Personalize the Forms - How to
Oracle Applications
Release 11.5.10

A Technical White paper
June 2005

Ramakrishna Goud
ramakrishna.goud@ge.com

Executive Overview

With the Oracle E-Business Suite release 11.5.10, the Oracle has introduced a mechanism which revolutionizes the way the forms can be customized to fulfill the customer needs. For many years, Oracle Applications has provided a custom library using which the look and behavior of the standard forms can be altered, but the custom library modifications require extensive work on SQL and PL/SQL. In the release 11.5.10, Oracle has provided a simple and easy feature to implement the customer specific requirements without modifying the underlying forms code or CUSTOM library. Although CUSTOM library still can be used for forms customization to implement the complex business logic, the personalization feature provided in the latest release is easy, faster and requires minimum development effort.

The Need

Most of the customers wish to modify the way the forms look and behave. They want to change the labels on the form fields, hide fields, hide buttons, amend custom logic etc.

The motives could be

- The fields are unneeded
- To match the customer specific business requirements
- Or to match the lingo used by the customer

The Solution

Oracle Applications release 11.5.10 has provided a user interface "Personalization form" which will be used to define the personalization rules. These personalization rules establish the custom actions to be performed by the forms during runtime. The actions may include changing prompts, showing custom messages or prevent the user from doing the transactions etc. This feature also provides a well-built controlling mechanism which restricts the execution of the personalization rules at different levels (Industry, site responsibility and user). The personalization doesn't violate the support from Oracle and these rules also conceded with the future upgrades without any additional effort.

Targeted Audience

This document emphasizes the structural guidance on implementing personalization rules according to the customer requirements. The targeted audience is technical consultants who are familiar with Oracle Developer fundamentals. You can refer to the Oracle documentation for complete and accurate information on personalization.

Enable or Disable the personalization feature

The personalization form should be used to implement the custom rules on a specific form. The specific form refers to the desired form on which you want to apply the custom business logic or modify the form behavior.

The personalization form is invoked by…

Menu Navigation: Help → Diagnostics → Custom Code → Personalize

By default, this menu option is enabled when any of the standard forms are opened, but at other times this menu option is disabled. The personalization form cannot be opened when the user is on the responsibility navigator. This means the specific form should be opened first, then invoke the "Personalization form" using the above navigation. Then define the personalization rules and actions on the personalization form, validate and apply them which automatically add the custom rules to the underlying specific form.

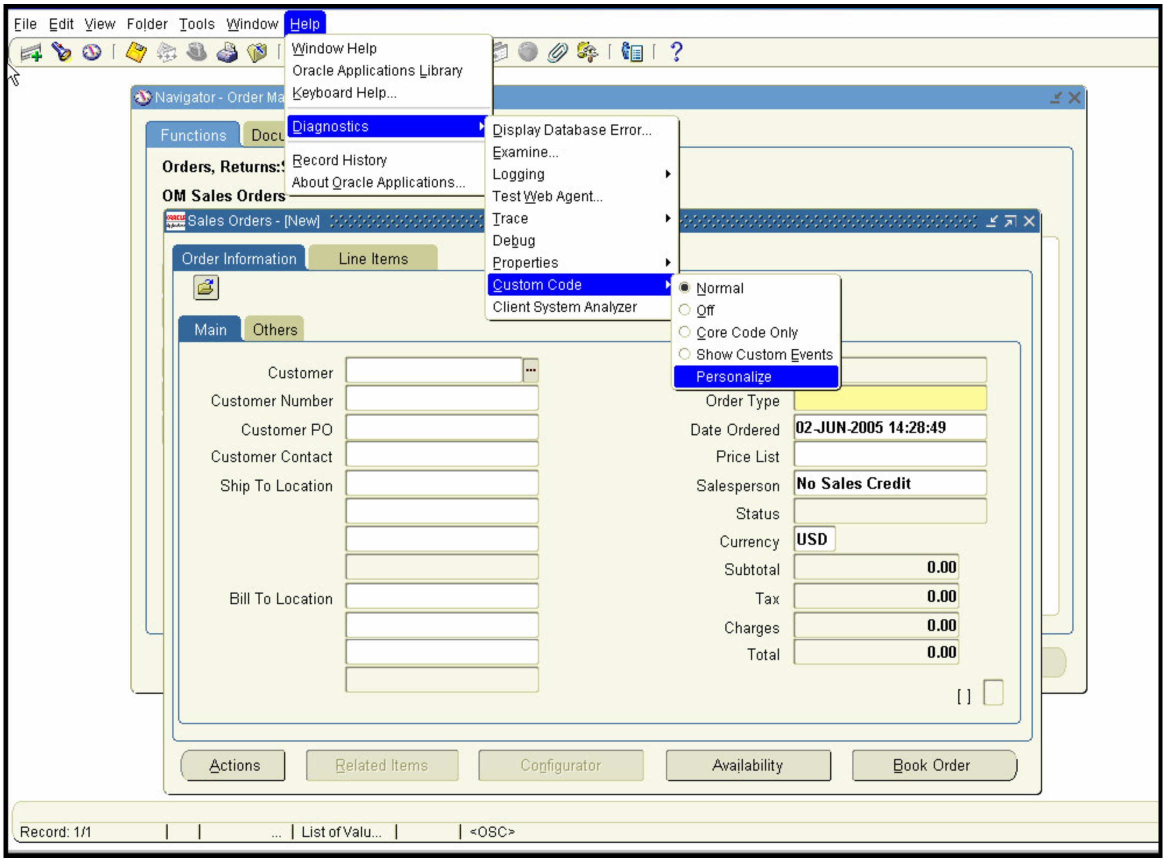
For example, if the specific form is "Sales Order"; Open the "Sales Order" form first, and then invoke the "Personalization Form" using the menu option.

By default the "Personalize" menu is visible to all the users; this can be controlled with help of profile options. By setting up the below profile options, the access to the Personalize menu can be limited for the authorized users. This will prevent unauthorized users from changing the look and behavior of the forms.

The profile options are…

- Utilities: Diagnostics = Yes/No
- Hide Diagnostics = Yes/No

The below picture depicts invoking the Personalization form on top of the Sales Order form. Open the Sales Order Form and click on "Help → Diagnostics → Custom Code → Personalize".



