Portrait
Foundation

Enhancing ASP.NET Applications User Guide

Edition 2.0

30 June 2014

Pitney Bowes
Software

Portrait Foundation
Enhancing ASP.NET Applications User Guide

Acknowledgement of trademarks

Other product names, company names, marks, logos and symbols referenced herein may be the trademarks or registered trademarks of their registered owners.

About Portrait Software

Portrait Software is now part of Pitney Bowes Software Inc.

Portrait Software enables organizations to engage with each of their customers as individuals, resulting in improved customer profitability, increased retention, reduced risk, and outstanding customer experiences. This is achieved through a suite of innovative, insight-driven applications which empower organizations to create enduring one-to-one relationships with their customers.

Portrait Software was acquired in July 2010 by Pitney Bowes to build on the broad range of capabilities at Pitney Bowes Software for helping organizations acquire, serve and grow their customer relationships more effectively. The Portrait Customer Interaction Suite combines world leading customer analytics, powerful inbound and outbound campaign management, and best-in-class business process integration to deliver real-time customer interactions that communicate precisely the right message through the right channel, at the right time.

Our 300 + customers include industry-leading organizations in customer-intensive sectors. They include 3, AAA, Bank of Tokyo Mitsubishi, Dell, Fiserv Bank Solutions, Lloyds Banking Group, Merrill Lynch, Nationwide Building Society, RACQ, RAC WA, Telenor, Tesco Bank, T-Mobile, Tryg and US Bank.

Pitney Bowes Software Inc. is a division of Pitney Bowes Inc. (NYSE: PBI).

For more information please visit: http://www.pitneybowes.co.uk/software/

UK

Portrait Software
The Smith Centre
The Fairmile
Henley-on-Thames
Oxfordshire, RG9 6AB, UK

Email: support@portraitsoftware.com
Tel: +44 (0)1491 416778
Fax: +44 (0)1491 416601

America

Portrait Software
125 Summer Street
16th Floor
Boston, MA 02110
USA

Email: support@portraitsoftware.com
Tel: +1 617 457 5200
Fax: +1 617 457 5299

Norway

Portrait Software
Portrait Million Handshakes AS
Maridalsveien. 87
0461 Oslo
Norway

Email: support@portraitsoftware.com
Tel: +47 22 38 91 00
Fax: +47 23 40 94 99

About this document

Purpose of document

Since the Portrait Foundation 4.3 release a lot of work has been done to improve the user experience of Portrait Foundation ASP.Net applications. This document describes the means by which the visual appearance of these applications can be modified.

Intended audience

This document is intended for use by anyone who wishes to change the visual appearance of their Portrait Foundation ASP.Net applications.

Related documents

ASP.Net Application Development User Guide

Generic Desktop 2 Overview

Software release

Portrait Foundation 5.0 or later.

Contents

1 Rich Internet Applications (RIA)

Portrait Foundation 4.3 delivered an optional change to the way that web pages are rendered so that technologies such as AJAX and jQuery can be used in Foundation MSHTA based applications.

The default behaviour is to make a request to the server, such as Start Operation, and the response will be raw HTML that is then written directly into a frame in a tab view or dialog.

This can cause problems with technologies such as AJAX and jQuery because the referenced script files don’t always load as expected and aren’t necessarily available in the DOM when the associated control is initialised.

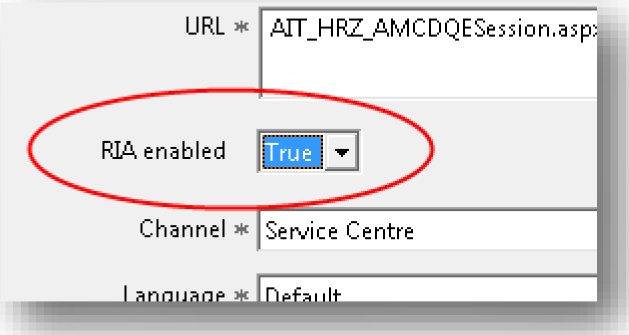
This behaviour can now be changed by redirecting the frame to the page and letting the browser take care of how and when things are loaded.

1.1 Enablement

There are a number of ways this behaviour can be enabled.

1.1.1 Implementation reference

A new property has been added to implementation references for custom interactions, generated interactions and custom views called ‘RIA enabled’.

