ILP PROGRAM - ORACLE APPLICATIONS

Tata Consultancy Services

AOL Study Guide

Author:	MD. Nazmul Hoda (TCS)
Creation Date:	Jan 7, 2015
Last Updated:	Jan 7, 2015
Document Ref:	ILP/ORACLEAPPS/AOL/01
Version:	DRAFT 1A

Approvals:

<Approver 1>	Santanu Sarkar (TCS)
<Approver 2>	Shubho Chakraborty(TCS)

Document Control

Change Record

Date	Author	Version	Change Reference
07-May-15	MD. Nazmul Hoda	Draft 1a	No Previous Document

Reviewers

Name	Position

Distribution

Copy No.	Name	Location
1	Library Master	Project Library
2		Project Manager
3		
4		

Note To Holders:

If you receive an electronic copy of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a hard copy of this document, please write your name on the front cover, for document control purposes.

Contents

Table of Contents

How to use this manual

Video1: Script: Vid1-Introduction to the chapter and its content – Face recording.

This vide will introduce the material covered in this pdf, the goals,

1. How this document is organized
2. What is the purpose of this document
3. What will you achieve after going through the document and related videos
4. How to read this document
5. How does it relate to the work you will be doing on real project
6. Reference to other reading materials for further references

This manual has been organized as a step by step guide to teach how to create reports using Oracle Developer Suite 10G. The target audience is new comes to Oracle Developer suite. It assumes that the reader has basic knowledge of Oracle concepts and PL/SQL. After completing this course, you will be able to crate variety of reports using Oracle Developer Suite 10G.

This manual is organized to be read in a serial fashion and follow the instructions given in the document as it is. Practical examples are given in each section to guide you through every step. The tables referred here are common (shared) tables used by different batches, so care should be taken not to delete or update the rows which does not belong to you, this may create problem for the other batches. At the end of the course, you should delete the data you have created.

There are several symbols used to designate particular sections, which are described below:

- Describes the purpose of the section.

- Notes relevent to the scetion above

- This denotes the task to be completed by the audience on his own PC. The layout of the output has to be followed as it is. For any confusion, the faculty shoul dbe contacted.

1. Introduction to AOL

The AOL is a collection of pre-built application components and facilities and it consists of forms, subroutines, concurrent programs and reports, database tables and objects, messages, menus, responsibilities, flexfiled definitions, various guides and library functions. AOL is also the short name for the System Administration module of Oracle which is where you control such things.

2. Terminology

In this section you will learn about the key terminology of the AOL, which will be explained in detail in subsequent sections

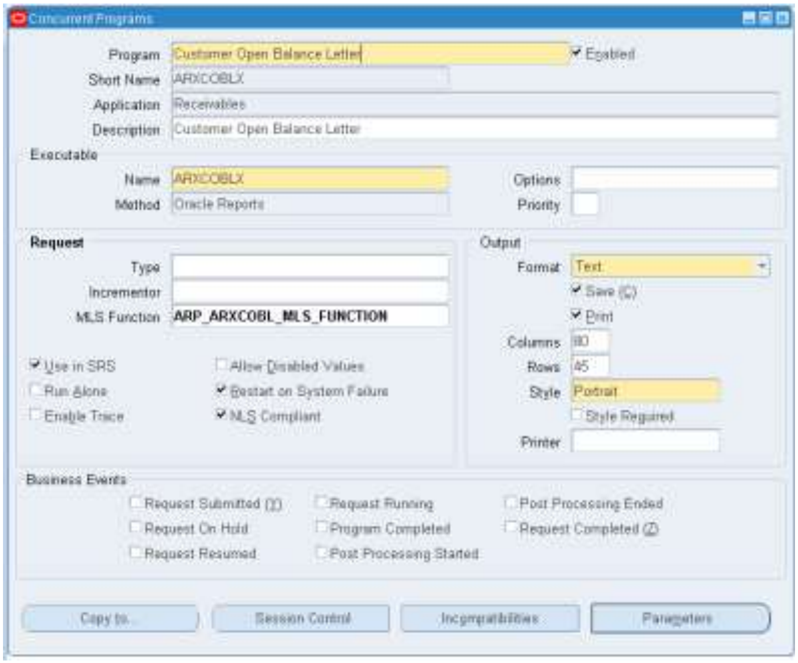
APPS Schema

An ORACLE schema that has access to the complete Oracle Applications data model. This is a database user having super user access. Further detail of this schema is available in the “Technical Architecture” manual.

Concurrent Program

A concurrent program is an executable, which is capable of running a job from within Oracle Applications. A concurrent program has to be submitted from within the application. It can accept parameters at the runtime, and is capable of executing a variety of codes like PL/SQL code, Database stored procedure and function, Shell scripts and binaries. Any piece of code that you want to run from within Oracle Applications has to be registered as a Concurrent Program. A concurrent program can be schedules to run at a particular time, and several options are available to control the behaviors of a concurrent program.

A Concurrent program looks like this:



Concurrent Programs

Program: Customer Open Balance Letter ☑ Enabled
Short Name: ARXCOBLX
Application: Receivables
Description: Customer Open Balance Letter

Executable
Name: ARXCOBLX
Method: Oracle Reports
Options:
Priority:

Request
Type:
Incrementor:
MLS Function: ARP_ARXCOBL_MLS_FUNCTION

☑ Use in SRS
□ Run Alone
□ Enable Trace
□ Allow Disabled Values
☑ Restart on System Failure
☑ NLS Compliant

Output
Format: Text
☑ Save (C)
☑ Print
Columns: 80
Rows: 45
Style: Portrait
□ Style Required
Printer:

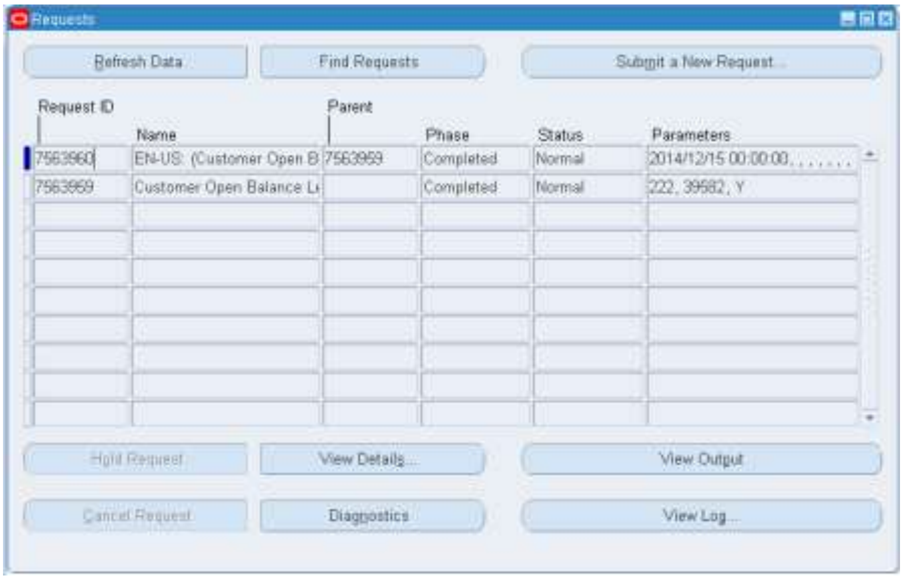
Business Events
□ Request Submitted (Y)
□ Request On Hold
□ Request Resumed
□ Request Running
□ Program Completed
□ Post Processing Started
□ Post Processing Ended
□ Request Completed (Z)

The details will be explained in next chapters.

Concurrent Request

When a program is submitted for running, it is called a Concurrent Request.

The below diagram shows two programs which has completed.



Request ID	Name	Parent	Phase	Status	Parameters
7563960	EN-US: (Customer Open B	7563959	Completed	Normal	2014/12/15 00:00:00, , , , , , ,
7563959	Customer Open Balance L		Completed	Normal	222, 39582, Y

There are few things to note here:

Request ID: Whenever a concurrent program is submitted for run, the system allocates a new request id.

Name: This is the name of the program.

Parent: A Concurrent Program often submits another Concurrent Program, In this case request id – 7563960 has been invoked by request id – 7563959.

Phase: When a Concurrent Program is submitted, it may not start running immediately, and goes through different phases – Pending, Running and Completed/Error. A program can be in pending state for many reasons, for example: Concurrent Manager is not ready to take the job. Then it goes into running state.

Status: This shows that the current status of concurrent request, If everything is fine, it will show – Normal , else it may show – Standby (meaning waiting for Concurrent manager), or Error (if a problem has occurred while running)

Parameters: The parameter values which has been supplied while submitting the program.

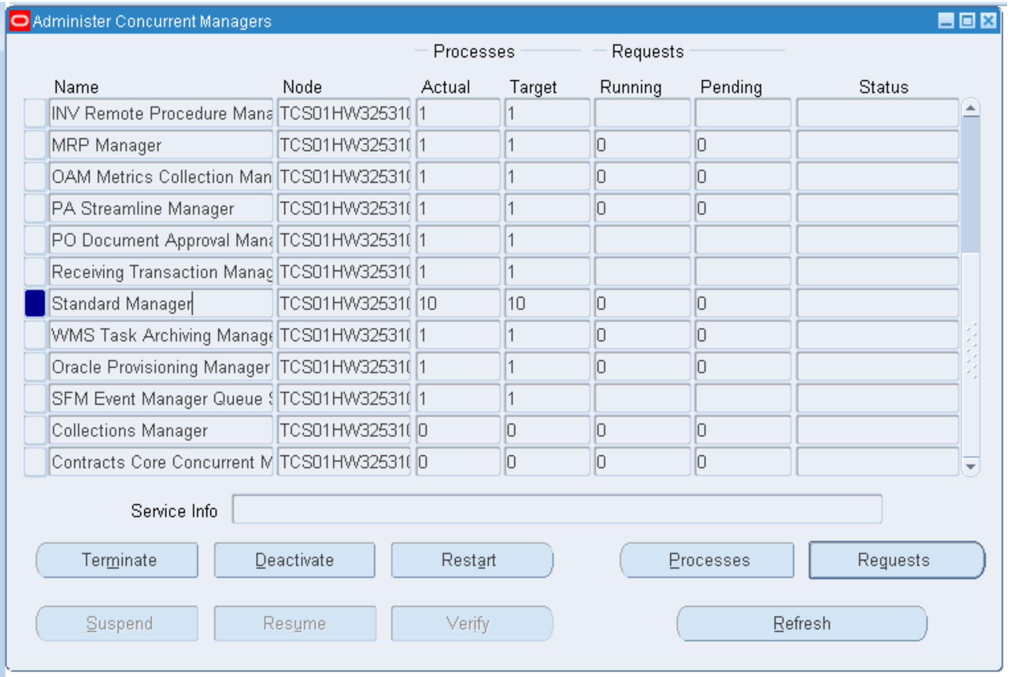
Video1: Script:

In this video, we will see how a concurrent program looks like, and what a concurrent request is.

