Forms Personalization

Presented by

Talbott Jones
Introduction

• Talbott Jones
  – Consultant for Colibri Information Management
  – BSBA 1978 – Old Dominion University
  – MBA 1990 – Old Dominion University
  – CPA 1985 – Virginia
  – 1997 Began working with Oracle Applications
    • General Ledger
    • Payables
    • Assets
    • Purchasing
    • Inventory
  – tjones@Colibrilimited.com
• Provides services “Exclusively” for Oracle Applications and RDBMS
• Each consultant has a minimum of 8 years Oracle experience.
• Web Site
Topics

• Basics
• Create Personalizations
  – Field Required
  – Insert Default
  – Display Message
  – Hide Button
  – Populate Field With One Of Three Possible Values
  – Display Warning
• Documentation
• Migration
Basics - Access

• Access to Forms Personalization
  – Requires Apps Password
  
  or

  – Profile setting Utilities:Diagnostics = Yes

• Access to the form being Personalized
Access

- Navigate to the form being personalized
  - Help
  - Diagnostics
  - Custom Code
  - Personalize
Forms Personalization Form

Form Personalizations (Purchase Orders)

Function Name: PO_POXPOEPO
Form Name: POXPOEPO
Debug Mode: Off

Seq | Description                  | Level     | Enabled
--- |------------------------------|-----------|--------
1   | Revision Required            | Function  | ✔️
2   | Revision Default to most recent | Form     | ✔️
3   | Item Pending Effectivity     | Form      | ✔️
4   | ASL Lookup                    | Form      | ✔️

Condition | Actions
--- | -------------------------------
Trigger Event: WHEN_VALIDATE_RECORD (You can enter additional event names.)
Trigger Object: PO_LINES
Condition: Not in Enter-Query Mode

Processing Mode: Not in Enter-Query Mode
Context: Site
Level: Value

Insert 'Get' Expression...  Insert Item Value...  Validate  Apply Now
# Personalization Header

- Function Name – How you got here
- Form Name – How Oracle identifies this form
- Input Area
  - Seq – Order in which personalizations are performed
  - Description
  - Level – Form / Function

<table>
<thead>
<tr>
<th>Seq</th>
<th>Description</th>
<th>Level</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Revision Required</td>
<td>Function</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Revision Default to most recent</td>
<td>Form</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Item Pending Effectivity</td>
<td>Form</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>ASL Lookup</td>
<td>Form</td>
<td>✓</td>
</tr>
</tbody>
</table>
• Select Form or Function
  – Form if this personalization is to be done every time the form is used
  – Function, if just when this function is being called.

A form can be called by more than one function. This helps you be more selective about when a Forms Personalization is invoked.
• Form / Function – Does it matter?
  – If you are not sure where a form is used, select Function.
  – If you want to cover every possible use of the form, select form.
  – If you use Function and look at the form’s personalizations in another function, you will see gaps in your list.
Condition Tab

- Trigger Event: WHEN-VALIDATE-RECORD
- Trigger Object: PO_LINES
- Processing Mode: Not in Enter-Query Mode
- Context:
  - Site
  - [Additional Options]
Condition Tab Elements

• **Trigger Event** — What event will invoke this personalization?

• **Trigger Object** — The event relates to this object.

• **Condition** — What must be true to allow your personalization to be invoked?

• **Processing Mode** — This personalization works in Query Mode / Not in Query Mode or in either mode.
  – We will not be discussing Query Mode Personalizations today.
Trigger Events

This event invokes your personalization

- New form
- Navigate to a new block
- Navigate to a new item
- Validate a record
- Menus and Special
  - (not covered today)
Trigger Object

- Trigger Object relates to the Trigger Event
  - If the event is a new block, the object is a block.
  - If the event is a new item, the object is a block and item combination.
  - Some events do not require an object.
Trigger Object

• How do you know what to use for a block or item?

Focus on desired field.

Help
Diagnostics
Examine
Careful With Examine!

• Look but don’t touch.
Condition

• Null (blank) means every time.
• You can add restrictions.
• Data from the form is referenced by a colon “:”.
• Partial SQL statements

Below, the condition says “if the lot number is 0”.