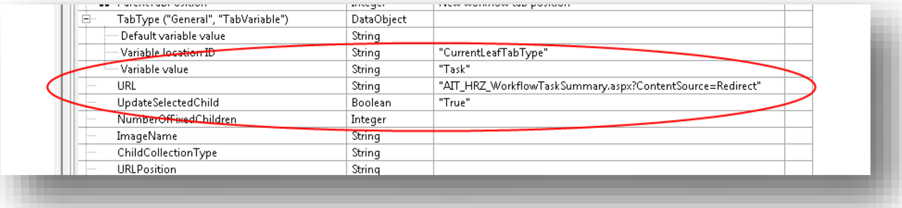


URL *	AIT_HRZ_AMCDQESession.asp
RIA enabled	True
Channel *	Service Centre
Language *	Default

This is a boolean property that defaults to False, but when set to True will cause the page to be loaded in the new way, by redirecting the frame.

1.1.2 Constant model mapping

If a page is being used that doesn’t have an associated Implementation reference, such as the URL property in a tab data object, the value of the URL can be used to trigger the behaviour.



ParentTabPosition	Integer	New workflow tab position
TabType ("General", "TabVariable")	DataObject	
Default variable value	String	
Variable location ID	String	"CurrentLeafTabType"
Variable value	String	"Task"
URL	String	"AIT_HRZ_WorkflowTaskSummary.aspx?ContentSource=Redirect"
UpdateSelectedChild	Boolean	"True"
NumberOfFixedChildren	Integer	
ImageName	String	
ChildCollectionType	String	
URLPosition	String	

By adding a query string parameter called ContentSource with a value of Redirect the page will be rendered by redirecting the frame. For example:

```
MyWebPage.aspx?ContentSource=Redirect
```

1.1.3 Web.config

This new behaviour can be applied across all generated and custom interactions and/or across all custom views by making use of settings in the application web.config file. These settings will override the configuration.

In the AIT.Portrait.Web section, the settings will be seen as:

```
<AIT.Portrait.Web>
    ...
    <add key="DialogContentSource" value="Request"></add>
    <add key="ViewContentSource" value="Redirect"></add>
    ...
</AIT.Portrait.Web>
```

DialogContentSource applies to all generated and custom interactions while ViewContentSource applies to all custom views.

The value of these settings should be either Request or Redirect. Request will render the UI component based on the configuration and Redirect will override the configuration and make all UI components render by redirecting the frame.

By default both settings are set to Request.

1.2 Controls

A number of jQuery plugins have been added to the Foundation UI code that can be used in existing and new custom interactions and generated interactions.

More information about jQuery can be found here.

Although the out-of-the-box Foundation UI does not use AJAX, there is no reason why it can't be used when the implementation reference is RIA enabled.

1.2.1 Custom interactions

To use the jQuery plugins in custom interactions all you need to do is include the relevant script and css files, and then add the correct style name to the control you want the plugin to be applied to.

The includes that you need to use are as follows:

```
<script src="includes/jQuery/jquery-1.5.js" type="text/javascript"></script>
<script src="includes/jQuery.ui.enhancements.js" type="text/javascript"></script>
```

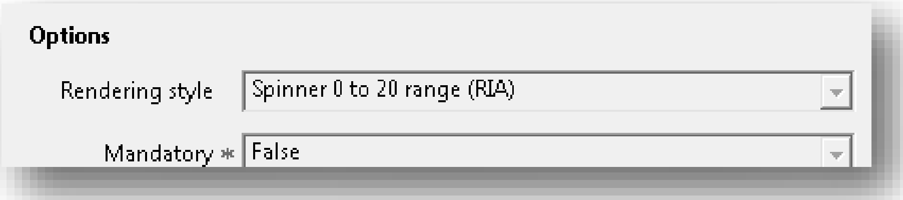
The css class names that can be used are listed below, but as an example the following code extract shows a control with a CssClass value that includes DatePicker10YrRange. This means that the control will be rendered as a Date picker with a selection restricted to a date within 10 years of the current date.

```
<amc:dateedit id="ToDateBox" runat="server" Width="150px" CssClass="InputEdit
DatePicker10YrRange">
```

1.2.2 Generated interactions

To use the jQuery plugins in generated interactions you need to select the relevant Rendering style when configuring the questions.

As an example, the selected style shown below will render a numeric text box as a spinner control that restricts the input data to a number between 0 and 20.