Concurrent Manager

This is a tool which actually runs the Concurrent Programs which has been submitted. This has an in built scheduler, which can control when a program will run, and on what conditions.

The below picture shows the concurrent managers:



Name	Node	Processes		Requests		Status
		Actual	Target	Running	Pending	
INV Remote Procedure Mana	TCS01HW32531(	1	1			
MRP Manager	TCS01HW32531(	1	1	0	0	
OAM Metrics Collection Man	TCS01HW32531(	1	1	0	0	
PA Streamline Manager	TCS01HW32531(	1	1	0	0	
PO Document Approval Mana	TCS01HW32531(	1	1			
Receiving Transaction Manag	TCS01HW32531(	1	1			
Standard Manager	TCS01HW32531(	10	10	0	0	
WMS Task Archiving Manage	TCS01HW32531(	1	1	0	0	
Oracle Provisioning Manager	TCS01HW32531(	1	1	0	0	
SFM Event Manager Queue (	TCS01HW32531(	1	1			
Collections Manager	TCS01HW32531(	0	0	0	0	
Contracts Core Concurrent M	TCS01HW32531(	0	0	0	0	

Service Info

Note the following:

Name: Name of the Concurrent Manager

Node: The server where the manager is running.

Actual: The actual number of processes, the manager has is running. This means the number of Concurrent Programs the manager can run parallel. In this case, the MRP Manager can run one and the Standard Manager can run 10 Concurrent Programs parallel.

Target: The number of processes the Concurrent Manager has been asked to run. This should be same as the Actual column in normal conditions, but will differ if some of the processes have died.

Running: The number of Concurrent Programs, the manager is actually running at this moment. In this case, none of the managers are running anything, meaning no one has submitted any Concurrent Requests.

Pending: Number of Concurrent Programs waiting to be run

Status: Status of the Concurrent Manager, will show null, if everything is fine, else will show 'Deactivated', if the manager has been deactivated.

Application

An application is a collection of Forms, Reports and PL/SQL objects to meet a particular business objective. There are large numbers of these objects in Oracle Applications, which has been categorized into different applications.

For example, our current installation contains:

- 3978 Forms
- 2273 Reports
- 574690 Database Objects(PACKAGE BODY,TABLE,INDEX,SEQUENCE,PACKAGE,TABLE PARTITION,LOB)

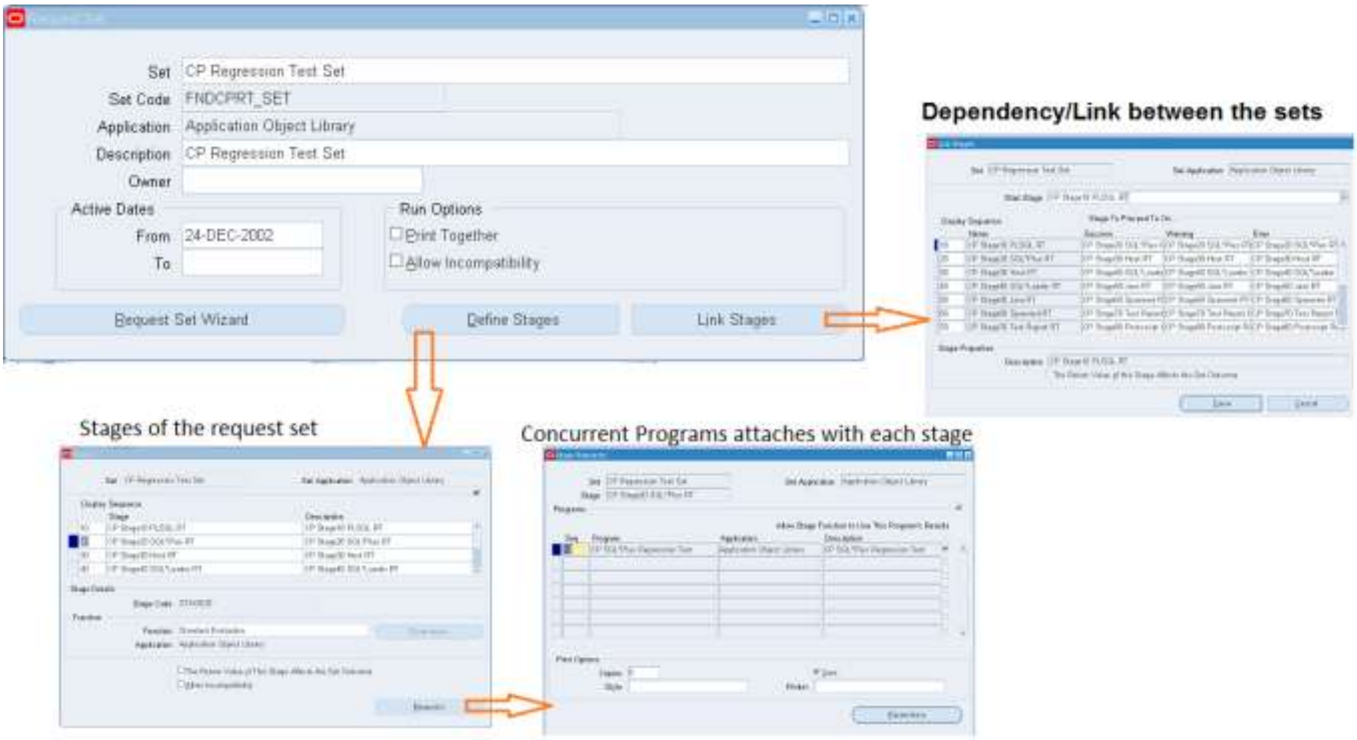
Out of which, the application "**Payables"** holds the following:

- 54 Forms
- 110 Reports
- 2 PACKAGE BODY
- 345 TABLE
- 620 INDEX
- 139 SEQUENCE
- 2 PACKAGE
- 64 TABLE PARTITION
- 4 LOB

Request Set

A Request set is a collection of concurrent programs with optionally linked of execution sequence. When we need to executed a set of programs in a particular order, we create a request set. The programs within a request set can be executed in a particular order and depending on certain conditions.

A sample of Request Set has been illustrated in the following diagram:



Set: CP Regression Test Set

Set Code: FNDCPRT_SET

Application: Application Object Library

Description: CP Regression Test Set

Owner:

Active Dates

From: 24-DEC-2002

To:

Run Options

□ Print Together

□ Allow Incompatibility

Dependency/Link between the sets

Stages of the request set

Concurrent Programs attaches with each stage

Note the following in the above diagram:

Request Set named “**CP Regression Test Set**” has stages namely:

CP Stage10 PLSQL RT

CP Stage20 SQL*Plus RT

CP Stage30 Host RT

CP Stage40 SQL*Loader RT

Each stage has a concurrent program attached to it. In this diagram, we are showing the Concurrent Program attached to the stage “**CP Stage20 SQL*Plus RT**”, and we can see that the Concurrent Program named “**CP SQL*Plus Regression Test**” is attached to this stage.

The diagram also shows the link between the stages. This is done by clicking the **Link Stage button**.

Look at the first row of the Link Stage window, this means:

If Stage **CP Stage10 PLSQL RT** completes successfully, run the stage **CP Stage20 SQL*Plus RT**

If Warning run the stage **CP Stage20 SQL*Plus RT**

If Error run the stage **CP Stage20 SQL*Plus RT**

Responsibility

A collection of functions within an Oracle Application. A responsibility is assigned to a user. A user can have multiple responsibility. User can change the responsibility to get a different set of privileges. This functionality is called segregation of duties.

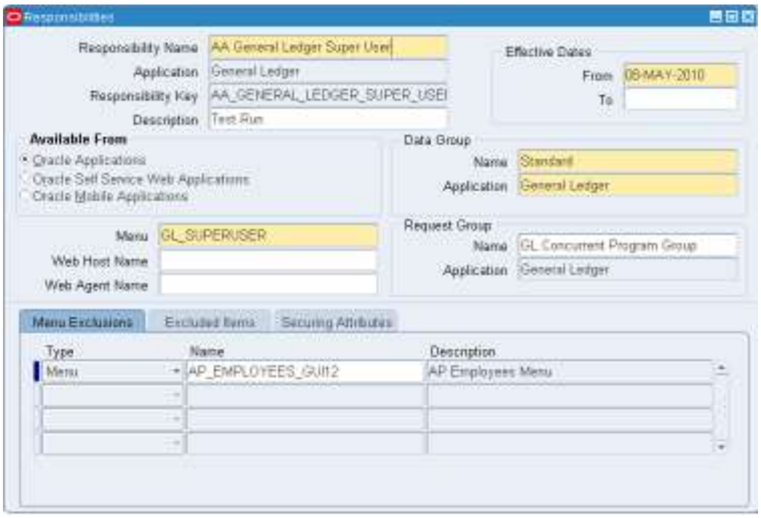
The user gets to do the following functions by virtue of the responsibility assigned to him/her:

1. What form/functions he will be able to access
2. What Concurrent Programs he will be able to run
3. What data will he be able to access through the above cause

The important attribute of a responsibility are:

1. Menu – this decided the navigation menu and submenu the user gets when he logs into Oracle Applications
2. Request Group – this decided what Concurrent programs the user will be able to run
3. Data Group – this decides what data the user will have access to.
4. Menu Exclusion – What function or submenu will be excluded (from the menu assigned to it) to this responsibility.

