

# Advanced Security Controls without Customization (Forms Personalization in 11.5.10)

Joe Spina/Susan Behn  
Solution Beacon, LLC

SROAUG  
October 21, 2005  
[www.solutionbeacon.com](http://www.solutionbeacon.com)

# Introduction

- Solution Beacon, LLC
  - ◆ Joe Spina
  - ◆ Susan Behn

# Agenda

- Standard Application Security
  - ◆ Menu Driven Security
  - ◆ HRMS Security Options
  - ◆ Oracle Label Security (OLS)
- Personalized and Custom Security Options
  - ◆ CUSTOM.pll
  - ◆ Logical Apps AppsRules
  - ◆ Forms Personalization – New for 11.5.10
- Additional Resources

# Standard Application Security

- Menu Driven Security
  - ◆ Group functions into logical menu structures
    - ◆ Include all functions for a logical group
    - ◆ Do not include sub functions
    - ◆ Create a functions with the parameter `QUERY_ONLY="YES"`
  - ◆ Group concurrent requests into request groups
  - ◆ Assign request groups to responsibilities
  - ◆ Assign menus to responsibilities
  - ◆ Assign responsibilities to users
- Problem: Seeded menus, request groups and responsibilities do not always comply with regulations regarding separation of duties

# Standard Application Security

- Responsibilities – Additional Security
  - ◆ Function Menu Exclusions – System Administrator excludes functions and subfunctions (form,tab,button)
  - ◆ Excluded Items (Self Service Only)
    - ◆ Provides column level security
    - ◆ Exclude specific attributes
  - ◆ Securing Attributes (Self Service Only)
    - ◆ Provides row level security
    - ◆ iExpenses example- TO\_PERSON\_ID and ICX\_HR\_PERSON\_ID are set to the employee person id

# Standard Application Security

- Standard HRMS Security
  - ◆ User is assigned to a responsibility that is assigned a security profile which included a business group
  - ◆ Separate responsibilities are required for each security profile
- Security Groups Enabled Security
  - ◆ Responsibility = what you can do
  - ◆ Security Profile = who you can do it to
  - ◆ A responsibility includes multiple security groups
  - ◆ Assign Security Profile form - Links a security profile to a user's responsibility and business group
- Only one business group can be viewed at a time
- Security profiles are not easily changed

# Standard Application Security

## ■ Oracle Label Security

- ◆ Provides powerful row level security in Oracle9i
- ◆ Data is secured with a combination of data labels and up to 3 associated components (levels, compartments, groups)
- ◆ Policies define security for a collection of labels
- ◆ Requires PL/SQL to design and assign labels and assign them to tables
- ◆ Requires significant time to design and maintain
- ◆ OLS does not have a front end interface

# Security Requirement

**What do we need?**

**...to be able to secure any data item on any form for any user or group of users**

# Advanced Security Solutions

## How can we get what we need?

- CUSTOM.pll - Library available in \$AU\_TOP/resource is modified to provide customizations to Oracle Application forms
- Logical Apps AppsRules
- Forms Personalization – Forms 6i – 11.5.10
- OA Framework Personalization – Java Forms (not included in this presentation)

# CUSTOM.pll

- Library available in \$AU\_TOP/resource is modified to provide customizations to Oracle Application forms
- Use forms builder *6i* to modify package body
- Examples
  - ◆ Hide fields, tabs
  - ◆ Make fields required
  - ◆ Restrict update or insert
  - ◆ Change prompts, tab labels
  - ◆ Alter LOVs
  - ◆ Create zooms and tool bar menu selections
  - ◆ Validate and format
  - ◆ Almost anything you can do in PL/SQL
- Reference: Application Developer Guide – Chapter 28

# CUSTOM.pll – Methodology

## ■ Challenges

- ◆ Traditional implementation of customizations in CUSTOM.pll only allows one developer at a time to make modifications
- ◆ Size limitations can be an issue
- ◆ Keeping code modular is difficult
- ◆ Testing requirements can be significant for subsequent modifications to CUSTOM.pll

