**ORACLE**
PeopleSoft

# PeopleTools 8.61: Fluid User Interface Developer's Guide

**January 2024**

**ORACLE**

## Documentation Accessibility

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

## Access to Oracle Support

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

# Contents

# Preface

---

## Understanding the PeopleSoft Online Help and PeopleBooks

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

## Hosted PeopleSoft Online Help

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

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

## Locally Installed PeopleSoft Online Help

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

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

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

## Downloadable PeopleBook PDF Files

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

## Common Help Documentation

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

- Application Fundamentals

- Using PeopleSoft Applications

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

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

## Field and Control Definitions

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

## Typographical Conventions

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

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

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

# ISO Country and Currency Codes

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

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

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

# Region and Industry Identifiers

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

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

## Region Identifiers

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

- Asia Pacific

- Europe

- Latin America

- North America

## Industry Identifiers

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

- USF (U.S. Federal)

- E&G (Education and Government)

## Translations and Embedded Help

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

# Using and Managing the PeopleSoft Online Help

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

- Using the PeopleSoft Online Help.

- Managing hosted Online Help.

- Managing locally installed PeopleSoft Online Help.

# PeopleTools Related Links

PeopleTools 8.61 Home Page

PeopleSoft Search and Insights Home Page

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

PeopleSoft Online Help

PeopleSoft Information Portal

PeopleSoft Spotlight Series

PeopleSoft Training and Certification | Oracle University

My Oracle Support

Oracle Help Center

# Contact Us

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

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

# Follow Us

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

**Chapter 1**

# Getting Started with 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.

- Application Designer.

- PeopleCode.

- PeopleTools portal technology including the 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 Android 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 system recognizes the device used to access it, and renders fluid pages so that the application appears naturally, as expected by the user of a certain device type. For example, the following shows a PeopleSoft fluid application page appearing on a browser on a laptop or a tablet device.

This example shows a fluid page displayed on a tablet in landscape mode.
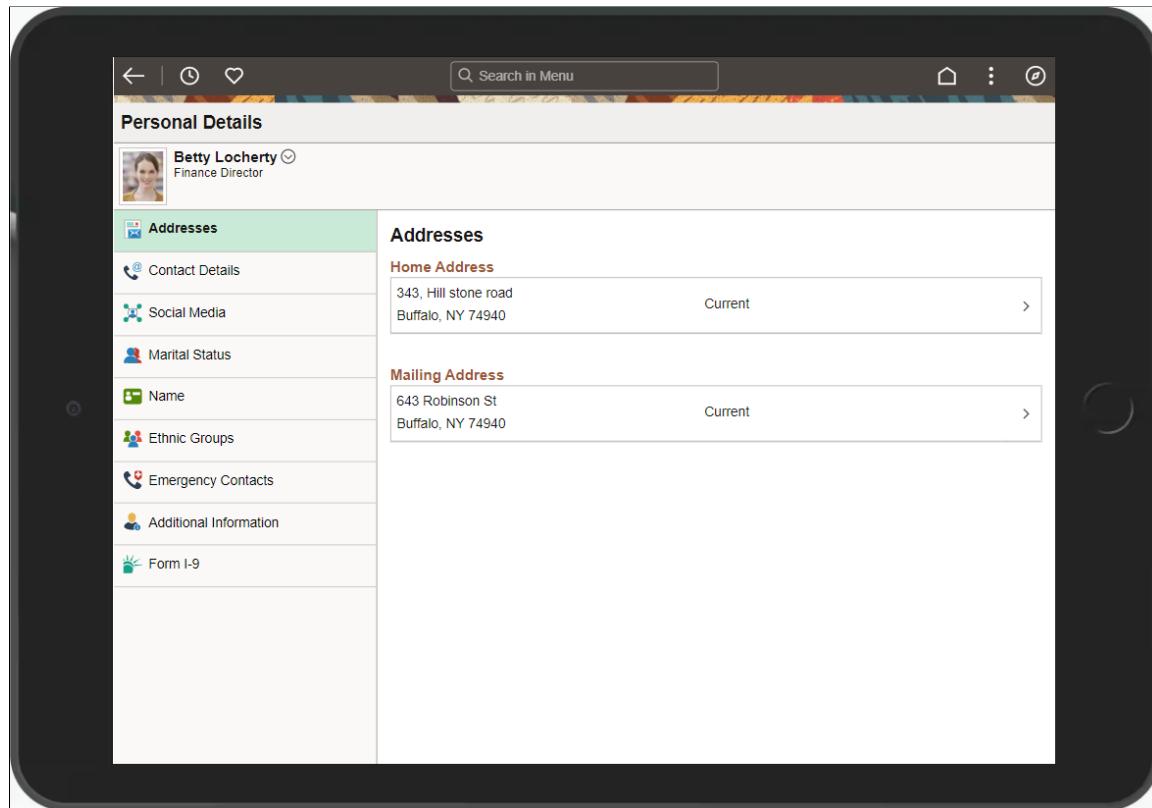


If you access that same page using a cell phone and view it in landscape mode, it would appear similar to the following example.

The following example shows how a fluid page may appear on a smart phone in landscape mode.
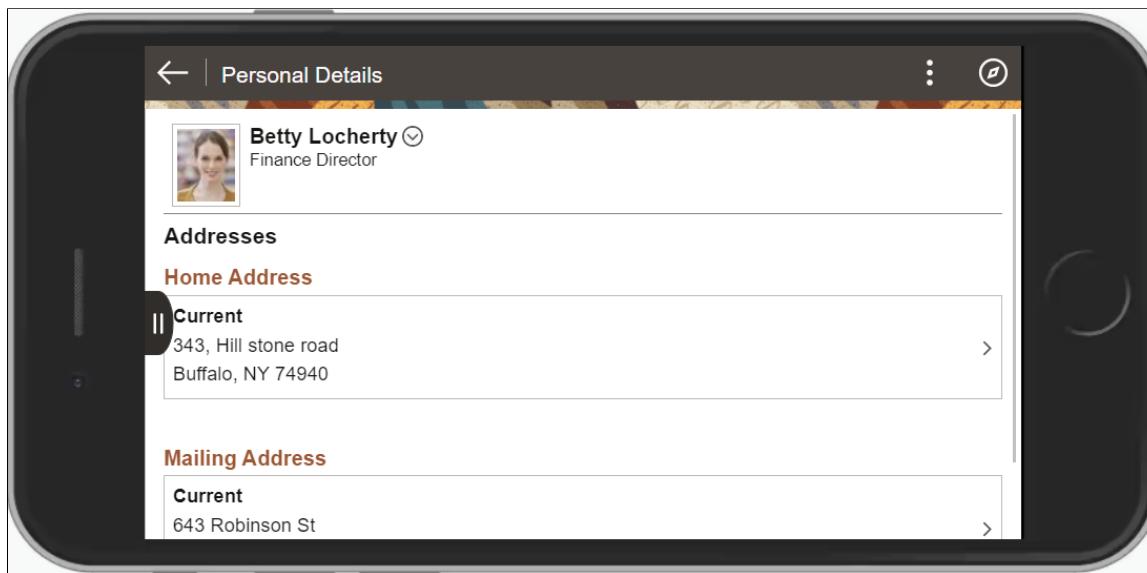


Notice that the page shrinks to fit the screen size of the device. Which controls are displayed in the banner varies depending on the width of the browser window. As the width decreases, controls are moved from the banner to become links in the Actions menu. The user can use typical gestures, such as swiping (on homepages only), to scroll through the fluid page.

If you access the same page using a mobile phone and view it in portrait mode, you will notice additional changes as the page further adapts and responds to the adjusted screen display. All controls except for the Back button have been moved into the Actions menu. The PeopleSoft Fluid User Interface is *adaptive* in that the system detects the device and generates the page rendering 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.

The following example shows a fluid page displayed on a smart phone in portrait mode.
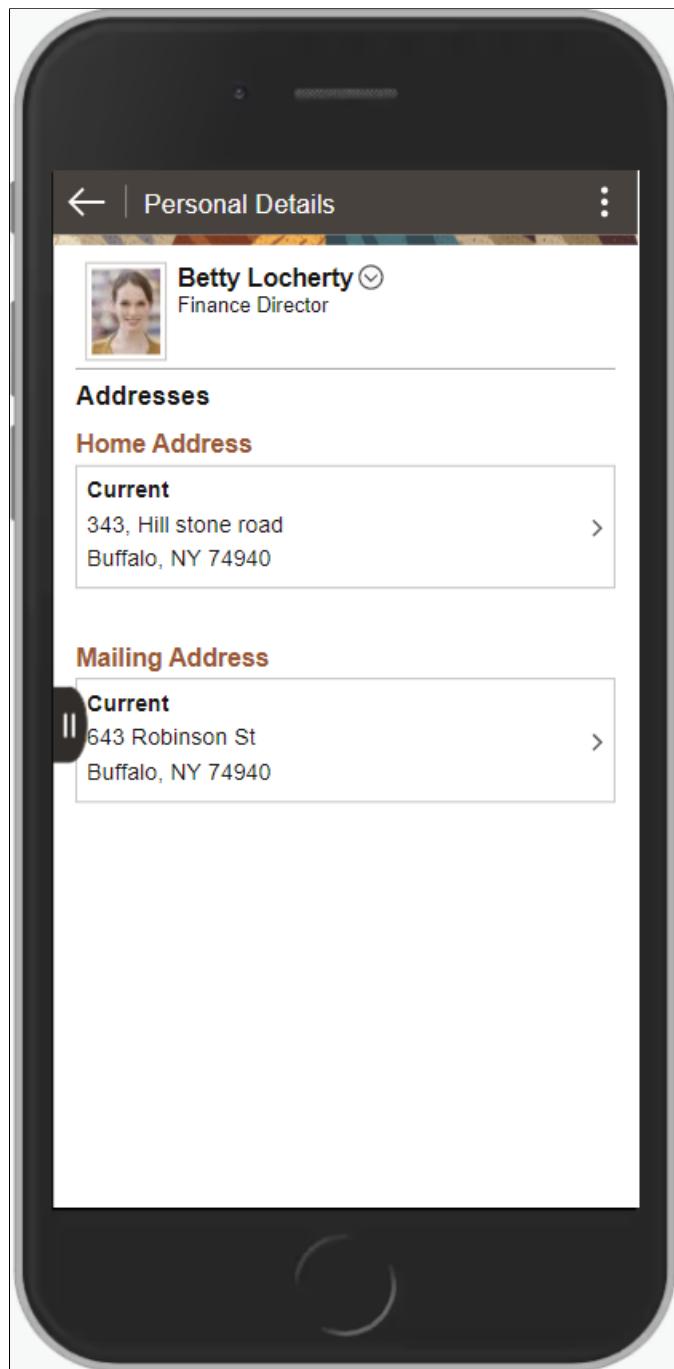


Notice that the system further shrinks the page, while maintaining the fundamental structure.

## Related Links
"Working with Fluid Homepages and Dashboards" (Applications User's Guide)
"Working with Fluid Pages and Controls" (Applications User's Guide)
"Using the Actions Menu" (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 PeopleSoft classic 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 fluid page definitions, the approach and layout used are completely different than what are used for traditional PeopleSoft page development. In addition, 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 as a fluid application. |
| 7. Register components. | Classic and fluid applications both need to have the components registered using the registration wizard. |
| 8. Test the application. | Use actual mobile phones, tablets, laptops, and desktop computers to test your application.<br><br>**Note:** While you can use device emulators for some preliminary testing, these emulators do not fully replicate the actual device and its native operating system. In particular, iOS testing requires an actual iOS device. |

**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 classic 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 or customize PeopleSoft fluid applications will need expertise in these additional areas (listed in order of importance and recommended experience):

- CSS3

- HTML5

- JavaScript

---

**Important!** With the flexibility provided through creating and customizing HTML, CSS, 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. |
| 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, while you cannot disable fluid deployment at the system level, you can direct all users to the classic homepage at signon. See Setting Device Homepage Defaults for more information. |
| 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. Plus, some pages are administrative in nature.<br><br>Would these pages benefit from a simpler reworking to make them classic plus, but consequently foregoing the responsive and adaptive nature of fluid applications? See Converting Existing Classic Pages to Fluid for more information. |
| Does an application *require* functionality that does not exist in the fluid user interface? | For example, some features, like spell check, trees, and others are not available in fluid. See Page Controls Not Supported in Fluid Applications for more information. |

**Chapter 3**

# Creating Pages for Fluid Applications

## Creating Fluid Pages

### Creating a New Main Fluid Page

The main fluid page is the main content area, or target page. The main page is used directly within the component.

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

1.  In Application Designer, select File, New.

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

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

    **Note:** Most main fluid pages are based on the PSL_APPS_CONTENT layout page.

    See Choosing a Fluid Layout Page.

4.  Enter the page name in 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).

    **Note:** Typically, click Yes to include any PeopleCode from the layout page.

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

### Creating Other Fluid Page Types

To create other fluid pages that can be created from a page template:

1.  In Application Designer, select File, New.

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

3.  Select an initial layout page (template) from the Choose Layout Page that corresponds to the page type that you want to create—for example, select *PSL_SIMPLE_SBP* to create a fluid subpage.

    See Choosing a Fluid Layout Page.

4.  Click Choose and enter the page name in 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).

   **Note:** Typically, click Yes to include any PeopleCode from the layout page.

6. Open the Page Properties dialog box.

7. On the Use tab, select the page type—for example, select *Subpage* to create a fluid subpage, or select *Secondary* to create a fluid secondary page, and so on.

   **Important!** Do not select the Disable Display in Modal Window When Not Launched by DoModal Peoplecode check box for fluid secondary pages. Fluid secondary pages must always be displayed in a modal window.

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

### Creating Other Fluid Pages Not Based on a Template

Certain fluid page types (such as footer pages, search pages, and so on) are not created from a fluid page layout template. Instead, you create these from the standard Page menu item.

To create other fluid pages that cannot be created from a page template:

1. In Application Designer, select File, New.

2. Select *Page* in the New Definition dialog box, and click OK.

   **Note:** Do not select *Page (Fluid)*.

3. Open the Page Properties dialog box.

4. On the Use tab, select the page type—for example, select *Footer Page*. Also, select the *Fluid Page* check box.

5. Insert a group box on the page that encompasses the entire page content area.

### Related Links
Working with Subpages
Working with Secondary Pages

## Choosing a Fluid 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.

**Important!** Do not include a layout page in any component definition. Layout pages are templates only for the generation of a new fluid page definition.

This example illustrates the layout page options provided by the Choose Layout Page dialog box.
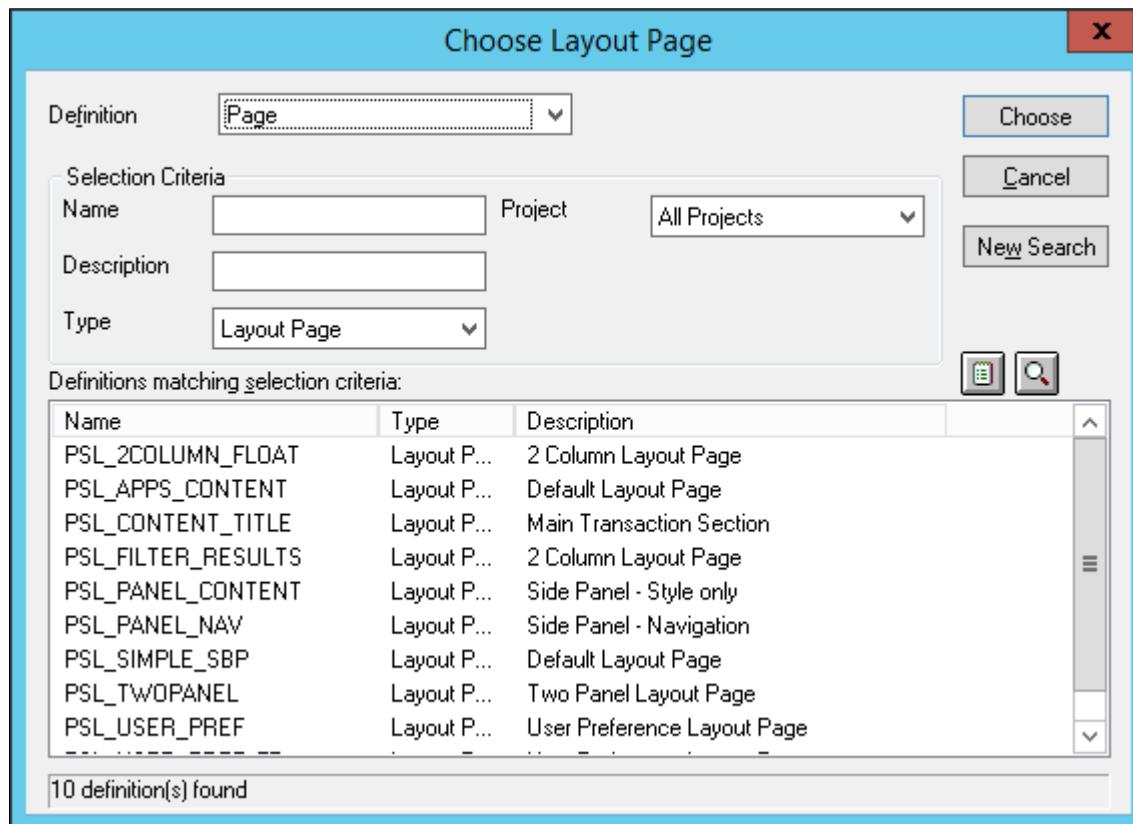


In your new page definition, you may add or remove elements as needed for your purposes. However, make sure to examine the template pages so that you understand how any existing PeopleCode or CSS styles affect the interface.

---

**Important!** Not all of the delivered layout pages are intended for external usage.

---

## Recommended Layout Pages

This table describes the layout pages suitable for fluid application development.

| Layout Page | Description |
|---|---|
| PSL_APPS_CONTENT | Use PSL_APPS_CONTENT as the basis for most main fluid pages of type Standard or Secondary. This layout 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 outer flexbox necessary for all fluid applications. |
| PSL_2COLUMN_FLOAT | This layout page contains nested layout group boxes inside of the outermost group box that has the *ps_apps_content* style applied.<br><br>If you're using this layout page, you need to supply the styling for the child containers based on your needs. |

| Layout Page | Description |
|---|---|
| PSL_CONTENT_TITLE | This layout page includes a special Page Title group box to display a page title on the page. Use this layout page for pages that are displayed inside two-panel wrappers such as activity guides or master/detail components. |
| PSL_PANEL_CONTENT | This layout page includes a single container having the default styling for the side panel (left or right panel) usage.<br><br>If you're 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 target page content. Role (ARIA) automatically applied to indicate that the contents will be navigational in nature. |
| PSL_SIMPLE_SBP | Use PSL_SIMPLE_SBP as the basis for fluid subpages.<br><br>**Note:** After creating a subpage from this layout template, you must change the page type from *Standard* to *Subpage*. See Creating Other Fluid Page Types. |
| 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 this layout page for creating user preferences to be included as part of My Preferences. |
| PSL_USER_PREF_FT | Use this layout page for creating a footer page for user preferences. |

## Layout Pages Reserved for Internal Usage

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

### Related Links
Selecting Fluid Page Types

# Converting Existing Classic Pages to Fluid

When you're evaluating whether to convert your existing custom classic applications to fluid, you should consider these questions:

- Will the application be used on smart phones and tablets as well as laptops and desktop computers, thereby taking advantage of the responsive and adaptive nature of fluid applications across all form factors?

- Will the application benefit from easy access via a tile on a homepage or dashboard?

- Or, if the application is administrative in nature, would it benefit from a simpler reworking to make it classic plus, but consequently foregoing the responsive and adaptive nature of fluid applications?

Then, consider these implementation approaches:

- For applications that do not require all of the benefits of the responsive and adaptive nature of fluid applications, rework the classic pages to classic plus.

- For simple classic applications with a simple page structure and few page controls, create a new fluid component and page, and then copy all of the existing classic page controls into the new fluid page.

- For more complex classic applications with complex page structure, consider starting anew by redesigning the pages as a fluid application, taking full advantage of all fluid page design features.

**Related Links**

PeopleSoft Spotlight Series: Developing Fluid Applications
"Reworking Components for Classic Plus Display" (Application Designer Developer's Guide)

## Selecting Fluid Page Types

When developing fluid applications, 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 a standard page that has been designated a Fluid Page on the Use tab of the Page Properties. The main fluid page definition contains the bulk of the transaction fields and controls, which are contained within an outermost layout group box that has the *ps_apps_content* style applied. 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.

The following example shows the Page and Page (Fluid) definition types in the definition list in Application Designer.
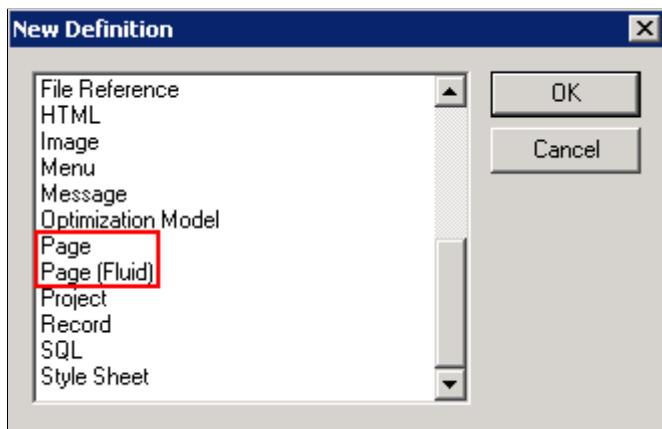


You select the remaining page types for fluid mode, from the Page Type drop-down list on the Use tab of the Page Properties dialog.

This example shows the fluid page types listed in the Page Type drop-down list on the Page Properties, Use tab.
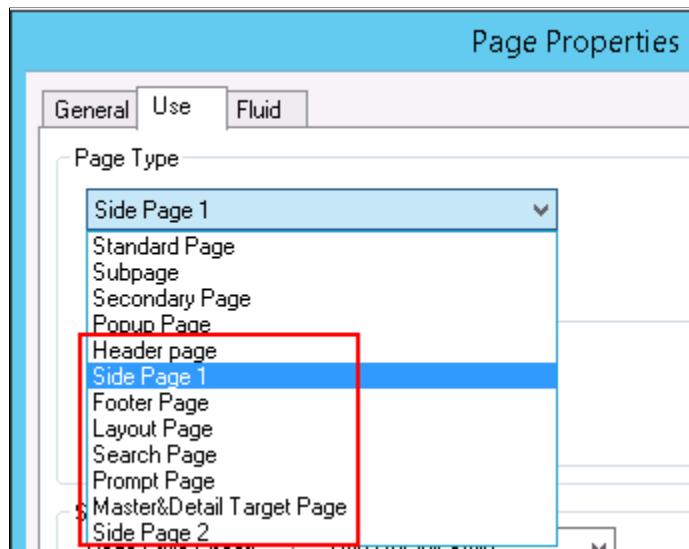


The following table provides a brief description of each fluid page type and the layout page to be used when creating a new page definition.

| Page Type | Layout Page | Description |
| --- | --- | --- |
| Standard Page | PSL_APPS_ CONTENT* | The main fluid page is the main content area, or target page; the main page is used directly within the component.<br><br>The main fluid page contains the outermost group box, which acts as the overall flexbox container for the fluid page content. This group box has the *ps_apps_content* style applied. |
| Subpage | PSL_SIMPLE_SBP | Fluid subpages can be used to reduce clutter, group related fields, and segregate business logic. |
| Secondary Page | PSL_APPS_ CONTENT* | Fluid secondary pages are typically displayed in a modal window. See Implementing Transfers and Modal Windows with PeopleCode. |
| Header Page | NA | 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.<br><br>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 | PSL_PANEL_ CONTENT<br><br>PSL_PANEL_NAV | Page displayed on the left side off the screen, often containing left-side navigational constructs. Users can slide it open and shut. |

| Page Type | Layout Page | Description |
|-----------|-------------|-------------|
| Footer Page | NA | 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. |
| Layout Page | NA | You can designate a fluid page definition as a layout page for use as a template for development of new fluid page definitions. Do not include layout pages within a fluid component definition.<br><br>**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). |
| Search Page | NA | Page generated in <aside> section containing search pages, which include Find an Existing Value and Keyword Search.<br><br>**Note:** An Add a New Value page is not provided by the search page definitions provided by PeopleTools. |
| Prompt Page | NA | The definition for a fluid prompt page.<br><br>PeopleTools supplies a default prompt page, PT_PROMPTPAGE , which can be cloned to create a new prompt page definition. |
| Master&Detail Target page | NA | 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 | PSL_PANEL_ CONTENT | 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. |

[*] Typically, PSL_APPS_CONTENT is used to create pages of this type.

The following diagram illustrates the *general* layout of the various fluid page types.
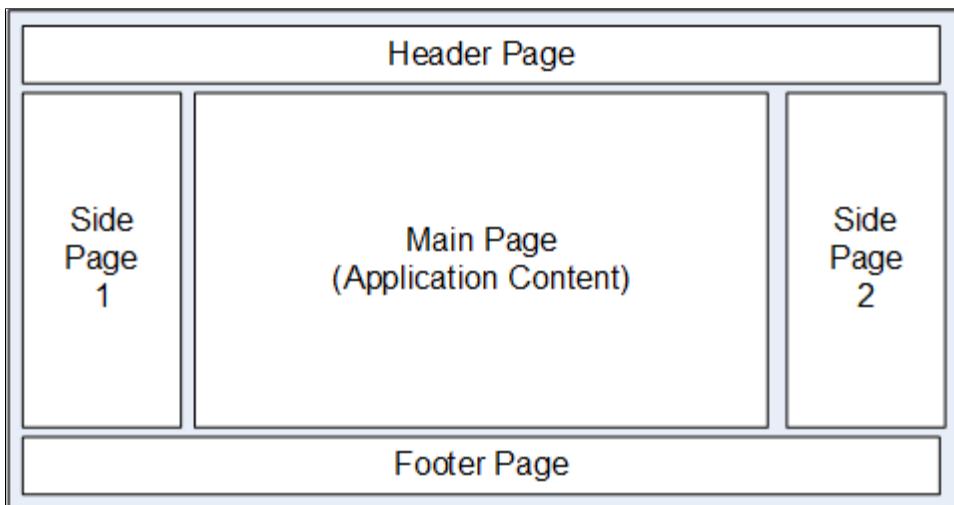


**Note:** Not all page types are required to be used. The application content area displays the content presented by the main fluid page.

### Related Links

Choosing a Fluid Layout Page

"GetPageType" (PeopleCode Language Reference)

# 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" (Application Designer Developer's Guide)

"Creating New Page Definitions" (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 fluid applications (versus classic applications).

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.
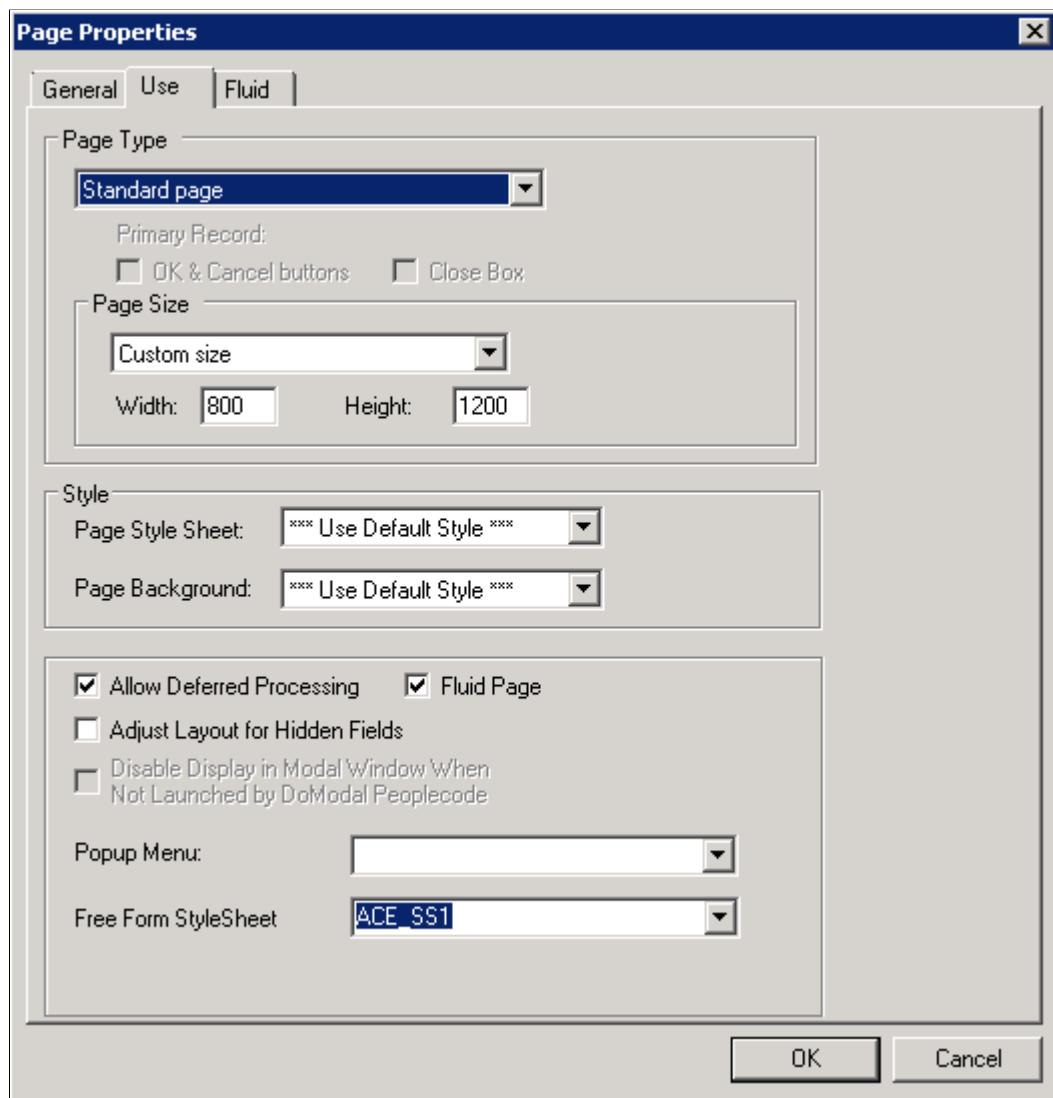


| Field or Control | Description |
|---|---|
| **Page Type** | Select the type of page for use in the fluid application, such as header, footer, search, and so on.<br><br>For more information on page types used for fluid development, see Selecting Fluid Page Types. |
| **Fluid Page** | For a page to 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.