The below diagram is an example of a responsibility definition:



Responsibilities

Responsibility Name	AA General Ledger Super User
Application	General Ledger
Responsibility Key	AA_GENERAL_LEDGER_SUPER_USEI
Description	Test Run
Effective Dates From	08-MAY-2010
Effective Dates To	

Available From

- ● Oracle Applications
- ○ Oracle Self Service Web Applications
- ○ Oracle Mobile Applications

Data Group Name	Standard
Data Group Application	General Ledger
Menu	GL_SUPERUSER
Web Host Name	
Web Agent Name	
Request Group Name	GL Concurrent Program Group
Request Group Application	General Ledger

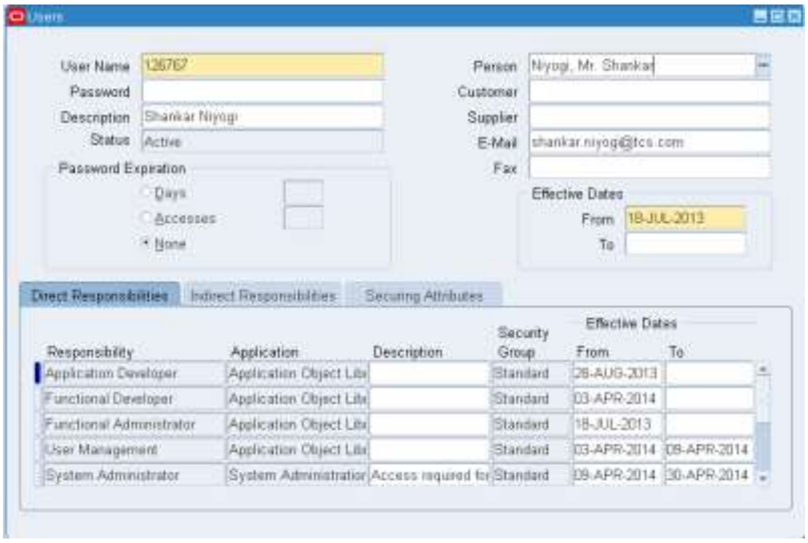
Menu Exclusions | Excluded Items | Securing Attributes

Type	Name	Description
Menu	AP_EMPLOYEES_GUI12	AP Employees Menu

User

A user is the login id for Oracle Applications. A user has to supply a password along with user id to access the application. A user has a set of responsibilities.

The person field is the HR users created for the user in the Human Resource system. The Oracle Application user is linked to this through this form, but all users do not have a person attached to it.



Users

User Name	126767
Password	
Description	Shankar Niyogi
Status	Active
Person	Niyogi, Mr. Shankar
Customer	
Supplier	
E-Mail	shankar.niyogi@tcs.com
Fax	
Effective Dates From	18-JUL-2013
Effective Dates To	

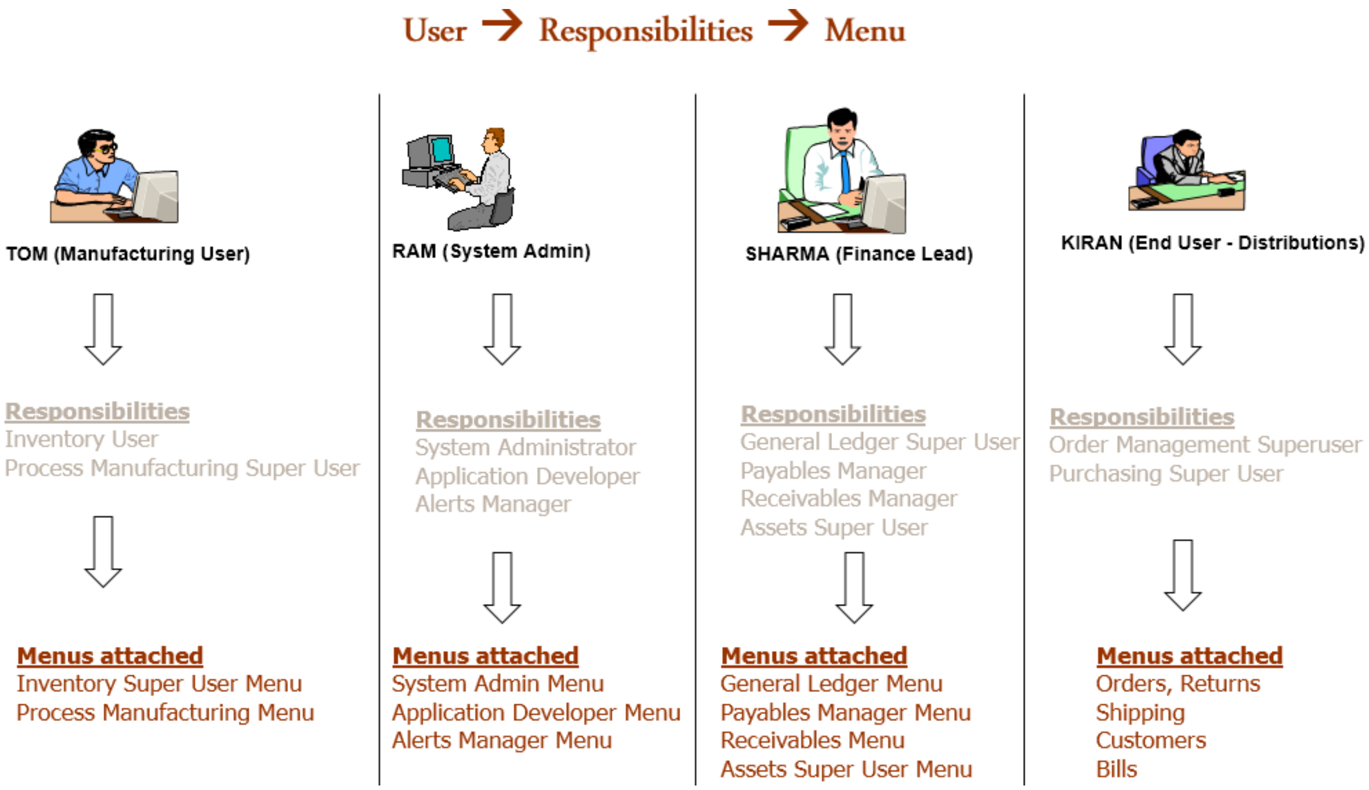
Password Expiration

- ○ Days
- ○ Accesses
- ● None

Direct Responsibilities | Indirect Responsibilities | Securing Attributes

				Effective Dates	
Responsibility	Application	Description	Security Group	From	To
Application Developer	Application Object Libı		Standard	28-AUG-2013	
Functional Developer	Application Object Libı		Standard	03-APR-2014	
Functional Administrator	Application Object Libı		Standard	18-JUL-2013	
User Management	Application Object Libı		Standard	03-APR-2014	08-APR-2014
System Administrator	System Administratior	Access required for	Standard	09-APR-2014	30-APR-2014

The relationship between User > Responsibility and Menu is depicted in the diagram below



User → Responsibilities → Menu

TOM (Manufacturing User)

Responsibilities
Inventory User
Process Manufacturing Super User

Menus attached
Inventory Super User Menu
Process Manufacturing Menu

RAM (System Admin)

Responsibilities
System Administrator
Application Developer
Alerts Manager

Menus attached
System Admin Menu
Application Developer Menu
Alerts Manager Menu

SHARMA (Finance Lead)

Responsibilities
General Ledger Super User
Payables Manager
Receivables Manager
Assets Super User

Menus attached
General Ledger Menu
Payables Manager Menu
Receivables Menu
Assets Super User Menu

KIRAN (End User - Distributions)

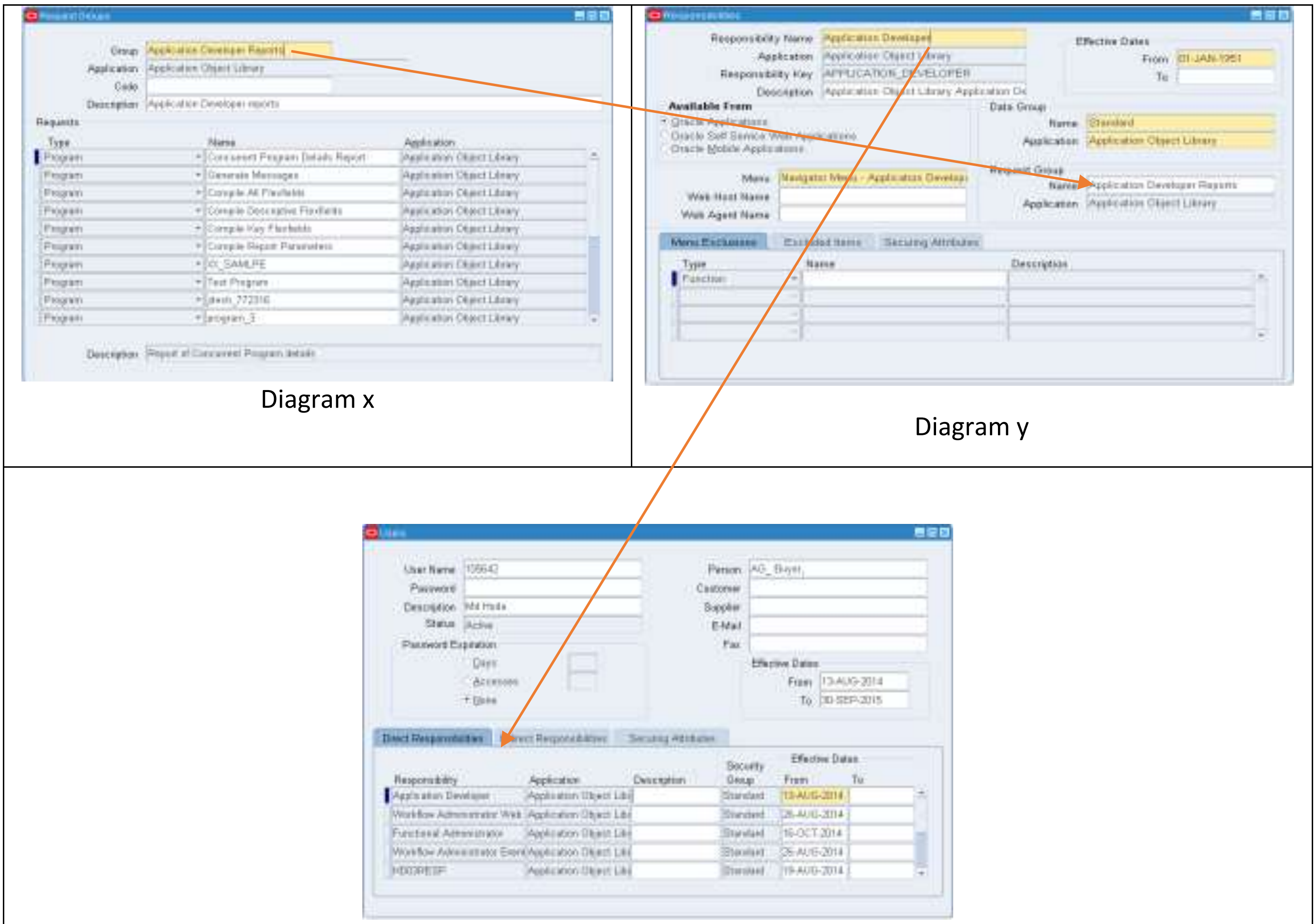
Responsibilities
Order Management Superuser
Purchasing Super User

Menus attached
Orders, Returns
Shipping
Customers
Bills

Request Group

A request group is a collection of Concurrent Programs. The request group is assigned to a responsibility. User having this responsibility can run these programs.