# CUSTOM.pll – Methodology

## Multi-Developer Solution – Supplier Form Example

- Create a separate library (.pll) for each form to be customized

### Package Spec

```
PACKAGE XXXXXAPXVDMVX IS
  Procedure event(event_name VARCHAR2);
END;
```

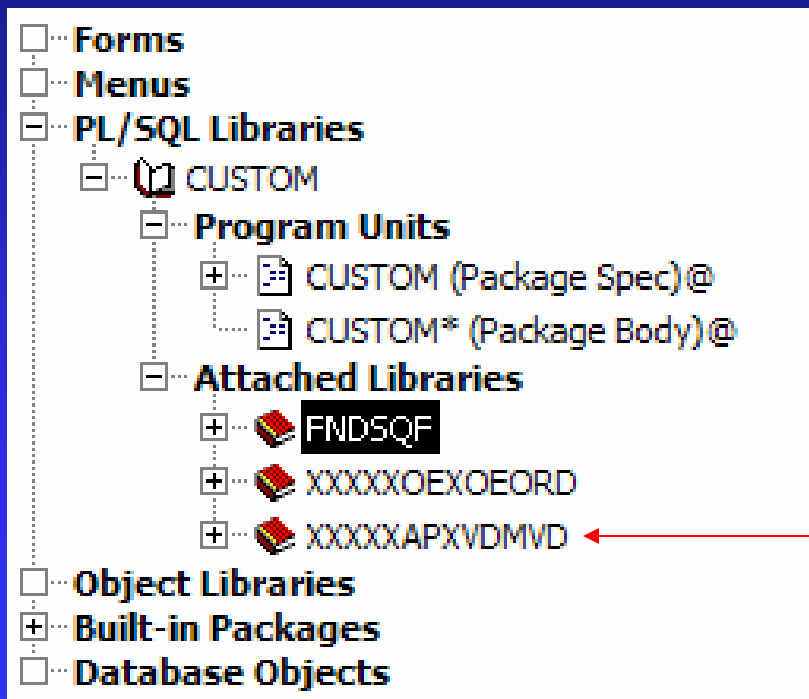
### Package Body

```
PACKAGE BODY XXXXXAPXVDMVX
IS
PROCEDURE event (event_name VARCHAR2) IS
BEGIN
  IF event_name = 'WHEN-NEW-FORM-INSTANCE' THEN
    \*Make the vendor type field required*\
    APP_ITEM_PROPERTY2.SET_PROPERTY('VENDOR_TYPE_DISP',REQUIRED,PROPERTY_TRUE);
  END IF;
END event;
END XXXXXAPXVDMVX;
```

# CUSTOM.pll – Methodology

Multi-Developer Solution – Supplier Form Example

- Attach your new library to CUSTOM.pll



Navigator view of  
CUSTOM.pll

Your new library

# CUSTOM.pll – Methodology

## Multi-Developer Solution – Supplier Form Example

- Add a call to your new library in CUSTOM.pll

### Package Body of CUSTOM.pll

```
Form_name      varchar2(30) := name_in('system.current_form');  
Begin  
  If form_name = 'APXVDMVD' THEN  
    xxxxapxvdmvd.event(event_name);  
  Elself form_name = 'OEXOEOORD' THEN  
    xxxxoexoeord.event(event_name);  
  end if;  
end event;
```

# CUSTOM.pll – More Examples

\\*Force Upper Case for Supplier Name\*\

```
APP_ITEM_PROPERTY2.SET_PROPERTY('VNDR.VENDOR_NAME_MIR',CASE_RESTRICTION,UPPERCASE);
```

\\* Hide the tax payer id\*\

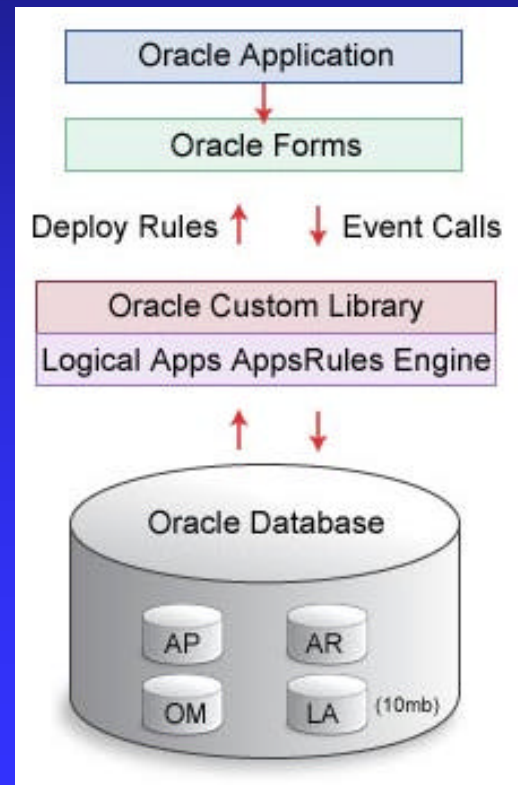
```
APP_ITEM_PROPERTY2.SET_PROPERTY('VNDR.NUM_1099_MIR',DISPLAYED,PROPERTY_OFF)
```

\\*Change the prompt\*\

```
APP_ITEM_PROPERTY2.SET_PROPERTY('VNDR.  
END_DATE_ACTIVE_MIR',PROMPT_TEXT,'Inactive Date')
```

# Logical Apps AppsRules

- Configurable Security for *6i* forms
- Centralized Rules Repository
- Front end interface to CUSTOM.pll
- Recommended for pre 11.5.10 installations



# Logical Apps AppsRules

Examples of National Institutes of Health Security Rules  
Implemented using AppsRules

- Security
  - ◆ Prevent update to fields and blocks
  - ◆ Hide Fields and Tabs
- List of Values on field without existing LOV
- Messages
- Change prompts
- Set default values
- Limit access for:
  - ◆ Responsibility
  - ◆ Profile Option
  - ◆ Email Address
  - ◆ User

# Logical Apps Apps Rules

## ■ AppsRules Security Tab

Business Rule Details

Form Name: **APXVDMVD**    Element Description: **Secure fields and alter I**    Event: **When New Form**

