Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Alpha and Beta Draft Documentation Notice

If this document is in preproduction status:

This documentation is in preproduction status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.
## Contents

Preface ........................................................................................................................................ vii
  Understanding the PeopleSoft Online Help and PeopleBooks ..................................................... vii
  PeopleSoft Hosted Documentation .................................................................................................. vii
  Locally Installed Help ..................................................................................................................... vii
  Downloadable PeopleBook PDF Files .............................................................................................. vii
  Common Help Documentation .......................................................................................................... vii
  Field and Control Definitions ........................................................................................................... viii
  Typographical Conventions ............................................................................................................. viii
  ISO Country and Currency Codes .................................................................................................... ix
  Region and Industry Identifiers ........................................................................................................ ix
  Access to Oracle Support .................................................................................................................. ix
  Documentation Accessibility .......................................................................................................... x

Using and Managing the PeopleSoft Online Help ........................................................................ x
PeopleTools Related Links ............................................................................................................. x
Contact Us ....................................................................................................................................... x
Follow Us .......................................................................................................................................... x

**Chapter 1: Getting Started with Fluid User Interface Development** ........................................ 11
  Fluid User Interface Overview ......................................................................................................... 11
  Implementing PeopleSoft Fluid User Interface .............................................................................. 11

**Chapter 2: Understanding PeopleSoft Fluid User Interface** ................................................... 13
  Understanding the Need for a Fluid User Interface ......................................................................... 13
  Understanding the Characteristics of Fluid Applications .................................................................. 14
  Understanding How to Build Fluid Applications ............................................................................ 16
  Considerations for PeopleSoft Fluid Application Implementation ................................................ 18

**Chapter 3: Creating Pages for PeopleSoft Fluid Applications** .................................................. 19
  Selecting Page Types ...................................................................................................................... 19
  Creating the Primary Fluid Page ....................................................................................................... 21
  Setting Properties for Fluid Pages .................................................................................................... 22
    Setting Use Properties .................................................................................................................... 23
    Setting Fluid Properties ................................................................................................................ 24
  Defining the Layout of the Main Fluid Page ....................................................................................... 25

**Chapter 4: Defining Components for Fluid Applications** ......................................................... 31
  Adding Fluid Pages to a Fluid Component ....................................................................................... 31
  Setting Component Properties for Fluid Components .................................................................... 32
  Working with Search Pages ............................................................................................................. 34

**Chapter 5: Adding Page Controls** ........................................................................................... 37
  Managing Common Control Concepts and Settings for Fluid Pages ............................................ 37
    Understanding the Makeup of a Control on a Fluid Page .............................................................. 37
    Setting Common Page Control Fluid Properties .......................................................................... 37
  Working with Group Boxes ............................................................................................................. 39
    Using Group Boxes for Fluid Pages .............................................................................................. 40
    Setting Group Box Properties ....................................................................................................... 42
  Working with Layout Group Boxes .................................................................................................. 44
  Working with Container Group Boxes ............................................................................................. 45
  Working with Accordion Group Boxes ........................................................................................... 45
  Creating Custom Headers ................................................................................................................ 45
  Working with Related Field Group Boxes ....................................................................................... 46
<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing Group Box Layout</td>
<td>46</td>
</tr>
<tr>
<td>Working with Grouplets</td>
<td>47</td>
</tr>
<tr>
<td>Working with Other Page Controls</td>
<td>50</td>
</tr>
<tr>
<td>Working with Grids</td>
<td>50</td>
</tr>
<tr>
<td>Working with Edit Boxes and Long Edit Boxes</td>
<td>51</td>
</tr>
<tr>
<td>Working with Subpages</td>
<td>52</td>
</tr>
<tr>
<td>Working with HTML Areas</td>
<td>52</td>
</tr>
<tr>
<td>Working with Check Boxes</td>
<td>52</td>
</tr>
<tr>
<td>Working with Push Button/Hyperlink</td>
<td>52</td>
</tr>
<tr>
<td>Working with Static Images and Static Text</td>
<td>53</td>
</tr>
<tr>
<td>Working with Frames</td>
<td>53</td>
</tr>
<tr>
<td>Working with Horizontal Rules</td>
<td>54</td>
</tr>
<tr>
<td>Applying Styles</td>
<td>54</td>
</tr>
<tr>
<td>Working with JavaScript</td>
<td>55</td>
</tr>
<tr>
<td>Delivered JavaScript</td>
<td>56</td>
</tr>
<tr>
<td>Custom JavaScript Guidelines</td>
<td>57</td>
</tr>
<tr>
<td>Chapter 6: Working with PeopleCode for the Fluid User Interface</td>
<td>59</td>
</tr>
<tr>
<td>Working with Fluid Page and Component PeopleCode</td>
<td>59</td>
</tr>
<tr>
<td>PeopleCode Processing Order</td>
<td>59</td>
</tr>
<tr>
<td>Page Type PeopleCode</td>
<td>59</td>
</tr>
<tr>
<td>Using Fluid Component Level PeopleCode</td>
<td>59</td>
</tr>
<tr>
<td>Determining Browser and Device Type</td>
<td>60</td>
</tr>
<tr>
<td>Retrieving Device Type</td>
<td>61</td>
</tr>
<tr>
<td>Retrieving Browser Type</td>
<td>61</td>
</tr>
<tr>
<td>Retrieving Browser Platform</td>
<td>61</td>
</tr>
<tr>
<td>Using Drag and Drop PeopleCode</td>
<td>61</td>
</tr>
<tr>
<td>Working with Grouplet PeopleCode</td>
<td>62</td>
</tr>
<tr>
<td>Working With DoModal Popup PeopleCode</td>
<td>65</td>
</tr>
<tr>
<td>Working with the Back Button</td>
<td>65</td>
</tr>
<tr>
<td>Setting Free Form Page Style</td>
<td>66</td>
</tr>
<tr>
<td>Chapter 7: Managing Fluid User Interface System Settings</td>
<td>67</td>
</tr>
<tr>
<td>Configuring Fluid Mode PeopleTools Options Settings</td>
<td>67</td>
</tr>
<tr>
<td>Configuring Fluid Mode Web Profile Settings</td>
<td>68</td>
</tr>
<tr>
<td>Configuring Tile Options</td>
<td>69</td>
</tr>
</tbody>
</table>
Preface

Understanding the PeopleSoft Online Help and PeopleBooks

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

PeopleSoft Hosted Documentation

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

Locally Installed Help

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

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

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

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

Downloadable PeopleBook PDF Files

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

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.