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.



| Field or Control | Description |
|---|---|
| **Style Classes** | Enter the application-specific style classes to apply *to extend* the default system-specific styles (PeopleTools styles applied by default). Separate multiple style classes with a space. |

| Field or Control | Description |
| --- | --- |
| **Form Factor Style Class Override** | Use to override any application-specific styles based on the form factor (device size) of the device accessing the page. Form factors are represented in these ranges:<br><br>• *Small:* Applies to most smartphones.<br><br>• *Medium:* Applies to 7-inch tablets.<br><br>• *Large:* Applies to 10-inch tablets and laptops.<br><br>• *Extra Large:* Applies to large monitors.<br><br>For example, assume that the application-specific Style Classes field contains the style, psc_margin-bottom05em, which you intend to apply to all form factors except for the small form factor. In this case, for the Small field in the Form Factor Style Class Override group box, you enter the override style class to apply when a small form factor device accesses the page. |
| **Suppress System-Specific Style Classes** | **Important!** This setting applies at the page control level only. At the page level, this setting is ignored. |

# 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 applications. In a classic 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.

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 for the user, the layout, spacing, order, color, and so on is for the most part identical between the design time definition and the runtime presentation.

This example illustrates the same page as depicted in the previous example at run time. 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 page.

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 user.
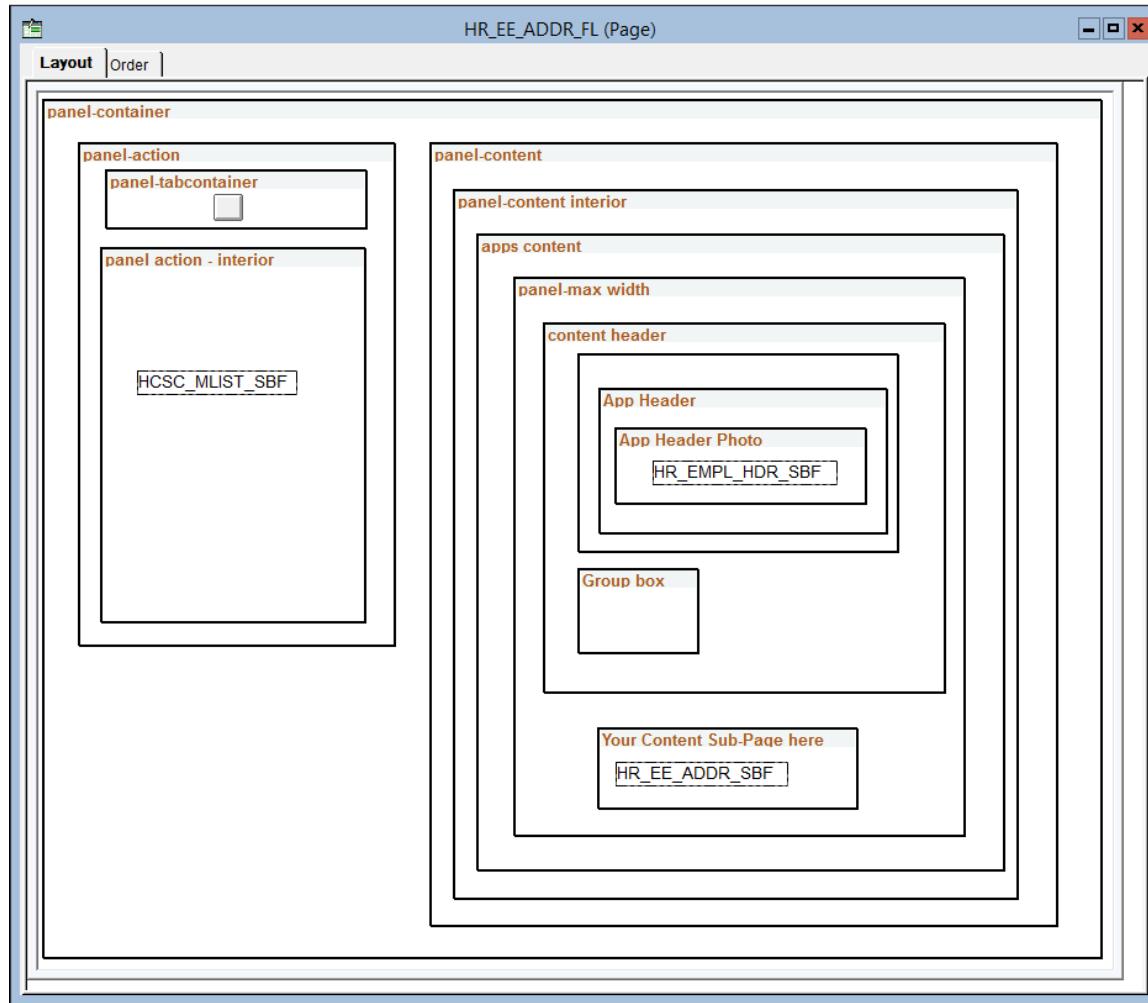


The layout of the fluid application is not what you would call a WYSIWYG approach. If user were to access a page in this state, it would be impossible to complete a transaction. Notice the boxes that are used to indicate levels of containment and to provide visual cues, making each element from high-level 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.

This example illustrates the fluid rendering of this very same page definition (above) in a browser.
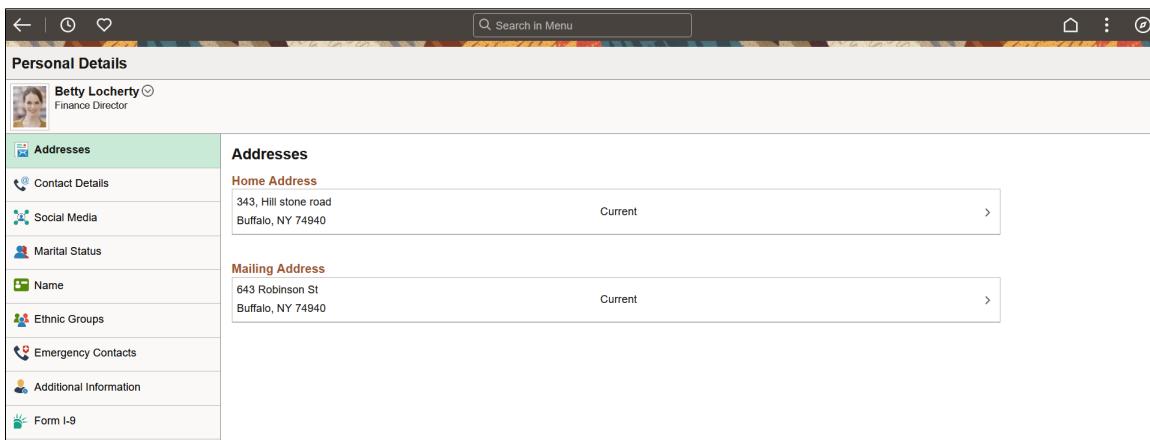


At a high-level, you can map the rendering in the browser to the layout as:

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

- The area showing the navigation on the left is represented in the layout as the group box labeled psc_panel-action.

- The area on the right currently titled Addresses is represented in the layout as the group box labeled Your Content Sub-Page here. When 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.

**Note:** The blue bar running across the top displays the fluid banner, which is the default PeopleTools, or system, header page definition (PT_HEADERPAGE). 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.

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.

During the page design process, you might want to periodically preview how your page looks in the browser. How your page appears in Application Designer at design time may look different when viewed online. This is particularly true for fluid pages, which do not follow the pixel perfect WYSIWYG paradigm of classic pages. With fluid pages, CSS determines how the page and how the page elements are displayed to the user. The physical placement of the page elements often has no bearing on how the fluid page will appear at run time. Therefore, you must either preview fluid pages from Application Designer or view them in a runtime browser session. You can use the View in Browser menu item in Application Designer to preview the look of your fluid pages. See "Previewing Fluid Pages" (Application Designer Developer's Guide) for more information.

**Chapter 4**

# Defining Components for Fluid Applications

## Adding Fluid Pages to a Fluid Component

Like classic 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 component buffer. Having these pages all in the same component 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.

**Note:** Do not include layout pages (templates) within a fluid component definition.

## Setting Component Properties for Fluid Components

This section describes component properties from a fluid perspective.

## Setting Properties on the Fluid Tab

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

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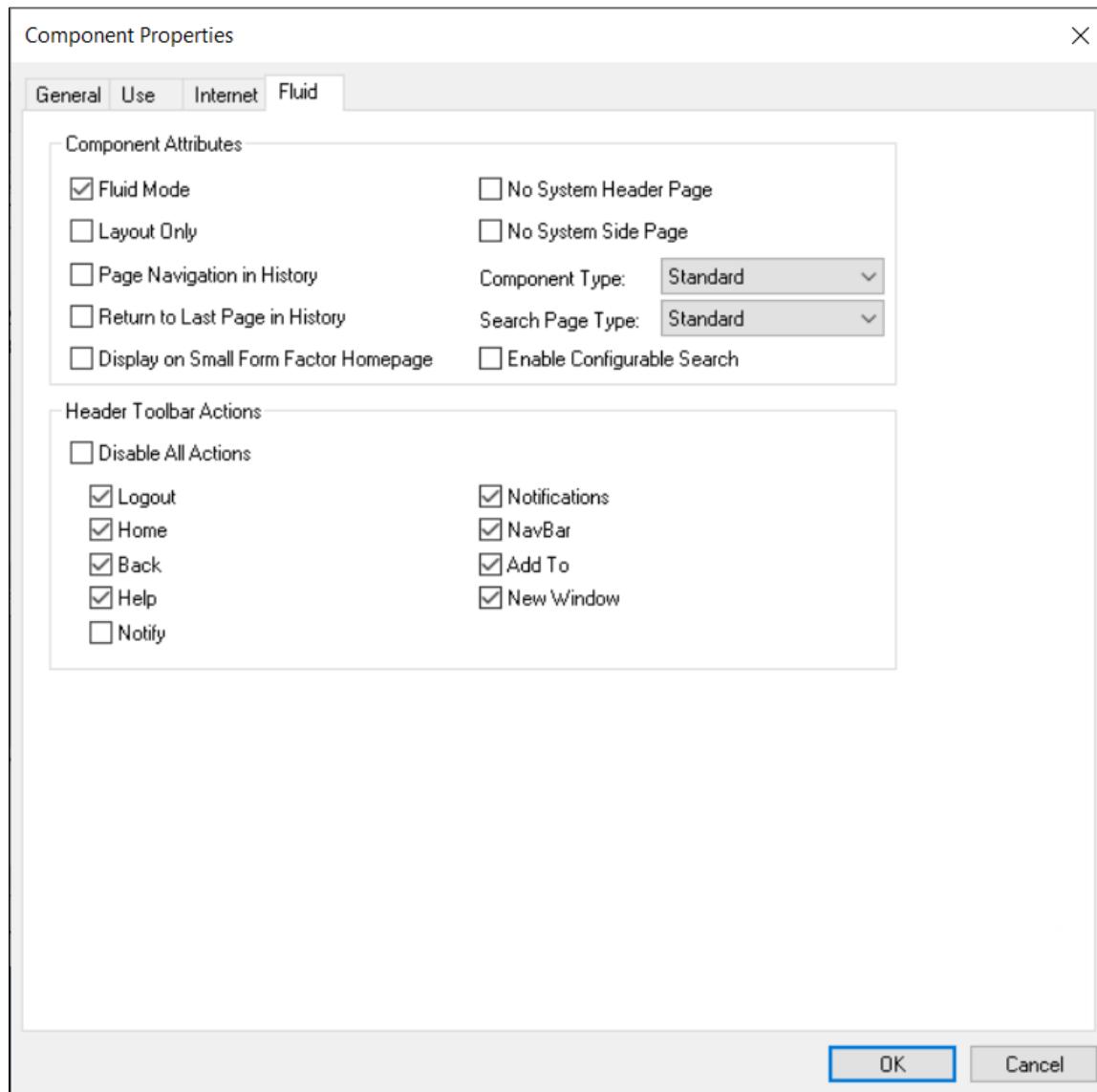


## Component Attributes

| Field or Control | Description |
| --- | --- |
| **Fluid Mode** | Indicates that this is a fluid component. Selecting this check box *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. |

| Field or Control | Description |
|---|---|
| **Page Navigation in History** | Enables the system to store the individual pages that are visited within this component in the Back button history. This can also be enabled using the SetTransferAttributes PeopleCode built-in function. <br><br> **Note:** By default, only the visit to the component is stored in the Back button history. <br><br> For more information, see "SetTransferAttributes" (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. <br><br> 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. <br><br> This can also be enabled using the SetTransferAttributes PeopleCode built-in function. <br><br> For more information, see "SetTransferAttributes" (PeopleCode Language Reference). |

| Field or Control | Description |
|---|---|
| **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.<br><br>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.<br><br>If a fluid homepage does not contain *any* tiles for components with Display on Small Form Factor selected, that fluid homepage displays an error message on the small form factor device, reading *This Homepage is Empty: None of the tiles configured for this homepage are supported on smart phones.*<br><br>**Note:** If a tile for a fluid component is not displayed on the fluid homepage, users can still access the component using other navigation features, such as the Navigator menu.<br><br>When a user personalizes a fluid homepage, a visual cue in the form of a smartphone icon appears next to a tile on the Add Tile dialog box indicating which tiles can be displayed on the homepage for a small form factor device. The smartphone icon appears for tiles associated with components with Display on Small Form factor selected.<br><br>When personalizing a homepage using a smartphone, tiles associated with components that do not have Display on Small Form Factor Homepage selected display a note reading *Tile is not supported on this device.*<br><br>See "Working with Tiles on Small Form Factor Devices" (Applications User's Guide). |
| **No System Header Page** | Select if you have a custom header page that you want to display instead of the system header page (the default is PT_HEADERPAGE). |
| **No System Side Page** | Select if you have a custom side page for the right panel that you want to display instead of the system side page. The default, PT_SIDE2PAGE, defines the layout of the Related Information panel, which can include fluid related content and simplified analytics. |

| Field or Control | Description |
|---|---|
| **Component Type** | Select the type of component you are building:<br><br>• *Standard.* Similar to standard, classic components, this type is used for most fluid components that will include fluid application pages.<br><br>• *Activity Guide.* This component type applies to custom fluid activity guides—that is, those not created using the **PeopleTools** > **Activity Guides** > **Activity Guide Templates** menu item. Activity guides aggregate a step-wise procedure or activity that spans multiple components.<br><br>See "Understanding Activity Guides" (Portal Technology) for more information on how to create activity guide templates instead of creating activity guide components.<br><br>• *Master/Detail.* Used for a special type of fluid component, similar to an activity guide, master/detail components provide access to multiple components, aggregating the navigation to these components in a master list, which remains displayed in the left panel. 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. |

| Field or Control | Description |
|---|---|
| **Search Page Type** | Select the type of search the user will interact with when accessing the component:<br><br>• *None.* Disables search pages for the component.<br><br>• *Standard.* Displays a "standard" search page, which provides users with search results that they click, and drill into component pages and data replacing the entire browser window.<br><br>• *Master/Detail.* Displays a two-panel search results page similar to fluid master/detail components, which also provides 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 detail area.<br><br>For Standard and Master/Detail search types, either a Search Framework search definition or a pivot grid model needs to be associated with the component. For more detailed information on these options and how they are configured, see <u>Implementing Search for Fluid Components</u>.<br><br>**Note:** When you enable either a *Standard* or *Master/Detail* search type, the system displays Next in List, Previous in List, and Return to Search buttons by default. |
| **Enable Configurable Search** | Enables configurable search for the component, however you must exercise caution before you select this option because you can also enable configurable search for a component using the Manage Configurable Search page.<br><br>See "Guidelines on Using Configurable Search for Fluid Components" (Search Technology).<br><br>The Enable Configurable Search check box is enabled when the Search Page Type is set to Standard.<br><br>The Enable Configurable Search check box is unavailable when the Search Page Type is set to Master/Detail or None. |

## Header Toolbar Actions

Select the options to be displayed in the fluid banner:

| Field or Control | Description |
|---|---|
| **Disable All Actions** | Select to deselect all header toolbar actions. At runtime, the system displays a fluid banner containing no buttons. |
| **Logout** | Select for the Sign Out item to appear in the Actions menu.<br><br>This feature signs the user out of the PeopleSoft application, returning the user to the system signon page. |
| **Home** | Select for the Home button to appear on the fluid banner.<br><br>This feature 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 fluid banner.<br><br>This feature returns the user to the previously visited item.<br><br>**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. |
| **Help** | Select for the Help item to appear in the Actions menu for the fluid pages in this component.<br><br>**Important!** This option is reserved for PeopleSoft-delivered components only.<br><br>When a user clicks Help, the system displays the PeopleSoft Online Help in a new browser page regardless of the form factor. The Help URL value must be specified the General tab of the web profile to enable any online help (same as for classic components and pages). For more information on configuring online help URL for your PeopleSoft system, see "Configuring General Portal Properties" (Portal Technology).<br><br>**Note:** For fluid applications, you cannot launch online help from modal pages.<br><br>**Note:** For the Safari browser, running on iOS in full screen mode, PeopleSoft Online Help is displayed in a full modal page. |

| Field or Control | Description |
|---|---|
| **Notify** | Select for the Notify item to appear in the Actions menu.<br><br>When a user selects the Notify item, the Notify window opens and enables a user to send notifications to other users.<br><br>See "Sending Ad Hoc Notifications from Fluid Components" (Applications User's Guide).<br><br>See Managing Global Settings.<br><br>**Note:** The Notify option is not available for Activity Guide because Activity Guide has its own mechanism to notify. |
| **Notifications** | Select for the Notifications button to appear on the fluid banner. When a user clicks the Notification button, the Notification window shows the list of push notifications for that user, such as actions and alerts. See "Using the Notifications Panel" (Applications User's Guide) for more information.<br><br>Deselect this option to disable notifications while within a specific component. |
| **NavBar** | Select for the NavBar button to appear on the fluid header, providing users with additional navigation features, such as the Navigator menu, Favorites, and other NavBar tiles. |
| **Add To** | Select for Add To items to appear in the Actions menu.<br><br>This feature enables a user to add a tile for the component to a fluid homepage, the NavBar, or Favorites.<br><br>Users granted permission to the WEBLIB_PTNUI web library will have access to all three Add To links from the Actions menu unless their permissions are modified by one or more of these roles:<br><br>• DisableAddToHomepage – Disables access to the Add To Homepage link.<br><br>• DisableAddToNavBar – Disables access to the Add To NavBar link.<br><br>• DisableAddToFavorites – Disables access to the Add To Favorites link. |

| Field or Control | Description |
| --- | --- |
| **New Window** | Select for the New Window item to appear in the Actions menu. This feature allows the user to open the current component in a new browser tab (or window). |

### Related Links
"Working with the Controls in the Fluid Banner" (Applications User's Guide)

## Setting Properties on the Use Tab

**Note:** In the current release, spell check is not supported at the component level for fluid components. Do not select the Mandatory Spell Check option on the Use tab.

## Setting Properties on the Internet Tab

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" (Application Designer Developer's Guide).

**Note:** In the current release, spell check is not supported at the component level for fluid components. Do not select the Spell Check toolbar option on the Internet tab.

## Setting the Viewport

Setting the viewport for each fluid component ensures that your fluid applications operate properly across all device types, particularly on small form factor devices such as smartphones.

The viewport meta tag in HTML is a crucial element of delivering responsive web applications. Without it, text and other page elements are not rendered at the correct size across all device types. To ensure that your fluid applications are rendered correctly across all device types, you must set the viewport. The recommended methodology is to use the SetDefaultViewport method in the component PostBuild event or the page Activate event:

```
import PT_PAGE_UTILS:*;
```

```
(create PT_PAGE_UTILS:Utils()).SetDefaultViewport();
```

**Note:** You can use the SetCustomViewport method as an alternative to SetDefaultViewport.

**Note:** Similar to fluid components, you must also set the viewport for each modal secondary window to ensure that they also operate properly across all device types. See <u>Working with Secondary Pages</u>.

### Related Links

"SetDefaultViewport" (PeopleCode API Reference)

# 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 components, the page tabs appear across the top of the page, always visible to the end user.

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 interface.

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.



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.



---
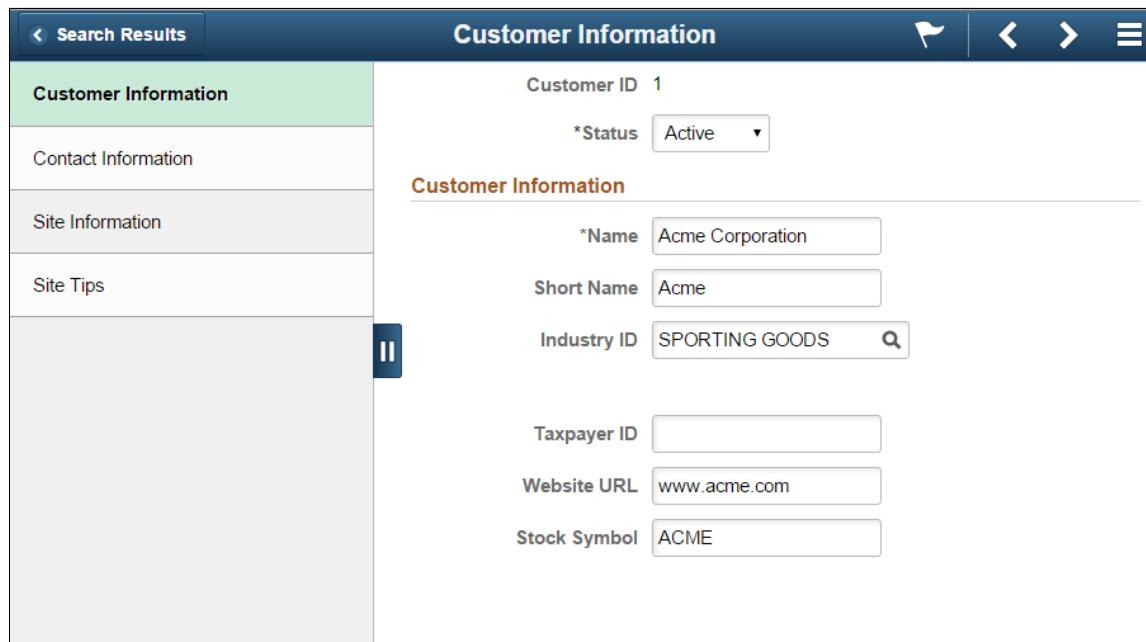
# Implementing Search for Fluid Components

This topic discusses implementing search for fluid components.

### Related Links
"Working with Search in Fluid Applications" (Applications User's Guide)

## 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.<br><br>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.<br><br>See Setting up Keyword Search for Fluid Components. |
| Configurable Search | Configurable search enables you to configure the presentation of your search page (standard or keyword) for an individual component or system wide.<br><br>See "Guidelines on Using Configurable Search for Fluid Components" (Search Technology). |

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

For classic components, 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 "Creating a Component Pivot Grid Model Using the Pivot Grid Wizard" (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" (Pivot Grid)
"Creating a Component Pivot Grid Model Using the Pivot Grid Wizard" (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. 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 component.

| Task | Documentation Cross Reference |
|------|-------------------------------|
| Create a query for the search definition. | See "Understanding Search Definition Queries" (Search Technology). |
| Create a search definition. | See "Creating Search Definitions" (Search Technology). |
| Map the search definition to your fluid component. | See "Mapping Components to Search Definitions" (Search Technology). |
| Create a search category. | See "Understanding Search Categories" (Search Technology). |
| Deploy and build your search index. | See "Administering Search Definitions and Search Categories" (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 or by specifying a search record that does not have a search key defined.<br><br>• *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.<br><br>• *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.<br><br>**Note:** Standard or Master/Detail search types may be used for both real-time or keyword search options. |
| Enable Configurable Search | Enables configurable search for the component, however you must exercise caution before you select this option because you can also enable configurable search for a component using the Manage Configurable Search page.<br><br>See "Guidelines on Using Configurable Search for Fluid Components" (Search Technology). |
| 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" (Application Designer Developer's Guide)
"Working with Configurable Search" (Search Technology)

## Setting Up Configurable Search for Fluid Components

Configurable search enables you to configure search pages for classic and fluid components. Administrators can configure search pages for components to use either a search record (standard search) or a search definition (keyword search).

For guidelines on using configurable search for fluid components and for instructions on configuring search pages, see "Working with Configurable Search" (Search Technology)

### Related Links
"Understanding Configurable Search" (Search Technology)

## Embedding the Suggestions Search Box

You can embed the suggestions search box on a page within a component to provide search capabilities in addition to the search configured for the component. The suggestions search box functionality is provided by PTS_INTELLISCHS_FL, a PeopleTools-delivered subpage. Similar to keyword search, you must use the Search Framework to create a search index for use with the suggestions search box. That index serves as the basis for the suggestions displayed as the user types.

In this example, the search suggestions are retrieved from the PTPORTALREGISTRY index.



Use this process to embed the suggestions search box on a page within a component:

1. Use the Search Framework to create a search definition and category, and then index the searchable data.

2. Insert a container group box for the suggestions search box. On the Fluid tab, enter *pts_pagesearch_div* as an application-specific style class.

3. Insert the PTS_INTELLISCHS_FL subpage into the group box.

   If you want to display a search drop-down list box from which you can select search categories that are different from the search context, you should insert a subpage in the PTS_INTELLISCHS_FL subpage. The subpage uses a dynamic view.

4. Define page Activate PeopleCode to associate the search index with the suggestions search box and set other properties for its operation. For example:

```
import PTSF_SEARCHUI:SearchbarContext;
Component PTSF_SEARCHUI:SearchbarContext &sc;

&sc = create PTSF_SEARCHUI:SearchbarContext();
&sc.Placeholder = MsgGetText(240, 4250, "Search for Component Cref");
```

```
&sc.CustomSearchCategory = "PTPORTALREGISTRY";
&sc.IsIncludeHiddenCref = True;
&sc.IsComponentOrGenericURL = True;
&sc.ShowGridIcons = False;
&sc.ShowCategoryDropDown = False;
&sc.ShowGlobalSearchButton = False;
&sc.SetSearchbarContext();

SetCursorPos(%Page, PTS_INTSRCH_WK.PTS_KEYWORDS_SS);
```

**Note:** Certain properties of the PTSF_SEARCHUI:SearchbarContext class apply to the PTPORTALREGISTRY index only.

To implement search categories drop-down list box, use the ArrCustomSearchCategory property.

```
import PTSF_SEARCHUI:SearchbarContext;

&sc = create PTSF_SEARCHUI:SearchbarContext();
&sc.KeywordLabel = "Keywords ";
&sc.SrchWidgetId = %Request.FormName | "divPTS_SEARCHWIDGET";
&sc.CustomSearchCategory = "PTPORTALREGISTRY";

&sc.ArrCustomSearchCategory.Push("QE_PIASRCH_JOB_OPENINGS");
&sc.ArrCustomSearchCategory.Push("QE_FILE_PIASRCH");
&sc.ArrCustomSearchCategory.Push("QE_NUI_ITM_NM");

&sc.SetSearchbarContext();
```

5. If your use of search suggestions is to allow the user to navigate to a search result, then no additional PeopleCode is required. However, if you intend to use the selected search result for other purposes (for example, to create a configuration based on the selected content reference), then you will need to write additional PeopleCode to make use of the selected result.