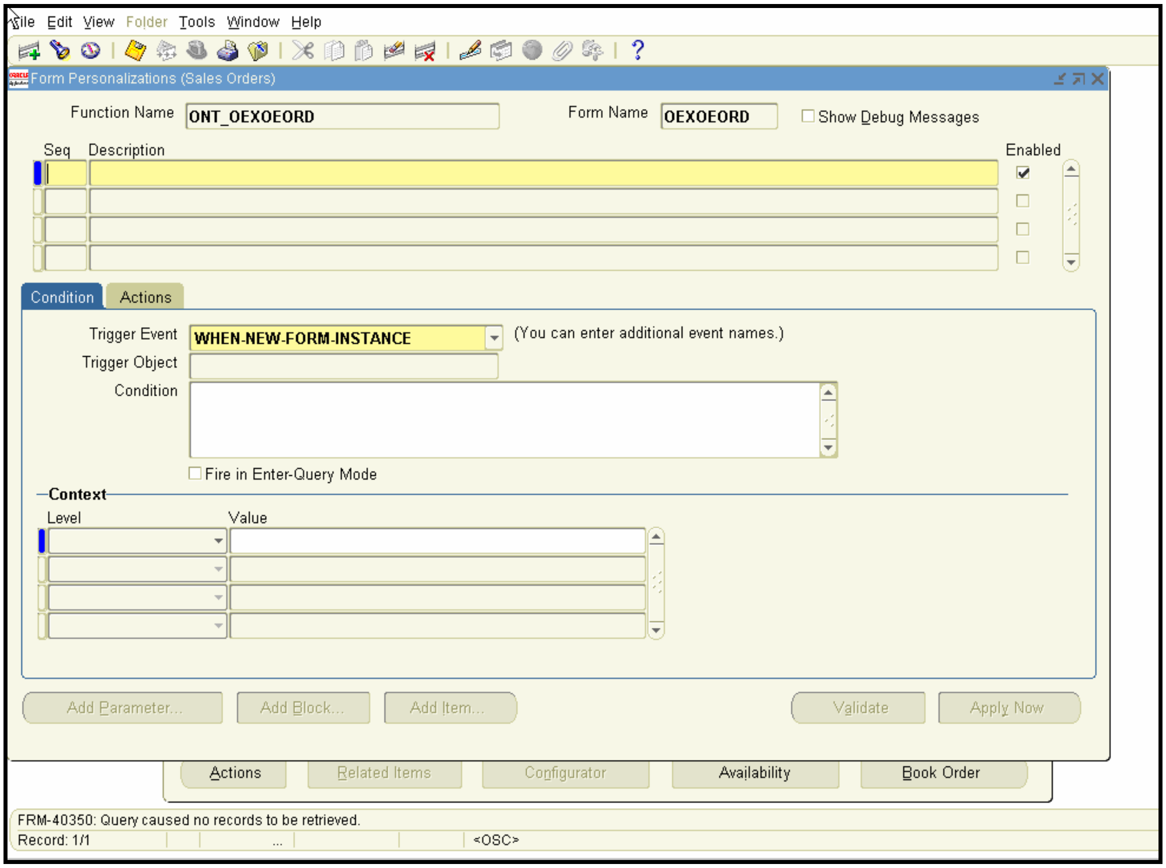
Date Ordered: 02-JUN-2005 14:28:49
Salesperson: No Sales Credit
Currency: USD
Subtotal: 0.00
Tax: 0.00
Charges: 0.00
Total: 0.00
Custom Code: ● Normal
Record: 1/1

Figure 1: Invoking Personalization on top of the Sales Order Form

Understand the Personalization Form

When invoked the personalization form automatically queries the function, form and pre-defined personalization rules, if any exists for the specific form.

The below picture shows the form name as OEXOEORD i.e. Sales Order form on which the personalization form is opened.



Function Name: ONT_OEXOEORD

Form Name: OEXOEORD

□ Show Debug Messages

Enabled: ☑

Trigger Event: WHEN-NEW-FORM-INSTANCE (You can enter additional event names.)

Trigger Object:

Condition:

□ Fire in Enter-Query Mode

FRM-40350: Query caused no records to be retrieved.

Record: 1/1 ... <OSC>

Figure 2: Personalization Form

The form mainly contains four sections. They are…

- Rules
- Conditions
- Context
- Actions

The purpose of each section is explained below…

Understand the Rules

Rules administer the personalization needs to be implemented on the form. Each rule contains a sequence number and the description. The rule can be activated or de-activated using the "Enabled" checkbox. The rule can be deleted when no longer needed.

For example, a rule can be "Change the Order Number prompt to Claim Number"

Defining rules doesn't identify when the rule should get evaluated or applied. For each rule, there should be conditions attached which power the execution of the rule.

Understand the Conditions

Conditions decide the event the rule to be executed. Each condition mainly contains three sections i.e. Trigger Event, Trigger Object and Condition.

Trigger Event specifies the occurrence where the rule should be executed.

Trigger Object is the object on the form to determine at what level the rule should be executed. The values can be "<Form Name>", '<Block Name>" or "<Item Name>"

Condition is any SQL statement to control the execution of rule when the mentioned criterion is met.

Example:
Trigger Event = WHEN-NEW-ITEM-INSTANCE
Trigger Object = ORDER_NUMBER
Condition = where user <> 'SYSTEM'

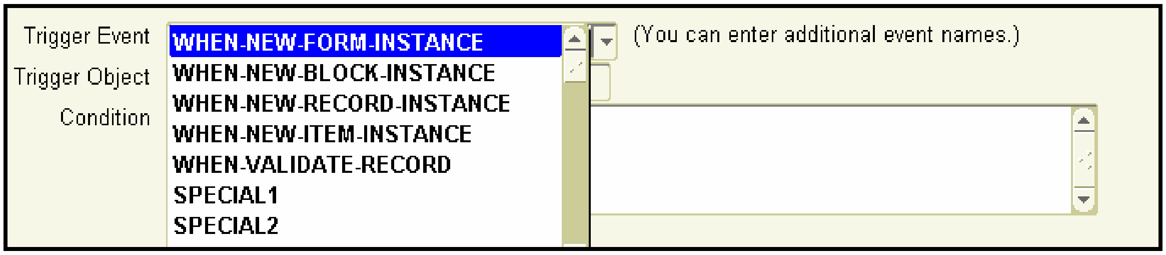


Figure 3: Conditions section on the Personalization Form

Understand the Context

Context manages to whom the personalization should apply. This is similar to the concept of using profile options in Oracle Applications. The various levels are Site, Responsibility, Industry and User. During runtime, the values provided in the context are evaluated and personalization rules will be applied. Usage of context is very vital in implementing the personalization to prevent the inappropriate users accessing these customizations of the form.

Example:

Context = Responsibility
Value = Order Management Super User

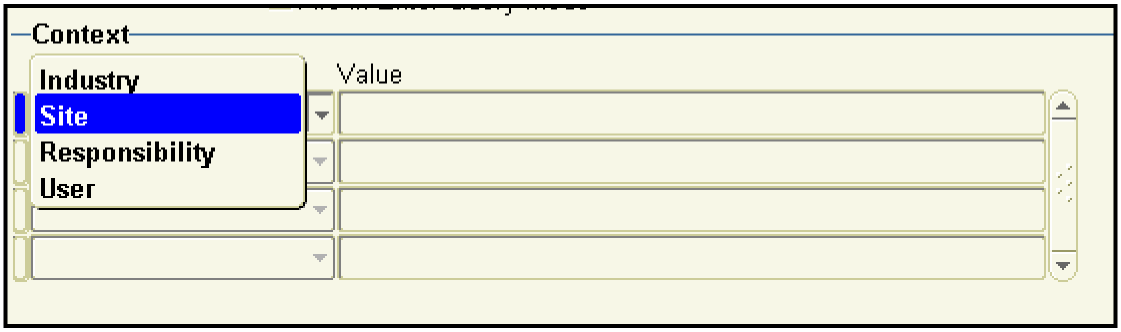


Figure 4: Context section on the Personalization Form

Understand the Actions

Actions decide the exact operation to be performed when the conditions and context return true during the runtime. Usually each rule should be associated with at least one action.

The types of actions available are…

- Property
- Message
- Builtin
- Special

Each action contains a sequence number, description and language. Actions can be enabled, disabled or deleted. Based on the action type selected, the fields on the right sector of the actions tab will be refreshed where the action parameters are entered.