Block Name:     Field Name:

Security    Navigation    Messages    Default Values    List of Values    Field Attributes    SQL    AppsFlow

Type	Block/Tab	Field Name	Case	Required	No Update	No Insert	No Delete	Hide	Active
Field	VNDR	INVOICE_MATCHING_I		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Block	VNDR	...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Default Where: **where vendor\_type\_disp != 'Employee'**

Order By:

Done



# Forms Personalization

- Personalizations are declarative changes to forms delivered with the E-Business Suite
- Forms Personalizations declaratively alter the behavior of Forms
- User must understand Forms and PL/SQL
- Most changes traditionally done using CUSTOM.pll can be accomplished using Forms Personalization
  - ◆ Must use CUSTOM.pll to alter LOVs
  - ◆ CUSTOM.pll allows all PL/SQL capabilities, builtins and SQL
- Forms Personalizations are effective immediately – no compiling
- Forms Personalizations fire prior to CUSTOM.pll for the same event

# Forms Personalization Profile Options

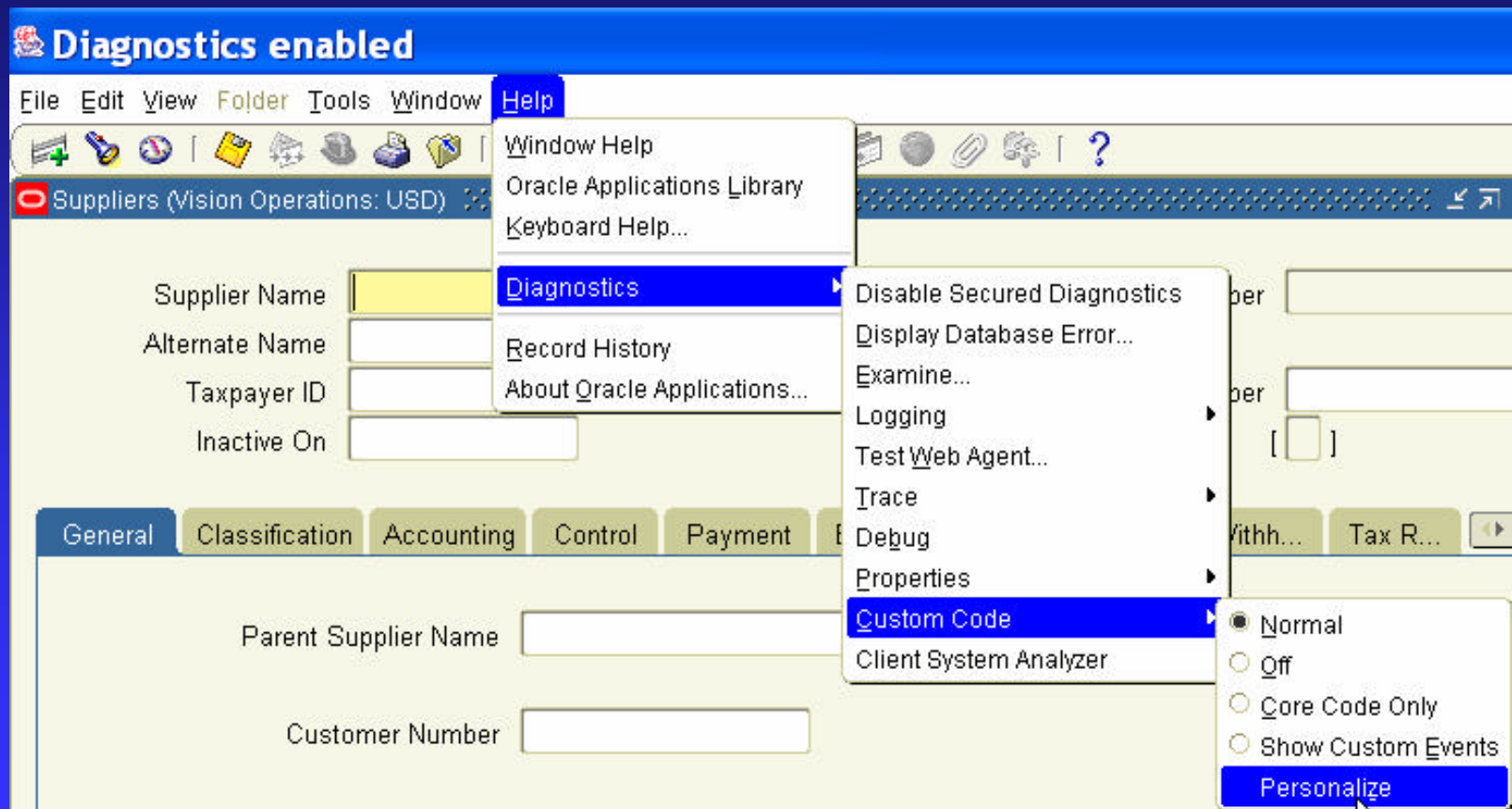
- Today's Demo environment – 11.5.10 with CU1 patch applied  
(Forms Personalization in CU1 changed significantly with CU1 patch)
- Set Profile Option *Hide Diagnostics menu entry* to *No*  
Yes will hide the diagnostics menu.
- Profile Option *Utilities: Diagnostics* – if set to *No*, apps password is required

The screenshot shows a window titled 'System Profile Values' with a table of profile options. The table has columns for Profile, Site, Application, Responsibility, and User. The 'Site' column contains values 'No', 'No', and 'No' for the three rows shown.

