Customizing Oracle Applications with CUSTOM Library (CUSTOM.pll)

Biju Varghese
University of Virginia, Charlottesville, Virginia

Introduction:

The University of Virginia was founded by Thomas Jefferson in 1819. The University is made up of ten schools in Charlottesville, plus the College at Wise in southwest Virginia. UVa offers 51 bachelor's degrees in 47 fields, 83 master's degrees in 66 fields, six educational specialist degrees, two first-professional degrees (law and medicine), and 57 doctoral degrees in 55 fields. The University has more than 20,000 students and almost 15,000 employees.

University of Virginia implemented Oracle ERP software in the year 2001 with Oracle Apps 11.0.3 version. We upgraded to 11.5.9 in 2004 and to 11.5.10 in 2006. We have most of the financial modules, HR, and Payroll under Oracle Apps. We have used CUSTOM.pll extensively to customize our Oracle forms to match the business requirement of the University. We have more than 25 customizations done through CUSTOM.pll in Human Resource and other financial modules.

This presentation will cover many topics of CUSTOM.pll customizations, including technical details of each form event and CUSTOM.pll coding standards and design considerations. Attendees will learn CUSTOM.pll programming and techniques and will have an opportunity to understand real life customizations done at the University of Virginia Integrated System. Trouble shooting tips will also be included.

Pre-Requisites

The custom library is used only for Oracle applications. So, you should have sound knowledge in Oracle application technology. CUSTOM.pll programming is a combination of Forms programming and PL/SQL.In order to work on complex customizations using CUSTOM.pll, deep knowledge of Oracle forms and event driven programming are required.

Overview

Custom.pll is an attached library of an Oracle Form. Oracle Application forms are located at $AU_TOP/forms/US directory in the UNIX system. If you take an Oracle form from this directory and open it in the Forms Builder, you can see many attached libraries like APPCORE, FNDSQF, GLOBE, CUSTOM etc. CUSTOM is nothing but CUSTOM.pll, which is one of those libraries attached to any Oracle Application form. All the libraries are located under $AU_TOP/resource directory in UNIX. The file extension '.pll' is the source code and '.plx' is the executable file. See the following;

```
$cd  $AU_TOP/resource
$ls -ltr CUSTOM.*

rwxrwxrwx   1 isd20    dba       32768 Feb 20 11:14 CUSTOM.pll
rwxrwxrwx   1 isd20    dba       28672 Feb 20 11:16 CUSTOM.plx

$cd  $AU_TOP/forms/US
$ls -ltr APXINWKB.fmb

-rwxrwxrwx   1 isd20    dba      10231808 Feb 25 16:57 APXINWKB.fmb
```

Custom library works the same way as Oracle forms, but Forms has higher priority than CUSTOM.pll code when it is running. The real beauty of CUSTOM.pll is you can customize Oracle forms without modifying Oracle seeded forms and these customizations are mostly protected from Oracle patches. So, you don't need to re-customize your Oracle forms, when you are applying patches. This is one of the major advantages of using CUSTOM.pll in your Oracle ERP implementation project.

Common Customizations

Some of the common customizations are disabling fields and blocks, hiding descriptive flex fields and confidential data like SSN, defaulting values to a field, enabling the Zoom icon and controlling the navigation of forms from one to another etc. You can also add business validations and custom error messages on certain forms and responsibilities. If you want to do some complex customizations, you need to understand the technical details of the underlying form.

Events and Triggers

Oracle Forms has hundreds of triggers for various purposes. But, you cannot use all the forms triggers in your custom library. Oracle doesn't support all the forms triggers in custom library for technical reasons.
You can use the following triggers in custom library as per Oracle guidelines;

- WHEN-NEW-FORM-INSTANCE
- WHEN-NEW-BLOCK-INSTANCE
- WHEN-NEW-RECORD-INSTANCE
- WHEN-NEW-ITEM-INSTANCE
- WHEN-VALIDATE-RECORD
- ZOOM

WHEN-NEW-FORM-INSTANCE trigger is executed when you enter into an Oracle form. Then it goes to block level, record level and finally to an item. WHEN-NEW-ITEM-INSTANCE is executed for each item in the block. The WHEN-VALIDATE-RECORD fires when you leave from one record to another. The Zoom event is used to navigate from one form to another without exiting from it. It helps users to save time because of less keystrokes and mouse clicks. Depending on your business requirement, you need to find out the suitable trigger logic. You should execute your logic only for the required form and block. You have to use IF conditions to verify the current form and block before you insert your logic in custom library. The custom library template, which comes with the installation, has some sample customizations (commented out by Oracle) which will be very useful when you are coding. See the following sample code;

```
procedure event(event_name varchar2) is

   form_name      varchar2(30) := name_in('system.current_form');
   block_name     varchar2(30) := name_in('system.cursor_block');

  begin
   if (event_name = 'ZOOM') then
     if (form_name = 'DEMXXEOR' and block_name = 'ORDERS') then
        ………
   elsif (event_name = 'WHEN-VALIDATE-RECORD') then
     if (form_name = 'APXVENDR') then

         if (block_name = 'VENDOR') then
             copy(upper(name_in('VENDOR.NAME')), 'VENDOR.NAME');

          end if;

     end if;
   else
         null;
   end if;

  end event;
```

If you want to see the custom events which are executed from your form, you can go to the Oracle Applications Menu: Help->Diagnostics->Custom Code->Show Custom Events. This feature is really useful when you are debugging CUSTOM.pll code. This feature will also help you to understand the flow of trigger logic in your form.

