

Customization Reduction with Forms and OA Framework Personalizations

Susan Behn

Infosemantics

Introduction

Oracle provides two different personalization tools. OA framework personalizations are for the new OAF forms that were introduced first in the self-service applications such as iExpense and iProcurement and are now utilized within many other modules. Professional Forms personalizations are for the forms built using the Forms Builder 6i/10g tool. These are many of the core application forms such as invoice workbench and other workbench type forms. The 6i version is for E-Business (EBS) version 11i and the 10g version is for EBS version 12. Professional form personalizations work the same in both 6i and 10g.

The Oracle E-Business Suite is known for its flexibility and configurability. Most Oracle Applications users are familiar with using profile options, setups, key and descriptive flexfields, folder tools, etc. to achieve this flexibility. Users can consider personalizations as another type of configuration. Prior to 11i, professional forms could be personalized through the use of a special library named CUSTOM.pll. The declarative forms personalizations can replace most of the code in CUSTOM.pll. The purpose of this paper is to show examples, which can potentially replace forms customizations, extensions and CUSTOM.pll with personalizations.

Form Personalizations and OA Framework Personalizations allow EBS users to meet specific business needs such as protecting sensitive data, validations, modified value lists and more. Specific examples and recommendations implemented at multiple client sites will be presented for both types of personalizations. Some of these examples can be implemented in 11i and will migrate to R12 allowing for simplification of the upgrade process. Keep in mind however that for OAF personalizations, the same personalization is often made on multiple pages. As Oracle adds pages to workstreams, particularly in HRMS, additional work will be needed to update new pages.

The multiple business benefits of using personalizations are shown in Figure 1. These will be identified in the specific examples as well.

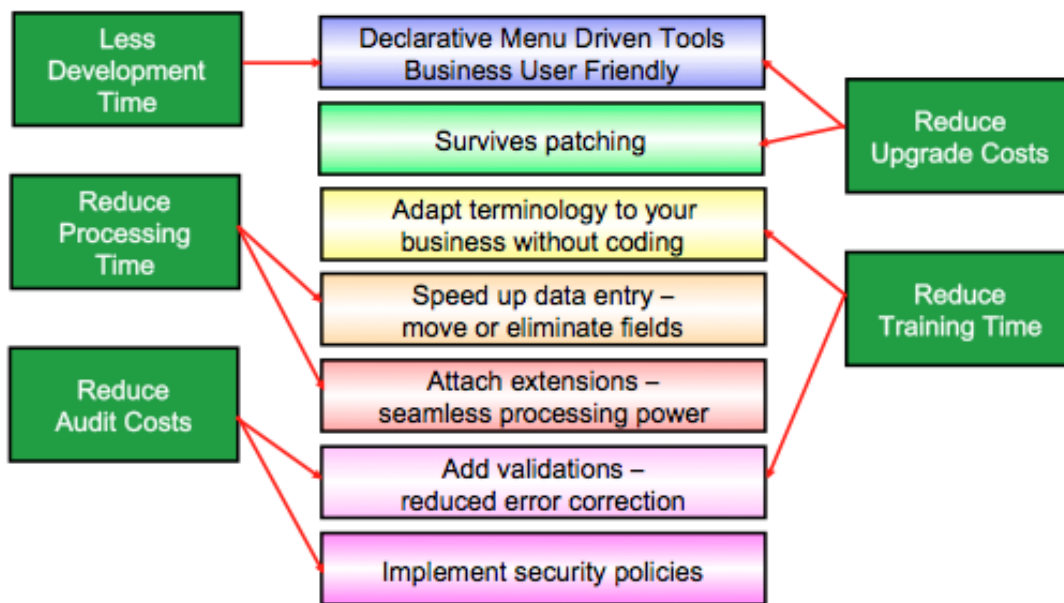


Figure 1

Definitions

Functionality of Oracle Applications can be altered using a variety of methods. As the Applications mature, some of the terminology has matured as well. For the purposed of this paper, these methods are defined as follows:

Extensions

Extensions are additional code or objects added to the E-Business suite that add to the existing functionality. Examples include new forms created from TEMPLATE.FMB, new reports, interfaces and other new objects such as views used for custom reporting.

Customizations

Customizations are changes to forms, reports or any other objects delivered with the E-Business Suite. Direct modifications of forms, reports, packages or other objects is not supported or protected during patching. Customization of the E-Business Suite is not recommended and should be used only as a last resort.

Personalizations

Personalizations are changes to existing forms using tools provided by Oracle to alter the appearance or behavior of the form or function. Upgrades or patches do generally not impact these changes, and the use of personalizations is supported by Oracle with some limitations. Personalizations for 6i/10g forms are implemented using CUSTOM.pll or the new forms personalization tools provided by Oracle beginning in release 11.5.10.

Forms 6i/10g Personalization Overview

Forms personalizations declaratively alter the behavior of 6i/10g forms delivered with the E-Business Suite. Although the declarative nature of this tool does not require the use of pl/sql, it is generally recommended that the user have some understanding of pl/sql and forms in order to understand the impact of the changes. Forms personalizations are effective immediately as opposed to CUSTOM.pll, which must be compiled.

Most changes traditionally done using CUSTOM.pll can be accomplished using forms personalization; however, users may still choose to utilize CUSTOM.pll for complex modifications. For the same event, the declarative forms personalizations will fire prior to the CUSTOM.pll for the same event.

Profile Options Impacting Forms Personalization

Hide Diagnostics menu entry – This profile options must be set to “No.” Setting this profile options to “Yes” hides the diagnostics menu.

Utilities: Diagnostics – If this profile options is set to “Yes” the apps password is not required to use diagnostic features including forms personalization and examine. These are very dangerous utilities. Set this profile option to “No” in production environments.

Getting Started

First access the form you wish to personalize. The following examples will utilize the Supplier form. After opening the form, go to Help → Diagnostics → Custom Code → Personalize to navigate to the forms personalization tool as shown in Figure 2. The forms personalization form is shown below in Figure 3.