Profile	Site	Application	Responsibility	User
FND: Diagnostics	No			
Hide Diagnostics menu entry	No			
Utilities:Diagnostics	No			

# Forms Personalization – Access

- Access the form or function needing personalization
- Help → Diagnostics → Custom Code → Personalize



# Forms Personalization

Form Personalizations (Suppliers)

Function Name  Form Name  Debug Mode

Seq	Description	Level	Enabled
10	Susan's Test	Form	<input checked="" type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

**Condition** | **Actions**

Trigger Event  (You can enter additional event names.)

Trigger Object

Condition

Processing Mode

**Context**

Level	Value
User	SBEHN

# Forms Personalization – Rules

- Personalization Rule Header
  - ◆ Sequence
    - ◆ Rules run in sequence
    - ◆ Sequence numbers (1-100) are not unique
  - ◆ Description – free form entry
  - ◆ Level – Form or Function (CU1 patch)
  - ◆ Debug mode
    - ◆ Off
    - ◆ Step-by-Step – shows events impacted by rule (CU1 patch)
    - ◆ Show Debug Messages – shows messages with type = debug
  - ◆ Enabled – checked

# Forms Personalization – Conditions Tab

## Trigger Events

- Use generic trigger events available in most forms
- Use specific events unique to the form (with caution)
- Find events using Help → Diagnostics → Custom Code → Show Custom Events
- Trigger Events are not validated from the LOV

# Forms Personalization – Conditions Tab

Trigger Events – Generic to *almost* all forms

- WHEN-NEW-FORM-INSTANCE
  - ◆ Security rules
  - ◆ Navigation rules
  - ◆ Visual attributes
  - ◆ Avoid message rules at this level
- WHEN-NEW-BLOCK-INSTANCE
  - ◆ Same as WHEN-NEW-FORM-INSTANCE
  - ◆ Message rules
- WHEN-NEW-RECORD-INSTANCE
  - ◆ Default values

# Forms Personalization – Conditions Tab

Trigger Events – Generic to *almost* all forms

- WHEN-NEW-ITEM-INSTANCE

- ◆ Message rules
- ◆ Default values dependent on entry of another item

- WHEN-VALIDATE-RECORD

- ◆ Populate hidden fields
- ◆ Additional validations

- SPECIALn

- ◆ Populate tools menu (MENU1-15) (CU1 patch)
- ◆ Populate tools menu (SPECIAL 1-15)
- ◆ Populate reports menu (SPECIAL 16-30)
- ◆ Populate actions menu (SPECIAL 31-45)

# Forms Personalization – Conditions Tab

Trigger Events – Generic to *almost* all forms

- ZOOM – recommend using MENU<sub>n</sub> or SPECIAL<sub>n</sub> rather than zoom
- KEY-Fn

# Forms Personalization – Conditions Tab

## Trigger Object

- Required if LOV is available
- Requires Block Name
  - ◆ WHEN-NEW-BLOCK-INSTANCE
  - ◆ WHEN-NEW-RECORD-INSTANCE
  - ◆ WHEN-VALIDATE-RECORD
- Requires Block.field name
  - ◆ WHEN-NEW-ITEM-INSTANCE
- May be required for other events specific to form

# Forms Personalization – Conditions Tab

## Conditions

- Optional SQL code fragment to limit scope of rule
- References bind variables (:block.field)
- Examples
  - ◆ Use to limit scope based on profile option values
  - ◆ GL Journal Entry – Remind users to change the period name the first 10 days of the fiscal year

The screenshot shows the 'Conditions' tab in the Oracle Forms Personalization tool. It contains the following fields:

- Trigger Event:** A dropdown menu set to 'WHEN-NEW-ITEM-INSTANCE'. A note next to it says '(You can enter additional event names.)'
- Trigger Object:** A text box containing 'BATCH.DEFAULT\_PERIOD\_NAME'.
- Condition:** A large text area containing the SQL code: `to_char(sysdate,'MON') = 'SEP' and to_char(sysdate,'DD') < 20`. There is a vertical scrollbar on the right side of this text area.
- Processing Mode:** A dropdown menu set to 'Not in Enter-Query Mode'.

At the bottom left of the form, there is a label '-Context'.

# Forms Personalization – Conditions Tab

Context – who does this rule apply to?

