ORACLE®

FUSION APPLICATIONS
DEVELOPER RELATIONS

Using Page Composer with Fusion Cloud Applications

Tips, Tricks and Best Practices

Follow: @fadevrel

Read: blogs.oracle.com/fadevrel

Watch: youtube.com/FADeveloperRelations

Discuss: bit.ly/fadevrelforum

ORACLE®

Contents

Using Page Composer

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Introduction

This guide provides additional tips, tricks, and advice on using Page Composer to tailor your Fusion Cloud Applications (SaaS). It has been compiled from information and experiences from multiple teams within Oracle and our partners.

This document is not a training guide or a general description of the features of Fusion Applications or Page Composer. For this please refer to the Extending the Applications product documentation.

Please use the table of contents to identify areas of interest and browse through the suggestions therein. The Fusion Applications Developer Relations team intend to update this document with new recommendations and advice regularly. If you have anything to suggest, please contact us.

The content is based on testing with Release 8, although most parts will apply for later releases also. If there is anything release-specific it will be mentioned.

The content here is provided as-is, and for use at your own risk.

Page Editing and Property Security

Each-and-every page (.jspx) sets its own properties that determine if it can be edited in page composer.

These properties (*customization allowed* and *isPersonalizableInComposer*) are not available to change at run-time.

It should be noted that the Global Administration menu item option with the name "Customize [the current page]" is also linked to these page properties, and not to any security role or privilege.

In addition to this, there are roles related to the whole Administration menu itself, and details are given in this blog post.

In addition, each J2EE Application has a pe_ext.xml file which controls which component properties are exposed in Page Composer. This file is not available to edit in Cloud environments, and to eliminate potential inconsistencies it will be standardized into one global pe_ext.xml file.

Sandboxes: Before You Publish

It is recommended that before you publish a sandbox that you export it and keep a local copy. Enter the sandbox details page and click "Download All". This forms a backup and record of the customization, and can then be re-imported and adjusted if problems occur.

Once you have downloaded your sandbox check the file size is not 0 bytes and that it contains the related XML data, such as below.
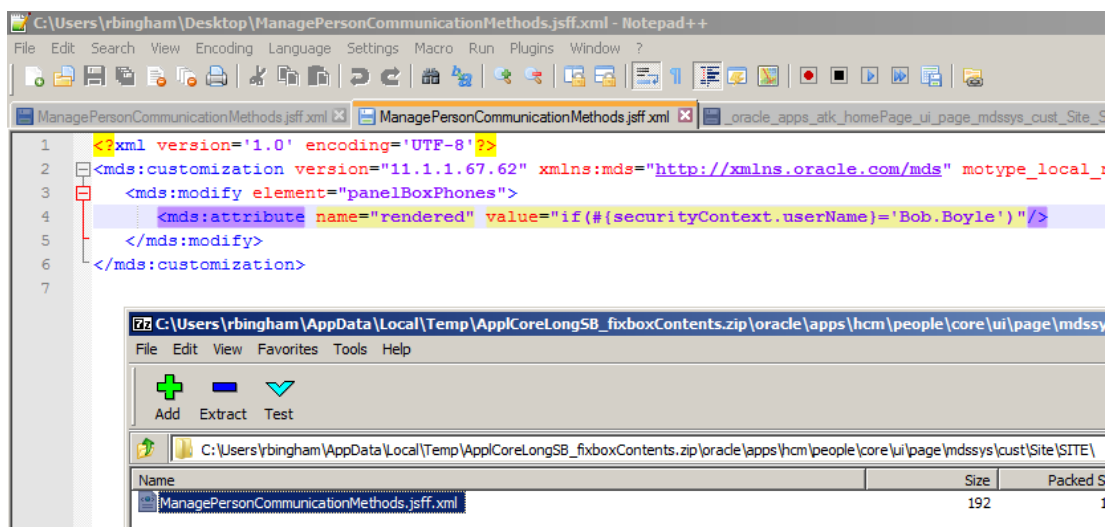
In addition, if your sandbox does not have too many MDS artefact files in, consider taking screenshots to capture the file and path.

Creating Sandboxes: Keep It Simple

Don't make a single sandbox hold too many customizations. Distribute them across several sandboxes based on a logical commonality, such as a set of pages that support a single business task.

