Form Personalization is done through the Front End in Oracle Apps. Personalization is just changing the properties with out making any changes. When you change the look and feel of the system. Say hide new fields, change tips, label of the field etc

When using Form Personalization, it is recommended to take some precautions when implementing it.

Ensure to set system profile **'Hide Diagnostics menu entry'** to 'No' and **'Utilities:Diagnostics'** to 'Yes' at the user level.

**Hide Diagnostics Menu Entry**

This profile option determines whether users can access the Diagnostics menu entry from the Help menu. The default value is Yes, with the Diagnostics menu entry is hidden. If it is set to No, the Diagnostics menu entry is visible

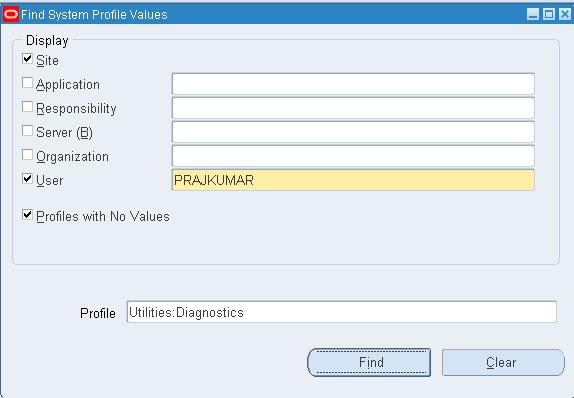
**Utilities: Diagnostics**

Utilities: Diagnostics determines whether a user can automatically use the Diagnostics features. If Utilities:Diagnostics is set to Yes, then users can automatically use these features. If Utilities:Diagnostics is set to No, then users must enter the password for the APPS schema to use the Diagnostics features.

1. Only trusted users should have access to this feature. Use the system profiles: **'Hide Diagnostics menu entry'** and **'Utilities:Diagnostics'** to control the visibility of the menu and the direct access to the feature.

Login as hp/hemant1

(N)System Administrator > Profile> System>



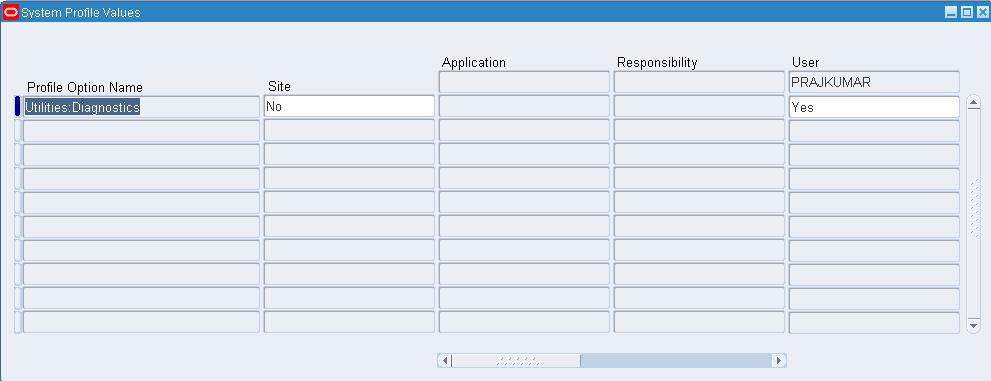
Check the “Check Boxes” for Site and User

Enter Application User (in our case hp) for whom you want to enable Diagnostics-> Examine

Enter profile name: Utilities:Diagnostics. Click the

(B)Find

Give Yes at User level and Save(B) the Changes



**Note –**

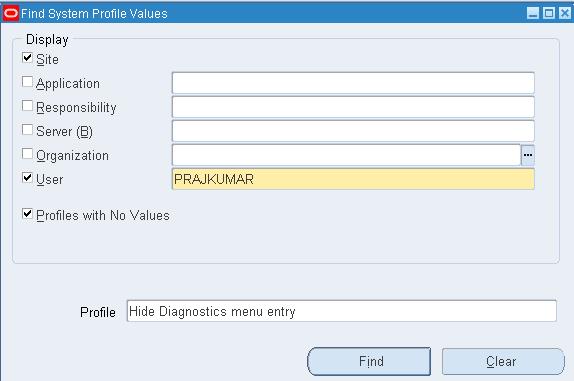
You can set Yes at Site level also if you want to enable this option for all Oracle application users