- Multiple scope rows are allowed
- Level at which the rule will apply
  - ◆ Site
  - ◆ Responsibility
  - ◆ User – Use this for testing rules
  - ◆ Industry (For future use)
- Value – choose from LOV

**Tip:** For initial development, set scope to your user id

Level	Value
Industry	SBEHN
Site	
Responsibility	
User	

# Forms Personalization – Property Actions

- Sequence
  - ◆ Rules will run in sequence
  - ◆ Sequence number not unique

- Type

- ◆ Property
- ◆ Message
- ◆ Built-in
- ◆ Menu

Seq	Type	Description	Language	Enabled
10	Property		All	<input checked="" type="checkbox"/>
	Message			<input type="checkbox"/>
	Builtin			<input type="checkbox"/>
	Menu			<input type="checkbox"/>

- Description
- Language – use when changing prompts for a specific language
- Enabled – checked

# Forms Personalization – Property Actions

- Property Type Fields
  - ◆ Object Type
    - ◆ Item , Window, Block, Tab Page, Parameter
    - ◆ Radio Button, View, :GLOBAL Variable, Canvas
    - ◆ LOV, Local Variable (CU1 patch)
  - ◆ Target Object – LOV will include valid values for most types
  - ◆ Property Name – LOV available
  - ◆ Value
- Use *Select by Text* button to select the target object by prompt name
- Use *Get Value* button to get the current value

# Forms Personalization – Property Strings

- Rules for fields that accept strings
  - ◆ Start with =
    - ◆ String evaluated at run time
    - ◆ Can use bind variables, operators, etc
    - ◆ Can use server side functions without out variables
    - ◆ Prior to CU1 patch, SQL statements starting with =Select require "A" alias (=Select meaning A from fnd\_lookups where...\_
  - ◆ Does not start with =
    - ◆ String is taken as a literal exactly as you type it

# Forms Personalization – Property Actions Example

Force Upper Case for the Vendor Name

- Type = Property
- Object Type = Item
- Target Object = VNDR.VENDOR\_NAME\_MIR
- Property Name = CASE\_RESTRICTION
- Value = UPPERCASE

The screenshot shows the 'Actions' tab in Oracle Forms Personalization. A table lists the configured actions, and a configuration panel on the right shows the details for the selected action.

Seq	Type	Description	Language	Enabled
10	Property	Vendor Name Upper	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Configuration Panel:

- Select By Text...
- Object Type: **Item**
- Target Object: **VNDR.VENDOR\_NAME\_MIR**
- Property Name: **CASE\_RESTRICTION**
- Value: **UPPERCASE**
- Get Value

# Forms Personalization – Property Actions Example

Change the window title

- Type = Property
- Object Type = Window
- Target Object = VENDOR
- Property Name = TITLE
- Value = Suppliers (Oracle Open World)

The screenshot shows a configuration window for a Property Action. At the top is a button labeled "Select By Text...". Below it are four fields:

- Object Type:** A dropdown menu with "Window" selected.
- Target Object:** A text field containing "VENDOR".
- Property Name:** A text field containing "TITLE".
- Value:** A text area containing "Suppliers (Oracle Open World)".

At the bottom of the form is a button labeled "Get Value".

# Forms Personalization – Actions Example

Prevent Insert in the Bank Accounts Tab  
(require users to do this in Bank setup)

- Type = Property
- Object Type = Block
- Target Object = VNDR\_USES
- Property Name = INSERT\_ALLOWED
- Value = False

The screenshot shows a configuration window for a form action. At the top is a button labeled "Select By Text...". Below it are four rows of configuration fields:

Object Type	Block
Target Object	VNDR_USES
Property Name	INSERT_ALLOWED
Value	False

At the bottom of the window is a button labeled "Get Value".

# Forms Personalization – Variables

## Global Variables

- Used to pass values between forms
- Max length is 255 bytes
- Prepend the name of the variable with XX

## Local variables

- Used when you need to refer to a variable multiple times
- Specific to local form
- Max length is 4000 bytes
- Prepend the name of the variable with XX

# Forms Personalization – Property Actions Example

Set a global variable to the value of the email address in FND\_USERS and display this value in a message

- Type = Property
- Object Type = :GLOBAL Variable
- Target Object = XX\_USER\_EMAIL
- Property Name = VALUE
- Value = 

```
=SELECT Nvl(Email_Address,'NO_EMAIL')  
FROM fnd_user  
WHERE user_id = fnd_global.user_id
```

# Forms Personalization – Property Actions Example

## ■:GLOBAL Variable

Object Type	:GLOBAL Variable
Target Object	XX_USER_EMAIL
Property Name	VALUE
Value	=SELECT Nvl(Email_Address,

Editor

```
=SELECT Nvl(Email_Address,'NO_EMAIL_ADDRESS')  
FROM fnd_user  
WHERE user_id = fnd_global.user_id
```

## ■Message

Message Type	Show
Message Text	=:global.xx_user_email

# Forms Personalization – Messages

- Message Type Fields
  - ◆ Message Type
    - ◆ 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
  - ◆ Message Text
- Do not use messages for WHEN-NEW-FORM events

# Forms Personalization – Message Examples

## ■ Debug message

The screenshot shows the 'Condition' tab of the Forms Personalization tool. It features a table with columns for Seq, Type, Description, Language, and Enabled. The table contains three rows of messages. To the right of the table, there are fields for 'Message Type' and 'Message Text'.