When starting a new customization, even when testing, always create a new sandbox and do not reuse an existing one used for another purpose. Upon publishing, old customizations in your re-used sandbox may cause conflicts with changes made since they were created.

Wherever possible, publish your sandbox to the same release and patch bundle level in which it was created. This will also avoid potential conflicts.
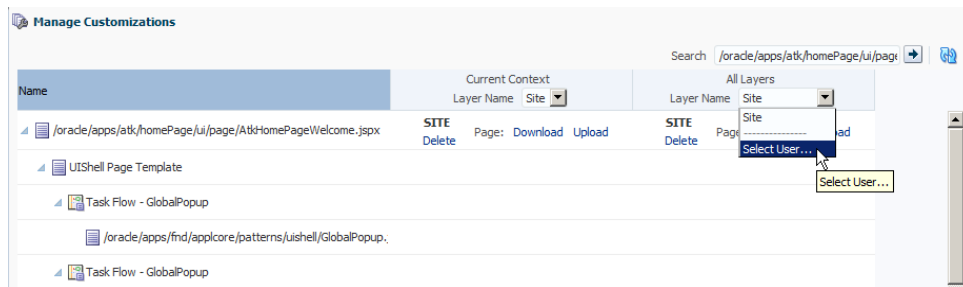
Finding Problem Customizations

There are several options for finding and removing problematic customizations:

- Use Source View mode to find accidentally hidden fields.

- You can only download/import customizations to your sandbox for those done at the USER or SITE level. You can only use the Customization Migration features for those done at the SITE level.

- If they are still in a Sandbox (unpublished) then you can either delete the related metadata files from the sandbox contents screen (shown below), or delete the entire sandbox. It may not always be easy to identify the metadata rows related to a single customization however, in which case deleting the whole sandbox is recommended. You can also limit the metadata file shown in the sandbox by picking the layer/user.
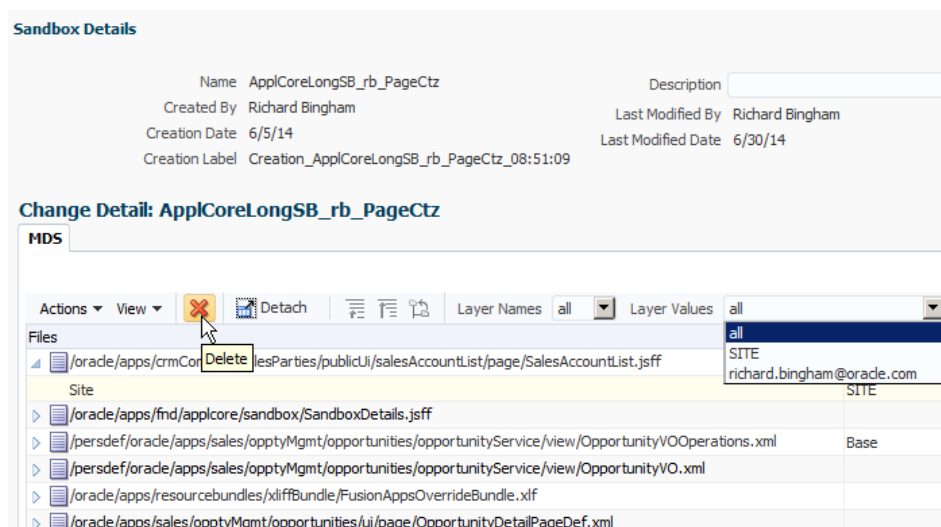
- If you have published the sandbox, then you can use the Manage Customizations popup (Administration menu) to identify the customization and delete it. This page has the option to search by user (from the layer dropdown) which can be helpful.

Using Page Composer

- Sometimes it is hard to get to the page/component where the problem exists. This is desirable as it is easier to open the Manage Customizations page with the right context (current page). If it's not possible then to manually determine the full path/name of the related jspx page for use in the search

  o Use the screenshots taken of your original sandbox to determine the path and artifact.

  o Open your downloaded sandbox zip and find the related artifact/file, noting the path.

- When you cannot access the customization using the Manage Customizations popup, the final option is to use Weblogic Scripting Tool (WLST) commands (i.e. *deleteMetadata*) to remove the metadata from the repository. This option is only available to on-premise system administrators or Oracle Cloud Operations.

Track Your Customizations

It is strongly recommended to keep a record of the customizations made to each environment, with dates, contacts, details, sandbox data, the downloaded files, and so on. This helps administrators understand the existing system implementation and improves the support for future changes.