6. Create a FieldChange PeopleCode program on the *component* record field, PTS_INTSRCH_WK.PTPORTAL_OBJNAME_S.

   The following example is an excerpt of the relevant code demonstrating how to use the value returned in the PORTALOBJSS field to search in the portal registry for the content reference (CREF) associated with this portal object ID. The code excerpt then hides the container group box created in step 2 above and reveals a work grid for completing some configuration associated with the CREF.

```
Component string &PortalObjName, &strPnlGrpName, &strMenuName;

/* The CrefName/Portal obj name of the component selected from the suggestions⇒

 will be stored in this field by the search code via their subpage. */
Local string &PortalObjNm = GetPageField(%Page, "PORTALOBJSS").Value;

Local ApiObject &Portal1, &Cref;

&Portal1 = %Session.GetPortalRegistry();
Local boolean &IsOpen = &Portal1.Open(%Portal);


If &IsOpen And
      All(&PortalObjNm) Then
   &Cref = &Portal1.FindcrefByName(&PortalObjNm);

   /* Hide the group box containing the suggestions subpage. */
   Local Field &SrchGrpBox = GetPageField(%Page, "PT_EVM_SRCH_GRPBOX");
   &SrchGrpBox.Visible = False;

   /* Reveal the group box containing the work grid. */
   Local Field &GridGrpBox = GetPageField(%Page, "PTCS_EVMAP_CFGSGBX");
   &GridGrpBox.Visible = True;
```

```
      /* Application-specific configuration based on the selected CREF. */

   End-If;
   &Portal1.Close();
```

### Related Links

"SearchbarContext Class Properties" (PeopleCode API Reference)
"Working with Suggestions Search" (Search Technology)

# Understanding Delivered Search Pages

The search pages the system displays for classic 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_CFGSRCH_FL | This is a delivered search page for configurable search. You can use this page to set up standard and keyword search pages.<br><br>For more information on configurable search and how to enable configurable search for fluid components, see "Guidelines on Using Configurable Search for Fluid Components" (Search Technology). |
| PTS_NUI_SEARCH | Standard search page.<br><br>**Note:** If configurable search is enabled for a component or a fluid content reference is set up for configurable search through browser, this search page is replaced with the search page delivered for configurable search (PTS_CFGSRCH_FL). |
| PTS_NUI_SEARCH_S | Master - Detail search page.<br><br>**Note:** When Master - Detail search page is selected, you cannot enable configurable search. |
| PTS_NUI_GBLSRCH | Fluid global search page.<br><br>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 Searchbar in the banner.<br><br>For more information on global search, see "Working with Global Search" (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 Page 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 search 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 applications and fluid applications, not every search feature that is available for classic applications 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 applications and the fluid applications.

| *Feature* | *Fluid Applications* | *Classic Applications* |
|---|---|---|
| Images in results | Yes | Yes |
| Related actions in component search results | Yes | Yes |
| Related actions in keyword search results | Yes | Yes |
| Bulk related actions | Yes | No |
| Bulk related actions (configurable search enabled) | No | No |
| Conditional Global Search | Yes | Yes |
| Facets in component search results | Yes | No |
| Facets in keyword search results | Yes | Yes |
| List view in component search results | Yes | No |

| Feature | Fluid Applications | Classic Applications |
|---|---|---|
| List view in keyword search results | Yes | Yes |
| Chart view in component search results | Yes | No |
| Chart view in component search results (configurable search enabled) | No | No |
| Add a new value | Yes | Yes |
| Persistent search | No | No |
| Search suggestions | Yes | Yes |

**Important!** Persistent search, the drop-down menu, and other forms of classic navigation are no longer supported. The default navigational interface for PeopleSoft applications in the current release is based on the fluid banner, which can be used for both classic and fluid applications.

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 |
| Related Actions (Configurable search enabled) | Yes | Yes | Yes |
| Bulk Related Actions | Yes | Yes | No |
| Bulk Related Actions (Configurable search enabled) | No | No | No |
| Chart interaction in component search | Yes | Yes | No |
| Chart interaction in component search (Configurable search enabled) | No | No | 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. In this regard, master/detail components provide WorkCenter-like functionality for fluid applications. 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 typically navigate to these components and pages separately via the Navigator menu as needed.

### Example of a Master/Detail Application

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.
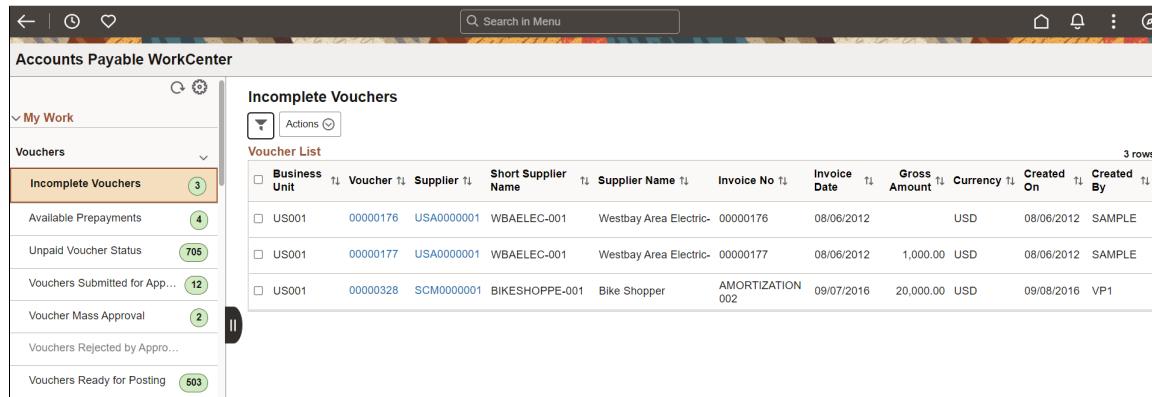


## Guidelines and Limitations for Master/Detail Components

Note the following guidelines and limitations for target content and master/detail components:

•   For a fluid component to be included as target content, it must be defined as a *Standard* fluid component.

> **Important!** A fluid wrapper component (that is, a fluid activity guide, another master/detail component, or fluid navigation collection) can never be included as target content inside a master/detail component.

•   Typically, for each page defined within the fluid component, the outermost group box must include the ps_apps_content style or another styling appropriate to the page design.

- Unless AJAX transfers are enabled on the master/detail component, the fluid component should not perform a transfer to another component. This includes transfers via a hyperlink, a push button, or a PeopleCode program. Such transfers will result in the user exiting the master/detail component. See <u>AJAX Transfers</u> for more information.

- If the target content is a fluid component that employs an internal two-panel construct (that is, it either includes a Side Page 1 page, was built from the PeopleTools 8.54 PSL_TWOPANEL layout page, or was designed with an internal left panel similar to PSL_TWOPANEL), you must enable left panel collision handling with the slide-out left panel on the master/detail component. See <u>Left Panel Collision Handling with Slide-out Left Panels</u> for more information.

- The fluid component must have the search page type set to *None* to be displayed directly as target content within a master/detail component. Otherwise, if the search page type is set to *Standard* or *Master/Detail,* the fluid component cannot be displayed as target content within a master/detail component. However, a link for the component can be included in the navigation list as long as the link destination is programmatically set to a new window or to replace the current window. See "SetGroupletDisplayIn" (PeopleCode API Reference).

- The fluid component definition must not include a Header Page page definition.

- Including a Footer Page page definition within a fluid component displayed as target content inside of a master/detail component may or may not follow usability standards and could create a confusing user experience.

- To use a multi-page fluid component in a master/detail component, you can do one of the following to maintain a clear and consistent user experience:

  - Reparent the page tabs to a group box of type Page Tab that is included on *each* page definition.

  - Include PT_SIDE_PAGETABS in the component definition to reparent the page tabs to the left panel. You must also enable left panel collision handling with the slide-out left panel on the activity guide. See <u>Left Panel Collision Handling with Slide-out Left Panels</u> for more information.

  - Hide the page tabs. You can do so in the component definition by deselecting the Display Folder Tabs (top) on the Internet tab. Alternatively, you can use delivered styles to hide the page tabs.

- If you want the master/detail component to maintain the same title in the fluid banner regardless of which page is displayed as target content, then set the title in whichever definition generates the content for the master list using the PeopleTools-delivered SetSideTopTitle JavaScript function. For example:

```
AddOnLoadScript("SetSideTopTitle('" | EscapeHTML("My M/D Component's Title") |⇒

 "');");
```

- If your master/detail component controls the title in the fluid banner (see the preceding bullet item), for each page defined within the target component, you should include a group box of type Page Title that is defined to be visible when the page is rendered within a master/detail component. This Page Title group box provides context that is lost when the page title is not displayed in the fluid banner. To hide this page title if the page is rendered outside of a master/detail component, use the psc_hideinmode-notmd style.

# Understanding the Elements of Master/Detail Components

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

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.
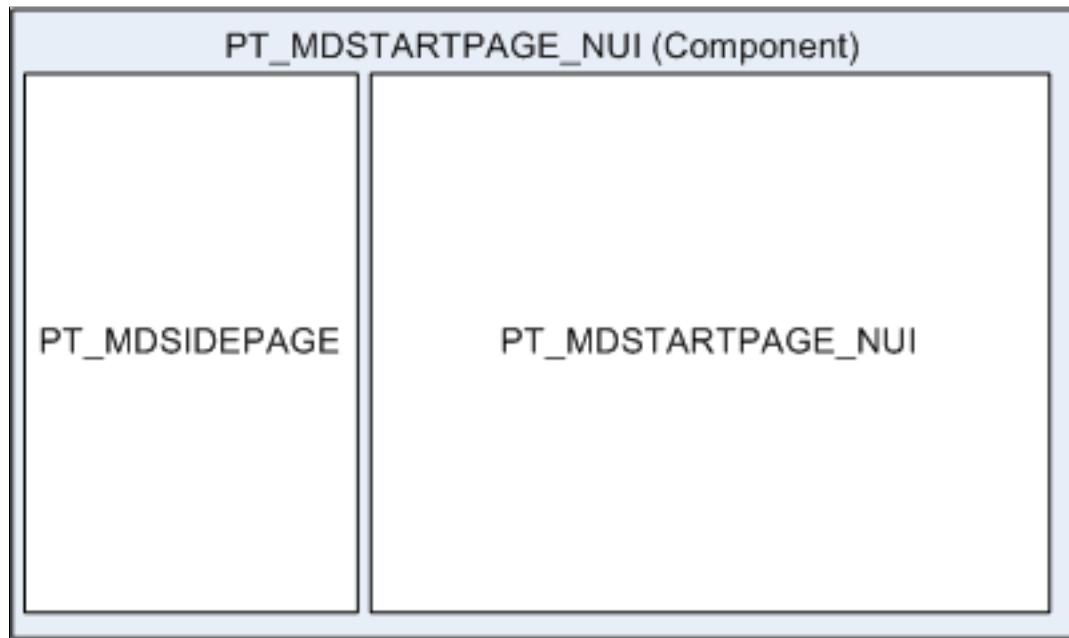


## Master/Detail Component

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.

This example illustrates the PT_MDSTARTPAGE_NUI component definition.
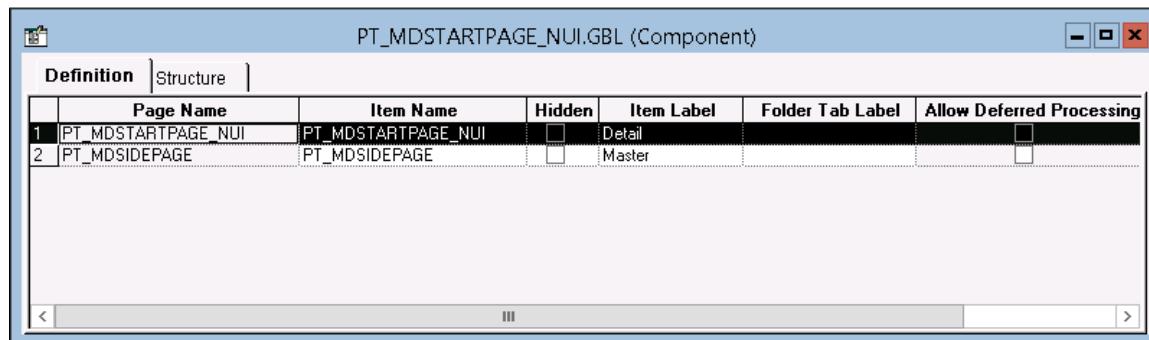


## Master/Detail Side Page

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.

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*.
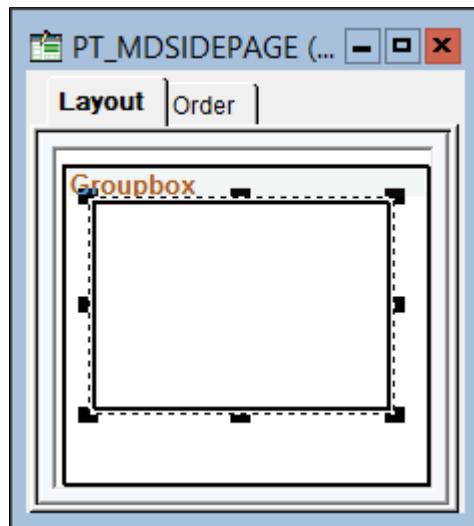


## Master/Detail Target Page

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.

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*.
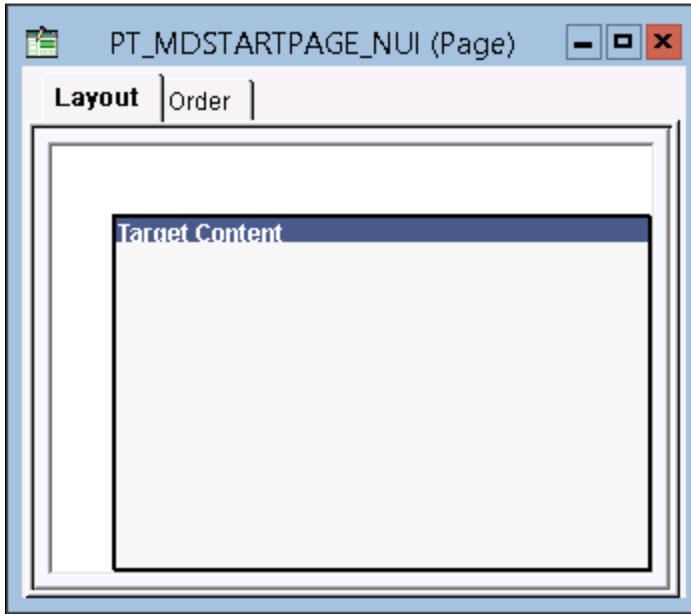


## Implementing Master/Detail Components

While you can clone the delivered PT_MDSTARTPAGE_NUI component and the pages within that component, the following procedure demonstrates how you can reuse the delivered definitions to load the master list from a different component.

To load the master list from a different component:

1.  Create a dynamic view record definition—for example, MY_NAVLIST_DVW. Add these fields to the dynamic view:

    •  A numeric field for the hyperlink to the target content—for example, LINENUM.

    •  A numeric field if a dynamic counter needs to be displayed along with the link—for example, COUNTER.

2.  Create a derived/work record definition—for example, MY_MDLAYOUT.

    Add one field to this derived/work record: GROUPBOX1, a character field.

3.  Create a standard page definition—for example, MY_NAVLIST. On the Use tab, select the Fluid Page check box.

4.  Add a group box that covers the entire page. On the Fluid tab, define these properties of the group box:

    •  In the Style Classes field, enter *psc_scroll*.

    •  For the Group Box Type field, select *Layout Only*.

- Select the Scroll Sizing by Style check box.

5. Add a grid to the page.

6. Add a group box as the first column to the grid. Associate it with the MY_MDLAYOUT.GROUPBOX1 field. Define these page field properties of the group box:

   - On the Fluid tab, in the Style Classes field, enter *ps_vtab*.

   - On the Fluid tab, for the Group Box Type field, select *Layout Only*.

7. Define these properties of the grid:

   - On the General tab, select the Unlimited Occurs Count (rows) check box.

   - On the Label tab, click the Display Navigation Bar: Properties button; on the Row Cntr tab, select the Invisible check box. Click OK to close the dialog box.

   - On the Label tab, deselect the following three check boxes: Show Border, Show Row Headings (runtime), and Show Column Headings (runtime).

   - On the Use tab, select the following two check boxes: No Auto Select, and No Auto Update.

   - On the Use tab, for the Grid Layout field, select *List Grid Layout (Unordered)*.

   - On the Fluid tab, in the Style Classes field, enter *psc_list-linkmenu psc_wrap psc_list-has-icon psc_list-separator-dark psc_border-standard psc_border-bottomonly psc_list-icons-active*.

   - On the Fluid tab, in the Custom Grid Header field, select *0 | None*.

8. Add a push button/hyperlink as the second column in the grid. Associate it with the MY_NAVLIST_DVW.LINENUM field. Define these properties of the hyperlink:

   - On the Type tab, for the Type field, select *Hyperlink*.

   - On the Label tab, select the Both check box.

   - On the Fluid tab, for label rendering, select the After Control radio button.

9. If you wish to display a dynamic counter as part of the link to the target content, add an edit box as the third column in the grid. Associate it with the MY_NAVLIST_DVW.COUNTER field. Define these properties of the edit box:

   - On the Fluid tab, in the Style Classes field, enter *psc_list_count*.

   - On the Fluid tab, for the Input Type field, select *text*.

10. Insert a group box around the grid. On the Fluid tab, for the Group Box Type field, select *Layout Only*.

This example illustrates the appearance of the MY_NAVLIST page definition after all items have been added and defined.
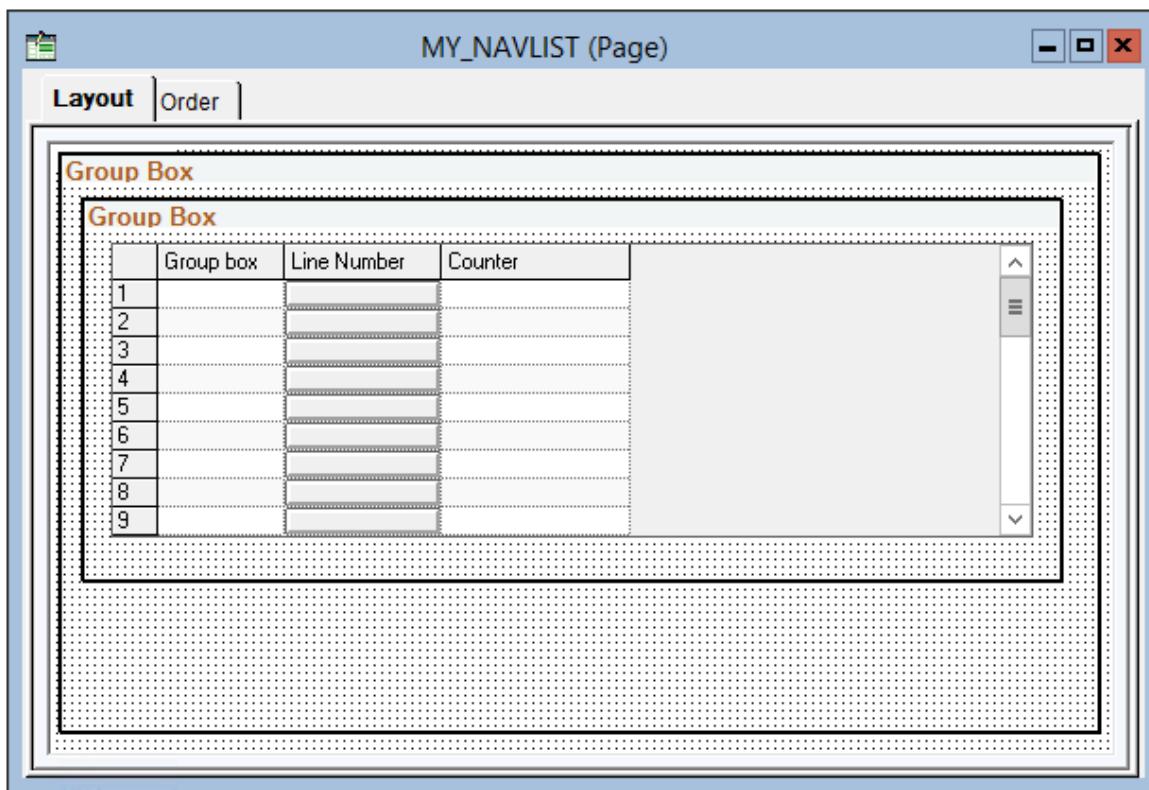


11. Add a page Activate PeopleCode program to the page definition. Your PeopleCode program must:

- Get the initial row from the URL parameters and then set the active row in the master list.

- For each link in the master list, add a row to the rowset representing the grid on your page.

- For each link, generate the URL to the target content. For example, use the GenerateComponentContentURL built-in function.

- Specify this URL as the action URL of the MY_MDLAYOUT.GROUPBOX1 group box field.

- For the MY_NAVLIST_DVW.LINENUM hyperlink field, specify the item number in the master list, the label to be displayed for this list item, and an optional image.

- For the optional the MY_NAVLIST_DVW.COUNTER dynamic counter, retrieve and set the dynamic value.

The following example page Activate program creates four items in the master list: a fluid component, a fluid component with additional URL parameters, a classic component, and a fluid dashboard. The same optional image is reused for each list item. In addition, the counter for each item is hard-coded and not dynamically retrieved.

```
Local string &sNewURL, &sMenu, &sComp, &sPage;
Local number &nRow = 0;

Local Rowset &rs = GetLevel0()(1).GetRowset(Scroll.MY_NAVLIST_DVW);

If &rs <> Null Then
```

```
    /* Get the initial row number from the URL. */
    If (%Request.GetParameter("nRow") <> "") Then
        &nRow = Value(%Request.GetParameter("nRow"));
    End-If;
    /* Set the active row in the master list. */
    If (IsGroupletRequest()) Then;
        AddOnLoadScript("DoRowAction(" | &nRow | ");");
    End-If;

    /* Use SetSideTopTitle to maintain the same title in the fluid banner regar⇒
dless */
    /* of what is displayed as target content. Use a Message Catalog entry.   ⇒
     */
    /* AddOnLoadScript("SetSideTopTitle('" | EscapeHTML("My M/D Component's Tit⇒
le") | "');"); */

    /* Specify the first row in the master list. */
    Local integer &n = &rs.ActiveRowCount;
    &sMenu = "MenuName.MY_MENU";
    &sComp = "Component.MY_COMP1_FL";
    &sPage = "Page.MY_PAGE1";
    &sNewURL = GenerateComponentContentURL(%Portal, %Node, @&sMenu, %Market, @&⇒
sComp, @&sPage, "U");
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletActionUrl(&sNewURL);
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletDisplayIn(%GroupletActionMDTargetF⇒
luid);
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.IsGroupletLabelOnly = True;
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.Visible = False;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Value = &n;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Label = "Fluid Component 1";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.HoverText = "Fluid Component 1";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.LabelImage = @("Image.CHECKLIST_64");
    &rs(&n).MY_NAVLIST_DVW.COUNTER.Value = 5;

    /* Insert another row in the master list. */
    &rs.InsertRow(&n);
    &n = &rs.ActiveRowCount;
    &sMenu = "MenuName.MY_MENU";
    &sComp = "Component.MY_COMP1_FL";
    &sPage = "Page.MY_PAGE2";
    /* Includes additional URL parameters. */
    &sNewURL = EncodeURL(GenerateComponentContentURL(%Portal, %Node, @&sMenu, %⇒
Market, @&sComp, @&sPage, "U") | "&CONNGATEWAYID=LOCAL");
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletActionUrl(&sNewURL);
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletDisplayIn(%GroupletActionMDTargetF⇒
luid);
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.IsGroupletLabelOnly = True;
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.Visible = False;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Value = &n;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Label = "Fluid Component 2";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.HoverText = "Fluid Component 2";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.LabelImage = @("Image.CHECKLIST_64");
    &rs(&n).MY_NAVLIST_DVW.COUNTER.Value = 12;

    /* Insert the third row in the master list. */
    &rs.InsertRow(&n);
    &n = &rs.ActiveRowCount;
    &sMenu = "MenuName.MY_OTHER_MENU";
    &sComp = "Component.MY_CLASSIC_COMP";
    &sPage = "Page.MY_PAGE3";
    &sNewURL = GenerateComponentContentURL(%Portal, %Node, @&sMenu, %Market, @&⇒
sComp, @&sPage, "U");
```

```
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletActionUrl(&sNewURL);
    /* Classic components require a different value. */
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletDisplayIn(%GroupletActionMDTargetC⇒

lassic);
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.IsGroupletLabelOnly = True;
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.Visible = False;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Value = &n;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Label = "Classic Component 1";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.HoverText = "Classic Component 1";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.LabelImage = @("Image.CHECKLIST_64");
    &rs(&n).MY_NAVLIST_DVW.COUNTER.Value = 1;

    /* Insert the fourth row in the master list. */
    &rs.InsertRow(&n);
    &n = &rs.ActiveRowCount;
    &sMenu = "MenuName.PT_FLDASHBOARD";
    &sComp = "Component.PT_FLDASHBOARD";
    &sPage = "Page.PT_LANDINGPAGE";
    /* Specify which fluid dashboard as a URL parameter. */
    &sNewURL = EncodeURL(GenerateComponentContentURL(%Portal, %Node, @&sMenu, %⇒

Market, @&sComp, @&sPage, "U") | "&DB=QE_CHART_DB");
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletActionUrl(&sNewURL);
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.SetGroupletDisplayIn(%GroupletActionMDTargetF⇒

luid);
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.IsGroupletLabelOnly = True;
    &rs(&n).MY_MDLAYOUT.GROUPBOX1.Visible = False;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Value = &n;
    &rs(&n).MY_NAVLIST_DVW.LINENUM.Label = "Chart Dashboard";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.HoverText = "Chart Dashboard";
    &rs(&n).MY_NAVLIST_DVW.LINENUM.LabelImage = @("Image.CHECKLIST_64");
    &rs(&n).MY_NAVLIST_DVW.COUNTER.Value = 0;

End-If;
```

This example illustrates the rendering of the master list based on the preceding PeopleCode example. The initial list item (nRow=2) is selected by default when the master/detail application is opened.
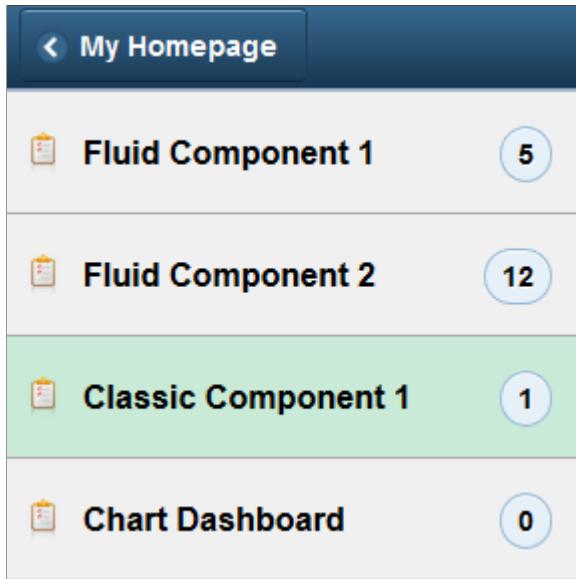


12. If you want the master/detail component to maintain the same title in the fluid banner regardless of which page is displayed as target content, then remove the comment and use the SetSideTopTitle JavaScript function to set the title.

13. Create a fluid component definition—for example, MY_NAVLIST:

    • On the Use tab, select *INSTALLATION* as the search record.

    • Insert the MY_NAVLIST page into the component.

14. Use the component Registration Wizard to add the component to a menu and permission list.

    **Important!** Do not register the component in the portal registry.

15. Manually, create a content reference definition to launch your master/detail application:

    a. In the browser, select **PeopleTools** > **Portal** > **Structure and Content**.

    b. Create the content reference definition in the menu folder where you want the menu item to appear.

       **Note:** To make your master/detail application available as a tile only and not as a menu item in the menu structure, create the content reference definition in the tile repository. See "Understanding Tile Definitions and the Tile Repository" (Portal Technology).

    c. In the content reference definition, specify the delivered PT_MDSTARTPAGE_NUI component in the URL information as follows:

| Field or Control | Description |
|---|---|
| Node Name | LOCAL_NODE |
| URL Type | PeopleSoft Component |
| Menu Name | NUI_FRAMEWORK |
| Market | GBL |
| Component | PT_MDSTARTPAGE_NUI |

This example illustrates the fields in the URL Information group box for a master/detail application based on the delivered PT_MDSTARTPAGE_NUI.

d. Add four additional parameters to the URL definition to identify the component you created to define your master list:

| Field or Control | Description |
|---|---|
| nRow | Specify the list item to be displayed when the master/detail is initially opened. A value of 0 represents row 1 in the list. |
| GMenu | (Required) Specify the menu name you used to register your component. |
| GComp | (Required) Specify the name of the component containing the master list. |
| GPage | Specify the name of the page containing the master list. |

e. Specify any additional URL parameters required for correct rendering and operation of your master/detail application—for example, whether AJAX transfers are allowed or whether to use the slide-out left panel. See Working with Two-Panel Implementations for more information.