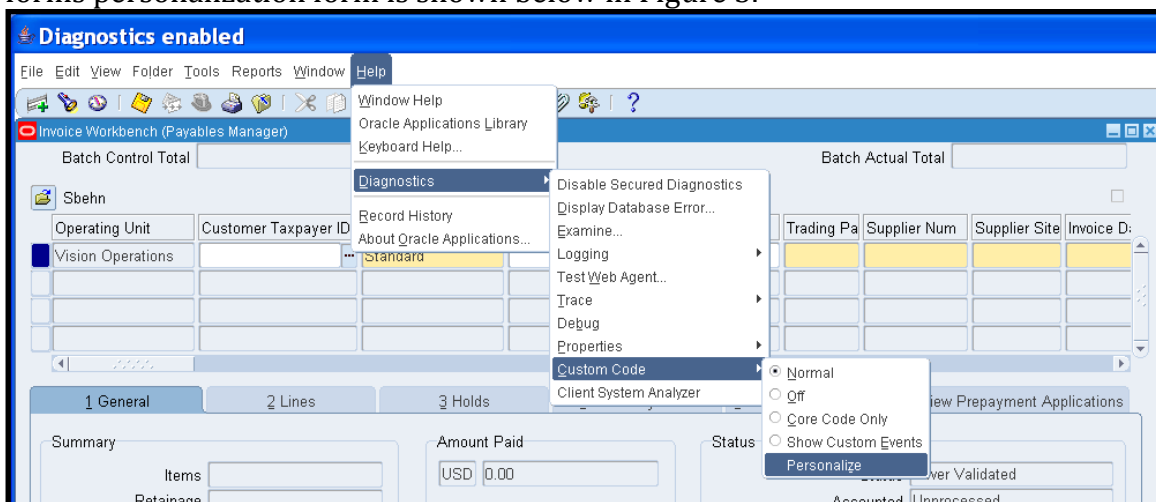


Figure 2

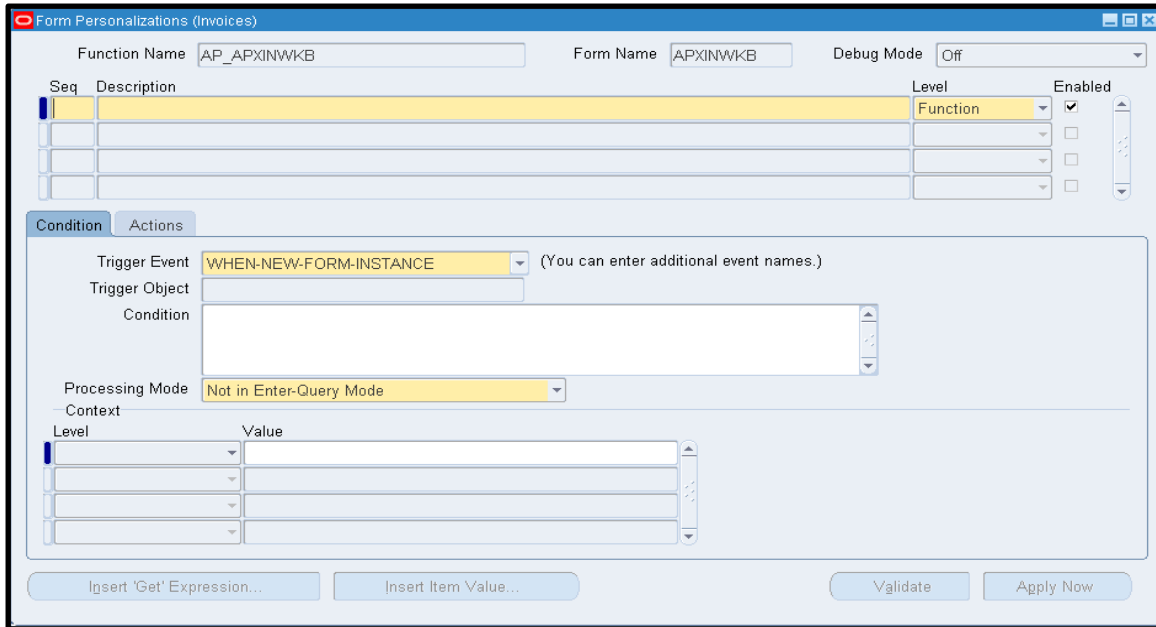


Figure 3

Header

The function name and form name will default to the current function and form. These values may not be changed. The debug mode can be set to off, Step-by-Step or Show Debug Messages. Step-by-Step will display a pop-up window showing events impacted by the rule. Show Debug Messages will show messages where the type = debug.

Sequence numbers are set between 1 and 100 and can be reused. Rules will run in sequence.

Description is free form entry.

Level can be set to form or function. For example, you may want to create a personalization that applies only to the query-only function for the Supplier form. Checking the Enabled check box enables personalizations.

Condition

Trigger Event

The following page provides some guidelines on which standard trigger event to use for specific types of personalizations. The events included in this list are the events Oracle generally provides in all forms. However, test to make sure the trigger event is actually firing. To find events that are firing, go to Help → Diagnostics → Custom Code → Show Custom Events. As you navigate through the form, a pop-up window will display the trigger events as they fire. You may also choose to use any other event fired in the form, however, Oracle does not guarantee that other events will be retained in patches, so these personalizations may not be protected.

- WHEN-NEW-FORM-INSTANCE

- Security rules
- Navigation rules
- Visual attributes
- WHEN-NEW-BLOCK-INSTANCE
 - Same as WHEN-NEW-FORM-INSTANCE
 - Message rules
- WHEN-NEW-RECORD-INSTANCE
 - Default values
- WHEN-NEW-ITEM-INSTANCE
 - Message rules
 - Default values dependent on entry of another item
- WHEN-VALIDATE-RECORD
 - Populate hidden fields
 - Additional validations
- MENU_n
 - Populate tools menu (MENU1-15) (CU1 patch)
- SPECIAL_n
 - Populate tools menu (SPECIAL 1-15)
 - Populate reports menu (SPECIAL 16-30)
 - Populate actions menu (SPECIAL 31-45)
- ZOOM – recommend using MENU_n or SPECIAL_n rather than zoom
- KEY-Fn

The trigger event LOV contains the standard trigger events above, however, since Oracle allows you to use other non-standard events, the trigger event field does not validate from the list of values. You can choose a value from the list, but if you are trying to use a standard event and type it incorrectly, your personalization will not work.

Use MENU1-15 before SPECIAL1-45 to avoid conflict with any pre-existing menu items seeded by Oracle.

Trigger Object

The trigger object may be required depending on the trigger event. If the list of values is available when the cursor is in the trigger object field, than the trigger object is required. For example, the following events will require a block name.

- WHEN-NEW-BLOCK-INSTANCE
- WHEN-NEW-RECORD-INSTANCE
- WHEN-VALIDATE-RECORD

The WHEN-NEW-ITEM-INSTANCE trigger event will require a block and field name separated by a period. Other non-standard events may also require a trigger object depending on the event. In this case, the absence of a list of values does not necessarily mean the trigger object is not required.

Condition

The condition is an optional sql code fragment used to limit the scope of the personalization. For example, you can limit a message to appear only certain times of the year based on the system date. You may reference the value of a field in the current record in the format :blockname.fieldname.

Processing Mode

The processing mode determines when this personalization is applicable. Options are Only in Enter-Query Mode, Not in Enter-Query Mode or Both.

Context

The context determines who the personalization applies to. Multiple rows are allowed and will be processed as “and” statements. Applicable levels are site, responsibility, and user. Industry exists in the list of values, but is reserved for future use. If the context is null, the personalization will apply to all.

TIP: When testing new personalizations, set the context so the personalization only applies to your own user id.

Actions

The Actions tab shows the actions on the left and details about the action on the right. There are 4 action types – property, message, builtin, and menu, which will be described in more detail in the examples. The details on the right side of the tab page will change depending on the type of action. Figure 4 shows the tab page for an action type of property.

The screenshot shows a software interface with two tabs: 'Condition' and 'Actions'. The 'Actions' tab is active. On the left, there is a table with columns: Seq, Type, Description, Language, and Enabled. The first row is selected, showing a 'Property' action type, an empty description, 'All' language, and an enabled checkbox. A dropdown menu is open over the 'Type' column, showing options: Property, Message, Builtin, and Menu. To the right of the table is a configuration panel for the selected action. It includes a 'Select By Text...' button, a dropdown for 'Object Type' (set to 'Item'), a text field for 'Target Object', a text field for 'Property Name' (set to 'PROMPT_TEXT'), and a text area for 'Value'. Below these fields is a 'Get Value' button. At the bottom of the interface are four buttons: 'Insert 'Get' Expression...', 'Insert Item Value...', 'Validate', and 'Apply Now'.

Figure 4

Actions have a non-unique sequence number from 1 to 100. The description is free form entry. You may choose a language if your site is enabled for multiple languages, and you may choose to enable or disable an action using the check box.

Property action type

Property action types require an object type, target object, property name and value. The properties for object types listed below are available for personalization.

- Item – size, position, visibility, insert, update, delete, value, case restriction, LOV, format mask and more
- Window – size, position, title, visibility
- Block – size, position, default where clause and order by, navigation, insert, update, delete
- Tab Page – enabled, displayed, label
- Canvas – size, position, visibility
- Radio Button – size, position, label, prompt, visibility
- View – size, position
- Global Variable – initial value, value
- Local Variable – value
- LOV – size, position, group name

A list of values is available for the object type, target object and property name. Use the Select by Text button to select the target object by prompt name. Use the Get Value button to retrieve the current value. Enter the value for the property selected. Values starting with an = sign will be evaluated at run time. Values that do not start with an = sign are taken as a literal.

Message action type

Message action types require a message type and message. Messages can be displayed at trigger events other than WHEN-NEW-FORM-INSTANCE. The following list describes the different message types.