Action Type: Property

The action type "Property" is used to set the properties of the objects. The various objects include "Item, Window, and Block etc". Actual object name defined in the form should be entered after selecting the object type.

Property Name is the property to be changed
Value is the new desired value for the property.

The current value of the property will be displayed when clicked on the "Get Value" button.

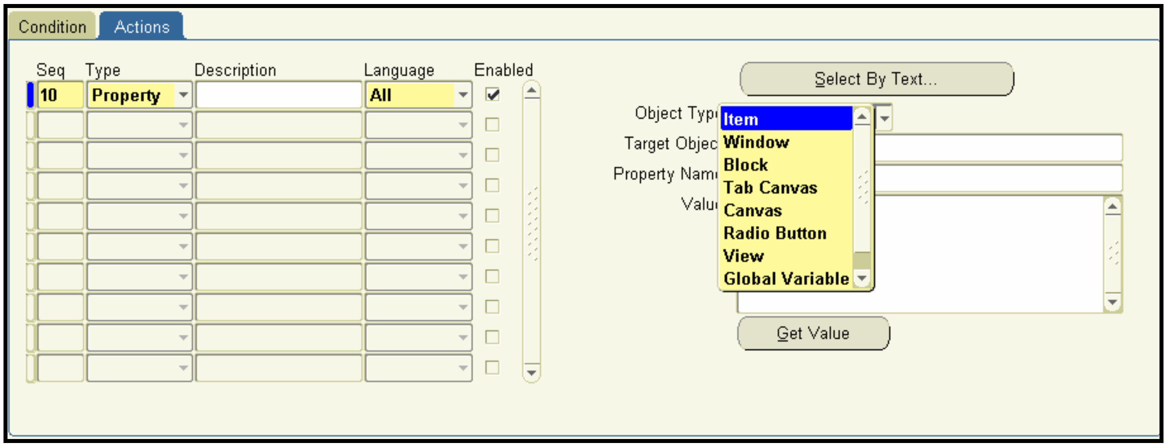
Example:

Object Type = "Item"
Target Object = ":ORDER.ORDER_NUMBER"
Property Name = "PROMPT_TEXT"
Value = "Claim Number"

The value is interpreted at runtime, so you can use SQL functions and operators. Any value started with "=" operator will be interpreted at runtime; otherwise the value is treated as is entered in the value field

Example:

Value => ='Welcome to the Sales Order Entry Mr. ' | | user



Seq	Type	Description	Language	Enabled
10	Property		All	☑

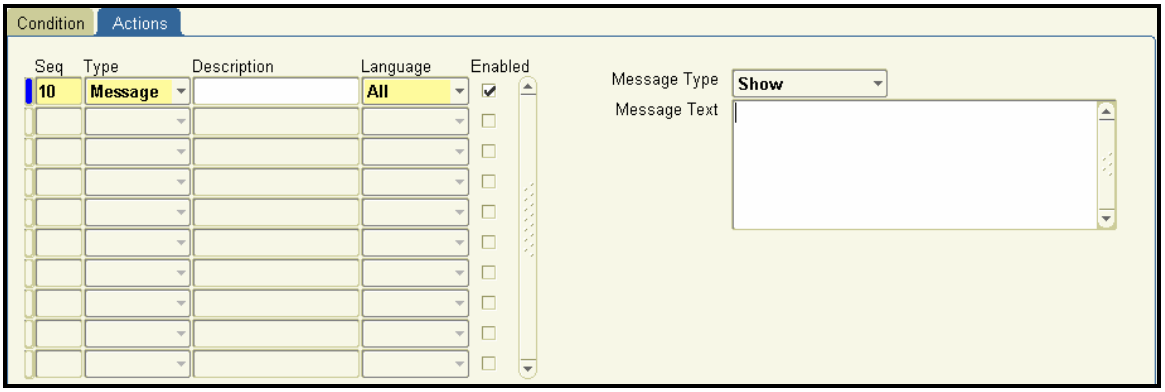
Figure 5: Action Type "Property"

Action Type: Message

The action type "Message" is used to display custom messages during runtime. Message Type and Description should be entered after selecting the action type as "Message". The available message types are "Error, Warning, Hint, Question and Debug". Depending on the response expected from the user, the appropriate message type should be selected.

Example:

Message Type = "Hint"
Message Text = "Please Follow the Date format DD-MON-YYYY"



Seq	Type	Description	Language	Enabled
10	Message		All	☑

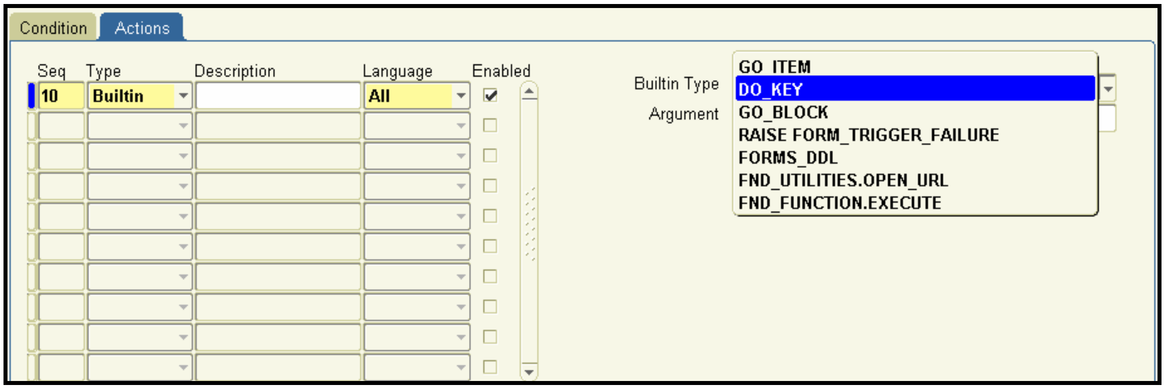
Message Type: Show

Figure 6: Action Type "Message"

Action Type: Builtin

The action type "Builtin" is used to execute the form and AOL API's. Depending on the API type selected, the parameters should be entered.

Example:
Builtin Type = FND_UTILITIES.OPEN_URL
Argument = http://www.oracle.com



Seq	Type	Description	Language	Enabled
10	Builtin		All	☑
				□
				□
				□
				□
				□
				□
				□
				□
				□

Builtin Type: DO_KEY
Argument:

Figure 7: Action Type "Builtin"