Coding Standards and Restrictions

There is a chapter on custom library in the Oracle Application Developers Guide. There are certain standards you should follow when you are programming in CUSTOM.pll. You cannot use SQL statements directly, but you can call a PL/SQL function. Also, use the copy command instead of an assignment operator.

Example: copy('N','CTRL.SINGLE_ONLY');

You cannot use forms built-ins to create blocks and items, but you can modify the property of blocks and items using forms built-in functions. Also, use app_item_property( ) instead of set_item_property( ).

Troubleshooting Ideas

There are different ways you can troubleshoot a CUSTOM.pll bug or a problem. You can display error messages inside your code using a quick command called fnd_message.debug( ). If your forms not working properly, you should disable the custom code first by going to the menu: Help->Diagnostics->Custom Code-> Off. After disabling the custom code, if you are getting the same error again, then it is coming from the Oracle seeded form and you may have to open a Service request with Oracle. Otherwise, the error is because of custom library and you need to fix your program. You can also use Forms run-time Diagnostics (FRD) to find out the root cause of the issue. To generate a FRD trace file, you need to assign values to the profile 'ICX: Forms Launcher' at the user level. Once you set the values to the user profile and run through your test case, it will generate a trace file under the directory $FORMS60_TRACE_PATH,which can be useful to find out the exact location of the error.

Tips and Suggestions

If you are using hard-coded values (Constant values) in your program, it will be costly for maintenance and future enhancements. Instead of using constant values, you can create user profiles and store those values in the profile and read those with a PL/SQL function. This is more efficient and cost-effective method. Use the following SQL statement to read a profile value from a profile:

```
SELECT profile_option_value
  FROM fnd_profile_option_values
 WHERE profile_option_id  IN (
                              SELECT profile_option_id
                              FROM fnd_profile_options
                              WHERE profile_option_name = 'UVA_HR_APPL_ACCESS'
                              AND TRUNC (SYSDATE) >= start_date_active
                               AND TRUNC (SYSDATE) < NVL (end_date_active, c_eternity)
                              )
   AND level_id = 10003
   AND level_value = p_resp_id;
```

Sometimes, after you have applied a mega patch or a rollup patch, your CUSTOM.pll customizations may not work properly. This is happening because Oracle should have changed their Forms design or attributes. If Oracle re-designs a form completely, and if you have built logic based on the Oracle form logic, you have to test your customizations thoroughly after applying the patch in your development instance. In this situation, you may even need to revisit your logic and come up with new trigger or logic to tackle the new Oracle forms behavior. This has happened many times at UVa !

Oracle Forms has higher precedence over CUSTOM.pll. So, your custom logic can be overridden by Oracle Forms triggers at the same scope/level. For example, if you have business logic in WHEN-NEW-BLOCK-INSTANCE and suddenly, after applying a mega patch, it stops working! Why? It may be because Oracle added some extra forms logic in their WHEN-NEW-BLOCK-INSTANCE trigger for the same block. To tackle this situation, you should move your logic to an item level trigger, if possible, so that your business logic will get executed again for each item. You might have to tackle it differently, depending on the problem or situation. Also, there might be times when you have to open the underlying Oracle form and understand the technical details of it to effectively program in custom library.

Another idea of managing CUSTOM.pll is to divide the library into small libraries for each module for HR, GM, AP, PO etc. We have one CUSTOM.pll and a dozen other sub-libraries for each module (UVAHR, UVAGMS etc). If you are dividing this way, different developers can work on different libraries at the same time, which will help to get more productivity from developers.

Future of Oracle Forms & CUSTOM.pll

Oracle is gradually converting Forms to Self service screens using JAVA and framework objects. Oracle Fusion may be a 100% self service based application without any Oracle forms. If that happens in Fusion, Oracle Forms will become obsolete and naturally custom library will become history. Custom library will be there as long as Oracle forms exist. The self service Forms are created using Oracle Application framework technology. So, if you want to customize a self service screen, you have to do this with Page personalization or OA framework extensions using JDeveloper, depending on the requirement. Technical knowledge is not required for page personalization, but you should know JAVA and object oriented programming to customize OA framework Page using JDeveloper.

Conclusion

CUSTOM.pll is fun! CUSTOM.pll programming can become tricky and complex depending on the situation, but it is challenging and interesting. It is an excellent tool to customize Oracle Applications software to match user business requirement without much impact. Also, you can make your users very happy with the little tiny coding of CUSTOM.pll. All the best!

Thank you.