<table>
<thead>
<tr>
<th><strong>Typographical Convention</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Key+Key</td>
<td>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.</td>
</tr>
<tr>
<td>. . . (ellipses)</td>
<td>Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.</td>
</tr>
<tr>
<td>{ } (curly braces)</td>
<td>Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (</td>
</tr>
<tr>
<td>[ ] (square brackets)</td>
<td>Indicate optional items in PeopleCode syntax.</td>
</tr>
<tr>
<td>&amp; (ampersand)</td>
<td>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.</td>
</tr>
<tr>
<td>⇒</td>
<td>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.</td>
</tr>
</tbody>
</table>
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)

Access to Oracle Support

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

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

Using and Managing the PeopleSoft Online Help

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

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

PeopleTools Related Links

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

PeopleSoft Information Portal

My Oracle Support

PeopleSoft Training from Oracle University

PeopleSoft Video Feature Overviews on YouTube

Contact Us

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

Follow Us

Get the latest PeopleSoft updates on Facebook.

Follow PeopleSoft on Twitter@PeopleSoft_Info.
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.

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 class Peoplesoft applications. A working knowledge and some expertise in the most current, industry-standard internet development tools and techniques is also required. The following list contains the main tools and technologies Oracle recommends developers are familiar with prior to beginning a fluid application implementation.

- PeopleSoft Application Designer
- PeopleCode
- PeopleTools Portal Technology
- PeopleSoft Pure Internet Architecture
- HTML 5.0
- Cascading Style Sheets 3.0
• JavaScript
Understanding the Need for a Fluid User Interface

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

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

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

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

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

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

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

Image: Fluid page

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

**Image: Fluid page in landscape mode**

The following example shows a fluid page on a smartphone displayed in landscape mode.

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

If you access the same page using a cell phone and view it in portrait mode, you will notice additional changes as the page further adapts and responds to the adjusted screen display. The PeopleSoft Fluid User Interface is *adaptive* in that the system detects the device and generates the UI appropriate to that device, and it is *responsive* in that the CSS running on the client device that responds to user actions, such as changing from portrait to landscape mode.
**Note:** The PeopleSoft Fluid User Interface adaptive qualities require a server trip, while the responsive qualities do not.

**Image: Fluid page in portrait mode**

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

Notice that most of the buttons contain only images and no text. Also notice that the grid contains even fewer columns, and the group boxes are aligned one below the other.

**Related Links**

"Working With Fluid Homepages" (PeopleTools 8.54: Applications User's Guide)
"Working With Fluid Pages and Controls" (PeopleTools 8.54: Applications User's Guide)

---

**Understanding How to Build Fluid Applications**

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

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

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

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

- CSS 3
- HTML 5
- JavaScript

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

The majority of PeopleSoft applications, components, and pages will continue to be accessed in classic mode, which is the way PeopleSoft applications have been accessed on 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 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, such as page layout, some elements, such as PeopleCode business logic, can be re-used 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.

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not all applications (currently) are candidates for fluid deployment.</td>
<td>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.</td>
</tr>
<tr>
<td>Are mobile devices such as cell phones and tablets used widely to access PeopleSoft applications at your site?</td>
<td>If your user base already accesses or has expressed a demand for accessing your PeopleSoft applications using mobile devices, fluid deployment would be a good option for you. However, if mobile devices are not part of your user base or application delivery strategy, you can elect to disable fluid deployment at the system level.</td>
</tr>
<tr>
<td>Application usage and workload?</td>
<td>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.</td>
</tr>
<tr>
<td>Does an application require functionality that does not exist in the fluid user interface?</td>
<td>For example, some features, like spell check, rich text editor, Find In for grids, Download to Excel, modal movement or resizing, are not available in fluid.</td>
</tr>
</tbody>
</table>
Creating Pages for PeopleSoft Fluid Applications

Selecting Page Types

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

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

Image: New Definition dialog box

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

**Image: Page Properties dialog box**

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

![Image of Page Properties dialog box](image)

While each fluid page type will be discussed separately in detail, the following table provides a brief description of each fluid page type.

<table>
<thead>
<tr>
<th><strong>Page Type</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluid Page</td>
<td>A fluid page is a standard page that has been designated a Fluid Page on the Use tab of the Page Properties. The fluid page definition contains the bulk of the transaction fields and controls.</td>
</tr>
<tr>
<td>Header Page</td>
<td>Page displayed in &lt;header&gt; section of the HTML acting as the banner area fixed at the top of every page. It does not scroll with the page. The header page appears at the top of the browser interface over the main page, containing elements used for basic navigation and orientation, like a back button, page title, home button, search button, and so on. PeopleTools supplies a default header page, PT_HEADERPAGE, but this can be overridden for a custom header page to be used by selecting No System Header Page in the Component Properties dialog box.</td>
</tr>
<tr>
<td>Side Page</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>Footer Page</td>
<td>Page displayed in the &lt;footer&gt; 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.</td>
</tr>
<tr>
<td>Search Page</td>
<td>Page generated in &lt;aside&gt; section containing search pages, which include Find an Existing Value and Keyword Search. <strong>Note:</strong> An Add a New Value page is not provided by the search page definitions provided by PeopleTools.</td>
</tr>
<tr>
<td>Page Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Layout Page</td>
<td>You can designate a page definition as a layout page for use as a template for development reuse or for testing purposes, such as finalizing CSS work.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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).</td>
</tr>
</tbody>
</table>

### Creating the Primary Fluid Page

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

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

1. In Application Designer, select File, New.
2. Select Page (Fluid) on the New Definition dialog box, and click OK.
3. Select an initial layout page (template) from the Choose Layout Page and click Choose.
4. Enter the page name on the Save As dialog box.
5. Indicate whether you’d like to save the PeopleCode associated with the layout page for your new page (Yes/No).
6. Define layout and add page controls according to the fluid techniques and guidelines described in this documentation.

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

1. Open the existing standard page in Application Designer.
2. (Recommended) Select File Save As, and rename the page to distinguish it from the standard, classic PIA page.

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

3. Define layout and add page controls according to the fluid techniques and guidelines described in this documentation.
4. Access the Page Properties dialog box, select the Use tab, and enable the Fluid Page check box.