The below diagram shows the Concurrent Programs attached to the Request Group “**Application Developer Reports”** (Diagram x) **.** The User Name – 105642 has the responsibility Application Developer (Diagram z). This responsibility is attached to the request group “**Application Developer Reports**” (Diagram y). This Request Group has a set of programs attached to it (Diagram x), so, the user **105642** will be able to run all of these Concurrent Programs.



Group: Application Developer Reports
Application: Application Object Library
Code:
Description: Application Developer reports

Requests

Type	Name	Application
Program	Concurrent Program Details Report	Application Object Library
Program	Generate Messages	Application Object Library
Program	Compile All Flexfields	Application Object Library
Program	Compile Descriptive Flexfields	Application Object Library
Program	Compile Key Flexfields	Application Object Library
Program	Compile Report Parameters	Application Object Library
Program	XX_SAMLPE	Application Object Library
Program	Test Program	Application Object Library
Program	desh_772316	Application Object Library
Program	program_3	Application Object Library

Description: Report of Concurrent Program details

Diagram x

Responsibility Name: Application Developer
Application: Application Object Library
Responsibility Key: APPLICATION_DEVELOPER
Description: Application Object Library Application De
Effective Dates From: 01-JAN-1951 To:
Available From: ● Oracle Applications ○ Oracle Self Service Web Applications ○ Oracle Mobile Applications
Data Group Name: Standard
Data Group Application: Application Object Library
Menu: Navigator Menu - Application Develop
Web Host Name:
Web Agent Name:
Request Group Name: Application Developer Reports
Request Group Application: Application Object Library

Type	Name	Description
Function		

Diagram y

User Name: 105642
Password:
Description: Md Huda
Status: Active
Password Expiration: ○ Days ○ Accesses ● None
Person: AG_ Buyer,
Customer:
Supplier:
E-Mail:
Fax:
Effective Dates From: 13-AUG-2014 To: 30-SEP-2015

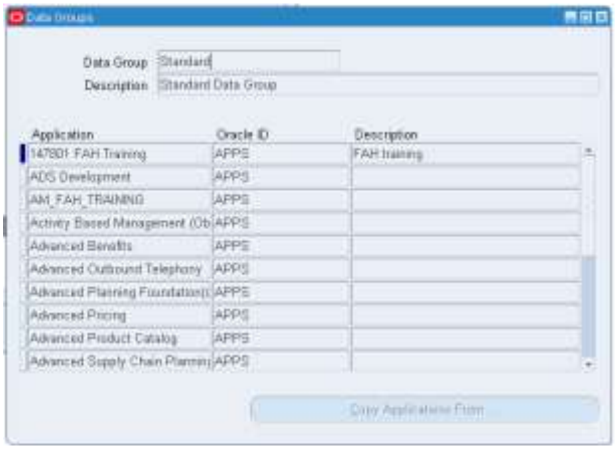
Responsibility	Application	Description	Security Group	Effective Dates From	To
Application Developer	Application Object Lib		Standard	13-AUG-2014	
Workflow Administrator Web	Application Object Lib		Standard	26-AUG-2014	
Functional Administrator	Application Object Lib		Standard	16-OCT-2014	
Workflow Administrator Event	Application Object Lib		Standard	26-AUG-2014	
HDCRESP	Application Object Lib		Standard	19-AUG-2014	

Diagram z

Data group

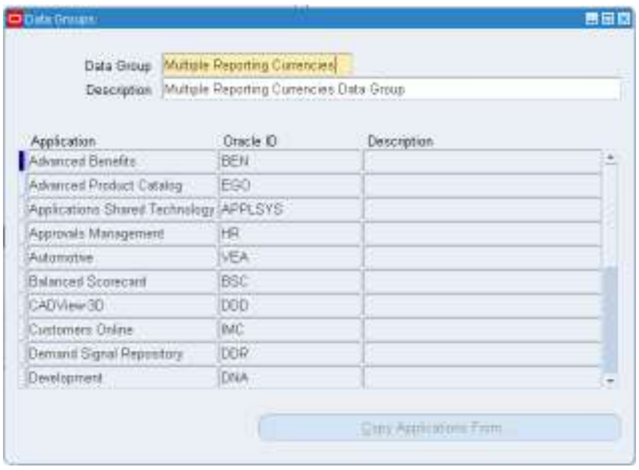
A Data Group defines the mapping between Oracle Applications products and ORACLE user IDs. A Data Group determines which Oracle database accounts will be sued by an application to connect through forms or concurrent programs to the database.

Data group named **Standard** is created by default, and other data group can be created as needed.



Data Group: Standard
Description: Standard Data Group

Application	Oracle ID	Description
147801 FAH Training	APPS	FAH training
ADS Development	APPS	
AM_FAH_TRAINING	APPS	
Activity Based Management (Ob	APPS	
Advanced Benefits	APPS	
Advanced Outbound Telephony	APPS	
Advanced Planning Foundation(	APPS	
Advanced Pricing	APPS	
Advanced Product Catalog	APPS	
Advanced Supply Chain Plannin	APPS	



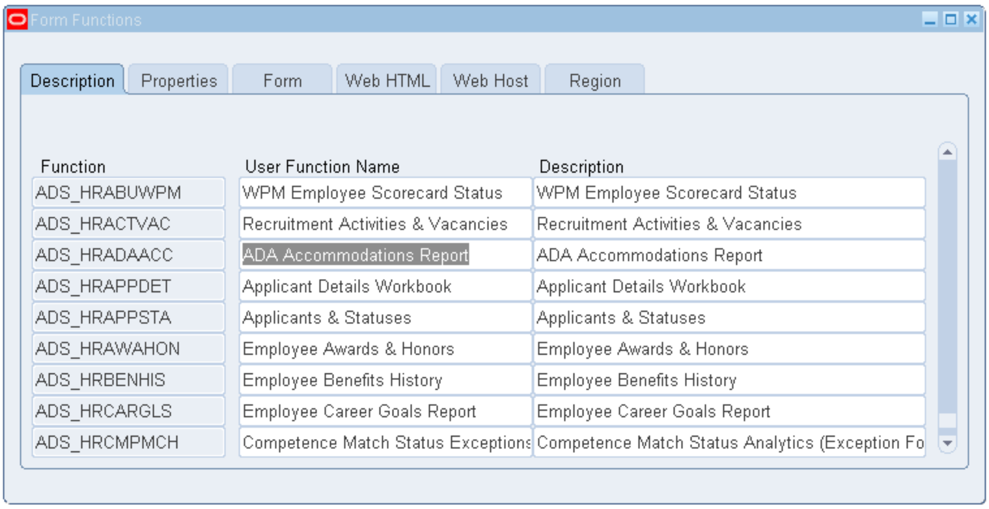
Data Group: Multiple Reporting Currencies
Description: Multiple Reporting Currencies Data Group

Application	Oracle ID	Description
Advanced Benefits	BEN	
Advanced Product Catalog	EGO	
Applications Shared Technology	APPLSYS	
Approvals Management	HR	
Automotive	VEA	
Balanced Scorecard	BSC	
CADView-3D	DDD	
Customers Online	IMC	
Demand Signal Repository	DDR	
Development	DNA	

The above diagram shows two data groups defined in our instance.

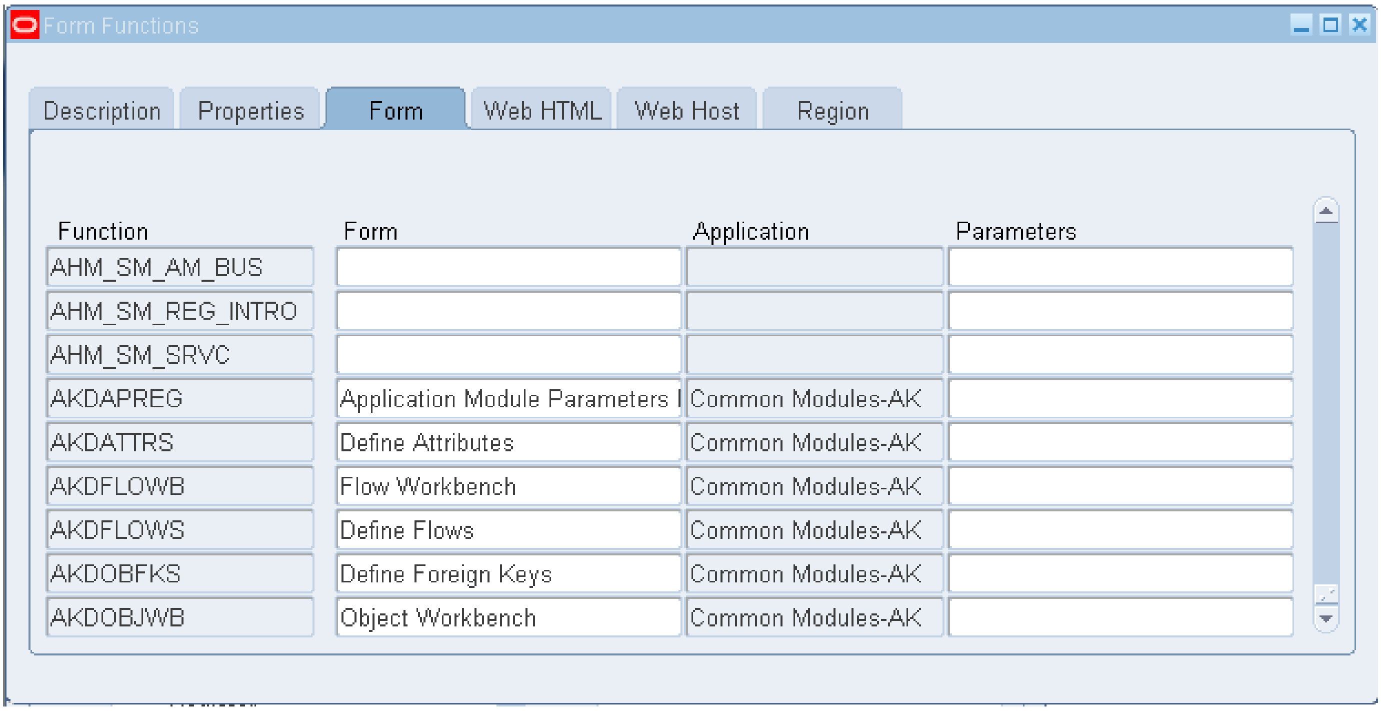
Function

A function is attached to a menu. When the user clicks on a menu item, he invokes a function. A function can be a form, or a URL or jsp page.