- Show – Informational Message
- Hint – Appear on status bar
- Error – Requires user response
- Debug – Only displays if debug mode is set to Show Debug Messages
- Warn – Informational message with caution symbol

Builtin action type

The following builtin types are available:

- Launch SRS Form – launches a concurrent program
- Launch a Function – launches another form function
- Launch a URL – launches any URL
- DO_KEY – execute a form builtin
- Execute a Procedure – execute any procedure using syntax exactly as you would in pl/sql code
- GO_ITEM – navigate to a specific item
- GO_BLOCK – navigate to a specific block
- FORMS_DDL – issue dynamic sql statements
- RAISE FORM_TRIGGER_FAILURE
- EXECUTE_TRIGGER – call a trigger
- SYNCHRONIZE – synchronize form on client and middle tiers
- Call Custom Library – call a specific event you have coded directly in CUSTOM.pll
- Create Record Group from Query – creates a record group dynamically

Parameters for builtins vary based on the builtin type. Several examples are provided later.

Menu action type

Using the menu action type only displays the menu in the appropriate location on the tool bar. (The builtin to execute the functionality behind this menu item is covered in an example.) Menus are typically established during the WHEN-NEW-FORM-INSTANCE or WHEN-NEW-BLOCK-INSTANCE event. Select a menu entry and give it a label. You may also choose an icon and choose which blocks should include this menu item. The Add Block button will provide a list of blocks. To select multiple blocks separate block names with a comma or use the Add Block button multiple times to select multiple blocks. Leaving the Enabled in Block(s) field null will enable the menu item in all blocks.

Professional Forms 6i/10g Examples

Example 1 – Display a Message

The reason for this personalization is to display a reminder message for users who were repeatedly making the same mistake. During year in close, the final period is open longer resulting in mistakes because the user would forget to change the period when making a prior period journal entry. Simply displaying a reminder message during the first few days of the New Year reduced these errors resulting in a reduced load for IT support which equated to cost savings.. Figure 5 shows the configuration for this personalization. This personalization was set to occur as the user entered the header name field due to specific code that was executed after the WHEN-NEW-FORM-INSTANCE and WHEN-NEW-BLOCK-INSTANCE events. Figure 6 shows the results of this personalization.

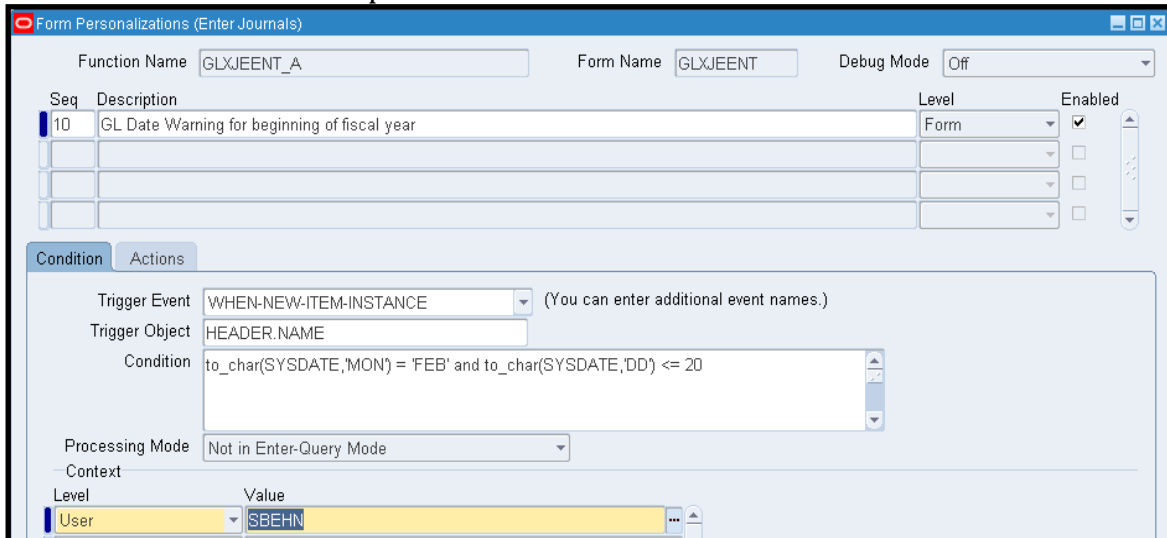


Figure 5

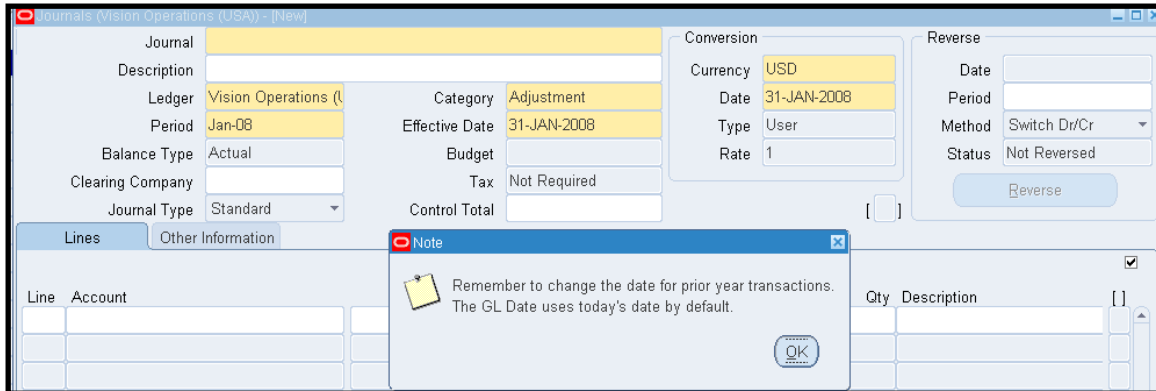


Figure 6

Example 2 – Data Access Restrictions

In this example, the client wanted to restrict users from entering or retrieving their own invoice that was in the payables system to pay expense reports. This provided better audit control and theft prevention. This example does require some minimal sql skills. This personalization was created at the WHEN-NEW-BLOCK-INSTANCE event because this is where the default clause for the query is executed. Figure 7 shows the header and condition tabs.

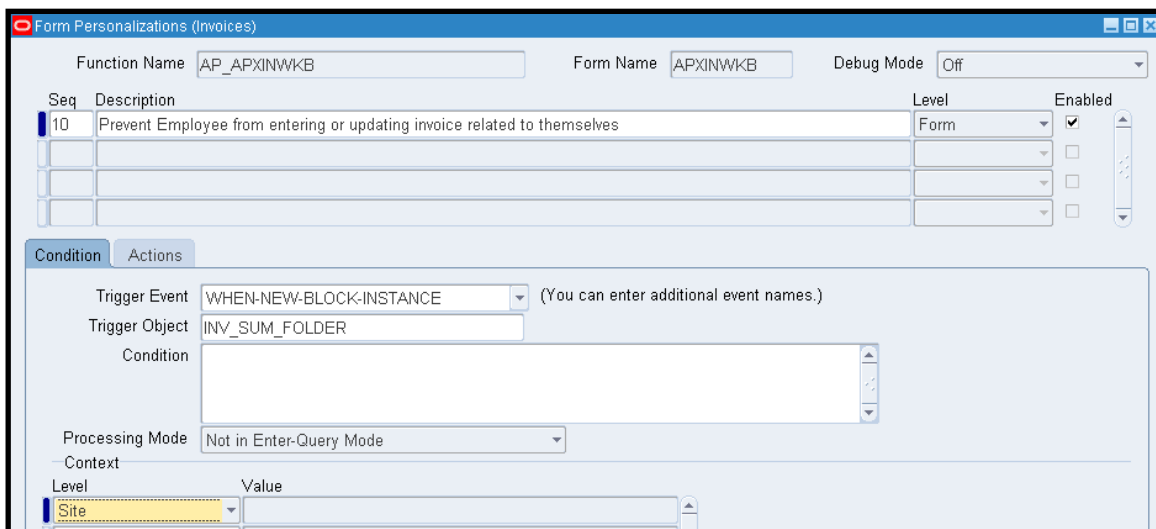
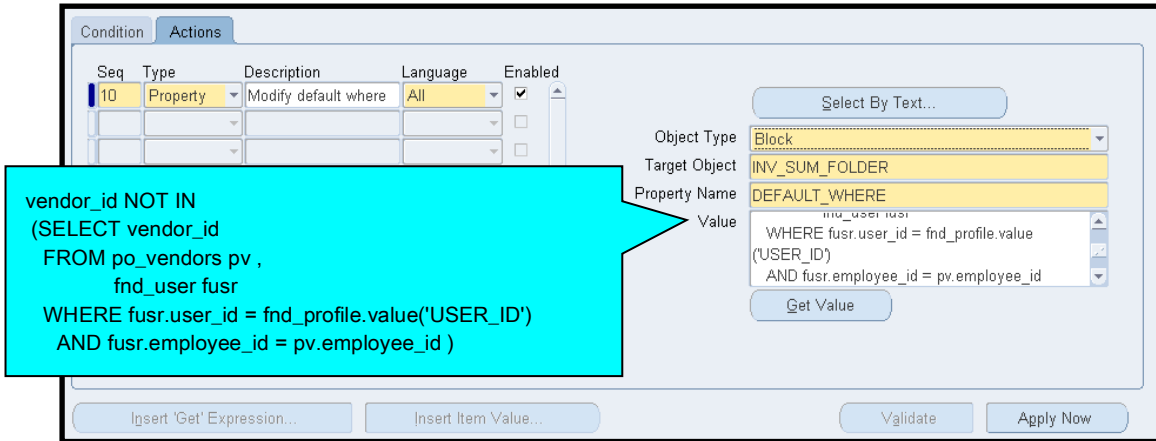