### When to Clone the Delivered PT_MDSTARTPAGE_NUI Definitions

In some circumstances, to implement a master/detail application, you may want to clone the PT_MDSTARTPAGE_NUI component and the page definitions it includes. These situations include:

- You need to execute additional initialization PeopleCode for the master/detail component that is not in the PeopleTools-supplied component.

- You need to use more than one grouplet for the navigation list in the left panel.

### Related Links
"GenerateComponentContentURL" (PeopleCode Language Reference)

# 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. AJAX transfers apply to four PeopleCode functions only: Transfer, TransferExact, ViewContentURLClassic, and ViewContentURLFluid.

When AJAX transfers are not enabled (the default), the action exits the fluid wrapper and the content replaces the current window. If the transfer function is set to open a new window, the action opens a new window instead.

When AJAX transfers are enabled, the new content is displayed in the target content area of the fluid wrapper when the target is set to current window. If the transfer function is set to open a new window, this new window setting is overridden and the new content is also displayed in the target content area of the fluid wrapper *except* when the transfer is from classic content to other classic content. In that case, the new window setting is honored and the classic content is opened in a new window.

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.
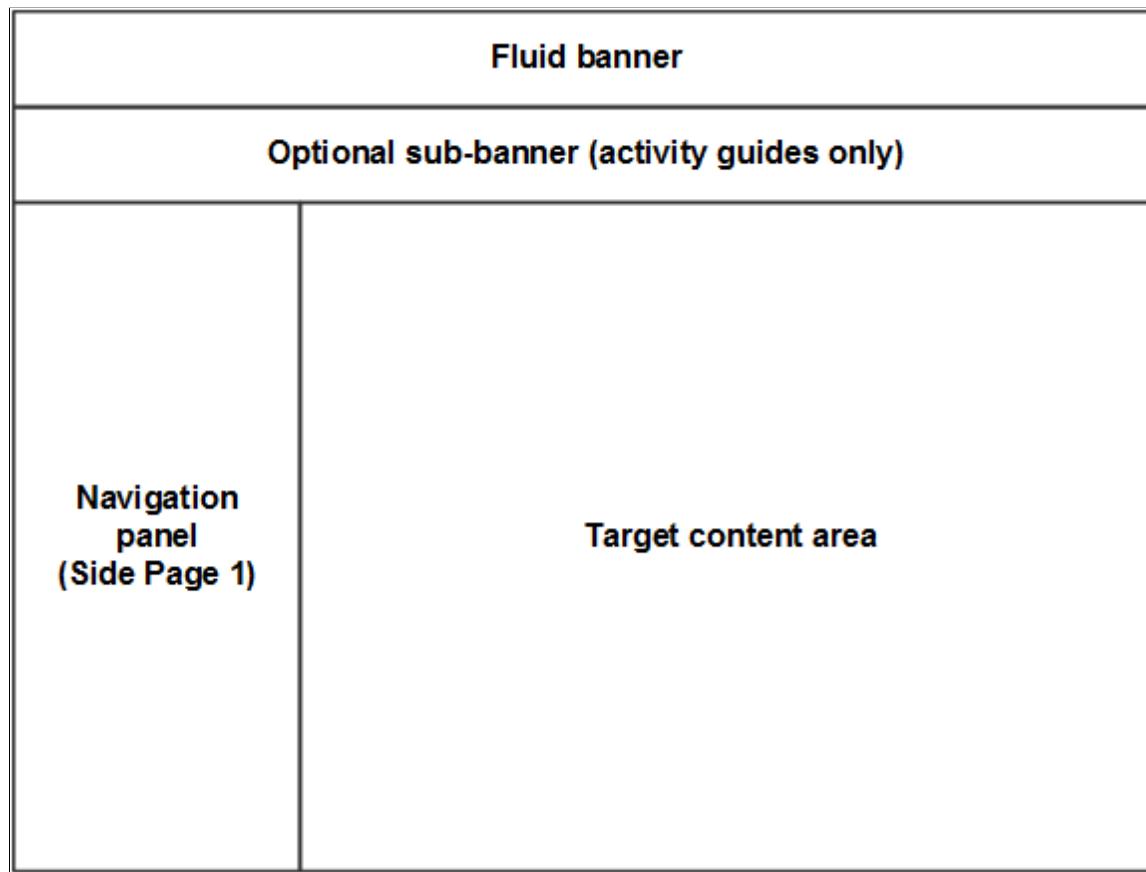


Since AJAX transfers are disabled by default, you must explicitly enable AJAX transfers on each fluid wrapper where the behavior is desired:

- 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" (PeopleCode Language Reference)
"SetMDListSlideout" (PeopleCode Language Reference)
"Transfer" (PeopleCode Language Reference)
"TransferExact" (PeopleCode Language Reference)
"ViewContentURLClassic" (PeopleCode Language Reference)
"ViewContentURLFluid" (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 one 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.

---

**Important!** A fluid wrapper component (that is, a master/detail component, a fluid activity guide, or a fluid navigation collection) can never be displayed inside another fluid wrapper component. Enabling the slide-out left panel cannot be used to overcome this inflexible limitation.

---

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.



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:

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" (PeopleCode Language Reference)
"IsSide1CollisionInterPage" (PeopleCode API Reference)
"IsSide1CollisionIntraPage" (PeopleCode API Reference)
"IsSide1HasCollision" (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.

**Important!** By default, the form factor trace option is disabled (hidden). To enable the Form Factor drop-down list to display on the sign-in page, you need to add a custom web property to the current web profile. Add ShowFormFactor and set it to *true.* For information on adding custom web profile properties, see "Configuring Custom Properties" (Portal Technology).

**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.

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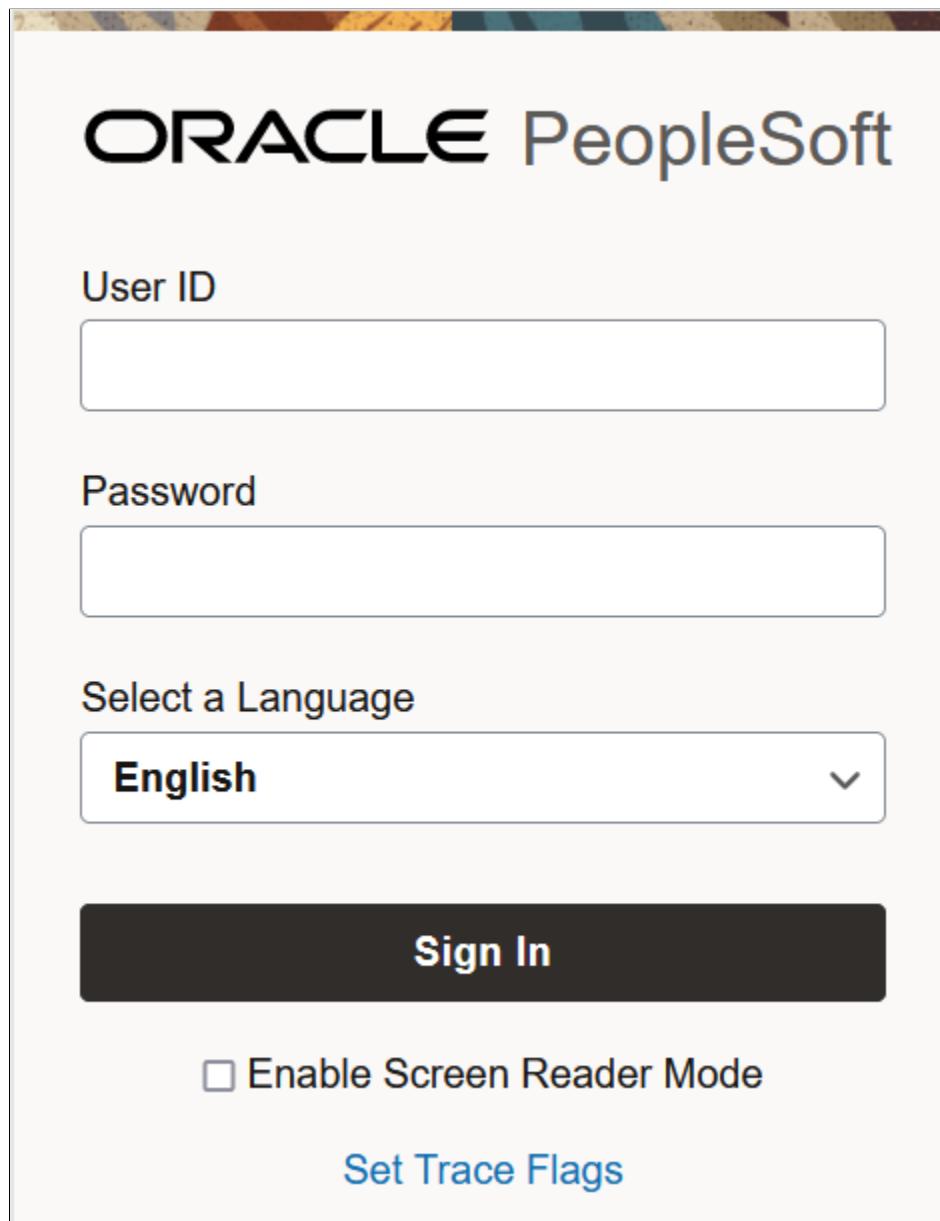


Once you click the Set Trace Flags link, the Select a Language drop-down list becomes the Form Factor drop-down list from which you can select the desired form factor.

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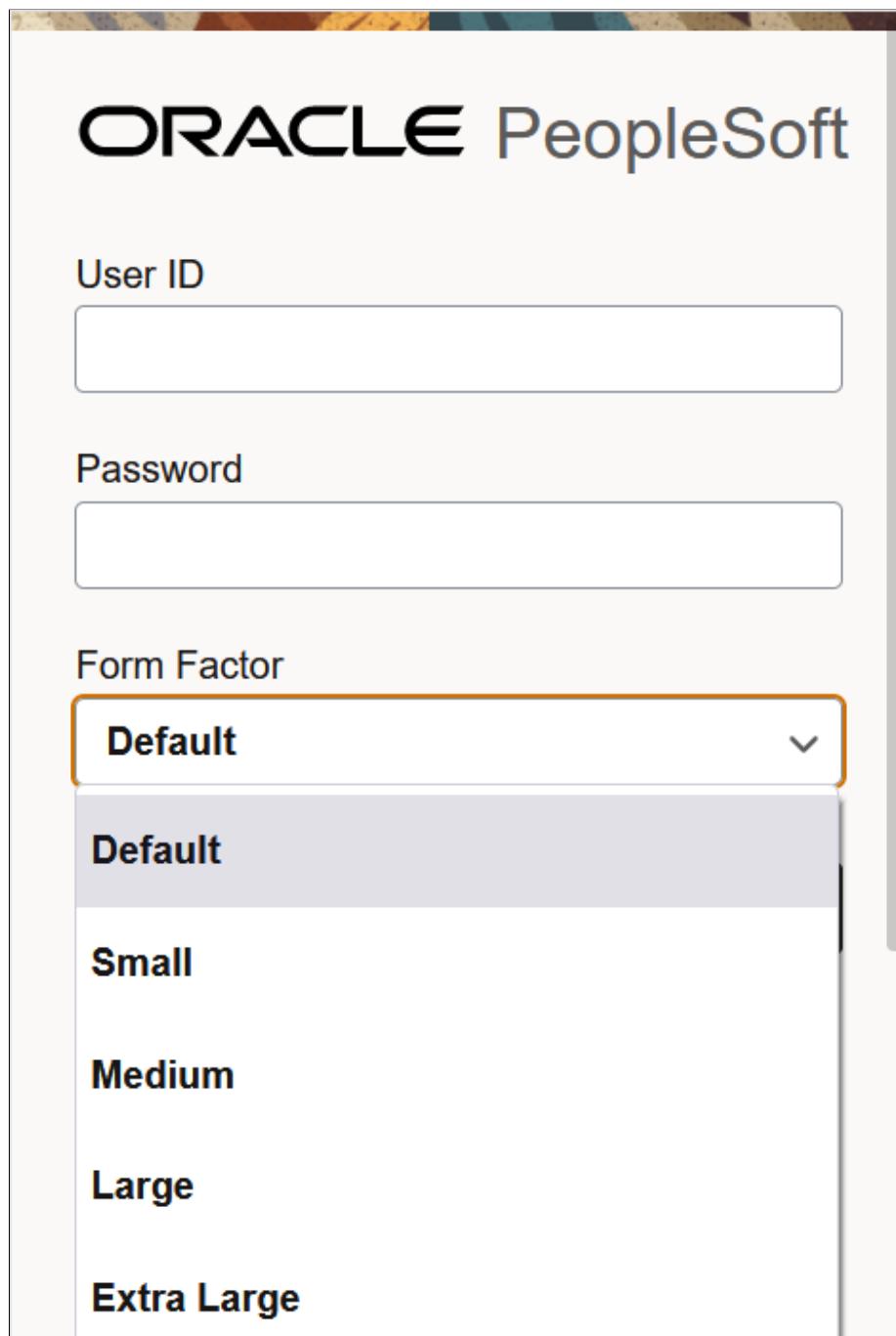


After you select a particular form factor, the system displays a second browser window rendered in the dimensions of the form factor you selected, displaying the sign-in page.

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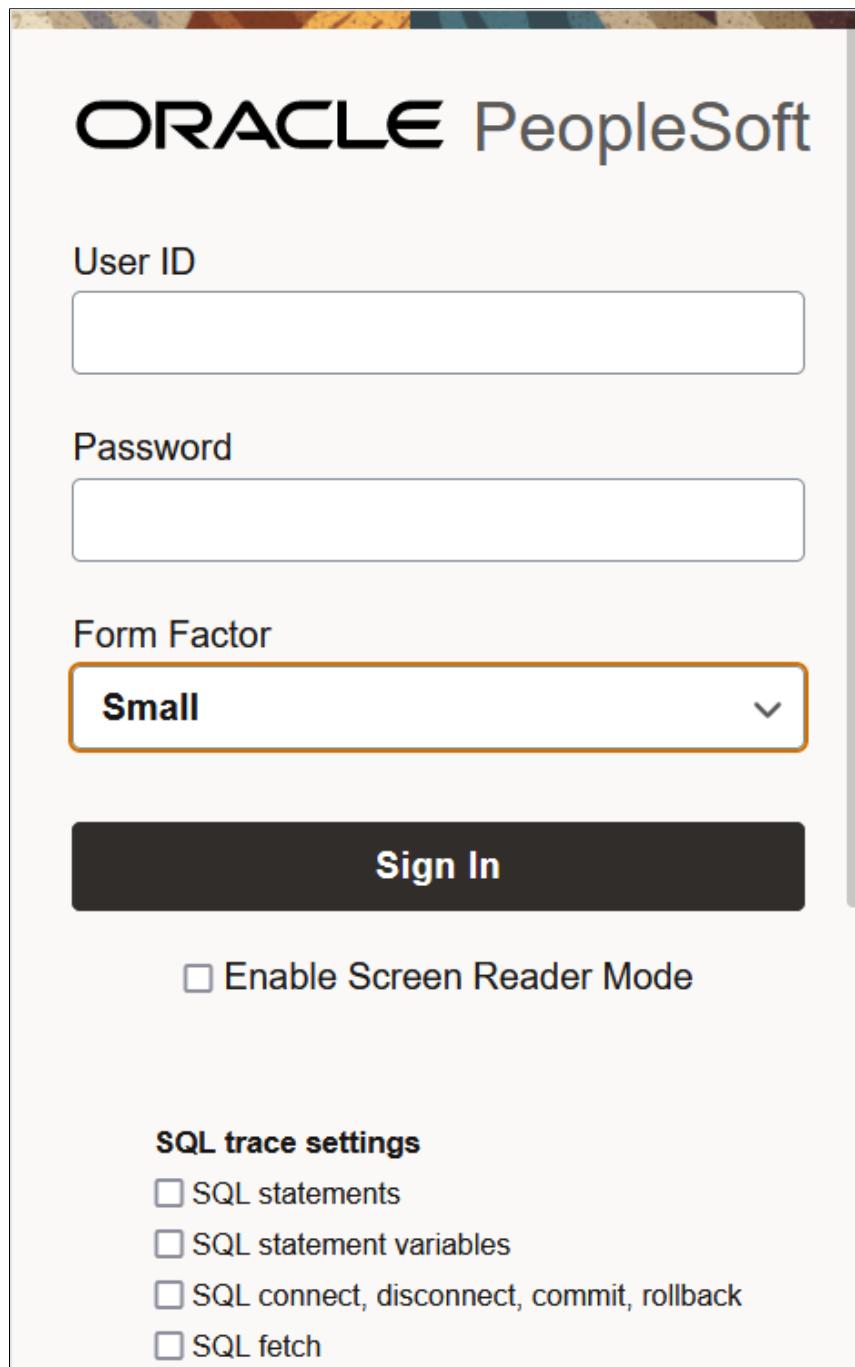


After signing in, you can navigate to and view the layout of a fluid page for the form factor you selected.

This example illustrates a fluid page rendering in the form factor you selected from the Form Factor trace option.
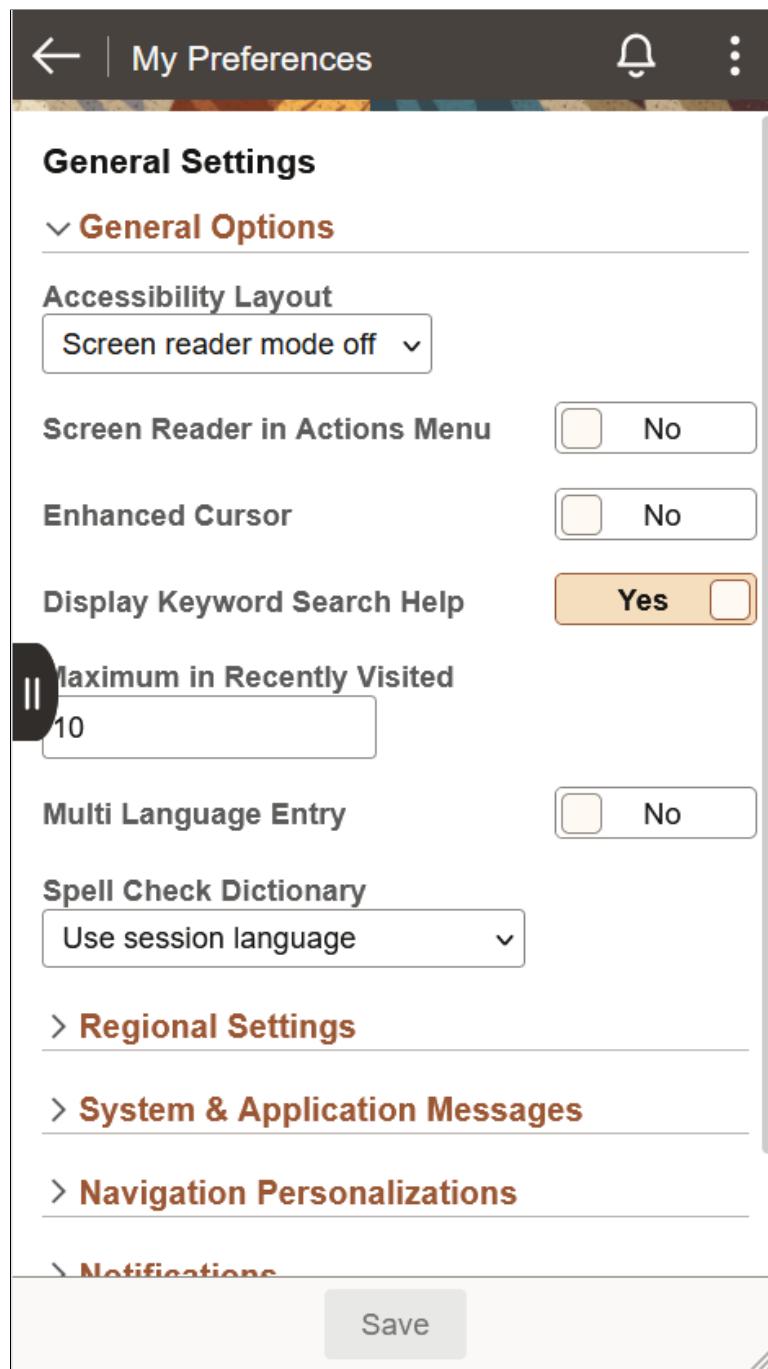
**Chapter 5**

# Adding Page Controls

## Managing Settings for Fluid Page Controls

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.

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 Properties on the Fluid Tab

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 page 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.

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



| Field or Control | Description |
| --- | --- |
| **Style Classes** | Enter the application-specific style classes to apply *to extend* the default system-specific styles (PeopleTools styles applied by default). Separate multiple style classes with a space. |

| Field or Control | Description |
| --- | --- |
| **Form Factor Style Class Override** | Use to override any application-specific styles based on the form factor (device size) of the device accessing the page. Form factors are represented in these ranges:<br><br>• *Small:* Applies to most smartphones.<br><br>• *Medium:* Applies to 7-inch tablets.<br><br>• *Large:* Applies to 10-inch tablets and laptops.<br><br>• *Extra Large:* Applies to large monitors.<br><br>For example, assume that the application-specific Style Classes field contains the style, psc_margin-bottom05em, which you intend to apply to all form factors except for the small form factor. In this case, for the Small field in the Form Factor Style Class Override group box, you enter the override style class to apply when a small form factor device accesses the page. |
| **Suppress System-Supplied Style Classes** | Select to suppress the system default styles applied by PeopleTools:<br><br>• If it is selected, it applies to *all* form factors, and the system applies no PeopleTools default classes for the control (neither does the system load state information). Any application-specific styles *override* the PeopleTools defaults.<br><br>**Warning!** When suppressing the default styles applied automatically by PeopleTools, you are then taking complete control of the CSS styling. This is not recommended.<br><br>• If it is not selected, any application-specific styles specified for Style Classes *extend* the default system styles, such that both styles are used. |
| **Suppress on Form Factor** | To prevent a page control from being displayed on a specific form factor, select the appropriate form factor. If selected, the page control 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 renders the display of a page control as a combined unit of both the control and its label. You can specify the position of the label.<br><br>*Before Control:* Select for the label to appear to the right of the control.<br><br>*After Control:* Select for the label to appear to the left of the control. |

| Field or Control | Description |
|---|---|
| **Control Structure** | *Basic:* Select to have the control appear on the fluid page just as it would appear on a classic page.<br><br>*Advanced:* Select to have the control appear on the fluid page with the look and feel of a modern, touch-enabled application. For example, a check box would appear as a toggle switch, rather than as a square box. |
| **Include Labels in Grid Cells** | If using a tab separator in your grid, you are allowed to generate field labels within a grid.<br><br>Typically, these are suppressed, but because you can put group boxes within grids, these labels are appropriate for the individual page fields inside the group box (depending on context). |

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.

This example illustrates how group boxes are used in the layout of the fluid page, clearly separating the page elements into distinguishable parts.
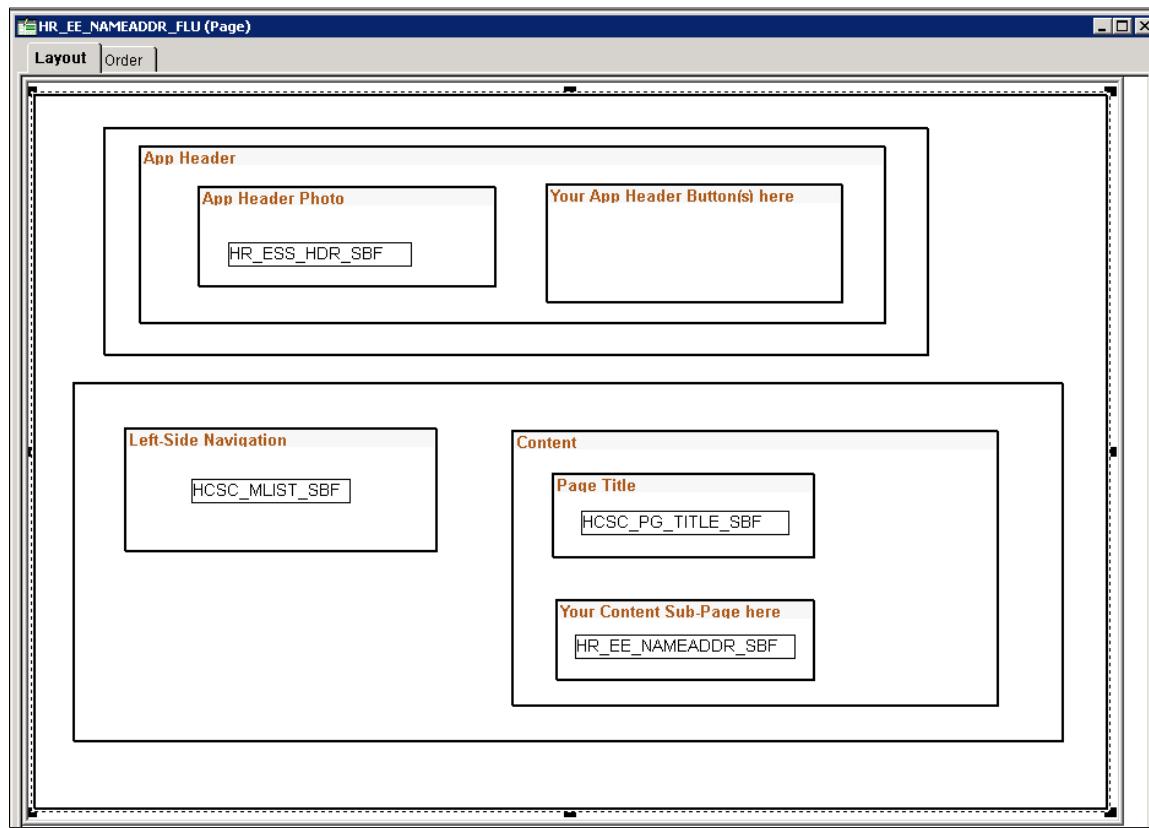


In general, it is recommended to be generous in the use of group boxes to layout and organize your page elements. However, group boxes are not just used for layout purposes. For some controls and widgets, it is required that they be encapsulated within a group box in order to apply behavior characteristics.

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.

**Note:** Embedded help is not supported for group boxes, grids, and scroll areas on fluid pages.

## Setting Group Box Properties

This section describes properties specific to group boxes on a fluid page that you can set on the Fluid tab of the Group Box Properties dialog box.