Function	User Function Name	Description
ADS_HRABUWPM	WPM Employee Scorecard Status	WPM Employee Scorecard Status
ADS_HRACTVAC	Recruitment Activities & Vacancies	Recruitment Activities & Vacancies
ADS_HRADAACC	ADA Accommodations Report	ADA Accommodations Report
ADS_HRAPPDET	Applicant Details Workbook	Applicant Details Workbook
ADS_HRAPPSTA	Applicants & Statuses	Applicants & Statuses
ADS_HRAWAHON	Employee Awards & Honors	Employee Awards & Honors
ADS_HRBENHIS	Employee Benefits History	Employee Benefits History
ADS_HRCARGLS	Employee Career Goals Report	Employee Career Goals Report
ADS_HRCMPMCH	Competence Match Status Exceptions	Competence Match Status Analytics (Exception Fo

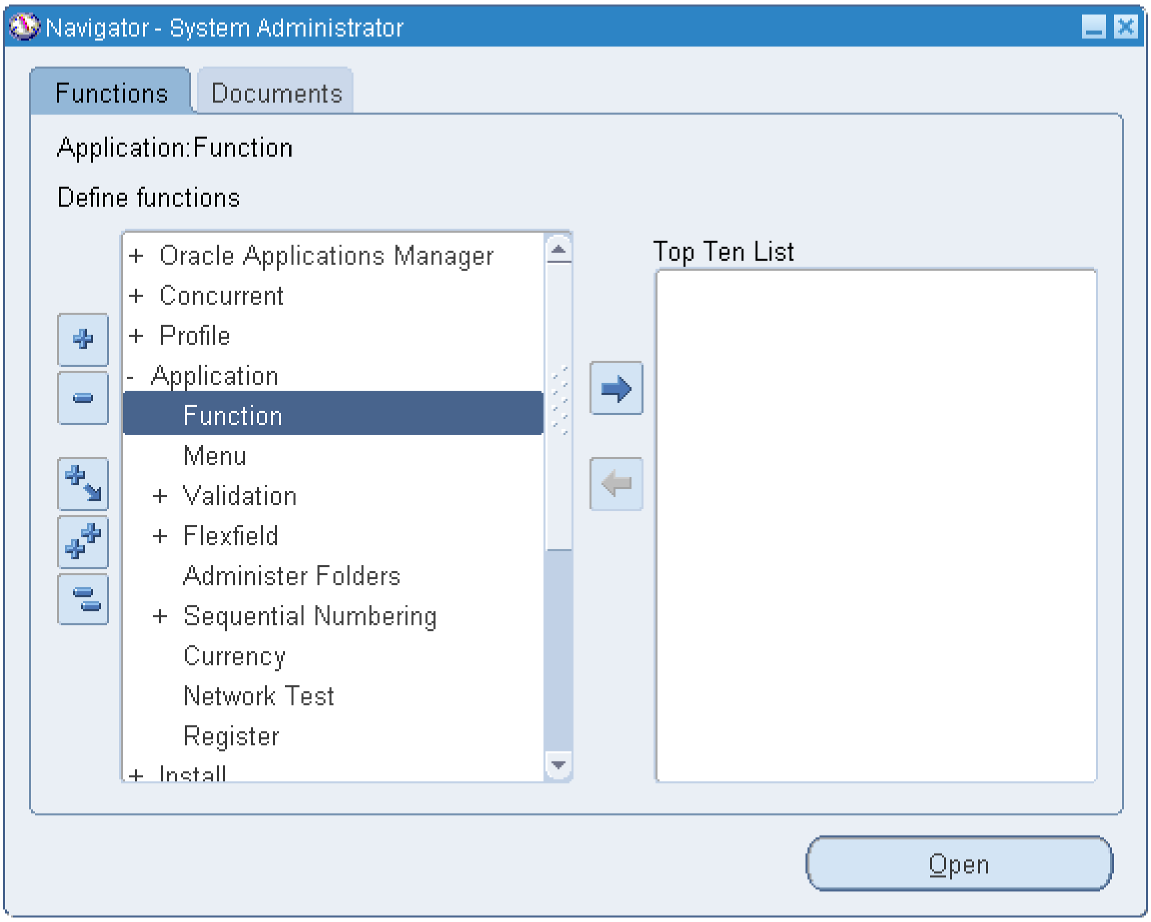
The above diagram shows the definition of functions, with different tabs. You need to click on the relevant tab for the registration. For example, if you want to define the function to call a form, click on the Form tab and enter the form detail which needs to be called. For example, the Form named 'Define Attributes' has been attached to the function – AKDATTRS.



Function	Form	Application	Parameters
AHM_SM_AM_BUS			
AHM_SM_REG_INTRO			
AHM_SM_SRVC			
AKDAPREG	Application Module Parameters I	Common Modules-AK	
AKDATTRS	Define Attributes	Common Modules-AK	
AKDFLOWB	Flow Workbench	Common Modules-AK	
AKDFLOWS	Define Flows	Common Modules-AK	
AKDOBFKS	Define Foreign Keys	Common Modules-AK	
AKDOBJWB	Object Workbench	Common Modules-AK	

Menu

Collection of its submenus and functions in the navigator window. In the below screenshot, a user is logged in with a responsibility – 'System Administrator' and he has a menu, which has items shown below. This is called a menu for the responsibility 'System Administrator'.

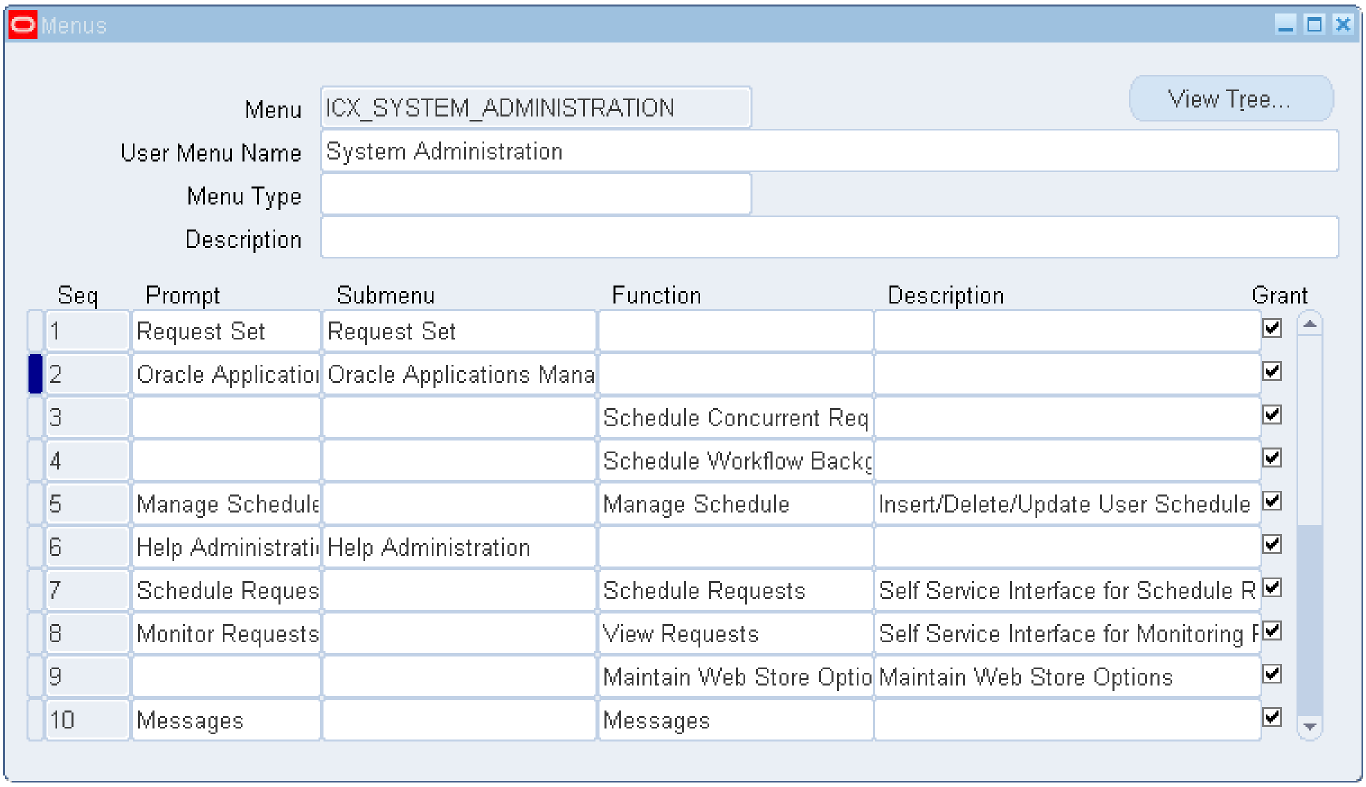


Application:Function

Define functions

- + Oracle Applications Manager
- + Concurrent
- + Profile
- - Application
  - Function
  - Menu
  - + Validation
  - + Flexfield
  - Administer Folders
  - + Sequential Numbering
  - Currency
  - Network Test
  - Register
- + Install

The below screenshot shows, the place where the menu structure has been defined:



Menu	ICX_SYSTEM_ADMINISTRATION
User Menu Name	System Administration
Menu Type	
Description	

Seq	Prompt	Submenu	Function	Description	Grant
1	Request Set	Request Set			☑
2	Oracle Applicatio	Oracle Applications Mana			☑
3			Schedule Concurrent Req		☑
4			Schedule Workflow Backg		☑
5	Manage Schedule		Manage Schedule	Insert/Delete/Update User Schedule	☑
6	Help Administrati	Help Administration			☑
7	Schedule Reques		Schedule Requests	Self Service Interface for Schedule R	☑
8	Monitor Requests		View Requests	Self Service Interface for Monitoring F	☑
9			Maintain Web Store Optio	Maintain Web Store Options	☑
10	Messages		Messages		☑