Figure 7

Change the default where clause for this block to eliminate invoices where the employee id in the supplier record matches the employee id in the user record for the user that is logged on. See Figure 8 for the specific SQL and other details on the action tab.



Example 3 – Using Menu Entries

In this example, the client wanted to create a menu entry to display the invoice id to improve the efficiency of workflow troubleshooting for the invoice approval workflow. There are two steps require to complete this personalization. First, establish the menu entry. Second, identify what should happen when the user clicks on the menu entry.

Establish the menu event at the WHEN-NEW-FORM-INSTANCE event by choosing a menu entry number and identifying the label. As shown in Figure 8, this only establishes the menu entry, not what happens when the menu entry is clicked.

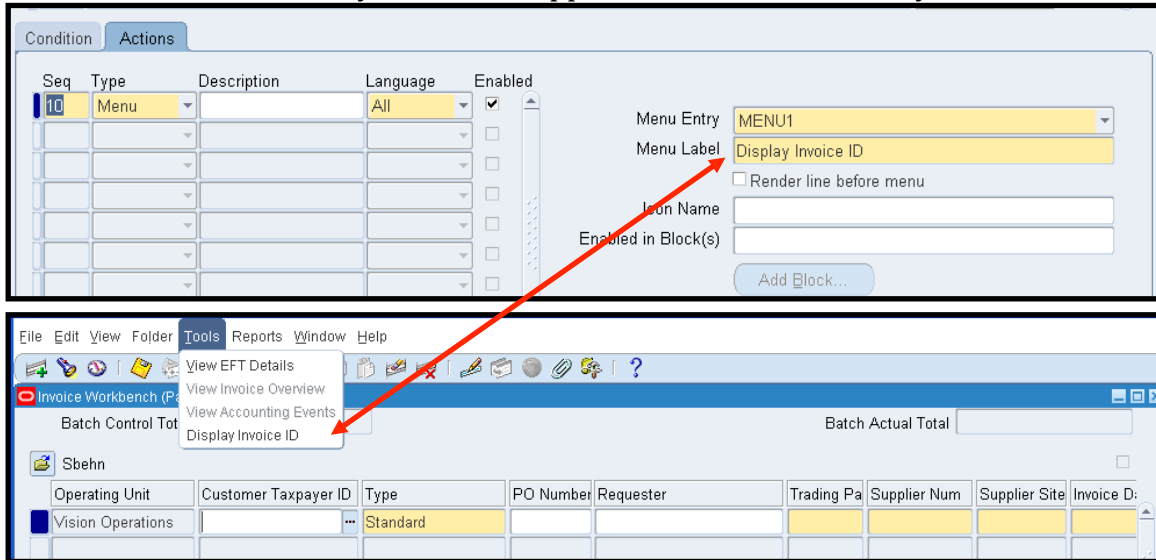


Figure 8

To define what will happen when the menu item is select, create a second personalization where the trigger event is the menu entry that was selected in figure 8. See Figure 9 for the header and condition tab for this personalization.

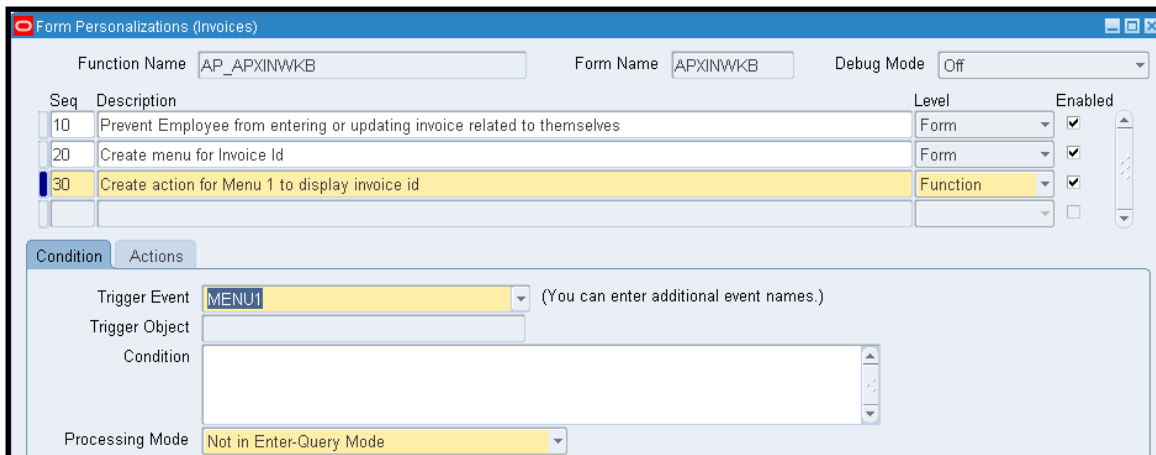


Figure 9

The action tab is shown in Figure 10. This is simply a display message using the bind variable :INV_SUM_FOLDER.INVOICE_ID. Note that the “Validate” button can be pressed to test the sql.

