Developing Oracle EBS Custom Extensions: The Right Way

Doug Manning
Keyur Pancholi

April 15th, 2008
Agenda

- Background
- Types of Oracle EBS Customizations
- Customization Considerations
- JHU-APL Custom Extensions
- Wrap-up
- Questions
Background

Johns Hopkins University Applied Physics Laboratory (JHUAPL or just APL)

• One of world’s premier university research labs (est. 1942)
• Home to ~4000 scientists, engineers and other staff
• Comprised of 11 separate business areas across 11 departments and actively involved with 400+ programs sponsored by US government & other international organizations
• Oracle Applications EBS customer since 1994 (Purchasing, AP, Fixed Assets, GL, Inventory, AR & Cash Mgt.)
• Over the years we have designed, developed and deployed many custom extensions to our Oracle EBS product
• Currently in process of migrating to Oracle EBS R12
General Reasons Why Oracle EBS Customers Customize

1. **Data** - Standard Oracle EBS product doesn’t capture the specific data that are desired and required.

2. **Functionality** – Standard Oracle EBS product does not possess the desired processes and/or features that are desired and required.
Types of Oracle EBS Customizations

1. **Configuration** - The use of delivered functionality to capture and process custom organization data [e.g., Profile Options, Descriptive Flex Fields (DFF’s) and EBS Security Setups (Function and Data)].

2. **Personalization** – The use of delivered functionality to modify the standard Oracle EBS visual, look & feel and data display characteristics [e.g., Forms Foldering and UI tailoring (color, display), etc.].

3. **Extensibility** – Developing and deploying custom or 3rd Party code to augment the delivered Oracle EBS product (e.g., Custom Forms, Reports, CUSTOM.PLL, 3rd Party Software integration, and other methods).
Oracle EBS Custom Extension Considerations – One (Don’t)

*If at all possible, do not develop custom extensions to your Oracle EBS product. This should be your last solution option.*

**Why?**

1. Increased costs of EBS ownership.
2. Increased risk of EBS ownership.
3. Current Oracle EBS contains enhanced data capture and processing functionality for non-standard custom data.
Oracle EBS Custom Extension Considerations – One (Don’t)

What are some specific Costs and Risks associated with developing custom extensions to your Oracle EBS/

1. Increased cost of ownership due to hiring, training and maintaining a competent IT staff who will have to develop, implement and maintain these customizations.
2. Increased cost of ownership since customizations need to be evaluated, tested, and possibly re-implemented in association with patch and upgrade activities.
3. Increased risk of ownership due to in-house developed customizations that break existing standard EBS product transaction and/or interface processes.
4. Increased risk of ownership due to Oracle patches that (when applied) break in-house custom developed transaction and/or interface processes.
5. Increased risk to your organization of Oracle not supporting your Oracle EBS investment due to your implementation of non-supported customizations.
Instead of developing custom extensions, change your business process to fit the standard Oracle EBS process.

Why?

1. Decreased EBS ownership cost and risk.
2. Your Oracle EBS processes will be more in line with other organizations and industries.
Oracle EBS Custom Extension Considerations – Two (Change BP)

What are some specific realities that may not allow you to change your current business processes to fit the standard Oracle EBS?

1. No support from your upper management.
2. No support from your organization Subject Matter Experts (SME’s) and/or functional super users.
3. The standard Oracle EBS process does not contain the vital and required data or functionality components that are critical to your organization.
4. The standard Oracle EBS process functionality is too burdensome to use or is just not usable.
Instead of developing custom extensions, customize/tailor your Oracle EBS product using Oracle’s built-in configuration features and functionality.

Why?

1. Decreased EBS Ownership Cost and Risk.
2. Easy.
Oracle EBS Custom Extension Considerations – Three (Configure)

What are some specific ways of using the configuration method to customize your Oracle EBS/

1. DFF’s.
2. Profile Options.
Oracle EBS Custom Extension Considerations – Four (Personalize)

Instead of developing custom extensions, personalize your Oracle EBS product’s visual look and feel and data display functionality to fit your organizations needs and desires.

Why?
1. Decreased EBS Ownership Cost and Risk.
2. Enhanced data access.
3. Enhanced data accuracy.
Oracle EBS Custom Extension Considerations – Four (Personalize)

What are some specific ways of using the personalization method to customize your Oracle EBS?

