

# Enhance Your Oracle Applications Using the CUSTOM Library

Scott D. Spivey  
*BOSS Corporation*

## Introduction

This paper explains how the CUSTOM library fits into the Oracle Applications forms processing, provides an understanding of the features and limitations of using the CUSTOM library and provide examples of using the CUSTOM library to enhance the functioning of the Oracle Application forms.

## What is the CUSTOM Library

The CUSTOM.pll library is a standard Oracle Forms PL/SQL library that is supplied by Oracle with the Oracle Applications. This is Oracle's built-in feature that allows the customer to enhance the standard functionality of the Applications by implementing site-specific business rules. Every Oracle Forms -based eBusiness screen, and any custom form developed using the Oracle Application development standards, will access the CUSTOM library. This makes an ideal point of creating business rules that effect the entire organization. This is the only method of forms enhancement whose functionality is supported by Oracle World Wide Support. Although any enhancements coded by the customer are not directly supported by Oracle World Wide Support.

## How to Use the CUSTOM Library

Since the CUSTOM library is a typical Forms PL/SQL library, it can include standard PL/SQL logic, Oracle Forms built-in commands and may have other Forms PL/SQL libraries attached to it. The base CUSTOM library is located in the \$AU\_TOP/resource directory on your forms server. Explore the CUSTOM.pll using the Oracle Forms Designer module to examine the sample code that exists in library. Once any enhancements are made, the library must be generated into an executable library (CUSTOM.plx) that then can be used by the Oracle Applications Forms runtime module. Since the CUSTOM library is loaded once for a given session, a user must log out of the application and sign-on again before any changes will become apparent. It is suggested that you also place a copy of the CUSTOM library in your customization directory as a safety precaution.

The CUSTOM PL/SQL library does have its limitations. It can not contain SQL commands. It can not have certain Oracle Applications Forms global libraries attached to it (such as APPCORE). The CUSTOM library is designed to be used solely with the Oracle eBusiness Applications and does not support the Oracle Self-Service Web Applications (OSSWA). Since the CUSTOM library's standard location is in the \$AU\_TOP/resource, it can be overwritten during an upgrade of the Applications.

Once enhancements are implemented, the CUSTOM library is accessed by the Oracle Applications based upon 'events'. The 'events' define the occasions when the Applications will look into the CUSTOM library for additional code to execute.

Some events are generic:

- WHEN-NEW -FORM -INSTANCE – initially entering a form
- WHEN-NEW -BLOCK -INSTANCE – entering a zone (or block) within a form
- WHEN-NEW -ITEM -INSTANCE – moving into a new field within the form
- WHEN-NEW -RECORD -INSTANCE - creating a new record
- WHEN-FORM -NAVIGATE – navigating thru a form using the mouse
- WHEN-VALIDATE-RECORD – saving (committing) the information to the database
- EXPORT – triggered by using the Export feature

Some events are field specific

- ZOOM – Pre-11 feature for moving to another form and querying up specific records

Some events are form specific

SPECIALn - (where n is a number between 1 and 45) used to generate entries in the 'Special' menu of the tool bar and the code is triggered by selecting a menu choices from the 'Special' option on the toolbar  
KEY-Fn - (where n is a number between 1 and 8) triggered by pressing the corresponding function key

Some events are application specific:

Application Object Library

WHEN-LOGIN-CHANGED - when a user logs on as a different user

WHEN-RESPONSIBILITY-CHANGED - when a user changes responsibilities

WHEN-PASSWORD-CHANGED - when a user changes their password

These are most of the documented events. Discussions from previous OAUG conferences suggest that many more events are passed to the CUSTOM.pll. The method used to find these events would be to create a generic piece of code to display the event name as it passes thru the CUSTOM library.

The PL/SQL logic within the library can be organized by these events and then all effected forms or by form and then all effected events. The Oracle development standard is to group forms logic within a specific event. This is a good solution if many forms are affected by the same event. Another development standard is to organize the logic by form and then specific events. This is a good solution if limited forms are affected by several events.

## Examples Used in the CUSTOM Library.

The CUSTOM library can be used to enforce business rules, secure standard features of the Oracle Applications (eliminate or restrict their use), and enforce naming standards.

Business rules can be enforced by adding logic that will be executed when a user attempts to save a record. The business rules can be as simple as determining that a user has the authority to perform the actions or that certain data has been entered or as complex as limiting field access and display based upon a user's security level. All of these items can be coded using PL/SQL logic and Forms built-in features. Since the CUSTOM library cannot contain SQL commands, that type of logic must be passed off to stored procedures or functions within the Oracle database.

This section of code looks for the event 'WHEN-NEW -FORM -INSTANCE' when the user is in the PO Supplier form (APXVDMVD). It then executes a custom stored PL/SQL function (FMS\_USER\_HAS\_RESP) to determine if the user has the proper authority to update the supplier information. If the user is in the Site portion of the form, then the custom procedure FMS\_WHEN\_NEW\_ITEM\_SITE is executed to allow or disallow changes to the Supplier Site information.

```
IF event_name = 'WHEN-NEW -FORM -INSTANCE' THEN
  If (form_name = 'APXVDMVD') THEN
    v_user_id := fnd_global.user_id;
    v_site_maint_ok := FMS_USER_HAS_RESP(v_user_id, 'Purchasing Vendor Site Maint');
    If (block_name = 'SITE' ) THEN
      FMS_WHEN_NEW_ITEM_SITE(v_site_maint_ok, v_new_record);
    End if; -- BLOCK=SITE
  End If; -- FORM=APXVDMVD
END IF;
```

Standard Oracle Application features can be secured or eliminated using standard Oracle Forms built-in features. A form button can have its logic disabled and even be removed from display by altering the button's characteristics.