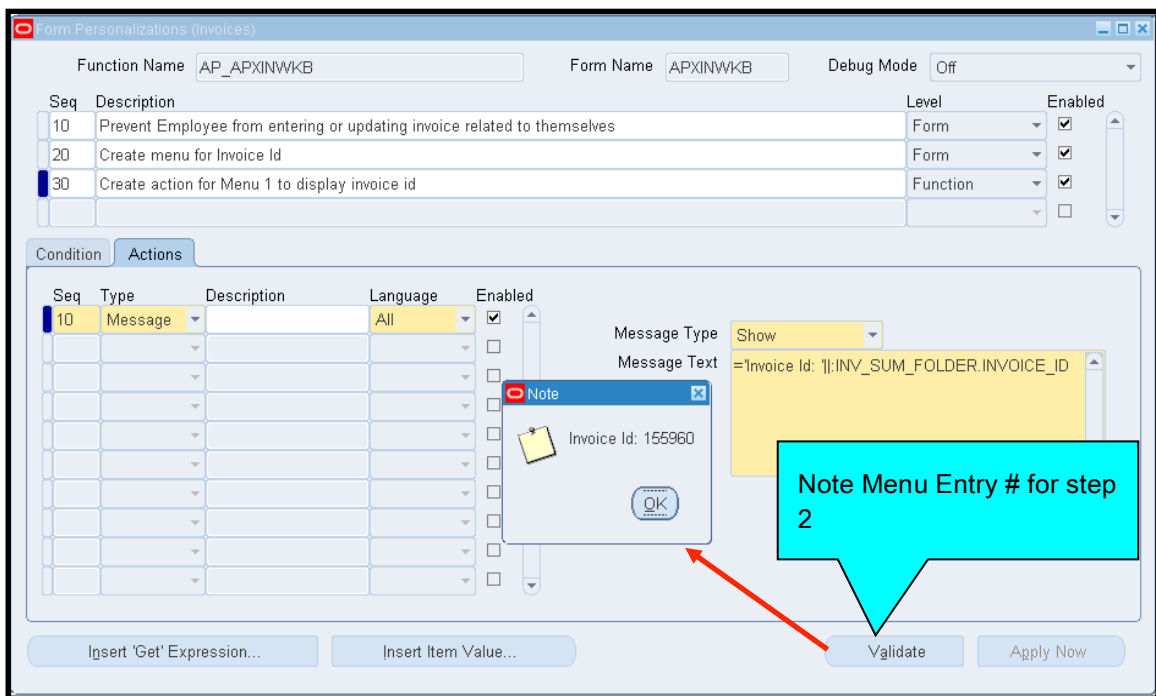


Figure 10

Example 4 – Masking Data and Hiding Fields

In 11i, the tax id is displayed on the supplier form, on the query by name query and the query by number list of values. This is a compliance issue in that this information should be secured. Note that this particular item is masked effectively in Release 12, however, there may be other examples where masking and hiding fields is applicable and this example is important for those who are choosing to migrate directly from 11i to Fusion Applications at some point. Visual changes should occur at the WHEN-NEW-FORM-INSTANCE trigger event whenever possible. To conceal the data, set the property “CONCEAL_DATA” to TRUE

as shown in Figure 11. If the requirement is to remove the field completely from the form because it is not applicable to your business process, set the property “DISPLAYED” to FALSE.

Also as shown in Figure 11, the Select By Text button can be used to find the item to be personalized by the prompt name. It is not very intuitive that the object name for Taxpayer ID is VNDR.NUM_1099_MIR. This improves the user friendliness of this tool enable better adoption by functional superusers.

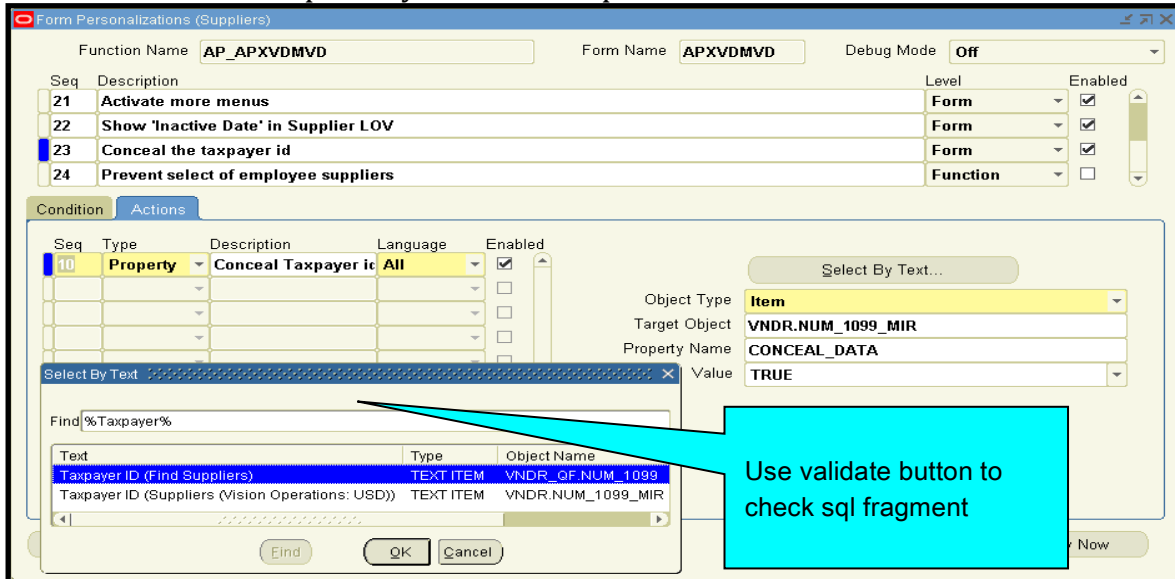


Figure 11

Example 5 – Changing LOVs

To further security sensitive data, the taxpayer ID should be removed from the List of Values (LOV) for both the query by name and query by number list of values for suppliers. For this personalization, a new record group must be created using the Builtin action with the Create Record Group from Query as shown in Figure 12. There are some limitations on changing LOVs. The headings cannot be changed and the number of fields returned cannot be changed. In this personalization, the taxpayer ID was replaced with the end date for the supplier. This served two purposes. First, the sensitive data was secured, and secondly, the payable clerks were able save time by avoiding incorrectly opening supplier records for end-dated suppliers.

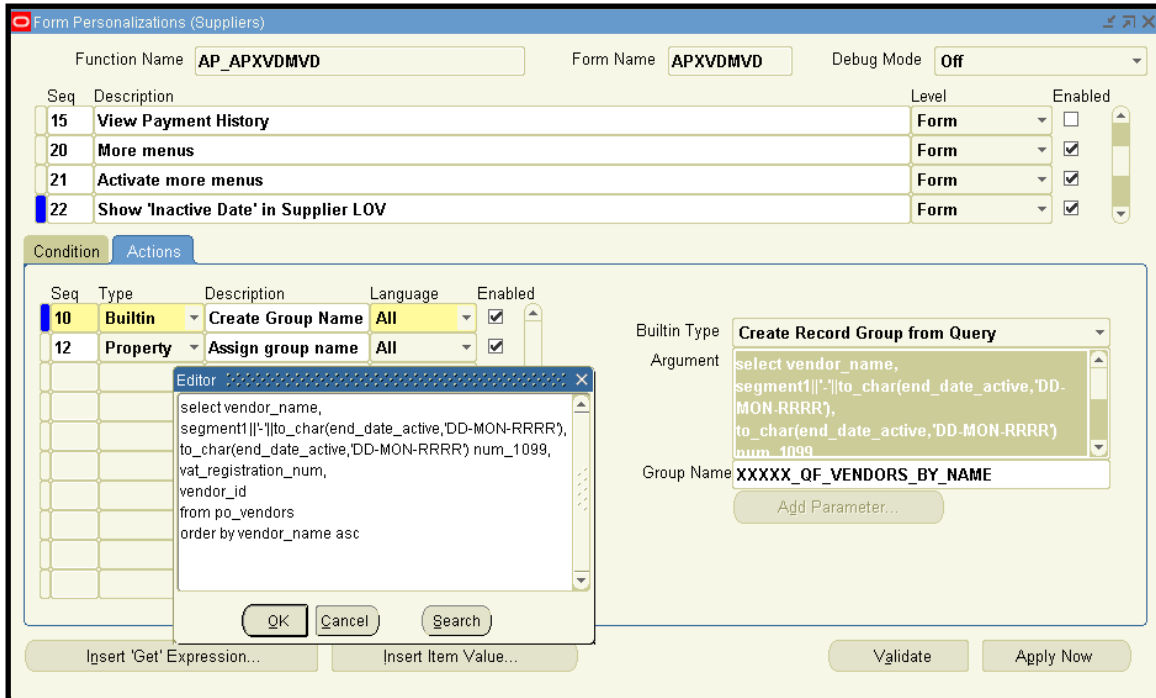


Figure 12

Note that the name used for the group in Figure 12 is XXXXX_QF_VENDORS_BY_NAME. The next part of this personalization is to assign this record group to the LOV. The action tab in Figure 13 shows the GROUP_NAME property for the LOV, QF_VENDRS_BY_NAME replaced with the name of the record group created above.