This example illustrates the options available on the Fluid tab of the Group Box Properties dialog box. Descriptions of the options follow the example.
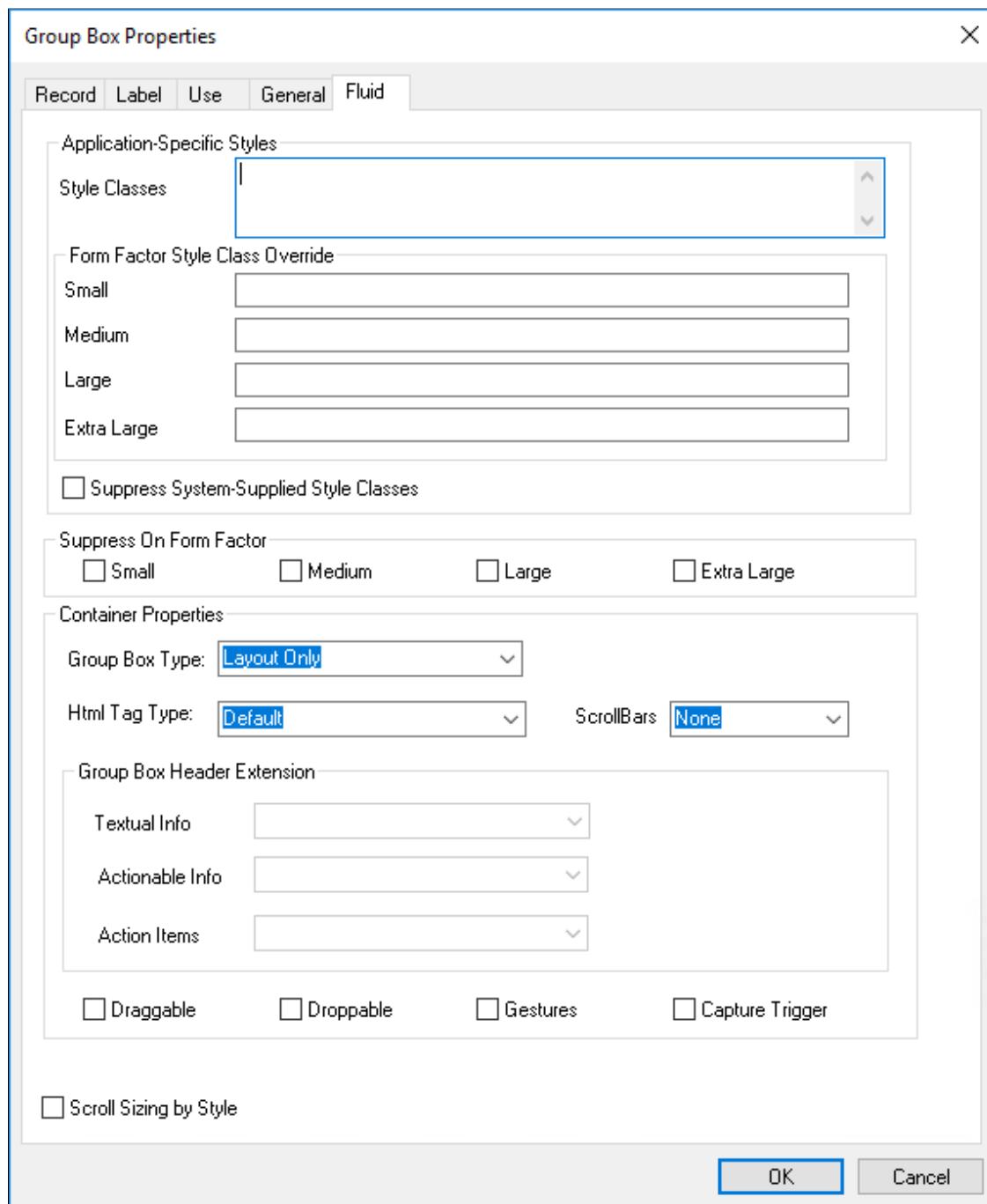


Style Classes and Form Factor Style Class Override are common properties to all fluid controls. See Setting Properties on the Fluid Tab.

| Field or Control | Description |
|---|---|
| **Group Box Type** | • *Default:* Same as *Container.* It includes both a header structure and a content structure.<br><br>• *Container:* Use as a standard container, just as in classic applications, to illustrate groupings of fields.<br><br>See <u>Working with Container Group Boxes</u>.<br><br>• *Accordion Vertical/Horizontal/Group Box:* Used to implement expandable sections.<br><br>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.<br><br>See <u>Working with Accordion Group Boxes</u>.<br><br>• *Layout Only:* Use for a group box only to contain and position child elements.<br><br>See <u>Working with Layout Group Boxes</u><br><br>• *Panel Control:* Use for containing content when creating a side tab.<br><br>• *Custom Header Left/Right/Middle/Bottom:* Use to add application controls to header page sections.<br><br>See <u>Creating Custom Header Group Boxes</u><br><br>• *Page Title:* Used to generate an H1 heading element (<h1>…</h1) that displays either the page title (as would be found in the fluid banner) or the content title (as replicated from the item selected in the left panel in a two-panel implementation). See <u>Working with Two-Panel Implementations</u>.<br><br>---<br>**Important!** Do not place any content or page controls within this group box container.<br>---<br><br>If you plan to allow the component that includes your page to be displayed within a fluid wrapper (a fluid activity guide or master/detail component), it is recommended that a Page Title group box (associated with a record field) be placed at the top of your page; this group box will display the content title. You should hide this Page Title group box when the component runs standalone and display it only when your component is displayed within a fluid wrapper. For activity guides (and fluid navigation collections), the activity guide framework automatically updates this page title found in the content area to match the selection made in the navigation panel.<br><br>• *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.<br><br>See <u>Working with Related Field Group Boxes</u><br><br>• *Toggle Container:* Use to switch between two group boxes and changes the label of the button controlling the toggle.<br><br>---<br>**Note:** Currently, do not use the Toggle Container option. This option is for internal use only until a future release.<br>--- |

| Field or Control | Description |
|---|---|
| | • *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.<br><br>• *Custom Action Menu:* Enables the transaction to add to the Actions List displayed in the banner.<br><br>• *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.<br><br>• *Custom Header Search:* Use to create a custom header for component search.<br><br>• *Page Tab:* Use to take page tabs running horizontally across the top of the component pages, and insert them into the left side page, creating left navigation. For sample usage, refer to PT_SIDE_PAGETABS.<br><br>• *Master/Detail Target:* Use to contain the target content to be displayed when an item in the master list of a master/detail component is selected. A group box of this type is required on a Master&Detail Target Page page type. In addition, this group box must:<br><br>    • Be associated with a record field on the Record tab.<br><br>    • Be named PAGECONTAINER_TGT on the General tab.<br><br>    • Have the ps_ag-targetcontent application-specific style applied on the Fluid tab.<br><br>For more information on the master/detail approach, see <u>Working with Master/Detail Components</u>.<br><br>• *Search Action:* Use to synchronize component search actions to the actions displayed in the Actions List (or other area in the header if preferred). This enables a PeopleCode event from the selected action, such as Return To List.<br><br>**Note:** When selected, also select *UL* as the **HTML Tag Type.**<br><br>• *Related Content:* Use to create drop zones for embedded related content or for dynamic subpages.<br><br>    • To create drop zones for embedded related content, see <u>Working with Embedded Related Content</u>.<br><br>    • To create drop zones for dynamic subpages, see "Inserting Configurable Drop Zones" (Application Designer Developer's Guide).<br><br>• *Menu* and *Menu Item:* Use as a container for a pop-up menu and for each menu item, respectively.<br><br>See <u>Creating Fluid Pop-up Menus</u>.<br><br>• *Tab Set* and *Tab Item:* Use as a container for a set of sectional tabs embedded on a page and for each tab, respectively.<br><br>See <u>Creating Sectional Tabs on a Page</u>. |

| Field or Control | Description |
|---|---|
| | • *Group Box Header Extension:* Use as a container for controls that can be used to extend the information displayed in the header of another group box.<br><br>See <u>Creating Group Box Header Extensions</u>. |
| **HTML Tag Type** | Selecting an HTML tag type enables you to create proper HTML structure as defined by HTML specifications.<br><br>• *Default:* Same as DIV.<br><br>• *DIV:* Set as an HTML 5.0 <div> tag.<br><br>• *Section:* Set as an HTML 5.0 <section> tag. If creating a section containing an article, you may consider using this group box type.<br><br>• *UL:* Set as an HTML 5.0 <ul> (unordered list).<br><br>• *OL:* Set as an HTML 5.0 <ol> (ordered list).<br><br>• *LI:* Set as an HTML 5.0 <li> (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>. |
| **Scroll Bars** | • *None:* Display group box with no scroll bar.<br><br>• *Vertical:* Display the group box with a scroll bar on the right side, running top-to-bottom.<br><br>• *Horizontal:* Display the group box with a scroll bar on the bottom, running left-to-right.<br><br>• *Both:* Display both vertical and horizontal scroll bars.<br><br>• *Carousel:* Display the scroll bar in carousel mode (slider mode), where records can be viewed in defined sets by spinning or swiping. |
| **Group Box Header Extension** | You can specify three different types of group box header extensions by selecting a group box of type *Group Box Header Extension* from the list of available extensions, which are identified by *tab_order_# | Group_Box_Label*.<br><br>• **Textual Info:** Select a group box containing one or more fields to be displayed as simple textual information.<br><br>• **Actionable Info:** Select a group box containing one clickable control that will display additional information.<br><br>• **Action Items:** Select a group box containing one or more controls representing actions associated with the contents of the primary group box (that is, the one with added extensions).<br><br>See <u>Creating Group Box Header Extensions</u>. |
| **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. |

| Field or Control | Description |
| --- | --- |
| **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.<br><br>For example, assume you want to allow dragging 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. |
| **Capture Trigger** | Select for a group box that acts as a container "hotspot" for multiple display-only columns and a hidden trigger field.<br><br>See Implementing Multiple Hotspots for Grid Rows for more information. |
| **Scroll Sizing by Style** | Indicates that the size of the scroll will be controlled by style.<br><br>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 Container Group Boxes

Container group boxes can be used as a standard container, similar to classic 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.

---

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.

Group Box Type: *Accordion Vertical*

Group Box Type: *Accordion Group Box*

Group Box Type: *Accordion Group Box*

Group Box Type: *Accordion Group Box*
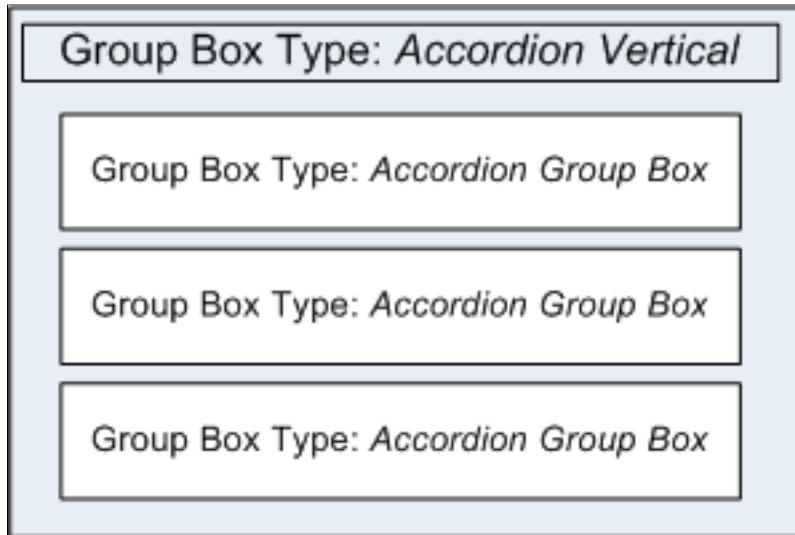
This example illustrates how an accordion group box appears initially, after page load, with all sections collapsed, by default.

| Location Information | ∨ |
|---|---|
| Telephone | ∨ |
| Street Address | ∨ |

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.

| Location Information | ∨ |
|---|---|
| Telephone | ∨ |
| Street Address | ∧ |

| Address Line 1 | 500 Broadway |
|---|---|
| Address Line 2 | |
| City | Death Valley |
| State | CA |
| Country | USA |
| Postal Code | 94564 |
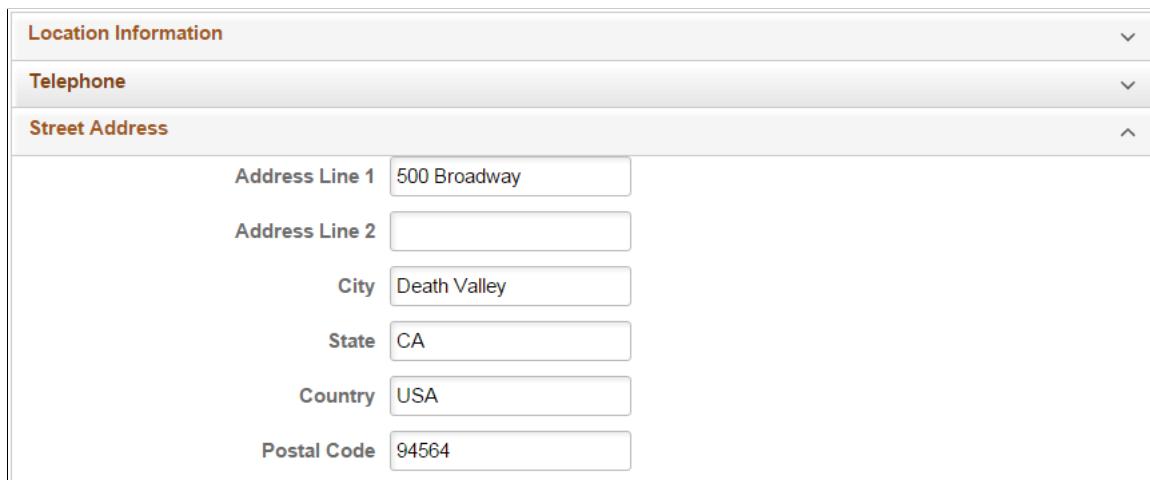
## Working with Layout Group Boxes

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.

## Creating Custom Header Group Boxes

Before creating any sort of custom header (also referred to as an "application header"), 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.

## 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.

## Working with Embedded Related Content

You can display embedded related content on a fluid page using the Related Content group box type. Then, existing tile definitions can be configured as embedded related content on these fluid pages either as an anonymous service or as an explicitly defined related content service using the PeopleSoft Tile URL type.

This example illustrates setting the Group Box Type value to *Related Content,*, which then displays the Autosize check box.
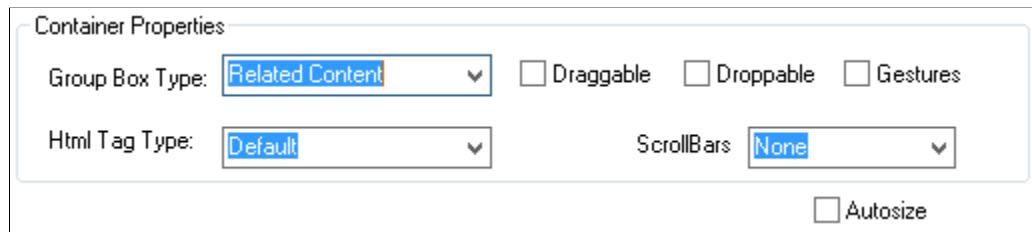


The following procedure provides an example of adding embedded related content to your fluid page. You will complete the first task in Application Designer and then the final task using the Manage Related Content Service component in the browser.

## Adding a Related Content Group Box for Embedded Related Content

To add a related content group box for embedded related content only:

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

2.  Double-click the group box to open the Group Box Properties dialog box. Then, on the Fluid tab, set the Group Box Type to *Related Content*.

3.  For interactive embedded related content, you can select the *Autosize* check box if you always want the content to be fully visible; for non-interactive embedded related content, leave this check box deselected.

4.  (Optional) Apply styling, if needed, to this group box using 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.

    Enter styles in the fields in the Application-Specific Styles area. For example, to use less than the full width of the container or page, enter the percentage of the width. For example: `psc_width-80pct.`

5.  Select the Label tab and enter a descriptive label to identify this related content group box as a drop zone. This label is not rendered at run time, but it will help at design time by:

    •   Differentiating this drop zone from any others on the page.

    •   Providing a label on the drop zone during fluid page preview.

    •   Providing a label for the drop zone during the related content configuration task.

This example illustrates a related content group box added to a fluid page with a label for identification as an embedded related content drop zone.
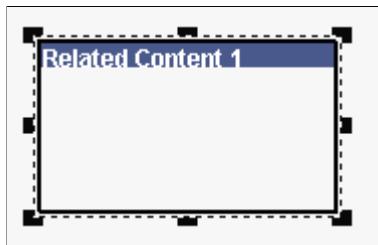


6. Save your changes.

---

**Note:** When a related content group box is configured in this manner, it can be used for embedded related content only. To create a related content group box that can be used for either dynamic subpages or embedded related content, see "Inserting Configurable Drop Zones" (Application Designer Developer's Guide).

---

## Assigning Page-Level Embedded Related Content

1. In the browser, 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 for your fluid page.

3. On the Configure Related Content page, configure the embedded related content in the Page Level Related Content group box as described in "Assigning Related Content to Fluid Components and Pages" (Portal Technology).

   The following example illustrates configuration of embedded related content on a fluid page.
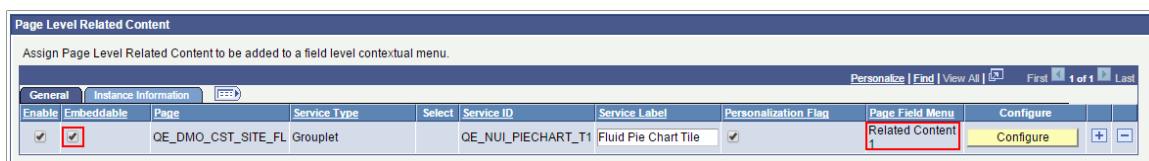


### Related Links

"Fluid Related Content" (Portal Technology)
"Assigning Related Content to Fluid Components and Pages" (Portal Technology)
"Previewing Fluid Pages" (Application Designer Developer's Guide)

# Creating Group Box Header Extensions

You can create group box header extensions to display additional information and actions related to the contents of the primary group box. Three types of header extensions can be defined:

- **Textual Info:** This textual information is placed immediately to the right of, and becomes part of, the group box title. Its purpose is to provide short, text-only information about the group box and its contents.

- **Actionable Info:** This clickable control is placed immediately to the right of the textual information (if that extension has been defined). Its purpose is to provide a clickable button or icon that displays additional information about the group box and its contents in a modal window or a message box, or that transfers the user to another page containing the additional information.

- **Action Items:** One or more action items are placed at the far right of the group box header. Each action item is a clickable button or icon representing an action associated with the group box contents. For example, actions can be, but are not limited to, adding items, deleting items, a pop-up menu, transferring to another page, and so on.

In this example, the primary group box title appears at the left followed by textual information. Each element of the textual information is preceded by a dot. The actionable information icon immediately follows the textual information. Two action items are displayed at the right margin of the group box header.
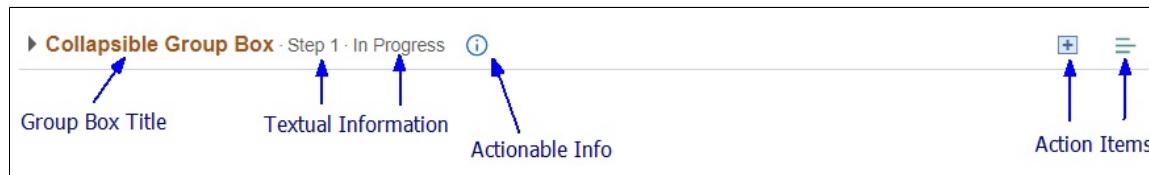


To create and associate header extensions with a primary group box:

1. Create one or more group boxes of type *Group Box Header Extension*.

2. Add one or more controls to each group box depending on the type of extension you are defining. See the following sections for additional information.

3. Create the primary group box if it does not exist already:

   - On the Label tab, specify the label for the group box.

   - On the Fluid tab, select either *Default* or *Container* as the group box type.

   - On the Fluid tab under **Group Box Header Extension**, select one or more extensions, which are identified by `tab_order_# | Group_Box_Label`.

     This example shows selection of group box header extensions.
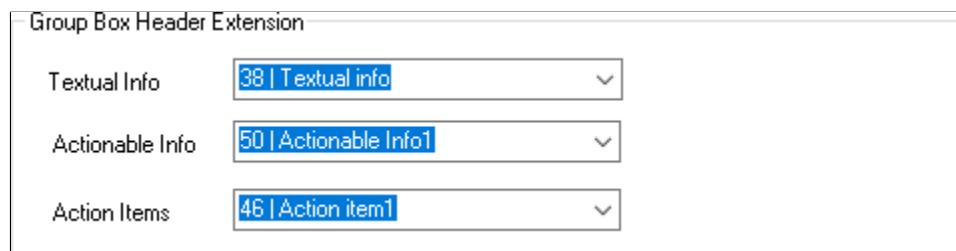


## Textual Info Extensions

Add one or more of these display-only text controls inside the group box:

- Check box – On the Use tab, also select *Text Only* as the display-only appearance.

- Drop-down list box – On the Use tab, also select *Text Only* as the display-only appearance.

- Edit box.

- HTML area – Use inline HTML tags only. No interactive HTML elements can be used. In addition, do not use block-level tags, such as div, p, section, and so on.

- Long edit box – On the Use tab, also select *Expand Field When Display Only*.

- Static text

In addition:

- All controls must be set to *Display Only*.

- If you wish to add a dot separator before each text field displayed in the group box header, add the *psc_dot-before* style to the extension group box itself. Alternatively, you can add the *psc_dot-before* style to individual text controls to add a dot before that control only. For an HTML area, you must add either *ps-text* or *ps_box-value* to the HTML elements inside the HTML area. For example:

```
<span class="ps-text">Simple <i>HTML</i> Area</span>
```

### Actionable Info Extensions

Add one actionable control that defines a clickable button or icon that displays additional information about the group box and its contents in a modal window or a message box, or that transfers the user to another page containing the additional information.

### Action Item Extensions

Add one or more actionable controls that defines an action item as a clickable button or icon representing an action associated with the group box contents. For example, actions can be, but are not limited to, adding items, deleting items, a pop-up menu, transferring to another page, and so on

### Additional Considerations

When implementing group box header extensions, consider this additional information:

- Group box header extensions are not supported for group boxes used as grouplets.

- Group box header extensions are also not supported for group boxes within grids. However, extensions are supported for group boxes within scroll areas.

- Do not assign the same group box header extension to more than one main group box on the page.

## 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.

This example illustrates where you enable group box layout testing on the Component Properties dialog box, by selecting *Layout Only.*
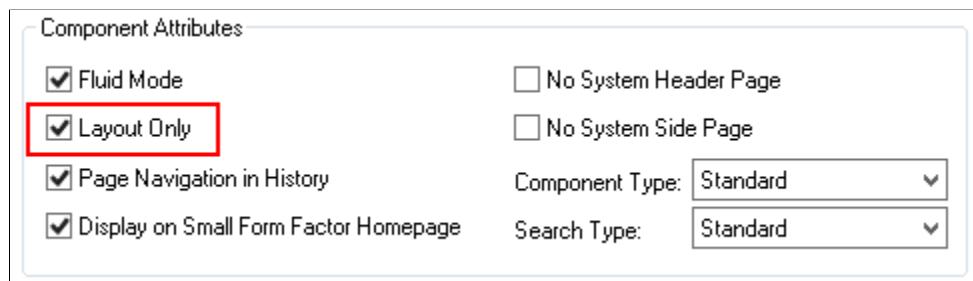


At run time, the system displays the fluid page in the browser using different colored boxes to depict the scope of each group box on the page, providing visual indication for you to verify the scope and contents of each group box container.

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).



---
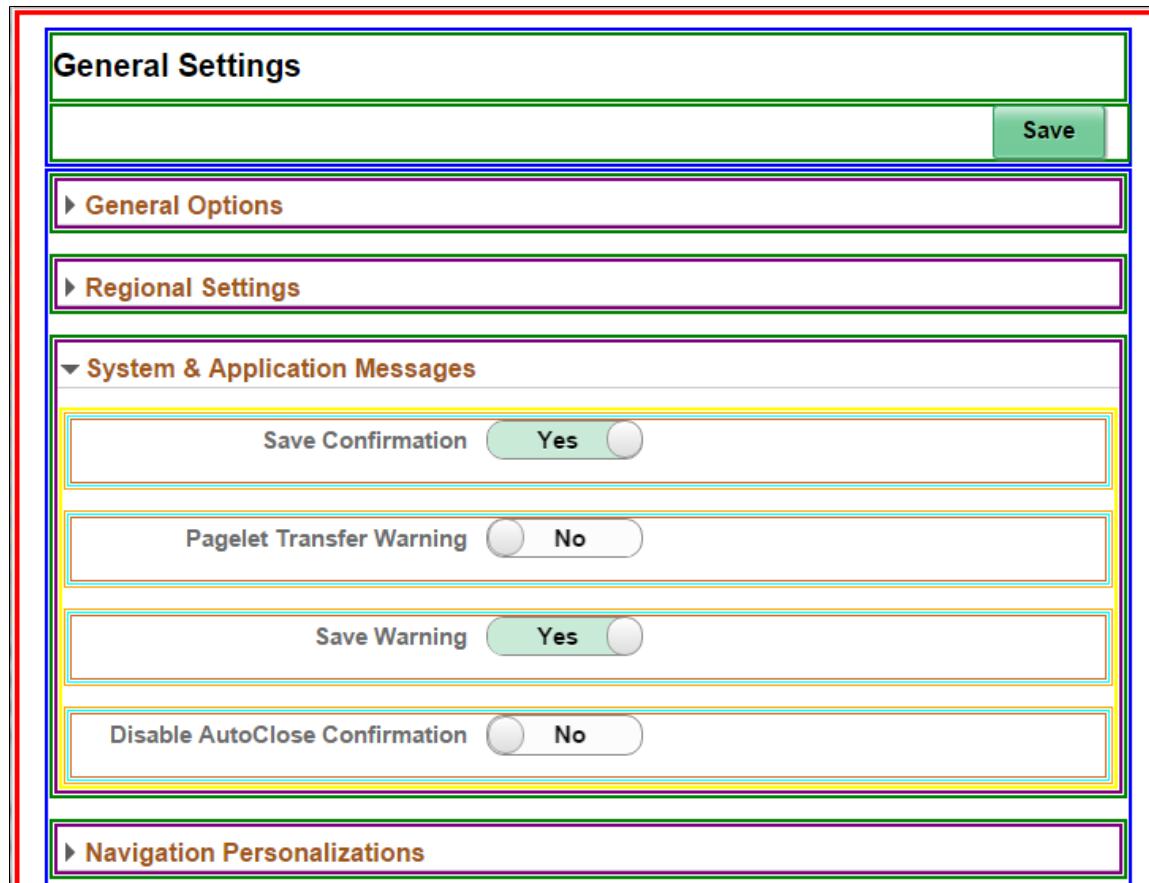
# Working with Other Page Controls

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 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 Label Properties

Fluid grids support these property settings:

- Row count (*n* rows).

- Previous, Next, and View All.

  Combined, these properties enable lazy scrolling on fluid grids; see Enabling Lazy Scrolling for Scrollable Grids for more information. However, unlike classic grids, no separate page controls are displayed for these features.

- Find.

- Download to Excel.

- Personalize, which displays two options for fluid grids:

  - *Personalize Columns* – Select to display the Personalize button in the grid header. The Personalize button allows users to change the order of grid columns, hide columns, and specify the sort order on up to three columns.

  - • *Save Sort* – Select to display the Sort button in the grid header. The Sort button allows users to specify a sort on one column and save this sort personalization.

    ---
    **Note:** Save Sort is selected by default on fluid grids.

    ---

  However, do not enable the Personalize property on a fluid grid if:

- Grid columns are hidden or shown programmatically.

- The Show Column When Cells Hidden check box is selected on the grid's Use tab.

  In addition, do not enable the Personalize property on a fluid grid if the page or component includes multiple grids based on the same record except when you implement a common list of grid columns across all grids using this record *and* you restrict the sort to only those common columns.

Personalization of a "stacked" grid is disabled because it could cause unintended display issues in the grid. The styling that achieves the stacking is imposed by group box columns in the grid. The styling may be different from one collection of grouped columns to the next. If an end user moves a column from one collection of grouped columns to another, it can cause unintended display issues in the grid. Therefore, the Personalize button does not appear on a "stacked" grid.

These features are not supported and are ignored (if enabled) on fluid grids:

- Top and Bottom (First and Last links).

- Row count (*x–y* of *n*).

- Footer.

For more information on setting grid label properties, see "Setting Grid Label Properties" (Application Designer Developer's Guide)

## Selecting Grid Layout

Fluid grids support these layout options:

- Data Grid Layout.

- Div Grid Layout.

- Flex Grid Layout.

- List Grid Layout (Ordered).

- List Grid Layout (Unordered).

- Menu Grid Layout.

- Tab Set Grid Layout.

For more information on selecting a grid layout and setting other grid use properties, see "Setting Grid Use Properties" (Application Designer Developer's Guide).

## Setting Fluid Properties for Grids

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

| Term | Definition |
|------|------------|
| **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:<br><br>• If the custom style does not control the scroll container size successfully, the scroll may not appear or work as desired.<br><br>• Your custom style must be able to handle window size and orientation change.<br><br>• If there is no overflow amount (data rows exceeding occurs limit), lazy scrolling may not be triggered. |
| **Custom Grid Header** | Select a group box of type Custom Grid Header to specify a custom grid header. See Setting Group Box Properties for more information on the Custom Grid Header group box type. |

---

**Note:** In some situations, such as when displaying a page on a smart phone, you may want grid scrolling disabled. Add the psc_noscroll style to Small in the Form Factor Style Class Override group box to disable scrolling for that grid on small form factor devices. Do not disable scrolling for lazy scroll grids.

---

### Other Unsupported Grid Features

These grid features are not supported in addition to the grid label features that aren't supported:

*   Odd/even row styles.

*   Persist in menu.

*   Collapsible data areas.

### Related Links
"Working with Fluid Grids" (Applications User's Guide)

## Enabling Lazy Scrolling for Scrollable Grids

Lazy scrolling provides the ability for the PeopleSoft application to dynamically load grid content as the end user scrolls downward. Rather than loading all content up front, which would degrade performance, the fluid application loads the content as needed. The fluid application loads the next set of scrollable content automatically as the user reaches the end of the currently displayed content. The fluid application displays the total row count for the grid on the same line as the grid title.

To enable lazy scrolling:

1.  On the Grid Properties > General tab, set the Occurs Count to the number rows to be sent to the browser when the grid appears initially.

    ---

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

    ---

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

    *   Previous

    *   Next

    *   View All

3.  On the Use tab, in the Grid Style group box, select the Vertical check box.

## Implementing Multiple Hotspots for Grid Rows

You can add container group boxes to organize grid columns into "hotspots," or clickable areas. Then, you add a hidden trigger field (a hyperlink or push button) to each group box to define the hotspot's action.

To create multiple clickable hotspots in grid rows:

1.  Within the grid, insert a group box before the grid columns to be included within that hotspot.

a. On the Fluid tab of the group box's properties, set the type to *Container* or *Default*.

b. Also, select the **Capture Trigger** check box.