Note the following:

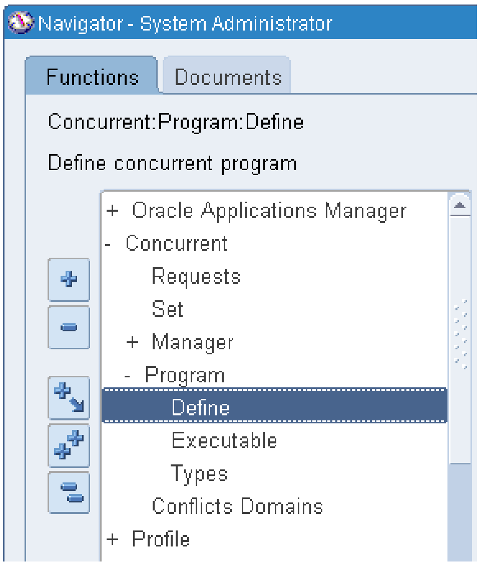
- A menu has a name (here - ICX_SYSTEM_ADMINISTRATION), a prompt which the user will see . Try to correlate the entries appearing in previous diagram and these sentries.
- A menu can have a submenu(optional), which can have further submenus
- A submenu will appear with a '+' symbol, and will open up another menu tree when clicked.
- Each entry in a menu has a sequence number, item will appear in the same sequence at runtime
- Each menu item should have a function attached.
- When a user clicks on a menu item, the function attached to it fires and invoked the Form or Web page or a jsp page attached to it. After finishing the task, the control comes back to the menu.

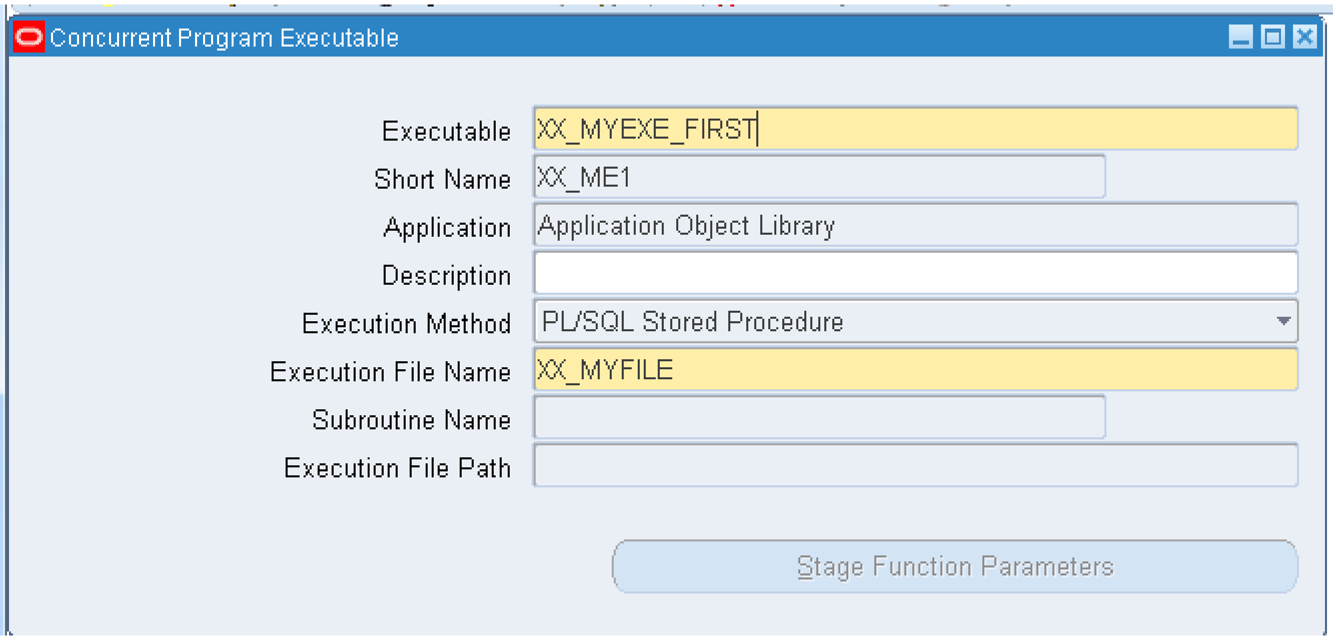
3. Concurrent Program registration

In this section you will learn how to register a concurrent program in detail. This will cover the important fields and navigation to be followed.

Step1: Define Concurrent Program Executable

You need to change the responsibility to '**System Administrator'** or '**Application Developer**' and follow the navigation: Concurrent > Program > Executable

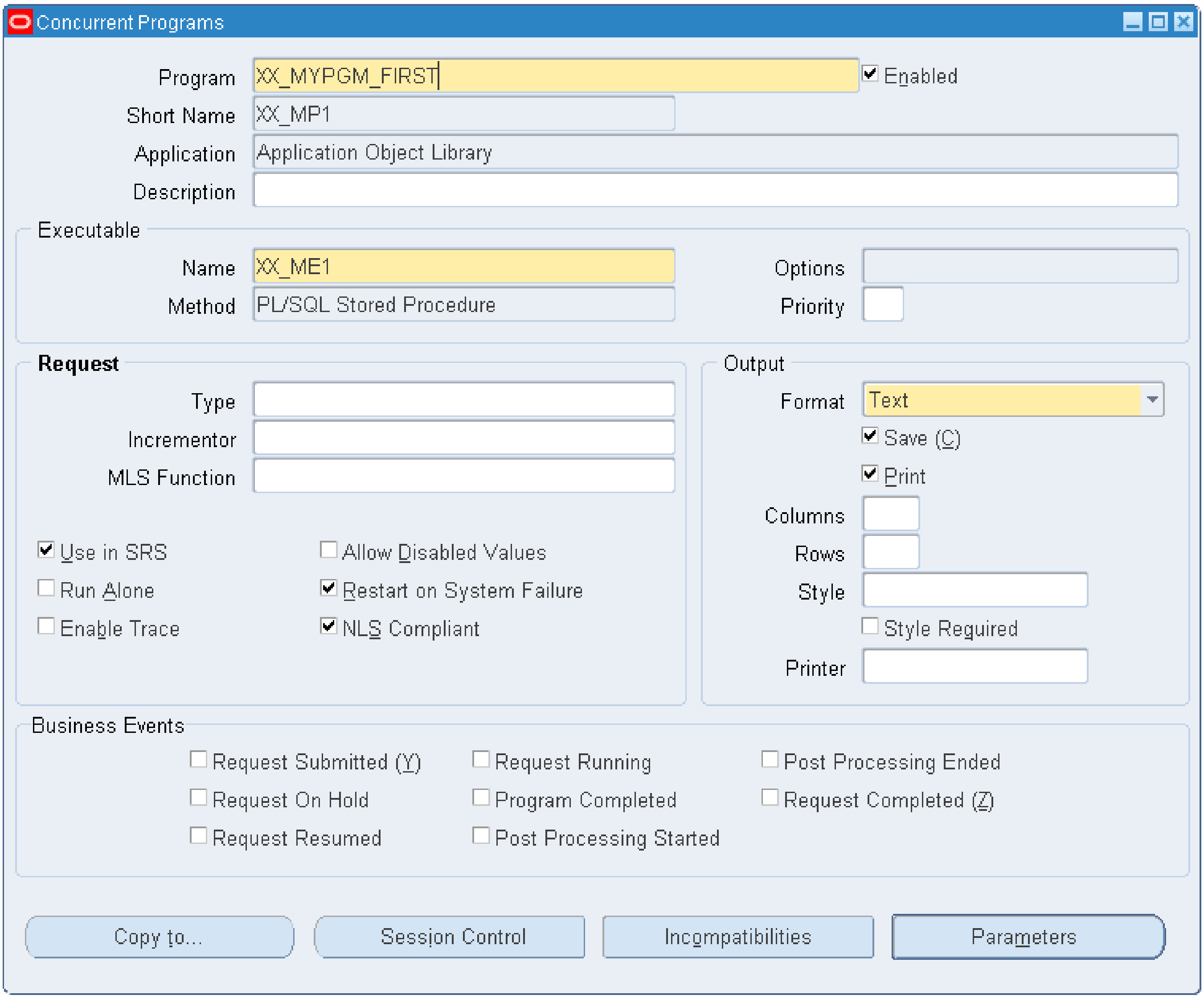

Navigator - System Administrator
Functions
Documents
Concurrent:Program:Define
Define concurrent program
+ Oracle Applications Manager
- Concurrent
Requests
Set
+ Manager
- Program
Define
Executable
Types
Conflicts Domains
+ Profile



Concurrent Program Executable

Executable	XX_MYEXE_FIRST
Short Name	XX_ME1
Application	Application Object Library
Description	
Execution Method	PL/SQL Stored Procedure
Execution File Name	XX_MYFILE
Subroutine Name	
Execution File Path	

Step2: Define Concurrent Program



Program: XX_MYPGM_FIRST ☑ Enabled

Short Name: XX_MP1

Application: Application Object Library

Description:

Executable

Name: XX_ME1

Options:

Method: PL/SQL Stored Procedure

Priority:

Request

Type:

Incrementor:

MLS Function:

☑ Use in SRS ☐ Allow Disabled Values

☐ Run Alone ☑ Restart on System Failure

☐ Enable Trace ☑ NLS Compliant

Output

Format: Text

☑ Save (C)

☑ Print

Columns:

Rows:

Style:

☐ Style Required

Printer:

Business Events

☐ Request Submitted (Y) ☐ Request Running ☐ Post Processing Ended

☐ Request On Hold ☐ Program Completed ☐ Request Completed (Z)

