply to long edit boxes.

Note: In the current release, spell check is not optimized for fluid mode. It is not recommended to enable spell check at the component level or at the field level for fluid applications.

Note: The rich text editor is not supported in fluid applications. Do not enable the rich text editor for long edit boxes on fluid pages.

Working with Number Fields

When you select *number* as the Input Type, the system uses an HTML 5 number input which renders in a fashion specific to the browser/device accessing the page. For example, on a touch device the system displays the numeric keyboard when the number field is in focus, and a browser may display a spinner (spin-button) for a number field.

Note: Browser rendering and HTML 5 rendering may change after the publication of this document.

When setting Input Type to number, consider that HTML 5 restricts the data that can be used as input or displayed for number fields to the following set of characters:

- digits 0 through 9

- "-" for negative numbers

- "." for a decimal point

Adding additional characters (such as a currency symbol, a thousand's separator, and so on) or additional field formatting is not supported for a field with the Input Type set to *number*. For example, if you set Input Type to *number,* the numeric keyboard will be displayed by default for that field on touch devices, however you also introduce the HTML 5 number field character restrictions. In this case, any special characters or formatting applied in Edit Box Properties, field definition formatting, or PeopleCode (such

as currency symbols, thousands separators, and so on) will not be displayed or accepted as input for that field.

On the other hand, if you want to allow additional formatting or characters for numerical values, then set the Input Type to *text,* keeping in mind that with the Input Type set to *text* you lose the device/browser-specific representation (such as the numeric keyboard for example).

A general recommendation would be that you should only set a field's Input Type to number if the underlying field definition Field Type is set to *Number* in Application Designer, or if the Field Type is set to *Character,* you have also set the Format Type to *Numbers Only.*

Working with Subpages

Using subpages can help to reduce clutter in the layout page, and it clearly separates related fields into separate entities. In addition, it enables you to logically separate business logic.

When using subpages, you will notice that the footprint is small in terms of the space it consumes in the layout. The width of the subpage displayed on the primary page is based on the width of the label when you insert into the page (fluid page).

Subpages must be enclosed in a group box of the type layout.

There can be multiple levels of subpages.

Working with HTML Areas

In general, HTML Areas should be used sparingly, if at all, with fluid applications. If used, keep these guidelines in mind:

- Select the HTML Tags Only option on the HTML tab of the HTML Area Properties dialog box for performance reasons. This limits the processing and parsing tasks that the system needs to perform, especially in the case of large grids containing numerous input fields.

- Do not use/reference external JavaScript style sheets in the HTML Area.

Working with Check Boxes

With check boxes on a fluid page, there are differences based on the Control Structure settings on the Check Box Properties > Fluid tab.

The default (Advanced) shows the actual translate values (On/Off, Active/Inactive, whatever they happen to be). With Basic selected, the system shows just a check box, similar to classic rendering.

Label positioning can be important with these options, as you typically want labels to appear after a standard check box but before the control with the Advanced option selected.

Image: Advanced and Basic Check Box Rendering

This example illustrates the differences in rendering and label placement of the Advanced and Basic Control Structure settings for check boxes. The Advanced selection shows the toggle switch, while the Basic selection displays a more standard check box.

Working with Push Buttons and Hyperlinks

In addition to the existing push buttons used in classic applications, these push buttons are available for use with fluid user interface pages.

- Home

- Sign Out

- NavBar

- Add To

- Search

You can create toolbars containing push buttons that appear in custom locations, such as sidebars for example, using your own actions, if needed. Push buttons are not required to appear only in the toolbar at the top of the page. Options that can be found on the Use tab that may also be of interest for fluid pages include:

- Execute PC on Row\Group click: Runs PeopleCode as soon as user clicks on row\group).

- Execute SearchSave Event: When a user runs a search, the system saves the page data prior to running the search so it is not lost. This option applies only to the search button, not every button on the page.

Note: For a button to display an image and text (a combination button), on the Label tab, select Both for the Label Type, then specify the Alt tag for image and Label Image.

Working with Static Images and Static Text

Static controls can be used for fluid pages. They have a different structure than what you see on classic pages, and you can apply style classes to these page elements (to the outer wrapper as with all other controls).

Image: Static Text

This example illustrates the static text control represented in the generated HTML for a fluid page.

```
<div class="ps_box-text" id="win0div$ICField3">
  <span class="ps-text">This is Static Text</span>
</div>
```

Image: Static Image

This example illustrates the static image control represented in the generated HTML for a fluid page.

```
▼<div class="ps_box-staticimg" id="win0div$ICField4">
     <img src="/cs/ps/cache/A_DOWN_1.gif" class="ps-staticimg" alt="Alternate Text for Image" title="Alternate Text for Image">
  </div>
```

Working with Horizontal Rules

The horizontal rule control can be used and you can apply a style on the Horizontal Rule Properties > Fluid tab to be applied to the horizontal rule. Styles are applied at the ps_box-hr level (as with all controls, styles applied in IDE are found in the outer wrapper of the control).

Working with Charts

When inserting charts onto a fluid page, note that there is no Fluid tab on the Chart Properties dialog box. The properties applied to a chart on a fluid page are applied using the Fluid tab of the chart container (the group box surrounding the chart.)

For more information, see "Fluid Mode Considerations for Charts and Gauges" (PeopleTools 8.55: PeopleCode API Reference).

Page Controls Not Supported in Fluid Applications

The following page controls are not supported in fluid applications:

• Frames – Use a group box of type layout instead.

• Pop-up menus – The use of pop-up menus can produce unpredictable results with other aspects of fluid applications.

• Scroll bars.

• Trees.

Applying Styles

Understanding Fluid Styling

The layout of all page elements in a fluid application is controlled by free from style sheets using CSS 3.0 standards. This is one of the most critical differences to consider when comparing classic PIA applications to fluid applications. With each control that you add to a page, PeopleTools *automatically* applies a style class to that control structure, and that style class enables that control to behave and render successfully on a fluid page.

In some cases, a single style class creates a change in behavior, color, or look and feel of a page element. In other cases, depending on the control, a structure of multiple style classes are applied to create the behavior and look and feel of a control. It is critical that before you attempt to override any delivered default styles that you observe the structure of the control in the HTML source of the rendered page

so that you can gain insight into the elements involved in the control's structure and the styles that are currently being applied, not to mention the order in which they are applied.

When the system applies a style, by default or override, the style is applied to the container in which the control resides, and those styles apply to all elements in that container, which includes the structure of the control, the label, the field, and so on. In a fluid application, you do not apply a style to discreet elements, such as just to the label for instance.

Identifying PeopleTools Style Sheets

PeopleTools provides a main style sheet for fluid pages and controls, while also supplying additional style sheets for various fluid features, such as the NavBar.

Style Sheet	Description
PSSTYLEDEF_FMODE	General style sheet for fluid applications (included on every page), containing the majority of fluid style classes.
PTNUI_NAVBAR_CSS	Contains the styling used for the Navigation Bar (NavBar).
PTPN_NOTIFYWIN_CSS	Notification/Alerts styling. Included on every page where notifications are enabled.
PTNUI_PINTO_CSS	Controls the styling for the "Add to…" functionality. Included on pages with the Add to feature enabled.
PTS_NUI_FFSTYLE	Used by both global search and component search features.
PTPG_MENUSTYLE	Used by pivot grid search features.
PTPG_NUI_GRID_CSS	Used by pivot grid display.

Overriding Delivered Style Classes

The system default style sheet is PSSTYLEDEF_FMODE, which contains all the default, delivered styles for all page elements and controls.

You can create custom free form style sheets in Application Designer, and you can choose to include them and override or extend the default styles at varying levels for page elements.

The system-wide style sheet can be overridden by way of the PeopleTools options page, by selecting the custom style sheet and assigning it to the Default StyleSheet Name option. This is a global, system-wide setting. If set, the system completely ignores any style contained in the delivered style sheet, PSSTYLEDEF_FMODE. Otherwise, if not set, the system applies PSSTYLEDEF_FMODE by default.

Note: Overriding the delivered system-wide style sheet is *not* recommended.

Note: Use PeopleTools, Portal, Branding, Branding System Options to set fluid branding options in the Fluid Component group box.

To incorporate additional style sheets at either the component or page level, use AddStyleSheet PeopleCode:

```
AddStyleSheet(StyleSheet.stylesheetname);
```

AddStyleSheet adds the reference to the style sheet to the header section of the HTML page.

Image: Style Sheets in HTML head section

This example illustrates style sheets added into the HTML <head> section.

```
▼<head>
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta name="viewport" content=" user-scalable=yes,initial-scale=1.0,minimum-scale=1.0,width=device-width">
  ▶<script language="JavaScript">…</script>
    <link id="PSSTYLEDEF_FMODE_2036_css" rel="stylesheet" type="text/css" href="/cs/ps/cache/PSSTYLEDEF_FMODE_2036.css">
    <link rel="stylesheet" type="text/css" href="/cs/ps/cache/HR_ALERTS_FL_2037.css">
    <link rel="stylesheet" type="text/css" href="/cs/ps/cache/PTNUI_PINTO_CSS_2027.css">
    <link rel="stylesheet" type="text/css" href="/cs/ps/cache/PTNUI_NAVBAR_CSS_1479.css">
```

In Application Designer, you can also elect to override or extend the default styles, using the Fluid tab for any page or control use the Free Form Style group box to call out styles to be modified. To completely ignore the default style, select Override PeopleTools Style, otherwise you extend or add to the default style. For situations where you have adjustments to make depending on the form factor of the device, you can set those in the Form Factor Override fields for the various form factors.

When working with the order in which style sheets are applied, consider:

• When you call AddStyleSheet, the system checks if there is a theme with a mapped, skinned style sheet for *that* style sheet. If so, the system loads that style sheet after the style sheet passed in the argument.

• The order in which you call AddStyleSheet determines the order in which the style sheets appear in the stylesheet list.

• The system determines the order in which the PeopleTools style sheets are applied.

• Regardless of custom changes you may make, standard CSS rules *always* apply. It is very important for the developers to understand how CSS and style sheets work.

Image: Form Factor Override

This example illustrates an approach to adjusting the layout of controls based on the form factor of the device accessing the fluid page.

For details on the override options, see Setting Common Page Control Fluid Properties.

Free form styles set in Application Designer are static and can't be changed, however, styles set using PeopleCode are dynamic, meaning they can be switched programmatically at runtime. Styles set in Application Designer can't be changed or queried using PeopleCode at runtime. Dynamic styles are applied *only* with PeopleCode; they can be added, changed, deleted or queried at runtime.

It is not recommended to add style to the system containers. Add your own group boxes, and add style to that.

Example:

```
&fld1.AddFFClass("addffclass");
  &fld1.ReplaceFFClass("addffclass", "replaceffclass");
  &fld1.FreeFormStyleName = "pc_ffclass";
```

In your implementation, Oracle strongly recommends that you only override style classes with respect to coloring, theme, and branding; a practice often referred to as "skinning." Oracle strongly discourages overriding delivered style classes with regard to page layout (positioning, padding, filling, and so on).

You only skin complete style sheets, not the sub style sheets contained within a style sheet. To extend (or skin) a delivered style sheet, you want the delivered style sheet to appear first in the style sheet list, followed by the override or "skinned" style sheet. For example, PSSTYLEDEF_FMODE should appear before PSSTYLEDEF_FMODE_skinned, allowing the original to be applied, and the "skin," containing the supplemental changes, to be applied after.

Note: With CSS style sheet application, the last style sheet in the sequence to be applied takes precedence.

It is recommended to skin or extend only one style sheet at a time. As in, if you have multiple style sheets, have the skinned style sheet appear just after (under) the style sheet it extends or overrides. If the extended or override style sheet appears beneath other style sheets in the list (assuming there are multiple style sheets), any override in the skinned style sheet will apply to any style sheet specified before it, which may result with unintended style overrides.

If you intend on extending or overriding the delivered styles, consider these items:

- You need to be aware of all the styles applied in any previously applied style sheet(s).

- If overriding styles, it is recommended to change only color characteristics.

- It is not recommended to override any styles related to structural (layout or positioning) characteristics or behaviors applied to page elements.

- Do not modify a delivered style sheet. Any changes will be overridden in any future upgrade.

- Styles are associated directly to structure (container, label, control, and so on), and any change PeopleSoft makes to structure in future releases can affect any custom styles you've implemented.

Setting Min/Max Width on Content

Not all content displays as desired when stretched to a maximum size of a device or reduced (shrunk) too much to fit a smaller space, or device. There are functional minimum and maximums, which should be defined on the content so that scrolling occurs or the size of the area no longer grows in terms of min/max values for width. When creating custom fluid pages, or customizing delivered pages, it is your responsibility to manage the styling for these min/max values.

It is recommended that you *do not* set the min/max values directly on the group box with the style ps_apps_content, as this can cause browser issues.

The ps_apps_content style implements the flex model. Pages are laid out as a column based on flex model with horizontal stretching. This can cause varying results per browser type when attempting to stretch the width fully while restricting the width based on a max-width value.

It is recommended to include a layout group box within the ps_apps_content group box containing your page content, and apply the max/min widths on the interior group box.

Layout Element	Example
Structure and Style Class	```GROUPBOX – class=ps_apps_content GROUPBOX – class=psa_my_content <page content goes here>```
Style	```.psa_my_content { min-width:320px; max-width: 1440px; margin:0 auto; /* center content horizontally */ }```

Hiding Columns in a Flex Grid Using Style

You can style an entire column in a flex grid by using the page field name of the field (or group box) within a grid. The page field name will be applied directly to the column (both the table header [TH tag] and the cells [TD tag]) as a style. At that point, you can affect styling an entire column (centering, font weight, and so on).

You can hide an entire column stylistically, using the page field name (as opposed to hiding the column using PeopleCode, which can prohibit complete rendering of the column).

If you are hiding a column using style, you must consider the following items.

Consideration	Description
Accessibility	If you stylistically hide the trigger column used for the row action, you must make sure that you un-hide the trigger column in Accessibility Mode.

This is because in accessibility mode, the row action is disabled and the column used as the trigger field now becomes actionable and can receive focus. In accessibility mode, the row action trigger should have the following characteristics:

• It must be visible.

• It must have text (alt text or image) that explains the action.

For example, if you have a field on a grid that has a page field name of "ABC_XYZ", the style applied on all references to that column is "ABC_XYZ" (TD and TH tags in the table). To properly hide the entire column, use the following:

`:root:not(.psc_mode-access) .ABC_XYZ { display:none;` |
| Grid Header Alignment | If you stylistically hide a column, *do not* stylistically hide the last column in a grid. Doing so can adversely affect grid header layout.

PeopleTools uses the last column of the grid to adjust the grid header if it needs to have an overhang for the scroll bar, enabling the grid scroll bar to appear as part of the grid. If you hide the last column, the overhang is assigned to the last column and if it is hidden, the overhang can be lost, resulting in a misaligned grid header. |

Working with Flex Grid Column Alignment

A flex grid consists of two table structures that support both the fixed grid header and a scrollable grid body. The column styes are applied to both table constructs that make up this page element. However, these styles can have adverse effects on the grid header, causing alignment issues if you attempt to adjust column dimensions.

If you are attempting to set any dimensions on columns, make sure you are familiar with the limitations of using min/max values on grid columns.

If you attempt to set a dimension of any type (width, min-width, max-width, and so on), make sure you affect only the real table and not the header-only table, which the system uses for header alignment.

For example, assume a column named XYZ_ABC as a page field name in the grid. That class is assigned to all "TD" and "TH" tags both in the real table (ps_grid-flex) as well as the table used internally for fixed header implementation (ps_grid-flex-head). To style a dimension and change the text color, see the following:

```
.ps_grid-flex .XYZ_ABC { min-width:10em; } /* set dimension only on real table */
.XYZ_ABC {color: #A00;} /* set color both real and header-only table */
```

Working with JavaScript
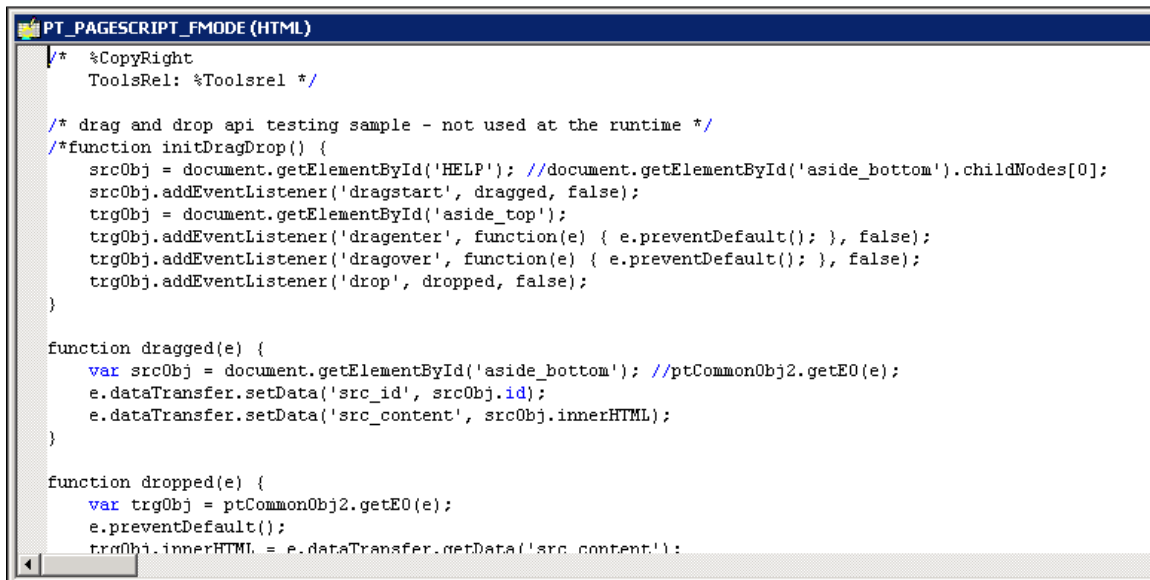
This topic discusses information related to using JavaScript with fluid applications.

Delivered JavaScript.

PeopleTools delivers an HTML definition, PT_PAGESCRIPT_FMODE, containing all the provided JavaScript functions.

Image: PT_PAGESCRIPT_FMODE

This example illustrates the delivered JavaScript in PT_PAGESCRIPT_FMODE.

These JavaScript functions control a wide range of page behavior, including (but not limited to):

• Scrolling for group boxes

• Accordion widget

• Toggle widget

• Grouplet (like a pagelet – smaller)

• Drag and drop group boxes

• Grid lazy scroll

• Carousel – (used only for scroll areas)

• Transfer animation

• File attachment

• Search page and nav bar

• Related action menu

- Side page tabs

- New action buttons – Back, home, logout

- Search side page

Custom JavaScript Guidelines

Incorporating custom JavaScript programs is supported, but application developers and implementation teams should use custom JavaScript programs judiciously, and they should be well aware of any implications introduced by the custom code.

With the architecture of the PeopleSoft Fluid User Interface, PeopleTools provides increased flexibility and opportunity to customize. However, with that flexibility comes increased responsibility for developers to test and manage any custom code. PeopleTools can only impose logical restrictions and processing constraints on the code it delivers and tests.

When considering adding custom JavaScript programs, consider the following items:

- Include JavaScript files in component PeopleCode, such as PostBuild, for better performance. Otherwise, the system needs to perform extra processing, such as inserting references into the header and checking to make sure all of the styles have been downloaded and applied, and before completing the page load\activate process.

- Avoid using custom JavaScript events on a control.

- If there are any JavaScript errors from application onload, the page will also encounter problems loading.

- Do not use JavaScript to change the structure of the document.