2. Within that group box, insert a push button/hyperlink control:

---

**Note:** The order of this trigger control within the group box does not matter.

---

a. On the Type tab, ensure that the destination is *PeopleCode Command*.

b. On the Use tab, select the **Execute PC on Row/Group Click** check box.

c. On the Fluid tab, stylistically hide the control using *psc_hidden*.

3. Ensure that all visible columns within the hotspot are set to display only and are not interactive.

4. Repeat steps 1 through 3 for each hotspot within the grid.

## Additional Considerations

When implementing multiple hotspots for grid rows, consider this additional information:

- Multiple hotspots and the row click action are mutually exclusive features. You cannot implement both on the same grid.

- The hotspot is rendered in HTML with the ARIA role="button". If link is a more appropriate term for screen reader users, or if further explication is required, use the "AriaAttributes" (PeopleCode API Reference) property of the Field class to describe the ARIA role. For example, either of the following will more accurately describe the control to screen reader users:

```
&fld.AriaAttributes = "aria-roledescription='information button'";
&fld.AriaAttributes = "aria-roledescription='link'";
```

# Working with Scroll Areas

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

| Term | Definition |
|------|------------|
| **Layout** | Select one of these layout options:<br><br>• *Default* – Displays the rows of the scroll in the standard, default format.<br><br>• *List (Unordered)* – Displays the rows of the scroll as a bulleted (unordered) list.<br><br>• *List (Ordered)* – Displays the rows of the scroll as a numbered (ordered) list.<br><br>• *Menu* – Use as a container for a pop-up menu.<br><br>See <u>Creating Fluid Pop-up Menus</u>.<br><br>• *Tab Set* – Use to as a container for a set of sectional tabs embedded on a page.<br><br>See <u>Creating Sectional Tabs on a Page</u>. |

**Note:** Collapsible data areas are not supported for fluid scroll areas.

### Related Links

"Using Scroll Areas and Scroll Bars" (Application Designer Developer's Guide)

## 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:

| Term | Definition |
|------|------------|
| **Application-Specific Styles: Style Classes** | Similar to classic pages, when the character counter is enabled, it is displayed beneath the bottom right corner of the long edit box. If you want the character counter to be displayed beneath the bottom left corner, use the *psc_counter-left* style. Oracle recommends that you do not use a psc_control-width*pct style to control the width of the long edit box itself because you will then need to develop and apply custom styling to get the character counter aligned with the long edit box. |

| Term | Definition |
|---|---|
| **Place Holder Text** | Display default, place holder, text in the edit box as follows:<br><br>• *None:* Disables the display of place holder text.<br><br>• *Static:* Select to enable the Static Text box, and enter the desired text.<br><br>• *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.<br><br>For example, you can add a default value or sample format of the field data to assist users.<br><br>HTML example:<br><br>`<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.<br><br>Setting input types for forms allow for better input control and validation for various data types, including:<br><br>• text<br><br>• date<br><br>• datetime<br><br>• email<br><br>• number<br><br>• range<br><br>• search<br><br>• time<br><br>• url<br><br>**Note:** The Input Properties setting does not apply to long edit boxes. |

**Note:** The Show Label option is not supported for edit boxes on fluid pages.

**Note:** The Display in Modal Window option is not supported for long edit boxes on fluid pages. Therefore, the character limit counter cannot be displayed for long edit boxes in fluid grids.

### 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:

* 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.*

### Related Links
"Using Edit Boxes and Long Edit Boxes" (Application Designer Developer's Guide)

## Working with Subpages

Using subpages can help to reduce clutter on the main or primary fluid page, and they clearly separate related fields into separate entities. In addition, they enable you to segregate business logic into separate, reusable modules.

When using subpages, you will notice that the footprint is small in terms of the space they consume on the page that includes the subpage. Every subpage control must be enclosed by a group box on the primary page where the subpage has been included.

Finally, subpages can be nested—that is, one subpage can include a subpage control for another subpage, and so on.

### Related Links
Creating Other Fluid Page Types

## Working with Secondary Pages

Similar to classic secondary pages, use secondary pages in fluid applications to:

- Reduce clutter of the main or primary fluid page.

- Display ancillary information in a separate modal window.

- Provide a data entry area for secondary or supplemental information.

Unlike classic secondary pages, fluid secondary pages must always be displayed in a modal window. When creating a fluid secondary page:

1. Select *Page (Fluid)* as the definition type in Application Designer.

2. Select *PSL_APPS_CONTENT* as the layout page.

   **Note:** Creating your page in this manner applies the *ps_apps_content* style to the outermost group box container on the page.

3. On the Use tab of the Page Properties dialog box, set the Page Type to *Secondary Page.*

4. On the Use tab, determine whether to retain settings for PeopleTools-produced buttons:

   **Important!** Do not select the Disable Display in Modal Window When Not Launched by DoModal Peoplecode check box for fluid secondary pages. Fluid secondary pages must always be displayed in a modal window.

   - Deselect the OK & Cancel buttons check box if you plan to add custom button definitions directly to the secondary page definition.

     **Note:** On fluid modal secondary windows, the OK button is actually rendered with the label Done.

   - Deselect the Close Box check box when the OK & Cancel buttons check box is selected or when you plan to add a custom Cancel button directly to the secondary page definition.

     This item renders an X in the upper left corner of the modal window that can be used to cancel and close the window.

5. Set an appropriate minimum size for the modal window for each form factor, which is inclusive of its title bar. Specify a size that is neither too small nor too large for the content you are displaying in the modal window. Otherwise, the modal window will be rendered incorrectly, sometimes with scroll bars and sometimes not.

   Oracle recommends that you use styles to set the minimum size for the modal window. Specify styles on the Fluid tab of the Page Properties dialog box.

For example, *page-secmin30x25em* sets the minimum size to 30 ems wide by 25 ems tall:
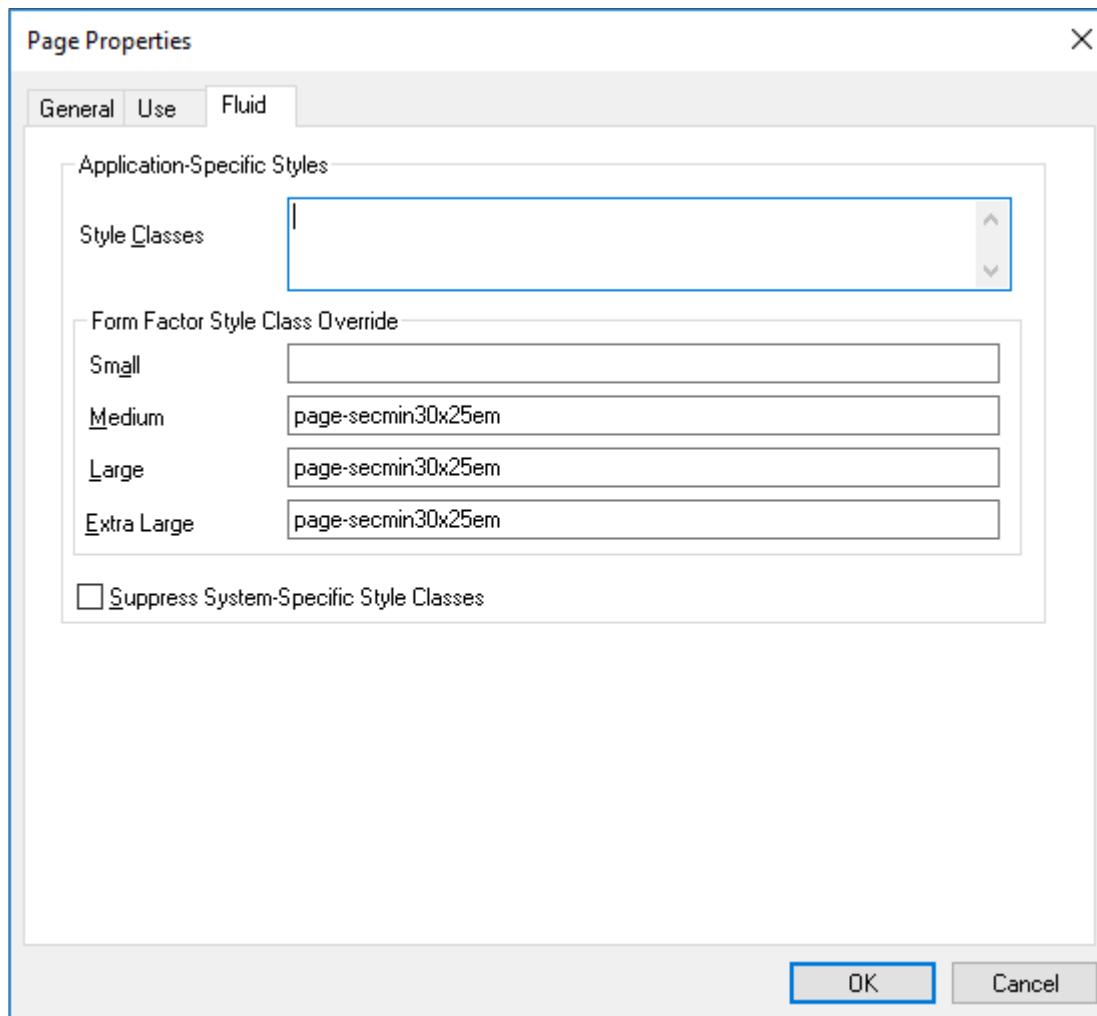


Typically, modal windows on small form factor devices should occupy the full screen. Therefore, no style override was specified in the preceding example.

---

**Note:** You will encounter some Oracle-delivered fluid applications that set width and height in PeopleCode using modal options. Nevertheless, the recommended approach is to use styles as stated here.

---

6. Add page Activate PeopleCode to set the viewport for this secondary page.

Setting the viewport for each fluid component and secondary page ensures that your fluid applications operate properly across all device types, particularly on small form factor devices such as smartphones. The recommended methodology is to use the SetDefaultViewport method:

```
import PT_PAGE_UTILS:*;
(create PT_PAGE_UTILS:Utils()).SetDefaultViewport();
```

## When ps_apps_content Is Not Applied to the Outermost Container

When you create your secondary page definition using the *PSL_APPS_CONTENT* layout page as described in preceding section, *ps_apps_content* is applied to the outermost group box container.

However, in certain cases, *ps_apps_content* may not be appropriate for the outermost container—for example, if your modal page design includes fixed areas (group boxes) at the top, bottom, or sides of the page. If your secondary page does not have *ps_apps_content* applied to the outermost group box container, you might have to apply additional styles to ensure that the modal window is sized correctly:

- If *ps_apps_content* is applied to another group box within the page, for each fixed group box, apply one of these styles:

  - For a fixed horizontal area (for example, at the top or bottom), add the *psc_modal-addheight* style to the container group box to add the height of the group box to the computed modal size.

  - For a fixed vertical area (for example, at either side), add the *psc_modal-addwidth* style to the container group box to add the width of the group box to the computed modal size.

  - Add the *psc_modal-addall* style to the container group box to add both the height and the width of the group box to the computed modal size.

- If you have a scrollable area but are not using the *ps_apps_content* style, then apply the *ps_scroll-target* to this group box. Then, if you have other fixed areas on the page, see the usage notes above for *psc_modal-addheight*, *psc_modal-addwidth*, and *psc_modal-addall*.

### Related Links
Creating Other Fluid Page Types
"SetDefaultViewport" (PeopleCode API Reference)

## 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 reference external JavaScript or external style sheets in the HTML area.

### Related Links
"Using HTML Areas" (Application Designer Developer's Guide)

## 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.

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.



### Related Links

"Using Check Boxes" (Application Designer Developer's Guide)

## Working with Push Buttons and Hyperlinks

In addition to the existing push buttons types used in classic applications, these toolbar action types are available for use in fluid applications only: *Back*, *Clear Prompt*, *Help Link*, *Home*, *IN Prompt*, *Logout*, *Nav Bar*, *New Window*, *Search*, *Search Cancel*, and *SUOW Save*.

You can create tool bars 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.

**Note:** The Open in New Window and Replace Current Window options on the Push Button/Hyperlink Properties dialog box Use tab are not supported for fluid pages. To open a page in a new window for a fluid page, use PeopleCode, such as ViewURL or ViewContentURL.

### Toggle Buttons

A toggle button is a two-state button indicating whether the button is pressed or not pressed (for example, on or off). Pressed (or on) means that the functionality implemented by the button is enabled. Not pressed (or off) means that the functionality implemented by the button is not enabled (or the default functionality is applied).

Toggle buttons can be created from simple text buttons (that is, a text button with the *psc_button-simple* style) or from image-only buttons. The toggle button states are manipulated by these toggle button methods of the PT_PAGE_UTILS:Utils class: "GetButtonState" (PeopleCode API Reference), "SetButtonState" (PeopleCode API Reference), and "ToggleButtonState" (PeopleCode API Reference).

For example, use the following code excerpt in a FieldChange PeopleCode program to toggle the button state each time the button is tapped:

```
import PT_PAGE_UTILS:*;
```

```
Local boolean &bState = (create PT_PAGE_UTILS:Utils()).ToggleButtonState(GetField()⇒
);
/* Do additional processing associated with the toggle button. */
```

### Transparent, Borderless Buttons

You can style push buttons to be transparent and borderless using the *psc_button-transparent* style. This style can be used on standard text buttons, text and image buttons, and image-only buttons. While this style can be used on simple buttons (that is, a button styled with *psc_button-simple*), it is not recommended. Do not use this style on primary buttons or on toggle buttons.

### Related Links
"Using Push Buttons and Links" (Application Designer Developer's Guide)

## 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).

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>
```

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>
```

### Related Links
"Using Static Text" (Application Designer Developer's Guide)
"Using Images" (Application Designer Developer's Guide)

## 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 Application Designer are found in the outer wrapper of the control).

### Related Links
"Using Horizontal Rules" (Application Designer Developer's Guide)

# 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" (PeopleCode API Reference).

# Creating Fluid Pop-up Menus

You can create a pop-up menu on a fluid page with menu items performing any action or navigation. Fluid pop-up menus can contain only a single level of menu items. Using the page controls and techniques described in this section will create pop-up menus that are rendered, styled, and tagged to standards that work with assistive technology.

The following example illustrates a fluid pop-up menu:
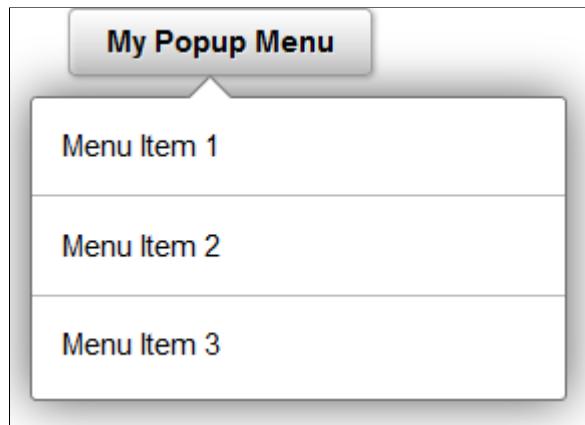


A fluid pop-up menu is not a separate, distinct page control found in Application Designer. Instead, you can use several methodologies to create fluid pop-up menus, including a parent group box (described here), a hyperlink or button, and others. The procedure described here relies on a parent group box of type Popup to serve as the button to activate the pop-up menu. This procedure requires no further coding by you to create a control and menu that is generated to standards that work with assistive technology. In addition to the parent Popup group box, you can use one of three methods to create the pop-up menu:

- A group box of type Menu. Each menu item is a hyperlink contained within a group box of type Menu Item.

- A grid with a layout of Menu Grid Layout. Each row in the grid is a menu item.

- A scroll area with a layout of Menu. Each row in the scroll is a menu item.

To create a fluid pop-up menu:

1. Create a parent group box of type Popup to serve as the button to activate the pop-up menu:

   - On the Label tab, specify a text label for this button.

   - On the Use tab, specify an image for this button if applicable.

- On the Fluid tab: Specify any additional styles for this button—for example, psc_image_only.

- On the Fluid tab: Oracle recommends that you enable a vertical scroll bar for this menu; whether a vertical scroll bar is displayed depends on how the page is displayed (in a modal window, for example) and form factor usage (smart phone, for example).

- Using the SetGroupletMOptions PeopleCode built-in function, set modal popup options enabling bPopupMenu on the record field. For example:

```
MY_RECORD.POPUP_MENU_GBX.SetGroupletMOptions("bPopupMenu@1;");
```

**Note:** When bPopupMenu is enabled (set to 1), the system will automatically set bHeader to 0 (disabled), bAutoClose to 1 (enabled), bMask to 1 (enabled), and sMaskStyle to ps_masktrans; these options cannot be overridden. In addition, bVertical will be set to 1 (which is recommended), but it can be overridden.

2. Within the popup group box, create the menu container using one of three options:

   - To create a static menu, use a group box of type Menu.

   - To create a dynamic menu, use a grid with a layout of Menu Grid Layout.

   - Alternatively, use a scroll area with a layout of Menu to create a dynamic menu.

3. Define menu items within the menu container depending on the option selected in step 2:

   - If the menu container is a group box, create a group box of type Menu Item for each static menu item. Within each menu item group box, insert a single control, a hyperlink that executes the PeopleCode for this menu item. If you require a complex grouping of controls, use a grid or scroll as the menu container.

   - If the menu container is a grid or a scroll area, you can create a simple grid or scroll with one column: a hyperlink that executes the PeopleCode for this menu item.

   - Alternatively, if the menu container is a grid or a scroll area, you can also create a grouping of columns to be acted upon as a single menu item: a layout group box as the first column in the grid (in a scroll, this layout group box contains the other columns), one hyperlink that executes the PeopleCode for this menu item, and any additional controls such as images or counters (badges) to adorn the menu item.

### Related Links
"Navigating Fluid Pop-up Menus Using Keyboard Shortcuts" (Accessibility Guide)

## Creating Sectional Tabs on a Page

You can create sectional tabs embedded on a fluid page to control which portions of the page data are displayed depending on the tab selected. Using the page controls and techniques described in this section will create sectional tabs that are rendered, styled, and tagged to standards that work with assistive technology.

The following example illustrates sectional tabs:

| Tab 1 | **Tab 2** | Tab 3 |
|---|---|---|

DateTime Created

*Description

Sectional tabs are not a separate, distinct page control found in Application Designer. Instead, you can use one of three methods to create sectional tabs:

- A group box of type TabSet. Each tab item is a hyperlink contained within a group box of type TabItem.

- A grid with a layout of Tab Set Grid Layout. Each row in the grid is a tab item.

- A scroll area with a layout of Tab Set. Each row in the scroll is a tab item.

To create sectional tabs:

1. Create the tab set container using one of three options:

   - To create a static set of tabs, use a group box of type TabSet.

   - To create a dynamic set of tabs, use a grid with a layout of Tab Set Grid Layout.

   - Alternatively, use a scroll area with a layout of Tab Set to create a dynamic set of tabs.

2. Define tabs within the tab set container depending on the option selected in step 1:

   - If the tab set container is a group box, create a group box of type TabItem for each static tab. Within each tab item group box, insert a single control, a hyperlink that executes the PeopleCode for this tab. If you require a complex grouping of controls, use a grid or scroll as the tab set container.

   - If the tab set container is a grid or a scroll area, you can create a simple grid or scroll with one column: a hyperlink that executes the PeopleCode for this tab.

   - Alternatively, if the tab set container is a grid or a scroll area, you can also create a grouping of columns to be acted upon as a single tab: a layout group box as the first column in the grid (in a scroll, this layout group box contains the other columns), one hyperlink that executes the PeopleCode for this tab, and any additional controls such as images or counters (badges) to adorn the tab.

### Related Links
"Navigating Fluid Sectional Tabs Using Keyboard Shortcuts" (Accessibility Guide)

## Page Controls Not Supported in Fluid Applications

The following page controls are not supported in fluid applications:

- Embedded help in grids, scroll areas, and group boxes.

- Frames – Use a group box of type Layout instead.

- Pop-up menus – The use of the classic pop-up menu page control can produce unpredictable results with other aspects of fluid applications. Pop-up menus on fluid pages are created in a different manner.

  See Creating Fluid Pop-up Menus.

- Pop-up pages – Mouse-over pop-up pages are not supported in 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 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). |

| Style Sheet | Description |
|---|---|
| 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 the AddStyleSheet built-in PeopleCode function:

```
AddStyleSheet(StyleSheet.STYLESHEET_NAME);
```

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

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.

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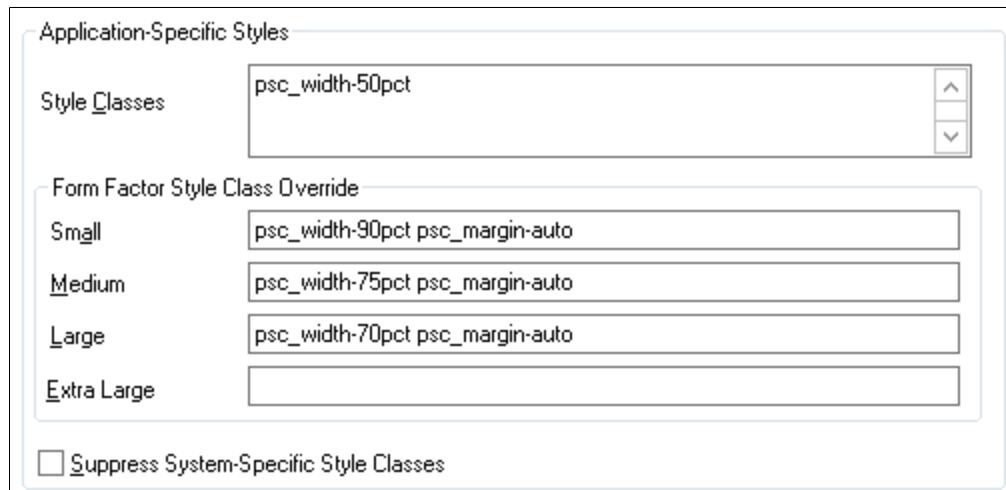


For details on the override options, see Setting Properties on the Fluid Tab.

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 Minimum and Maximum 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 minimum and maximum values for width. When creating custom fluid pages, or customizing delivered pages, it is your responsibility to manage the styling for these minimum and maximum values.

It is recommended that you *do not* set the minimum and maximum 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 minimum and maximum 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 expose the trigger column in screen reader mode.

This is because in screen reader mode, the row action is disabled and the column used as the trigger field now becomes actionable and can receive focus. In screen reader 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 { d⇒

isplay:none; }
``` |

| Consideration | Description |
|---|---|
| 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 minimum and maximum 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 includes an HTML definition, PT_PAGESCRIPT_FMODE, which serves as the master list referencing other definitions containing the actual JavaScript function definitions. These JavaScript functions control a wide range of page behavior, including (but not limited to):

- Scrolling for group boxes.

- Accordion widget.

- Toggle widget.

- Grouplet.

- Drag and drop group boxes.

- Grid lazy scroll.

- Carousel – (used only for scroll areas).

- Transfer animations.

- File attachments.

- Search page and NavBar.

- Related action menu.

- Side page tabs.

- New action buttons – Back, Home, Logout

- Search side page.

Use the AddOnLoadScript PeopleCode built-in function to invoke delivered JavaScript functions. The following example illustrates two invocations of AddOnLoadScript:

```
If (IsGroupletRequest()) Then;
   AddOnLoadScript("DoRowAction(" | &nRow | ");");
End-If;

AddOnLoadScript("SetSideTopTitle('" | EscapeHTML("My M/D Component's Title") | "');⇒