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

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

Related Links

"Understanding Page Design and Development" (PeopleTools 8.54: Application Designer Developer's Guide)
"Creating New Page Definitions" (PeopleTools 8.54: Application Designer Developer's Guide)
Setting Use Properties

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

Image: Page Properties > Use tab

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

Page Type

Select the type of page for use in the fluid application, such as header, footer, search, and so on.

For more information on page types used for fluid development, see Selecting Page Types .

Fluid Page

For a page to be displayed and rendered in fluid mode, the Fluid Page check box must be selected. This is required.
Setting Fluid Properties

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

**Image: Page Properties > Fluid tab**

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

![Page Properties dialog box](image)

**Free Form Style**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Override Tools Style</strong></td>
<td>Select to bypass the system default style completely.</td>
</tr>
<tr>
<td></td>
<td>If not selected, the custom free form style specified will be appended to</td>
</tr>
<tr>
<td></td>
<td>the default system style, such that both style sheets are used.</td>
</tr>
<tr>
<td><strong>Default Style Name</strong></td>
<td>Enter the custom default style name to include.</td>
</tr>
</tbody>
</table>

**Form Factor Override**

<table>
<thead>
<tr>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td></td>
</tr>
<tr>
<td>ExtraLarge</td>
<td></td>
</tr>
</tbody>
</table>
**Form Factor Override**

Use to override the style based on the form factor (device size) of the device accessing the page. Form factors are represented in these ranges:

- Small: Applies to most smartphones.
- Medium: Applies to 7-inch tablets.
- Large: Applies to 10-inch tablets and laptops.
- Extra Large: Applies to large monitors.

**Suppress on Form Factor**

Use to prevent the page from displaying for specified form factors. Options are:

- Small
- Medium
- Large
- Extra Large

For example, if you select Extra Large, the page would not appear in fluid mode on the desktop PC. It would only appear in fluid mode on a phones, tablets, and laptops. This also aids performance because if it is not set to appear on a specific device it doesn’t just prevent display of the page to the end user, the system does not send that page to the device at all, meaning the system saves on network traffic, rendering, and CSS processing.

**Defining the Layout of the Main Fluid Page**

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

**Image: Classic page definition at design time**

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

**Image: Classic page at runtime**

This example illustrates the same page as depicted in the previous example at runtime. The design time and runtime appearance is practically identical in terms of layout, positioning, sizing, color, font, and so on.
When performing layout for a fluid page, you will notice that design representation in no way resembles the design representation for the classic PIA page.

**Image: Fluid page at design time**

This example shows a fluid page layout at design time in Application Designer, with various elements contained in nested group boxes and little if any indication regarding how the page might appear at run time to the end user.

The layout of the fluid application is not what you would call a WYSIWYG approach. If an ender user were to access a page in this state, it would be impossible to complete a transaction. Notice the boxes that are used to indicate levels of containment and to provide visual cues, making each element from
high-level to low-level a self contained and discrete. Group boxes are used to achieve this effect. The rectangular boxes within some of the group boxes represent subpages.

**Image: Fluid page at run time**

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

The strip running across the top displays the default PeopleTools, or system, header page definition. Because it is a separate page definition entirely, it is not depicted in the layout view of the main page (the content below it) in Application Designer.

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

- The area showing the employee photo is represented in the layout as the App Header Photo group box.
- The area showing the navigation on the left is represented in the layout as the Left Side Navigation group box.
- The area showing the Name and Addresses fields is represented in the layout as the Content group box. When Names and Addresses is selected from the left hand navigation, it appears in the Content container. When Contact Details is selected, the contact information is displayed in the Content container, and so on.

While there are many factors contributing to the appearance of a fluid page, which will be discussed in this document, the main elements of fluid page layout are:
<table>
<thead>
<tr>
<th>PeopleSoft Design Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>group boxes</td>
<td>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.</td>
</tr>
<tr>
<td>style sheets</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.
Adding Fluid Pages to a Fluid Component

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

Note: When using the default system pages, as in the default header page provided by PeopleTools, the system automatically loads those pages into the component. If you intend to use those pages, they do not need to be added manually to the component.
Setting Component Properties for Fluid Components

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

Image: Component Properties dialog box: Fluid tab

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

Component Attributes

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

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

**Page Navigation in History**
Select to record the navigation among the fluid pages for use with the Back button.

**Small Form Factor Optimized**
Select if the pages in the component have been designed specifically for optimum display and performance on a smartphone. If selected, this component will be available on small form factor devices. Otherwise, on small form factor devices, this component would not be available as a tile on the homepage, for example, but could still be accessed using the Navigator.

**No System Header**
Select if you have a custom header page that you want to display instead of the default, system header.

**Enable Search Page**
Select to enable search pages for the component, which, just like classic PIA applications, include the Find an Existing Value page, Add a New Value page, and Keyword Search page.

**Header Toolbar Actions**

**Disable All Actions**
Select to deselect all header toolbar actions. At runtime the system displays a header page toolbar containing no buttons.

**Logout**
Select for the Logout button to appear on the header toolbar.
Use to logout, or sign off, of the application, returning the user to the fluid login page.

**Home**
Select for the Logout button to appear on the header toolbar.
Returns the user to the default fluid homepage, as if they’ve just signed on.

**Back**
Select for the Back button to appear on the header toolbar.
Returns the user to the previous page.

**Add To**
Select for the Pin (Add To) button to appear on the header toolbar.
Enables a user to add a link to the component to the homepage or the Navbar or Favorites.

**Notifications**
Select for the Notifications button to appear on the header toolbar. Enabled by default.
The Notification button shows the list of notifications for the logged in user, such as events and notifications from AWE.
Use this option to disable notifications while within a specific component (exception to rule and also allows debugging).
The banner uses this option to determine whether to hide or show the notification button and to enable the appropriate JavaScript.

**NavBar**

Select for the NavBar button to appear on the header toolbar.

**Note:** For the current release, the NavBar is not supported for right-to-left languages, such as Arabic.

**Search**

Select for the Global Search button to appear on the header toolbar.

Enables free form search for content using the Search Framework.

**Next in List**

Select for the Next in List button to appear on the header toolbar.

Takes the user to the next search result on a search page.

**Previous in List**

Select for the Previous in List button to appear on the header toolbar.

Takes the user to the next search result on a search page.

---

**Working with Search Pages**

If your component requires a search page, check “Enable Search Page” in component properties.