Seq	Type	Description	Language	Enabled
19	Message	Debug 1	All	<input checked="" type="checkbox"/>
20	Property	Hide tax id	All	<input checked="" type="checkbox"/>
21	Message	Debug 2	All	<input checked="" type="checkbox"/>

Message Type: **Debug**  
Message Text: **Debug 1 - before hiding tax id**

## ■ Training reminders

The screenshot shows the 'Form Personalizations (Enter Journals)' window. At the top, there are fields for 'Function Name' (GLXJEENT\_A), 'Form Name' (GLXJEENT), and 'Debug Mode' (Show Debug Messages). Below this is a table with columns for Seq, Description, Level, and Enabled. The table contains one row of messages. Below the table, there are tabs for 'Condition' and 'Actions'. The 'Actions' tab is active, showing a detailed view of message 10.

Function Name: **GLXJEENT\_A**    Form Name: **GLXJEENT**    Debug Mode: **Show Debug Messages**

Seq	Description	Level	Enabled
10	Message to check period name first 10 days of fiscal year	Function	<input checked="" type="checkbox"/>

Message Type: **Show**  
Message Text: **Reminder - The period name defaults to the current period. Please remember to change the period when you are entering journals for the prior period.**

# Forms Personalization – Menu Example

Create a menu entry to submit payment history report

- Type = Menu (Prior to CU1 patch, Type = Special)
- Menu Entry = MENU1 – MENU15 or SPECIAL1-45
- Menu Label = Supplier Payment History
- Icon = null
- Enabled in Blocks = VNDR, SITE
  - ◆ Separate by comma
  - ◆ Use *Add Block* Button to choose blocks
- Note – this action only displays the menu entry.

Seq	Type	Description	Language	Enabled
10	Menu	Payment History	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Menu Entry	<input type="text" value="MENU1: Supplier Payment History"/>
Menu Label	<input type="text" value="Supplier Payment History"/>
	<input type="checkbox"/> Render line before menu
Icon Name	<input type="text"/>
Enabled in Block(s)	<input type="text" value="VNDR , SITE"/>
	<input type="button" value="Add Block..."/>

# Forms Personalization – Built-ins Example

Activate the menu entry to execute the concurrent request *Supplier Payment History*

- Trigger Event = MENU1

The screenshot shows the 'Form Personalizations (Suppliers)' window. At the top, the 'Function Name' is 'AP\_APXVDMVD', the 'Form Name' is 'APXVDMVD', and 'Debug Mode' is 'Off'. Below this is a table of personalization items:

Seq	Description	Level	Enabled
20	Property changes	Form	<input checked="" type="checkbox"/>
30	Show error message	Form	<input type="checkbox"/>
40	Menu entries	Form	<input checked="" type="checkbox"/>
50	Built in	Function	<input checked="" type="checkbox"/>

The 'Built in' item (Seq 50) is selected. Below the table, the 'Condition' tab is active. The configuration for this item is as follows:

- Trigger Event: MENU1 (You can enter additional event names.)
- Trigger Object: (Empty)
- Condition: (Empty)
- Processing Mode: Not in Enter-Query Mode
- Context:
  - Level: User
  - Value: SBEHN

At the bottom of the window, there are buttons for 'Insert 'Get' Expression...', 'Insert Item Value...', 'Validate', and 'Apply Now'.

# Forms Personalization – Built-ins Example

Activate the menu entry to run Supplier Payment History Report

- Type = Builtin
- Builtin Type = Launch SRS Form (CU1 patch)
- Program Name = Supplier Payment History

The screenshot shows the 'Actions' tab in the Oracle Forms Personalization tool. It features a table with columns for 'Seq', 'Type', 'Description', 'Language', and 'Enabled'. The first row is highlighted with a blue selection bar and contains the following values: Seq: 10, Type: Builtin, Description: Menu1, Language: All, and Enabled: checked. To the right of the table, there are two dropdown menus: 'Builtin Type' set to 'Launch SRS Form' and 'Program Name' set to 'Supplier Payment History'.

Seq	Type	Description	Language	Enabled
10	Builtin	Menu1	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Builtin Type: **Launch SRS Form**

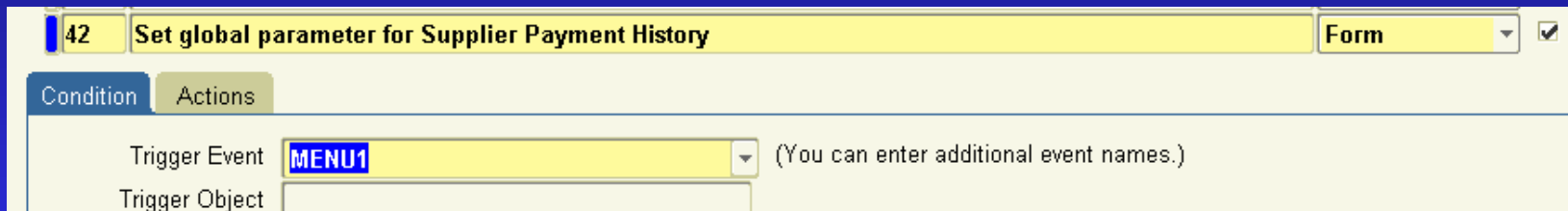
Program Name: **Supplier Payment History**