Block = LOT_ENTRY
Item = LOT_NUMBER
Format
:<BLOCK>.<ITEM>
Processing Mode

Forms Personalizations can be done in...

- Query Mode
- Not in Query Mode
- Both

Today, we will be talking about “Not in Query Mode”.
Context

For whom does the personalization work?

• **Site** – Everyone
• **Responsibility** – Designate a responsibility in the value column
• **User** – Designate a user in the value column.
Context Hint

When developing, assign the personalization to yourself to avoid affecting other users (even though you should be in a test instance).
Actions Tab
Actions Tab Elements

- **Seq** – Multiple actions can be assigned. This indicates the sequence of these actions.
- **Type** – Action types are
  - Property – assigns a property to a target object.
  - Message – Displays a message
  - Builtin – calls custom code or function (not covered today)
  - Menu – change or add a menu selection (not covered today)
- **Description** – Optional
- **Language**
- **Enabled**
The right hand side of the Actions tab changes depending upon the Action Type.

Today, we will discuss two action types:

- Property
- Message
Action Type - Message

If the Action Type is Message, select a message type:

- **Show** – display only
- **Hint** – Not covered today.
- **Error** – Stops commit (save) and gives a reason (maybe).
- **Debug** – Not covered today
- **Warn** – Allows the user to choose to proceed or not.
Message Text

Text can be simple, or it can have elements of SQL.

This example uses information from the form in the text displayed. The Item Number is imbedded in the message.
Action Type - Property

There are 50 available properties.

Examples:
• Value – Contents of the field.
• Required – True/False (required or not required)
• Enterable – True/False (field can be entered or cannot be entered)
Validate

Use the Validate button to test your Personalization

Click “Validate” and if your Personalization works, the message pops up.

- Item 16650197-01 is pending effectivity
Examples

Purchase Order form

1. Field Required
   - Item Revision

1. Insert Default Value
   - Revision Default

1. Display a message
   - Pending Effectivity

Receiving Transactions form

4. Hide Button
   - Inspection Button

5. Populate Field
   - Lot Number

5. Display Warning
   - Supplier Lot missing
Ex. 1 – Field Required

Navigate to the form to be personalized. In this case, the Purchase Order.
Ex. 1 – Field Required

Navigate to Forms Personalization

- Help
- Diagnostics
- Custom Code
- Personalize
Ex. 1 – Field Required

Our example is *PO Revision Required*

Why: For every item ordered, the business requires a revision number. On this form, the revision number is not required.
Ex. 1 – Field Required

Form Personalizations (Purchase Orders)

Seq | Description | Level | Enabled
--- | --- | --- | ---
1 | Revision Required | Function | ✓
2 | Revision Default to most recent | Form | ✓
3 | Item Pending Effectivity | Form | ✓
4 | ASL Lookup | Form | 

Condition

Trigger Event: WHEN-VALIDATE-RECORD
Trigger Object: PO_LINES
Condition: Not in Enter-Query Mode

Level | Value
--- | ---
Site | 

Insert 'Get' Expression... Insert Item Value... Validate Apply Now
Ex. 1 – Field Required

- Complete the Header section.
  - Seq = 1
  - Description = Revision Required
  - Level = Function
  - Enabled is checked
Ex. 1 – Field Required

Complete the Condition Tab.

Trigger Event = When Validate Record
Trigger Object = PO_LINES
Condition is null
Processing Mode = Not in Enter Query Mode
Context = Site (no value needed)
### Ex. 1 – Field Required

**Complete Actions Tab**

- **Seq = 1**
- **Type = Property**
- **Object Type = Item**
- **Target Object = PO_LINES.ITEM_REVISION**
- **Property Name = Required**
- **Property Value = True**

---

<table>
<thead>
<tr>
<th>Seq</th>
<th>Type</th>
<th>Description</th>
<th>Language</th>
<th>Enabled</th>
<th>Object Type</th>
<th>Target Object</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Property</td>
<td></td>
<td>All</td>
<td></td>
<td>item</td>
<td>PO_LINES.ITEM_REVISION</td>
<td>REQUIRED</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
Ex. 1 – Field Required