Moving Customizations

It is always recommended to make Customizations on a test environment first, and then migrate them to the production environment. In an emergency a customization made to production should then be made to test, to keep them in-sync (required for the Customization Migration feature).

To avoid rekeying customizations there are three options available:

1) Sandboxes can be downloaded and re-imported. If the import button is missing from the Manage Sandboxes page, check Profile Option "FND_DISABLE_SANDBOX_IMPORT" is not set to Y.

Using Page Composer

2) For on-premises environments, you can use Enterprise Manager (Fusion Applications Control) to download and upload a set of customizations. In addition in the Manage Customizations page use the "Download Customizations for All Layers" option to download published customizations, however be aware that at this time there is no equivalent wholesale upload process for this file. As such this option can be used for recording and troubleshooting customizations.

3) The Customization Migration feature (Navigator menu, under Customization) supports moving many types of SITE layer published customizations, including menus and all those made by Page Composer. This is the recommended method of moving published customizations between environments.

Note: Flexfields and setup data is migrated using Functional Setup Manager.

Preventing a Lost Worklist

To prevent users from accidently deleting the Worklist region on the homepage using the X in the customization/personalization screens, you can do the following:

1. Log in to the welcome dashboard using a customization user. From the user menu, choose Administration and Customize Workarea Pages

2. Select the layer, either User or Site depending on your requirements.

3. In the "Worklist: Notifications and Approvals" region, click the edit wrench button just before the "X" button.

4. In the properties popup, go to Display Options and uncheck the "Allow Remove" checkbox. Click OK and save. Now users cannot remove this component from the page.

Using Page Composer
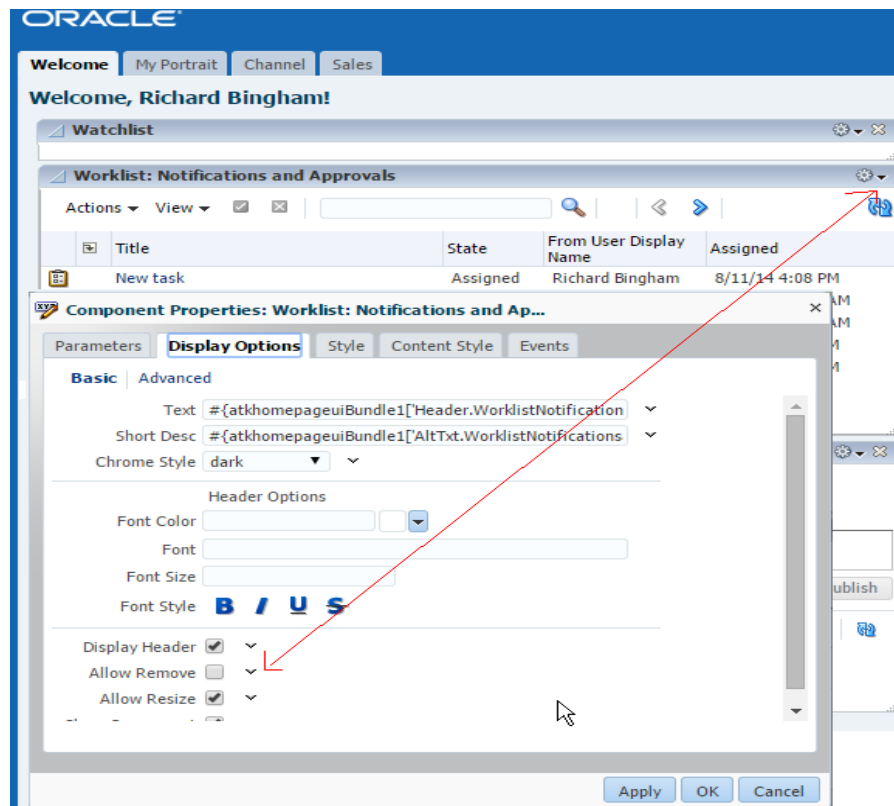
Debugging

Download each specific customization from your sandbox and open the related XML file from the zip.

Check its contents for extra characters like encoded html and special characters. Look for problems in the XML structure that may suggest corruption and check for any conflicting modification entries.

If problems remain, simply delete it from the sandbox and redo your customization again.