- Don't use JavaScript to add/change the free form styles on the controls. Use PeopleCode to change/ add free form style.

- Only add JavaScript events on the controls that do not have any system generated script.

Considering Accessibility for Fluid Pages

When building any application, developers need to consider accessibility requirements always. This topic describes some key concepts to keep in mind with regard to accessibility while developing fluid pages.

For more information on accessibility for PeopleSoft applications in general, see "Accessibility in PeopleSoft Applications" (PeopleTools 8.55: Accessibility Guide).

Accessibility Consideration	Description
ARIA live regions	Use Accessible Rich Internet Applications (ARIA) live regions sparingly.

Accessibility Consideration	*Description*
Background and foreground colors	If you specify a foreground color, it is strongly recommended that you also specify a background color and vice versa. Otherwise, users changing their system colors may produce invisible text. Also, be cautious when using any of the following CSS techniques: • Background images • Gradients • RGBA (transparent or semi-transparent) colors These all make it difficult to test for color contrast, if not applied appropriately.
Background images	Do not use background images for informational icons. They are not visible in high contrast mode. Only do this for decorative images.
Button or hyperlink row actions	If you have a button or hyperlink and the control is used for a row action, you must supply a translatable label for the item regardless of whether it's visible or hidden. This label is needed for screen reader users.
Consistency with images and control labels	Use images and controls consistently throughout your application. For example, do not include a button labeled "Search" on one page and a button labeled "Find" on another page of the same application.
Creating custom controls	If you create a custom control using HTML Area, which is discouraged, the "Name, Role, Value" rule has to be in place. For example: • Custom controls need a meaningful programmatically exposed name. • Custom controls have appropriate roles specified using the role attribute. • Custom controls have an appropriate value or state and property information conveyed using the relevant WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications) or HTML 5 attributes.
Drag-and-drop	If you are leveraging drag-and-drop capabilities on your page, you must provide an accessible mechanism that non-sighted users can use. A similar mechanism must be available. For example, with homepage personalization, users can drag-and-drop tiles into their own specific layout and order. For non-sighted users, who cannot use drag-and-drop, a Personalize Homepage interface is available where users can also use the keyboard to arrange pagelets.
Dynamic information	User should be able to stop, pause, hide information that is moving, blinking, or auto-updated if: • Starts automatically. • Lasts more than five seconds.

Accessibility Consideration	Description
Focus indicator	An end user must be able to see when focus is set for a control, for example a box surrounding the widget: • If the widget is in HTML area, provide your own focus indicator. • If you style a delivered widget in a unique way so it no longer looks like others of its kind, you also need to make sure to style a clear focus indicator.
Group box popups	Do not use the word "MENU" in a title for a group box popup. Otherwise, a screen reader user will expect a keyboard that works with MENU to be available (up/down/left/right arrow keys), which is not available yet.
Hidden fields	When a fields is not visible (for example, within closed accordion content or within group box that is itself set as invisible), the field should not be tab-able. It also must not be set to have focus.
HTML areas	The controls in an HTML area must be able to be operated using the keyboard only, with no reliance on the mouse. Use `tabindex="0"` for controls that are not accessible using TAB, but should be (those that can receive user input).
Links versus buttons	If a link doesn't transfer users to a different location when they press or tap it, it is strongly recommended that you use a button, rather than a link.
List items	List items must have one interactive field only, which becomes the target of the row action for the list item. In addition, limit the number of display-only fields (attributes) to three or less per list item. Because each list item is treated as a single cell, all of the fields are read by the screen reader for each list item when it is selected. More than four fields presents too much information for the reader to retain and act on. If you must present more than four fields, create a grid instead of a list. That way, each grid cell can be read individually.
Lists	If you have a list, use list grid layout to render it, because it is a list, not a table, you are presenting. In addition, when you have only one column, it is not a table, it is a list. If you want to show how many items are in the list, do not use the word "rows" for the counter. *Row* implies tables, which is confusing for screen reader users.
Plug-ins	If you use plug-ins, verify that content rendered with those plug-ins is accessible.
Read-only group box content	For group boxes that contain read-only elements and have a row action, it is recommended to add an aria-label and set the role using PeopleCode: `&Fld.JavaScriptEvents = "aria-label=""Partial Days None""role="`
Row headers	Fluid pages should specify a row header for tables, specified in Application Designer. Use the Row Header property on the Grid Properties > Use tab.

Accessibility Consideration	Description

Screen reader mode	• If your fluid component and classic component are exactly the same in functionality, you might want to display the classic component when a user is in fluid screen reader mode for desktop and tablet users; classic pages do not display well on smart phones. This can eliminate the need to design fluid pages for regular and screen reader mode.
	
	• For selected list items, add the following ARIA attribute:
	
	`aria-current="true"`
	
	• If you use styles to hide a column, ensure that you don't hide the column in screen reader mode. If you always hide it, there will be no action available on the row. For example, the use of none in the following example is incorrect:
	
	`:root:not(.psc_mode-access) .GRID1_EDIT,`
	`:root:not(.psc_mode-access) .GRID1_EDIT {`
	` display:none !important`
	`}`

Sensory descriptors	Avoid instructions with sensory characteristics or descriptions of page elements, such as: • Color (red, blue, and so on). • Visual location (above, below, and so on) • Orientation (right, left) • Size and shape (small, larger, round, arrow, and so on). Use field identification instead to point users to page elements.
Table summaries	Provide table summaries for table. Use Application Designer to populate them. Use the Summary attribute for Accessibility property on the Grid Properties > Use tab.
Tab order and field focus	With fluid pages, the tab order cannot be set by the application developer, unlike classic pages. The tab order is based strictly on a top-to-bottom, left-to-right sequence. For fluid pages, design in such a way that the logical tab order is maintained as much as possible, leveraging group boxes to gather page elements together as necessary. Pay attention to field focus by tabbing to make sure the tab order is in a logical sequence. Pay close attention to these situations: • Default focus location for initial page display. By default, the focus will be set to the first interactive field found on the page. However, if this field is not visible because it is within an invisible element (for example, a group box that is set to invisible) or within collapsed accordion content, then focus will be lost. • Focus location after refreshing part of a page. • Focus location before and after pop-up windows are displayed.

Accessibility Consideration	Description
Testing	Perform the following additional testing: • Test your application in Windows high contrast mode. • If you build your fluid application for tablets or phones, test using a native screen reader. For example, use VoiceOver (OSX native screen reader).
Toggle buttons	If you have a toggle button with different visual styles for on and off, you should do one of the following: • Change the label of the button when on versus off. • Code WAI-ARIA states aria-pressed="true/false".

Chapter 6

Working with PeopleCode for the Fluid User Interface

Working with Fluid Page and Component PeopleCode

This topic discusses PeopleCode constructs and usage specific to working with fluid pages and fluid components.

Execution Order for Page Activate Events

The PeopleCode page Activate events for fluid pages execute in the following:

1. Header page (PT_HEADERPAGE is the PeopleTools-delivered fluid header page).

2. Side pages (Side 1 page and Side 2 page).

3. Footer page.

4. Main page (Standard Page page type).

Note: The Activate event is not supported for subpages, fluid or classic.

Related Links
Selecting Page Types

Built-In Functions

This table lists the PeopleCode built-in functions provided for use with fluid applications at the component or page level.

Function	Description
"AddJavaScript" (PeopleTools 8.55: PeopleCode Language Reference)	Add JavaScript to the document or page.
"AddMetaTag" (PeopleTools 8.55: PeopleCode Language Reference)	Add a meta-tag to the document or page.
"AddOnLoadScript " (PeopleTools 8.55: PeopleCode Language Reference)	Add the script to execute in the onload event of the document.

Function	*Description*
"AddStyleSheet" (PeopleTools 8.55: PeopleCode Language Reference)	Add a free form style sheet to the HTML document for the page or component.
"CreateSearchRowset" (PeopleTools 8.55: PeopleCode Language Reference)	Create the rowset based of the search record and key list values.
"DoModalComponentPopup" (PeopleTools 8.55: PeopleCode Language Reference)	Display a secondary fluid component in a modal, secondary window with a child relationship to the parent window. See Transfer and Modal Functions for a complete list of functions available for implementing transfers and creating modal windows.
"DoModalPopup" (PeopleTools 8.55: PeopleCode Language Reference)	Display a secondary fluid page in a modal, secondary window with a child relationship to the parent window.. See Transfer and Modal Functions for a complete list of functions available for implementing transfers and creating modal windows.
"ExecuteSearchSavePC" (PeopleTools 8.55: PeopleCode Language Reference)	Execute the SearchSave PeopleCode on the search page.
"GetAddSearchRecName" (PeopleTools 8.55: PeopleCode Language Reference)	Return the add search/prompt record name.
"GetComponentTitle" (PeopleTools 8.55: PeopleCode Language Reference)	Return the name of the component (from the component definition).
"GetPagePrefix" (PeopleTools 8.55: PeopleCode Language Reference)	Return the prefix for the page based on the page type as set on the Use tab.
"GetPageTitle" (PeopleTools 8.55: PeopleCode Language Reference)	Return the title of the page.
"GetPageType" (PeopleTools 8.55: PeopleCode Language Reference)	Return the page type—for example, header, footer, prompt, and so on.
"GetPanelControlStyle" (PeopleTools 8.55: PeopleCode Language Reference)	Return the styles set by the system that control the state of the left and right panels.
"GetSearchRecName" (PeopleTools 8.55: PeopleCode Language Reference)	Return the search/prompt record name.
"GroupletRequestSource" (PeopleTools 8.55: PeopleCode Language Reference)	Return an Integer value indicating the source page (location) for the grouplet.

Function	Description
"IsAddEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether Add mode has been enabled on the component's Use tab.
"IsAddMode" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the component build process is in add mode.
"IsAGComponent" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the component type is set to Activity Guide on the Fluid tab.
"IsAGRequest" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the request was initiated by an activity guide wrapper.
"IsBackEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether back navigation is enabled on the component.
"IsFluidMode" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether Fluid Mode is enabled on the component.
"IsFluidSearchStart" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the Fluid Search is started on the component.
"IsGroupletRequest" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the request was initiated by a tile (also referred to as a grouplet).
"IsHomeEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the Home button is enabled on the component.
"IsIScriptAuthorized" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the current user is allowed to access the specified iScript.
"IsLogoutEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether Logout is enabled on the component.
"IsMDComponent" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the component type is set to Master/Detail on the Fluid tab.
"IsMDGuided" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the master/detail or activity guide wrapper is in guided mode (that is, Previous and Next are displayed in the fluid banner).
"IsMDListPopup" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value whether the master/detail or activity guide wrapper is in non-optimized mode (that is, navigation to pages is presented in a drop-down list in the fluid banner, not in the left panel).

Function	Description
"IsMDNonOptimized" (PeopleTools 8.55: PeopleCode Language Reference)	This function is equivalent to the IsMDListPopup function.
"IsMDRequest" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the request was initiated by a component from within the master/detail wrapper.
"IsMDSearchEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the search type is set to Master/Detail for the current component.
"IsModeless" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the target content for the tile is currently being displayed in a modal window—that is, the Display In setting for the tile is set to *Modal*.
"IsNavBarEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the NavBar is enabled on the component.
"IsNextInListEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether NextInList is enabled on the component.
"IsNotificationEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the Notifications button is enabled on the fluid component. This corresponds to the Notifications toolbar option on the component's Fluid tab.
"IsPinEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the Add To functions are enabled on the component.
"IsPrevInListEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether PreviousInList is enabled on the component.
"IsSaveEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether Save is enabled on the component.
"IsSearchEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	**Important!** In PeopleTools 8.55 or later, use either the IsMDSearchEnabled function or the IsStandardSearchEnabled function instead to determine which type of search is enabled. Return a boolean value indicating whether search is enabled on the component.
"IsSingleComponentAG" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether this is a single component activity guide.
"IsSingleUnitOfWork" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether this is a single unit of work activity guide.
"IsSmallFFOptimized" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the Display on Small Form Factor Homepage option has been selected on the Fluid tab of the component properties.

Function	Description
"IsStandardSearchEnabled" (PeopleTools 8.55: PeopleCode Language Reference)	Return a boolean value indicating whether the search type is set to Standard for the current component.
"SetAddMode" (PeopleTools 8.55: PeopleCode Language Reference)	Set the AddMode attribute in the panel processor.
"SetMDGuided" (PeopleTools 8.55: PeopleCode Language Reference)	Set whether the master/detail or activity guide wrapper is in guided mode (that is, Previous and Next buttons are displayed in the fluid banner).
"SetMDListPopup" (PeopleTools 8.55: PeopleCode Language Reference)	Set whether the master/detail or activity guide wrapper is in non-optimized mode (that is, navigation to pages is presented in a drop-down list in the fluid banner, not in the left panel).
"SetPanelControlStyle" (PeopleTools 8.55: PeopleCode Language Reference)	Specify a String value that updates the system with styles that control the state of the left and right panels.
"SetThemeId" (PeopleTools 8.55: PeopleCode Language Reference)	Set the theme ID for the page so that proper theme style sheets get loaded.
"SetTransferAttributes" (PeopleTools 8.55: PeopleCode Language Reference)	Set certain fluid-related properties prior to executing a transfer to a different page or component.

Related Links

"Understanding Activity Guide Templates and Template Types" (PeopleTools 8.55: Portal Technology)

PT_PAGE_UTILS Application Package

The fluid page utilities classes of the PT_PAGE_UTILS application package provide tools to manage certain aspects of your fluid applications:

- PT_PAGE_UTILS:Banner —

  Use the PT_PAGE_UTILS:Banner class to control items in the fluid banner including the title as well as visibility of some of the items that appear in the Actions list.

  Many methods of this class 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 list such as in an activity guide, in a master/detail component, or in a modal window.

- PT_PAGE_UTILS:PanelController —

  Use the PT_PAGE_UTILS:PanelController class to inquire on and update the modes and visibility of the Side Page 1 (left panel) and Side Page 2 (right panel). Because this class spawns Panel objects for each side panel, you do not need to instantiate Panel objects directly.

  The system determines the initial characteristics of each panel based on the configuration of the component, activity guide instance, or master/detail instance. The system also maintains the panels' runtime characteristics. The panel state is not maintained within the PanelController object (or

any Panel objects it spawns). Therefore, you must ensure that any changes are updated back to the server to effect an actual change either by directly invoking the UpdatePanel method or indirectly by invoking the Initialize method with auto-update set to true.

- PT_PAGE_UTILS:Utils —

  The PT_PAGE_UTILS:Utils class provides general utility methods for working with viewport settings, displaying confirmation messages in the banner, and working with toggle buttons.

Important! To be rendered correctly across all device types, each fluid application should set the viewport. The "SetDefaultViewport" (PeopleTools 8.55: PeopleCode API Reference) method is the recommended way to do this.

Related Links

"Banner Class" (PeopleTools 8.55: PeopleCode API Reference)
"PanelController Class" (PeopleTools 8.55: PeopleCode API Reference)
"Utils Class" (PeopleTools 8.55: PeopleCode API Reference)

Additional Field Class Methods and Properties

This section lists other methods and properties of the Field class used in fluid development.

See "AddFFClass" (PeopleTools 8.55: PeopleCode API Reference).

See "Field Class Properties" (PeopleTools 8.55: PeopleCode API Reference).

See "AriaWrapperAttributes" (PeopleTools 8.55: PeopleCode API Reference).

See "FreeFormStyleName" (PeopleTools 8.55: PeopleCode API Reference).

See "GetBaseName" (PeopleTools 8.55: PeopleCode API Reference).

See "HtmlAttributes" (PeopleTools 8.55: PeopleCode API Reference).

See "HtmlInputType" (PeopleTools 8.55: PeopleCode API Reference).

See "IsHyperlink" (PeopleTools 8.55: PeopleCode API Reference).

See "JavaScriptEvents" (PeopleTools 8.55: PeopleCode API Reference).

See "PlaceholderText" (PeopleTools 8.55: PeopleCode API Reference).

See "ReplaceFFClass" (PeopleTools 8.55: PeopleCode API Reference).

See "SetImageUrl" (PeopleTools 8.55: PeopleCode API Reference).

Additional Grid Class Properties

This section lists other properties of the Grid class used in fluid development.

See "Layout" (PeopleTools 8.55: PeopleCode API Reference).

See "Grid Class Properties" (PeopleTools 8.55: PeopleCode API Reference).

See "Selector" (PeopleTools 8.55: PeopleCode API Reference).

Additional Meta-HTML

Additional meta-HTML you can use with fluid development include these constructs.

See "%ExplainMessage" (PeopleTools 8.55: PeopleCode Language Reference).

See "%FORMFACTOREXTRALARGE" (PeopleTools 8.55: PeopleCode Language Reference).

See "%FORMFACTORLARGE" (PeopleTools 8.55: PeopleCode Language Reference).

See "%FORMFACTORMEDIUM" (PeopleTools 8.55: PeopleCode Language Reference).

See "%FORMFACTORSMALL" (PeopleTools 8.55: PeopleCode Language Reference).

See "%BB" (PeopleTools 8.55: PeopleCode Language Reference).

See "%BP" (PeopleTools 8.55: PeopleCode Language Reference).

See "%BV" (PeopleTools 8.55: PeopleCode Language Reference).

Determining Browser and Device Type

Depending on the device or browser accessing the application, you may want to set various options or styles catered to that device type, browser type, or browser platform (operating system).

See "Request Class Methods" (PeopleTools 8.55: PeopleCode API Reference).

See "GetDeviceInfo" (PeopleTools 8.55: PeopleCode API Reference).

See "BrowserDeviceType" (PeopleTools 8.55: PeopleCode API Reference).

See "BrowserDeviceFormFactor" (PeopleTools 8.55: PeopleCode API Reference).

See "BrowserFluidCapable" (PeopleTools 8.55: PeopleCode API Reference).

See "BrowserTypeClass" (PeopleTools 8.55: PeopleCode API Reference).

See "BrowserPlatformClass" (PeopleTools 8.55: PeopleCode API Reference).

See "ExtraLarge" (PeopleTools 8.55: PeopleCode API Reference).

See "Large" (PeopleTools 8.55: PeopleCode API Reference).

See "Medium" (PeopleTools 8.55: PeopleCode API Reference).

See "Small" (PeopleTools 8.55: PeopleCode API Reference).

Using Drag and Drop PeopleCode

The PeopleCode used to handle drag and drop functionality needs to be component-level PeopleCode and to be added to the PT_WORK.PT_BUTTON_DND field. With drag and drop, you drag data from a field, row, or rowset in a source group box and drop it onto a field, row, or rowset in a target group box.

See "GetDNDField" (PeopleTools 8.55: PeopleCode API Reference).

See "GetDNDRow" (PeopleTools 8.55: PeopleCode API Reference).

See "GetDNDRowset" (PeopleTools 8.55: PeopleCode API Reference).

See "GetDNDTargetField" (PeopleTools 8.55: PeopleCode API Reference).

See "GetDNDTargetRow" (PeopleTools 8.55: PeopleCode API Reference).

See "GetDNDTargetRowset" (PeopleTools 8.55: PeopleCode API Reference).

See "Draggable" (PeopleTools 8.55: PeopleCode API Reference).

See "Droppable" (PeopleTools 8.55: PeopleCode API Reference).

Creating Grouplets with PeopleCode

A grouplet is the encapsulation of content from one source (for example, a component, an iScript, or external content) that is presented within the context of another fluid application page. Tiles, embedded related content, and related information are all examples of grouplets. In each of these scenarios, the grouplets are created by the system at runtime based solely on configuration details.