Options

Rendering style: Spinner 0 to 20 range (RIA)

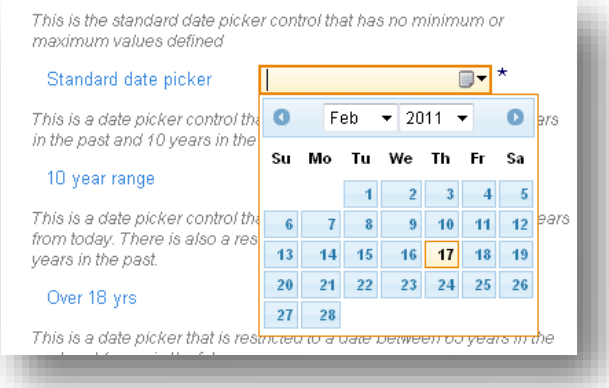
Mandatory *: False

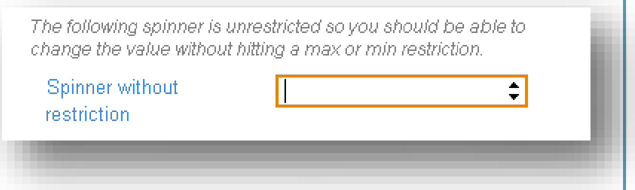
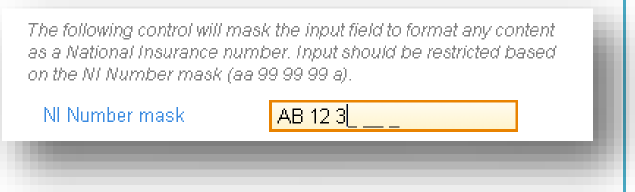
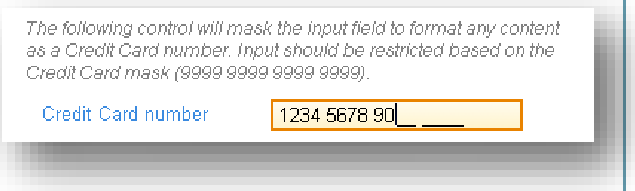
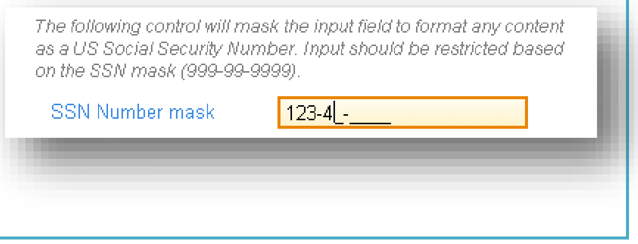
1.2.3 Custom views

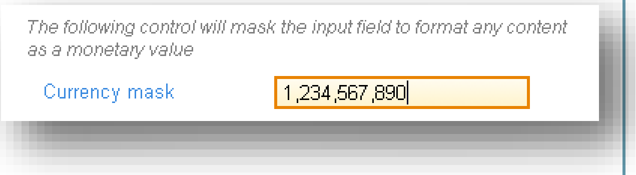
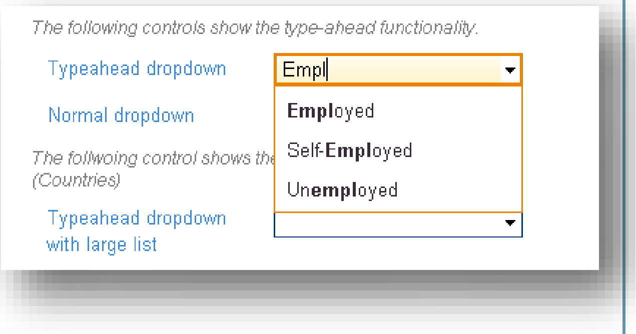
Custom views in Foundation UI aren't typically used for data entry, but the steps detailed for custom interactions above will work for custom views too.

1.2.4 Available jQuery plugins

The available CSS style names and rendering style names are listed below with the associated behaviour of the plugin.

CSS Name	Rendering style name	Description
Date picker plugin (More info can be found here)		
DatePicker	Date picker (RIA)	The Date Picker plugin renders an edit field with a calendar icon that displays a pop-up calendar when clicked.  *This is the standard date picker control that has no minimum or maximum values defined* Standard date picker * *This is a date picker control tha ars in the past and 10 years in the* 10 year range *This is a date picker control tha ears from today. There is also a res years in the past.* Over 18 yrs *This is a date picker that is restricted to a date between 65 years in the*
DatePicker10YrRange	Date picker 10 yr range (RIA)	Looks the same as the DatePicker style but restricts the dates that can be chosen to be within the range + or - 10 years of the current date.
DatePickerOver18Yrs	Date picker over 18 yrs (RIA)	Looks the same as the DatePicker style but restricts the dates that can be chosen to be 18 years or more in the past. The minimum date that can be selected is set to be 118 years in the past.
DatePickerEmploymentStart	Date picker employment start (RIA)	Looks the same as the DatePicker style but restricts the dates that can be chosen to be within 65 years in the past and 1 year in the future.
Spinner plugin (More info can be found here)		