Upload In Manage Customizations

This method forms an invasive customization as it allows the modification of the XML page source stored in MDS.

It goes beyond what is possible via Page Composer and as such the results cannot be supported by Oracle.

9

Using Page Composer

Customizing Field Labels

Always use 'Select Text Resource' option as it allows for different values based on the user language. These must be entered separately and apply only for the current session language.

The Select Text Resource popup can be used in the following ways:

1) First use the search to try to find an existing text string that matches what you need. The search covers the KEY and the Display Value fields.

2) You can create new labels EITHER using the Create button above the search table OR by simply entering the Key and Display Values at the top of the table. This dialog looks like the table is a set of child records however it is NOT, each key only has one display value per language.

Using Page Composer
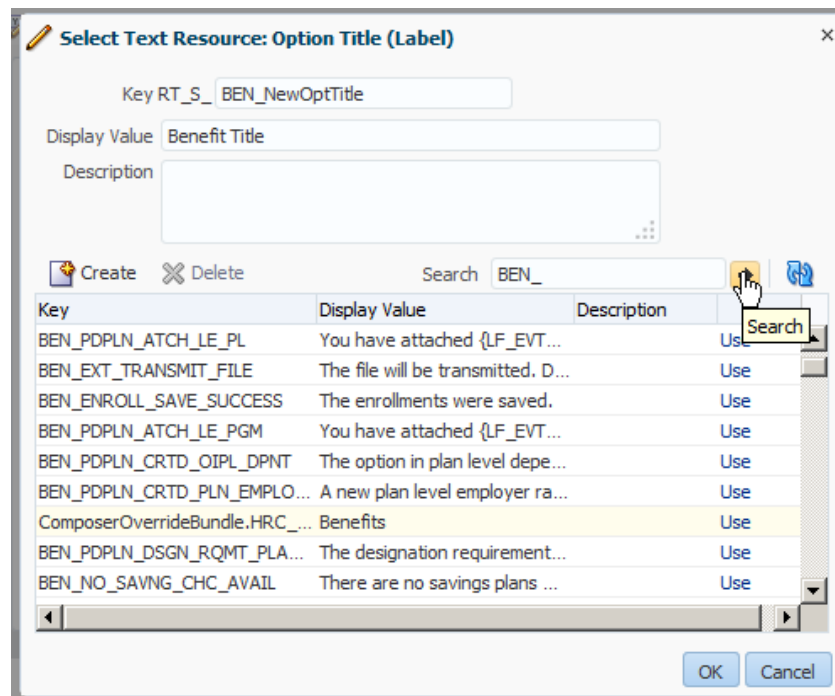
Customizing Fields That Are Tricky To Select/Find

Sometimes components are very hard to select on the user interface in Page Composer; however there are a few tricks you can use to access the component and its properties.

- Zoom (ctrl+mouse-scroll) to make the page as large as possible to select the component required.

- Use Design Mode (from the Views menu) to navigate around in full screen mode and then switch to Source Mode once you have the component selected. Use Design view again to preview the change just made and for navigating to the part of a page that needs customizing.

- For pop-ups, select the area around the intended component (fields, buttons or linked images) to access the popup node in the hierarchy (or its taskflow).

- When you want to navigate/scroll down a page to make a customization - use the mouse wheel - as any clicks assume you want to edit that taskflow and switching between design-source views reloads the page taking you back to the top.

- Select parent container components related to panels and regions as these often hold the child components and taskflows that you wish to edit. Example shown below where the fields in the popup from the red dot is only accessible by finding the Edit Task Flow for the related panelLabelAndMessage and not from selecting the outputText or the red dot itself.
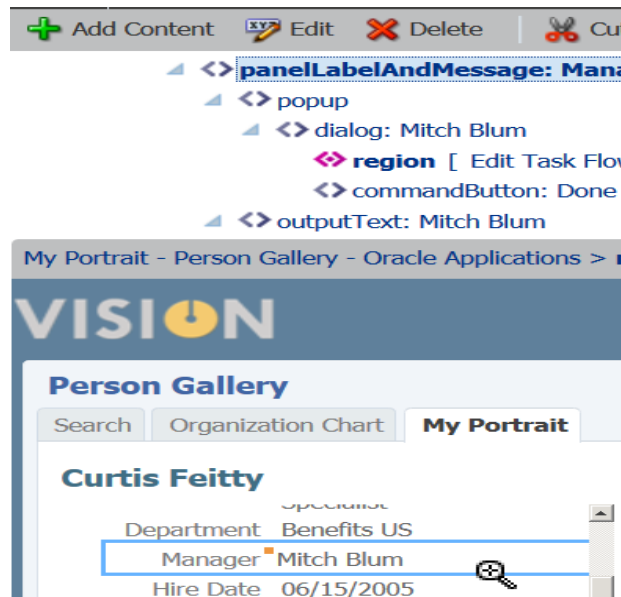
Source view is available for all non-CRM application pages, and for all editing of the UI Shell template.