The classic search page is not available for fluid applications. Depending on what type of search capability you want to provide for the component you have these options for search pages:

<table>
<thead>
<tr>
<th>Search Page Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT_SEARCHPAGE</td>
<td>The fluid equivalent of the classic component search page. When Enable Search Page is selected, the system automatically loads this search page definition for end users to use for retrieving information.</td>
</tr>
<tr>
<td>PTS_NUI_SEARCH</td>
<td>Provides a more full-featured set of search results for using SES-based Search Framework searches as well as a pivot grid-based component search, which resembles the SES based search results. If you want this option, you must add the PTS_NUI_SEARCH page to the component definition so that it will be loaded instead of the default, PT_SEARCHPAGE, when Enable Search Page is selected.</td>
</tr>
</tbody>
</table>

The delivered search page definitions (PT_SEARCHPAGE or PTS_NUI_SEARCH) must be the base for any custom or system search page as it contains the basic framework definition and structures to use as a reference. For any custom search page, make sure to select Search Page as the page type on the Page.
Properties, Use tab. You will need to also use the pivot grid publishing wizard to select the options for fluid search pages.
Chapter 5

Adding Page Controls

Managing Common Control Concepts and Settings for Fluid Pages

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

Understanding the Makeup of a Control on a Fluid Page

A basic input page control consists of two items:

• The control.
• The label.

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

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

DIV.ps_box-edit
  DIV.ps_box-label
    LABEL.ps-label
  DIV.ps_box-control
    INPUT[type=text].ps-edit

Image: HTML for fluid page controls

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

```
<dir id="win0divEMERGNCY_CNT_VW_CONTACT_NAME" class="ps_box-edit psc_fld-standard">
  <div id="win0divEMERGNCY_CNT_VW_CONTACT_NAME1bl" class="ps_box-label">.
    </div>
  <div id="win0divEMERGNCY_CNT_VW_CONTACT_NAMEctrl" class="ps_box-control"/>
</dir>
```

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

Setting Common Page Control Fluid Properties

For any control that you can add to a page, such as a group box, an image, a grid, and so on, there is a Fluid tab on the properties dialog box for that control. All of the controls share some properties, while others have additional properties that apply only to the function of that control.
Note: The options documented below apply to all fluid controls. Depending on the control selected for the page, additional properties may appear. Properties specific to an individual control type are documented within a section devoted to that control type.

Image: Common Fluid tab Options

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

Override Tools Style
Indicates to the system not to render the default class for that specific control type (e.g. the ps_box-edit in the previous example). If this is applied, it applies to all form factors.

Default Style Name
Enter additional class specifications used to adjust the display or behavior of a specific control. This setting applies to all form factors.
factors unless specifically overridden by subsequent form factor overrides.

**Form Factor Override**
Applies classes specifically for adjusting the display (and possibly behavior) of an element on a specific form factor. If no value is specified at the form factor override, the default Style Name will be applied.

**Suppress on Form Factor**
To prevent a page element from being displayed on specific form factor (device size), select the form factor. If selected, the page element is never rendered on the client device, similar to setting a field as invisible for a classic page. Suppression can improve throughput as the size of the HTML document is reduced.

**Label Rendering**
The system processes the display of a control as a combined unit of both the display and its label. You can control the position of the label.

*Before Control:* Select for the label to appear to the right of the control.

*After Control:* Select for the label to appear to the left of the control.

**Control Structure**
*Basic:* Select to have the control appear on the fluid page just as it would appear on a classic PIA page.

*Advanced:* Select to have the control appear on the fluid page having the look and feel of a typical touch-screen application for a tablet or cell phone. For example, a check box would appear as a sliding switch, rather than an empty square box.

---

**Working with Group Boxes**

This topic discusses:

- Using Group Boxes for Fluid Pages
- Setting Group Box Properties
- Working with Layout Group Boxes
- Working with Container Group Boxes
- Working with Accordion Group Boxes
- Creating Custom Headers
- Working with Related Field Group Boxes
- Using a Toggle Container
• Testing Group Box Layout

• Working with Grouplets

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 start off with a containing group box as well.

**Image: Group Boxes define fluid page layout**

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

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

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

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

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

Image: Group Box Properties Dialog Box: Fluid Tab

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

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

**Group Box Type**

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

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

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

**Note:** Currently, do not use Accordion Horizontal.

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

**Tab Container:** Use for creating a side tab.

**Note:** Side tabs should not be used in the current release.

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

**Page Title:** This is a container for the page title. The header description is generated in a H1 tag (\texttt{<H1>..\texttt{<H1>}). It is used by the header page. Related Field Group: Use to group a set of fields.

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

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

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

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

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

**Custom Header Search:** Use to create a custom header for component search.
### HTML Tag Type

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

*Default:* Same as DIV.

**DIV:** Set as an HTML 5.0 DIV tag.

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

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

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

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

### Draggable

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

### Droppable

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

### Scroll Bars

*None:* Display group box with no scroll bar.

*Vertical:* Display the group box with a scroll bar on the right side, running top-to-bottom.

*Horizontal:* Display the group box with a scroll bar on the bottom, running left-to-right.

*Both:* Display both vertical and horizontal scroll bars.

*Carousel:* Display the scroll bar in carousel mode (slider mode), where records can be viewed in defined sets by spinning or swiping.

### Scroll Sizing by Style

Indicates that the size of the scroll will be controlled by style.

By default, PeopleTools attempts to control the sizing of the group box to accommodate the size of the current page size. If you want to restrict the size of the group box stylistically (using classes and CSS), you can do this so that the system does not attempt to change the size of the group box as needed.

### Working with Layout Group Boxes

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

- No default look and feel style is applied, such as no border.
• No header will be generated.
• The label set for the group box will be ignored at runtime, but can be used at design time for reference.

Working with Container Group Boxes

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

When using container group boxes, keep these items in mind:
• A default, system style provides a standard look and feel.
• A header will be generated if a label is defined.
• Custom styles can be used to override default styles.
• The default HTML tag used for group box is “div,” which you can override with the HTML Tag Type option.
• You can set the group box as Section and Article elements.
• Can be used as a collapsible data area.

Working with Accordion Group Boxes

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

When working with the accordion feature:
• Set the parent group box container as Accordion Vertical for vertical accordion behavior (up and down).
• 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.

Creating Custom Headers

If you do not want to use the provided default system header page, you can use the custom header group boxes to create your own. Before doing so, become familiar with the default system header page definition, PT_HEADERPAGE.