Spinner	Spinner (RIA)	The spinner (also known as a spinbox) plugin renders an edit control that accepts numeric input with up/down arrows that increment/decrement the field value.  *The following spinner is unrestricted so you should be able to change the value without hitting a max or min restriction.* Spinner without restriction
Spinner0To20Range	Spinner 0 to 20 range (RIA)	Looks the same as the Spinner style but restricts the input to be a number between 0 and 20.
Field mask plugins (More info on the masked input plugin can be found here and the masked money plugin here)		
NationalInsuranceNumberMask	National insurance number mask (RIA)	Uses the Mask plugin on an edit control to restrict data entry to be in the format of a UK National Insurance number.  *The following control will mask the input field to format any content as a National Insurance number. Input should be restricted based on the NI Number mask (aa 99 99 99 a).* NI Number mask AB 12 3_ __ _
CreditCardNumberMask	Credit card number mask (RIA)	Uses the Mask plugin on an edit control to restrict data entry to be in the format of a Credit Card number. Note: This mask may not be valid for all credit card numbers as different card providers do use different formats. For example, VISA use 16 numbers and American Express use 15 numbers.  *The following control will mask the input field to format any content as a Credit Card number. Input should be restricted based on the Credit Card mask (9999 9999 9999 9999).* Credit Card number 1234 5678 90__ ____
SocialSecurityNumberMask	Social security number mask (RIA)	Uses the Mask plugin on an edit control to restrict data entry to be in the format of a US Social Security Number.  *The following control will mask the input field to format any content as a US Social Security Number. Input should be restricted based on the SSN mask (999-99-9999).* SSN Number mask 123-4_-____

CurrencyMask	Currency mask (RIA)	Uses the MaskMoney plugin on an edit control to format numeric values as they are typed. Note: Different currency formats will need to be added on an implementation basis.  The following control will mask the input field to format any content as a monetary value Currency mask 1,234,567,890
Type-ahead plugin (This plug-in uses the jQueryUI autocomplete functionality that can be found here)		
TypeAheadDropdownList	Type ahead dropdown list (RIA)	Uses the TypeAhead plugin to add type-ahead functionality to a dropdown list control .  The following controls show the type-ahead functionality. Typeahead dropdown Empl Employed Self-Employed Unemployed Normal dropdown The follwoing control shows th (Countries) Typeahead dropdown with large list
Page refresh plugin		
RefreshPageOnChange	Refresh page on change (RIA)	Causes the page to be refreshed when the onchange event is fired on the control. Anticipated use would be on dropdown list and listbox controls within a Generated Interaction. There is an example of this plugin on the Employment details page, displayed during the Amend personal details operation in the All applications workspace.
RefreshPageOnClick	Refresh page on click (RIA)	Causes the page to be refreshed when the onclick event is fired on the control. Anticipated use would be on checkboxes, radio buttons and checkbox/radio lists within a Generated Interaction.
Validation plugin (More info can be found here) The validation plugin has been wired into the existing Foundation client validation framework so that validation messages look consistent with existing functionality.		
ValidateAsEmailAddress	Validate as email address (RIA)	Validates that the data entered is a valid email address.

ValidateMaxLength254	Validate max length 254 (RIA)	Validates that the data entered is no longer than 254 characters.
ValidateMaxLength2000	Validate max length 2000 (RIA)	Validates that the data entered is no longer than 2000 characters.
UKPostcode	Validate as UK postcode (RIA)	Validates that the data entered is a valid UK postcode.
USZipCode	Validate as US ZIP Code (RIA)	Validates that the data entered is a valid US ZIP code.
AustralianPostalCode	Validate as Australian postal code (RIA)	Validates that the data entered is a valid Australian postal code.
The validation plugin is also used to validate data entered in the datepicker and spinner controls to confirm that the values fall within specified ranges.		

1.2.4.1 Styling

The datepicker and type-ahead dropdown controls come from the jQuery UI library and therefor use the underlying jQuery UI theme capability. The themes are highly configurable and this is explained in more detail here.

The out-of-the-box Foundation client uses the 'Cupertino' theme.

Note: Some jQueryUI styles have been overridden to be consistent with the rest of the Foundation client UI and these can be found in the standard application.css file.

1.2.4.2 Using the plugins

Use of these plugins requires some common sense.

For example, the datepicker or spinner plugin should only be used on an edit control. If you apply the plugin to something like a label control you are likely to face problems.

Plugins that cause the page to refresh when an event is fired may also cause performance problems when used incorrectly. For example, if you want a page to refresh when selection in a dropdown list changes, use the onchange variant of the refresh plugin. If you choose the onclick version you will generate significantly more post-backs to the server.