For non-CRM applications the Page Composer view menu also offers a "Source Position" option which just moves the display around the screen.

This can be handy to move from the default top when the page is complex and you need to see more of the hierarchy.

Using Page Composer
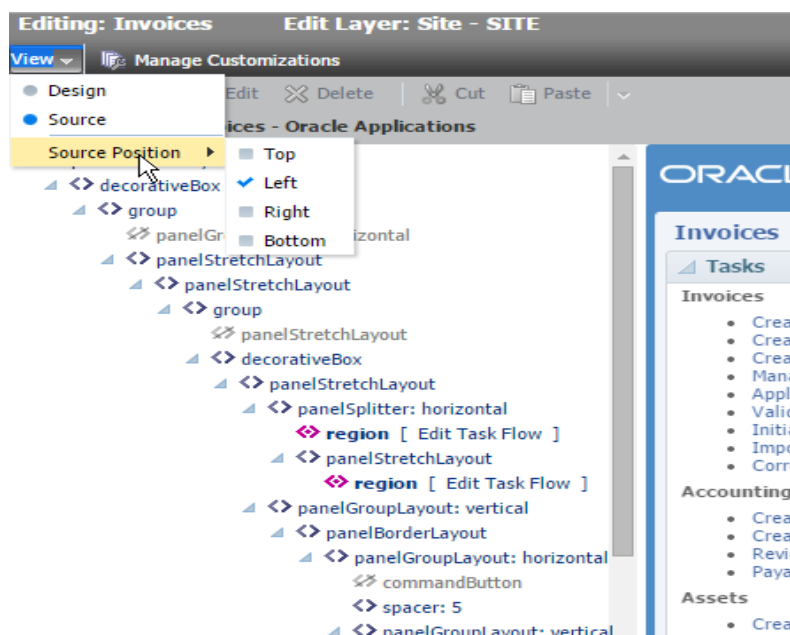
Creating Translations

If you customize text/label components then you need to always do the English language change first, then change the users session language (under Preferences), then make the same update for the new language using the same Key value.

Setting the field property in English

Then in a French user session, with same Key value.

All language customizations get stored in the following files. In the last example, you can replace _fr with any other language translation. This information can be useful to know when looking at your sandbox metadata content.

- Base: /oracle/apps/resourcebundles/xliffBundle/FusionAppsOverrideBundle.xlf

- English: /oracle/apps/resourcebundles/xliffBundle/FusionAppsOverrideBundle_en.xlf

- French: /oracle/apps/resourcebundles/xliffBundle/FusionAppsOverrideBundle_fr.xlf

Missing iFrame Content

When you use page composer to add a region to a dashboard page and if the content is hosted from a server that uses another security mechanism than the host (or none) then all modern browsers will invoke a safety feature that blocks the content.

Unfortunately it's not possible to avoid this entirely, however if this occurs users can click the shield icon in the browser address bar and choose to allow the content. After this the browser should remember that this content is permitted, however any changes may invoke the security feature again.

You can read more about this here.

Customizing the Navigator Menu

As a popular way to invoke additional applications, the navigator menu setup is done using Functional Setup Manager (from 'Manage Menu Customizations' task), and has some fields that require some explanation. See an example here.

- Focus View ID = the page (known as a view) internal ID.

- Web Application = the J2EE application where the page is running. Fortunately it's an LOV.

- Secured Resource Name = the artefact to which the iDM security applies.

- Application Stripe = the iDM domain to uses for applying security (such as hcm, crm or fscm)

- Destination = used when you're not pointing at a Fusion Applications page. You can use the JSON Web Token (JWT) user security token with this also.

The best method of obtaining the values for Fusion Applications pages, copy them from an existing menu item that points to the same page.