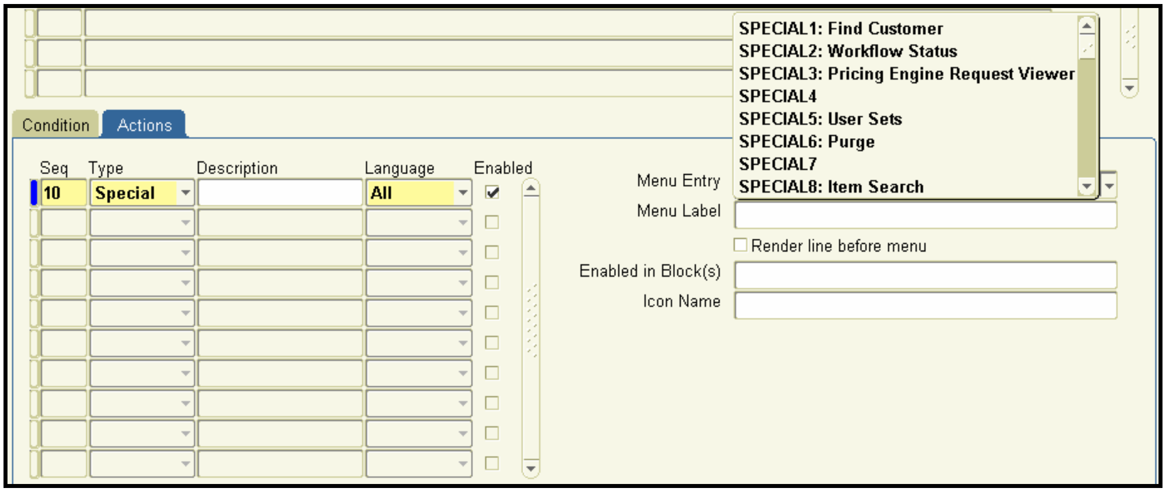
Action Type: Special

The action type "Special" is used to activate the available special menus on the Tools menu. Oracle Applications provide 45 special menus under Tools menu which can be used by customers based on their requirements.

Select the SPECIAL menu which is not used by the form. Menu label is the prompt which appears to the users when Tools menu is invoked, block specifies the blocks for which the special menu should be activated and Icon name is the .ico file name.

A separator can be created above the activated special menu by selecting the "Render line before menu" checkbox.

Example:
Menu Entry = SPECIAL4
Menu Label = Additional Order Header Information
Enabled in Blocks(s) = ORDER
Icon Name = Flower.ico



Seq	Type	Description	Language	Enabled
10	Special		All	☑

Menu Entry

Menu Label

□ Render line before menu

Enabled in Block(s)

Icon Name

Figure 8: Action Type "Special"

Validate and Test

All the rules and actions can be validated and tested immediately by clicking on the "Validate" and "Apply Now" buttons. Navigate or reopen the specific form to observe the result of the personalization.

Add Parameters, Add Blocks and Add Items buttons are used to select the Parameters, Blocks and Items from the desired form respectively.

Examples: Step by step

The examples provided in this document are executed on the Order Management "Sales Order" form. Screenshots are also provided for the reference

Example1: Change the "Order Number" label to "Claim Number"

Step1: Open the Sales Order Form

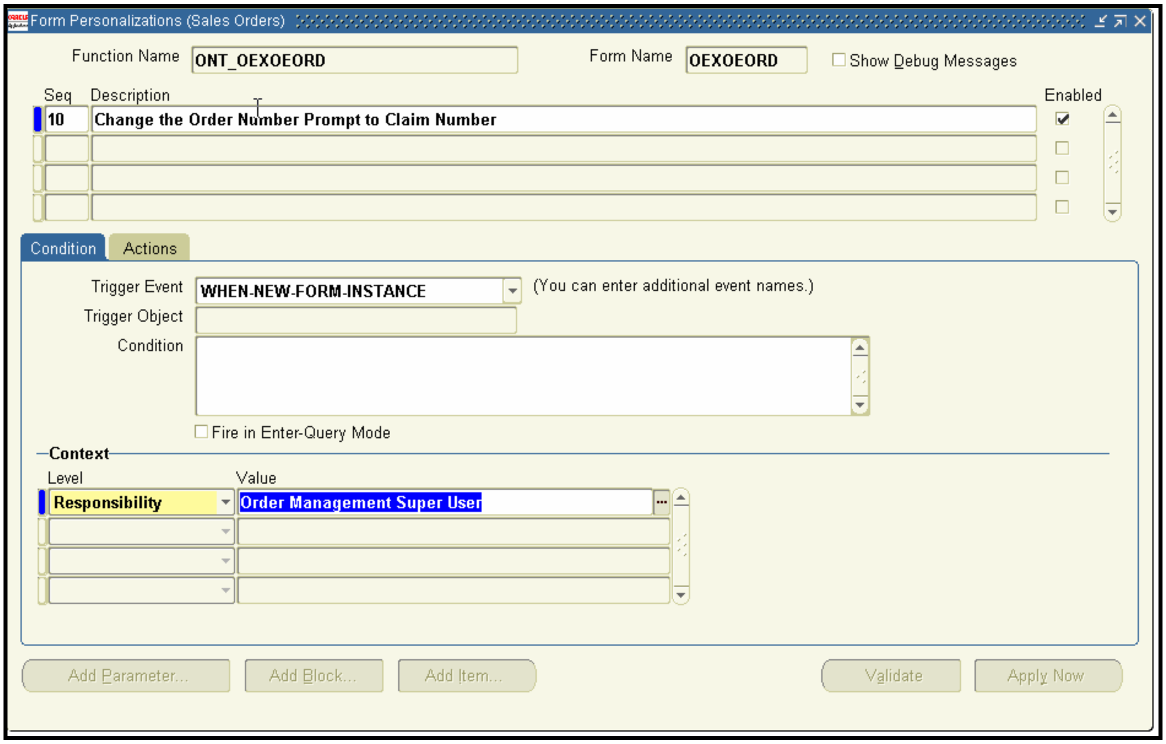
Step2: Open the Personalization form using the navigation "Help → Diagnostics → Custom Code → Personalize"

Step3: Enter the sequence number as 10 and description as "Change the Order Number Prompt to Claim Number"

Step4: Select the Trigger Event as "WHEN-NEW-FORM-INSTANCE"

Step5: Select or enter the following values under Context section

- Level = Responsibility
- Value = Order Management Super User



Function Name: ONT_OEXOEORD

Form Name: OEXOEORD

□ Show Debug Messages

Seq	Description	Enabled
10	Change the Order Number Prompt to Claim Number	☑
		□
		□
		□

Trigger Event: WHEN-NEW-FORM-INSTANCE (You can enter additional event names.)

Trigger Object:

Condition:

□ Fire in Enter-Query Mode

Context

Level	Value
Responsibility	Order Management Super User

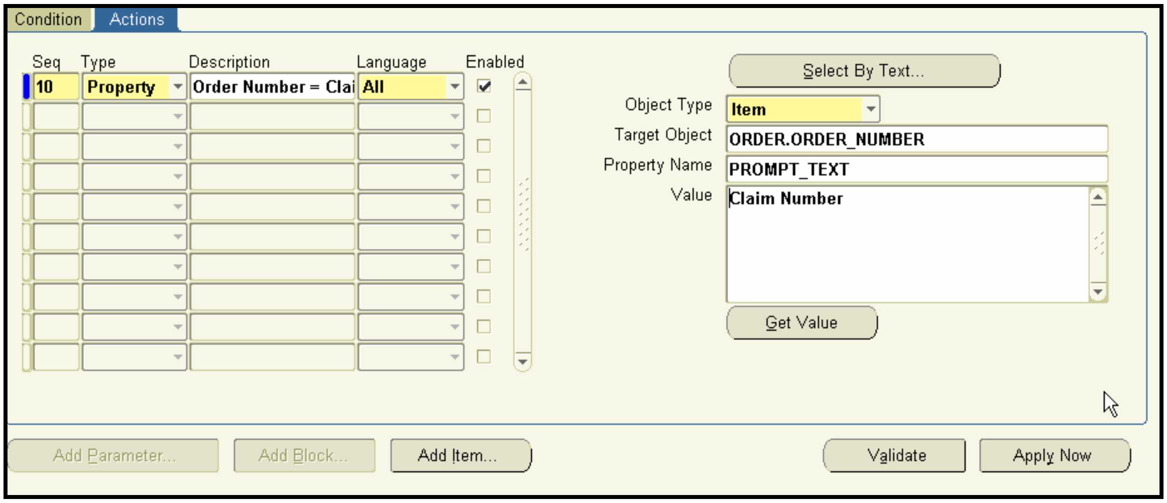
Figure 9: Example for changing the Order Number prompt to Claim Number