1.3 Updated pages

The following pages have been updated to use the jQuery plugins mentioned above.

- Pick work search criteria (AIT_HRZ_PickWorkSearchCriteria.aspx)
- Generated interaction (AIT_HRZ_AMCDQESession.aspx)

Generic Desktop (GenericDesktop.aspx) has been updated to use a jQuery plugin that darkens the background when a dialog is active. This feature uses a plugin called BlockUI and more info can be found here.

Some pages have been updated to remove the use of the Microsoft IE Web Controls (v1.0).

The following now use the jQueryUI tab control.

- Workflow summary (AIT_HRZ_WorkFlowTaskSummary.aspx)

- Party Workflow summary (AIT_HRZ_TaskDetails.aspx)
- Case Workflow summary (AIT_HRZ_TaskDetails.aspx)

The following now use the jQueryUI accordion control.

- Party Contact details (AIT_HRZ_ContactDetails.aspx)

2 Cascading Style Sheets (CSS)

The entire colour styling for Portrait Foundation applications is controlled through the use of a Cascading Style Sheets (CSS). The main style sheet used by the .Net application framework is called **application.css** and contains colour, font, style and layout information that is applied to different sections of the application as identified by classes or specific tags. For the Foundation 4.3 release we consolidated the styles in this file and removed all redundant entries. Some styles were also moved into **application.css** from **framework.css**. Details of the exact changes were included in the Portrait Foundation 4.3 release notes.

2.1 Restyling Foundation ASP.Net Applications

There are two sets of assets which influence the colours of the Foundation Application framework:

- The application.css style sheet which is located in the ***<application>*\includes** directory for the relevant application.
- A number of images that are used throughout the application which are located in the ***<application>*\images** directory for the relevant application.

In order to change the colours you can manually examine the colour references in the style sheet file or use an online tool such as CSS Color Editor. This tool allows you to upload a CSS file, for which it then provides a list of the colours used. You can either manually key in new colours or choose replacements from a colour picker dialog. You can leave any colours you do not want to change blank. The tool will then produce a new CSS file which can be saved with the appropriate colour changes. This process can be applied in a few minutes to give visual feedback on a chosen colour scheme.

The images that ship with a Portrait Foundation application can either be replaced with customer specific images that have been designed for the application, or can be customised using a graphics package such as Paint.Net, Photoshop or GIMP. Often the images contain single blocks of colour which can be easily replaced using the colour dropper and paint fill tools of these applications. Anything more complex than this typically requires a design resource as opposed to a development resource.

2.2 Choosing a colour scheme

More than likely the customer will already have a colour scheme in mind or present already within their corporate environment. Usually their web homepage will give a good idea of a colour scheme. From this it is possible to generate a colour scheme diagram which shows the palette of colours to be used and their hex values for inclusion in the CSS file or use in graphics packages. Any specific icons or images can also be included in this diagram.

If a colour scheme is not available or if you want to check on the appropriateness of the colours then there are a number of tools available which can assist. An example would be Color Scheme Designer which is an online tool that provides set of complementary colours and various tools which can check for issues in the colour scheme relating to visual conditions such as colour blindness.

When undertaking a 're-branding' exercise it is important to take into consideration the needs of colour-deficient users. For further information Q42 offers a colour blindness check or Vischeck have a colour vision simulator.

2.3 Application styles

In the Foundation 4.3 release a number of changes were made to application.css to remove redundant styles and make it easier to re-brand Portrait Foundation ASP.NET applications. Details of the exact changes can be found in the 4.3 release notes.

2.3.1 Button styles (MS theme)

All Portrait Foundation web pages have been updated to include a new HTML Meta tag to enable Microsoft Windows styling. When adding this tag, the **head** DHTML object must be set to "runat" server.

```
<head runat="server">
    …
    <meta http-equiv="MSThemeCompatible" content="Yes" />
```

To get Microsoft styled (curved) buttons in the new fresher white colour scheme, the styling has been removed from the button classes in application.css.

NB: These Microsoft themes only apply to newer Windows operating systems like Windows 7.

2.3.2 Important styles

To get started, below is a summary of the key styles that are used by all Portrait Foundation .Net applications.

Area	Style	Description
Logon & Select role	.ApplicationBackground	Background style for logon and select role custom interactions.
Logon & Select role	.DialogPanel	Style for the panel that contains the controls for logon and select role.
Logon & Select role	#VersionFooter	Controls location of application version number.
Generic desktop	#GenericDesktopBody	Background style for the application desktop.
Generic desktop	#LogoImage	Controls the logo image and its location.
Generic desktop	#AppVersion	Controls location of application version number.
Generic desktop	#divAgentName	Controls location of the agents name.
Generic desktop	.TS_TL	Unselected left tab image and text style.
Generic desktop	.TS_TL_SEL	Selected left tab image and text style.