To create a custom header page, create a new page with a Page Type of Header page. You will notice that the header page contains a left box, middle box, right box and a bottom. These correspond directly to the Custom Header Left/Middle/Right/Bottom group box types. Use these group boxes to inject the application controls into the header page sections. At runtime, the content of the group box will replace the content of the custom section in the 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
<pre><code>&lt;fieldset&gt;...&lt;/fieldset&gt;</code></pre>
The group header will be within <legend> instead &lt;h..&gt;.
The Collapsible Data Area option should not be used for related field group boxes.

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
General tab of the Component Properties dialog box.

Image: Enabling Group Box Layout Testing

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

**Image: Group Box Testing Display**

This example illustrates the different colored boxes indicating the scope and coverage of each group box.

---

**Working with Grouplets**

A grouplet is a group box on a fluid page that is comparable to a pagelet.

These are the grouplet types:

- [ ] On
- [ ] Off
- [ ] All Browsers
- [ ] Exclude Netscape
- [ ] Do Not Compress
• Consumer Grouplet: The tile is a super container that will consume content from another local/remote component, iScript, or third party content.

• Producer Grouplet: Produces display-only content with no form fields. External URLs, email links and related content actions are supported for hot spots.

• Grouplet/Tile: Can use an event to display more details in a modal, a new window session, or replace the current window to a target group box from an action URL.

Use the Pagelet Wizard to publish the grouplets, similar to how you publish pagelets. Publishing as a grouplet is supported for the HTML data type.
On step 6 of the Pagelet Wizard, you select the Embeddable Pagelet option, and the system displays the Publish as Grouplet link. Click the link to open the Publish Grouplet Definition.

**Image: Pagelet Wizard: Publish Grouplet Definition**

This example illustrates the fields and controls on the Publish Grouplet Definition page.

**Pagelet Wizard**

**Publish Grouplet Definition**

Create or update content reference object using this item as grouplet

<table>
<thead>
<tr>
<th>Target Page Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Page Type</strong></td>
</tr>
<tr>
<td><strong>Target Page Name</strong></td>
</tr>
<tr>
<td><strong>Target Page Label</strong></td>
</tr>
<tr>
<td>Long Description</td>
</tr>
<tr>
<td>Object Owner ID</td>
</tr>
<tr>
<td><strong>Parent Folder</strong></td>
</tr>
<tr>
<td><strong>Node Name</strong></td>
</tr>
<tr>
<td>Sequence Number</td>
</tr>
</tbody>
</table>

**Image: Pagelet Wizard: Publish Grouplet Definition**

This example illustrates the fields and controls on the Publish Grouplet Definition page.

<table>
<thead>
<tr>
<th>Grouplet Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grouplet Height</strong></td>
</tr>
<tr>
<td><strong>Grouplet Width</strong></td>
</tr>
<tr>
<td>Image Name</td>
</tr>
<tr>
<td><strong>Current Window</strong></td>
</tr>
<tr>
<td><strong>New Window</strong></td>
</tr>
<tr>
<td><strong>Modal Window</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tile Content Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Pagelet ID</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
</tr>
</tbody>
</table>
Note: Grouplets can be used also in classic PIA to share information between components. If implemented in a classic PIA application, it is recommended to share only lightweight, well-defined information in a grouplet.

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 PIA 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 classic PIA page development and all of the various page control options provided by Application Designer prior to beginning a fluid development or customization project.

Working with Grids

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

**Scroll Sizing by Style**

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

- If the custom style does not control the scroll container size successfully, the scroll may not appear or work as desired.
- Your custom style must be able to handle window size and orientation change.
- If there is no overflow amount (data rows exceeding occurs limit) lazy scrolling may not be triggered.

**Custom Grid Header**

Select to specify a custom grid header.

Note: If you want to add the scrolling feature to a grid (or scroll area), the grid must be placed within a group box.

**Include Labels in Grid Cells**

If using a Tab Separator in your grid, you are allowed to generate field labels within a grid.

Typically, these are suppressed but because we have the ability to put group boxes within grids, the labels are appropriate to display what the individual fields are (depending on context).
Working with Edit Boxes and Long Edit Boxes

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

**Place Holder Text**

Display default, place holder, text in the edit box. For example, add a default value or sample format of the field data to assist users.

- None: Disables the display of place holder text.
- Static: Select to enable the Static Text box, and enter the desired text.
- Message Catalog: Select to enable the Message Set/Number boxes. Enter the message set and message number of the message catalog entry that contains the place holder text.

HTML example:

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

**Input Properties**

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

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

- text
- date
- datetime
- email
- number
- range
- search
- time
- url

*Note:* Does not apply to long edit boxes.
Working with Subpages

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

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

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

There can be multiple levels of subpages.

Working with HTML Areas

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

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

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

Working with Check Boxes

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

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

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

**Image: Advanced and Basic Check Box Rendering**

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

```
Active Flag  Active

☑ Agreement to Conditions
```

Working with Push Button/Hyperlink

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

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

- Execute PC on Row\Group click: Runs PeopleCode as soon as user clicks on row\group).
- Execute SearchSave Event: When a user runs a search, the system saves the page data prior to running the search so it is not lost. This option applies only to the search button, not every button on the page.

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

### Working with Static Images and Static Text

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

**Image: Static Text**

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

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

**Image: Static Image**

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

```html
<div class="ps_box-staticing" id="win0div$ICField4">
  <img src="/PRODUCT/SAVE%20FULL%20TEXT/<br class="ps-staticing" alt="Alternate Text for Image" title="Alternate Text for Image" class="ps-staticing-alt">
</div>
```

### Working with Frames

Similar to classic PIA pages, using the frame control is not necessarily recommended, but it can be used a container. Rather than using the frame control, it is recommended to use a group box of type layout instead.
Working with Horizontal Rules

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

Applying Styles

The layout of all page elements is controlled by free from style sheets using CSS 3.0 standards. 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 and varying levels.

The system 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, $PSSTYLEDEF\_FMODE$.

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

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

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

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

Image: Style Sheets in HTML head section

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

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 the
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.

**Image: Form Factor Override**

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

<table>
<thead>
<tr>
<th>Form Factor Override</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>psc_border</td>
</tr>
<tr>
<td>Medium</td>
<td>psc_border psc_width-75pct psc_margin-auto</td>
</tr>
<tr>
<td>Large</td>
<td>psc_border psc_width-50pct</td>
</tr>
<tr>
<td>Extra Large</td>
<td></td>
</tr>
</tbody>
</table>

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

Free form styles set in Application Designer are static and can’t be changed, however, styles set using PeopleCode are dynamic.