• Save your work!
• When this form is used in the future, the field will be required.
Ex. 2 – Insert Default Value

Revision Default to Highest Value

Why: The business wants to minimize mistakes on purchase orders. Unless there is a special need, the default revision should be the most recent.
Ex. 2 – Insert Default Value

We are already in the form from example 1.
• Complete the Header section.
  – Seq = 2
  – Description = Revision Default is most recent
  – Level = Form
  – Enabled is checked
Ex. 2 – Insert Default Value

Complete the Conditions Tab

- Trigger Event = When New Item Instance
- Trigger Object = PO_LINES.ITEM_REVISION
- Context Level = Site
- No Context Value is needed.

Condition =

:PO_LINES.ITEM_REVISION IS NULL AND
:PO_LINES.ITEM_NUMBER IS NOT NULL
Ex. 2 – Insert Default Value

Condition explained…

:PO_LINES.ITEM_REVISION IS NULL AND
:PO_LINES.ITEM_NUMBER IS NOT NULL

This means…

The item revision is blank, but the item number is not blank. If this is true, the personalization will be invoked.
Ex. 2 – Insert Default Value

Complete the Actions Tab

- SEQ = 1
- Type = Property
- Object Type = Item
- Object Target = PO_LINES.ITEM_REVISION
- Property Name = Value
Ex. 2 – Insert Default Value

Actions Tab Continued

Complete the Value section...
• A SQL statement
Ex. 2 – Insert Default Value

This SQL statement selects the highest valued revision, according to the method used by Oracle.

A SQL statement for this purpose can only return a single value, otherwise an error results. This makes since, because only one value can be placed into the field.

The item number as delivered by the form is used at the bottom. You can see the :PO_LINES, which is the block.

```
=SELECT A
  from ( 
    select 
      I.SEGMENT1
    , max(R.revision) A
  FROM
    INV.MTL_ITEM_REVISIONS_B R
  ,APPS.MTL_SYSTEM_ITEMS_FVL I
  WHERE
    R.inventory_item_id (+) = I.inventory_item_id
    AND R.organization_id (+) = I.organization_id
    AND I.ORGANIZATION_ID = 167
    AND SEGMENT1 = :PO_LINES.ITEM_NUMBER
  group by segment1
  )
```
Ex. 3 – Display a Message

Display a Pending Effectivity Message

Why? The business wants purchasing agents to be aware of an expected change to an item when placing an order. They may decide to order less of the item, or defer the purchase altogether.
Ex. 3 – Display a Message

We are already in the form from example 1.
• Complete the Header section.
  – Seq = 3
  – Description = Item Pending Effectivity
  – Level = Form
  – Enabled is checked
Complete the Conditions Tab

- Trigger Event = When New Item Instance
- Trigger Object = PO_LINES.ITEM_NUMBER
- Context Level = Site
- No Context Value is needed.

Condition = a SQL statement (next slide)
Ex. 3 – Display a Message

This Condition is a SQL Statement.

This is looking for a flag in a descriptive flex field (DFF) for the item.

If the item number is among those that have this attribute, the personalization will be invoked.

:PO_LINES.ITEM_NUMBER in
(SELECT
  SEGMENT1
FROM
  APPS.MTL_SYSTEM_ITEMS_B
WHERE
  ATTRIBUTE17 = 'Y')
Ex. 3 – Display a Message

Complete the Actions Tab.
- Seq = 1
- Type = Message
- Message Type = Show
- Message Text = a partial SQL statement
Ex. 3 – Display a Message

Message Text

`='Item '||:PO_LINES.ITEM_NUMBER||' is pending effectivity'`

This message text includes elements of SQL, but is very simple.

```
"=" tells the system this is going to be a SQL statement.

`Item` is plain text.

|| concatenates text with :PO_LINES.ITEM_NUMBER

(::<Block>..<Item>)

|| attaches the "is pending effectivity"
```
Ex. 4 – Hide Button

Hide Inspection Button on The Receiving Transactions Form

Why? Warehouse receivers are not permitted to also inspect. That duty is restricted to QA. However, access to the form is needed for other reasons. Thus, the business wants to hide the button only for a particular responsibility.