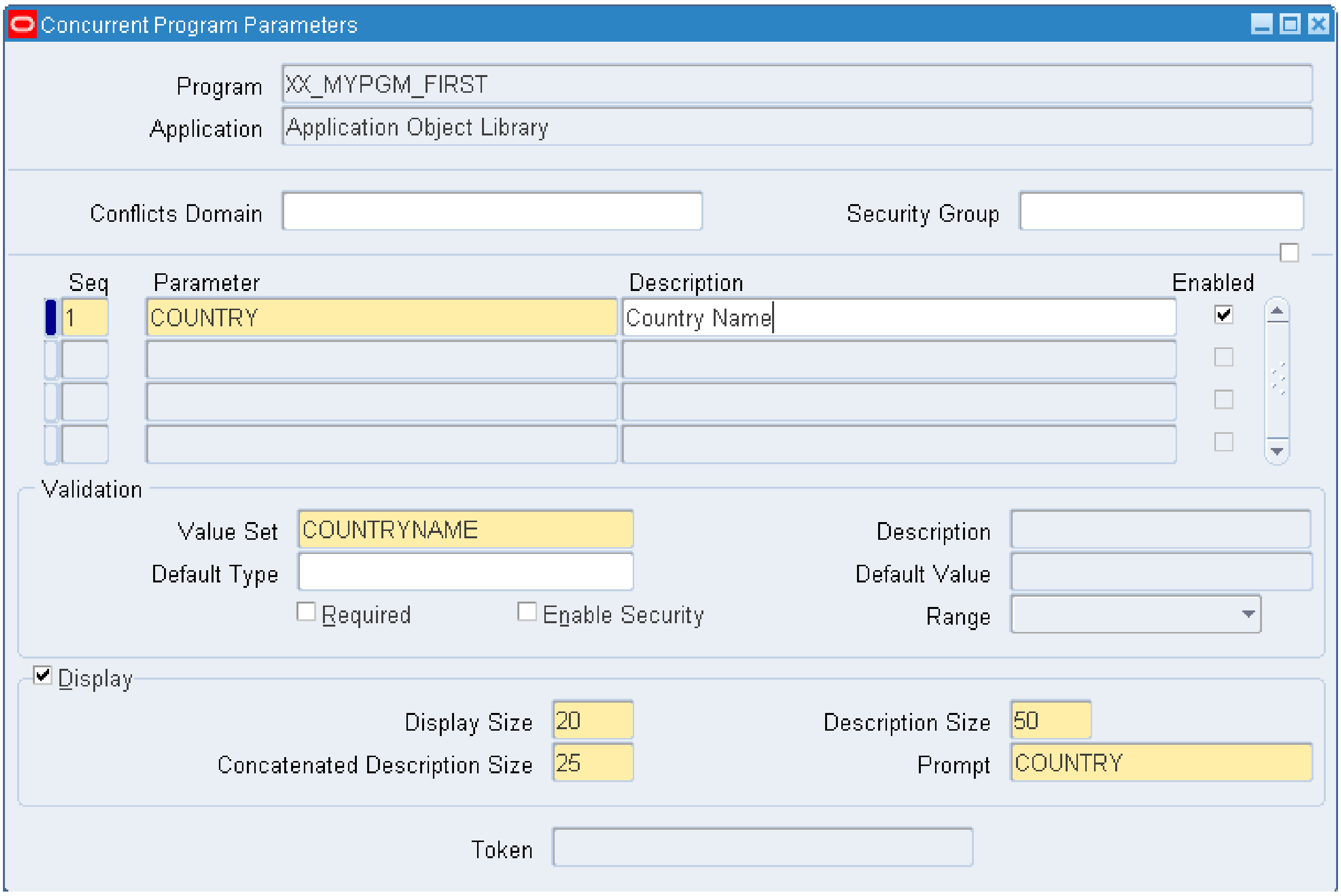
☐ Request Resumed ☐ Post Processing Started

Note: The **Short Name** field in the executable definition screen, must match the **Name** field in the Concurrent Program definition screen.

Click on the Parameter button to crate runtime parameters (if any) to be passed to the Concurrent Program.

Step3: Define Parameters (Optional)

Parameters have a sequence number and they will appear to the user at runtime in the same sequence.



Concurrent Program Parameters

Program: XX_MYPGM_FIRST
Application: Application Object Library

Conflicts Domain:
Security Group:

Seq	Parameter	Description	Enabled
1	COUNTRY	Country Name	☑
			□
			□
			□

Validation

Value Set: COUNTRYNAME
Default Type:
□ Required □ Enable Security
Description:
Default Value:
Range:

☑ Display

Display Size: 20
Concatenated Description Size: 25
Description Size: 50
Prompt: COUNTRY

Token:

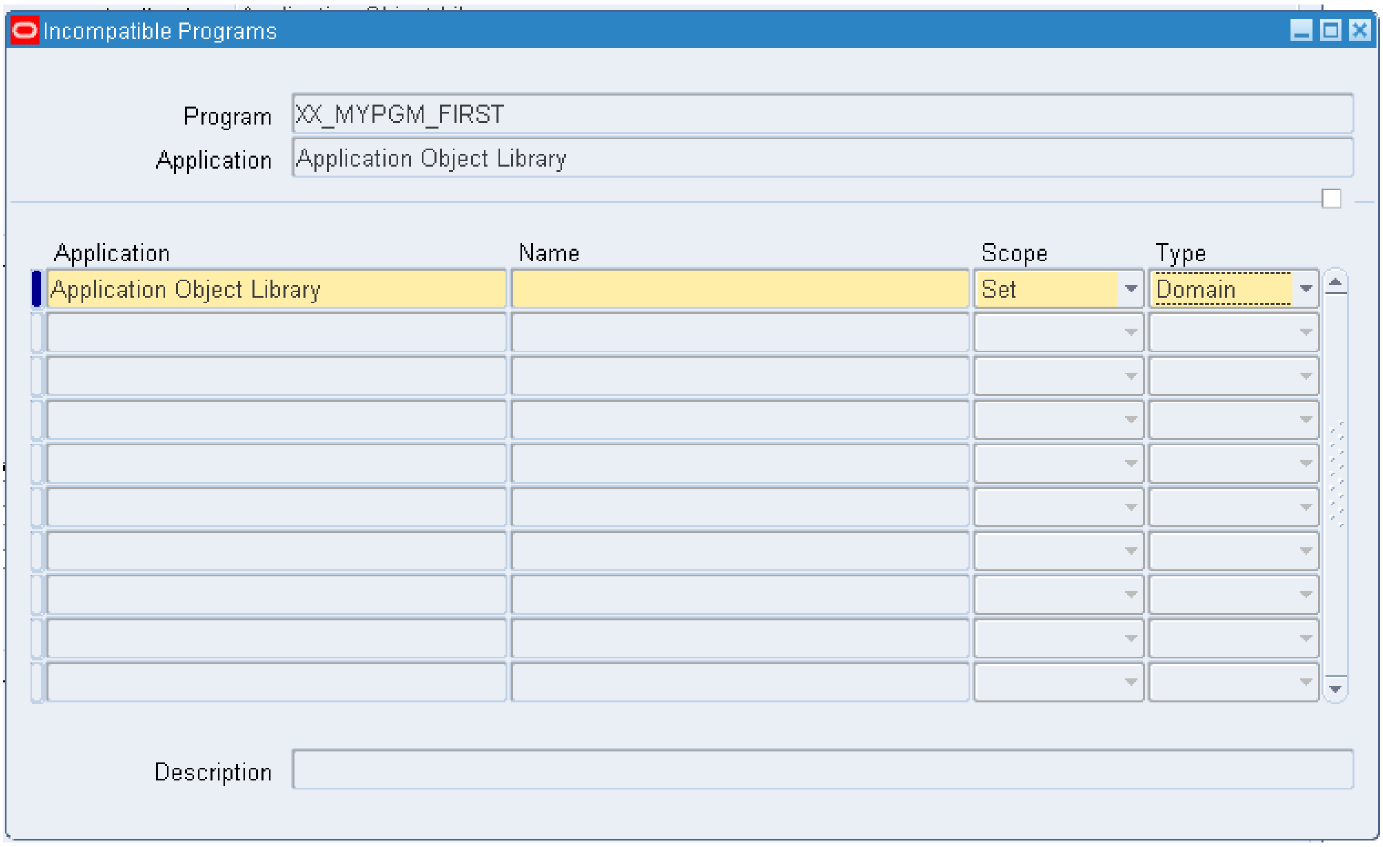
Step4: Attach valueset to parameters(optional)

A Parameter is validated against a value set. I.e , if you select a valuset = **50 Characters**, then at runtime, you have to enter 50 characters to this parameter.

Step4: Define Incompatibilities (optional)

If a program **'A'** is defined to be incompatible with program **'B'**, then program **'A'** cannot run, when program 'B' is running.

To do this, click on the **Incompatibilities** button and define the incompatible program.



Program: XX_MYPGM_FIRST

Application: Application Object Library

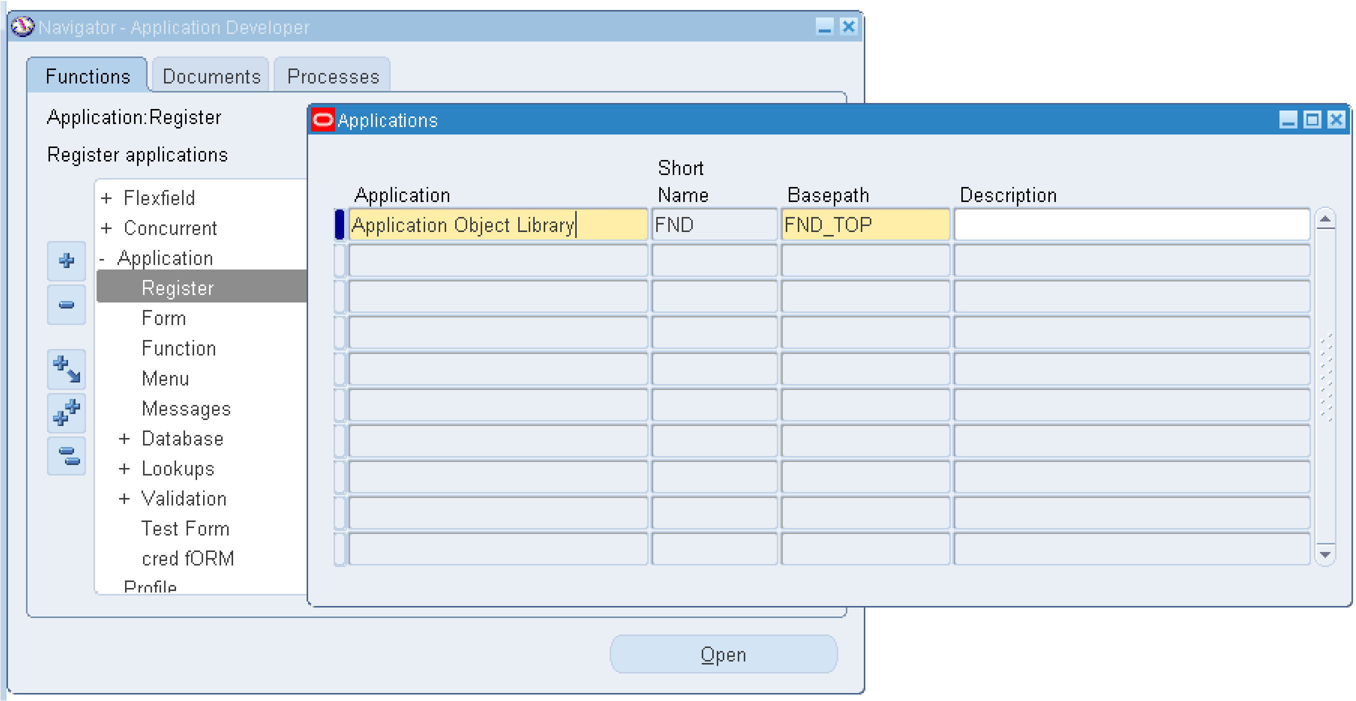
Application	Name	Scope	Type
Application Object Library		Set	Domain