This section of code hides the unreserved fields on the Purchase Order form (POXPOEPO) when the user first accesses the form (WHEN-NEW -FORM -INSTANCE event).

```
IF event_name = 'WHEN-NEW -FORM -INSTANCE' THEN
-- SECURE STANDARD ORACLE FEATURE
-- Turn off the Reserve/Unreserve functionality
```

```

If form_name = 'POXPOEPO'
And block_name = 'PO_APPROVE' then
  SET_ITEM_PROPERTY('PO_APPROVE.UNRESERVE',DISPLAYED,PROPERTY_FALSE);
  SET_ITEM_PROPERTY('PO_APPROVE.UNRESERVE_DATE',DISPLAYED,PROPERTY_FALSE);
End If;
END IF;

```

Since the event logic can be limited specifically to a given form and even a specific zone within a form, it is possible to create validation logic that can enforce naming standards and data entry standards for a type of transaction. One of the samples in the CUSTOM library provides logic to force a supplier's name to be uppercase.

This code changes the properties of several columns on the Purchase Order form to force uppercase data entry when the purchase order type is 'CONTRACT'. Notice that logic is required to reset the column properties for other purchase order types.

```

IF event_name = 'WHEN-NEW -FORM -INSTANCE' THEN
  If form_name = 'POXPOEPO' THEN
    -- Code to set the item properties for contract documents to Uppercase and Required.
    If (block_name = 'PO_HEADERS'
    And name_in('PO_HEADERS.TYPE_LOOKUP_CODE') = 'CONTRACT') THEN
      SET_ITEM_PROPERTY('PO_HEADERS.COMMENTS',CASE_RESTRICTION,UPPERCASE);
      SET_ITEM_PROPERTY('PO_HEADERS.START_DATE',REQUIRED,PROPERTY_TRUE);
      SET_ITEM_PROPERTY('PO_HEADERS.END_DATE',REQUIRED,PROPERTY_TRUE);
      SET_ITEM_PROPERTY('PO_HEADERS.AMT_LIMIT_DSP',REQUIRED,PROPERTY_TRUE);
    Else
      -- Code to set the item properties for NON contract documents to Uppercase and NON Required.
      SET_ITEM_PROPERTY('PO_HEADERS.COMMENTS',CASE_RESTRICTION,UPPERCASE);
      SET_ITEM_PROPERTY('PO_HEADERS.START_DATE',REQUIRED,PROPERTY_FALSE);
      SET_ITEM_PROPERTY('PO_HEADERS.END_DATE',REQUIRED,PROPERTY_FALSE);
      SET_ITEM_PROPERTY('PO_HEADERS.AMT_LIMIT_DSP',REQUIRED,PROPERTY_FALSE);
      v_doc_type := name_in('PO_HEADERS.TYPE_LOOKUP_CODE');
    END IF;
  END IF; -- Form = 'POXPOEPO'
END IF; -- EVENT=NEW -FORM

```

During the conference presentation, these examples and others will be discussed in more detail and shown in the entire context of the CUSTOM.pll library.

## Suggestions and Tips

To safeguard against accidental loss of your personalized CUSTOM library, place a copy in your customizations directory forms subdirectory. If you wish to execute the CUSTOM library from within your customization directory, you will need to add your customization forms directory path to the beginning of the FORMS60\_PATH o/s variable. This method will find your personalized version first and execute it instead of the standard version located in the \$AU\_TOP/resource directory.

To discover any hidden events that may be passed into the CUSTOM library, you could add a section of code to display the event name. Caution: since everyone that accesses the Applications will use the CUSTOM library, limit this experiment to off-hours or to limited use instances.

## Conclusion

Although this paper is limited in its ability to accurately show real-life examples of how the CUSTOM library can be used, it has explained what the CUSTOM library is, examined the various uses for the CUSTOM library and how

to implement extensions to the Oracle Applications. Further information concerning the CUSTOM library can be found in the Oracle Applications Developers Guide.

## **Company Profile**

**BOSS Corporation** is a services company devoted exclusively to Oracle Applications and Technology. BOSS was founded in 1995 on the premise that there should be something **BETTER**. The founders of BOSS were simply not satisfied with the level of services seen in our market, and they formed a company with the purpose of providing **BETTER** services to Applications customers...hence, **Better Organization Service Solutions** was formed and became known as BOSS

The author, Scott Spivey, is a certified Oracle professional and applications technical consultant with sixteen years systems analysis, application development, database design and leadership experience in implementation and enhancement of business and technical solutions with Oracle products. Scott has performed these duties in a variety of industries, from a state government agency and defense contractors to large-scale manufacturers and multi-media entertainment companies. Scott earned a Bachelor of Science Degree in Computer Science from Mississippi State University. He has an Oracle Education Master certification and two OCP certifications. Scott has presented at OAUG conferences in Atlanta and San Diego, while participating on discussion panels at OAUG conferences in San Diego, Honolulu, Philadelphia and Atlanta. Scott has also presented at the local Oracle user's group in Nashville, TN.