However, you can use methods and properties of the Field class to generate grouplets from group box fields programmatically. The following example with comments illustrates how to create a grouplet based on component content. The behavior of the grouplet can be enhanced by optional method invocations to set a target URL, where the target content is displayed, and so on.

Example

```
Declare Function PSURLtoNewWin PeopleCode FUNCLIB_PTPP.PTPP_HYPERLINK FieldFormula;

Local string &src_URL = PSURLtoNewWin(GenerateComponentContentURL(%Portal, %Node, M⇒
enuName.MY_MENU, %Market, Component.COMPONENT_A, Page.MAIN_A, %Action_UpdateDisplay⇒
));

Local string &tgt_URL = PSURLtoNewWin(GenerateComponentContentURL(%Portal, %Node, M⇒
enuName.MY_MENU, %Market, Component.COMPONENT_B, Page.MAIN_A, %Action_UpdateDisplay⇒
));

Local Field &fld = GetField();

/* Use the following three methods to create a display-only grouplet.    */
&fld.SetGroupletType(%GroupletNoStyle);
/* Specifies the grouplet's source content */
&fld.SetGroupletUrl(&src_URL);
/* Specifies the group box's page field name. Not used with interactive  */
/* grouplets, which render the entire component.                         */
&fld.SetGroupletID("GRPLTC_NOINTER");
```

```
/* Optional. Specifies the target URL, which makes the entire grouplet tappable. */
&fld.SetGroupletActionUrl(&tgt_URL);

/* Optional. Specifies how the action URL is opened. Default is current window.  */
&fld.SetGroupletDisplayIn(%GroupletActionInNewWindow); /*  */

/* Enable the icon for zooming to open the modal – similar to related          */
/* information in the right panel.                                             */
<* &fld.IsGroupletZoomIconEnabled = True;
/* Modal options if %GroupletActionInModal is used for SetGroupletDisplayIn.    */
&fld.SetGroupletMOptions("bAutoClose@0;"); *>
```

Related Links

"SetGroupletActionUrl" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletDisplayIn" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletEventMsg" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletID" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletImage" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletImageUrl" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletMOptions" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletTargetID" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletTimer" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletType" (PeopleTools 8.55: PeopleCode API Reference)

"SetGroupletUrl" (PeopleTools 8.55: PeopleCode API Reference)

"IsGroupletRequest" (PeopleTools 8.55: PeopleCode Language Reference)

Limitations on Accessing a Component as a Grouplet and Then in a Standalone Manner

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.

Specifically, standalone access refers to navigation to the fluid component from the Navigator menu, a homepage, a dashboard, a component transfer, execution of DoModalComponentPopup, and so on. The same fluid component cannot be specified as a grouplet (that is, as dynamic tile content, embedded related content, related information, or programmatically via grouplet PeopleCode) and it cannot be displayed in a "modeless" manner (such as the zoom on related information, the modal target for a tile, through ViewContentURL, and so on).

Implementing Transfers and Modal Windows with PeopleCode

Transfer and Modal Functions

Eighteen PeopleCode functions are available to use for implementing transfers and creating modal windows. Some functions are available for use in fluid applications only, in classic applications only, or in other specific contexts. The information on these functions is presented in a table that has been split into four parts. See the legend that follows the table for an explanation of terms and abbreviations.

PeopleCode	Function	Source Location					Action				Target			Rel
Function	Description	FS	CS	AG/MD	IG	MW	F2F	F2C	C2C	C2F	CW	NW	2W	Avail
"DoModal" (PeopleTools 8.55: PeopleCode Language Reference)	Use the DoModal function to display a secondary page in a modal, secondary window with a child relationship to the parent window.	Y	Y	*	*	*	Y	N	Y	N	N	N	Y	pre-8.54
"DoModalComponent" (PeopleTools 8.55: PeopleCode Language Reference)	Use the DoModalComponent function to display a secondary component in a modal, secondary window with a child relationship to the parent window.	Y	Y	*	*	*	Y	N	Y	N	N	N	Y	pre-8.54
"DoModalComponentPopup" (PeopleTools 8.55: PeopleCode Language Reference)	Use DoModalComponentPopup as the recommended function to display a secondary fluid component in a modal, secondary window with a child relationship to the parent window. This function includes a *modal_options* parameter, which allows you to specify characteristics of the modal window such as height and width.	Y	N	*	*	*	Y	N	N	N	N	N	Y	8.54.00
"DoModalPopup" (PeopleTools 8.55: PeopleCode Language Reference)	Use DoModalPopup as the recommended function to display a secondary fluid page in a modal, secondary window with a child relationship to the parent window. This function includes a *modal_options* parameter, which allows you to specify characteristics of the modal window such as height and width.	Y	N	*	*	*	Y	N	N	N	N	N	Y	8.54.00

PeopleCode Function	Function Description	Source Location					Action				Target			Rel
		FS	CS	AG/MD	IG	MW	F2F	F2C	C2C	C2F	CW	NW	2W	Avail
"DoModalX" (PeopleTools 8.55: PeopleCode Language Reference)	Use the DoModalX function to display a secondary page modally when you do not want it to display in a modal, secondary window.	N	Y	*	N	N	N	N	Y	N	Y	N	N	pre-8.54
"DoModalXComponent" (PeopleTools 8.55: PeopleCode Language Reference)	Use the DoModalXComponent function to display a secondary component modally when you do not want it to display in a modal, secondary window.	N	Y	*	N	N	N	N	Y	N	Y	N	N	pre-8.54
"RedirectURL" (PeopleTools 8.55: PeopleCode API Reference)	Use the RedirectURL function in an iScript to navigate to the location specified by the URL parameter.	N	N	N	N	N	N	N	N	N	Y	N	N	pre-8.54
"Transfer" (PeopleTools 8.55: PeopleCode Language Reference)	Use the Transfer function to exit the current context and transfer the user to another page.	Y	Y	Y[1]	Y[2]	Y[2]	Y	Y	Y	Y	Y[1]	Y[3]	N	pre-8.54
"TransferExact" (PeopleTools 8.55: PeopleCode Language Reference)	Use the TransferExact function to exit the current context and transfer the end user to another page using an exact match of all keys in the optional key list.	Y	Y	Y[1]	Y[2]	Y[2]	Y	Y	Y	Y	Y[1]	Y[3]	N	pre-8.54
"TransferModeless" (PeopleTools 8.55: PeopleCode Language Reference)	Use the TransferModeless function to open a new page in a modeless window on top of the parent window.	N	Y	*	*	N	N	N	Y	N	N	N	Y	pre-8.54
"TransferTop" (PeopleTools 8.55: PeopleCode Language Reference)	Use the TransferTop function to exit the current context and transfer the user to another page by always replacing the current window.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	8.55.13

| PeopleCode | Function | Source Location | | | | | Action | | | | Target | | | Rel |
Function	Description	FS	CS	AG/MD	IG	MW	F2F	F2C	C2C	C2F	CW	NW	2W	Avail
"ViewContentURL" (PeopleTools 8.55: PeopleCode Language Reference)	Use the ViewContentURL function to open a new browser window and navigate to the location specified by the URL parameter.	N	Y	N	N	N	N	N	Y	N	N	Y	N	pre-8.54
"ViewContentURLClassic" (PeopleTools 8.55: PeopleCode Language Reference)	Use the ViewContentURLClassic function to navigate to the classic content specified by the URL parameter.	Y	Y	Y[1]	Y	Y	N	Y	Y	N	Y	N	N	8.55.14
"ViewContentURLFluid" (PeopleTools 8.55: PeopleCode Language Reference)	Use the ViewContentURLFluid function to navigate to the fluid content specified by the URL parameter.	Y	Y	Y[1]	Y	Y	Y	N	N	Y	Y	N	N	8.55.14
"ViewContentURLModeless" (PeopleTools 8.55: PeopleCode Language Reference)	Use the ViewContentURLModeless function to open a modeless modal window displaying the PeopleSoft component content (either classic or fluid) specified by the URL parameter.	Y	N	Y	Y	N	Y	Y	N	N	N	N	Y	8.55.11
"ViewURL" (PeopleTools 8.55: PeopleCode Language Reference)	Use the ViewURL function to navigate to the location specified by the URL parameter.	Y	Y	N	N	N	Y	Y	Y	Y	Y	Y[3]	N	pre-8.54
"ViewURLModeless" (PeopleTools 8.55: PeopleCode Language Reference)	Use the ViewURLModeless function to open a modeless modal window displaying the non-component content (for example, an iScript or external content) specified by the URL parameter.	Y	N	Y	Y	N	N	N	N	N	N	N	Y	8.55.11

PeopleCode	Function		Source Location						Action					Target			Rel
Function	Description		FS	CS	AG/ MD	IG	MW	F2F	F2C	C2C	C2F	CW	NW	2W	Avail
"ViewURLTop" (PeopleTools 8.55: PeopleCode Language Reference)	Use the ViewURLTop function to exit the current context and navigate to the location specified by the *URL* parameter by always replacing the current window.		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	8.55. 13

Legend	Definition
FS	Fluid standalone
CS	Classic standalone
AG/MD	Activity guide (incl. fluid navigation collections) or master/detail
IG	Interactive grouplet
MW	Modeless window
F2F	Fluid-to-fluid
F2C	Fluid-to-classic
C2C	Classic-to-classic
C2F	Classic-to-fluid
CW	Current window
NW	New window
2W	Secondary window (modal or modeless)
*	Refer to the values for the source type as a standalone component
1	When AJAX transfers are not enabled (default), the action exits the fluid wrapper and the content replaces the current window. When AJAX transfers are enabled, the new content is displayed in the target content area of the fluid wrapper.
2	When the target setting is current window, the content replaces the current content.
3	On iOS only, when the Safari browser is in full screen mode and when content has been set to open in a new window, the content replaces the current window instead.

Modal Options

Modal windows can be launched from fluid components in four different ways. Each of these four scenarios allows you to specify modal options:

• Programmatically, use one of the following PeopleCode built-in functions: DoModalComponentPopup, DoModalPopup, ViewContentURLModeless, or ViewModeless. Modal options are specified by the *modal_options* parameter.

• For a group box defined as a grouplet, use the SetGroupletMOptions method to set modal options. (See Creating Grouplets with PeopleCode for more information on creating grouplets.)

• On the Fluid Attributes page of a content reference definition, if you specify *Modal* for the Display In option, then the Modal Parameters field is displayed.

• For a Tile Wizard definition, at Step 4, if you specify *Modal* for the Display In option, then the Modal Parameters field is displayed.

Important! Always use a semicolon to terminate each modal option. For example:

```
&sPopupOptions = "bAutoClose@1;bPopup@1;";
```

The following table lists modal options in alphabetical order. For Boolean options, specifying 1 enables the option; specifying 0 disables it:

Option	Values	Default	Description
bAutoClose	0 | 1	0	Enable automatically closing the modal window when the user clicks anywhere outside the modal window.
bAutoCloseWarn	0 | 1	0	Enable display of a save warning if content in the modal window has been changed when bAutoClose has been enabled. **Note:** This option is applicable only when bAutoClose has been enabled by DoModalComponentPopup or DoModalPopup.
bCenter	0 | 1	1	Display the modal window near the center of the main window.
bClose	0 | 1	1	Enable the close icon in the header of the modal window. **Note:** This option is valid only for popup group boxes, only with the SetGroupletMOptions method, and only when the header is displayed; otherwise, it is ignored in all other contexts. **Important!** Do not disable both of auto-close and the close icon.
bFullScreen	0 | 1	0	Display the modal window so that it covers the entire screen.
bHeader	0 | 1	1	Display the header for the modal window. **Note:** This option is ignored by DoModalComponentPopup and DoModalPoup because the header content is displayed within the same iframe as the target content within the modal window. You must use styling on the page definition to hide the header with these PeopleCode functions.
bMask	0 | 1	1	Display a transparent mask over the parent window. **Important!** Disabling the mask is not recommended. Moreover, a mask is required when bAutoClose is enabled.
bPopup	0 | 1	0	Display the modal window adjacent to the trigger field with a chat dialog-style arrow pointing to the trigger field.

Option	Values	Default	Description
sTitle	String	""	Set the title in the header of the modal window.
			Note: This option is ignored by DoModalComponentPopup and DoModalPoup because the header content is displayed within the same iframe as the target content within the modal window.
			Note: To allow spaces in the title and to properly escape special characters, store the title as a message catalog entry. Then, use the EscapeHTML built-in function on the retrieved message. For example:
			`&sMenuTitle = EscapeHTML(MsgGetText(11124, 525, "Message Not` `&sOptions = "sTitle@" | &sMenuTitle | ";");`
sWidth sHeight	String	none	Set the height and width of the modal window as a string value. You can specify any CSS supported unit of measure such as em, in, cm, px, %, and so on. For example: `sWidth@50em;sHeight@100em;` **Note:** The recommended approach for sizing a modal window is to apply styles at the page level.
width height	Number	none	Set the height and width of the modal window in pixels. For example: `width@400;height@400;` **Note:** The recommended approach for sizing a modal window is to apply styles at the page level.

Related Links

"DoModalComponentPopup" (PeopleTools 8.55: PeopleCode Language Reference)

"DoModalPopup" (PeopleTools 8.55: PeopleCode Language Reference)

"SetGroupletMOptions" (PeopleTools 8.55: PeopleCode API Reference)

"ViewContentURLModeless" (PeopleTools 8.55: PeopleCode Language Reference)

"ViewURLModeless" (PeopleTools 8.55: PeopleCode Language Reference)

Working with the Back Button

The Back button is displayed on the left side of the fluid banner (or the left side of the fluid-like portal header on classic pages). The Back button is used to navigate back to the previously visited item (for example, component, homepage, activity guide, navigation collection, WorkCenter, or MAP application). The label on the Back button varies to indicate the item to which the user will return.

The navigation history stack tracks navigation from component to component, including navigation from standard components, homepages. dashboards, activity guides, navigation collections, WorkCenters, and MAP applications. When a user navigates from one of these items, the Back button will be labeled appropriately and will point back to that item. However, page transfers within a component are not tracked in the navigation history stack. To support navigation tracking to remote content in a clustered PeopleSoft environment, an additional HTTP request, PTGETREMOTEDATA, is issued from the local

site to the remote site. For more information on the Back button's behavior in specific circumstances, see "Using the Back Button" (PeopleTools 8.55: Applications User's Guide).

Application developers can override the default Back button behavior (for example, specifying a custom Back button label or indicating that a specific transfer is to be ignored) using the SetTransferAttributes PeopleCode function, which alters the entries written to the navigation history stack. In addition, component property settings or the SetTransferAttributes function can be used to enable tracking of page-to-page transfers within a component.

Related Links

"Setting Internet Properties" (PeopleTools 8.55: Application Designer Developer's Guide)

"SetTransferAttributes" (PeopleTools 8.55: PeopleCode Language Reference)

Chapter 7

Managing Fluid User Interface Deployments

Configuring Web Profile Settings for Fluid Applications

Web Profile Configuration - General Page

Navigation

Access the Web Profile Configuration page. (Select PeopleTools, Web Profile, Web Profile Configuration.)

If you intend to enable online help for your fluid components, you must specify a Help URL value in the web profile on the General page.

For information on all of the fields on the Web Profile Configuration - General page, see "Configuring General Portal Properties" (PeopleTools 8.55: Portal Technology).

For more information on enabling help for a fluid page, see Setting Fluid Properties.

Web Profile Configuration - Look and Feel Page

Navigation

Access the Web Profile Configuration - Look and Feel page. (Select PeopleTools, Web Profile, Web Profile Configuration. Select the Look and Feel page.)

Image: Web Profile Configuration - Look and Feel page, fluid properties

The following example illustrates the Fluid Homepage property that you can set for fluid applications.

Fluid Homepage	Displays the name of the component used to render fluid homepage tabs.
	The default value is *NUI_FRAMEWORK.PT_LANDINGPAGE.GBL*.
Override	Click to change the default fluid homepage. The Override Fluid Homepage Component page is displayed with these fields:

• Menu Name: The default value is *NUI_FRAMEWORK*.

• Component: The default value is *PT_LANDINGPAGE*.

• Market: The default value is *GBL*.

• Fluid Homepage: This field is blank by default.

Important! When this value is set, it appears as `?LP=LB_PAGE_ID` and it overrides any values set on the Assign Branding Themes page. See "Assigning Branding Themes" (PeopleTools 8.55: Portal Technology) for more information on defining the default fluid homepage for a portal and by user attribute.

The values of these four fields are concatenated to produce the value of the Fluid Homepage field on the Look and Feel page.

For information on all of the fields on the Web Profile Configuration - Look and Feel page, see "Configuring Appearance and Character" (PeopleTools 8.55: Portal Technology).

Configuring PeopleTools Options for Fluid Applications

The system-wide options for fluid applications appear on the PeopleTools Options page in the Fluid Mode Properties group box.

Image: PeopleTools Options: Fluid Mode Properties

This example illustrates the properties related to fluid applications on the PeopleTools Options page. Descriptions of the properties appear after the example.

Default Style Sheet Name	Displays the default, system-wide style sheet for fluid components, which is defined on the Branding System Options page. See "Administering System Branding" (PeopleTools 8.55: Portal Technology)
	Note: If no style sheet is specified, then PSSTYLEDEF_FMODE is used as the default style sheet for all fluid components.
Header Page Name	The default header page is PT_HEADERPAGE. If you want to override this page, enter your custom page.
Side Page Name	The default side page is PT_SIDEPAGE.

	If you want to override this page, enter your custom page.
Prompt Page Name	The default prompt page is PT_PROMPTPAGE.
	If you want to override this page, enter your custom page.
Footer Page Name	The default footer page is PT_FOOTER_SAVEONLY.
	If you want to override this page, enter your custom page.
Search Page Name	The default search page is PTS_NUI_SEARCH.
	If you want to override this page, enter your custom page.

Managing Fluid Homepages

Creating, configuring, and managing fluid homepages is similar to managing classic homepages, which is performed in the PeopleTools Portal administration interface. When working with fluid homepages, make sure you are familiar with the concepts and procedures of the portal registry.

Related Links
"Managing Homepages" (PeopleTools 8.55: Portal Technology)

Working with Fluid Homepages

PeopleTools provides a default homepage, named DEFAULT_LP. The underlying fluid page and component are both named PT_LANDINGPAGE. (The term landing refers to where the user "lands" after signing on to the system).

All fluid homepages are content references under the Fluid Homepage - Hidden folder. Set the content reference permission to control the access.

Default PeopleSoft User (PTPT1000) permissions are granted to the component. Managing and administering the system delivered fluid homepages requires the Portal Administrator permission list (PTPT1300).

Renaming a homepage is controlled by the All Rename option on the Homepage tab attributes for a homepage content reference.

When managing fluid homepages on small form factors (smartphones), keep these items in mind:

• Personalizing fluid homepages *is not* supported from smartphones.

• For small form factor devices, the system only displays fluid homepages that contain tiles optimized for small form factors. If a fluid homepage contains tiles optimized for small form factors, the system displays only those tiles optimized for small form factors on small form factor devices. (Has the Display on Small Form Factor Homepage option selected for the component properties). For example, assume a fluid homepage contains a mixture of "regular" tiles and tiles optimized for small form factors. If an end user accesses that fluid homepage using a smartphone, that user will see only the tiles optimized for the small form factor device.

See Setting Component Properties for Fluid Components for more information on fluid component properties.

Setting Device Homepage Defaults

By default, all devices are set to display the fluid homepage after an end user signs on to the PeopleSoft system. If the classic homepage is preferred for any or all device types, you can override the fluid homepage default using the Define Personalization page.

Navigation

Select PeopleTools, Personalization, Personaliztaion Options, select PPTL.

Image: Define Personalizations page

This example shows using the HPPC and HPTABLET options to set system defaults for homepage display.