");
```

### Related Links
"AddOnLoadScript " (PeopleCode Language Reference)

# 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:

- All JavaScript shares the same namespace. That includes PeopleTools-delivered JavaScript, PeopleSoft application-delivered JavaScript, and your custom JavaScript. Therefore, to avoid overwriting global variables, functions, and objects defined by PeopleTools or PeopleSoft applications, you must prefix your global variable, function, and object names. For example, rather than naming your function showModal(), which would overwrite the PeopleTools-delivered showModal() function, name your function my_showModal() instead.

- 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 loading, 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" (Accessibility Guide).

| *Accessibility Consideration* | *Description* |
|---|---|
| ARIA live regions | Use Accessible Rich Internet Applications (ARIA) live regions sparingly. |
| 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:<br><br>• Background images<br><br>• Gradients<br><br>• RGBA (transparent or semi-transparent) colors<br><br>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.<br><br>For example, do not include a button labeled "Search" on one page and a button labeled "Find" on another page of the same application. |

| Accessibility Consideration | Description |
|---|---|
| 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:<br><br>• Custom controls need a meaningful programmatically exposed name.<br><br>• Custom controls have appropriate roles specified using the role attribute.<br><br>• 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 employing drag-and-drop capabilities on your page, you must also provide the ability to perform the same functions for users using the keyboard only.<br><br>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:<br><br>• Starts automatically.<br><br>• Lasts more than five seconds. |
| Focus indicator | An end user must be able to see when focus is set for a control, for example a box surrounding the widget:<br><br>• If the widget is in HTML area, provide your own focus indicator.<br><br>• 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 pop-up menus | Use only the techniques described in Creating Fluid Pop-up Menus to create pop-up menus on fluid pages. |
| 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.<br><br>Use `tabindex="0"` for controls that are not accessible using TAB, but should be (those that can receive user input). |
| Links versus buttons | Use links for transferring users to a different location; use buttons for actions on the current page. |

| *Accessibility Consideration* | *Description* |
|---|---|
| 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.<br><br>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:<br><br>`&Fld.JavaScriptEvents = "aria-label=""Partial Days⇒`<br><br>`None""role=""button""";` |
| 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. |
| 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.<br><br>• For selected list items, add the following ARIA attribute:<br><br>`aria-current="true"`<br><br>• 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:<br><br>`:root:not(.psc_mode-access) .GRID1_EDIT,`<br>`:root:not(.psc_mode-access) .GRID1_EDIT {`<br>` display:none !important`<br>`}` |

| Accessibility Consideration | Description |
|---|---|
| Sensory descriptors | Avoid instructions with sensory characteristics or descriptions of page elements, such as:<br><br>• Color (red, blue, and so on).<br><br>• Visual location (above, below, and so on)<br><br>• Orientation (right, left)<br><br>• Size and shape (small, larger, round, arrow, and so on).<br><br>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.<br><br>Pay attention to field focus by tabbing to make sure the tab order is in a logical sequence. Pay close attention to these situations:<br><br>• Default focus location for initial page display.<br><br>   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.<br><br>• Focus location after refreshing part of a page.<br><br>• Focus location before and after pop-up windows are displayed. |
| Testing | Perform the following additional testing:<br><br>• Test your application in Windows high contrast mode.<br><br>• If you build your fluid application for tablets or phones, test using a native screen reader. For example, use VoiceOver (Mac OS X 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:<br><br>• Change the label of the button when on versus off.<br><br>• 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.

## Processing Order for Page Activate Events

The PeopleCode page Activate events for fluid pages are processed in the following order:

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 Fluid 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" (PeopleCode Language Reference) | Add JavaScript to the document or page. |
| "AddMetaTag" (PeopleCode Language Reference) | Add a meta tag to the document or page. |
| "AddOnLoadScript " (PeopleCode Language Reference) | Add the script to run in the onload event of the document. |

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

| *Function* | *Description* |
|---|---|
| "IsAddMode" (PeopleCode Language Reference) | Return a boolean value indicating whether the component build process is in add mode. |
| "IsAGComponent" (PeopleCode Language Reference) | Return a boolean value indicating whether the component type is set to Activity Guide on the Fluid tab. |
| "IsAGRequest" (PeopleCode Language Reference) | Return a boolean value indicating whether the request was initiated by an activity guide wrapper. |
| "IsBackEnabled" (PeopleCode Language Reference) | Return a boolean value indicating whether back navigation is enabled on the component. |
| "IsFluidMode" (PeopleCode Language Reference) | Return a boolean value indicating whether Fluid Mode is enabled on the component. |
| "IsFluidSearchStart" (PeopleCode Language Reference) | Return a boolean value indicating whether the Fluid Search is started on the component. |
| "IsGroupletRequest" (PeopleCode Language Reference) | Return a boolean value indicating whether the request was initiated by a tile (also referred to as a grouplet). |
| "IsHomeEnabled" (PeopleCode Language Reference) | Return a boolean value indicating whether the Home button is enabled on the component. |
| "IsIScriptAuthorized" (PeopleCode Language Reference) | Return a boolean value indicating whether the current user is allowed to access the specified iScript. |
| "IsLogoutEnabled" (PeopleCode Language Reference) | Return a boolean value indicating whether Logout is enabled on the component. |
| "IsMDComponent" (PeopleCode Language Reference) | Return a boolean value indicating whether the component type is set to Master/ Detail on the Fluid tab. |
| "IsMDGuided" (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" (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). |
| "IsMDNonOptimized" (PeopleCode Language Reference) | This function is equivalent to the IsMDListPopup function. |

| Function | Description |
|---|---|
| "IsMDRequest" (PeopleCode Language Reference) | Return a boolean value indicating whether the request was initiated by a component from within the master/detail wrapper. |
| "IsMDSearchEnabled" (PeopleCode Language Reference) | Return a boolean value indicating whether the search type is set to Master/Detail for the current component. |
| "IsModeless" (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" (PeopleCode Language Reference) | Return a boolean value indicating whether the NavBar is enabled on the component. |
| "IsNextInListEnabled" (PeopleCode Language Reference) | Return a boolean value indicating whether NextInList is enabled on the component. |
| "IsNotificationEnabled" (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" (PeopleCode Language Reference) | Return a boolean value indicating whether the Add To functions are enabled on the component. |
| "IsPrevInListEnabled" (PeopleCode Language Reference) | Return a boolean value indicating whether PreviousInList is enabled on the component. |
| "IsSaveEnabled" (PeopleCode Language Reference) | Return a boolean value indicating whether Save is enabled on the component. |
| "IsSearchEnabled" (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. <br><br> Return a boolean value indicating whether search is enabled on the component. |
| "IsSingleComponentAG" (PeopleCode Language Reference) | Return a boolean value indicating whether this is a single component activity guide. |
| "IsSingleUnitOfWork" (PeopleCode Language Reference) | Return a boolean value indicating whether this is a single unit of work activity guide. |
| "IsSmallFFOptimized" (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" (PeopleCode Language Reference) | Return a boolean value indicating whether the search type is set to Standard for the current component. |
| "SetAddMode" (PeopleCode Language Reference) | Set the AddMode attribute in the panel processor. |
| "SetMDGuided" (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" (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" (PeopleCode Language Reference) | Specify a String value that updates the system with styles that control the state of the left and right panels. |
| "SetThemeId" (PeopleCode Language Reference) | Set the theme ID for the page so that proper theme style sheets get loaded. |
| "SetTransferAttributes" (PeopleCode Language Reference) | Set certain fluid-related properties prior to transferring to a different page or component. |

### Related Links
"Understanding Activity Guide Templates and Template Types" (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 menu.

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 menu 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" (PeopleCode API Reference) method is the recommended way to do this.

  ---

### Related Links

"Banner Class" (PeopleCode API Reference)
"PanelController Class" (PeopleCode API Reference)
"Utils Class" (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" (PeopleCode API Reference).

See "AriaAttributes" (PeopleCode API Reference).

See "AriaWrapperAttributes" (PeopleCode API Reference).

See "FreeFormStyleName" (PeopleCode API Reference).

See "GetBaseName" (PeopleCode API Reference).

See "HtmlAttributes" (PeopleCode API Reference).

See "HtmlInputType" (PeopleCode API Reference).

See "IsHyperlink" (PeopleCode API Reference).

See "JavaScriptEvents" (PeopleCode API Reference).

See "PlaceholderText" (PeopleCode API Reference).

See "ReplaceFFClass" (PeopleCode API Reference).

See "SetImageUrl" (PeopleCode API Reference).

## Additional Grid Class Properties

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

See "Layout" (PeopleCode API Reference).

See "FreeFormStyleName" (PeopleCode API Reference).

See "Selector" (PeopleCode API Reference).

## Additional Meta-HTML

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

See "%ExplainMessage" (PeopleCode Language Reference).

See "%FORMFACTOREXTRALARGE" (PeopleCode Language Reference).

See "%FORMFACTORLARGE" (PeopleCode Language Reference).

See "%FORMFACTORMEDIUM" (PeopleCode Language Reference).

See "%FORMFACTORSMALL" (PeopleCode Language Reference).

See "%BB" (PeopleCode Language Reference).

See "%BP" (PeopleCode Language Reference).

See "%BV" (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 "FormFactor" (PeopleCode API Reference).

See "GetDeviceInfo" (PeopleCode API Reference).

See "BrowserDeviceType" (PeopleCode API Reference).

See "BrowserDeviceFormFactor" (PeopleCode API Reference).

See "BrowserFluidCapable" (PeopleCode API Reference).

See "BrowserTypeClass" (PeopleCode API Reference).

See "BrowserPlatformClass" (PeopleCode API Reference).

See "ExtraLarge" (PeopleCode API Reference).

See "Large" (PeopleCode API Reference).

See "Medium" (PeopleCode API Reference).

See "Small" (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" (PeopleCode API Reference).

See "GetDNDRow" (PeopleCode API Reference).

See "GetDNDRowset" (PeopleCode API Reference).

See "GetDNDTargetField" (PeopleCode API Reference).

See "GetDNDTargetRow" (PeopleCode API Reference).

See "GetDNDTargetRowset" (PeopleCode API Reference).

See "Draggable" (PeopleCode API Reference).

See "Droppable" (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" (PeopleCode API Reference)

"SetGroupletDisplayIn" (PeopleCode API Reference)

"SetGroupletEventMsg" (PeopleCode API Reference)

"SetGroupletID" (PeopleCode API Reference)

"SetGroupletImage" (PeopleCode API Reference)

"SetGroupletImageUrl" (PeopleCode API Reference)

"SetGroupletMOptions" (PeopleCode API Reference)

"SetGroupletTargetID" (PeopleCode API Reference)

"SetGroupletTimer" (PeopleCode API Reference)

"SetGroupletType" (PeopleCode API Reference)

"SetGroupletUrl" (PeopleCode API Reference)

"IsGroupletRequest" (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.

| Function | Description | Source Location | | | | | Action | | | | Target | | | Rel |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | FS | CS | AG/MD | IG | MW | F2F | F2C | C2C | C2F | CW | NW | 2W | Avail |
| "DoModal" (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" (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" (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" (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 |

| Function | Description | Source Location | | | | | Action | | | | Target | | | Rel |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FS | CS | AG/MD | IG | MW | F2F | F2C | C2C | C2F | CW | NW | 2W | Avail |
| "DoModalX" (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" (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" (PeopleCode API Reference) | Use the RedirectURL function in to navigate to the location specified by the URL parameter. | N | Y | N | N | N | N | N | Y | Y | Y | N | N | pre-8.54 |
| "Transfer" (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[1,3] | N | pre-8.54 |
| "TransferExact" (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[1,3] | N | pre-8.54 |
| "TransferExactTop" (PeopleCode Language Reference) | Use the TransferExactTop function to exit the current context and transfer the user to another page using an exact match of all keys in the optional key list and by always replacing the current window . | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | N | N | 8.59.01 |
| "TransferModeless" (PeopleCode Language Reference) | Use the TransferModeless function to open a new page in a modeless window on top of the parent ... | N | Y | * | * | N | N | N | Y | N | N | N | Y | pre-8.54 |

| Function | Description | Source Location | | | | | Action | | | | Target | | | Rel |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FS | CS | AG/MD | IG | MW | F2F | F2C | C2C | C2F | CW | NW | 2W | Avail |
| "TransferTop" (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 |
| "ViewContentURL" (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" (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" (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 | Y | Y | Y | N | N | 8.55.14 |
| "ViewContentURLModeless" (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" (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 |

| Function | Description | Source Location | | | | | Action | | | | Target | | | Rel |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FS | CS | AG/MD | IG | MW | F2F | F2C | C2C | C2F | CW | NW | 2W | Avail |
| "ViewURLModeless" (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 |
| "ViewURLTop" (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* | |
|---|---|
| FS | Fluid standalone |
| CS | Classic standalone |
| AG/MD | Activity guide (including 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 (the default), the action exits the fluid wrapper and the content replaces the current window. If the transfer function is set to open a new window, the action opens a new window instead.<br><br>When AJAX transfers are enabled, the new content is displayed in the target content area of the fluid wrapper when the target is set to current window. If the transfer function is set to open a new window, this new window setting is overridden and the new content is also displayed in the target content area of the fluid wrapper *except* when the transfer is from classic content to other classic content. In that case, the new window setting is honored and the classic content is opened in a new window. |
| 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. |

### Related Links

[AJAX Transfers](#)

# 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. <br><br> **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. <br><br> **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. <br><br> **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. |

| Option | Values | Default | Description |
|--------|--------|---------|-------------|
| bHeader | 0 \| 1 | 1 | Display the header for the modal window.<br><br>**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.<br><br>**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. |
| bPopupMenu | 0 \| 1 | 0 | Display a modal window containing a pop-up menu adjacent to the trigger field with a chat dialog-style arrow pointing to the trigger field. For example:<br><br>`MY_RECORD.POPUP_MENU_GBX.SetGroupletMOptions("⇒`<br><br>`bPopupMenu@1;");`<br><br>**Note:** When bPopupMenu is enabled (set to 1), the system will automatically set bHeader to 0 (disabled), bAutoClose to 1 (enabled), bMask to 1 (enabled), and sMaskStyle to ps_masktrans; these options cannot be overridden. In addition, bVertical will be set to 1 (which is recommended), but it can be overridden.<br><br>See Creating Fluid Pop-up Menus. |
| bVertical | 0 \| 1 | 0 | Specify the orientation of the modal window's tail, which points back to the trigger control: a vertical orientation (1) means that the tail can point up or down; horizontal (0) means that the tail can point left or right.<br><br>**Note:** If bVertical is set in conjunction with bPopupMenu to override the default vertical orientation set by bPopupMenu, then bVertical *must* follow bPopupMenu in the invocation. For example:<br><br>`MY_RECORD.POPUP_MENU_GBX.SetGroupletMOptions("⇒`<br><br>`bPopupMenu@1;bVertical@0;");` |
| bSkipFocus | 0 \| 1 | 0 | When you enable bSkipFocus, the focus is set on the close button in the header if modal is a modeless window or if modal has a close icon in the header. |

| 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⇒`<br><br>`, "Message Not Found: My Title"));`<br>`&sOptions = "sTitle@" | &sMenuTitle | ";");` |
| sWidth<br><br>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:<br><br>`sWidth@50em;sHeight@100em;`<br><br>**Note:** The recommended approach for sizing a modal window is to apply styles at the page level. |
| width<br><br>height | Number | none | Set the height and width of the modal window in pixels. For example:<br><br>`width@400;height@400;`<br><br>**Note:** The recommended approach for sizing a modal window is to apply styles at the page level. |

## Related Links

"DoModalComponentPopup" (PeopleCode Language Reference)

"DoModalPopup" (PeopleCode Language Reference)

"SetGroupletMOptions" (PeopleCode API Reference)

"ViewContentURLModeless" (PeopleCode Language Reference)

"ViewURLModeless" (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 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 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" (Applications User's Guide).

Application developers can override the default Back button behavior (for example, 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" (Application Designer Developer's Guide)
"SetTransferAttributes" (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" (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.)

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

| Start Page | | |
|---|---|---|
| Page: | WEBLIB_PTBR.ISCRIPT1.FieldFormula.IScript_StartPage | Override ? |
| Fluid Homepage: | NUI_FRAMEWORK.PT_LANDINGPAGE.GBL | Override ? |

| Field or Control | Description |
|---|---|
| **Fluid Homepage** | Displays the name of the component used to render fluid homepage tabs.<br><br>The default value is *NUI_FRAMEWORK.PT_LANDINGPAGE.GBL.* |

| Field or Control | Description |
|---|---|
| **Override** | Click to change the default fluid homepage. The Override Fluid Homepage Component page is displayed with these fields:<br><br>• **Menu Name:** The default value is *NUI_FRAMEWORK*.<br><br>• **Component:** The default value is *PT_LANDINGPAGE*.<br><br>• **Market:** The default value is *GBL*.<br><br>• **Fluid Homepage:** This field is blank by default.<br><br>---<br>**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" (Portal Technology) for more information on defining the default fluid homepage for a portal and by user attribute.<br><br>---<br>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 Look and Feel" (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.

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

**Fluid Mode Properties**

| | | |
|---|---|---|
| Default Style Sheet Name | | |
| Embedded Stub Subpage | PT_ERCSUBPAGE_STUB | |
| Header Page Name | | Footer Page Name |
| Side Page Name | | Pivot Grid Search Page Name — PTS_NUI_SEARCH |
| Prompt Page Name | | |

| Field or Control | Description |
|---|---|
| **Default Style Sheet Name** | Displays the default, system-wide style sheet for fluid components, which is defined on the Branding System Options page.<br><br>**Note:** If no style sheet is specified, then PSSTYLEDEF _FMODE is used as the default style sheet for all fluid components.<br><br>See "Configuring Branding System Options" (Portal Technology) |
| **Embedded Stub Subpage** | The default subpage stub is PT_ERCSUBPAGE_STUB.<br><br>This subpage stub must be included in a drop zone group box in order to create configurable drop zones. See "Inserting Configurable Drop Zones" (Application Designer Developer's Guide). |
| **Header Page Name** | The default header page is PT_HEADERPAGE.<br><br>If you want to override this page, enter your custom page. |
| **Side Page Name** | The default side page is PT_SIDEPAGE.<br><br>If you want to override this page, enter your custom page. |
| **Prompt Page Name** | The default prompt page is PT_PROMPTPAGE.<br><br>If you want to override this page, enter your custom page. |
| **Footer Page Name** | The default footer page is PT_FOOTER_SAVEONLY.<br><br>If you want to override this page, enter your custom page. |
| **Pivot Grid Search Page Name** | The default search page is PTS_NUI_SEARCH.<br><br>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" (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.

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** > **Personalization Options** > **select PPTL**.

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

## Define Personalizations

Option Category Level:   PeopleTools

Define Personalizations                              Personalize | Find | View All | 🖻 | 🆔      First ◀ 26-50 of 65 ▶ Last

| User Option | Field Format | Format Length | Record (Table) Name | Field Name | Option Default Value | Set Option Default Value | | |
|---|---|---|---|---|---|---|---|---|
| CUSTOMPGSET | ▼ | | PSXLATITEM 🔍 | PSYESNO 🔍 | Yes | Set Option Default Value | + | − |
| DCSP | Mixedcase ▼ | 1 | | | . | Set Option Default Value | + | − |
| DFRMT | ▼ | | PSXLATITEM 🔍 | PT_DATE_FORMAT 🔍 | MMDDYY | Set Option Default Value | + | − |
| DICTIONARY | Uppercase ▼ | 1 | | | N | Set Option Default Value | + | − |
| DTSP | Mixedcase ▼ | 1 | | | / | Set Option Default Value | + | − |
| EXPERT | ▼ | | PSXLATITEM 🔍 | PSYESNO 🔍 | Yes | Set Option Default Value | + | − |
| GRDRWS | Numbers O ▼ | 6 | | | 100 | Set Option Default Value | + | − |
| GRDTAB | ▼ | | PSXLATITEM 🔍 | PSYESNO 🔍 | No | Set Option Default Value | + | − |
| HDRICN | ▼ | | PSXLATITEM 🔍 | PSYESNO 🔍 | No | Set Option Default Value | + | − |
| HPPC | ▼ | | PSXLATITEM 🔍 | PTHOMEPAGE 🔍 | Fluid | Set Option Default Value | + | − |
| HPTABLET | ▼ | | PSXLATITEM 🔍 | PTHOMEPAGE 🔍 | Fluid | Set Option Default Value | + | − |
| KEYWORDSRCH | ▼ | | PSXLATITEM 🔍 | PSYESNO 🔍 | Yes | Set Option Default Value | + | − |

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

These personalization settings are applied to the user's session based on the PeopleSoft system's determination of the browserDeviceTypeClass as defined in the web server's browscap file. (See "Browser Capabilities" (System and Server Administration) for more information on the location and contents of the browscap file.) Three device type classes are defined in the browscap file:

| browserDeviceTypeClass | Homepage personalization option |
| --- | --- |
| pc | HPPC |
| tablet | HPTABLET |
| phone | None – Fluid is always the default homepage type. |

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



**Related Links**

"Understanding Navigation Personalizations" (Security Administration)

# 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" (Portal Technology).

**Related Links**

"Managing Homepages" (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.

**Chapter 8**

# Working with the Push Notification Framework

## Understanding the Push Notification Framework

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

- Tuxedo event broker on the application server.

- WebSocket push technology on the web server.

# Push Notification Framework Architecture

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

*   Server layer.

*   Web server layer.

*   Browser layer



Notifications are generated on the server domains, pushed to the web server, and then on to client browsers.

## Server Layer

The server layer of the Push Notification Framework includes these elements:

*   One or more application server domains.

*   (Optional) One or more Process Scheduler server domains.

- (Optional) Domains gateways for inter-domain communications between the Process Scheduler server and application server domains.

  If more than one application server domain is configured per Process Scheduler server domain, then the additional application server can serve as a fail-over server in the event that initial application server becomes unavailable. In the preceding diagram, this fail-over connection is depicted with a dashed line.

- Event publishers.

- Tuxedo event broker on the application server to route notifications to Jolt on the web server.

### Web Server Layer

The web server:

- Receives browser endpoint subscriptions via WebSocket.

- Receives notifications routed by the Tuxedo event broker to Jolt.

- Routes notifications to subscribing browser endpoints via WebSocket.

### Client Layer

Notifications are routed to the clients for display to users:

- Browser endpoints (the Notification window, pages, and others) are subscribed to events (or event collections).

- Browser endpoints receive notifications from the web server via WebSocket.

## Administration and Development Task Overview

To configure the system for push notifications, the system administrator:

- Configures one or more application server domains.

- Optionally, configures one or more Process Scheduler domains.

- Optionally, configures a cluster environment for push notifications.

Once the system is configured, the system administrator also:

- Specifies global settings.

- Creates and maintains notification configurations.

- Manages workflow notifications.

- Archives and purges notification data.

- Reviews the user data mapping configuration.

- Configures inbound message handlers.

To create notifications, the application developer:

- Creates event definitions.

- Creates event subscriptions.

- Publishes events using PeopleCode.

# Configuring the System for Push Notifications

This topic contains an overview and discusses:

- Configuring an application server domain for push notifications.

- (Optional) Configuring the Process Scheduler domain for push notifications.

- (Optional) Completing inter-domain configuration.

- (Optional) Configuring Process Scheduler system settings for push notifications.

- (Optional) Adding a PeopleTools option to disable and re-enable push notifications.

- (Optional) Configuring push notifications in a cluster.

- Troubleshooting the notifications configuration.

## 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 the application server domain and optionally from the Process Scheduler domain. By enabling push notifications within a domain, the Tuxedo Event Broker will be started in the domain, which is the TMUSREVT server process.

As of PeopleTools 8.58, you do not need to explicitly configure the web server for push notifications. The EnablePNSubscriptions custom web profile property is no longer used, and is ignored regardless of its setting.

**Note:** Events raised in a Process Scheduler domain are ultimately routed through an application server domain. For these inter-domain events, you have to set inter-domain 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" (System and Server Administration).

## Configuring an Application Server Domain for Push Notifications

Each application server domain that will trigger and publish events must be configured for push notifications using psadmin.

To enable push notifications in an application server domain:

1. Run psadmin.exe.

2. Select the application server domain.

3. Select **Configure this domain.**

4. Enter *13* (Push Notifications) to enable push notifications from the Quick-configure menu.

    The push notifications option is set to No (disabled) by default.

5. Optionally, if the Process Scheduler domain will also trigger and publish events, then inter-domain configuration between the server domains is also required. Enter *12* (Domains Gateway) on the quick-configure menu.

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

7. Boot (or reboot) the domain and verify that the TMUSREVT server process starts along with the other domain server processes. In addition, if the domains gateway was enabled, verify that the GWTDOMAIN process is also started.

## (Optional) Configuring the Process Scheduler Domain for Push Notifications

Each Process Scheduler server domain that will trigger and publish events must also be configured for push notifications using psadmin.

To enable push notifications in a Process Scheduler domain:

1. Run psadmin.exe.

2. Select the Process Scheduler domain.

3. Select **Configure this domain.**

4. Enter *4* (Domains Gateway) on the quick-configure menu.

5. Enter *5* (Push Notifications) to enable push notifications from the Quick-configure menu.

6. Enter *6* to load the modified configurations settings.

7. Boot (or reboot) the domain and verify that the TMUSREVT and GWTDOMAIN server processes start along with the other domain server processes.

**Note:** After the Process Scheduler server is configured with a Domains Gateway, additional configuration of both servers is required to complete inter-domain configuration.

## (Optional) Completing Inter-Domain Configuration

To complete the inter-domain configuration, you need to copy the Process Scheduler access point ID to each application server's configuration. Then, you need to copy the application server access point ID to the Process Scheduler's configuration. By default on PeopleSoft systems, the Process Scheduler listens on port 8988 for inter-domain connections; and the application server listens on port 7988. After rebooting both domains, check the Tuxedo log files to ensure that the inter-domain connection has been established.

An application server domain can be configured to receive push notification events from one or more Process Scheduler domains simultaneously. While a Process Scheduler domain can be configured with

more than one application server domain, the Process Scheduler domain connects to only one application server domain the when the Process Scheduler domain is booted. The other configured application server domains remain available as fail-over servers. However, the configuration does not specify a primary server and fail-over servers explicitly. When the Process Scheduler domain is booted, a connection is established by Tuxedo to one of the available servers.

As a first step as described in previous tasks, you must configure the application server and Process Scheduler domains to generate the inter-domain events configuration (interdom.gbb files for each domain).

This example illustrates how the configuration settings for inter-domain events can be coped from a Process Scheduler domain to an application server domain and vice versa. The access point IDs and port values must match on both domains to ensure successful transactions.



## Configuring the Application Server Domain for Inter-Domain Events

**Note:** When your environment consists of multiple application server domains, you can configure any one of the application server domains for inter-domain events.

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

1. Run psadmin.exe.

2. Select the Process Scheduler domain.

3. Select **Configure this domain.**

4. Enter the *15* the Custom configuration option.

5. 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.

6. Review and modify as needed the values in inter-domain events:

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

| *Field or Control* | *Description* |
|---|---|
| **Process Scheduler Credentials** | Specify the credentials for each Process Scheduler domain from which notifications will be received in the format:<br><br>`Process Scheduler Credentials=`*`ACCES⇒`*<br><br>*`SPOINTID`*`\|`*`HOST_NAME`*`:`*`PS_port`*<br><br>• *ACCESS_POINT_ID*: Open the PS_CFG_HOME \appserv\prcs\\*DOMAIN_ID*\interdom.gbb file for the Process Scheduler domain. Locate the section in the file that is similar to the following::<br><br>`*DM_LOCAL`<br>`"PRCS" GWGRP=GWTGROUP1 ACCESSP⇒`<br><br>`OINTID="PRCS_SERVER01_PSNT1234⇒`<br><br>`"`<br><br>---<br>**Note:** The ACCESSPOINTID must be 30 bytes or fewer in length—that is, it must be 30 characters or fewer (including the trailing Null character).<br>---<br><br>• *HOST_NAME*: The host name of the machine that is running Process Scheduler domain.<br><br>• *PS_port*: The port number provided in the interdom.gbb file for the Process Scheduler domain. Each Process Scheduler domain must have a unique port number.<br><br>The completed configuration values for the Process Scheduler domain will look similar to the following example:<br><br>`Process Scheduler Credentials=PRCS_⇒`<br><br>`SERVER01_PSNT1234\|SERVER01:8988`<br><br>---<br>**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_SERVER01_PSNT1234\| SERVER01:8988,PRCS_SERVER01_PSNT5678\| SERVER01:8999`<br>--- |
| **Application Server Port** | You can retain the port number as is. |

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

8.  Boot the application server domain.

## Configuring the Process Scheduler Domain for Inter-Domain Events

To enable inter-domain events for a Process Scheduler domain:

1.  Run psadmin.exe.

2.  Select the Process Scheduler domain.

3.  Select **Configure this domain.**

4.  Enter the *7* the Custom configuration option.

5.  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.

6.  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
    ```

| Field or Control | Description |
|---|---|
| **Process Scheduler Port** | Specify the port through which this Process Scheduler domain sends event notifications. |
| | **Note:** You can leave the port value as is if you are configuring a single Process Scheduler domain. If you are configuring inter-domain events for multiple Process Scheduler domains on the same machine, then specify a unique port number for each domain. |

| Field or Control | Description |
|---|---|
| **Application Server Credentials** | Specify each application server domain to which this Process Scheduler domain can connect in the following format:<br><br>`Application Server Credentials=`*`ACCE`*⇒<br><br>*`SSPOINTID`*`|`*`HOST_NAME`*`:`*`AS_port`*<br><br>• *ACCESS_POINT_ID*: Open the PS_CFG_HOME \appserv\*DOMAIN_ID*\interdom.gbb file for the application server domain. Locate the section in the file that is similar to the following:<br><br>`*DM_LOCAL`<br>`"AS_T85704AA_41162" GWGRP=GWTG`⇒<br><br>`ROUP ACCESSPOINTID="T85704AA_4`⇒<br><br>`1162"`<br><br>---<br>**Note:** The ACCESSPOINTID must be 30 bytes or fewer in length—that is, it must be 30 characters or fewer (including the trailing Null character).<br><br>---<br>• *HOST_NAME*: The host name of the machine, which is running application server domain.<br><br>• *AS_port*: The port number provided in the interdom.gbb file for the application server domain. You can leave the default value of the port as is.<br><br>The completed configuration values for the application server will look similar to the following example:<br><br>`Application Server Credentials=T857`⇒<br><br>`04AA_41162|SERVER01:7988` |

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

8. Boot the Process Scheduler domain.

9. View the TUXLOG files to verify that the inter-domain configuration is valid and that the domains are connected to each other:

   • `PS_CFG_HOME\appserv\`*`DOMAIN_ID`*`\LOGS\TUXLOG.`*`TODAY`*

   • `PS_CFG_HOME\appserv\prcs\`*`DOMAIN_ID`*`\LOGS\TUXLOG.`*`TODAY`*

10. Search for GWTDOMAIN nearest the end of the log file and evaluate the log entry to determine the status of the connection. The following examples are from a Process Scheduler log file:

• *Success:*