Note: No SQL is involved in this personalization.
Ex. 4 – Hide Button

Navigate to the Receiving Transactions Form

The Inspect Button appears at the bottom right of the form.
Ex. 4 – Hide Button

Navigate to Forms Personalization
- Help
- Diagnostics
- Custom Code
- Personalize
Ex. 4 – Hide Button

- Complete the Header section.
  - Seq = 1
  - Description = Inspection Button
  - Level = Form
  - Enabled is checked
Ex. 4 – Hide Button

Complete the Condition Tab
• Trigger Event = When New Form Instance
• Context Level = Responsibility
• Context Value = Warehouse Receiving
Ex. 4 – Hide Button

Complete the Actions Tab

- Seq = 1
- Type = Property
- Object Type = Item
- Target Object = RCV_CONTROL.INspeCTION_BUT
- Property Name = Displayed
- Value = False (not displayed)
Ex. 4 – Hide Button

Result - Button is gone
Ex. 5 – Populate Field

Supplier Lot Number is to be populated with one of three values.

• Receipt Number
• Supplier Lot Number – if a DFF flag indicates this is desired
• If Supplier Lot Number is indicated by DFF, but not provided, populate with zero
Ex. 5 – Populate Field

We are already in the form from example 4.

• Complete the Header section.
  – Seq = 2
  – Description = Lot Number Determination
  – Level = Form
  – Enabled is checked
Ex. 5 – Populate Field

Complete the Condition Tab

- Trigger Event = When New Block Instance
- Trigger Object = LOT_ENTRY
- Context Level = Site
Ex. 5 – Populate Field

Complete the Actions Tab

- Seq = 1
- Type = Property
- Object Type = Item
- Target Object = LOT_ENTRY.LOT_NUMBER
- Property Name = Value
- Value = a SQL statement

```
= Select A from (select CASE WHEN ATTRIBUTE11 = 'Y' THEN nvl(:RCV_TRANSACTION.VENDOR_LOT_NUM,0) ELSE :SUBROUTINE_AREA.RECEIPT_NUM END) AS Result
```
Ex. 5 – Populate Field

The value is a SQL statement.
Reference is made to a descriptive flexfield. If its value is Y, then use the supplier lot number. Otherwise, use receipt number.
If DFF = ‘Y’ but there is no value in supplier lot number, use ‘0’.

```sql
= Select A from (select
    CASE WHEN ATTRIBUTE11 = 'Y' THEN
        nvl(:RCV_TRANSACTION.VENDOR_LOT_NU M,'0') ELSE
        :OVERFLOW_AREA.RECEIPT_NUM
    END A
from
    apps.mtl_system_items
where segment1 =
    :LOT_CONTEXT.ITEM
and organization_id = 168)
```
Ex. 6 – Display Warning

To fill a gap on Ex. 5…

What if the supplier lot number is not there when needed?

Using a zero for the lot number is not desired.

The system should provide a warning.
Ex. 6 – Display Warning

<table>
<thead>
<tr>
<th>Seq</th>
<th>Description</th>
<th>Level</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inspection Button</td>
<td>Form</td>
<td>✔</td>
</tr>
<tr>
<td>2</td>
<td>Lot Number Determination</td>
<td>Form</td>
<td>✔</td>
</tr>
<tr>
<td>3</td>
<td>Warning Supplier Lot Missing</td>
<td>Form</td>
<td>✔</td>
</tr>
</tbody>
</table>

- Complete the Header section.
  - Seq = 3
  - Description = Warning Supplier Lot Missing
  - Level = Form
  - Enabled is checked
Ex. 6 – Display Warning

Complete the Condition Section

- Trigger Event = When Validate Record
- Trigger Object = LOT_ENTRY
- Condition = :LOT_ENTRY.LOT_NUMBER = '0'
- Context Level = Site
Ex. 6 – Display Warning

Complete the Actions Tab

- Seq = 1
- Type = Message

- Message Type = Warn
- Message Text = The lot number selected appears to be incorrect. Click "Cancel" to go back and reenter your data.
Ex. 6 – Display Warning

• If a lot number of “0” is encountered, the system displays this message.