Important! The HPPC and HPTABLET user options are system-wide settings. They cannot be exposed to end users to adjust as needed.

HPPC applies to laptop and desktop computers and HPTABLET applies to mobile devices. If you want to change the default homepage displayed for a device, click the Set Option Default Value link, and select the desired homepage.

Image: Set Option Default Value page

This example illustrates overriding the default homepage of *fluid* and setting it to *classic* for PC's (laptops and desktop computers).

Providing Guest User Access to Fluid Homepages

PeopleTools provides a feature where you can provide guest access to the system. This applies to both fluid and classic users where a user may not be tied to a user profile in the system, but can access specific pages in the system as needed. Configuring this option is discussed in detail in the product documentation for PeopleTools Portal Technology.

For more information, see "Implementing Guest User Access" (PeopleTools 8.55: Portal Technology).

Managing Mobile Device Access

PeopleTools delivers a default icon (LOGO_FAVICON) that will be used when a user adds (pins) a link to any PeopleSoft URL on the home screen of their mobile device. If you replace the PeopleTools default icon with a custom image, that customization will be overwritten after applying a PeopleTools update or upgrade.

         121

Chapter 8

Working with the Push Notification Framework

Understanding the Push Notification Framework

The Push Notification Framework updates the user interface with real time changes that have occurred on the server tier. The framework enables the PeopleSoft Internet Architecture to push data directly to the end user's display. To bring this feature to your PeopleSoft application, PeopleTools takes advantage of these elements in our server architecture:

- WebSocket push technology on the web server.

- Tuxedo event broker on the application server.

Image: Push Notification Framework

The following example illustrates how the Push Notification Framework architecture operates within these layers of the PeopleSoft infrastructure:

• Browser layer

• Web Server layer.

• Server Runtime layer.
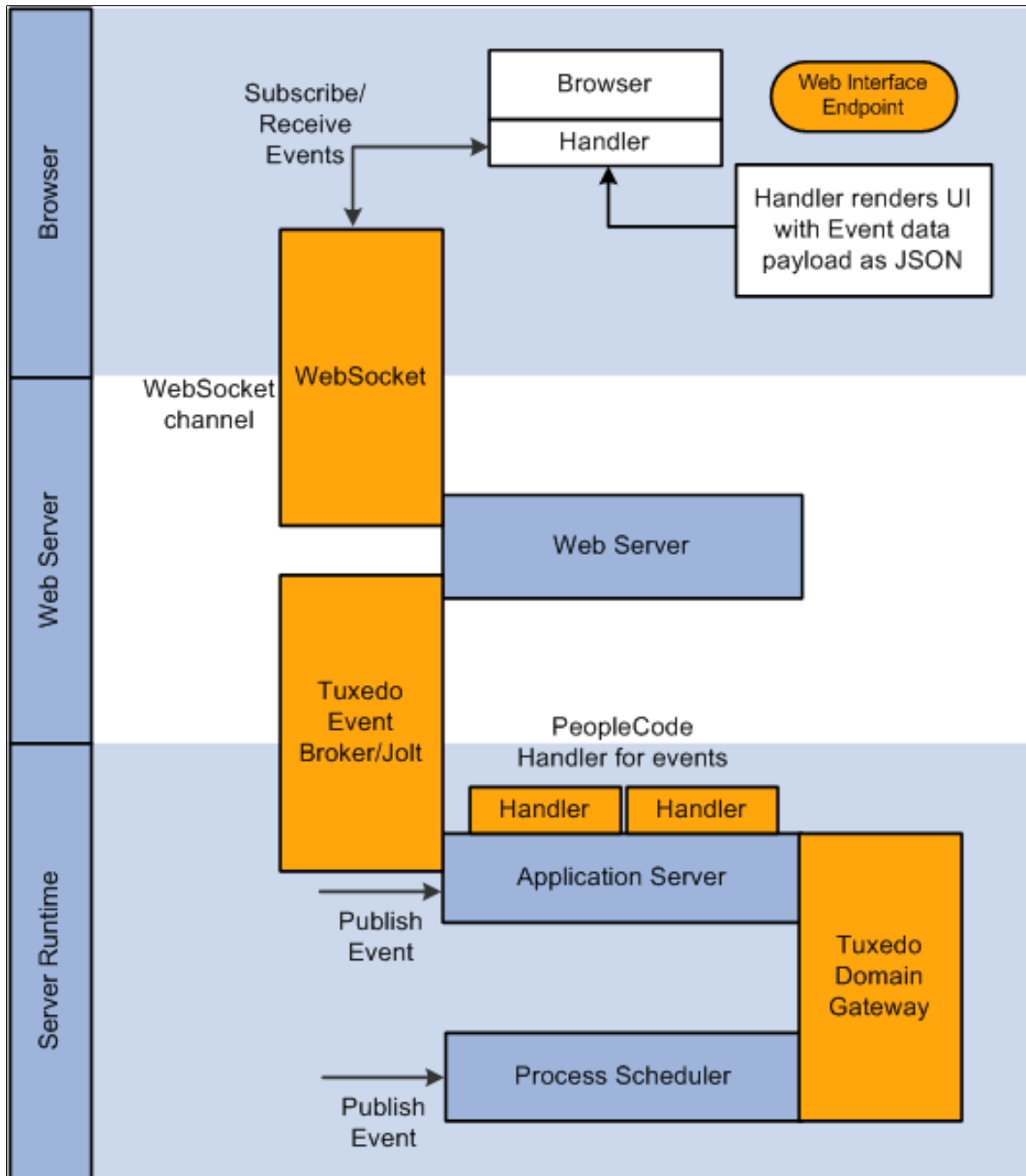
In the architecture shown the example, the Tuxedo Event Broker running on the application server interacts with websockets in web server to dispatch events from the server to the end user's browser.

The key elements in this framework are:

- Event: A state change in the system, causing the notification publication.

- Publisher: Publisher raises the event during the state change on the application server.

- Subscriber: Interested in events occurring on the server or browser.

- Websockets: Provides full duplex communication channels (simultaneous in both directions) between browsers and web servers.

Note: For the Push Notification feature to work, your browser should support the websocket technology. Newer browser versions support this technology.

- Tuxedo Event Broker: The Oracle Tuxedo Event Broker is provides asynchronous routing of application events among the processes running in a Oracle Tuxedo application. It also distributes system events to the application processes configured to receive them.

Development tasks for implementing the Push Notification feature can be categorized as follows:

Function	*Tasks*
Event Definition	- Define events. - Define event collections. - Associate events to collections.
Subscription	- Subscribe to an event. - Subscribe to an event collection.
Publishing	Use PeopleCode API.

System administration tasks for implementing the Push Notification feature include:

- Enabling Push Notifications in the active web profile for the PIA domain on the web server.

- Enabling Push Notifications for the application server domain(s).

- If events will be generated by processes run through Process Scheduler, you need to enable Push Notifications for the Process Scheduler domain(s), and the domain gateway feature between the Process Scheduler domain(s) and an application server domain enabled for Push Notification.

Setting Up Push Notification Configurations

This topic contains an overview and discusses:

- Enabling Push Notifications for the PIA Domain.

- Configuring an Application Server Domain for Push Notifications.

- Configuring Notifications from Process Scheduler.

Understanding Push Notification Configuration

To receive push notifications on a PeopleSoft system, you have to enable push notifications for the server domains where the events are being published. Events are published from application server domain and Process Scheduler domains. By enabling Push Notifications within a domain, the TMUSREVT server process will start in the domain (tmusrevt.exe), which is the Tuxedo Event Broker.

Note: Events raised in a Process Scheduler domain are ultimately routed through application server domain. For these inter-domain events, you have to set configurations for both the application server and the Process Scheduler.

This topic assumes you have a working knowledge of PeopleSoft server administration. For more information on managing PeopleSoft servers, see "Using PSADMIN" (PeopleTools 8.55: System and Server Administration).

Enabling Push Notifications for the PIA Domain

By default, Push Notifications are turned off for the PIA domain (web server layer). If Push Notifications are set at the server layer (application server and Process Scheduler), in order for the notifications to be distributed to the end user interface, you must enable Push Notifications for the PIA domain.

To enable Push Notifications for the PIA domain:

1. Select PeopleTools, Web Profiles, Web Profile Configuration.

2. Select the Custom Properties tab of the deployed web profile, add the custom property, *EnablePNSubscriptions,* and set it to *true.*

3. Save the web profile changes.

4. Restart the PIA domain for this change to take effect.

Configuring an Application Server Domain for Push Notifications

To enable Push Notifications in the application server domain:

1. Launch PSADMIN.

2. Select the appropriate domain(s) and select Configure this domain.

3. Enter *13* to enable Push Notifications from the Quick-Configure menu.

   The Push Notifications option is set to No (disabled) by default.

4. Enter *14* to load the modified configurations settings.

5. Boot (or reboot) the domain and verify that the TMUSREVT server process starts along with the other domain server processes.

Configuring Notifications from Process Scheduler

The inter-domain configuration allows events raised from Process Scheduler servers routed to application servers and finally routed to browsers. You can configure the following to allow push notifications from the Process Scheduler domain.

Image: Inter-Domain Event Configuration

This example illustrates the configuration settings for setting up inter-domain events between an application server domain and a Process Scheduler domain. The Process Scheduler port values set on both domains and the application server port values set on both domains must match to ensure successful transactions.
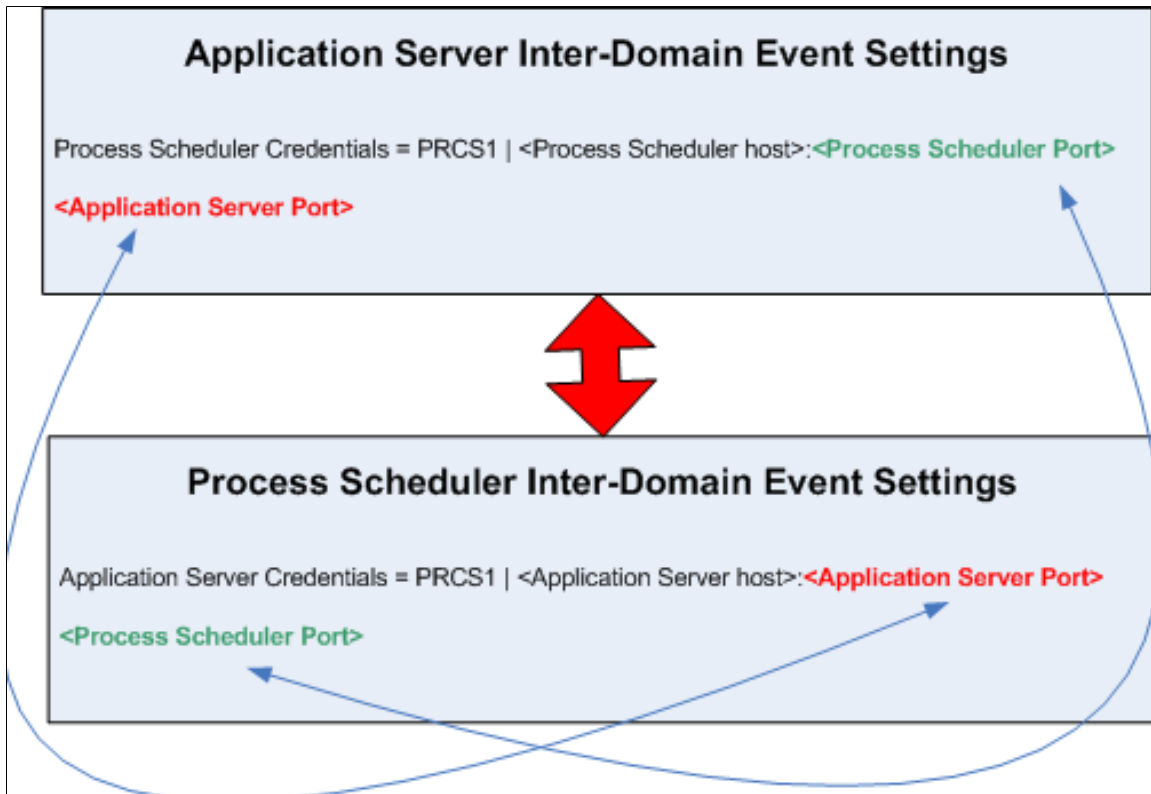
As a first step, you should configure the application server and Process Scheduler domains to generate inter-domain events configuration.

Generating Inter-Domain Events Configuration for an Application Server Domain

To configure inter-domain events for the application server domain:

1. Launch PSADMIN.

2. Select the appropriate application server domain and elect to configure the domain.

3. On the Quick-Configure menu, enable the Domains Gateway and Push Notifications options.

4. Select 14) Load Configuration.

Generating Inter-Domain Events Configuration for a Process Scheduler Domain

To configure inter-domain events for the Process Scheduler domain:

1. Launch PSADMIN.

2. Select the appropriate Process Scheduler domain and elect to configure the domain.

3. On the Quick-Configure menu, enable the Domains Gateway and Push Notifications options.

4. Select 6) Load Configuration.

When you complete these tasks, the system generates the tuxedo inter-domain events configuration for application server and Process Scheduler domains. Next, you need to configure values for the application server and Process Scheduler domains for inter-domain events.

Configuring the Process Scheduler Domain for Inter-Domain Events

To enable Inter-Domain Events for a Process Scheduler domain:

1. Launch PSADMIN.

2. Select the Process Scheduler domain and select to Configure this domain.

3. Enter the *Custom configuration* option.

4. Press the Enter key for Do you want to change any values (y/n/q)? [n], and continue to press the Enter key until the Values for config section – Inter-Domain Events appear.

5. Review and modify as needed the values in Inter-Domain Events.

```
Values for config section - Inter-Domain Events
    Process Scheduler Port=8988
    Application Server Credentials=APPDOM1|localhost:7988
```

Process Scheduler Port	Specify the port through which this Process Scheduler domain sends event notifications.
	Note: You can leave the port as it is if you are configuring a single Process Scheduler domain on a machine. If you are configuring multiple Process Scheduler domains on a machine for inter-domain events, then provide unique port number for each domain.
Application Server Credentials	Specify the application server domain to which this Process Scheduler domain will send event notifications in the format:
	`<domain name>|<hostname>:<application server port>`
	• domain name: Open the interdom.gbb file for the application server domain. The file is available in PS _CFG_HOME/appserv/<Domain>/folder. Locate the section in the file that is similar to the following:
	`*DM_LOCAL` `"AS_TESTSERV_41162" GWGRP=GWTGROUP ACCESSPOINTID="TESTSER`
	Copy the ACCESSPOINTID value and provide the configuration value as follows:
	`Application Server Credentials=TESTSERV_41162|<hostname>:`

- hostname: the host name of the machine, which is running application server domain.

- application server port: the port number provided in inter-domain events configuration for application server domain. You can leave the default value of the port as it is.

  The completed configuration values for the application server will look similar to the following example:

  ```
  Application Server Credentials=TESTSERV_41162|slc00fep:79
  ```

6. Enter *q* to exit the custom configuration interface, load any configuration changes, and return to the Domain Administration menu.

Note: If you have multiple process scheduler domains, then this configuration needs to be applied to each of the Process Scheduler domains.

Configuring the Application Server Domain for Inter-Domain Events

If the events occur by way of the Process Scheduler, then the system routes the events to an application server domain where they can be processed and distributed to the browser interface. For this to occur, you must enable and configure the settings for the Inter-Domain Events between the application server domain and the Process Scheduler domain.

When your environment consists of multiple application server domains, you can configure any one of the application server domains for inter-domain events. This ensures that any notification published from any of the Process Scheduler domains will be routed to the application server domain. Then, the events are routed to the connected PIA domain and finally to the browser clients.

To configure inter-domain events for the application server domain:

1. Launch PSADMIN.

2. Select to configure the appropriate application server domain and elect to Configure the domain.

3. Select the Custom Configuration option.

4. Press ENTER until you arrive at the Inter-Domain Events section.

5. Press *y* to change the values in Inter-Domain Events.

6. Review and modify as needed the Process Scheduler Credentials and Application Server Port settings.

   ```
   Process Scheduler Credentials=PRCS1|localhost:8988
   Application Server Port=7988
   ```

Process Scheduler Credentials	Specify the credentials for the Process Scheduler domain from which notifications will be received in the format:
	`<domain_name>|<hostname>:<process scheduler port>`
	- domain_name: Open the interdom.gbb file for the Process Scheduler domain. The file is available at PS_

CFG_HOME/appserv/prcs/<Domain>/ folder. Locate the section in the file that is similar to the following:

```
*DM_LOCAL
"PRCS" GWGRP=GWTGROUP1 ACCESSPOINTID="PRCS_SLC00FEP_PSNT"
```

Copy the ACCESSPOINTID value and provide the configuration value as follows:

```
Process Scheduler Credentials=PRCS_SLC00FEP_PSNT|<hostnam
```

- hostname: The host name of the machine where the Process Scheduler domain is running.

- process scheduler port: Process scheduler port is the port number provided in inter-domain events configuration for Process Scheduler domain.

The completed configuration values for the Process Scheduler domain will look similar to the following example:

```
Process Scheduler Credentials=PRCS_SLC00FEP_PSNT|slc00fep:89
```

Note: If there are multiple Process Scheduler domains from which you want to receive push notifications, then provide the configuration for each domain separated by a comma. For example: `Process Scheduler Credentials=PRCS_SLC00FEP_ PSNT|slc00fep:8988,PRCS_SLCFTJTH_PSNT| slcftjth:8988`

Note: If you have multiple Process Scheduler domains running on the same machine, then each Process Scheduler domain should have a unique port for inter-domain events configuration. For example: `Process Scheduler Credentials=PRCS_SLC00FEP_ PSNT|slc00fep:8988,PRCS_SLC00FEP_PSNT1| slc00fep:8989`

Application Server Port   You can retain the port number as it is.

Note: When you are connecting multiple process scheduler domains to a single application server domain for inter-domain events configuration, this is the recommended configuration.

---

7. Enter q to return to the PeopleSoft Domain Administration menu, and load any configuration changes.

   You have completed the configuration for push notifications on the application server domain and Process Scheduler domains. You need to boot the domains for the configurations to take effect.

Testing Push Notifications Published from Application Server Domain

To test the push notifications published from an application server domain:

1. Select PeopleTools, Portal , Inter Window Communication.

2. Search and select Portal Object Name: *PTSVRTESTEVENTS_GBL*.

3. In the Content Reference Message Events grid, click the Map icon for *SERVEREVENTTESTA*.

4. On the Map Event Data to Page Elements Modal window, click OK, and click Save.

    Note: You need to complete this only once on a new installation.

5. Using two different browser sessions navigate to PeopleTools , Utilities, Debug, Server Side Events Test Page.

6. Enter test values in the Event Data field and click Publish on one browser session.

7. On the other browser session, confirm that the values you entered previously, appear in the Event Targets fields.

Troubleshooting the Notification Configuration

If events are not appearing in the Notification window, check the following items:

*   Verify that the web profile passwords are consistent between the PIA domain and the application server domain.

*   Verify that your browser, web server, and any reverse proxy servers used all support websocket technology.

To troubleshoot the notification configuration for push notifications published from Process Scheduler domain, you also need to consider the following:

*   After booting up the domains, you can verify the configuration by ensuring that the domains have connected to each other. To verify the connection, you need to open the TUXLOG file for application server and Process Scheduler domains from their respective domain LOGS folders. You will see an entry in the file that the connection is established between the domains.

*   If you do not see connections being established, then reboot the domains after clearing IPC resources for the domains. Use the Clean IPC resources of this domain option in PSADMIN and then reboot the domains.