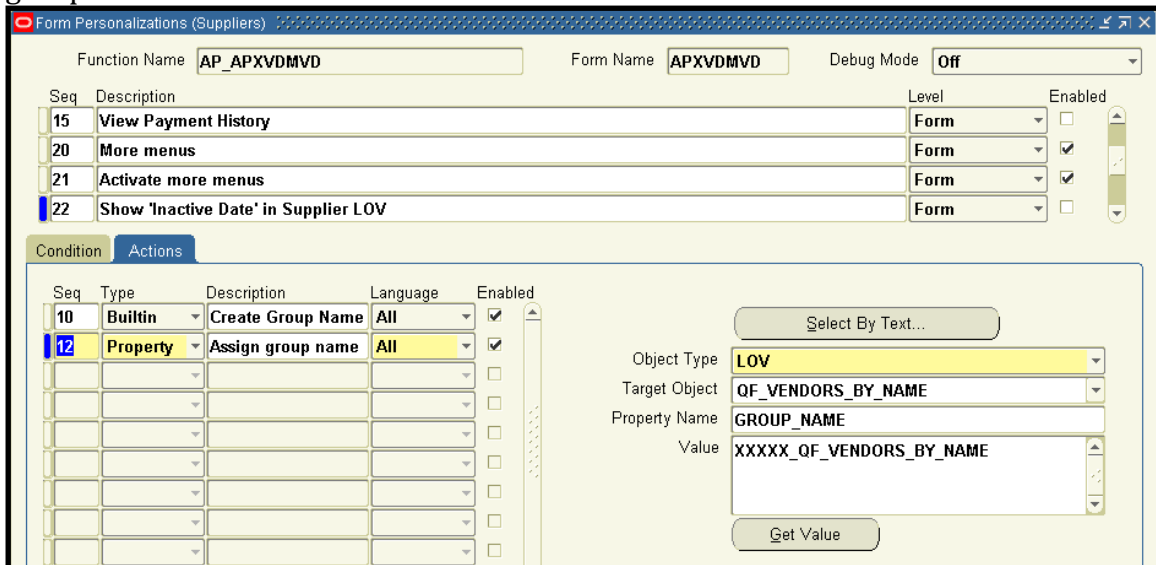


Figure 13

Professional Forms Personalization Tips

- If you disable a tab page, make sure the user cannot still navigate to the items on the tab page
- You may need to exit and re-open the form to see personalization changes

- After upgrades, go to the personalization for each form and choose Tools → Validate All
- Use debug message before and after events
- Use the *Validate* button to validate strings
- Conditions will return true, false or error
- Values will return the resulting string or an error
- Use the *Apply Now* button to apply the action now and see the results (does not always work if dependent on the results of another action)
- Use the Insert 'Get' Expression button to get any property of an item (CU1 patch)
- Turn custom code off to confirm any form problem is due to custom code
Help → Diagnostics → Custom Code → Off
- Use FNDLOAD as shown below to move personalizations to other instances
 - Download for a specific form:
FNDLOAD <userid>/<password> 0 Y DOWNLOAD
\$FND_TOP/patch/115/import/affrmcus.lct <filename.ldt>
FND_FORM_CUSTOM_RULES form_name=<form name>
 - Download all personalizations
FNDLOAD <userid>/<password> 0 Y DOWNLOAD
\$FND_TOP/patch/115/import/affrmcus.lct <filename.ldt>
FND_FORM_CUSTOM_RULES
 - Upload
FNDLOAD <userid>/<password> 0 Y UPLOAD
\$FND_TOP/patch/115/import/affrmcus.lct <filename.ldt>

OA Framework Personalization Overview

OA Framework personalizations are limited to changing the look and feel of the OA Framework forms. Advanced requirements such as additional validations require jDeveloper extensions. The scope of this white paper is limited to personalizations, not extensions.

Profile Options Impacting Forms Personalization

- **Personalize Self-Service Defn** – Set to “Yes” to allow ‘Personalize’ button to appear at the top of the page. Set this option at the user level for only a limited number of users.
- **FND: Personalization Region Link Enabled** – Set to “Yes” to display all the regional links above each region. This makes it easier to pinpoint tables, but I personally think this option clutters the screen. Set this option to “Minimal” to display only key region links. Set this option at the user level for only a limited number of users.
- **Disable Self-Service Personal** – Set this option to “Yes” to disable all personalizations at any level
- **FND: Diagnostics** – Set this option to “Yes” to turn on the “About this Page” link at the bottom left corner of self-service forms.

- **FND: Personalization Document Root Path** - Set this profile option to a tmp directory with open (777) permissions. This is where personalizations will be downloaded to or uploaded from during the migration process.

Getting Started

First access the form to be personalized. The first step in all personalizations will be to click the “Personalize” button at the top of the page as shown in Figure 14.

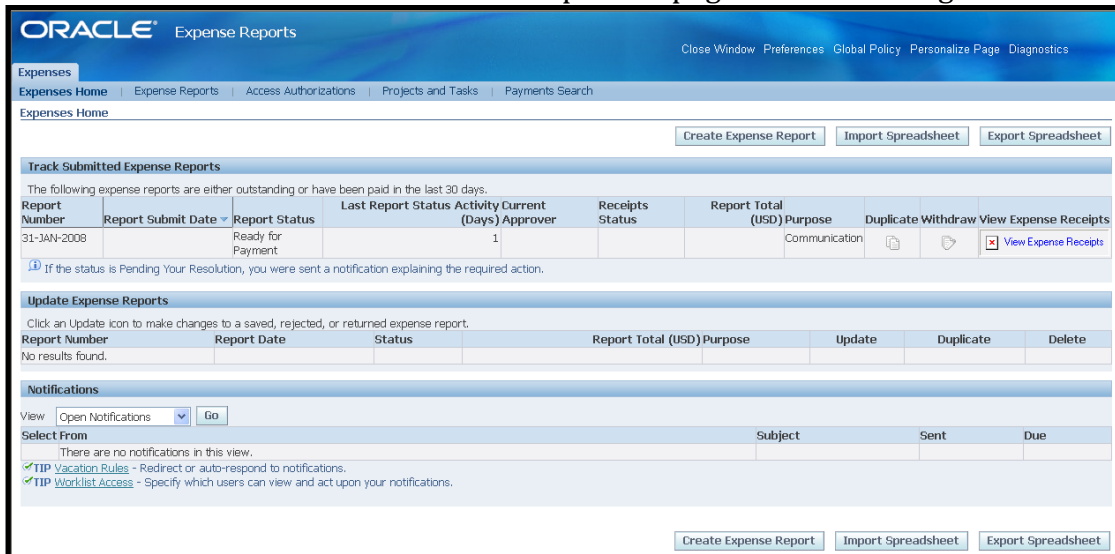


Figure 14

Example 6 – Remove a button

In this example, the client decided to utilize iExpenses for procurement card transactions. A separate responsibility was created for procurement cards and personalizations were created at the responsibility level to accommodate this requirement. This first requirement was specifically to remove the “Import Spreadsheet” button in the procurement card responsibility because this function does not apply to procurement cards. As you review these examples, think about how you can use personalizations to adapt to your own business processes without customizations.

The biggest challenge in OA Framework personalizations is finding the right item to personalize. Once you get to the right item, everything else is easy. My favorite way to find the right item is to click on the “Expand All” and “Complete View” radio buttons and then use <CTRL> F to find the label of the item to be personalized. Figure 15 is a quick view of the options.