- Note: Parameters are not automatically passed

# Forms Personalization – Built-ins Example

Supplier Payment History Report – Passing Parameters

- Create a new rule with a sequence before menu execute
- Trigger event = MENU<sub>n</sub> or SPECIAL<sub>n</sub>



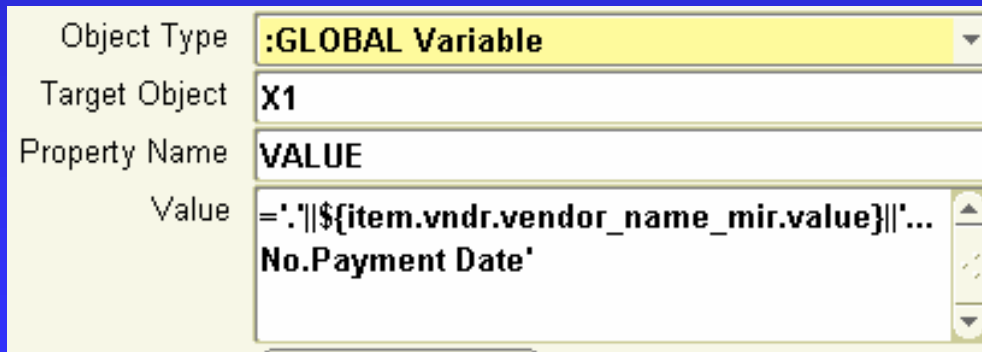
42 Set global parameter for Supplier Payment History Form

Condition Actions

Trigger Event **MENU1** (You can enter additional event names.)

Trigger Object

- Set :GLOBAL variable to value of parameters for report



Object Type **:GLOBAL Variable**

Target Object **X1**

Property Name **VALUE**

Value **= '||\${item.vndr.vendor\_name\_mir.value}||'...  
No.Payment Date'**

# Forms Personalization – Built-ins

Supplier Payment History Report – Passing Parameters

- Create a new rule for the Requests: Submit form

The screenshot shows the Oracle Forms Personalizations window for the 'Requests: Submit' form. The Function Name is 'FND\_FNDRSRUN' and the Form Name is 'FNDRSRUN'. The Debug Mode is set to 'Off'. A table lists personalization rules, with the first rule selected: Seq 20, Description 'Set parameters', Level 'Form', and Enabled checkbox checked. Below the table, the 'Condition' tab is active, showing a Trigger Event of 'WHEN-NEW-ITEM-INSTANCE', a Trigger Object of 'WORK\_ORDER.PARAMETERS', and a Condition of ':global.x1 is not null'. The Processing Mode is 'Not in Enter-Query Mode'.

The screenshot shows the 'Actions' tab of the Oracle Forms Personalizations window. The Object Type is 'Item', the Target Object is 'WORK\_ORDER.PARAMETERS', and the Property Name is 'VALUE'. The Value field contains the following SQL query: `=select nvl(#{item.global.X1.value},#{item.work_order.parameters.value}) from dual`. A red arrow points from the 'Condition' field in the previous screenshot to the 'Value' field in this screenshot.

Different syntax required

# Forms Personalization – Built-ins Example

Open a url

- Trigger Event = WHEN-NEW-FORM-INSTANCE
- Establish menu entry

The screenshot shows the 'Actions' tab in Oracle Forms Personalization. A table lists actions, with the first row selected:

Seq	Type	Description	Language	Enabled
10	Menu	Open a url	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Configuration fields for the selected action:

- Menu Entry: MENU2
- Menu Label: Google
- Render line before menu
- Icon Name: (empty)
- Enabled in Block(s): (empty)

- Trigger Event = MENU2

The screenshot shows the 'Actions' tab in Oracle Forms Personalization. A table lists actions, with the first row selected:

Seq	Type	Description	Language	Enabled
10	Builtin	open google	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Configuration fields for the selected action:

- Builtin Type: Launch a URL
- Argument: www.google.com
- Target: BLANK (circled in red)

CU1 Patch

# Forms Personalization – Built-ins Example

Launch a function (CU1 patch) to view payment history form

- Establish the Menu entry
  - ◆ Trigger Event = WHEN-NEW-FORM-INSTANCE
  - ◆ Action Type = Menu
  - ◆ MENU3 = Payment History

Seq	Type	Description	Language	Enabled
10	Menu	Payment History	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Menu Entry: **MENU3: Payment History**

Menu Label: **Payment History**

Render line before menu

Icon Name:

Enabled in Block(s):

# Forms Personalization – Built-ins Example

Launch a function to view payment history form

- Trigger Event = MENU3
- Set the :GLOBAL Variable
- Launch a Function

Condition Actions

Seq	Type	Description	Language	Enabled
10	Property	Global - vendor id	All	<input checked="" type="checkbox"/>
20	Builtin	Payment inquiry	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Select By Text...