Step6: On Actions tab, enter or select the following values

- Sequence = 10
- Type = Property
- Description = Order Number to Claim Number
- Language = All
- Enabled = Yes

Step7: On the right side of Actions tab, enter or select the following values

- Object Type = Item
- Target Object = ORDER.ORDER_NUMBER
- Property Name = PROMPT_TEXT
- Value = Claim Number



Seq	Type	Description	Language	Enabled
10	Property	Order Number = Clai	All	☑

Object Type: Item

Target Object: ORDER.ORDER_NUMBER

Property Name: PROMPT_TEXT

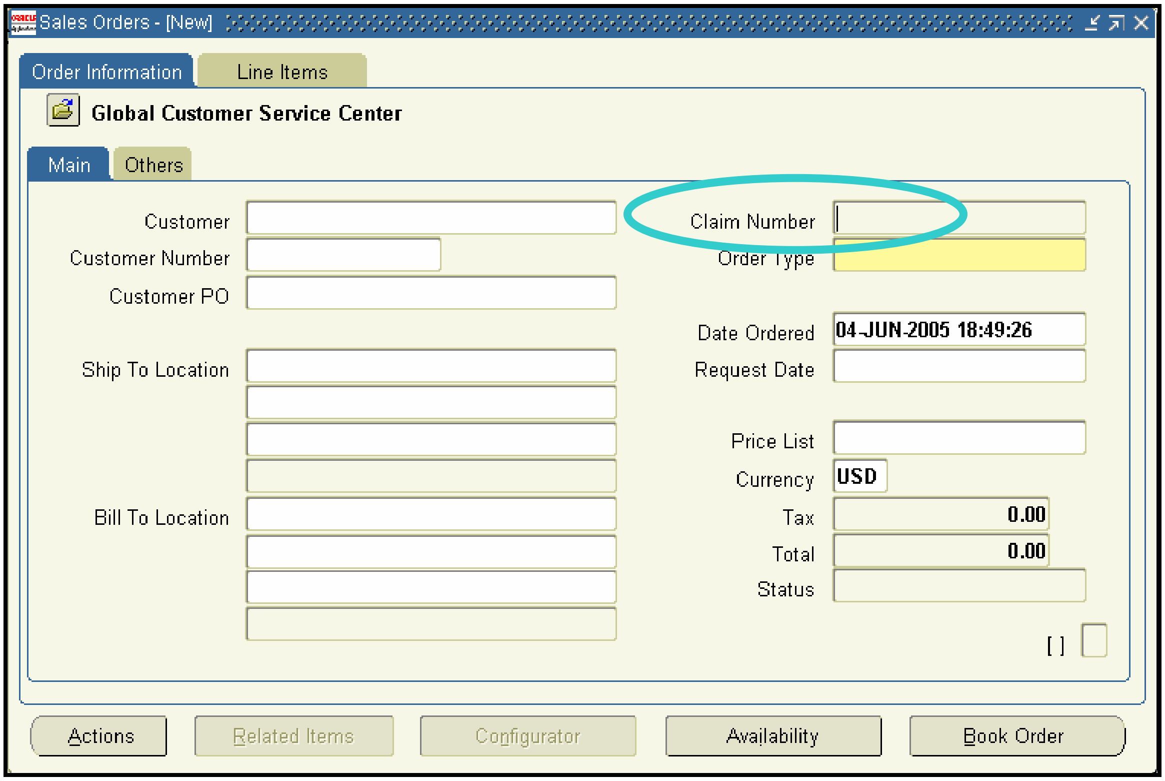
Value: Claim Number

Figure 10: Example 1 – Entering values for Action Type "Property"

Step8: Click on "Apply Now" button

Step9: Close both the forms and reopen the Sales Order form

Result: The Order Number label should reflect as Claim Number



Global Customer Service Center

Customer	
Customer Number	
Customer PO	
Ship To Location	
Bill To Location	
Claim Number	
Order Type	
Date Ordered	04-JUN-2005 18:49:26
Request Date	
Price List	
Currency	USD
Tax	0.00
Total	0.00
Status	

Figure 11: Result of Example 1

Example2: Warn the users not to enter Sales Orders on Sunday

Step1: Open the Sales Order Form

Step2: Open the Personalization form using the navigation "Help → Diagnostics → Custom Code → Personalize"

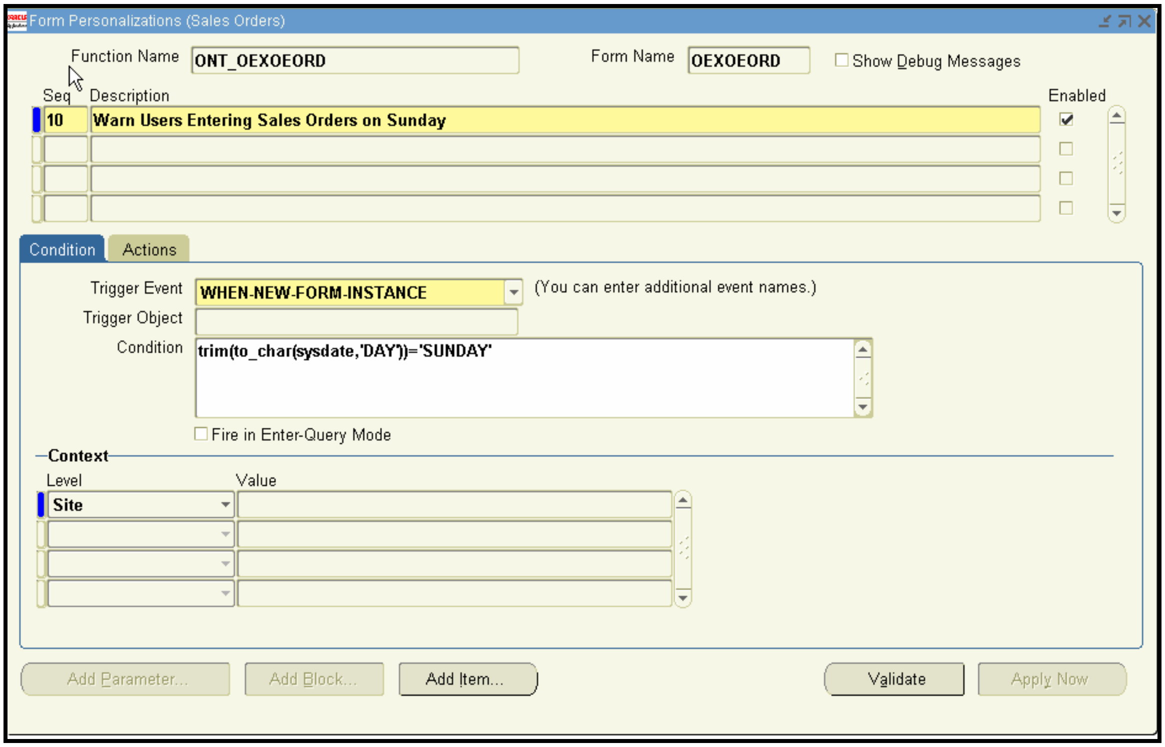
Step3: Enter the sequence number as 10 and description "Warn Users Entering Sales Orders on Sunday