Again navigate to System Administrator responsibility> Profile> System>

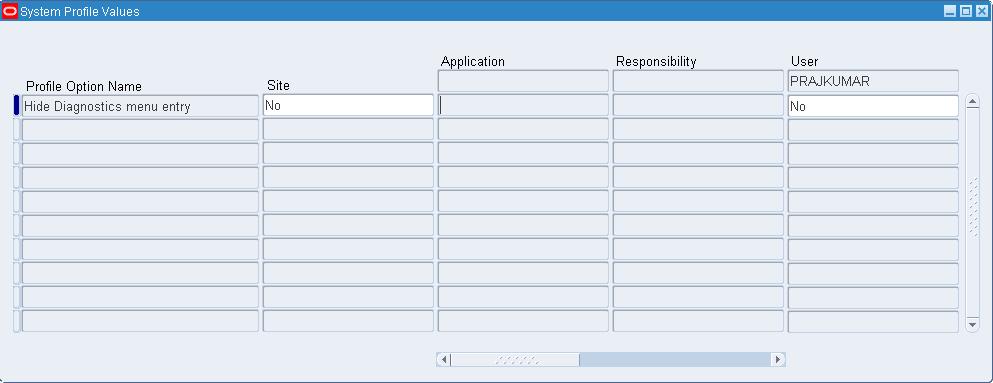
Enter Application User (in our case hp) for whom you do not want to hide Diagnostics menu entry

Enter profile name: Hide Diagnostics menu entry

(B)Find



Give No at User level and Save the Changes



**Note –**

You can set No at Site level also if you do not want to hide menu entry option for all Oracle application users

It is a good idea to test all Form Personalizations in a DEV or TEST environment before moving them to a Production environment. Remember that changes made could impact or interfere with the base code of a form;

**OVERVIEW AND BASIC CONCEPTS**

Form Personalization is a declarative feature that alter the behavior of [Oracle](http://it.toolbox.com/trd/95/7/9904/?lc=int_mb_1001) Forms-based screens, including changing properties, executing built-ins, displaying messages, and adding menu entries.

* **RULES**  
  For each function, you can specify one or more Rules. Each Rule consists of an Event, an optional Condition, the Scope for which it applies, and one or more Actions to perform.

An Event is a trigger point within a form, such as a startup (WHEN-NEW-FORM-INSTANCE), or a new record (WHEN-NEW-RECORD-INSTANCE). [Oracle forms](http://it.toolbox.com/trd/95/7/9870/?lc=int_mb_1001) sends standard and product-specific events.

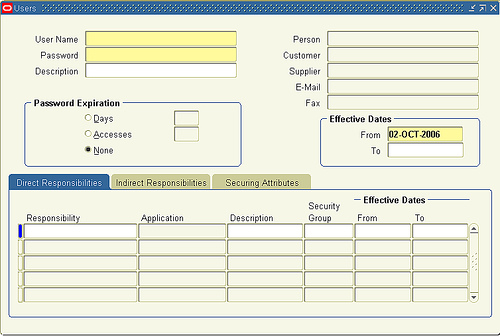
A Condition is an optional [SQL](http://it.toolbox.com/trd/95/7/9813/?lc=int_mb_1001) code fragment that is evaluated when an Event occurs. If the Condition evaluates to TRUE then the Actions are processed.  
  
A Scope is evaluated based on the current runtime context to determine if a Rule should be processed or not. The Scope can be Site, Responsibility or User. Each Rule can have one or more Scopes associated with it. When defining new Rules in your DEV or TEST environment, it is recommend setting the Scope to your User first just to be safe. If all is well with your personalization, then proceed in setting the Scope to it's intended level. If a Rule has a context of "Site", it will apply for everyone. A Rule with a context of Responsibility doesn't override Site. A Rule with a context of User doesn't override Site or Responsibility.

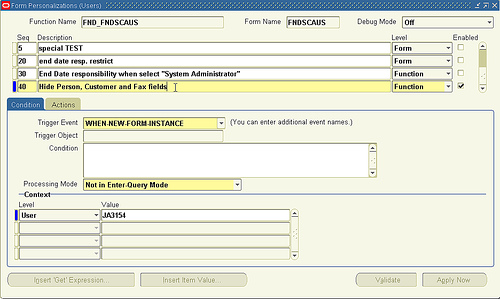
* **ACTIONS**  
  Each Action consists of **'Setting a property'**, such as making a required field or hiding a Tab page, **'Executing a Built-in'**, such as GO\_BLOCK, DO\_KEY or END\_FUNCTION.EXECUTE, **'Displaying a Message'** or **'Enabling a Special menu entry'**, such as a zoom.