Object Type: :GLOBAL Variable

Target Object: XX\_VENDOR\_NAME

Property Name: VALUE

Value: =:VNDR.VENDOR\_NAME\_MIR

Condition Actions

Seq	Type	Description	Language	Enabled
10	Property	Global - vendor id	All	<input checked="" type="checkbox"/>
20	Builtin	Payment inquiry	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Builtin Type: Launch a Function

Function Code: AP\_APXPWALL\_VIEW

Function Name: View Payment Overview

Parameters:

# Forms Personalization – Built-ins Example

Launch a function – target function rules

- Populate query find variable if the global variable is not null
  - ◆ Trigger Event = WHEN-NEW-ITEM-INSTANCE
  - ◆ Trigger Object = use the first item on the form

The screenshot shows the 'Form Personalizations' window for 'View Payment Overview'. The function name is 'AP\_AXPWALL\_VIEW' and the form name is 'APXPWALL'. The debug mode is set to 'Off'. A table lists the personalization rules, with the first rule selected:

Seq	Description	Level	Enabled
10	Execute query if vendor id is populated	Form	<input checked="" type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

Below the table, the configuration for the selected rule is shown under the 'Condition' tab:

- Trigger Event: WHEN-NEW-ITEM-INSTANCE (You can enter additional event names.)
- Trigger Object: PAYMENT\_QF.START\_CHECK\_NUMBER
- Condition: :global.XX\_VENDOR\_NAME IS NOT NULL

# Forms Personalization – Built-ins Example

Launch a function – target function rules

- Populate query find variable if the global variable is not null
  - ◆ Trigger Event = WHEN-NEW-BLOCK-INSTANCE
  - ◆ Action Type = Property

The screenshot shows the Oracle Forms Personalization tool interface. On the left, a table lists function rules:

Seq	Type	Description	Language	Enabled
10	Property	populate vendor id	All	<input checked="" type="checkbox"/>
11	Builtin	force query exec	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

On the right, configuration options are shown:

- Select By Text...
- Object Type: **Item**
- Target Object: **PAYMENT\_QF.DISPLAYED\_VENDOR\_NAME**
- Property Name: **VALUE**
- Value: `${item.global.xx_vendor_name.value}`
- Get Value

At the bottom, there are buttons: Insert 'Get' Expression..., Insert Item Value..., Validate, and Apply Now. A red arrow points from the 'CU1 Patch' box to the 'Insert 'Get' Expression...' button.

CU1 Patch

# Forms Personalization – Built-ins Example

Launch a function – target function rules

- Execute the DO\_KEY('NEXT\_BLOCK') built in to force query execution
  - ◆ Trigger Event = WHEN-NEW-BLOCK-INSTANCE
  - ◆ Action Type = Builtin

The screenshot shows the 'Actions' tab in Oracle Forms Personalization. It contains a table with the following data:

Seq	Type	Description	Language	Enabled
10	Property	populate vendor id	All	<input checked="" type="checkbox"/>
11	Builtin	force query exec[	All	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

To the right of the table, there are two configuration fields:

- Builtin Type: DO\_KEY
- Argument: NEXT\_BLOCK

# Forms Personalization – Built-ins

## Other Builtins

- GO\_BLOCK
- GO\_ITEM
- RAISE\_FORM\_TRIGGER\_FAILURE
- FORMS\_DDL
- EXECUTE\_TRIGGER (CU1 patch)
- SYNCHRONIZE (CU1 patch)

# 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
- Use Help → Diagnostics → Custom → Show Custom Events to determine what events are firing
- See MetaLink note 279034.1 for special rules for forms with folders
- After upgrades, go to the personalization for each form and choose Tools → Validate All
- Use debug message before and after events
- Initialize global variables to null in the navigator form using the WHEN-FORM-NAVIGATE trigger event

# Forms Personalization – Tips

- 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 dependant 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
- Set global values to null in the navigator form using the WHEN-FORM-NAVIGATE trigger

# Customizations / Personalizations

## WARNING

Customizations or Personalizations, whether they are protected or non protected, allow you to fundamentally change the behavior of the application. This could interfere with intended functionality.

**Use with caution!**

**TEST! TEST! TEST! TEST! TEST!**

# Customizations / Personalizations

**...THEN TEST IT AGAIN!**

# Forms Personalization – Moving to another instance

- 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>
```

# Other Sources of Information

- *Oracle Applications User Interface Standards for Forms-Based Products*
- *Oracle Applications Developer's Guide*
- *Oracle Applications System Administrator's Guide*
- *Oracle Applications User Guide*
- MetaLink note 279034.1 – Forms Personalization
- *Configuring, Reporting and System Administration in HRMS*
- *Oracle Self ServiceWeb Applications Implementation Manual*
- [www.solutionbeacon.com](http://www.solutionbeacon.com) – newsletters, free tools, white papers and presentations, Vision access

# Thank you!

If you have any questions or comments please contact:

Susan Behn     [sbehn@solutionbeacon.com](mailto:sbehn@solutionbeacon.com)  
Joe Spina       [jspina@solutionbeacon.com](mailto:jspina@solutionbeacon.com)

For free Release 11i Tools and helpful information,  
please visit our website at:

[www.solutionbeacon.com](http://www.solutionbeacon.com)

*Real Solutions for the Real World.®*