1. Develop custom Folder Forms.
2. Tailoring/Individualizing forms queries.
3. Tailoring/Individualizing the UI (Color, Form textures, etc.).
Oracle EBS Custom Extension Considerations – Five (Custom Extend)

*If your critical organization data and/or process is missing from the standard Oracle EBS product and:*

1. You cannot change your business process
2. You cannot configure the Oracle EBS
3. You cannot personalize the Oracle EBS
4. You are willing and able to increase Oracle EBS cost and risk
5. You have buy-in and approval from upper management and key personnel (SME’s and SU’s)

*Developing and deploying a custom extension to the Oracle EBS is the option to pursue.*
Types of Oracle EBS Custom Extensions

1. **Overriding Standard Functionality** - The use of custom code to override the standard delivered functionality of the Oracle EBS product (e.g., CUSTOM.PLL, custom developed object code, etc.).

2. **Custom Object & Process Augmentation** – The development and deployment of custom software objects and processes to enhance and expand upon standard delivered Oracle EBS functionality [e.g., Custom developed Oracle Forms, (Reports, Oracle Discoverer, Oracle OBIEE objects), Workflow, Oracle JDeveloper objects, etc.].

3. **Integration of 3rd Party Software Product Solutions** – Utilizing and deploying 3rd-Party software to enhance and expand upon delivered Oracle EBS product functionality.
Types of Oracle EBS Custom Extensions: Override Std. Functionality

*Using the built-in Oracle delivered CUSTOM.pll structure is the supported and best method for creating custom Oracle EBS forms functionality Overrides:*

1. It is a component of your Oracle EBS forms library.
2. Consists of user exit and execution code associated with Oracle EBS forms.
3. Custom code that you incorporate into this library will not be overwritten or affected by subsequent/current Oracle version bug fixes and patches.
4. *Will still be a part of Oracle’s R12 product, you just need to re-apply the custom extensions.*
Types of Oracle EBS Custom Extensions: Override Using CUSTOM.pll

Enter Requisitions Form – Supplier Field Entry Override
Types of Oracle EBS Custom Extensions: Override Using CUSTOM.pll

Enter Requisitions Form – Supplier Field Entry Override Code

```sql
IF In_Event = 'WHEN-NEW-FORM-INSTANCE'
THEN
  --
  IF in_form = 'POXROERQ'
  THEN
    --
    set_item_property ('LINES.SUGGESTED_VENDOR_NAME', validate_from_list, property_true);
    set_item_property ('LINES.SUGGESTED_VENDOR_LOCATION', validate_from_list, property_true);
    --
```
Types of Oracle EBS Custom Extensions: Override Using CUSTOM.plII

Enter Requisitions Form – Tools Menu Item Dropdown List & Code

```
app_special2.instantiate ('SPECIAL15', '6Print Requisition', 'POCROERQ', TRUE, 'LINE ');
app_special2.instantiate ('SPECIAL4', '6Requisition Routing', 'APLDisl', TRUE, 'LINE ');
app_special2.enable ('SPECIAL4', property_on);
```
Types of Oracle EBS Custom Extensions: Custom Object/Process Augmentation

*Using Oracle tools and associated Oracle coding standards is the supported and best method for creating custom forms and processes to enhance your standard Oracle EBS product:*

1. Oracle Forms and OAF.
2. Oracle JDeveloper.
3. Oracle Workflow.
4. Oracle (Reports, BI and OBIEE, Hyperion).

Copyright ©2008 The Johns Hopkins University - APL
Types of Oracle EBS Custom Extensions: Custom Objects/Processes

Automated Requisition Routing – Oracle Forms
Types of Oracle EBS Custom Extensions: Custom Objects/Processes

Automated Requisition Routing – Oracle Forms & MOD PL/SQL

Copyright ©2008 The Johns Hopkins University - APL
Types of Oracle EBS Custom Extensions: Custom Objects/Processes

Automated Requisition Routing – MOD PL/SQL
Types of Oracle EBS Custom Extensions: Custom Objects/Processes

Oracle JDeveloper – Supplier Rating
Types of Oracle EBS Custom Extensions: Custom Objects/Processes