```
192615.EDVMR1P0!GWTDOMAIN.4560.2512.0: LIBGWT_CAT:1129: INFO: Connection ⇒

established with domain (domainid=<T1B85701_42446>)
```

• *Failure:*

```
192356.EDVMR1P0!GWTDOMAIN.6700.6724.0: LIBGWT_CAT:1073: ERROR: Unable to ⇒

obtain remote domain id (T1B85701_32872) information from shared memory
192356.EDVMR1P0!GWTDOMAIN.6700.6724.0: LIBGWT_CAT:1509: ERROR: Error occu⇒

rred during security negotiation - closing connection
```

---

**Note:** If you have multiple Process Scheduler domains on the same machine, then this configuration needs to be completed on each of the Process Scheduler domains.

---

### Related Links
"PTPNInterdomainConfig" (Automated Configuration Management)

## (Optional) Adding a PeopleTools Option to Disable and Re-enable Push Notifications

---

**Note:** If you do not create this personalization option, push notifications remain enabled system wide.

---

You can add a PeopleTools personalization option to disable or re-enable push notifications system wide:

1. Select **PeopleTools** > **Personalization** > **Personalization Options.**

2. Search for and select *PPTL.*

3. Add a row to the grid with the following values:

| Field or Control | Description |
|---|---|
| **User Option** | *PNSUBSCR* |
| **Description** | *Push Notifications Enabled* |
| **Option Category Group** | *PS Internet Architecture* |
| **Option Category** | *General Options* |
| **User Option Type** | *System* |
| **Record (Table) Name** | *PSXLATITEM* |

| Field or Control | Description |
|---|---|
| Field Name | *PSYESNO* |
| Option Default Value | Click the Set Option Default Value link and select:<br><br>• *Yes* — Push notifications remain enabled system wide.<br><br>• *No* — Push notifications are disabled system wide until they are re-enabled by this personalization option. |

4. Save your changes.

## (Optional) Configuring Process Scheduler System Settings for Push Notifications

Use the Notification Configuration page to enable or disable notifications for specific Process Scheduler status changes on a system-wide basis.

Navigation:

**PeopleTools** > **Process Scheduler** > **Process Scheduler Settings** > **Notification Configuration**

This example illustrates the fields and controls on the Process Scheduler - Notification Configuration page. You can find definitions for the fields and controls later on this page.



When a status option is enabled *and* inter-domain communication has been configured, the notifications for that status change are propagated from the Process Scheduler server through the application server to the following channels: Notification window, pop-up notifications, and email.

For pop-up notifications, in addition to system-wide settings, each user must enable pop-up notifications as well as specific status options in My Preferences. However, each user does not need to configure the Notification window to receive notifications from the Process Scheduler.

### Related Links

"Defining Notification Configuration" (Process Scheduler)
"Specifying General Settings in My Preferences" (Applications User's Guide)

## (Optional) Configuring Push Notifications in a Cluster

A cluster of PeopleSoft systems consists of:

• A system designated as the portal (or hub).

• One or more content provider systems.

As of PeopleTools 8.56, any PeopleSoft application can be designated the portal system for configuration of unified navigation. PeopleSoft Interaction Hub is no longer required.

To create a cluster configuration:

1. Configure the integration gateway on each system.

2. Configure the integration network on each system.

3. Configure single signon (SSO) on each system.

See the steps outlined in "Configure Unified Navigation Fluid Navigation Collection" (Portal Technology) for information on how to create a cluster configuration.

Then, configure the cluster so that push notifications are propagated from the content providers to the designated portal system:

1. Add the ps_theme custom property to the active web profile on each content provider system to point to the default local node and URL for the portal system. Use the following format to set the value of this property:

   *DEFAULT_LOCAL_NODE*:http://*my_server*.*my_domain*:*port_no*/psp/*SITE_NAME*/

   For example:

   PS_NODE:http://server01.example.com:8000/psp/MYSITE01/

   For more information on setting custom web profile properties, see "Configuring Custom Properties" (Portal Technology),

2. Configure routings for the PTPN_ASYNC_PUBLISH service operation on each system:

   • On each content provider system, outbound routing as local to the portal system.

   • On the portal system, inbound routing as any to local.

   For more information on configuring service operations, see "Adding Routing Definitions" (Integration Broker).

Finally, synchronize notification preferences across the cluster:

1. Configure the PTPN_NOT_PREF_ASYNC service operation on each system:

   • Add the PTPT1000 permission list for service operation security.

   • On the portal system, configure outbound routings to be local (portal system) to each content provider; inbound routing as any to local.

   • On the each content provider system, configure the outbound routing to be local to the portal system; inbound routing as any to local.

2. On each content provider system only, run the PTPN_NOTCONF Application Engine program using the System Process Requests page.

## Troubleshooting the Notifications Configuration

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

- Verify that the web profile passwords are consistent between the web server 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 a 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.

### Testing Push Notifications Published from the 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 just entered appear in the Event Targets fields.

# Administering Notifications

Use the Notification Administration fluid navigation collection to administer notifications on your system.

Navigation:

**PeopleTools** > **Notifications** > **Notification Administration**

The following topics cover each of the components included in the Notification Administration fluid navigation collection including:

- Global Settings.

- Notification Settings.

- TriggerBusiness Event.

- Archive and Purge.

- User Data Mapping.

- Inbound Message Handler.



## Managing Global Settings

Use the Global Settings page to manage system-wide settings for push notifications.

Navigation:

In the left panel of the Notification Administration fluid navigation collection, select the Global Settings item.

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

**Global Settings**

Save

**System**

In-App    Enabled ˅
Email    Enabled ˅
Text    Enabled ˅

**In-App**

Setting for Notify    PTNotify    🔍
Maximum Action Count    150
Maximum Alert Count    150
Action View    ☐    No

When Action View is selected, it allows users to review and act on all actionable notifications within the two-panel action view without opening the Notifications panel

**Text**

Provide message for text consent
Message Set Number    282    🔍
Message Number    92    🔍
Select a Default URL identifier
URL Identifier    PTTEXTMESSAGING 🔍

## System

Use the fields in this section to enable or disable notification delivery options system wide.

| *Field or Control* | *Description* |
| --- | --- |
| **In-App** | Enables in-application delivery of notifications in the Notifications panel. See the "In-App" section below for additional fields that are revealed when this option is enabled.<br><br>**Important!** Even when in-app is disabled, certain notifications will be delivered to users' Notifications panel. For example, in the current release, alerts from the Process Scheduler are not managed via Notification Administration: Notification Settings. |
| **Email** | Enables delivery of notifications via email. |
| **Text** | Enables delivery of notifications via text messages. Additional configuration is required to complete system configuration for text messages. See the "Text" section below for additional fields that are revealed when this option is enabled. |

## In-App

| Field or Control | Description |
| --- | --- |
| **Setting for Notify** | Select PTNotify. It contains information on the notification channels, which you configure on the Notification Settings page.<br><br>See <u>Setting Properties on the Fluid Tab</u> and "Sending Ad Hoc Notifications from Fluid Components" (Applications User's Guide). |
| **Maximum Action Count** | Enter the maximum number of actions to be displayed to the user within the Notifications panel.<br><br>The default value is 150.<br><br>**Note:** Users can personalize the Notifications panel by specifying a different maximum number. See "Personalizing the Notifications Panel and Your Notifications" (Applications User's Guide). |
| **Maximum Alert Count** | Enter the maximum number of alerts to be displayed to the user within the Notifications panel.<br><br>The default value is 150.<br><br>**Note:** Users can personalize the Notifications panel by specifying a different maximum number. See "Personalizing the Notifications Panel and Your Notifications" (Applications User's Guide). |
| **Action View** | Enables the two-panel action view for all users system wide.<br><br>**Important!** Review the following important considerations in "Enabling the Two-Panel Action View" before enabling the two-panel action view on your system. |

## Text

To complete system configuration to support text messaging, each of the following values is required.

| *Field or Control* | *Description* |
|---|---|
| **Message Set Number** | Enter a number for the message set containing the message to be displayed to users when they consent to receive text messages. <hr> **Note:** The default value is 282. |
| **Message Number** | Enter a number for the message to be displayed to users when they consent to receive text messages. This message is displayed on the Notifications page of My Preferences, where users set their notification preferences. See "Personalizing Notifications" (Applications User's Guide). <hr> **Note:** The default value is 92. |
| **URL Identifier** | Select the URL identifier that defines your Twilio account details used for outbound and inbound text messaging. See the "Defining the PTTEXTMESSAGING URL Identifier" section below for instructions on how to configure PTTEXTMESSAGING correctly. <hr> **Note:** Select the default value of PTTEXTMESSAGING or define a custom URL identifier that has the same properties as PTTEXTMESSAGING. |

## (Optional) Defining the PTTEXTMESSAGING URL Identifier

**Important!** If you wish to provide your users with text messaging as a notification option for the delivery of push notifications or the ability to send inbound text messages, then you must establish an account with Twilio.

To define the PTTEXTMESSAGING URL identifier:

1.  Select **PeopleTools** > **Utilities** > **Administration** > **Maintain URLs**.

2.  Search for and select *PTTEXTMESSAGING*.

This example illustrates the fields and controls on the URL Maintenance page and URL Properties page for the PTTEXTMESSAGING URL identifier. You can find definitions for the fields and controls in the following section.



3. In the URLID field, replace the **{AccountSid}** placeholder with your Twilio account ID.

4. Click the URL Properties link and enter the following properties and values:

| Field or Control | Description |
|---|---|
| **ACCESSTOKEN** | The access token for your Twilio account. |
| | **Note:** Use the password encryption utility on this page to encrypt the access token. |
| **ACCOUNTID** | Your Twilio account ID. |
| **PROXY** (as required) | The fully qualified domain name of your proxy server. |
| **PROXYPORT** (as required) | The HTTP port on your proxy server. |
| **TEXTFROM** | The phone number associated with your Twilio account. |
| | Enter the phone number in international phone number format (known as E.164), which is a number of up to fifteen digits in length starting with a +. Do not include any other non-numeric characters such as spaces, dashes, periods, or parenthesis. For example: |
| | `+15555550100` |

## (Optional) Enabling the Two-Panel Action View

The two-panel action view uses the master/detail architecture. A user's list of notifications (the master) is displayed in the left panel, while the actionable transaction (the detail) is displayed on the right. The action view master/detail component has both AJAX transfers and the slide-out left panel enabled.

The following diagram illustrates the relationship of the items in the action view master/detail architecture:

- The action view master/detail component is the container.

- The master list of notifications is displayed in the left panel.

- If the notification target content includes an inner left panel, then the left panel collision handling that is set on the action view master/detail component enables the slide-out left panel.

- The component for the notification target is displayed on the right.



Any component and page that could be displayed as a detail transaction in the action view must conform to the general restrictions on being displayed within a master/detail component as described in Guidelines and Limitations for Master/Detail Components.

In addition to these general guidelines, evaluate each target component and page that could be displayed in the action view as the result of a notification:

- If the target is a fluid activity guide, another master/detail component, or a fluid navigation collection, it cannot be displayed within the action view; however, it can be set to display in a new browser window. You must add the &ICDisplayNotify=N query string parameter to the notification when it's published.

- Pages in the target fluid component cannot include an "application header"—that is, any of the Custom Header group box types (Left, Right, Middle, or Bottom) must be hidden when displayed in the action view.

- Since the action view master/detail component controls the title in the fluid banner, for each page defined within the target component, you should add a group box of type Page Title that is defined to be visible when the page is rendered within a master/detail component. This Page Title group box provides context that is lost when the page title is not displayed in the fluid banner. To hide this page title if the page is rendered outside of a master/detail component, use the psc_hideinmode-notmd style.

- If the target content is a fluid component that employs the 8.54 two-panel construct (that is, it was built from the PeopleTools 8.54 PSL_TWOPANEL layout page or it was designed with an internal left panel similar to PSL_TWOPANEL), your implementation must ensure that the internal two-panel is initialized (using initializeTwoPanel) so that left panel collision handling can occur.

### Related Links

Working with Two-Panel Implementations

"Using the Two-Panel Action View" (Applications User's Guide)

"PTPNSystemConfig" (Automated Configuration Management)

## Managing Notification Configurations

Use the Notification Settings page to define and maintain notification configurations that determine the notification type and personalization options by notification name.

Navigation:

In the left panel of the Notification Administration fluid navigation collection, select the Notification Settings item.

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

**Notification Settings**

Channel Settings for Notifications                                          6 rows

| Notification Name | | Description | Functional Category | In-App | Email | Text |
|---|---|---|---|---|---|---|
| BROADCAST | ⊙ | Broadcast Notifications | Notifications | Yes | No | No |
| PGThresholds | ⊙ | Personalized Analytic Notifications | Pivot Grid | Yes | Yes | Yes |
| PHCPushNotification | ⊙ | Health Center Notification | System Performance | Yes | Yes | Yes |
| PTAI_NOTIFICATION | ⊙ | Activity Guide - Notify | Activity Guide - Notify | Yes | Yes | Yes |
| PTNotify | ⊙ | Notify Options | Notifications | Yes | Yes | Yes |
| TriggerBusinessEvent | ⊙ | Trigger Business Event | Workflow | Yes | No | No |

| Field or Control | Description |
|---|---|
| **Notification Name** | The notification name must match exactly the name defined in the custom PeopleCode application class that implements the delivered PTPN_PUBLISH:PublishToWindow class. This key value is entered on the Add page, and cannot be edited later. For example, the following PeopleCode excerpt defines the BROADCAST notification for the SENDNOTE event:<br><br>`&wlSrch = create PTPN_PUBLISH:PublishToW⇒`<br><br>`indow("SENDNOTE", "BROADCAST");`<br><br>See Publishing Events. |
| **Description** | Specifies a description for this notification configuration. |
| **Functional Category** | Specifies a functional group name for this definition.<br><br>**Note:** At creation or configuration time, the functional category can be entered as free-form text or selected from existing values. |
| **In-App** | Specifies whether notifications for this notification are delivered to the Notifications panel. |
| **Email** | Specifies whether notifications for this notification are delivered by email. |
| **Text** | Specifies whether notifications for this category are delivered by text message.<br><br>**Important!** You must establish an account with Twilio in order to use text messaging as a notification option for the delivery of push notifications. See the section titled "Defining the PTTEXTMESSAGING URL Identifier" in Managing Global Settings. |
| Action | Bulk actions include: Deleting all selected notification configurations. |
| ⊻ | Related actions include:<br><br>• Configure – See "Adding or Updating Notification Configurations" in the following section.<br><br>• Delete – Deletes the current notification configuration.<br><br>• Add to ADS Project – Accesses the Add To Project page to add the current notification configuration to an ADS project. See "Adding Data Set Instances from Object Designer Pages" (Lifecycle Management Guide). |

**Important!** In the current release, alerts from the Process Scheduler cannot be managed via the Notification Settings page. No notification definition is delivered for these events and alerts. Alerts will be delivered to users' Notifications panel even though the system is not configured for in-app notifications.

## Adding or Updating Notification Configurations

Use the notification Add page to create new notification configurations; use the Configure page to update existing notification configurations. The Add and Configure pages have similar fields, except that the Notification Name is not editable on the Configure page.

**Note:** In releases prior to PeopleTools 8.59, the notification name had to match exactly the name defined in the custom PeopleCode application class that implements the delivered PTPN_PUBLISH:PublishToWindow class.

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



In addition to the fields that have already been defined for the Notification Settings page, adding or updating a notification configuration includes these additional fields:

| Field or Control | Description |
|---|---|
| **Object Owner ID** | Select an owner for this definition. |

| Field or Control | Description |
|---|---|
| **Mandatory** | Select the mandatory option to indicate whether the notification type is mandatory, which means that if personalization is allowed, each user must subscribe via at least one notification type.<br><br>**Note:** This setting is ignored if personalization is not enabled. |
| **Available** | Indicates whether the notification type is available for use for this notification type. |
| **Enable by Default** | Indicates that this notification type is on by default for all users. |
| **Allow Personalization** | Specify whether users can personalize this notification. |
| **Role/Permission List** | Enter one or more role names, one or more permission lists, or both to restrict which users can personalize this notification. If there are no entries in this grid, then all users are allowed to personalize this notification. |

## Managing Workflow Notifications

Use the TriggerBusiness Event page to enable or disable *workflow event* notifications selectively based on the business process, activity name, event name, and worklist name.

Navigation:

In the left panel of the Notification Administration fluid navigation collection, select the TriggerBusiness Event item.

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



| Field or Control | Description |
|---|---|
| **Enable** | Select Enable to enable or disable the notifications for this business process combination. |
| **Notification Configuration Override** | For this business process, select a notification configuration to override the default PeopleTools-delivered TriggerBusinessEvent notification configuration. |
| **>** | Click to configure the notification details. See the following section, "Workflow Notification Details Page." |

## Workflow Notification Details Page

Use the Notification Details page to configure the details for the selected workflow event.

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

| Cancel | **Notification Details** | Done |
|---|---|---|
| Priority | Default | |
| Notification Text Message Set Number | 9999 | |
| Notification Text Message Number | 9999 | |
| Notification Category Message Set Number | 9999 | |
| Notification Category Message Number | 9999 | |
| Notification Type | Alert | |
| Email Event | Email Note | |

| *Field or Control* | *Description* |
|---|---|
| **Priority** | Specify the priority for this notification: Default, High, Low, or Medium. |
| | **Note:** Notifications with a high priority are highlighted in the Notifications panel. All other priority settings are available as filters on the View All Notifications page. |
| **Notification Text Message Set/Message Number** | For each workflow event, specify the message set ID and number to form the notification message. |
| **Notification Category Message Set/Message Number** | **Note:** These fields are retained for compatibility with workflow event notifications created prior to PeopleTools 8.59. |
| **Notification Type** | Specify the notification type: Alert or Action. |

| Field or Control | Description |
|---|---|
| **Email Event** | For workflow events that include an email event in the workflow definition, you can select *Email Note* to use that event definition as the notification message. Otherwise, the Message Catalog entry specified on this page is used instead. |

# Archiving and Purging Notification Data

Use the Archive and Purge page to archive and purge notifications system wide. Archiving and purging notification data at regular intervals improves system performance.

Navigation:

In the left panel of the Notification Administration fluid navigation collection, select the Global Settings item.

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



| Field or Control | Description |
|---|---|
| **From Date, To Date** | Set the period to archive or purge data. |

| Field or Control | Description |
|---|---|
| Message State | Select one or more message states to archive or purge:<br><br>• *Dismissed* — Users can dismiss alert notifications so that they no longer appear in the Notification window. However, they are still viewable on the View All Notifications page.<br><br>• *New* — A notification is considered new until the user opens the Notification window, and then the notification becomes unread.<br><br>• *Read* — A notification is considered read if the user clicks on the notification within the Notification window.<br><br>• *Unread* — A notification is considered unread until the user clicks the notification within the Notification window, and then the notification becomes read. |
| Purge Archive Record | Select this option to also purge notifications from the archive table (PSPTPN_ARCHIVE). |
| Notification Name | Select a specific notification for archiving or purging.<br><br>**Note:** If no notification is selected, all notifications will be archived or purged. |
| Archive | Click to archive the data based on the selected options. When notifications are archived, the data is moved from the notification table (PSPTPN_MSG_TBL) to the archive table (PSPTPN_ARCHIVE). |
| Purge | Click to purge the data based on the selected options. When notifications are purged, the data is deleted from the notification table. Optionally, you can also select to purge data from the archive table.<br><br>**Important!** You cannot retrieve purged data. |

## Configuring User Data Mapping

Use the User Data Mapping page to configure an application package, class, and method to supply the email addresses and phone numbers that users select from when personalizing their notification preferences.

Navigation:

In the left panel of the Notification Administration fluid navigation collection, select the User Data Mapping item.

This example illustrates the fields and controls on the User Data Mapping page. You can find definitions for the fields and controls later on this page.

**User Data Mapping**

Save

**Application Class Configuration for User Data**

| | |
|---|---|
| Root Package ID | PTNC_UTILITY |
| Qualified Package/Class Path | : |
| Application Class ID | UserDataMapping |
| App Class Method Name | getUserNotificationOptions |

| Field or Control | Description |
|---|---|
| **Root Package ID** | Select the name of the application package that contains the method to be invoked. |
| **Qualified Package/Class Path** | Select the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon. |
| **Application Class ID** | Select the name of the application class that contains the method to be invoked. |
| **App Class Method Name** | Select the method to be invoked to retrieve and return user email address and phone number data. The method must return the data in a Rowset object based on the PTNC_USER_DATA record definition. |

PeopleTools delivers a default PTNC_UTILITY:UserDataMapping.getUserNotificationOptions method that retrieves email addresses from the user profile and a phone number from My Preferences if one has been defined there. PeopleSoft application pillars will deliver user data mapping that is specific to where users enter contact information in that application pillar.

## Configuring Inbound Message Handlers

Use the Inbound Message Handler page to configure message handlers for inbound text messaging.

**Important!** You must establish an account with Twilio in order to use inbound message handlers. See the section titled "Defining the PTTEXTMESSAGING URL Identifier" in <u>Managing Global Settings</u>.

Navigation:

In the left panel of the Notification Administration fluid navigation collection, select the Inbound Message Handler item.

This example illustrates the fields and controls on the Inbound Message Handler page. You can find definitions for the fields and controls later on this page.

**Inbound Message Handler**

Save

**Inbound Text Message Handler Configuration**

2 rows

| | Application Code ⌄ | Description ⌄ | Root Package ID ⌄ | Qualified Package/Class Path ⌄ | Application Class ID ⌄ | Application Class Method Name ⌄ | | |
|---|---|---|---|---|---|---|---|---|
| 1 | EXP | Expenses | FS_UTILITY 🔍 | : 🔍 | TextUtilities 🔍 | processExpense | ➕ | ➖ |
| 2 | RQT | Procurement | FS_UTILITY 🔍 | : 🔍 | TextUtilities 🔍 | createRequest | ➕ | ➖ |

| Field or Control | Description |
|---|---|
| **Application Code** | Enter a unique three-character code identifying the PeopleSoft application represented by this inbound message handler. |
| **Description** | Enter a description for this application. |
| **Root Package ID** | Select the name of the application package that contains the inbound message handler to be invoked. |
| **Qualified Package/Class Path** | Select the names of each subpackage in the application class hierarchy that define the location of the application class. Separate subpackage names by a colon. If the class is defined in the top-level application package, enter or select the colon. |
| **Application Class ID** | Select the name of the application class that contains the inbound message handler to be invoked. |
| **App Class Method Name** | Select the method to be invoked to process inbound text messages. The handler should process the three-character code and message payload to invoke the appropriate PeopleSoft transaction on behalf of the user associated with the sender's phone number. |

# Creating Push Notification Events

When creating push notification events, you:

1. Define the event.

   An event is associated with a status change on the server. When you define an event, you specify its data type and data structure.

2. (Optional) Add the event to an event collection.

   You can group a set of events that exhibit a common behavior as a collection. When an endpoint is subscribed to a collection, it receives notifications for each event included in the collection whenever they are published from the server.

3.  Publish the event using PeopleCode.

4.  Subscribe endpoints to the event or event collection.

5.  Define the notification configuration for the event.

# Defining Events

Use the Define Server Side Events page to define the data type and data structure for state changes that occur on the server. Examples of state changes include:

*   Process Scheduler process or job status change.

*   Approval notification for a workflow.

*   A custom state change such as adding or updating a row in an application data table.

Navigation:

**PeopleTools** > **Push Notifications** > **Notification Events.**

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



| Field or Control | Description |
|---|---|
| **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 defines how the system structures the event data while publishing the event. The value you assign determines what you enter for a data structure. |

| Field or Control | Description |
|---|---|
| Data Structure | Enter the Data Structure value, depending on the Data Type selected.<br><br>• *KeyValue:* Enter key names separated by commas.<br><br>• *PeopleCode Rowset:* Click the **Prompt** button to select the record name.<br><br>Use delivered PeopleCode application classes to set event data (in the form of key-value pairs or record field data) when publishing the event. |
| Adapters | **Note:** Mobile Application Framework (MAF) has been desupported as of PeopleTools 8.59.03. While the documentation of this feature has been removed at this time, the feature remains available in the PeopleSoft system. MAF will be fully removed in a future PeopleTools release.<br><br>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. |

## Maintaining Event Collections

Use the Define Server Event Collections page to define event collections and add defined events to a collection.

Navigation:

**PeopleTools** > **Push Notifications** > **Notification Event Collections**.

This example illustrates the fields and controls on the Define Server Event Collections Page. You can find definitions for the fields and controls later on this page.



| Field or Control | Description |
|---|---|
| Collection Name | Enter the unique name for the event collection. |
| Description | Enter a description of the event collection. |

| Field or Control | Description |
|---|---|
| Root Package ID | Select the PeopleCode application package that contains the application class that is the handler for this collection. |
| Application Class ID | Select the PeopleCode application class that handles the events that are part of the event collection when they are published. A valid handler class implements the PT_PNCOLL _HANDLER:Handler interface. |
| Events | Click the Events link to maintain the events for the event collection using the Add Events page. If an endpoint is subscribed to the collection, then it will receive the push notifications for each of the events listed in this page. |

## Add Events Page

Use the Add Events page to maintain the events associated with an event collection.

This example illustrates the fields and controls on the Add Events page.



# Publishing Events

You use PeopleCode to publish an event:

- For custom endpoint subscriptions, create an object of the PSEvent class to publish the event.

- For certain delivered endpoint subscriptions, instantiate and use delivered applications classes to publish to that endpoint—for example, to the Notification window.

## Publishing an Event for KeyValue Pairs

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. 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:

| Field or Control | Description |
| --- | --- |
| 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.

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");
```

### Publishing to the Notifications Panel

For example, the Notifications panel endpoint is delivered with a subscription to the PUSHNOTIFICATIONWINDOW event collection as seen previously on the Define Server Event Collections page. Once you have defined and added your custom event to the PUSHNOTIFICATIONWINDOW event collection, you can instantiate and use the PTPN_PUBLISH:PublishToWindow application class to publish your custom event to the Notifications panel.

The following PeopleCode excerpt illustrates the publication of a notification of type alert:

```
import PTPN_PUBLISH:PublishToWindow;

/* Creates and publishes an alert notification to the current user. */
Local string &KEY = "Student_ID" | PSU_STUDENT_TBL.STUDENT_ID.Value;
Local string &MSGINFO1;
&MSGINFO1 = "You added a new student named " | PSU_STUDENT_TBL.STUDENT_NAME.Value |⇒

 " with ID " | PSU_STUDENT_TBL.STUDENT_ID.Value | ".";
Local PTPN_PUBLISH:PublishToWindow &newStdnt_alert = create PTPN_PUBLISH:PublishToW⇒

indow("NEW_STUDENT", "Student Data");
&newStdnt_alert.SetCategoryTypeFyi(); /* Type for alerts. */
&newStdnt_alert.AddRecipient(%UserId, 1);
&newStdnt_alert.SetMsgKey(&KEY);
&newStdnt_alert.SetMsgInfo(&MSGINFO1);
&newStdnt_alert.SetMsgStateNew();
&newStdnt_alert.SetPriority("3"); /* Low priority */
&newStdnt_alert.Publish("");
```

# Subscribing to Events

You can use one of several methodologies to subscribe an endpoint to an event or event collection:

- Using JavaScript, the event data is published using the JavaScript Object Notation (JSON) format:

  - Subscribe to a specific event.

  - Subscribe to an event collection.

- Using PeopleCode.

  For custom endpoint subscriptions, create an object of the PSEvent class to subscribe to the event.

- Using Interwindow Communication (IWC).

## Subscribing to an Event Using JavaScript

Use the Subscribe JavaScript function to subscribe an endpoint to an event:

```
Subscribe(EVENT_NAME, Callback_Function_Name)
```

| Field or Control | Description |
|---|---|
| EVENT_NAME | The name of the event that the endpoint is being subscribed to. |
| Callback_Function_Name | JavaScript function invoked when the event is published on the server. |

For example:

```
Subscribe("PRCSEVENTS", ProcessHandler);
```

The ProcessHandler JavaScript function is then defined in the code:

```
function ProcessHandler(EventName, EventData)
{
    if(EventName == "PRCSEVENTS")
// Actual code omitted here ...
}
```

## Subscribing to an Event Collection Using JavaScript

Use the SubscribeCollection JavaScript function to subscribe an endpoint to an event collection:

```
SubscribeCollection(COLLECTION_NAME, Callback_Function_Name)
```

| Field or Control | Description |
|---|---|
| COLLECTION_NAME | The name of the event collection that the endpoint is being subscribed to. |
| Callback_Function_Name | JavaScript function invoked when the event is published on the server. |

For example, the Notification window has been subscribed to the PUSHNOTIFICATIONWINDOW event collection using this JavaScript code:

```
SubscribeCollection("PUSHNOTIFICATIONWINDOW",UpdatePNUI);
```

The UpdatePNUI JavaScript function is defined as follows:

```
function UpdatePNUI(EventName, EventData) {
    try {
        var parentNode = document.getElementById('PT_NOTIFY');
         var parentNode_classic=top.document.getElementById('pthdr2notify_div');
        if (parentNode != null  || parentNode_classic!=null) {
            NewNotification(EventData);
        }
    } catch (e) {}
```

```
}
```

## Subscribing to an Event Using PeopleCode

For custom endpoint subscriptions, create an object of the PSEvent class to subscribe to the event. Use the Subscribe method, which requires an application class method be defined as the callback:

```
APPLICATION_PACKAGE:ApplicationClass:MethodName
```

For example, the following code creates a subscription for the PRCSEVENTS event:

```
Local object &EventObject = CreateObject("PSEvent");
Local boolean &SubscriptionHandle = &EventObject.Subscribe("PRCSEVENTS", "PT_CONQRS⇒

:CQTEMPTABLES:OnEvent");
```

Then, implement an application class that defines the callback method for the event subscription:

```
method OnEvent
   /+ &EventObject as Object +/
   Local object &EventNotifObject = CreateObject("PSEvent");
   /* Actual code omitted here */
end-method;
```

## Subscribing to an Event Using Interwindow Communication (IWC)

Use IWC to subscribe a PeopleSoft page to a published event, 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.

To subscribe to an event through IWC:

1. In the browser, select **PeopleTools** > **Portal** > **Structure and Content**.

2. Navigate through the portal registry folders and select the content reference definition for the component.

3. Click the **IWC Message Events** link on the Content Ref Administration page.

4. Select the Add a New Value tab, and click Add.

5. Add a new row on the IWC Message Events page, specifying these values:

   • Event Name: Enter the name of a defined event.

   • Message Event Type: Select *Server Sub*.

6. Click the **Map** to display the Map Events Data to Page Elements page.

On the Map Events Data To Page Elements page you can map the data of a defined event to a page element.



| Field or Control | Description |
|---|---|
| **Event Name** | Displays the event name that was entered on the Content Reference Message Events page. |
| **Page Name** | Select the page where you want the event subscription added. |
| **Page widget Type** | Select either *Fields* or *Grid*. |
| **Event Data** | Select one of the data elements defined for 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 page record field name for the page element to be updated. |

# Sending Broadcast Messages

Use the Broadcast Message page as a testing utility to send broadcast messages to selected users by role or user ID.

Navigation:

**PeopleTools** > **Utilities** > **Debug** > **Send Broadcast Messages**

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



| Field or Control | Description |
| --- | --- |
| **Type of User** | Select from these types:<br><br>• *Role Name*: Sends the broadcast message to members of the selected role or roles.<br><br>• *User Name*: Sends the broadcast message to the specified user ID or user IDs. |
| **Role/User Name** | Select the role or user ID to be the recipient of the broadcast message. |

| Field or Control | Description |
|---|---|
| **Message** | Enter the message as free-form text. You can use the following variables within the message body:<br><br>• `%OPRID%`: The user's user ID as defined on the user profile.<br><br>• `%EmployeeName%`: The description of the user as defined on the user profile.<br><br>• `%EmployeeID%`: The employee ID for the user as defined on the user profile.<br><br>• `%EmailID%`: The user's primary email address as defined on the My System Profile page. |

## Using Application Data Set Definitions for Events

Use these Application Data Set (ADS) definitions to copy event and collection definitions between databases.

| Field or Control | Description |
|---|---|
| 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
"Application Data Set Overview" (Lifecycle Management Guide)