![Image of display warning message]

The lot number selected appears to be incorrect. Click "Cancel" to go back and reenter your data.
Documentation

• How you can document Forms Personalizations
  – Manual
  – Query

• How to verify your documentation
  – Oracle’s Find Personalizations Form
Documentation

• Forms Personalizations typically do not have a large volume of data. Cut and paste into a document can be useful
• A query can be useful too.
A Query

```sql
SELECT fcr.id,
       fff.user_function_name,
       FCR.FORM_NAME FORM,
       FCR.SEQUENCE SEQ,
       FCR.DESCRIPTION,
       case when fcr.rule_type = 'F' then 'FORM'
            when fcr.rule_type = 'A' then 'FUNCTION'
            ELSE 'UNKNOWN' END P_LEVEL,
       FCR.ENABLED,
       FU.USER_NAME,
       FCR.TRIGGER_EVENT,
       FCR.TRIGGER_OBJECT,
       FCR.LAST_UPDATE_DATE
FROM APPLSYS.FND_FORM_CUSTOM_RULES FCR,
     APPS.FND_USER FU,
     apps.FND_FORM_FUNCTIONS_VL FFF
WHERE fcr.function_name = fff.function_name
   AND fcr.last_updated_by = FU.USER_ID
   AND FCR.ENABLED LIKE :ENABLED
ORDER BY form_name, sequence;
```
Documentation

While in a Forms Personalization

• Tools
  – Administration
Documentation

The system lists all the forms with Forms Personalizations, with a count of each.

<table>
<thead>
<tr>
<th>Form</th>
<th>Function</th>
<th>User Function Name</th>
<th>Enabled Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVIDITM</td>
<td>INV_INVIDITM</td>
<td>Master Items</td>
<td>11</td>
</tr>
<tr>
<td>INVTTMTX</td>
<td>INV_INVTMTX_SUBIN</td>
<td>Subinventory Transfer</td>
<td>2</td>
</tr>
<tr>
<td>OEXOEBSO</td>
<td>ONT_OEXOEBSO</td>
<td>Sales Agreements</td>
<td>1</td>
</tr>
<tr>
<td>OEXOEBSO</td>
<td>ONT_OEXOEBSO_SUB</td>
<td>Sales Agreement Organizer</td>
<td>1</td>
</tr>
<tr>
<td>OEXOEORD</td>
<td>ONT_OEXOEORD_SUB</td>
<td>Order Organizer</td>
<td>1</td>
</tr>
<tr>
<td>POXPOEPO</td>
<td>PO_POXPOEPO</td>
<td>Purchase Orders</td>
<td>6</td>
</tr>
<tr>
<td>POXRQERQ</td>
<td>PO_POXRQERQ</td>
<td>Requisitions</td>
<td>2</td>
</tr>
<tr>
<td>POXSCASL</td>
<td>PO_POXSCASL</td>
<td>Define Approved Supplier List</td>
<td>2</td>
</tr>
<tr>
<td>POXSCERQ</td>
<td>PO_POXSCERQ</td>
<td>RFQ’s</td>
<td>1</td>
</tr>
<tr>
<td>RCVTXERT</td>
<td>RCV_RCVTXERT</td>
<td>Receiving Transactions</td>
<td>3</td>
</tr>
</tbody>
</table>
Migration to Production

There are two methods to migrate into production.

• Enter directly as shown in these slides.
• Use scripts which are available to copy from a test instance into production.
Cautions

• Future Oracle patches could impact your Forms Personalizations. Testing is needed when you patch.

• Don’t try to remove standard Oracle restrictions through Forms Personalization. If you want to skip a field that is required, try using an “auto-populate” method instead.

• It would be a good idea to restrict access to the “Utilities:Diagnostics” profile.
Q&A

- tjones@colibrilimited.com