Generic desktop	.TS_TC_SEL_OVR	Middle tab image and text style when hovering over a selected tab.
Generic desktop	.TS_TC_OVR	Middle tab image and text style when hovering over an unselected tab.
Generic desktop	.TS_TC	Unselected middle tab image and text style.
Generic desktop	.TS_TC_SEL	Selected middle tab image and text style.
Generic desktop	.TS_TR	Unselected right tab image and text style.
Generic desktop	.TS_TR_SEL	Selected right tab image and text style.
Generic desktop	.TS_TC_OVR	Middle tab image and text style when hovering over an unselected tab.
Generic desktop	.primaryBody	Background styling for single level desktop tab pages.
Generic desktop	.tabActive	Background styling for multi-level desktop tab pages.
Generic desktop	.BVOPBody	Background styling for the business view.
Dialogs	.TitleBar	Title bar image and text style used in all dialogs.
Dialogs	.dialogBody	Background styling for all Custom and Generated Interactions.
Dialogs	.giTopSection .giMiddleSection .giBottomSection	Controls the styles in the different sections of a Generated Interaction.
Dialogs	.dialogTopSection .dialogMiddleSection .dialogBottomSection	Controls the styles in the different sections of a Custom Interaction when using tables.
Dialogs	.dialogPageTextSpace	Controls row spacing between controls and buttons.
Dialogs	.ciTopSection .ciMiddleSection	Controls the styles in the different sections of a Custom Interaction when using divs.

	.ciBottomSection .ciButtonSectionLeft .ciButtonSectionRight .ciButtonSpacer	
Dialogs	.button .buttonDisabled	Controls the styling for all buttons in the application.
Dialogs	.MBInformation .MBPleasewait .MBQuestion .MBRetry .MBSystemerror .MBWarning	Message box image styles.
Controls	.gridHead .gridStatus	CustomGrid styles for the top and bottom sections of the grid.
Controls	.fxgHead .fxgStatus	FlexGrid styles for the top and bottom sections of the grid.
Controls	.MandatoryFieldEmpty .MandatoryFieldPopulated	Image style for mandatory control validation.

For examples of the changes required to achieve a different colour scheme compare the two application style sheets that are provided in the Portrait Platform "User Interface Utilities" package. See application.css and blue_application.css in the Configuration Repository under

Portrait_Platform\UserInterfaceUtilities_install\Release\WebServer\WebApps\ContactCentre.Net\includes

2.4 Using different colour schemes

The Portrait Foundation "All Applications" Implementation install now provides two different colour schemes, the original blue one and a new fresher white colour scheme.

To swap between colour schemes in your implementation, simply rename "application.css" in

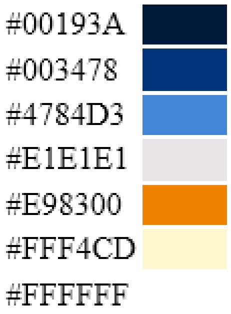
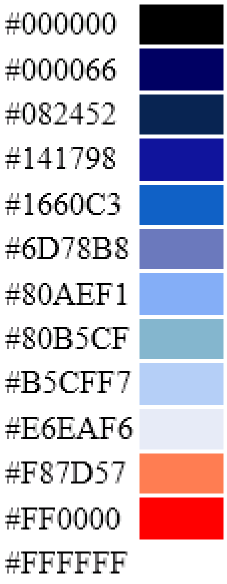
C:\Program Files\PST\Portrait Implementation\Systems\ <*system_name*>\Web Pages\Application Centre\includes

to "application_orig.css" and rename the other file (blue_application.css) to "application.css".

The new colour scheme will be picked up the next time the application is launched. If your changes are not picked up simply clear the IE browser cache of temporary internet files.

The main differences between these 2 files are the use of different colours and images. The new **application.css** uses a reduced set of basic colours in comparison to the original blue colour scheme. You can use tools like WinDiff (packaged with Visual Studio) to compare these files and see these differences.

Customers wishing to apply their own colour scheme should take a copy of the relevant application.css that best matches their own branding. Then edit this stylesheet replacing the colours and images where appropriate. The table below should be used to help identify the main colours that may need replacing.