*   If you reboot application server domain or Process Scheduler domains after a successful configuration, the domains may not connect to each other anymore. This happens because the ACCESSPOINTID, which you had entered for the inter-domain configuration changes during a reboot. In this case, review the configuration for the domains and provide the correct ACCESSPOINTID.

Defining Events

Before subscribing or publishing an event, you must define an event. Events are state changes that occur on the server during system runtime. Examples of a state change include:

- Process Scheduler process or job status change.

- Approval notification for a workflow.

You can define events on the Define Server Side Events page.

Access the Define Server Side Events page by selecting PeopleTools, Push Notifications, Define Server Events.

Image: Define Server Side Events Page

The following example shows the Server Side Events grid where you add server side events.

Name	Enter a unique name for the event.
Description	Enter a description of the event.
Data Type	Select between KeyValue Pair or PeopleCode Rowset. The Data Type value defines how the system structures the event data while publishing the event. The value you assign determines what you enter for a Data Structure.
Data Structure	Enter the Data Structure value, depending on the Data Type selected.

- *KeyValue:* Enter key names separated with comma.

- *PeopleCode Rowset:* Click the Prompt button to select the record name.

APIs are provided by the framework to set event data (in the form of KeyValue pairs or PeopleCode Rowset) while publishing the event.

| **Adapters** | Adapters are used for sending push notifications to mobile applications running in the Oracle Mobile Application Framework (MAF). An adapter is associated with a server side event. |

For more information on adapters and working with MAF, see <u>Creating an Adapter</u>.

Recipients

(Optional) Click the Recipients link associated with a server side event to open the Add Recipients page, where you can add either users *or* roles (but not both) to receive notifications for that event. If no recipients are provided, the system sends the event notification only to users who have subscribed to the event.

Publishing an Event

You can publish an event from server runtime using the PeopleCode APIs provided by the framework. The procedure depends on the data type: KeyValue or PeopleCode rowset.

Publishing an Event for KeyValue Pairs

If you are publishing an event where the data type is a KeyValue pair you follow these general steps:

1. Set values for the keys defined for the event.

2. Add a recipient user who has subscribed to the event.

   Note: Adding recipients to the event is optional. If you add users or roles using the APIs as recipients then the recipients added through web interface while defining the event are ignored.

3. Publish the event.

Example

This example demonstrates the publishing of the event PROCESSSTATUSCHANGE with the keys PROCESSID and RUNSTATUS and recipient user, VP1.

1. Create the event object.

   ```
   Local object &EventObject;
   &EventObject = CreateObject("PSEvent");
   ```

2. Set values for the keys defined in the event

   ```
   &EventObject.SetKeyValuePair("PROCESSID", PSPRCSSTATUS.PROCESSID);
   &EventObject.SetKeyValuePair("STATUS", PSPRCSSTATUS.RUNSTATUS);
   ```

3. (Optional) Add a recipient user for the event.

   In this case, the event will be sent only to user VP1 (if VP1 has subscribed to the event).

   ```
   &EventObject.AddRecipient("VP1", 1)
   ```

   The the parameters for AddRecipient are:

recipient	Specify the recipient using either a user name or a role name.
recipient type	Use a value of 1 to indicate a user.
	Use a value of 2 to indicate a role.

4. Publish the event.

```
&EventObject.Publish("PROCESSSTATUSCHANGE");
```

Publishing an Event for a PeopleCode Rowset

If the data type is PeopleCode rowset, you set the values for the fields in the record and add a recipient.

Example

This example illustrates publishing an event for a PeopleCode rowset. In this example, WORKFLOWAPPROVAL is the server event for which name and status fields are defined. All users assigned the role of Workflow Administrator are added as recipients.

1. Create the event object.

```
Local object &EventObject;
&EventObject = CreateObject("PSEvent");
```

2. Set the PeopleCode Rowset for the event.

```
Local object &EventObject;
&rs = CreateRowset(Record.PSPTWKLAPPVL);
&rs.InsertRow(&i);
&rs(&i).PSPTWKLAPPVL.NAME.Value = "Test Workflow";
&rs(&i).PSPTWKLAPPVL.STATUS.Value = 1;
&EventObject.SetRowsetData(&rs);
```

3. (Optional) Add a recipient role for the event.

   In this case, the event will be sent only to users who have subscribed to the event and belong to the role: Workflow Administrator.

```
&EventObject.AddRecipient("Workflow Administrator", 2);
```

4. Publish the event.

```
&ret = &EventObject.publish("WORKFLOWAPPROVAL");
```

Subscribing to an Event

This topic provides an overview of how you can create subscriptions to an event.

Subscribing to an Event through Interwindow Communication (IWC)

Use IWC to subscribe an event occurring on a PeopleSoft page, where the event model is mapped to page elements, such as fields and grids. Creating a subscription does not require any changes to the page. When the event is published from the server, the page is updated automatically with the event data.

See event data structure in, Defining Events.

To subscribe to an even through IWC:

1. Identify the component which needs to subscribe to the event.

2. Navigate to the content reference for the component using the portal Structure and Content interface (PeopleTools, Portal, Structure and Content).

3. Click theIWC Message Events link on the Content Ref Administration page of the component.

4. Select Add a New Value, and add a new row on the IWC Message Events page, specifying these values:

    •   Event Name: Enter a name to identify the event.

    •   Message Event Type: Select *Server Sub*.

5. Click the Map to display the Map Events Data to Page Elements page.

Image: Map Events Data To Page Elements Page

In this page you can map the data model of a published Event to a page element defined as *<Record Name>_<FieldName>*.

Event Name	Displays the Event Name that is defined in Content Reference Message Events page.
Page Name	Select the page where you want the event subscription added.
Page widget Type	Select from field or grid widget type inside the page which is auto-updated with the published event data.
Event Data	Select the meta-data of the event. The drop down will display key values if the event data type is KeyValue Pair and record fields if the event data type is PeopleCode Rowset.
Page Record Field Name	Select the record field name for the page element, which will update.

Related Links
"Publishing and Subscribing to IWC Messages" (PeopleTools 8.55: Portal Technology)

Subscribing to an Event Using JavaScript

When using programmatic subscription, you are able to render the published event data on the page in any form using the JavaScript Object Notation (JSON) data-interchange format. Use the following JavaScript API on an HTML embedded section on a page on the web interface to receive Push Notifications.

```
Subscribe(event name, callback function name)
```

event name	The event to be subscribed to.
callback function name	JavaScript function invoked when the event is published on the server.

Example

```
<script>
function OnEvent(EventName,EventData)
 { //EventName is the name of the published event
//EventData is the payload (or data) of the published event in
//JSON(JavaScript Object Notation) form.}
Subscribe("PROCESSSTATUSCHANGE",OnEvent)
</script>
```

Subscribing through Application Server Runtime

You can also subscribe to events from server runtime itself. This is beneficial for cases when you want to write the published event data to a database table or write to a log file.

Write a PeopleCode handler for subscription of an event, which also acts as the callback method for the event.

Subscription Code

```
Local object &EventObject;
&EventObject = CreateObject("PSEvent");
SubscriptionHandle = &EventObject.Subscribe("PROCESSSTATUSCHANGE", "PT_PRCS:Process⇒
```

```
Monitor:OnEvent");
```

The second parameter for the Subscribe method is the fully qualified name of a callback function in the form:

```
Application Package:Application Class:Function Name
```

This function is invoked when the event is published.

Syntax of a callback function

```
 method OnEvent(&EventObject As object);
end-class;
```

The &EventObject parameter represents the published event data.

Working with Event Collections

This topic provides an overview of the executables for working with event collections.

Understanding Event Collections

You can group a set of events that exhibit a common behavior as a collection. When you subscribe to a collection, you receive notifications for all the events included in the collection whenever they are published from the server runtime.

Defining an Event Collection

Use the Define Server Event Collections page to define an event collection.

Access PeopleTools, Push Notifications, Define Event Collections.

Image: Define Server Event Collections Page

This example illustrates how you can define an event collection.

	*Collection Name	*Description	Root Package ID	Application Class ID	Events		
1	PUSHNOTIFICATIONWINDOW	Push Notification Window	PTPN_WINDOW_HANDLER	NotificationWindowHandler	Events	+	−
2	SERVERTESTEVENTS	Server events test collection	PT_PNCOLL_HANDLER	ServerEventsTestHandler	Events	+	−

Collection Name	Enter the unique name for the event collection.
Description	Enter the description of the event collection.
Handler - Application Package	(Optional) Select the PeopleCode application package that contains the'Handler - Application Class.
Handler - Application Class	(Optional) Select the PeopleCode application class that can receive the events which are part of the event collection when they are published. A valid handler class implements the interface PT_PNCOLL_HANDLER:Handler.
Events	Click the link to add events to the event collection.

When you click Events for a collection, the Add Events page appears where you add events to the collection. If you subscribe to the collection, then you will receive the push notifications for all the events added in this page.

Subscribing to an Event Collection

You can subscribe to an Event Collection using the following JavaScript API.

```
SubscribeCollection(collection_name, callback_function_name)
```

collection_name	The name of the event collection to which you are subscribing.
callback_function_name	The function invoked if any of the events included in the collection are published on the server.

Collection-name is the name of the event collection to be subscribed. Callback function name is t

See Subscribing to an Event Using JavaScript for an example of how to add the callback function.

Using Application Data Set Definitions for Events

These Application Data Set (ADS) definitions are for events and collections. You can use them to copy event and collection definitions between databases.

PTSVR_EVENTS	ADS definition for events for copying event definitions and their corresponding IWC subscription configurations.
PTPN_EVENT_COLLECTION	ADS definition for event collections for copying both collection definitions as well as all the associated event definitions for that collection.

Related Links
"Understanding the PeopleSoft API Repository" (PeopleTools 8.55: PeopleCode API Reference)

System Requirements for Push Notification

Before making use of the Push Notification feature, verify that your infrastructure support WebSockets.

System Component	*Requirement*
Web Server	Weblogic 12.1.2 onwards. Not supported currently on Websphere.
Reverse Proxy Support	OTD from 11.1.1.7, WLS Plug-In 12.1.2+ for Apache 2.2, 2.4, OHS 12.1.3

Configuring Push Notification to Run within Oracle Mobile Application Framework

This topic provides an overview and describes the tasks to complete to enable push notifications to run within the Oracle Mobile Application Framework (MAF) on mobile devices, such as smartphones and tablets.

Understanding Push Notification and MAF

PeopleTools enables the Push Notification Framework to allow the PeopleSoft server runtime to push real-time notifications to MAF applications running on Android and iOS devices. The following example depicts the sequence of interaction for mobile devices configured for MAF and the PeopleSoft system.

Image: Push Notification using MAF

This example illustrates the sequence of events for establishing push notifications on a mobile device. The high level steps are described below.
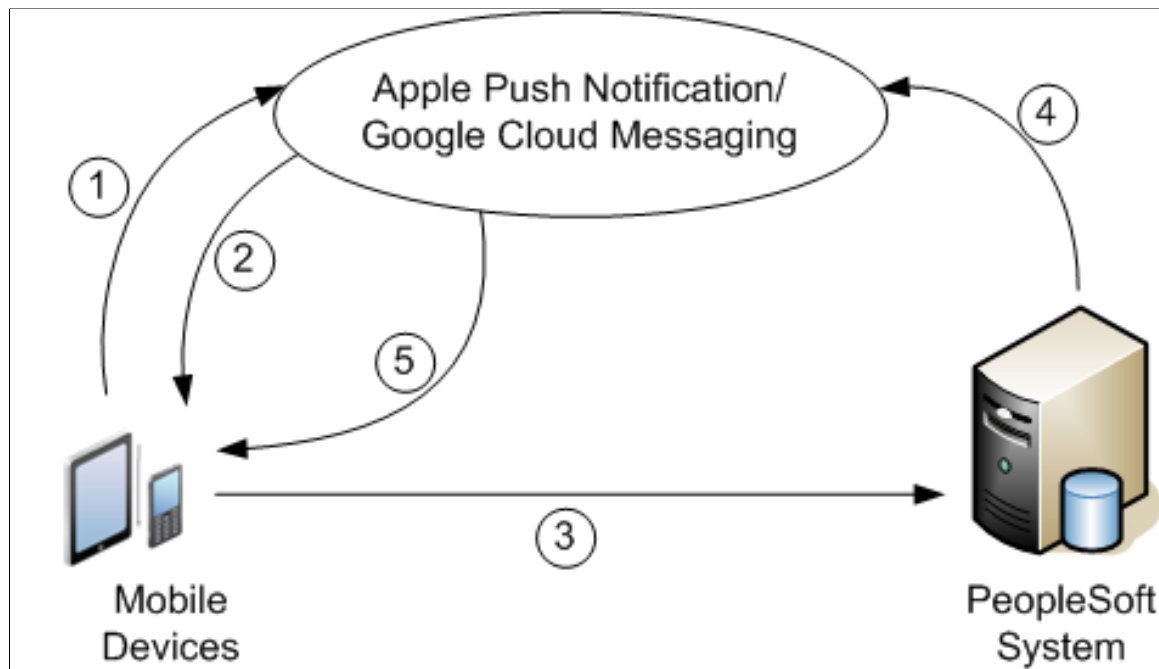
1	MAF application registers with the push notification provider: Google Could Messaging (GCM) for Android android or Apple Push Notification service (APNs) for iOS.
2	MAF application receives the registration ID (or device token) from the provider.
3	Mobile device passes registration ID to the PeopleSoft server.
4	When a defined server event occurs, PeopleSoft server runtime sends notification to the push notification providers.

5	Push notification provider pushes the notification to the MAF application.

Before you get started configuring push notifications for PeopleSoft on mobile devices, you need to:

*   Enable Integration Broker.

    See "Understanding PeopleSoft Integration Broker Administration" (PeopleTools 8.55: Integration Broker Administration).

    Note: If the application server is running inside a secure private network (such as VPN), the proxy server host and port needs to be configured in the Integration Broker Gateway configuration properties file. This is essential for the PeopleSoft server runtime to connect to the APNs server, which listens on particular port numbers that may be blocked by the internal firewall.

    ---

*   Install the MAF elements provided with your PeopleTools license.

    See Understanding Mobile Application Framework Support for PeopleSoft.

Gathering Push Notification Provider Information

Depending on your mobile device operating system, you will need a set of attributes related to your MAF application to configure push notification for your PeopleSoft application.

Required Attributes for Android

For the Android device, you need to provide these MAF application attributes:

*   Application Name

*   GCM Project Number

*   GCM API Key

The GCM Project Number for a MAF application is embedded into the application when it is packaged. The GCM API Key is used to establish connection to the provider. These attribute values are generated through Google API Console.

Required Attributes for iOS

For the iOS device, you need to provide these MAF application attributes:

*   Application Name

*   APNs Bundle ID

*   APNs Certificate (.p12 file)

*   APNs Certificate password

The APNs Bundle ID is a unique identifier provided by Apple for the MAF application, which is embedded into the application when it is packaged. PeopleSoft server runtime requires the APNs SSL Certificate (.p12 file) and the Certificate password to establish connection to the APNs server to send the notification. The APNs SSL Certificate is generated through iOS provisioning portal.

Configuring a MAF Application in PeopleSoft

Access the Configure MAF Applications page to register your MAF applications within PeopleSoft.

PeopleTools, Push Notifications, Mobile Notifications, Configure MAF Applications

Image: Configure MAF Application page

This example illustrates the fields and controls of the Configure MAF Application page.

Add the MAF Application Name for your application, and also add the required attributes for the provider through which you will receive push notifications on your MAF application running on a device.

If your MAF application is running on an android device, then you need to configure the GCM (Google Cloud Messaging) attributes for the MAF application, such as, GCM Project Number and GCM API Key.

If your MAF application is running on an iOS device, then you need to configure the APN (Apple Push Notifications) attributes for the MAF application, such as, APN Bundle ID, APN Private Key File Name, and APN Private Key Password.

Note: The GCM settings, discussed earlier, are also applicable if you are using Firebase Cloud Messaging (FCM), which is the new version of GCM.

See Gathering Push Notification Provider Information.

Use the Push Notification administrator credentials to configure MAF applications: user ID *PTPNADMIN* with password *PTPNADMIN*. The PTPNADMIN user ID has as one of its roles, *PN Administrator* (Push Notification Administrator).

The Push Notification administrator can enable or disable push notifications to a MAF application by selecting or deselecting the Enabled check box.

For APNs registration, you need to update the Integration Broker gateway properties file also. The exact path where the APNs Private key file is stored should be assigned to the ig.apnKeyStorePath property. For example:

```
ig.apnKeyStorePath=/usr/opt/APN
```

Configuring Push Notifications to Google Cloud Messaging (GCM) Through An HTTS Connection

PeopleSoft provides an Integration Broker service operation — PTPN_GCM_OPERATION_POST — to post push notifications to the Google Cloud Messaging (GCM) server.

To configure an HTTPS connection, complete the following steps:

1. Select PeopleTools, Integration Broker, Integration Setup, Service Operations, and select the PTPN_GCM_OPERATION_POST service operation.

2. On the Service Operations page, in the REST Resource Definition area, set the REST Base URL field to *https://gcm-http.googleapis.com*.

3. Add a certificate in the Integration Gateway keystore that validates the GCM SSL certificate because Integration Gateway needs to validate the SSL certificate of the GCM server.

   - Navigate to the location where the Integration Gateway keystore is located — (PS_CFG_HOME/webserv/peoplesoft/piaconfig/keystore).

   - Use the keytool utility to import the certificate into the keystore. For example, if the certificate name is geotrustca.cer, your command will be similar to the following.

     ```
     keytool -import -alias geotrust -file geotrustca.cer -keystore pskey
     ```

Creating an Adapter

The PeopleSoft server runtime sends push notifications to MAF applications using adapters. An adapter is a PeopleCode application class that uses Push Notification Framework APIs to send push notifications to MAF applications.

For more information on Push Notification Framework PeopleCode, see "Understanding the PSEvent Class" (PeopleTools 8.55: PeopleCode API Reference).

The following example illustrates an adapter class implemented for a PeopleSoft push notification event named SENDNOTE. Extensive comments are given in this example so that it can be used as a reference for writing adapter classes.

```
<* This is an example adapter class for pushing event: SENDNOTE to
MAF application.

Developers can use this class as a reference to implement custom adapter
classes for their events.

A custom adapter class extending from "MAFApplicationAdapter" should
have a constructor and implement the method "onNotification".  *>

import PTPN_ADAPTERS:MAFApplicationAdapter;

class SendNoteAdapter extends PTPN_ADAPTERS:MAFApplicationAdapter

    method SendNoteAdapter();

    method onNotification(&eventObject As PSEvent);

end-class;



<* Invoke MAFApplicationAdapter constructor by passing the MAF application
name, to which push notifications for SENDNOTE event is to be sent
and the fully qualified  name of the adapter class.  *>

method SendNoteAdapter

    %Super = create PTPN_ADAPTERS:MAFApplicationAdapter("PeopleSoft",
"PTPN_ADAPTERS:SendNoteAdapter");
end-method;



<* onNotification method will receive the published event instance:
eventObject.
```

Developers can use methods in PSEvent class to retrieve data from
the published event instance.

Use MAFApplicationAdapter methods and properties to set notification
data and send the push notification to the MAF application.

SendNotification method will return an error text incase of failures.
Refer PeopleBooks for methods and properties of PSEvent class. *>

method onNotification   /+ &eventObject as PSEvent +/;   Local
string &alert;   Local string &eventDataKey = "DETAIL";   Local
string &response;   &alert = &eventObject.GetValueOfKey(&eventDataKey);
 %Super.SetAlert(&alert);   &response = %Super.SendNotification();

end-method;

The MAFApplicationAdapter class provides the following APIs to construct a push notification to a MAF
application.