Once Rules are defined, save and close the form and re-run the function (the form). Then the Rules are automatically applied as Events occur within that form.

**EXAMPLE OF A BASIC FORM PERSONALIZATION**

For this example, we will define a basic Forms Personalization by hiding the Person, Customer and Fax fields of the User form. Follow the following steps.  
  
Login as hp/password. Go to the responsibility System Administrator---Security---User---Define. This is the basic Users form as shown here:

[](http://www.flickr.com/photos/34041331@N00/258728479/)

From the Help menu(M) > Diagnostics > Custom Code > Personalize.  
If some Rules are already defined in the Users form, you will see them in this window.  
Add a new Rule by selecting (M) File > New or by clicking the (+) sign while being on the existing last rule or simply click on a new blank row  
[](http://www.flickr.com/photos/34041331@N00/258786454/)

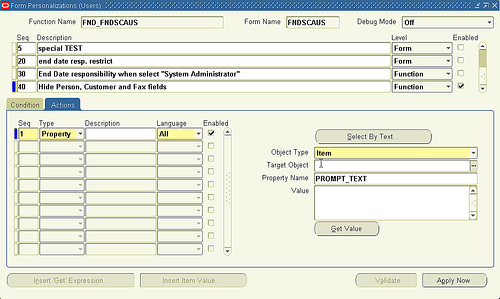
Rules Seq.: 30

Description: Hide Person, Customer and Fax fields

Trigger Event: WHEN-NEW-FORM-INSTANCE

Context or Scope: User level and Value : HP

Select the Actions tab. This is the Actions tab of Rule seq. 30

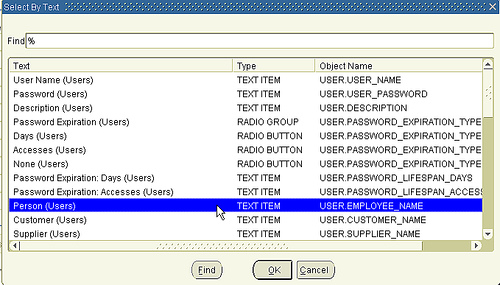
[](http://www.flickr.com/photos/34041331@N00/258804009/)

Action Seq.: 1

Type: Property

Object Type: Item

Select the **Select By Text** button

[](http://www.flickr.com/photos/34041331@N00/258811297/)

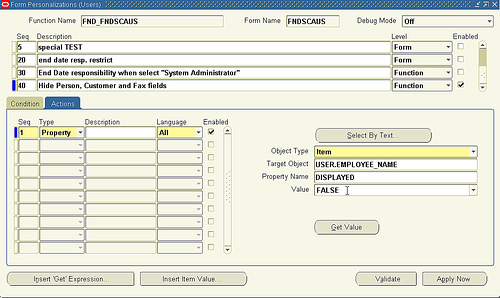
Select the **Person (Users)** item from the LOV

Property Name: DISPLAYED

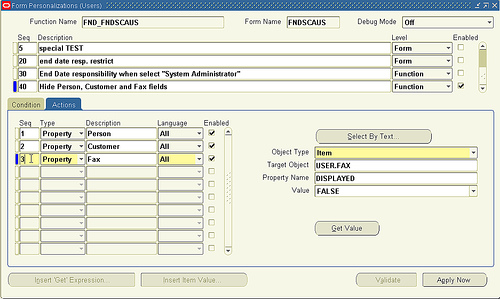
Value: False

Save it(CTRL+S), Click Apply Now and immediately in the background the concerned Text Item gets hidden

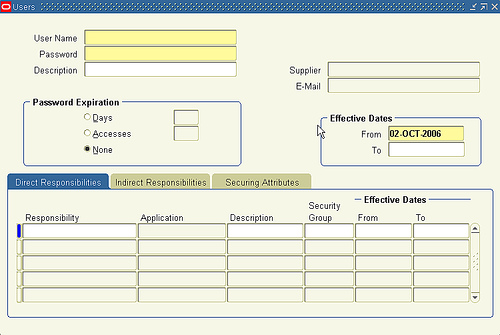
When you are done, it should look like this:

[](http://www.flickr.com/photos/34041331@N00/258825290/)

Repeat the Steps for the Customer and Fax items. Save all of your work. It should now look like this:

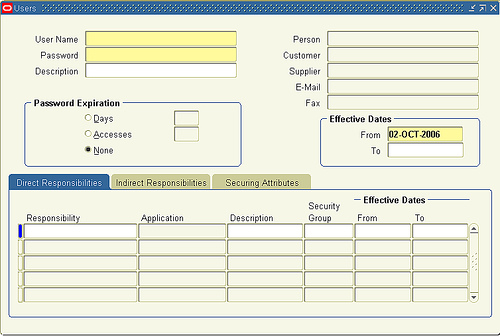
[](http://www.flickr.com/photos/34041331@N00/258834152/)

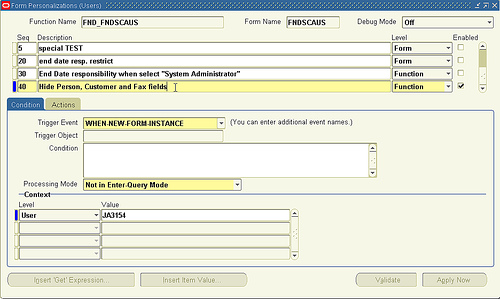
Close the Form Personalization form, close the User form and reopen it. Your User form should now look like this: Notice that the User, Customer and Fax fields are no longer displayed.

[](http://www.flickr.com/photos/34041331@N00/259740941/)

Other basic Forms Personalization are just as easy as this example, such as changing prompts, or hiding tabs.

Login as hp/password. Go to the responsibility System Administrator---Security---User---Define. This is the basic Users form as shown here:

[](http://www.flickr.com/photos/34041331@N00/258728479/)

From the Help menu(M) > Diagnostics > Custom Code > Personalize.  
If some Rules are already defined in the Users form, you will see them in this window.  
Add a new Rule by selecting (M) File > New or by clicking the (+) sign while being on the existing last rule or simply click on a new blank row  
[](http://www.flickr.com/photos/34041331@N00/258786454/)

Rules Seq.: 40

Description: Setting Email Address

Trigger Event: WHEN-NEW-ITEM-INSTANCE

Trigger Object :USER.EMAIL\_ADDRESS

Processing Mode : Not in Enter Query Mode

Context or Scope: User level and Value : HP

Select the Actions tab. This is the Actions tab of Rule seq. 40

Action Seq.: 1

Type: Property

Description : Set Email

Object Type: Item

Target Object : USER.EMAIL\_ADDRESS

Property Name : Value

Click Inside Value box. Press button “Insert ‘Get’ Expression”. In the dialog box which pops up :

Object Type :Item

Target Object : USER.USER\_NAME

Property : Value

Expression(will become) : ${item.user.user\_name.value}

Click OK Button

The value box for USER.EMAIL\_ADDRESS will now have : ${item.user.user\_name.value}

Change it to : concat(${item.user.user\_name.value},'@capgemini.com')

Click Apply Now(B) and Ctrl+s.

Ensure that in the first Window of Personalization, the check box is ticked/selected for every Rule that you want to be applied.

---------------

For the above Rule 40, put the following in the CONDITION, in the Condition tab :

:system.record\_status = 'INSERT' and (:USER.SUPPLIER\_NAME is null and :USER.CUSTOMER\_NAME is null)

Which means if he is a supplier or customer then you don’t want to set his email id.

**Profile Category**

Allocate the responsibility “Functional Administrator” to the user hp.

Login with the user hp. Go to the Responsibility Functional Administrator --- Core Services(Tab) --- Profile Categories(Tab)

Click on Create Profile Category button

Name : HP Profile Category

Code :HPPROFCAT

Application : AOL

Description : Hemant Pangam Profile Category

Click Apply button

On the Profile Categories Page, you may search the above created Profile category, by entering the code : HPPROFCAT and click the Go button.

Once the category is found, click on the Update button for “HP Profile Category”, to add some profiles to it.

In the next screen. Click the “Add Profiles” button

On the next screen, search by Name, by typing Java% in the text box, and click the Go button.

From the Java related profiles shown, select both the Java Color Scheme and Java Look and Feel, and click the Select button

Thus, both the above profiles have been added to your Profile category HP Profile Category. Click the Apply Button to save these two profiles under your profile category.

Customization is done by making the changes in the cutom.pll. Customization means making few changes according to client requirement to an already existing product/application. Customization is making the changes to the Oracle applications using Oracle tools like Forms, Reports. When you change the code of the forms/reports etc.