White (application.css)	Blue (blue_application.css)
 #00193A ■ #003478 ■ #4784D3 ■ #E1E1E1 ■ #E98300 ■ #FFF4CD ■ #FFFFFF	 #000000 ■ #000066 ■ #082452 ■ #141798 ■ #1660C3 ■ #6D78B8 ■ #80AEF1 ■ #80B5CF ■ #B5CFF7 ■ #E6EAF6 ■ #F87D57 ■ #FF0000 ■ #FFFFFF

3 Enabling Application Resizing

Since the Portrait Foundation 3.0 release it has been possible to resize ASP.NET applications. This section describes the web page changes required to make use of this feature. The **Generic Desktop 2** UI layout uses a different resizing mechanism so implementations no longer need to make changes to their View pages.

3.1 HTA application settings

Edit the HTA file that is used to launch your Portrait Foundation .Net application. At the top of the file edit the following HTA:APPLICATION settings.

- BORDER="thick" (use “none” when disabling resizing)
- CAPTION="yes" (use “no” when disabling resizing)

It’s important to note that SYSMENU is set to “no”**.** This removes the option for the user to close the session without going through log off. Unfortunately turning this off prevents the display of the maximize and minimize buttons as well so these cannot be present.

Go to the JavaScript section where the StartPage() function is defined and at the top of that section make sure useResizing is set to true. To disable resizing set this value to false.

```
var useResizing = true;
```

The application uses a cookie to store the values of the width and the height of the desktop window. The initial values should be set in the HTA (directly after useResizing).

```
var windowWidth = 1024;
var windowHeight = 768;
```

On application start, if the cookie doesn’t exist on client side then the application window is displayed with default size (see above), and a cookie is created with this value. If the cookie is already present, the application window is displayed with size as the cookie value, which would be window size before application closing on last visit.

3.2 Font resizing

The whole resizing concept is based around setting all the font sizes as relative units (in percentages) and then, based on the window size, set up a base font size (in points or pixels) and the browser will take care of the rest.

During development of this feature, the application.css style sheet was updated so that all font sizes used **em**’s rather than **pt**’s. A second style sheet (resize.application.css) was added that gets dynamically loaded when resizing is turned on. It contains relative units (in percentages) for most CSS classes used in Portrait Foundation. The sizes have been set up comparing their previous sizes (in points, pixels or ems).

For resizing to work correctly all project specific View (tab) pages should be updated to include ResizeGeneric.js and use CSS for positioning elements. In your custom Tab pages, the .aspx files for each **AIT.Portrait.Web.UI.TabPage** needs the following script included (see AIT_HRZ_EngagementHistory as an example).

```
<script type="text/javascript" src="/portrait_client/includes/ResizeGeneric.js"></script>
```

4 Menu Icons

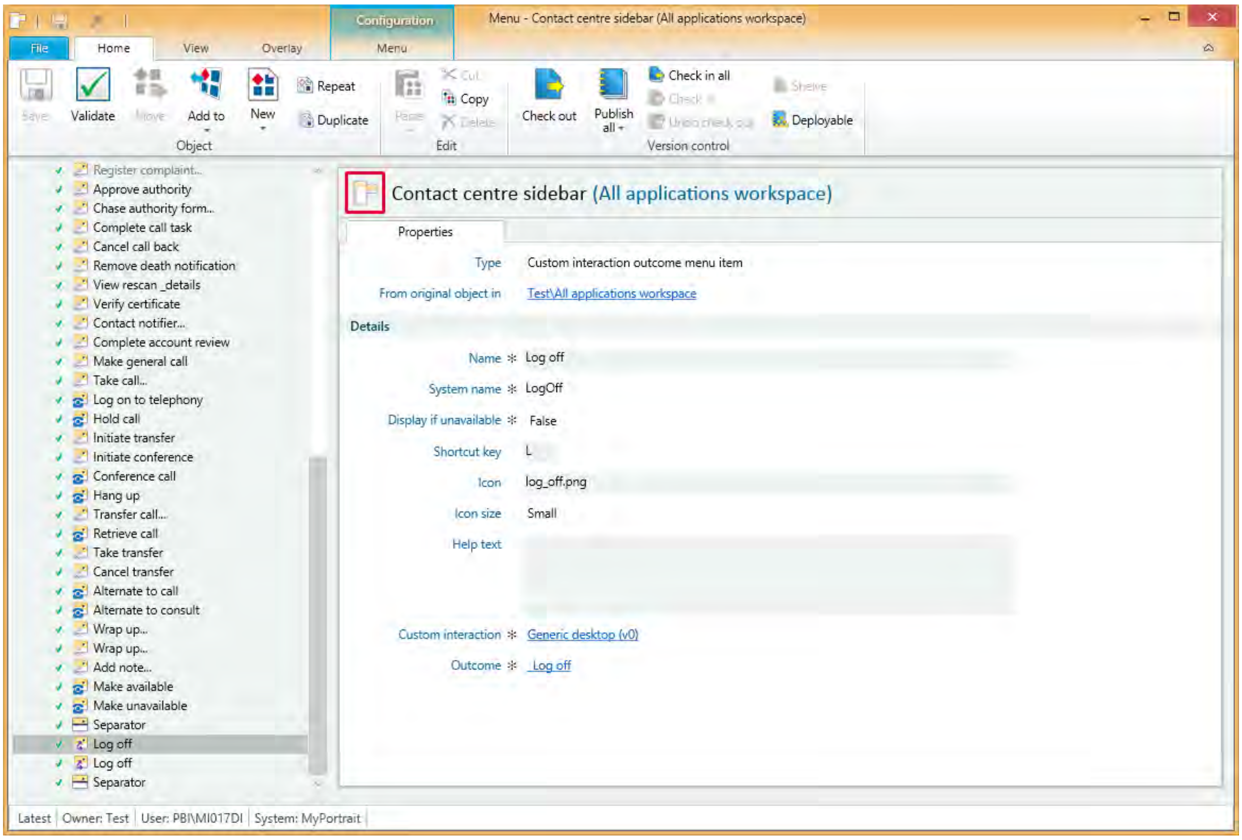
The menu sidebar within Portrait Foundation applications is used to display the options a user can select. It is possible to add icons to these options to enhance the user experience. Icons can be associated with menu items in the Configuration Suite so that they show to the left of the text. These icons are fully configurable and once the system is initially setup do not require a development resource to modify. Using this feature a client can achieve a consistent look and feel in their Portrait Foundation applications compared with other applications in their organisation.