API	Description
SetAlert(altert-text)	Alert text will be displayed as an alert message on the device receiving the notification.
SetSound(sound-file)	Plays an alert sound on the device receiving the notification. For example: SetSound("default")
SetIOSBadge(badge-number)	Displays the badge-number on the device, which receives the notification (works only on iOS devices). For example: SetIOSBadge(1)
SetUrl(url)	Sends a URL as part of the notification payload in the case where the user may have to navigate to a page to perform a task based on the notification.

Associating an Adapter to a Push Notification Event

PeopleTools, Push Notifications, Define Server Events, Click the Adapter icon

143

Image: Associating an Adapter to an Event

This example illustrates the fields and controls on the page used to associate adapters to events.

After clicking the Adapter icon for the desired event, use the Adapter drop-down to select the appropriate adapter. Add rows as needed.

The adapter will be executed when a selected event occurs in the PeopleSoft application server run-time.

Enabling Notifications

To enable notifications to a device use the Enable Notifications page.

PeopleTools, Push Notifications, Enable Notifications

Image: Enable Notifications Page

This example illustrates the fields and controls on the Enable Notifications page.

End users can configure which push notification events can send notifications to the MAP application on their phone.

Registering Devices

After an end user downloads the MAF application and registers for push notifications, the provide sends a device token and device information, which is passed to the PeopleSoft when the user signs in to the system. When the device token and device information is stored in the PeopleSoft Database, appears in the Register Device page, which displays a row for each new device token received.

Access PeopleTools, Push Notifications, Mobile Notifications, MAF Register Device

Image: Register Device Page

This example illustrates the fields and controls of the Register Device page.

Register Device					
					1 rows
	Device Name ◇	Application Name ◇	User Name ◇	Registration Status ◇	
1	srikbhat's iPad	Push Notification	PTDMO	Success	—

Save

# Using the Push Notification Window

The Push Notification framework helps in consolidating and updating new events in the PeopleSoft system in real time. When you login to the portal, you can see the number of new events or messages using the Notifications icon. When you press the Notifications icon, the Push Notification window appears below.

The system updates notifications as they occur (in real time) even if you have navigated away from the window or are currently displaying the window.

Note: Notifications are displayed in the language of the notification approver, which is specified on the General tab of the User Profiles page (PeopleTools, Security, User Profiles, User Profiles). For example, if the notification approver's language is set to English, the notifications will be displayed in English. When a notification is generated, the notification remains in that language and does not change depending on the user's language preferences or current PIA session language.

Image: Push Notification Window

The Push Notification window has list of new events and messages from the PeopleSoft system.
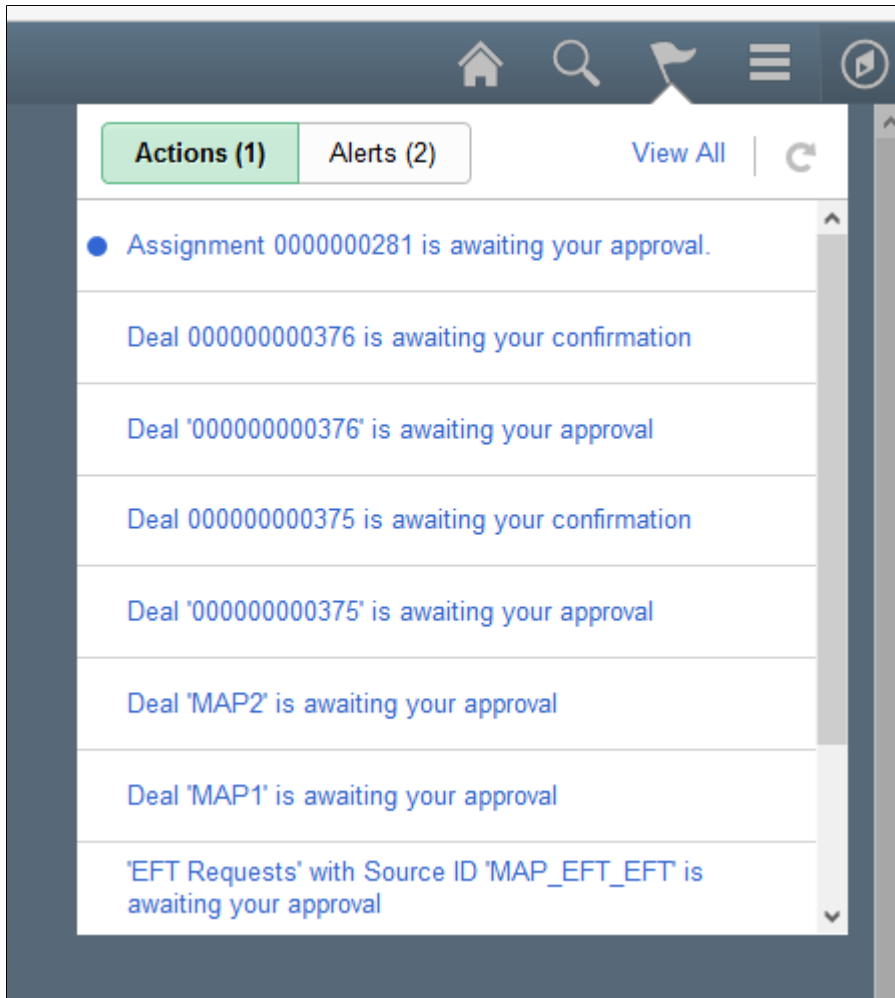
Actions	Select this tab to see the Actionable items. All actionable notifications are grouped under this tab.
Alerts	Select this tab to see all the informational messages like Alerts.
View All	Click the link to go to View All Notifications page.
●	Displays beside a notification item, indicating that it is new.
↻	Refreshes the notification window and lists all new notifications. The Refresh icon becomes enabled when you are on the Push Notification window and any new notification arrives.
‹	Opens the Actions Panel with Dynamic Label links. This icon is only visible when there are Dynamic Labels defined for that item.
Dynamic Labels	Use to take an action on the notification.

✕	Dismisses an alert. This icon is visible only for alerts on the Alert tab.

In both the Actions tab and Alerts tab, numbers of new items are displayed within parenthesis. Any notification item linked to a content reference is displayed as a hyperlink. Use the link to open the corresponding PeopleSoft page.

Viewing All Notifications

Use the View All link on the Notifications window header to open the View All Notifications page. Use the Show Filters button to expand the left pane. Use the left pane for faceted filtering of notifications.

Image: View All Notification Page

The View All Notifications page lists all Actions and Alerts together.

The system displays these filter categories.

Category Name	Allows filtering based on categories like Expense Approval, Promotion Code, Journal Approval, and so on.
Category Type	Allows filtering based on Actions or Alerts.
Message State	Allows filtering based on these message states: Read, Unread, and Dismissed.

Use these controls to work with notifications:

Select All	Enables the Actions button that allows you to take action on bulk notifications.
Actions	Displays the bulk actions popup menu where you can select an option to make the notifications read, unread, or dismissed.
Row Level Action	Displays the Dynamic Labels popup menu. Press a dynamic label to take a action on a notification item.

> **Note:** The Row Level Action button is visible only at the end of the row of a notification item if the item has Dynamic Labels defined.

Configuring the Notifications Window

This topic discusses:

- Setting System Configuration Options.

- Managing Notifications.

- Setting User Configuration Options.
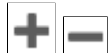
Setting System Configuration Options

Access the System Configurations page to configure the Notification Window. These options are set by a system administrator for end users of the system.

Select PeopleTools, Push Notifications, Notifications, System Configurations.

Image: System Configuration Page

The System Configuration page is used to configure Notification Window.

The table describes the names and definitions of fields in the System Configuration page.

Max Notification Items	Enter the number of notification to be displayed to the user. Default value for this option is 10.
Notification Items to Load	Enter the number of records that gets loaded to the system when the notification window is loaded. The default value for this option is 100. This configuration can be used to tune the performance of the notification window.
From Date, To date	Set the period to select the notification items for further actions.
Category Name	Select the category to archive or purge data.
+ —	Click to add or delete a category.
Archive	Click to archive the data based on the date and the category name. When notification items are archived, the data is copied into the archive table. Archiving the data at regular intervals

|	improves the system performance because the number of notification items in the table gets reduced on archival.
Purge Data	Click to purge the data based on the date and the category name. You cannot retrieve purged data. Purging the data at regular intervals improves the system performance because the number of notification items in the table gets reduced on purging.

Managing Notifications

Select Manage Notifications tab to access the Manage Notification page.

This page provides system-level configuration, allowing administrators to enable or disable TriggerBusinessEvent notification selectively based on the business process, activity name, event name and worklist name. By default, these options are disabled for all combinations of business process, activity name, event name and worklist name. When you enable TriggerBussinessEvent, you can then enable and apply options to each combination.

Image: Manage Notification page

This example illustrates the fields and controls on the Manage Notification page. Descriptions can be found below the example.
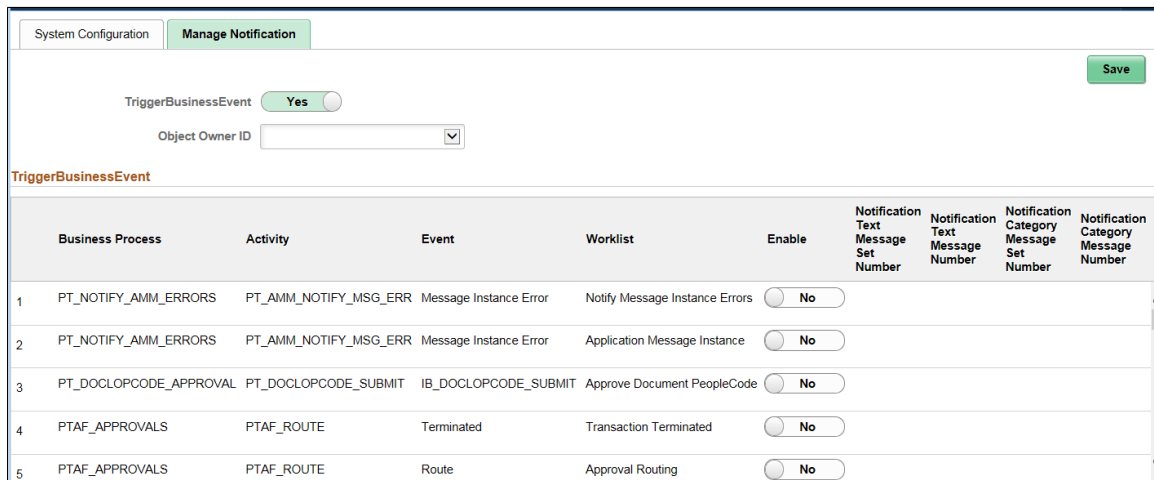
Enable	Slide the slider to enable or disable the notification feature for that business process combination.
Notification Message Set/Message Number	For each combination of business process, activity, event, and worklist, you can set the message set ID and number, based on which message catalog entry used to form the notification message and notification category.

Setting User Configuration Options

Access the User Configurations page to configure the Notifications window as an end user.

Select PeopleTools, Push Notifications, Notifications, User Configurations..

Image: User Configuration page

This example illustrates the User Configuration page, which end users access to set user options.

Notification Items to Display	Enter the number of notification items to display on the Notifications window.
Categories to Hide	Click the Add icon to add categories to prevent from being displayed. Any notification item from these categories is not displayed on the Notifications window.

Related Links

Understanding the Push Notification Framework
Fluid User Interface Overview

Chapter 9

Deploying Applications Using the Oracle Mobile Application Framework

Understanding Mobile Application Framework Support for PeopleSoft

PeopleTools provides support for customers who want to incorporate Oracle Mobile Application Framework (MAF) for their PeopleSoft Fluid or MAP deployments.

This documentation describes how to locate the elements and information required for using this option.

You can use Oracle MAF as described in the Mobile Application Framework and PeopleSoft license agreements. Using MAF *is an option* for deploying PeopleSoft Fluid User Interface applications or applications built with PeopleSoft Mobile Application Framework to mobile devices (smartphones and tablets).

Note: MAF *is not* required for deploying PeopleSoft applications to mobile devices. It is an option some customers may choose to pursue.

PeopleTools provides a restricted use of the Mobile Application Framework, which enables you to use *only* the Remote URL option for running PeopleSoft applications in the MAF container on the mobile device. An additional MAF license is required if you intend to use other MAF features or create new data controls.

This list outlines the main steps you need to complete to implement the MAF deployment option for PeopleSoft applications:

- Download the version of MAF provided in the PeopleTools Media Pack.

- Download the mobile application archive (.maa) provided with PeopleTools.

- Import the delivered mobile application archive into your IDE and modify as needed. JDeveloper is the recommended IDE. Eclipse may also be used, but see the note below.

- Deploy the application for end user download. For internal usage, deploy the application to your site's application repository. For external usage, deploy the application to Apple App Store or Google Play Store.

Note: Currently Oracle delivers the PeopleSoft mobile application archive (.maa) for JDeveloper. Eclipse may also be used, but you need to install Oracle Enterprise Pack for Eclipse (OEPE) and build the .maa manually.

Preparing to Use MAF for PeopleSoft Applications

After installing the current version of PeopleTools, before you begin working with MAF for PeopleSoft applications, you will need to have completed the tasks described in this topic.

Note: The version of MAF and JDeveloper used with your version of PeopleTools is important. Make sure to review the release notes, feature overview, licensing documentation, and Certifications on My Oracle Support to make sure you use the appropriate version combinations.

Using an Integrated Development Environment (IDE)

To configure and deploy a mobile application using the delivered .maa file, you will need to import the .maa file into an IDE. When working with Oracle MAF, currently, the primary IDE to use is JDeveloper.

Oracle delivers the PeopleSoft mobile application archive (.maa) for JDeveloper. Eclipse may also be used, but you need to install Oracle Enterprise Pack for Eclipse (OEPE) and build the .maa manually.

Note: This document does not seek to replace any documentation already existing for JDeveloper or Eclipse. It is assumed that if you are implementing the mobile application archive delivered by PeopleTools that you have a working knowledge of at least one of the IDE products listed in this document. This document outlines and describes the tasks required to configure and deploy the delivered .maa file in general terms. Always refer to the product documentation for your IDE for specific information.

Note: This document is written from the perspective of using JDeveloper. If you elect to use Eclipse, you are expected to be familiar with that product and use its documentation to achieve the same objectives described here for JDeveloper.

Installing MAF

Important! Only install the version of MAF supported with your version of PeopleTools. While there are more current versions of MAF available, install *only* the supported version of MAF for working with the delivered .maa file in your PeopleTools installation.

Note: MAF comes bundled with Oracle Enterprise Pack for Eclipse. You need to install the specific version of OEPE that includes the version of MAF supported by your PeopleTools version.

Downloading MAF

In most circumstances, MAF is already bundled with the PeopleTools files that you downloaded from the Oracle Software Delivery Cloud (OSDC). Therefore, you will not need to download MAF separately. This ensures that you receive the correct version of MAF supported with your version of PeopleTools. Store the MAF download within a local file system accessible from your JDeveloper installation.

In rare circumstances, the version of MAF you require is not yet available on OSDC and has not yet been bundled with PeopleTools. *In those circumstances only,* you can download the supported version of MAF from the Oracle Technology Network (OTN):

1. Download the supported version of MAF from OTN:

   http://www.oracle.com/webfolder/technetwork/jdeveloper/downloads/1213center.xml#oracle.maf

2. Store the MAF download within a local file system accessible from your JDeveloper installation.

Installing MAF from JDeveloper

To install MAF using JDeveloper:

1. In JDeveloper, select Help > Check for Updates.

2. On the Select update source screen, select Install From Local File, provide the path to the zip file containing the MAF download, and click Next.

   Image: JDeveloper — Installing MAF

   This example illustrates the interface used to install or update MAF using JDeveloper.
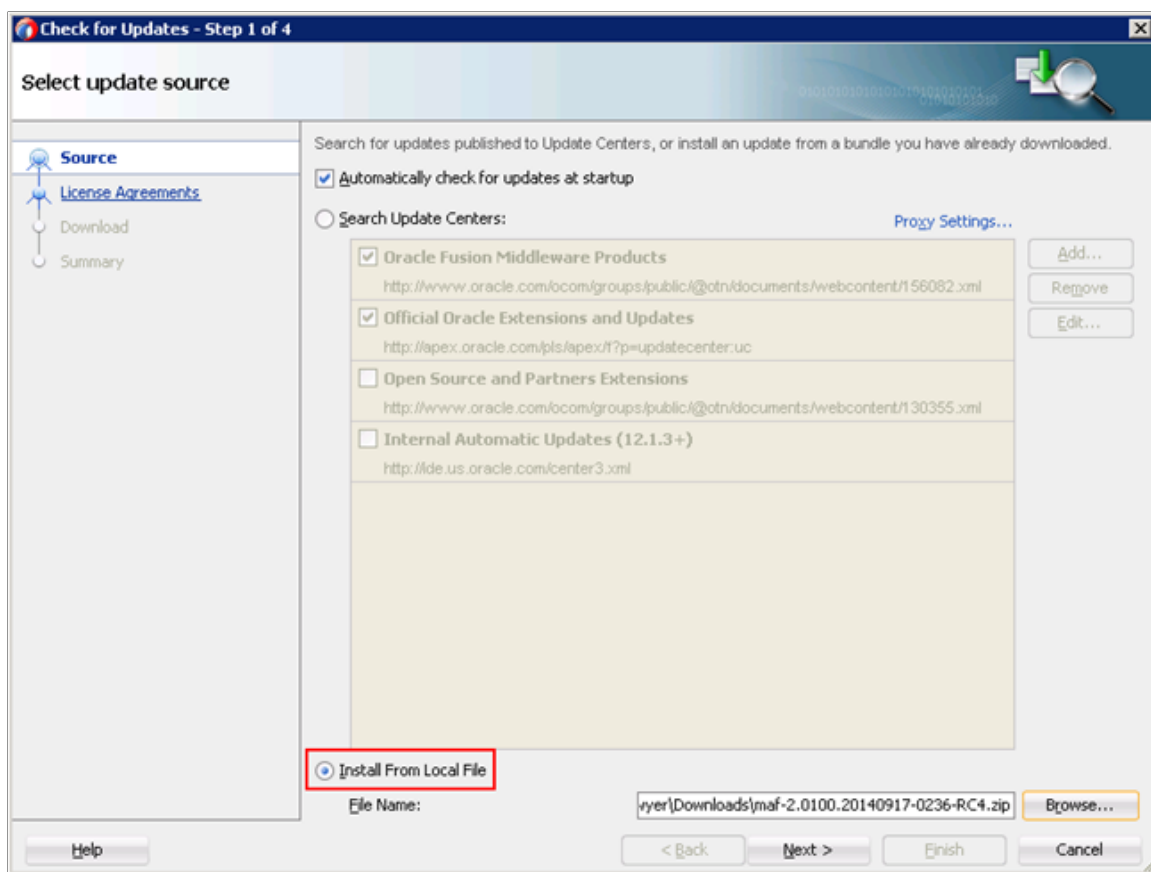
   Note: When using the *Check for Updates* option from within JDeveloper to install MAF, you need to specify *Install From Local File.* Using the *Search Update Centers* option applies the latest MAF version, which
   most likely is not the version certified for PeopleSoft use.

3. On the License Agreements screen, click *I Agree* (if the terms are suitable for your site), and click Next.

4. On the Summary screen, review your selections, and click Finish.

5.   To complete the MAF installation, you will need to restart JDeveloper.

Installing Target Operating System Development Requirements

Depending on the target operating system of the devices to which you will deploy the MAF application, you need to install any required development artifacts, such as SDK's and so on.