Step4: Select the Trigger Event as "WHEN-NEW-FORM-INSTANCE"

Step5: Enter the condition as trim(to_char(sysdate,"DAY")) = 'SUNDAY'

Step6: Click on Validate button, it should show the evaluation as True

Step7: Save the changes



Form Personalizations (Sales Orders)

Function Name: ONT_OEXOEORD

Form Name: OEXOEORD

□ Show Debug Messages

Seq	Description	Enabled
10	Warn Users Entering Sales Orders on Sunday	☑
		□
		□
		□

Trigger Event: WHEN-NEW-FORM-INSTANCE (You can enter additional event names.)

Trigger Object:

Condition: trim(to_char(sysdate,'DAY'))='SUNDAY'

□ Fire in Enter-Query Mode

Context

Level	Value
Site	

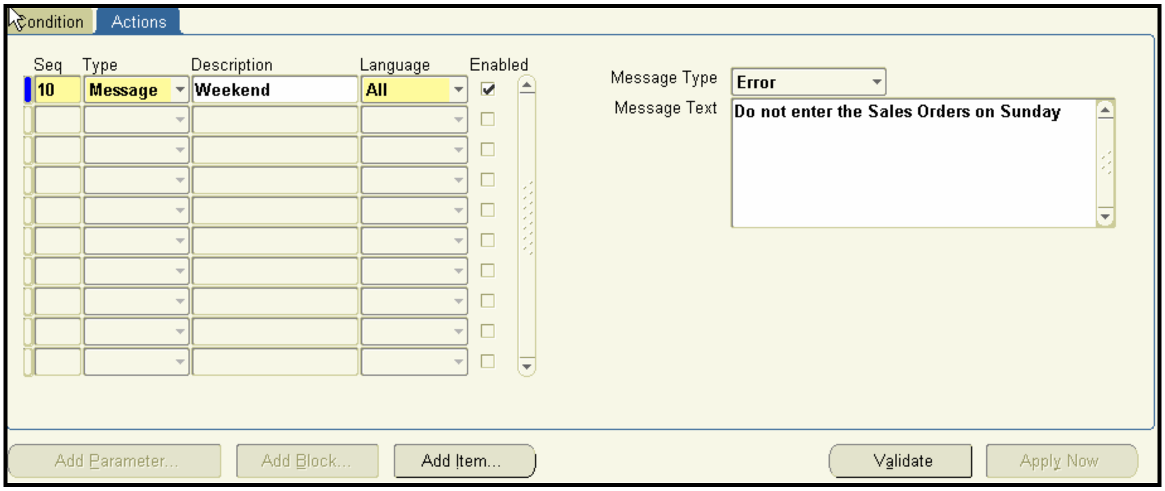
Figure 12: Example 2 Warn users not to enter Sales Orders on Sunday

Step8: On Actions tab, enter or select the following values

- Sequence = 10
- Type = Message
- Description = weekend
- Language = All
- Enabled = Yes

Step9: On the right side of Actions tab, enter or select the following values

- Message Type = Error
- Message Text = Do not enter the Sales Orders on Sunday



Seq	Type	Description	Language	Enabled
10	Message	Weekend	All	☑

Message Type: Error

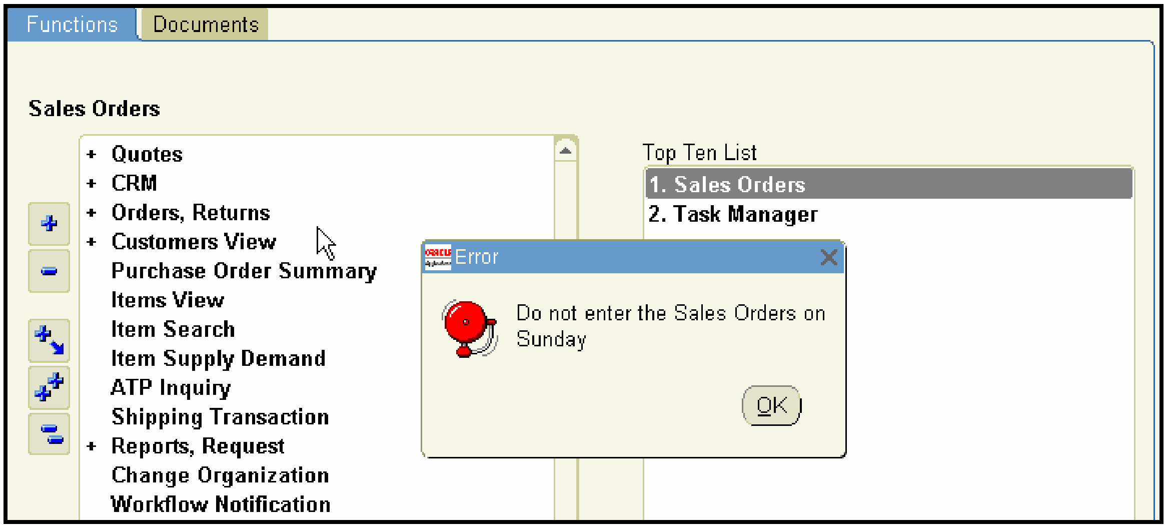
Message Text: Do not enter the Sales Orders on Sunday

Figure 13: Example 2 – Entering values for the Action Type "Message"

Step10: Click on the "Validate" button, then click on "Apply Now" button

Step11: Close both the forms and reopen the Sales Order form

Result: While opening the Sales Order form, it should prompt the given message



Do not enter the Sales Orders on Sunday

Figure 14: Result of Example 2

Example3: Activate Special Menu on the Sales Order Header and invoke a custom form when clicked on that special menu