The Contact Centre application included as part of the All Applications workspace contains 3 examples Lock desktop, Help and Log off.

4.1 Configuration

Specifying the icons for use in the application is done against the Menu Item in the Configuration Suite. Open the relevant package or workspace that contains the menu item definition and add the file name in the Icon field.

Open the Menu item object in the Configuration Suite. On the properties tab of each menu item you can specify an icon name and size. This should be the name and the extension of the image that you wish to use. The size is property is only used by the **Generic Desktop 2** UI layout.



Contact centre sidebar (All applications workspace)

Properties

Type: Custom interaction outcome menu item

From original object in: Test\All applications workspace

Details

Name *: Log off

System name *: LogOff

Display if unavailable *: False

Shortcut key: L

Icon: log_off.png

Icon size: Small

Help text:

Custom interaction *: Generic desktop (v0)

Outcome *: Log off

Latest | Owner: Test | User: PBI\MI017DI | System: MyPortrait

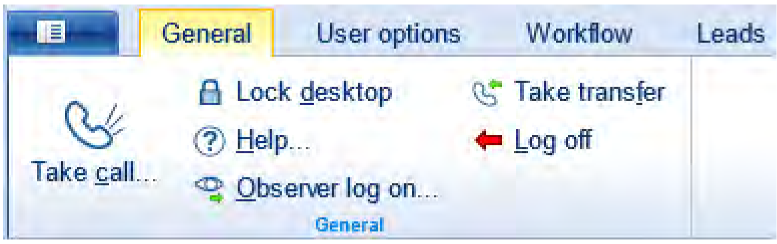
4.2 Icons

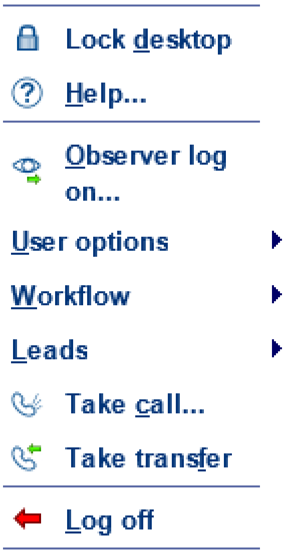
The Foundation Application framework will try and load the icon image from the ***<application>*/images** directory. So long as the icon exists in the appropriate directory and is spelled correctly it should then appear in the application the next time it is executed.

Icons should be roughly 20x20 pixels in dimension; however it is always best to perform a visual check once the icon has been included. There are a number of images provided out of the box.

The **Generic Desktop 2** UI layout uses a menu Ribbon with support for small and large icons; so different icon sizes are required. For more details please refer to the Generic Desktop 2 Overview document.

Examples


General
User options
Workflow
Leads
Take call...
Lock desktop
Help...
Observer log on...
Take transfer
Log off
General


Lock desktop
Help...
Observer log on...
User options
Workflow
Leads
Take call...
Take transfer
Log off