Customization Steps

Ensure that Oracle Developer suite is installed in your local system.  
Copying the TEMPLATE.fmb and APPSTAND.fmb files

Copy the TEMPLATE.fmb and APPSTAND.fmb files from server(/u05/oracle/R12VIS/apps/apps\_st/appl/au/12.0.0/forms/US) to your Local PC on any specific path like C:\DevSuiteHome\_5\forms

Copy all the required .pll’s from Server(/u05/oracle/R12VIS/apps/apps\_st/appl/au/12.0.0/resource) to Local PC in same location C:\DevSuiteHome\_5\forms. The .pll files can be copied using WINSCP and going to the concerned folder on the RHS and using the filter in the toolbar at the top and specifying the filter mask as \*.pll

Required Pll’s

APPCORE.pll  
FNDSQF.pll  
GLOBE.pll  
JE.pll  
JL.pll  
JA.pll  
VERT.pll  
GHR.pll  
APPCORE2.pll  
HRKPI.pll  
PQH\_GEN.pll  
PSAC.pll  
PSB.pll  
PSA.pll  
GMS.pll  
FV.pll

IGILUTIL.pll

IGILUTIL2.pll

IGI\_CBC.pll  
CUSTOM.pll

AXF\_CUSTOM.pll

IGI\_CC.pll

IGI\_CIS.pll

IGI\_DOS.pll

IGI\_EXP.pll

IGI\_IAC.pll

IGI\_MHC.pll

IGI\_SIA.pll

IGI\_STP.pll

OPM.pll

VERT1.pll  
VERT2.pll  
VERT3.pll  
VERT4.pll  
VERT5.pll

FNDMOAC.pll

APPDAYPK.pll

Setting Form’s Path in Registry

Open the default.env(environment file) from the location C:\DevSuiteHome\_5\forms\Server. For the variable FORMS\_PATH set or Append FORMS\_PATH C:\DevSuiteHome\_5\forms (“C:\DevSuiteHome\_5\forms”) is the location where all required plls are placed.

Before Appending “C:\oracle\_forms\plsql ”  
After Appending “C:\oracle\_forms\plsql; C:\DevSuiteHome\_5\forms”

Creation of table:

Connect xxdm@r12vis/xxdm

Create table emp as select \* from scott.emp;

Grant all on emp to apps;

Connect apps@r12vis/M0nday

Create synonym emp for xxdm.emp;

Develop the Form using oracle Form Builder

Open the TEMPLATE.fmb file from C:\DevSuiteHome\_5\forms. Save-As to change the file name, Say MYEMPLOYEES.fmb. In the Object Navigator, change the module name also to MYEMPLOYEES instead of TEMPLATE. Create New Block using wizard for emp(synonym) in apps for xxdm.emp table and select new canvas during wizard.

Setting Properties

For the new Canvas, set the Window property “BLOCKNAME” or you can create new window as well and assign it to your canvas.

From the Module property Change the First Navigation Data Block property to EMP (Emp is the Block you created earlier).

Apply property classes to each item to ensure the same look and feel as the rest of Oracle Apps forms. For each Module, Canvas, Window etc., apply the Property Classes that is most appropriate.