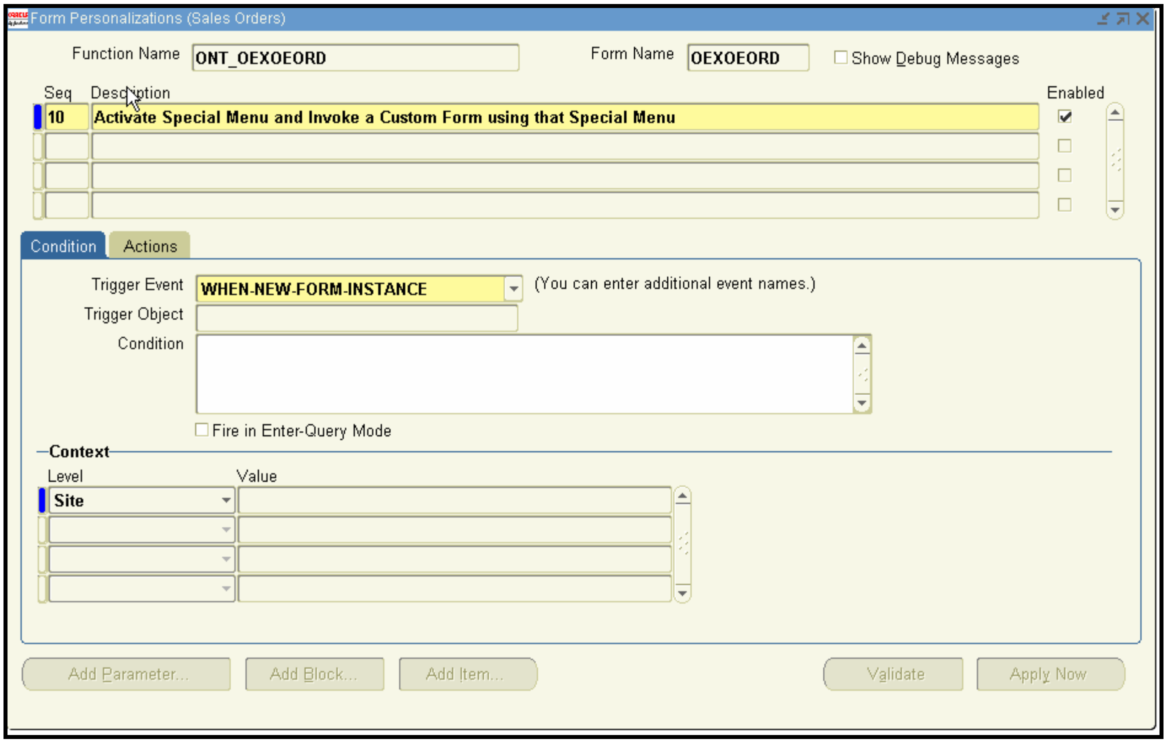
Step1: Open the Sales Order Form

Step2: Open the Personalization form using the navigation "Help → Diagnostics → Custom Code → Personalize"

Step3: Enter the sequence number as 10 and description "Activate Special Menu and Invoke a Custom Form using that Special Menu"

Step4: Select the Trigger Event as "WHEN-NEW-FORM-INSTANCE"

Step5: Save the changes



Form Personalizations (Sales Orders)

Function Name: ONT_OEXOEORD Form Name: OEXOEORD □ Show Debug Messages

Seq	Description	Enabled
10	Activate Special Menu and Invoke a Custom Form using that Special Menu	☑
		□
		□
		□

Condition | Actions

Trigger Event: WHEN-NEW-FORM-INSTANCE (You can enter additional event names.)

Trigger Object:

Condition:

□ Fire in Enter-Query Mode

Context

Level	Value
Site	

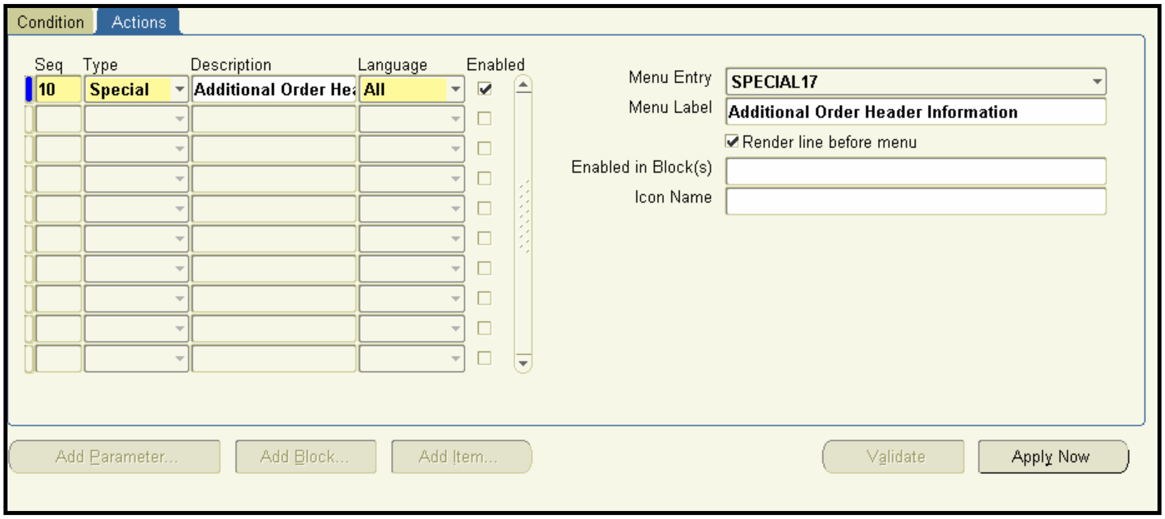
Figure 15: Example 3 Special Menu and AOL Builin

Step6: On Actions tab, enter or select the following values

- Sequence = 10
- Type = Special
- Description = Additional Header Information Special Menu
- Language = All
- Enabled = Yes

Step7: On the right side of Actions tab, enter or select the following values

- Menu Entry = SPECIAL7
- Menu Label = Additional Order Header Information
- Render line before menu = Yes



Seq	Type	Description	Language	Enabled
10	Special	Additional Order He:	All	☑

Menu Entry: SPECIAL17

Menu Label: Additional Order Header Information

☑ Render line before menu

Enabled in Block(s):

Icon Name:

Figure 16: Example 3 – Entering values for the Action Type "Special"

Step8: Click on the "Validate" button, then click on "Apply Now" button

Step9: Close both the forms and reopen the Sales Order form

Result: Check the Tools menu after opening the Sales Order form, there should be an additional menu showing the given prompt.