Example:

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

---

**Working with JavaScript**

This topic discusses:

- Delivered JavaScript.
- Custom JavaScript guidelines.
Delivered JavaScript.

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

**Image: PT_PAGESCRIPT_FMODE**

This example illustrates the delivered JavaScript in PT_PAGESCRIPT_FMODE.

```javascript
/* Copyright ToolsDel: %ToolsDel */

/* drag and drop api testing sample - not used at the runtime */

function initDragDrop() {
  var sccOb = document.getElementById('sccId'); // document.getElementById('aside_bottom').childNodes[0];
  sccOb.addEventListener('dragstart', dragged, false);
  sccOb.addEventListener('dragover', function (e) { e.preventDefault(); }, false);
  sccOb.addEventListener('drop', dropped, false);
}

function dragged(e) {
  var srcOb = document.getElementById('aside_bottom'); // ptCommonObj3.getElementById;
  e.dataTransfer.setData('src_content', srcOb.innerHTML);
}

function dropped(e) {
  var trgOb = ptCommonObj2.getElementById;
  e.preventDefault();
  trgOb.innerHTML = e.dataTransfer.getData('src_content');
}
```

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

- Scrolling for group boxes
- Accordion widget
- Toggle widget
- Grouplet (like a pagelet – smaller)
- Drag and drop group boxes
- Grid lazy scroll
- Carousel – (used only for scroll areas)
- Transfer animation
- File attachment
- Search page and nav bar
- Related action menu
- Side page tabs
- New action buttons – Back, home, logout
- Search side page
Custom JavaScript Guidelines

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

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

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

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

- Avoid using custom JavaScript events on a control.

- If there are any JavaScript errors from app onload, the page will also encounter problems loading. Do not use JavaScript to change the structure of the document.

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

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

PeopleCode Processing Order

The PeopleCode processing order for Page Activate is:

1. Header
2. Side
3. Footer
4. Main

Page Type PeopleCode

PeopleCode for page types of Header, Side and Footer:

- **Page.Visible**
  - Dynamically control page visibility.
- **GetPageType**
  - Retrieves the page type.
- **SetPageType**
  - Sets page type dynamically.

Using Fluid Component Level PeopleCode

The following table contains common component level PeopleCode used in fluid applications.

- **IsSaveEnabled()**
  - Checks if the Save button is enabled.
  - Returns true/false.
- **IsLogoutEnabled()**
  - Checks if the logout button is enabled.
  - Returns true/false.
- **IsHomeEnabled()**
  - Checks if the Home button is enabled.
  - Returns true/false.
IsBackEnabled() Checks if the Back button is enabled.
Returns true/false.

IsPinEnabled() Checks if the Pin button is enabled.
Returns true/false.

IsNotifyEnabled() Checks if the Notify button is enabled.
Returns true/false.

IsNavBarEnabled() Checks if the NavBar button is enabled.
Returns true/false.

IsNextInListEnabled() Checks if next in list for search is enabled.
Returns true/false.

IsPrevInListEnabled() Checks if previous list for search is enabled.
Returns true/false.

IsSearchEnabled() Checks if search is enabled.
Returns true/false.

IsReturnToSearchEnabled() Checks if return to search is enabled.
Returns true/false.

IsGroupletRequest() For checking if current request is a Grouplet request.
Returns true/false.

IsFluidMode() For checking if the component is fluid-enabled.
Returns true/false.

IsStartSearch() For checking if a search has been started but not entered the component yet.
Returns true/false.

GetPageTitle() Returns the current page’s title.
The title is from page menu label defined in the component.

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. At runtime, the HTML page-level class is determined by the device type, browser type, and browser platform (operating system).
Retrieving Device Type

To retrieve the device type, use:

\%Request.BrowserDeviceType

Returns are: desktop/tablet/phone

Examples:

If \%Request.BrowserDeviceType <> "phone" Then
   Local Page &pg = GetPage(@("Page.HGA_SS_ADDTNL_SPF"));
   &pg.FreeFormStyleName = "psa_pref_modal_sizing_addtnl_info";
End-If;
*************************************************************************
If \%Request.BrowserDeviceType <> "phone" Then /* STUB CHANGE TO FORMFACTOR SM⇒
ALL */
   &strColLabel = &EmpGrid.GetColumn("HR_PSEL_GRPBX_COL1").Label | " / " | &s⇒
trColLabel;
*************************************************************************
If \%Request.BrowserDeviceType = "desktop" Then
   &menuItemName = MsgGetText(268, 487, "(Message not found)Export Data");
   &hmBody.setMenuListItem(&menuItemName, "PTPG_NUI_WRK_PTPVG_EXPORT_EXCEL", &bA⇒
dmnShowExportData);
End-If;
*************************************************************************
If \%Request.BrowserDeviceType = "phone" Or
   \%Request.BrowserDeviceType = "tablet" Then
   &rl1().TL_TR_WEEK_WRK.RT_SOURCE.Value = "MD";
   /* default new entry source as mobile */
End-If;
*************************************************************************

Retrieving Browser Type

To retrieve the browser type, use:

\%Request.BrowserTypeClass

Returns are: chrome/safari/firefox/ie

Retrieving Browser Platform

To retrieve the browser platform, use:

\%Request.BrowserPlatformClass

Returns are: win/ios/mac/android

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.

For a source, use: For a source, use:
GetDNDRow Get the row where the drag initiated.

GetDNDRowset Get the rowset where the drag initiated.

GetDNDField Get the field where the drag initiated.

Example:
Local integer &fromRowNum = GetDNDRow().RowNumber;

For a target, use:

GetDNDTargetRow Get the row on which the data is being dropped.

GetDNDTargetRowset Get the rowset on which the data is being dropped.

GetDNDTargetField Get the field on which the data is being dropped.

Example:
Local Rowset &rs2 = GetDNDTargetRowset();

To restrict source and target use:

AddtoDNDSourceList(record.field) Ensures the drag and drop happens only from specific group boxes.

RemoveFromDNDSourceList(record.field) Removes a group box from the source list.