- Simple View
- Complete View
- Expand All
- <ctrl> F
- Collapse All
- Focus

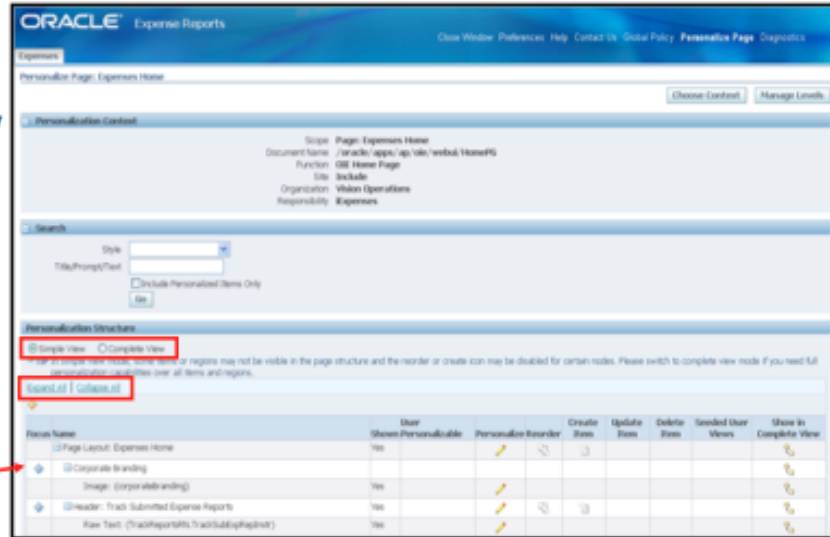


Figure 15

In this example, search for “import” and then click on the pencil to edit as shown in Figure 16.

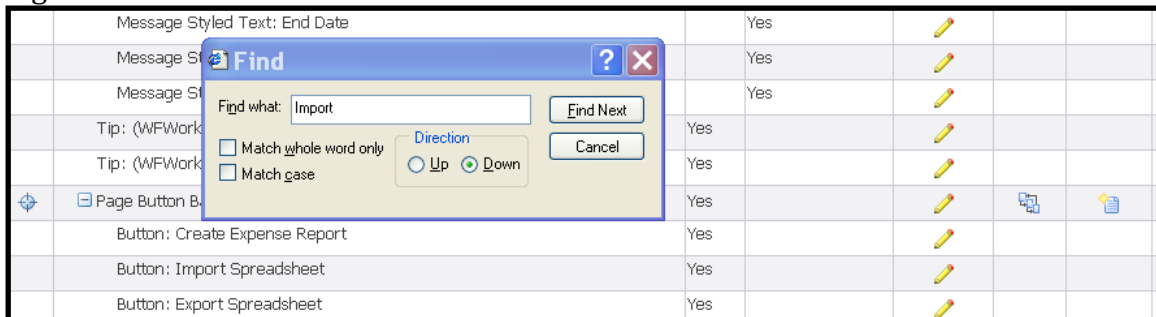


Figure 16

Set rendered = “False” at the appropriate level based on the requirement and then click the apply button as shown in Figure 16. Note that personalizations at a lower level will override personalizations at a higher level. Click the “Return to Application” button at the bottom left of the screen. The button in the application should be gone.

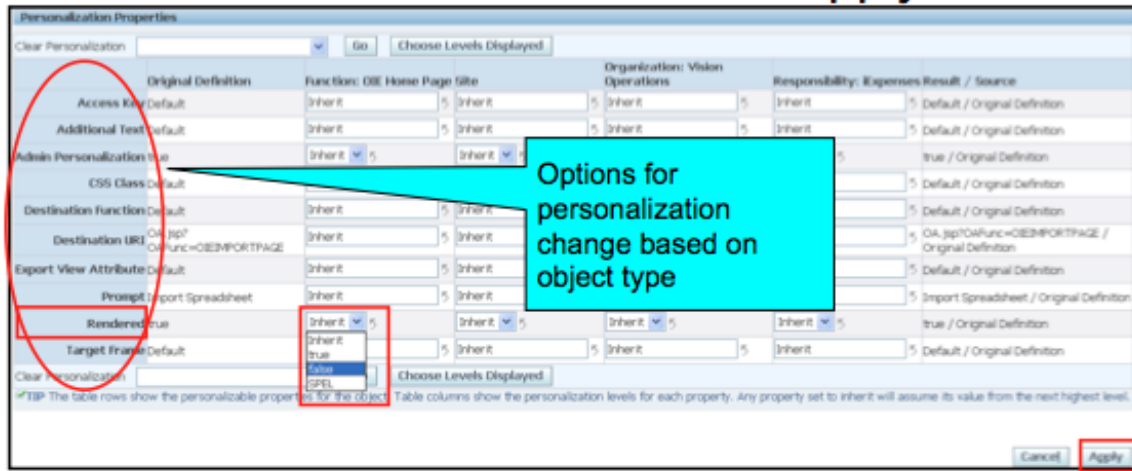


Figure 17

Example 7 – Change Prompts

In this example, the ID on the requisition notification was confusing. Many users assumed this was the requisition ID when in fact it is the notification ID. Changing the prompt clarifies the information to eliminate confusion and reduce calls to the IT support group. Changing prompts can also allow you to adapt forms to your own business terminology, which speeds up training of new employees and reduces error rates.

Remember the first step after accessing the form to personalize is to click on the “personalize page” button. In this case, my favorite <CTRL> F method to find the item to personalize isn’t going to work very well because the occurrences of “ID” will be too numerous. I still like <CTRL> F, but in this case, I search for a nearby prompt such as “Sent” to find the correct item to personalize as shown in Figure 18. Click the pencil to edit this item.

Focus Name	User	Shown Personalizable	Personalize	Reorder	Create Item	Update Item	Delete Item	Seeded User Views
Table Layout: (NtfDetailsFn.NtfDetailsAttr.leftTable)	Yes							
Message Styled Text: To	Yes							
Message Styled Text: Sent	Yes							
Message Styled Text: Due	Yes							
Message Styled Text: Priority	Yes							
Message Styled Text: Status	Yes							
Message Styled Text: Closed	Yes							
Message Styled Text: ID	Yes							

Figure 18

The prompt was changed to “Notif ID” at the site level to clarify the purpose of the field as shown in Figure 19. The results are shown in Figure 20.

Property	Original Definition	Function: Notification Details	Site	Organization: Vision Operations	Responsibility: System Administrator
Access Key	Default	Inherit	Inherit	Inherit	Inherit
Additional Text ID	Inherit	Inherit	Inherit	Inherit	Inherit
Admin Personalization	True	Inherit	Inherit	Inherit	Inherit
CSS Class	OraDataText	Inherit	Inherit	Inherit	Inherit
Destination Function	Default	Inherit	Inherit	Inherit	Inherit
Destination URI	Default	Inherit	Inherit	Inherit	Inherit
Export View Attribute	Default	Inherit	Inherit	Inherit	Inherit
Long Tip Region	Default	Inherit	Inherit	Inherit	Inherit
Maximum Length	Default	Inherit	Inherit	Inherit	Inherit
Prompt ID	Inherit	Inherit	Notif ID	Inherit	Inherit

Figure 19

ORACLE Administrator Workflow

Home | Logout | Preferences | Help | Personalize Page | Diagnostics

Home | Developer Studio | Business Events | Status Monitor | Notifications | Administration

Workflow Configuration | Vacation Rules | Notification Search | Signature Evidence Store

Administration: Notification Search >

Purchase Requisition 1443 for Behn, Susan (3,000.00 USD)

From: Behn, Susan
To: Brown, Casey
Sent: 20-Jan-2009 11:22:58
Notif ID: 2739761

Description: New Computer
Requisition Total: 3,000.00 USD
Non-Recoverable Tax: 0.00 USD

Approve | Approve And Forward | Forward | Reject | Reassign | Request Information | Printable Page

Line	Description	Supplier	Cost Center	Unit	Quantity	Price (USD)	Amount (USD)
1	Computer - Laptop		110				3,000.00

Figure 20

Example 8 – Hide/Reorder Columns in a Table

In example 8, we are hiding the “duplicate” in column in the iExpenses table because this functionality does not apply to the procurement card transaction. Fields within the table can also easily be re-ordered to speed up data entry or improve the view for your specific requirements.

Click the personalize button, find the item to edit and click on the pencil as shown in Figure 21.

Message Styled Text: Receipts Status	Yes	Yes	
Message Styled Text: Report Total		Yes	
Message Styled Text: Report Total	Yes	Yes	
Message Styled Text: Purpose		Yes	
Switcher: Duplicate	Yes	Yes	

Figure 21

Set rendered to false as shown in Figure 22 and the table column will disappear.