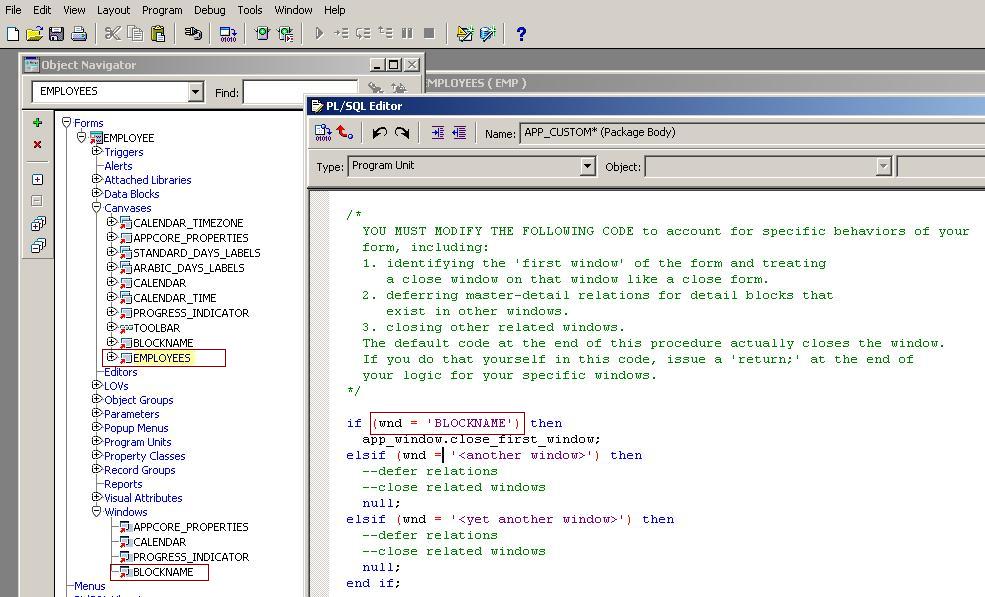
Item Type Property Class

---------------- ----------------------  
Module MODULE  
Canvas CANVAS  
Window WINDOW  
Data Block BLOCK  
Text Item TEXT\_ITEM  
Display Item DISPLAY\_ITEM or EXT\_ITEM\_DISPLAY\_ONLY  
Creation Date CREATION\_OR\_LAST\_UPDATE\_DATE  
Last Update Date CREATION\_OR\_LAST\_UPDATE\_DATE  
To apply the property class, click on the small button (on the right) on the “Subclass information” property of the property sheet of each item. In the “Subclass Information” Window choose the property class radio button. Then choose the correct property class for the current item.

Open the APP\_CUSTOM.CLOSE\_WINDOW program Unit and modify the block name as following.  
if (wnd = ‘BLOCKNAME’) then

app\_window.close\_first\_window;

Note that “BlOCKNAME” is window name which we assign to the new canvas



Now Compile your form Ctrl+Shift+K . It should compile successfully.

Now, using WINSCP, transfer your MYEMPLOYEE.fmb file to apps server in the path /u05/oracle/R12VIS/apps/apps\_st/appl/xxdm/12.0.0/forms/US. Set the properties-permissions to 777 for the MYEMPLOYEE.fmb file in the apps server.

Compiling on Apps Server

frmcmp\_batch.sh module=MYEMPLOYEE.fmb userid=apps/M0nday123 output\_file=MYEMPLOYEE.fmx module\_type=form batch=no compile\_all=special

Register the Form in Oracle Apps

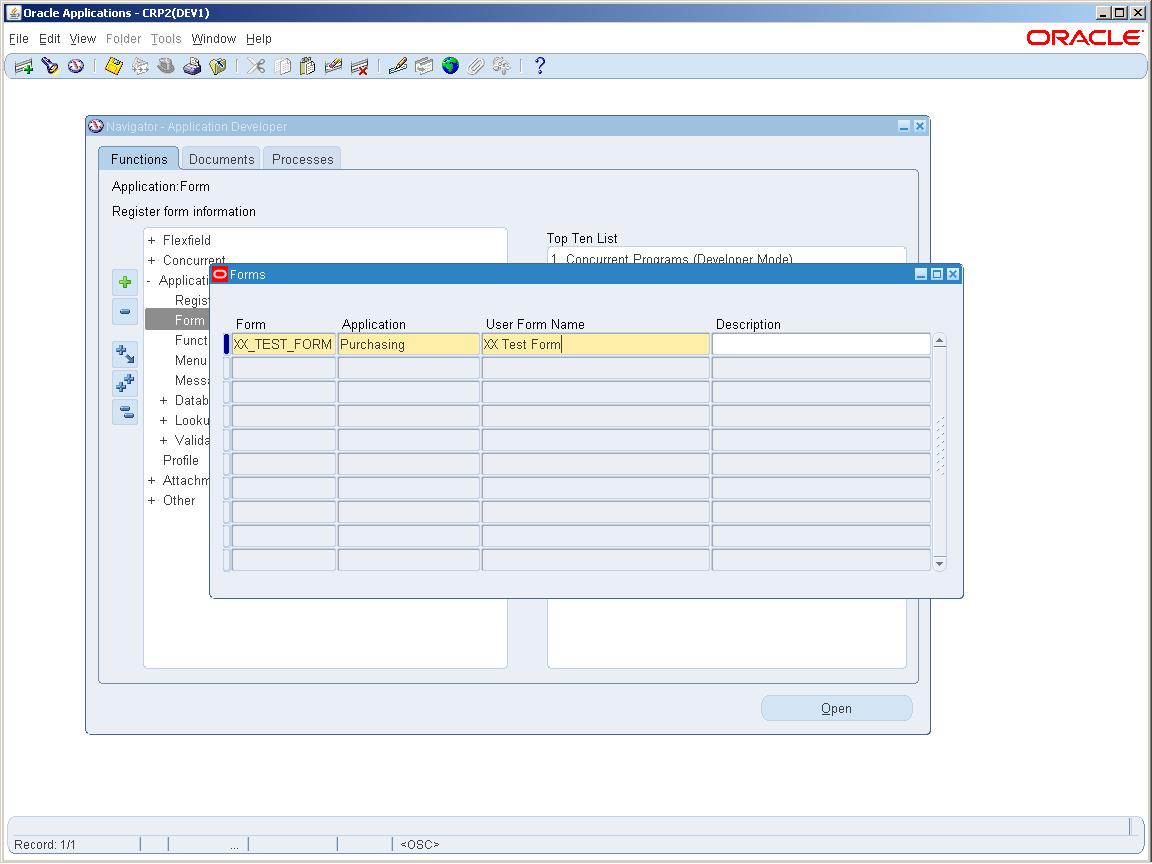
Login to Apps using hp/hemant1 or apps/M0nday123. Go to the Application Developer Responsibility---Application ---Form and define the form here.