Required Elements for iOS

For iOS, you will need these items in place in your development environment:

•   Xcode 6

    http://developer.apple.com/xcode/

•   iOS SDK

    http://developer.apple.com/devcenter/ios/

•   Provisioning profile

•   Any required client-side certificates

For more information see:

•   Setting Up Development Tools for iOS Platform

•   Provisioning and Development

Required Elements for Android

For Android, you will need to install the Android SDK. See:

•   Installing the Android SDK

•   Setting Up Development Tools for Android Platform

# Importing the Delivered Mobile Application Archive

After installing PeopleTools, there is an additional folder in the PS_HOME\setup directory, named \MAF. This is where the PeopleSoft mobile application archive file (.maa) resides.

To import the PeopleSoft archive file:

1.   In JDeveloper, select File > Import.

2.   From the Import dialog box, select *MAF Application from Archive File,* and click OK.

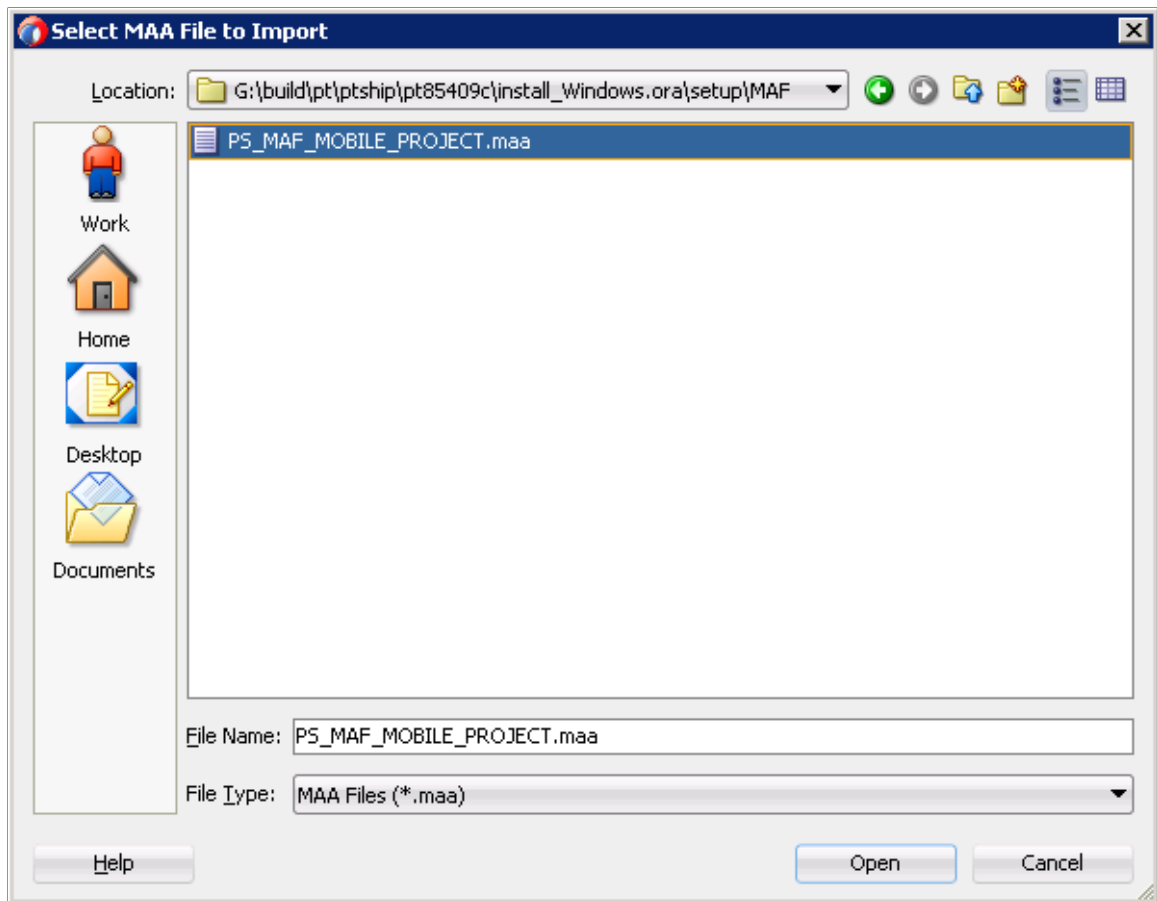**Image: Importing a MAF Application From Archive**

This example illustrates using the Import dialog box to import the .maa file.
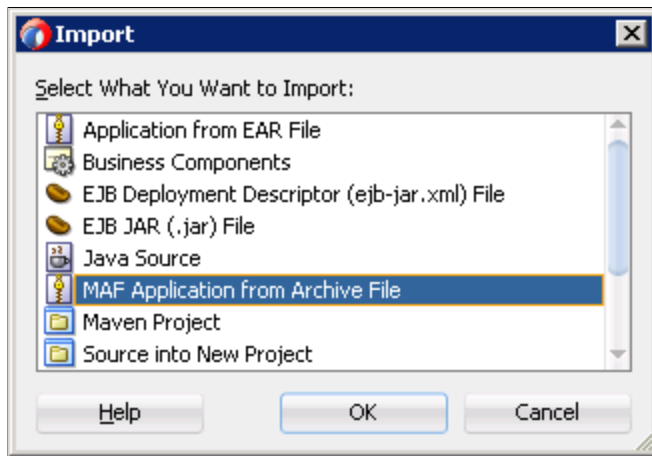
3. On the Select MAA File to Import dialog box, navigate to PS_HOME\setup\MAF and select
    PS_MAF_MOBILE_PROJECT.maa, and click Open.

Image: Selecting the PeopleSoft MAF Project

This example illustrates the PeopleSoft .maa file.

4. On the Location screen, enter a custom project name in the Application File field, and click Next.

   Your project name will be reflected automatically in the Directory field, where the project files will be stored. The default project name is the same as the archive file (PS_MAF_MOBILE_PROJECT).

   Image: MAF Application from Archive Dialog Box

   This example illustrates specifying the .maa file location as well as the location of the new application.
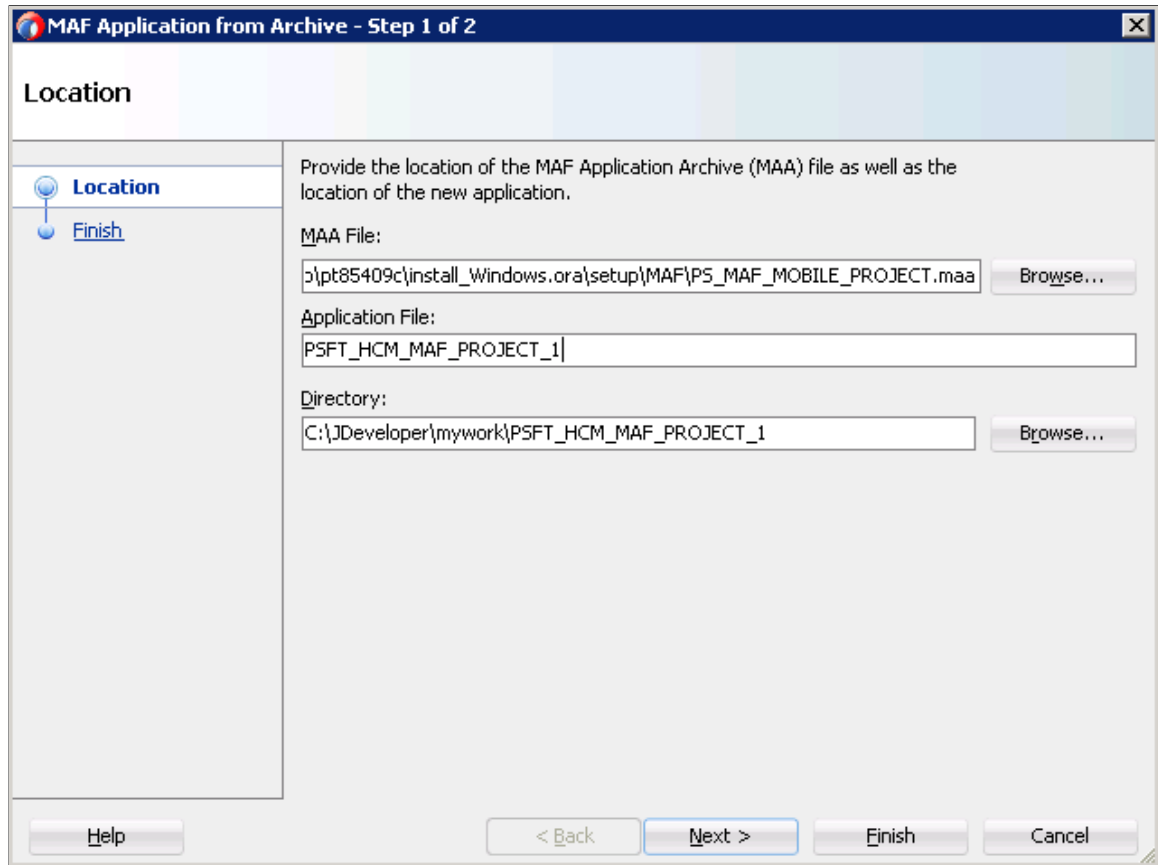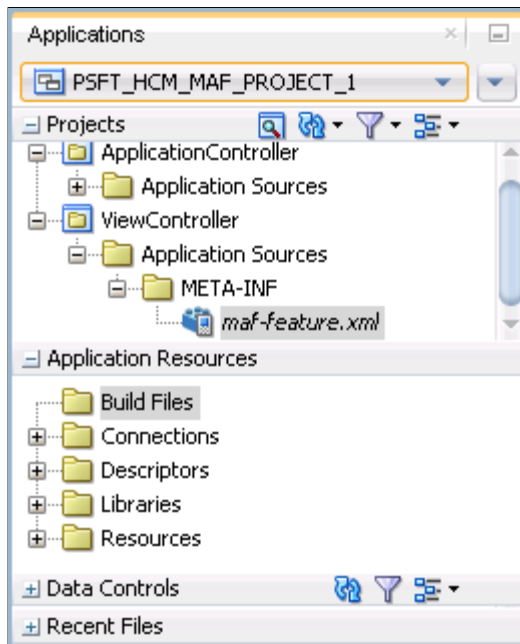

5. On the Finish screen, click Finish.

   After importing the archive you have the basis of a MAF application open in your IDE.

Image: MAF Feature Directory Structure

This example illustrates the maf-feature.xml file in the feature directory structure.

Also, the system creates the appropriate project directory structure, containing the required files for further configuration and deployment.

Image: JDeveloper Project Directory Structure

This example illustrates the project directory structure under JDeveloper\mywork.
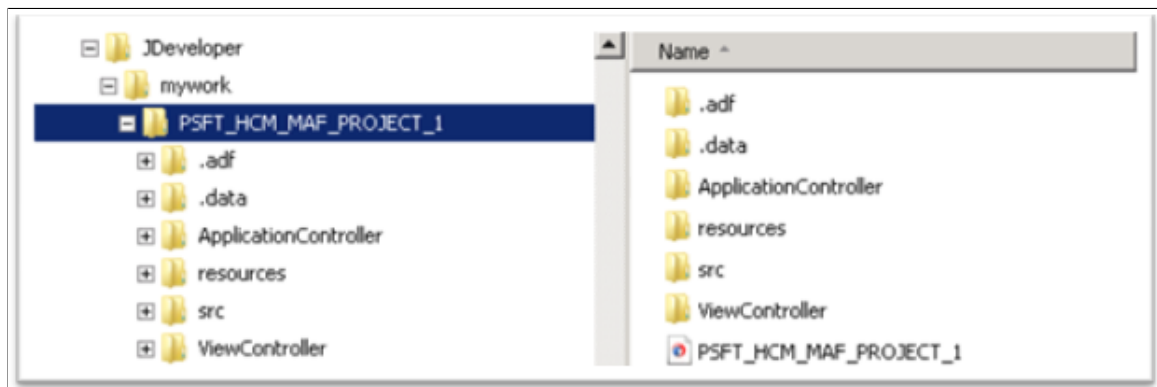
Configuring the Mobile Application Project for Your Site

This topic describes how you can configure the delivered mobile application project.
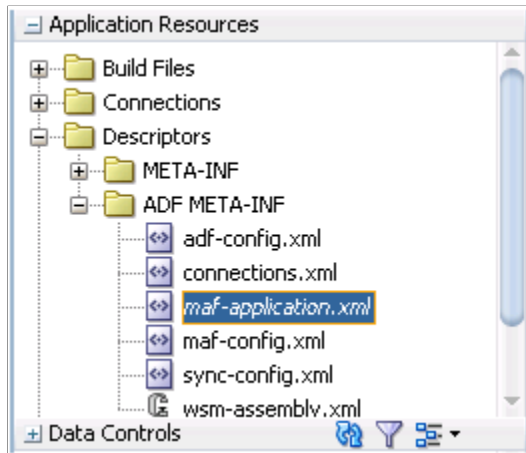
Setting the Application Name

To set the application name:

1. Expand the Application Resources pane.

2. Expand the Descriptors folder.

3. Expand the ADF META-INF folder.

4. Double-click maf-application.xml.

   **Image: IOpening maf-application.xml**

   This example illustrates navigating to and opening maf-application.xml.

5. On the maf-application.xml tab, select the Application tab:

   • Enter your custom application name value in the Name field.

   • Consider the Navigation Bar behavior. If Show Navigation Bar on Application Launch is selected, the system displays the Navigation Bar at the bottom of the application, where end users can configure the URL using the Settings tab.

Image: maf-application.xml tab

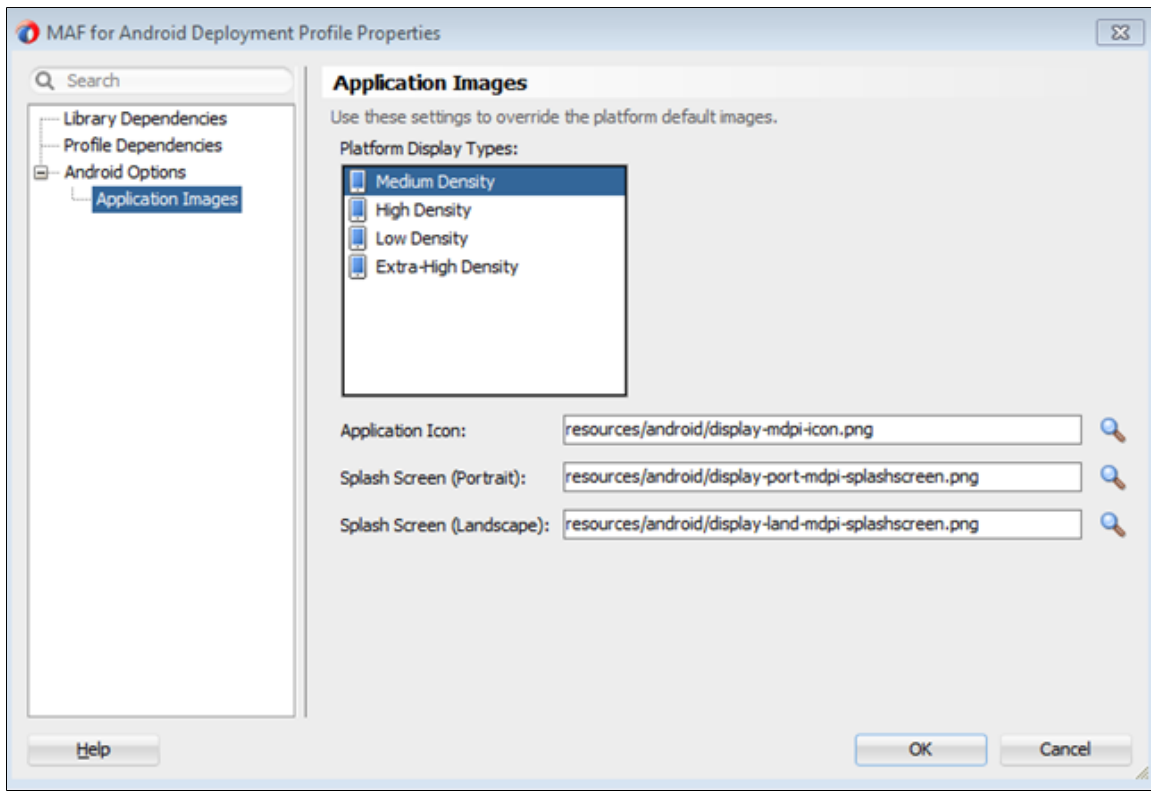This example illustrates selecting the maf-application.xml tab and naming the application.
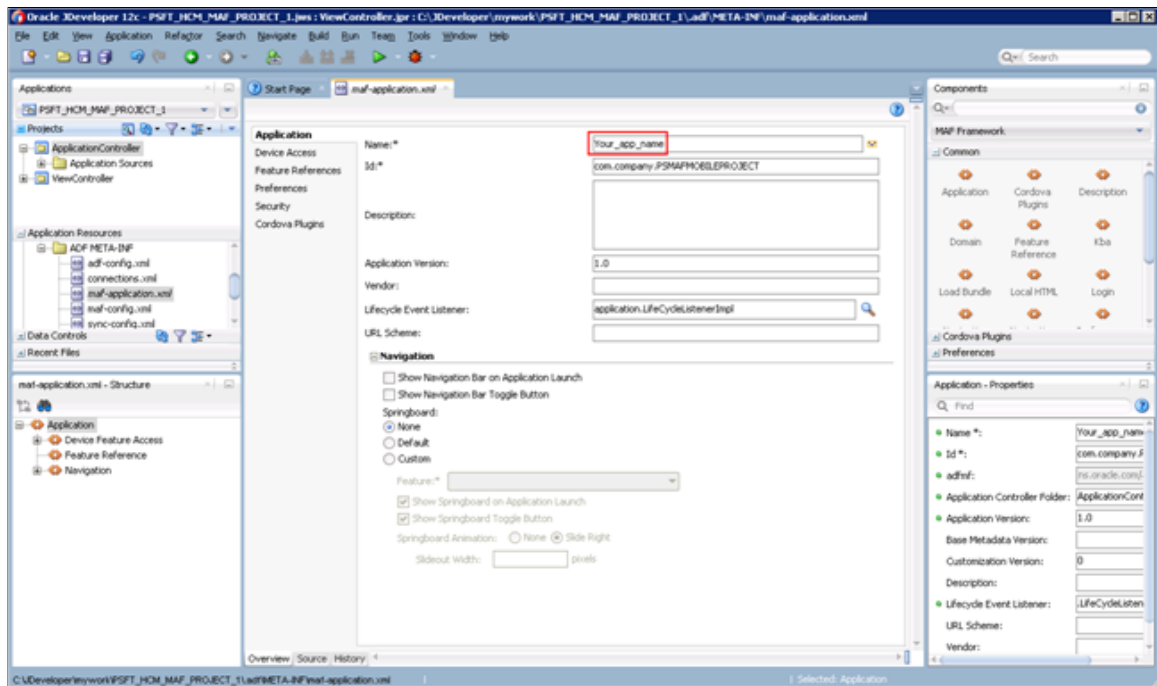
See Setting the Default Remote URL.

Selecting Application Icons

The application icon can be changed in the Application Images screen in the Deployment Profile Properties dialog box. Make sure to have the same images placed inside the resources folder of the deployment profile.

Image: Deployment Profile Properties dialog box

The following example illustrates modifying the application icon.

**Note:** While the example above depicts Android, this applies to iOS deployments as well.

Selecting Feature Icons

The feature icon can be changed using the Navigation Bar field on the General tab of a feature in the maf-feature.xml.