# Hiding Components and Regions

Although obvious, make sure any conditions you write that hide a component from view still allow someone access to it. For example if you specify a field to only show for a certain job role make sure that there is a user who has the job role and has customization privileges so to access it. If you cannot access it through the UI then even a customization user may not be able to select it to perform further customization.

Branding Tips

Use the Customize Global Page Template option to set the top level menu items and Desktop application logo icon (under the Style tab and the "Background Image" property. Use the "Other CSS" property to enforce the recommended size using "background-size:119px 25px;".

In the Simplified UI you can use the "Logo URL" field to select a remote image for your application, as there is no support for directly uploading new images. It is possible, however, to upload an image to the internal Webcenter Content Server and use the URL provided here.

At this time you cannot specify the branding for other than SITE level (e.g. change the branding based on Business Unit).
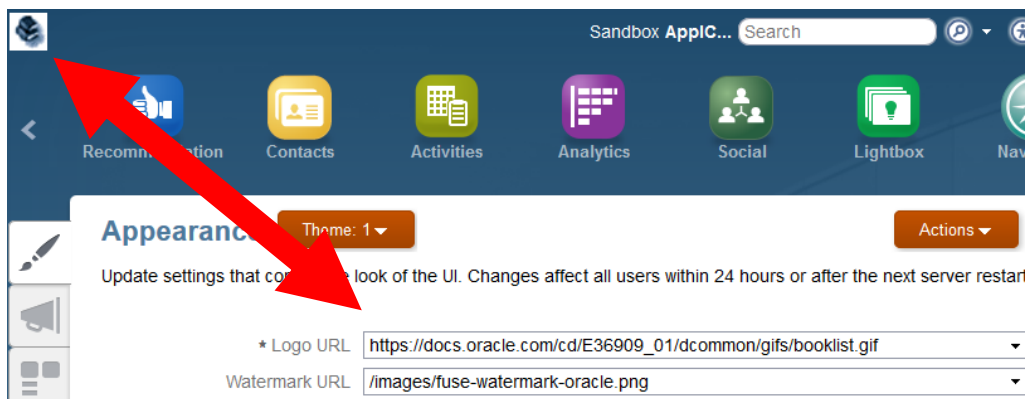
From Release 9 you can choose from a family of icons for the pages related to new Custom Objects. This is defined in the Custom Object screens in application composer.

There are plans to increase the number of icons available in future releases, however due to the variety of sizing requirements for images on various pages it is not possible to add your own icons at this time.

Always take care and use a sandbox when making general branding customizations, since problems may prevent users from accessing the links required to make corrections.

Using Page Composer

Page Composer Properties

The following table simplified which properties are most commonly used for customization. Whilst other properties may exist their use may not be supported by individual pages. See here for more details.

Property	Description	Example
Short Desc	Used for the tool tip text.	Any text
Visible	If it is shown on the page.	Use true|false or EL expression
Show Component	Is it shown on the page (same as visible).	Use true|false or EL expression
Required	Will show a validation message show if left blank.	Use true|false or EL expression
Show Required	Adds a star to the field label but doesn't implicitly make it required.	Use true|false or EL expression
Read Only	Shows the stored value but doesn't allow any UI edits.	Use true|false or EL expression
Columns / Rows	The displayed size of a field.	10 (character width)
Value	The actual value shown on the screen.	Use true|false or EL expression

Note: not all field types have all properties.

Flexfields

You first configure your flexfields using Setup and Maintenance tasks "Manage Descriptive Flexfields" and "Manage Extensible Flexfields". Once configured the next step is to expose them on the pages that support them, using Page Composer. For more information about adding flexfields, refer to chapter 5 of the extensibility guide, "Using Flexfields for Custom Attributes."

16

Checkbox and LOV Values

There are several types of LOVs commonly used, including *selectOneChoice*, *inputListOfValues* and *inputComboboxListOfValues*. The first has a *Value* attribute which allows you to determine the current data shown on the screen, whereas the others use an internal list object to hold the actual values and this is not readily accessible from page composer.

For checkboxes (*selectBooleanCheckbox*) the Value attribute can be get/set to "true" for checked and "false" for unchecked.

Customizing Pop-ups

Not all pop-ups are available to customize, however some are accessible using Page Composer Design View. The challenge is editing the right taskflow. Here are some tips.

- Sometimes you can get to the popup by editing the Child Components of the button that launches the popup.