Form name=MYEMPLOYEE(which you have saved at server side folder.)

Application : Hemant Pangam ERP (created earlier)

User Form name : HP MYEMPLOYEE

Save it.

[](http://www.oracleport.com/wp-content/uploads/2010/06/oracle-form-customization-2.jpg)

Define the Function for Form.

Go to the Application Developer Responsibility---Application ---Function

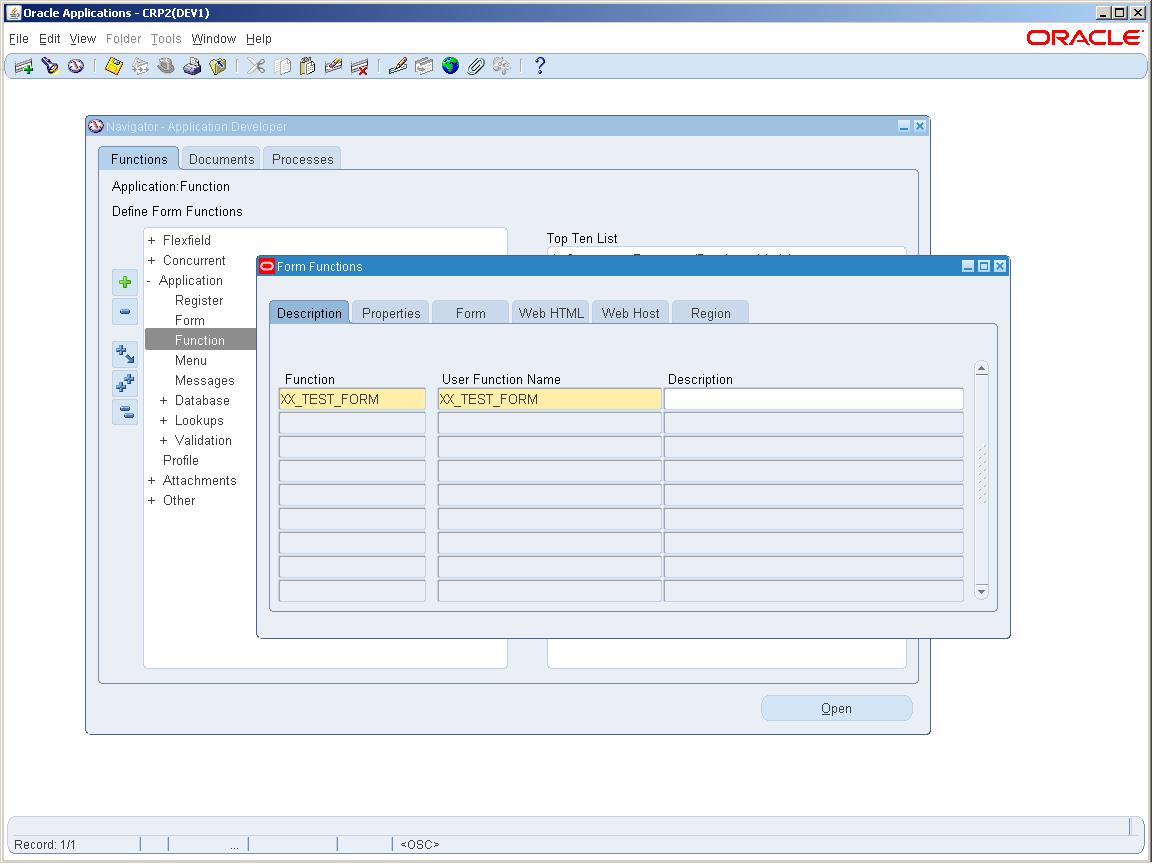
Function :MYEMPFUNC

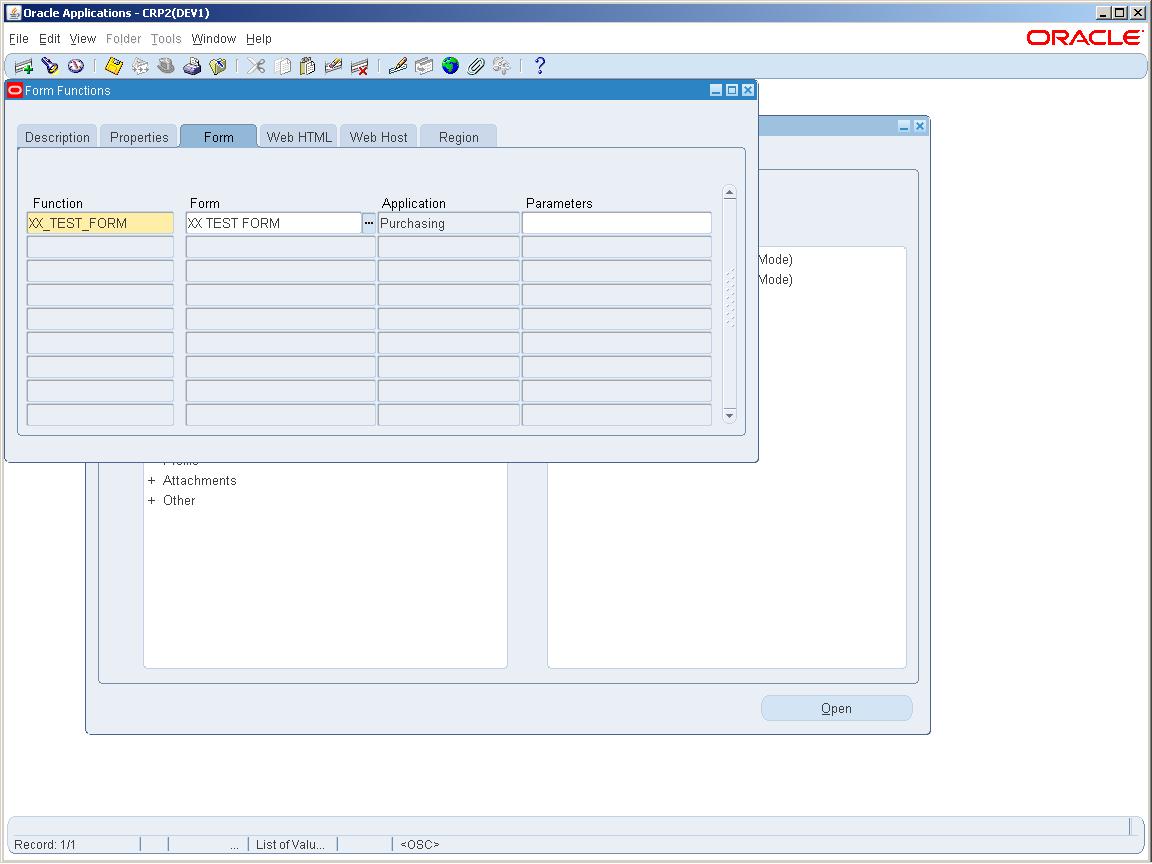
User Function name : HP MYEMPFUNC

Go to the Forms tab, to set the form for this functionality

Form :HP MYEMPLOYEE

Save it

[](http://www.oracleport.com/wp-content/uploads/2010/06/oracle-form-customization-4.jpg)

[](http://www.oracleport.com/wp-content/uploads/2010/06/oracle-form-customization-5.jpg)

Now to assign the form to a Menu

Go to the Application Developer Responsibility---Application ---Menu

Search the earlier created menu : HPERP\_CUSTOM\_MENU

Add a new menu-item or sequence

Seq : 200

Prompt : My Employees

Function :HP MYEMPFUNC

Save it

Login as the user hp/hemant1 to whom the responsibility is assigned which contains the application Hemant Pangam ERP (created earlier).

Identify the following form to customize

Example General Ledger – setup --- Currencies ---- Define

This is the form FNDNLDCX

Go to the AU\_TOP/forms/US

From here copy the FNDNLDCX.fmb to yo local machine. Set the permission to 755

Make some minor changes like changing the labels/prompts of a few text boxes

Save it, and CTRL+SHIFT+K

Now copy the FNDNLDCX.fmb back to AU\_TOP/forms/US

Compile to generate FNDNLDCX.fmx and copy it to FND\_TOP

Now, logout and login again to see the changes