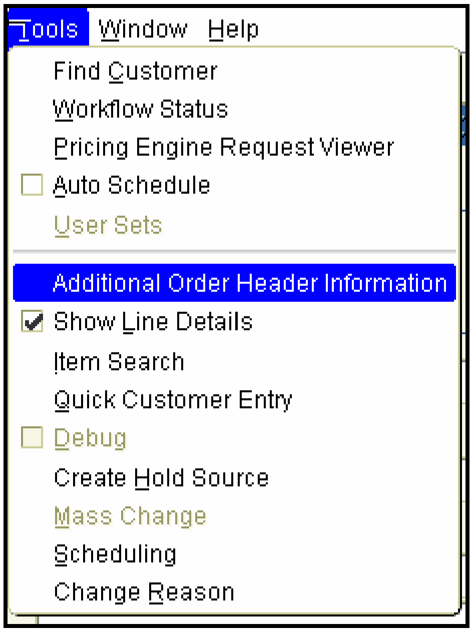


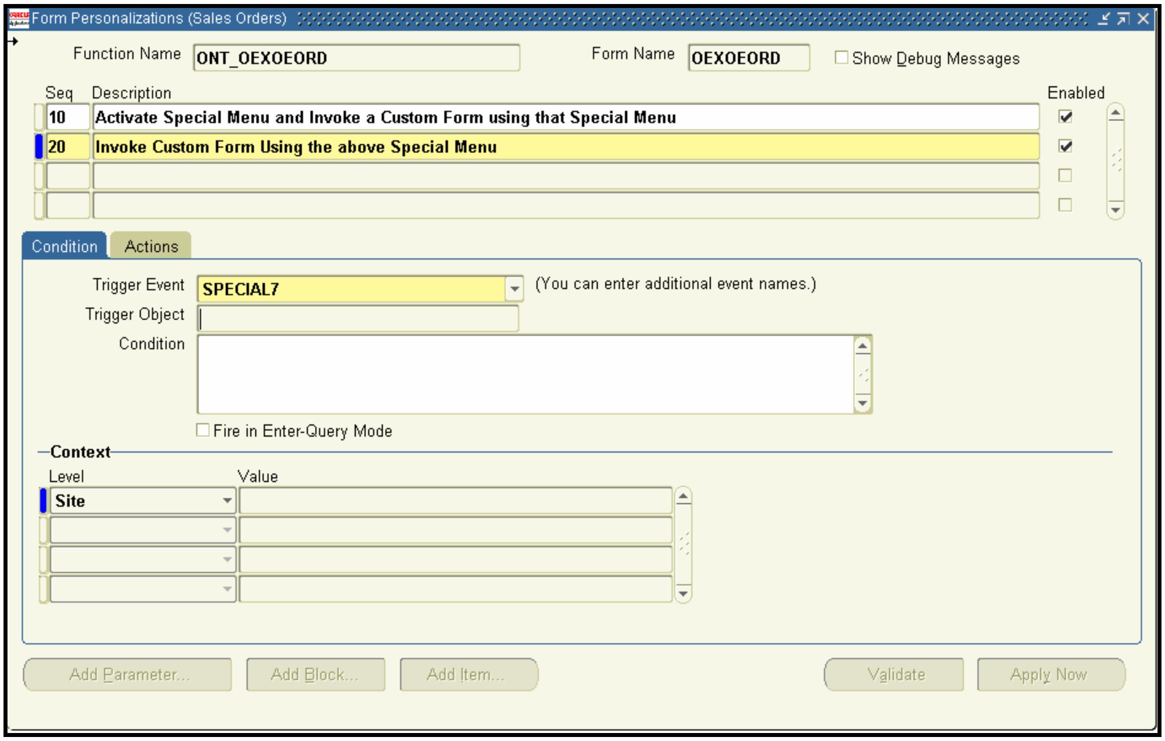
Figure 17: Result of Example 3 – Special Menu Activation

Step10: Open the personalization form and it should automatically query the existing rule to activate the special menu

Step11: Enter the sequence number as 20 and description "Invoke Custom Form Using the above Special Menu"

Step12: Select the Trigger Event as "SPECIAL7"

Step13: Save the changes



Form Personalizations (Sales Orders)

Function Name: ONT_OEXOEORD

Form Name: OEXOEORD

□ Show Debug Messages

Seq	Description	Enabled
10	Activate Special Menu and Invoke a Custom Form using that Special Menu	☑
20	Invoke Custom Form Using the above Special Menu	☑
		□
		□

Trigger Event: SPECIAL7 (You can enter additional event names.)

Trigger Object:

Condition:

□ Fire in Enter-Query Mode

Context

Level	Value
Site	

Figure 18: Example 3 – Invoking Custom via Special Menu

Step14: On Actions tab, enter or select the following values

- Sequence = 20
- Type = Builtin
- Description = Call Custom form
- Language = All
- Enabled = Yes

Step15: On the right side of Actions tab, enter or select the following values

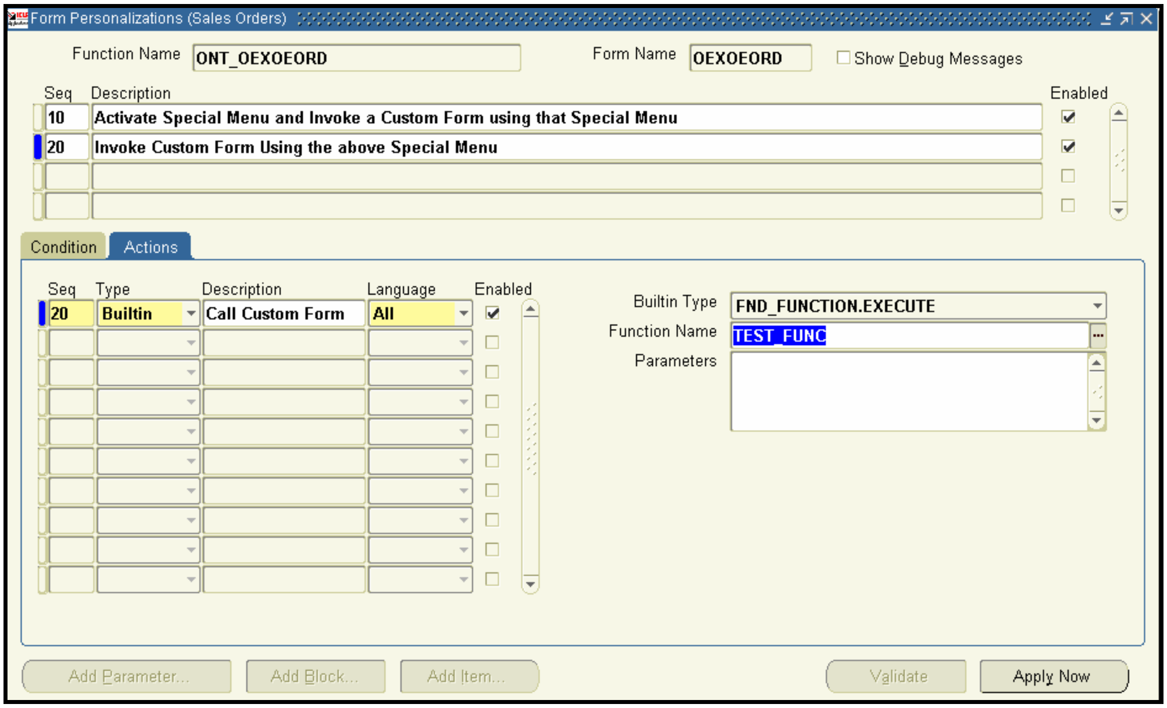
- Builtin Type = FND_FUNCTION.EXECUTE
- Function Name = TEST_FUNC

Note: Define TEST_FUNC as form function, associate with a custom form and add the TEST_FUNC function to the menu of "Order Management Super User" responsibility.

Step16: Validate, Apply and Save the changes

Step17: Close both the forms and reopen the Sales Order form

Result: When clicked on the Tools → Addition Header Information, it should invoke the custom form attached to the TEST_FUNC function.



Function Name: ONT_OEXOEORD

Form Name: OEXOEORD

□ Show Debug Messages

Seq	Description	Enabled
10	Activate Special Menu and Invoke a Custom Form using that Special Menu	☑
20	Invoke Custom Form Using the above Special Menu	☑

Seq	Type	Description	Language	Enabled
20	Builtin	Call Custom Form	All	☑

Builtin Type: FND_FUNCTION.EXECUTE

Function Name: TEST_FUNC

Parameters:

Figure 19: Example 3 – Entering Values for Action Type "Builtin"

Conclusion

The personalization feature allows the users to change the form behavior; this should be available only to the users who are authorized. Though this is easy and fast to implement, all the customization requirements like normal custom forms cannot be implemented. The design time activities like creating new items etc cannot be performed and implementing personalization requires the knowledge or experience of Oracle Developer and AOL.

On the whole, to the Oracle Applications customers, the personalization feature in release 11.5.10 offers an easy, fast and user friendly method to implement their specific requirements with less effort.