Oracle JDeveloper – Supplier Rating
Types of Oracle EBS Custom Extensions: Custom Objects/Processes

Oracle Workflow
Types of Oracle EBS Custom Extensions: 3rd Party SW Integration

Using Oracle certified 3rd Party software is an option that should be considered if the Oracle EBS is missing critical organization data and/or processes:

1. Data & processes that are missing are Organization critical.
2. Oracle Certified 3rd Party Software.
Types of Oracle EBS Custom Extensions: 3rd Party SW Integration

3rd Party Software – Parcel Tracking
Types of Oracle EBS Custom Extensions: 3rd Party SW Integration

3rd Party Software – Oracle EBS Integration

```
CREATE OR REPLACE PROCEDURE
INTRASYSDM.NJ_SP_IMPORTRECIPS_JH_PEOPLE(ERREBUF OUT VARCHAR2,
RETCODE OUT VARCHAR2) AS
ST00_SELCNT INTEGER;
ST00_ERROR INTEGER;
ST00_WARNCNT INTEGER;
ST00_ERRMMSG VARCHAR2(255);
...
-- 1) Pull recipient information from recipients (View) into a temp
table (RecipImportTable)
-- 2) Inserts any records from RecipImportTable into the Recipient
table that does not already exist
-- 3) Updates any records in the Recipient table that have changed
-- 4) Tags records as deleted (Status='Deleted') that are in the
Recipient Table but not in the RecipImportTable

BEGIN
APPS_FND.FND_FILE.PUT_LINE(APPS_FND.FND_FILE.OUTPUT,
'<<< Process Started: ' ||
TO_CHAR(SYSDATE, 'DD-MON-YYYY HH:MI:SS'));
-- Delete records from temp import table.
...
...
COMMIT;
...
APPS_FND.FND_FILE.PUT_LINE(APPS_FND.FND_FILE.OUTPUT,
'<<< Process Ended: ' ||
TO_CHAR(SYSDATE, 'DD-MON-YYYY HH:MI:SS'));

EXCEPTION
WHEN OTHERS THEN
ST00_WARNCNT := 0;
ST00_SELCNT := 0;
ST00_ERROR := SQLCODE;
ST00_ERRMMSG := SQLERRM;
RAISE_APPLICATION_ERROR(SQLCODE, SQLERRM, TRUE);
ROLLBACK;
END:
```
Types of Oracle EBS Custom Extensions: 3\textsuperscript{rd} Party SW Integration

3\textsuperscript{rd} Party Software – Oracle EBS Integration

Copyright ©2008 The Johns Hopkins University - APL
Oracle EBS Custom Extensions that Should Never be Performed

A discussion of Oracle EBS custom extensions would not be complete without mentioning customizations that should Never be performed:

1. Directly changing the functionality of object elements in an Oracle EBS delivered form (e.g., add, modify, delete: fields, buttons, regions, tabs, menu items, etc.).
2. Directly changing the functionality of object elements in an Oracle EBS delivered report (e.g., add, modify, delete: data elements, columns, titles, headers, footers, etc.).
3. Directly changing other delivered Oracle EBS software objects (e.g., triggers, functions, procedures, html and java objects, etc.).

Why?

Extreme risk and high probability of data integrity & process errors.
Extreme risk and high probability of not being able to apply Oracle bug fixes and CPU’s.
Inability to replicate custom functionality in future product releases.
Possibility of non-support from Oracle Corp.
R12 Custom Extension Considerations

- Expanded Configuration & Personalization
- CUSTOM.pll
- MOD PL/SQL (Available or Not?)
Wrap-up

Developing Oracle EBS custom extensions the right way not only involves the skill and know-how of performing the custom extension, but more importantly involves a methodology of progressive and comprehensive investigation to choose the correct solution for extending Oracle product functionality.

A custom extension checklist would include:

1. Before customizing your Oracle EBS, see if you can change your current business process to fit the Oracle EBS process.

2. Use Oracle EBS configuration and/or personalization functionality to capture and process unique data and procedures before writing custom extensions.

3. Custom extend the Oracle EBS (The Right Way) by using supported Oracle extensibility methods such as the CUSTOM.PLL, Oracle Tools (Forms, OAF, JDeveloper, Workflow, etc.) and 3rd Party software that is Oracle certified.
Questions?