Feature icons apply to both Andriod and iOS deployments. For iOS, you need to follow the standards provided by Apple as described in their developer's guide.

Image: Selecting Feature Icons

This example illustrates selecting feature icons with the navigation bar field.
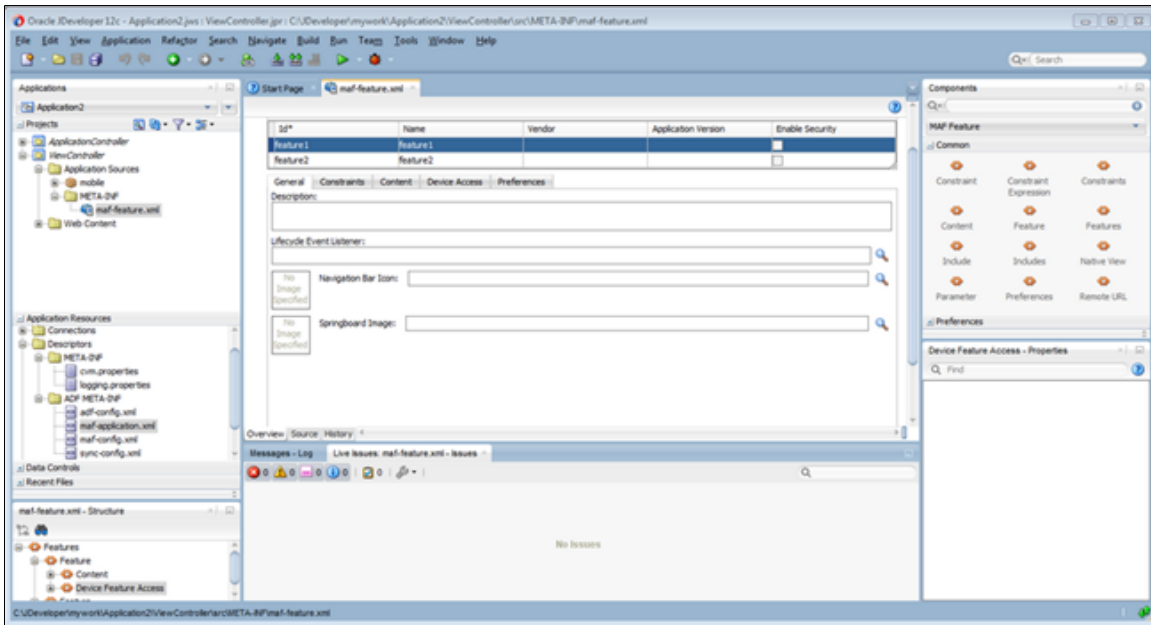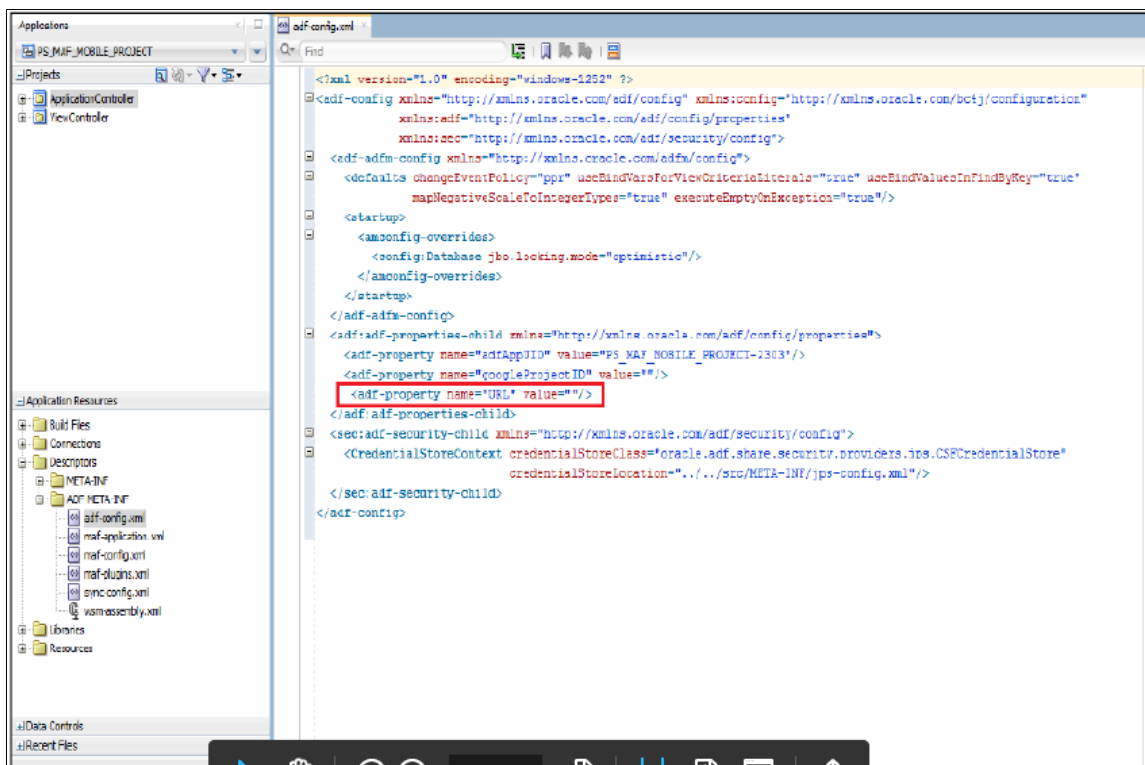

Setting the Default Remote URL

You can set the default URL for end users so that the specified URL value appears for the first launch of the application.

Set the default URL in the adf-config.xml file using as value of the URL property.

Image: adf-config.xml file

This example illustrates setting the default URL for end users.

When the user first launches the application, if you have configured the URL in the adf-config.xml, the application navigates to the default URL specified.

To change the URL, users can tap the Settings tab in the navigation bar, enter the new URL and tap Submit.

Users can also see the list of URLs they have used previously by tapping the Saved URLs list. Select a URL in the list and tap submit to navigate to that URL instead of the default URL.

If you have not provided a default URL value in the adf-config.xml file, when the user first clicks the application, the system displays an alert message indicating that the user needs to provide a URL to the PeopleSoft system.

After the user clicks OK, the system displays the Settings tab for the user to set the URL.

Setting the Remote URL Whitelist

Make sure to set the Remote URL Whitelist in the Security section in the maf-application.xml, as shown in the following example. Connections to domains on the white list open in the WebView within the MAF application container. Connections to domains that are not on the white list open in the device browser, outside of the MAF container.

Image: Remote URL Whitelist

This example illustrates the Remote URL Whitelist in maf-application.xml.
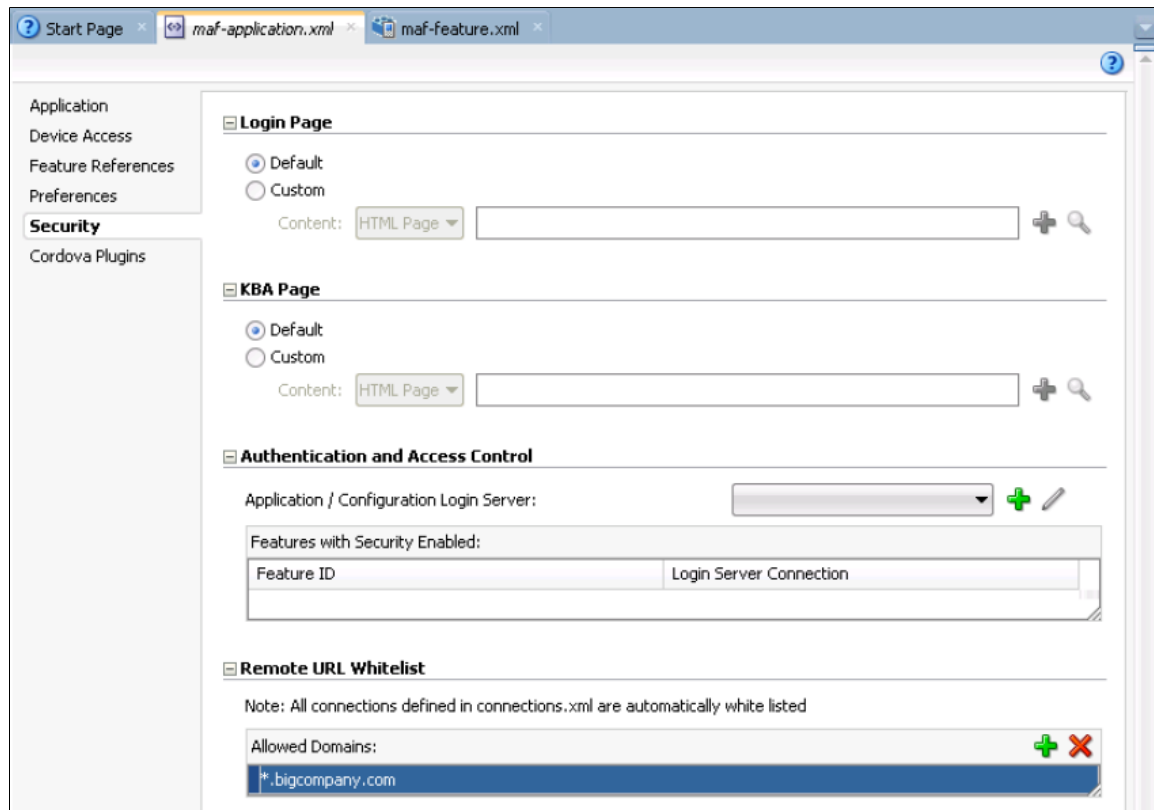
For more information, see Implementing Application Features as Remote URLs.

Enabling Push Notification for MAF

After creating a MAF application container for PeopleSoft, you can enable push notification for the MAF application. This allows the PeopleSoft system to send real-time notifications from the PeopleSoft Push Notification Framework to the mobile device. To implement this option, there are some tasks that need to be completed within the PeopleSoft Push Notification Framework and within MAF. This topic focuses on the tasks that need to be completed in MAF.

For information on the tasks that need to be completed in the Push Notification Framework, see Configuring Push Notification to Run within Oracle Mobile Application Framework.

Enabling Push Notification for iOS MAF Applications

For iOS, generate a provisioning profile that is Push Notification enabled. The push-enabled profile must be selected in JDeveloper > Preferences > Mobile Application Platform > iOS Platform to receive the device token, which is a unique alpha-numeric string used by the server to send the notification.

Enabling Push Notification for Android MAF Applications

To enable Push Notification for Android MAF applications:

1. Create a new project in the Google API console.

2. Enable Google Cloud Messaging for Android under Services.

3. Insert the Project ID into the value field of adf-property *googleProjectID* under adf-config.xml in your JDeveloper project.

Image: Setting adf-property

This example illustrates setting the adf-property to reflect the Google Project ID.

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<adf-config xmlns="http://xmlns.oracle.com/adf/config" xmlns:config="http://xmlns.oracle.com/bc4j/configuration"
            xmlns:adf="http://xmlns.oracle.com/adf/config/properties"
            xmlns:sec="http://xmlns.oracle.com/adf/security/config">
  <adf-adfm-config xmlns="http://xmlns.oracle.com/adfm/config">
    <defaults changeEventPolicy="ppr" useBindVarsForViewCriteriaLiterals="true" useBindValuesInFindByKey="true"
              mapNegativeScaleToIntegerTypes="true" executeEmptyOnException="true"/>
    <startup>
      <amconfig-overrides>
        <config:Database jbo.locking.mode="optimistic"/>
      </amconfig-overrides>
    </startup>
  </adf-adfm-config>
  <adf:adf-properties-child xmlns="http://xmlns.oracle.com/adf/config/properties">
    <adf-property name="adfAppUID" value="PS_MAF_MOBILE_PROJECT-2303"/>
    <adf-property name="googleProjectID" value="327307545797"/>
  </adf:adf-properties-child>
  <sec:adf-security-child xmlns="http://xmlns.oracle.com/adf/security/config">
    <CredentialStoreContext credentialStoreClass="oracle.adf.share.security.providers.jps.CSFCredentialStore"
                            credentialStoreLocation="../../src/META-INF/jps-config.xml"/>
  </sec:adf-security-child>
</adf-config>
```

Deploying Your Mobile Application

When distributing a MAF application you have the option to distribute the application internally using your own app repository, or you may elect to distribute the application externally to the Apple App Store or Google Play repositories.

Note: These steps are documented in detail in the MAF documentation. The sections in this document outline specific items to complete from a PeopleSoft perspective.

Related Links
Deploying Mobile Applications

Deploying Applications for Android

This section describes:

• Distributing an Android Application Externally.

• Distributing an Android Application Internally.

Distributing an Android Application Externally

To deploy an application for Android:

1. In JDeveloper, create a deployment profile, by selecting Application > Deploy > New Deployment Profile.

2. For the Profile Type, select *MAF for Android* from the drop-down list, and enter your Deployment Profile Name.

3. For the deployment of the Android application to package, it is recommended to set the deployment profile to Release mode instead of Debug mode.

   For this you need to sign the application, and map the same keystore in the Tools > Preferences > Mobile Application Framework > Android Platform and browsing through the keystore created in the Release tab under Signing Credentials.

   Image: Mobile Application Framework: Android Platform

   This example illustrates settings required for deploying applications for the Android platform.
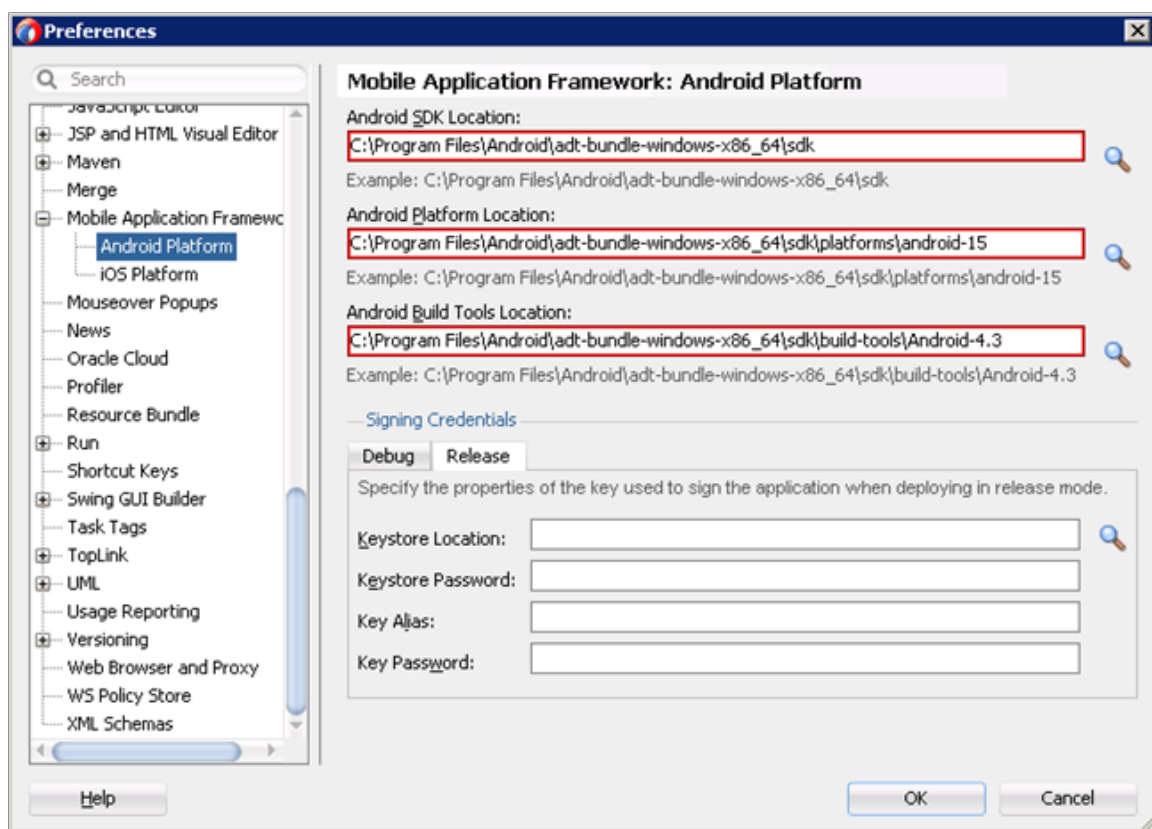
4. Deploy the Android application to distribution package.

   The result is the .apk file in the <path to app location>/deploy/<deployment profile>/<app name>.apk.

   You distribute the .apk file.

Deploying an Android Application Internally

The same .apk file generated using the steps in the previous section can be used to distribute internally. However, you would upload the file to an internal repository.

For example, you can store the .apk on your web server and have a link pointing to this .apk file from an HTML page for users to access.

Deploying Applications for iOS

This section describes:

- Distributing an iOS Application Externally.

- Distributing an iOS Application Internally.

Deploying an iOS Application Externally

To deploy an application for iOS:

1. Create a deployment profile in JDeveloper, by selecting Application > Deploy > New Deployment Profile.

2. For the Profile Type, select *MAF for iOS* from the drop-down list, and enter your Deployment Profile Name.

For the deployment of the iOS application to a package, you need to complete the steps in the following sections:

- Registering the Certificate.

- Registering the Provisioning Profile.

- Setting JDeveloper Preferences.

- Defining Your Application Bundle ID.

Registering the Certificate

To register the certificate:

1. From your Applications > Utilities folder open the KeyChain Access utility.

2. Ensure that the login keychain is the default (shown in bold).

3. Select the login keychain and choose File > Import Items from the Menu.

4. In the Import dialog select your certificate. Ensure that "login" is shown as the Destination Keychain in the drop down at the bottom of the dialog.

5. When prompted, enter the password.

Registering the Provisioning Profile

To register the provisioning profile:

1. Start XCode.

2. In XCode, choose XCode > Preferences.

3. On the Accounts tab, click the "+" button at the bottom, left corner.

4. Select *Apple ID* from the drop-down list options.

5. Sign in with your Apple ID.

Note: The certificate must be issued with this ID.

Setting JDeveloper Preferences

To set JDeveloper preferences:

1. Start JDeveloper and navigate to the preferences screen (Tools > Preferences).

2. In the Preferences screen navigator navigate to Mobile Application Framework > iOS Platform.

3. In the Device Signing section, for the Provisioning Profile field, click the Browse button and navigate to your provisioning profile.

4. In the Certificate field, copy and paste the certificate value from your keychain, making sure to include the entire certificate.You can right-click on the certificate and copy the text against Get Info.

5. Click OK to save the preferences.

Defining Your Application Bundle ID

To define the application bundle ID:

1. For every application that you want to deploy to a package with this profile, edit the deployment profile you are using through the Application Properties dialog box.

2. Select iOS Options in the Deployment Profile dialog box.

3. Ensure that the prefix of the value in the Application Bundle ID of the deployment profile property is taken from the link. Refer to your member center in the Apple developer site for more information.

4. Associated with the profile name, locate the ID, and append the Application Name to this ID.

Deploying the iOS Application Internally

To deploy the iOS application to the distribution package:

1. Open Xcode, and select File > Open, and navigate to <path to app location>/deploy/<deployment folder>/temporary_xcode_project/Oracle_ADFmc_Container_Template.xcodeproj.

2. When the project opens, change all the attributes under Signings in Build Settings to your provisioning profile. Also, ensure the app is reaching targeted customers by selecting the device support on the top after the app name.

3. Select Product > Archive, select Distribute, then click Next.

4. Select Save for Enterprise or Ad Hoc Deployment, and click Export.

5. Select your Mobile Provisioning Profile, and click Export.

6. Save the ipa into the desired location, and enter the file name.