Example:
Local Row &dragRow = GetDNDRow();
Local Rowset &rs = GetDNDRowset();
Local Field &field1 = GetDNDField();
Local integer &toRowNum = GetDNDTargetRow().RowNumber;
Local integer &fromRowNum = GetDNDRow().RowNumber;
Local Rowset &rs2 = GetDNDTargetRowset();

If &rs.Name = "PTNUI_LAND_REC" Then
    If &rs2.Name = "PTNUI_LAND_REC" Then
        &lpRS.moveToTile(&lp.ActiveTab, &fromRowNum, &lp.ActiveTab, &toRowNum);
    End-If;
    If &rs2.Name = "PTNUI_HPPNL2REC" Then
        &lpRS.addTileFromRS(&dragRow, GetDNDTargetRow(), -1);
        &lpRS.deleteTile(&fromRowNum);
    End-If;
Else
    If &rs2.Name = "PTNUI_LAND_REC" Then
        End-If;
    If &rs2.Name = "PTNUI_HPPNL2REC" Then
        &uhp2RS.MoveHP(&fromRowNum, &toRowNum);
    End-If;
End-If;

Working with Grouplet PeopleCode

This topic discusses PeopleCode used with grouplets.
### Setting Grouplet Type

`.SetGroupletType(n)` – Used to set grouplet type.

`.SetGroupletType(2)` – Grouplet No Style // default.

`.SetGroupletType(1)` – Grouplet with Style and JavaScript and onload JavaScript.

`.SetGroupletType(0)` – Not a grouplet.

N=1 – This producer grouplet provides content without style sheets. This is the recommended option.

N=2 – If this producer grouplet provides content containing the style sheets, they will be inserted in the consumer page. Be aware of style conflicts/overwriting. Not Recommended.

N=2 – If this producer grouplet provides content containing the JavaScript files, they will be inserted in consumer page. Be aware of function conflicts/overwriting. Not Recommended.

N=2 – If this producer grouplet provides content containing the JavaScript calls, they will be executed after processing the response and external files. Not Recommended.

### Setting Grouplet URL

`.SetGroupletUrl(url)` – Set producer content URL to retrieve dynamic content and display in the consumer group box/tile.

Example:


### Setting Grouplet ID

`.SetGroupletID(record_field)` – Used to set Producer Grouplet ID.

Example:

`&RS_OPTN(2).PSUSEROPTNDEFN.FIELDNAME.SetGroupletID("DERIVED_PSWD_OPERPSWDCONF")`

Grouplet ID is optional. If Grouplet ID is not provided, the top group box will be the default.

### Setting Grouplet Action URL

Use `.SetGroupletActionUrl(url)` to set the action URL to perform the action of the group box/tile when clicked or touched. The URL can be displayed in the Current/New/ModalWindow.

Example:


### Setting Grouplet Display Location

Use `.SetGroupletDisplayIn(n)` for Action URL with these options:
• Current Window(0)
• New Window(1)
• Modal(2)
• TargetGroup(3)

Examples:

Display in modal.
SetGroupletDisplayIn(2);

Display in a target group box.
SetGroupletDisplayIn(3);

Set Grouplet Target ID
Display in the target group box ID.
SetGroupletTargetID("TARGETGROUP$0");

Setting Grouplet Image
Use .SetGroupletImage(..) for loading the image to display on the grouplet.

.SetGroupletImage("ORG_CHART_128");

Setting Grouplet Content Reference
Use SetGroupletCRefID(..) to set the grouplet CRefID. Often used on the fluid homepage for folder navigation.

.SetGroupletCRefID("folder2");

Setting Grouplet Folder
Use .IsGroupletFolder to specify if it is a folder grouplet. Often used on the fluid homepage for folder navigation.

.IsGroupletFolder = True;

Setting Grouplet Label Only
.IsGroupletLabelOnly – Set if it is a folder grouplet with label only and no preload image. Commonly used by fluid homepage for folder navigation.

.IsGroupletLabelOnly = True;

Setting Grouplet Refresh
Use .SetGroupletTimer(n) to refresh the grouplet display in $n$ seconds. Refresh is performed by web worker.
Specifying Grouplet Request
Use .IsGroupletRequest() to determine if a request is a grouplet request (true/false).

---

Working With DoModal Popup PeopleCode

Use DoModalPopup to:

- Display a secondary page as a popup modal next to a parent control.
- Present a display-only secondary page as a DIV group container inserted into a parent page that can be cached.

DoModalPopup(sModalOptions, bDisplayOnly, bCache)

Where:

- sModalOptions – custom modal options.
- bDisplayOnly (True/False) – secondary page will be generated in display modal. It will be inserted on parent page and display it as a popup modal.
- bCache (True/False) – Used when bDisplayOnly=true. If bCache=true, the secondary page will be cached on the parent page. No server trip occurs if displayed again.

Examples:

```plaintext
&sModalOptions = "width@500;height@600;bAutoClose@1;bAutoCloseWarn@1;";
&sModalOptions =""; /* take all defaults */
If DoModalPopup(&sPopupOptions, "", False, False, Page.PSUSEROPTNDEFNDFLT, " ", - 1 ⇒ , - 1) = 1 Then ...
```

---

Working with the Back Button

The Back button enables you to configure the system to track user history and enable the user to retrace steps as needed. Including the Back button is an option for the system header page.

The main PeopleCode constructs used in this feature are:

- TransferPanel
- SetTransferAttributes

SetTransferAttributes(bAnimation, bEnableHistoryTrack, sLabel)

Where:

- bAnimation: Use to enable transfer animation, override the back label, or disable the back button on a specific transfer. (true/false)
- bEnableHistoryTrack: (true/false) Disable tracking the transfer for a back action. Default is true.
- sLabel: Use to include a string to customize the label appearing in the back button.

For example, setting a custom label:

```peoplecode
SetTransferAttributes(True, True, "My Page 2");
TransferPanel(Page.FSUSEROPTNDEFN2);
```

For example, disabling history tracking:

```peoplecode
SetTransferAttributes(False, False, "");
```

---

### Setting Free Form Page Style

Use FreeFormStyleName to set page style and define minimum width and height for modal pages.

Example:

```peoplecode
GetPage(Page.USER_PSWDHINT_NUI).FreeFormStyleName = "my_modalpage";
```

It is highly recommended to use page-level free form style to define a minimum width for better auto sizing of modal pages.