- For pop-ups launched from toolbars on tables, edit the taskflow for the region that contains the table, not the table or button itself, as shown below
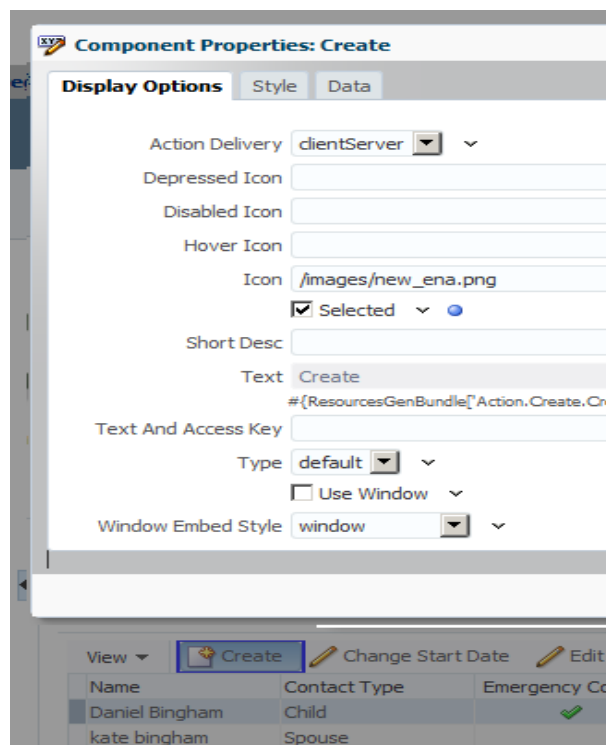
Using Page Composer
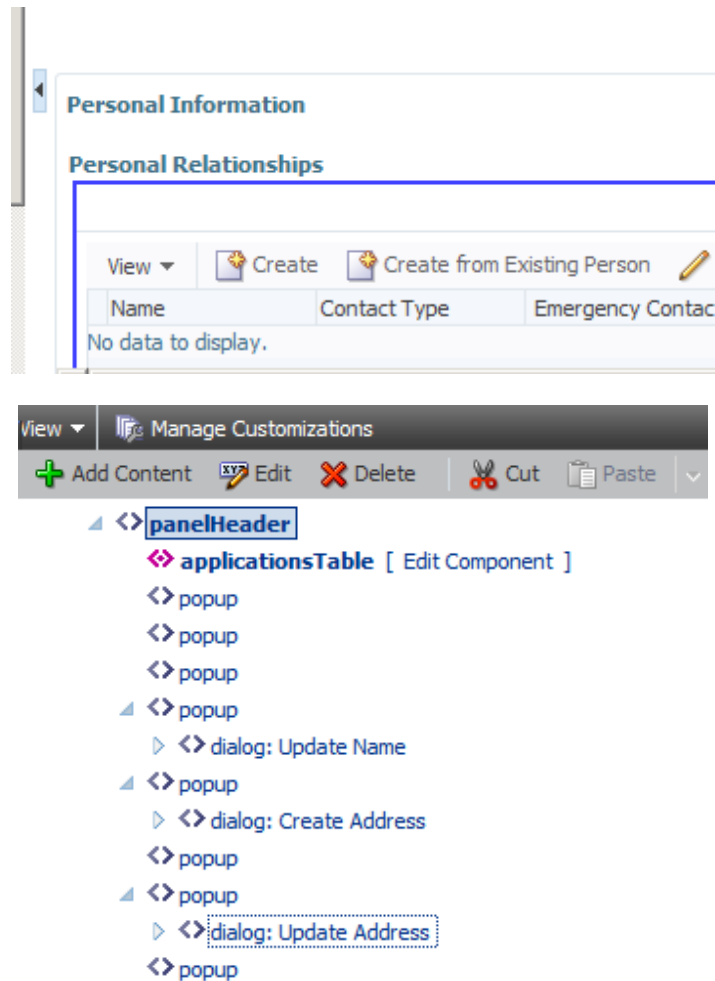
- Again as shown above popups may expose the dialog and its fields, full child taskflows, or actually very little like those at the top. You may have to look at the properties of each "popup" node to understand what they do (e.g. from looking at the "Popup Fetch Listener" property).

- For yes/no popups you can change the order of the child components (values yes/no) which reverses which is selected by default.