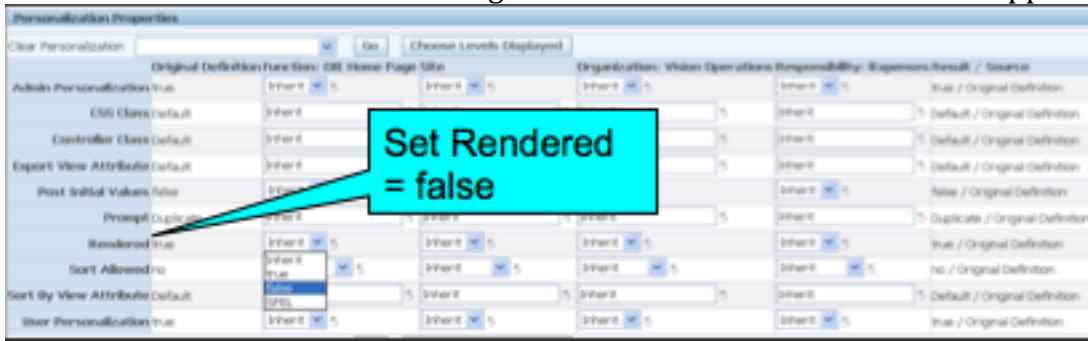


Figure 22

To change the order the columns in a table, find the table item and click the reorder icon as shown in Figure 23.



Figure 23

Use the arrows to move the columns around as shown in Figure 24.

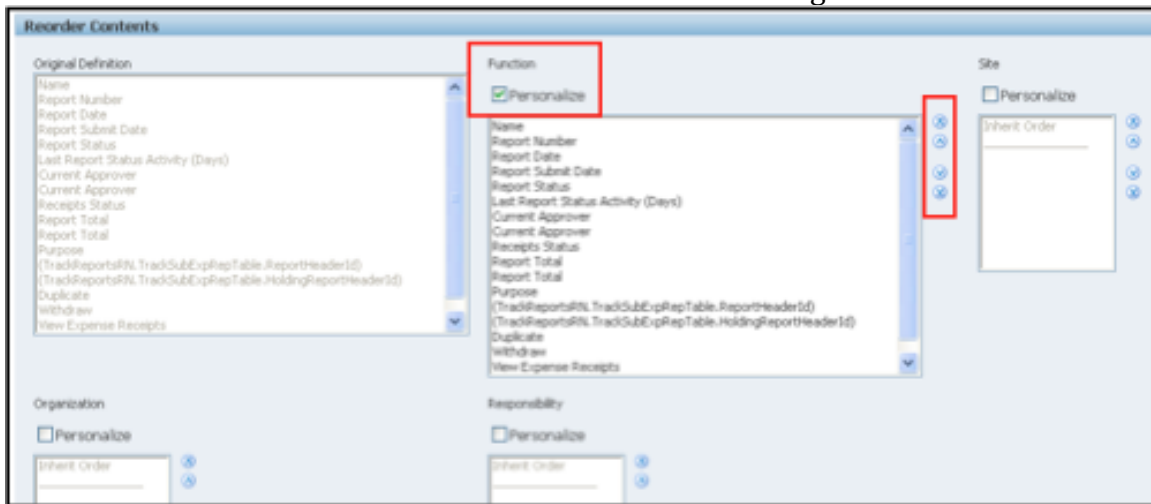


Figure 24

Example 9 – Set Initial Values

In this example, the initial value for the purpose was set based on the responsibility. For expense report responsibilities, the purpose was set to “Internet Expenses”. For procurement card responsibilities, the purpose was set to “Procurement Card.” This streamlines data entry and reduces errors.

Click the Personalize Page button...again...and find the item to edit. Change the initial value at the responsibility level as shown in Figure 25.

Original Definition	Function: OE Expenses Entry	Site	Organization: Vision Operations	Responsibility: Expenses	Result / Source
Access Key: Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
Additional Text: Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
Admin Personalization: true	Inherit	Inherit	Inherit	Inherit	true / Original Definition
CSS Class: OraFieldText	Inherit	Inherit	Inherit	Inherit	OraFieldText / Original Definition
Destination Function: Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
Export View Attribute: Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
Initial Value: Default	Internal Expenses	Inherit	Inherit	Inherit	Internal Expenses / Function
Long Tip Region: Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition

Figure 25

Example 10 – Add Item

In this example, we had a client that required receiving for every item ordered by over 10,000 employees. Training was obviously a challenge. To meet this challenge, web training was created for the receiving process and a tip was added to the self-service receiving form which contained a direct hyperlink to the training application. Adding information tips, even if they are not necessarily this elaborate, can reduce errors, reduce training costs and make users self sufficient.

We are going to slightly step outside the personalization world and go to the Functional Administrator responsibility to create the tip text as part of the message database as shown in Figure 26. Note that the “code” attribute is what will be referred to in the personalization to display the text in the “text” field.

ORACLE Applications Administration
 Home Logout Preferences Help Personalize Page Diagnostics

Security | Core Services | Personalization | File Manager

Lookups | Messages | Profile Categories | Profiles | Functions | Menus | Caching Framework | Personalization

Core Services: Messages >

Create Message

* Indicates required field
 ✓ TIP The 'Code' and 'Application Name' fields cannot be updated after message is created. [Cancel] [Apply]

* Code: SBRCVTIP2
 (Example: FII_AR_MESSAGE_NAME)

Description: Tip for Receiving Training Instructions

* Application Name: Purchasing

* Language: American English

Type: 30% Expansion Prompt

* Text: For detailed instructions on how to perform Receiving, please select 'Receipt Transaction Training' in the XXX Company Training Application

Figure 26

Now go back to the receiving page and click on “personalize page.” Find the item for “Tip” and click the “Create Item” icon shown in Figure 27.

Personalization Structure

Simple View | Complete View

Expand All | Collapse All

Page Layout: Oracle iProcurement: Receiving > Row Layout: (SearchResultTipRow) >

Focus Name	Shown User	Personalizable	Personalize	Reorder	Create Item
Cell Format: (SearchResultTipCell)	Yes				
Tip: Use the Previous/Next navigation tool to make selections across multiple pages	Yes				

Figure 27

Enter the item style, id, Text to identify the tip and the message name for the message created as shown in figure 28. Note that the name of the message code is entered as the “Tip Message Name.”

ORACLE iProcurement
 Shopping Cart Home Logout Preferences Help **Page** Personalize Diagnostics

Shop Requisitions Receiving Contractors

Create Item
 * Indicates required field Cancel Apply

Level: Function: POR:Receive Orders
 Item Style: Tip

Property	Value
* ID	SBRCVTIP2
Admin Personalization	true
Attribute Set	
CSS Class	
Comments	
Extends	
Rendered	true
Scope	,
Text	More Training
Tip Message Name	SBRCVTIP2

Figure 28

Migrating OA Framework Personalizations

Migrate OA Framework Personalizations from the Functional Administrator Responsibility → Personalization → Import/Export tab. Personalizations can be exported/imported at any level from a single page to the entire library of personalizations. The personalizations will be exported to the directory defined in FND: Personalization Document Root Path. The DBA will need to move these files to the same directory in the target instance. Then go to the Functional Administrator responsibility in the target instance to import.

Summary

Personalizations reduce customizations and extensions, are easy to implement and can ultimately save money currently through the limitation of complex programming and in the future by limiting exposure during patches and upgrades. Consider the advantages listed below.

- Less customizations result in lower maintenance cost and lower patching/upgrade costs
- Increase compliance reducing audit cost
- Speed up data entry
- Decrease error rates
- Reduce training costs

- Make users self sufficient

Additional Resources

The documentation listed below contains details related to extending and personalizing Oracle Applications.

- Oracle Applications User Interface Standards for Forms-Based Products
- Oracle Applications Developer's Guide
- Oracle Applications System Administrator's Guide
- Oracle Applications User Guide
- My Oracle Support Note ID: 279034.1 – Forms Personalization
- Configuring, Reporting and System Administration in HRMS
- Oracle Self Service Web Applications Implementation Manual