Example:

```peoplecode
If IsModalComponent() Then
    Local Page &pg = GetPage(@("Page." | %Page));
    &pg.FreeFormStyleName = "psa_pref_modal_sizing";
End-If;
```
Chapter 7

Managing Fluid User Interface System Settings

Configuring Fluid Mode PeopleTools Options Settings

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

Image: PeopleTools Options: Fluid Mode Properties

This example illustrates the Fluid Mode Properties on the PeopleTools Options page. Descriptions of the properties appear after the example.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Style Sheet Name</td>
<td>Select the layout style sheet. This is the equivalent of the system style sheet for classic pages.</td>
</tr>
<tr>
<td>Header Page Name</td>
<td>The default header page is PT_HEADERPAGE. If you want to override this page, enter your custom page.</td>
</tr>
<tr>
<td>Side Page Name</td>
<td>The default side page is PT_SIDEPAGE. If you want to override this page, enter your custom page.</td>
</tr>
<tr>
<td>Prompt Page Name</td>
<td>The default side page is PT_PROMPTPAGE. If you want to override this page, enter your custom page.</td>
</tr>
<tr>
<td>Footer Page Name</td>
<td>The default footer page is PT FOOTER_SAVEONLY.</td>
</tr>
<tr>
<td>Search Page Name</td>
<td>The default search page is PT_SEARCHPAGE</td>
</tr>
</tbody>
</table>
## Configuring Fluid Mode Web Profile Settings

Select PeopleTools, Web Profile, General to access the fluid mode web profile settings.

**Image: Fluid Mode Web Profile Properties**

The following example illustrates the web profile properties you can set for fluid applications.

<table>
<thead>
<tr>
<th>Disable Fluid Mode</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable Fluid On Desktop</td>
<td>?</td>
</tr>
</tbody>
</table>

### Disable Fluid Mode

Setting this option will disable all fluid pages and content references. Only the classic equivalent content references and links will be available throughout the entire system. This option allows system administrators to force classic-only mode for all users.

**Note:** This option will disable all fluid content references. Applications delivered only as fluid components will not be accessible with this option set.

### Disable Fluid on Desktop

Setting this option will display only classic pages on desktops and laptops.

By default this option is selected (enabled), which means that by default, browsers running on desktop computers display classic PIA pages, not the fluid equivalents.

**Note:** This option is effective only when the Disable Fluid Mode is not set.
Configuring Tile Options

To configure the behavior and appearance of tiles use the NUI Attributes page of the target content reference. (PeopleTools, Portal, Structure and Content, and drill into the folder hierarchy as needed, then click Edit on the appropriate content reference.)

**Image: Fluid (NUI) Attributes**

This example illustrates the fluid attributes for content references.

![Image of Fluid (NUI) Attributes](image)

### URL Information

<table>
<thead>
<tr>
<th><strong>Image Name</strong></th>
<th>To display a custom image on the tile, select the image from the drop-down list. Otherwise the default image will appear.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grouplet Height</strong></td>
<td>Use to adjust the height of the tile. The default dimensions of a tile are 1x1, with a maximum of 2 x 2.</td>
</tr>
<tr>
<td><strong>Grouplet Width</strong></td>
<td>Use to adjust the width of the tile. Note: While you can set the value to higher than 2, at runtime, the system enforces a limit of 2.</td>
</tr>
</tbody>
</table>
Note: While you can set the value to higher than 2, at runtime, the system enforces a limit of 2.

**Display In**
Control how the target transaction or content appears once a user presses the tile. Options are:

- Cur Window: target content appears in the current browser window.
- NavBar: applies only when the tile is added to the NavBar. If this is selected when the tile is on the homepage, the tile displays in a modal, when pressed.
- Modal: target content appears in a modal window.
- New Window: target content appears in a new tab in the browser.

**Grouplet Type**
Indicate whether styling is used. The default is No Style for improved performance. When selecting With Style, keep in mind that the system needs to perform additional processing to apply styling.

**Grouplet Timer (Seconds)**
To set an automatic refresh period for dynamic content on a tile, enter the time in seconds. When the timer limit has been reached, the system re-draws the tile so that it displays the current data, such as in the case with chart.

The system enforces a 10 second minimum limit. Any value entered less than 10 seconds will be ignored and treated as 10 seconds. When setting this value, be sure to monitor performance of the page refreshes and adjust accordingly.

The default value of 0, disables any automatic refresh.

**URL Type**
If the tile is a “live” tile, meaning it displays dynamic content, you have these options:

- PeopleSoft Component.
- PeopleSoft Generic URL.
- PeopleSoft Script.

See the following sections for information on each option.

Selecting None results in the system displaying the static image or a default static image if no image is defined on the content reference.

**Same as Main CREF**
If the desired parameters for the dynamic URL type for a component, a generic URL, or a script match those already entered on the Content Reference, General page.
Dynamic Tile Content: PeopleSoft Component

After selecting a URL Type of PeopleSoft Component, the Component Parameters group box appears. Select the menu name, market, and component name, and so on to use for this content reference.

If the parameters required do not differ from those specified for the main content reference on the General page, press Same As Main CREF.

Image: Dynamic Tile Content

This example illustrates the options for configuring dynamic tile content for components.

---

**Node Name**  
Select the node for the target component.

**Menu Name**  
Enter the menu for the target component.

**Market**  
Select the market for the target component.

**Component Name**  
Enter the menu for the component name.

**Page Name**  
Enter the fluid-enabled page name.

**Grouplet ID**  
Reserved for future use.

**Additional Parameters**  
Enter any additional parameters.

Dynamic Tile Content: PeopleSoft Generic URL

After selecting a URL Type of PeopleSoft Generic URL, options related to that URL appear. This usage type can be used in conjunction with a defined node. The result is the concatenation of the selected node's URI text and the portal URL that you enter.
If the parameters required do not differ from those specified for the main content reference on the General page, press Same As Main CREF.

**Image: Dynamic Tile Content: PeopleSoft Generic URL**

This example illustrates the options for configuring dynamic tile content for URLs.

![Dynamic Tile Content: PeopleSoft Generic URL](image)

**Dynamic Tile Content: PeopleSoft Script**

After selecting a URL Type of PeopleSoft Script, the iScript Parameters group box appears.

Select the record (table) and field names, PeopleCode event name, and PeopleCode function name that specifies the iScript to use, plus any additional parameters.
If the parameters required do not differ from those specified for the main content reference on the General page, press Same As Main CREF.

**Image: Dynamic Tile Content: PeopleSoft Script**

This example illustrates the options for configuring dynamic tile content for URLs.