Tables and Buttons – Set Default

You can set which columns are shown in a table by default by simply editing the columns on the table when in page composer.

Select the columns you wish to show, and then save by existing Page Composer. These become the defaults shown.

You can set the "Selected" property of buttons to promote the default usage (has focus). You can also use EL to conditionally decide which button is selected.

UI Text Customizations

If you want to make widespread (all pages) text changes, or cannot get to customize some text using Page Composer this tool (Navigator > Tools > Customization > User Interface Text) is the right tool to try.

Using Page Composer
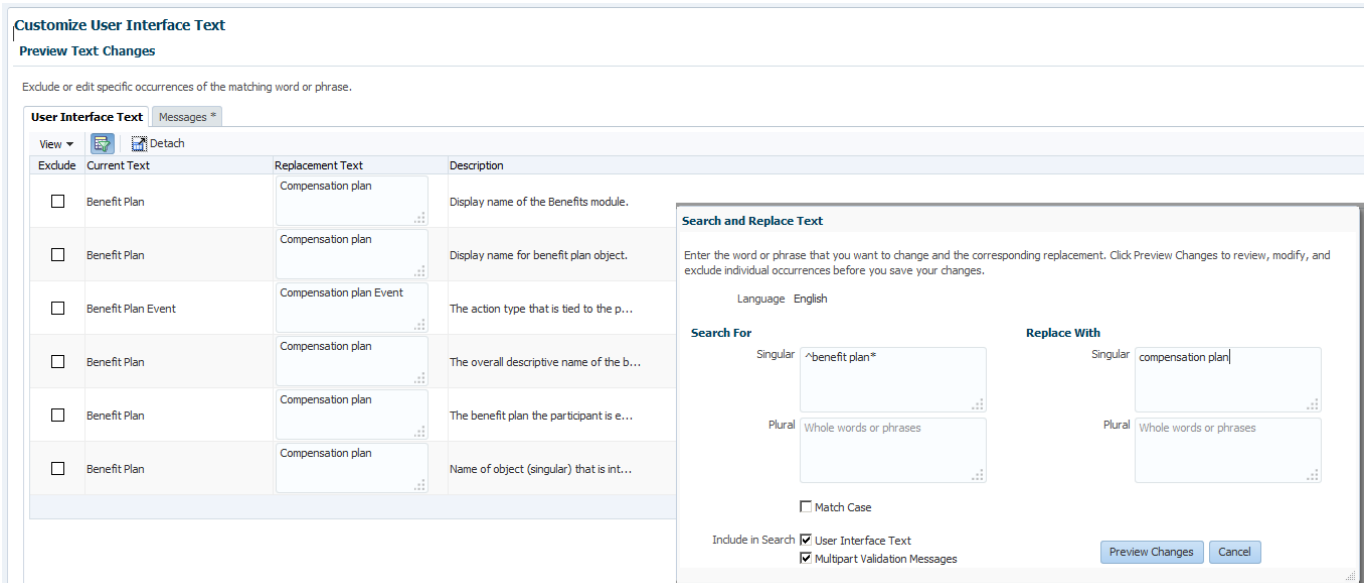
Here are some usage tips:

- Be as detailed as possible in searching for existing text strings, it can take a lot of time paging through many results and may be error prone.

- Ideally indentify the pages where the replacement should occur and test it first. Sometimes text strings vary only slightly between pages.

- Use this tool to search and replace labels and titles in pages, and the text of system response messages (confirmation, information, warning etc). The latter are known as Multipart Validation Messages. Don't forget to exclude texts/messages that you do not want changed.

- Perform the User Interface Text tasks again in another session language to customize each installed language in turn.

- Check that your "replace with" text has the correct capitalization/case for all usages.

- When searching for Multipart Validation Messages to customize, use the exact text as shown on the popup dialog, and be aware of runtime token data that is not stored in the message itself, such as names, titles, and ID's.

- Without "match case" selected, the replacement will capitalize the first character of the replacement string.

Using Page Composer

- The following Regular Expression-like characters can be used to help pinpoint texts and messages:

    o ^ - specifies the beginning of a line.

    o * - Specifies any sequence of following characters.

    o $ - specified the end of a line

    o . - matches any single alphanumeric space character

    o () - matches the word or words in the parenthesis

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

White Paper Title
November 2014
Author: Richard Bingham

CONNECT WITH US

blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com