Description:

4. Application registration

Change Responsibility to **‘Application Developer’**

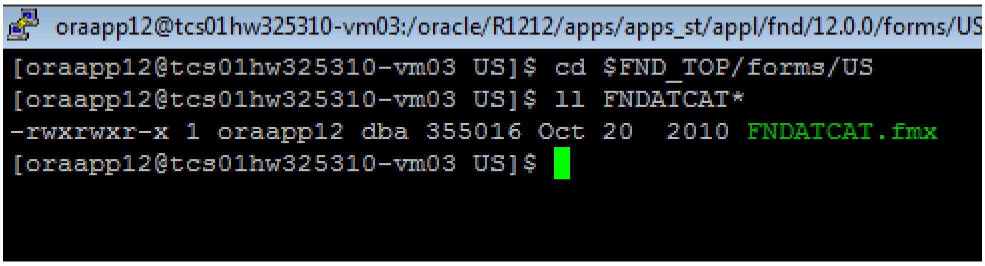
Navigate to Application > Register



Application	Short Name	Basepath	Description
Application Object Library	FND	FND_TOP	

Enter the name of your application and other fields as seen in the above screen. The **Basepath** is the file path on the server, where the executables belonging to this application will reside.

For example, in this case, if you register a form under this application, it will reside under the directory – FND_TOP/forms/US directory on the form server.


oraapp12@tcs01hw325310-vm03:/oracle/R1212/apps/apps_st/appl/fnd/12.0.0/forms/US

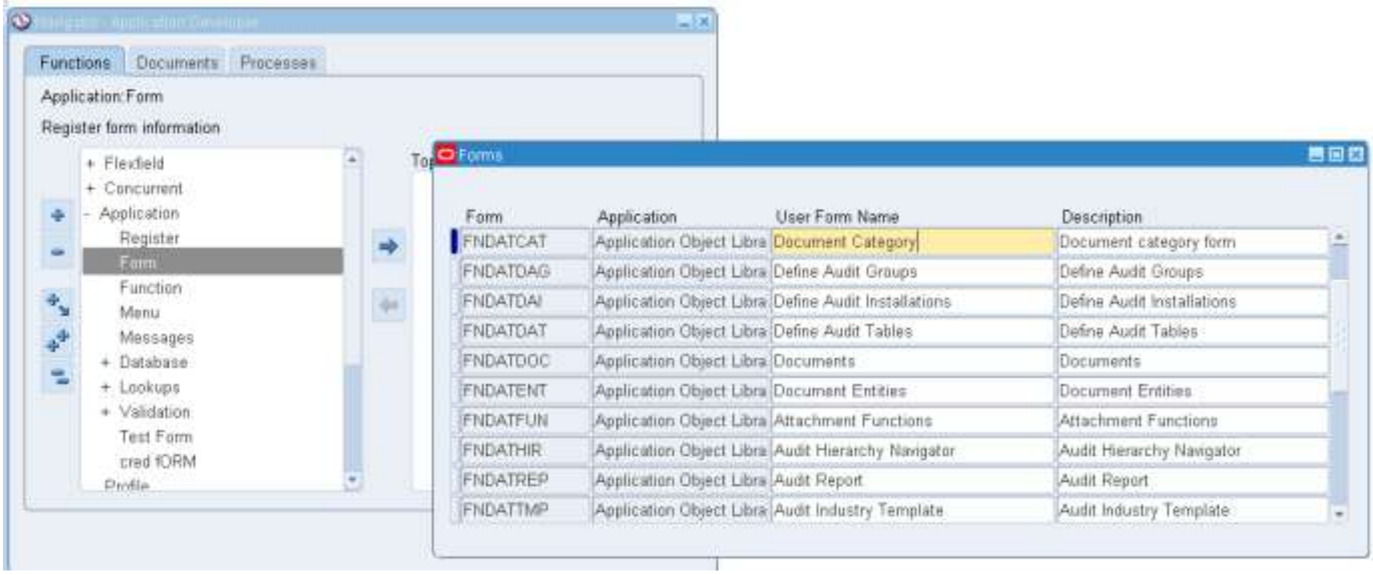
```
[oraapp12@tcs01hw325310-vm03 US]$ cd $FND_TOP/forms/US
[oraapp12@tcs01hw325310-vm03 US]$ ll FNDATCAT*
-rwxrwxr-x 1 oraapp12 dba 355016 Oct 20  2010 FNDATCAT.fmx
[oraapp12@tcs01hw325310-vm03 US]$
```

Note: the above screenshot is showing the application name , which comes be default with oracle applications, so do not try to create any application with this name, chose your own name.

5. Form/Function registration

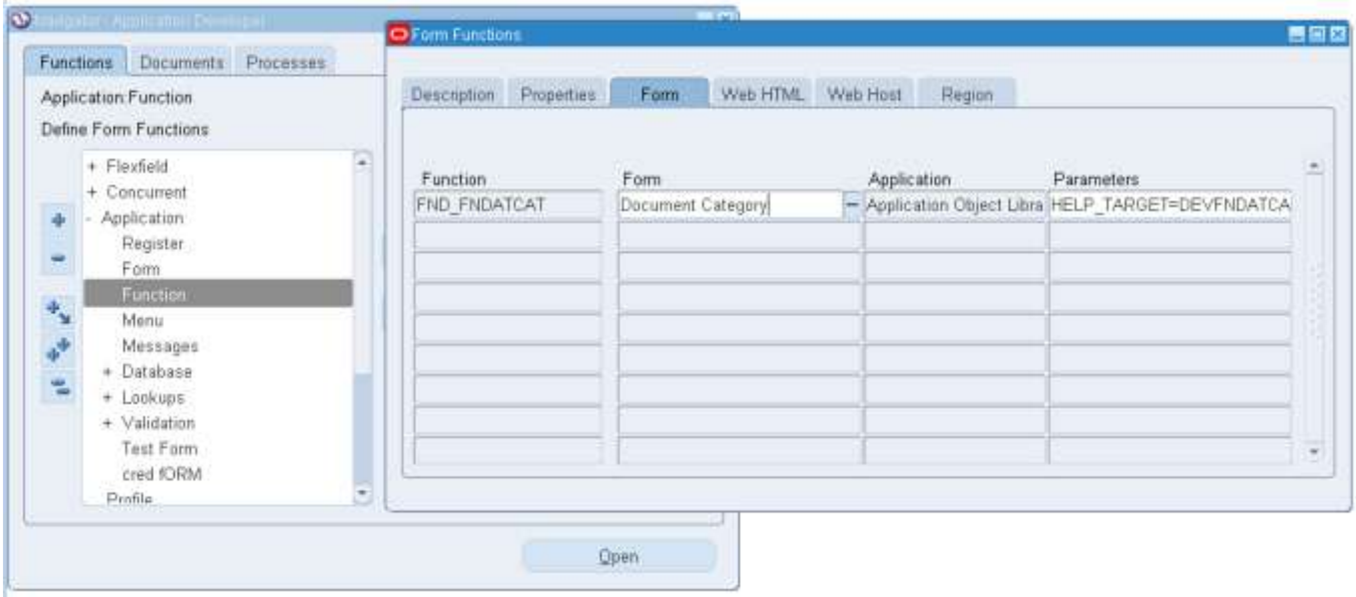
Change Responsibility to **‘Application Developer’**

Navigate to Application > Form



Form	Application	User Form Name	Description
FNDATCAT	Application Object Libra	Document Category	Document category form
FNDATDAG	Application Object Libra	Define Audit Groups	Define Audit Groups
FNDATDAI	Application Object Libra	Define Audit Installations	Define Audit Installations
FNDATDAT	Application Object Libra	Define Audit Tables	Define Audit Tables
FNDATDOC	Application Object Libra	Documents	Documents
FNDATENT	Application Object Libra	Document Entities	Document Entities
FNDATFUN	Application Object Libra	Attachment Functions	Attachment Functions
FNDATHIR	Application Object Libra	Audit Hierarchy Navigator	Audit Hierarchy Navigator
FNDATREP	Application Object Libra	Audit Report	Audit Report
FNDATTMP	Application Object Libra	Audit Industry Template	Audit Industry Template

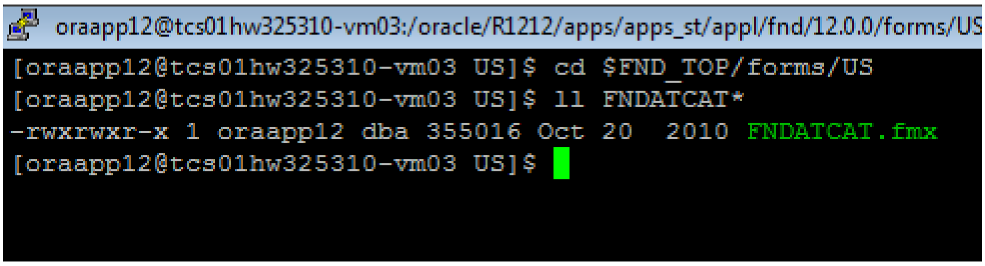
Enter the fields as shown above and save. We will take the example of form ‘**FNDATCAT’** and User Form Name= ‘**Document Category’** here.



Function	Form	Application	Parameters
FND_FNDATCAT	Document Category	Application Object Libra	HELP_TARGET=DEVFNDATCA

The ‘**User Form Name’** in the first screen should match **‘Form’** field in the second screen.

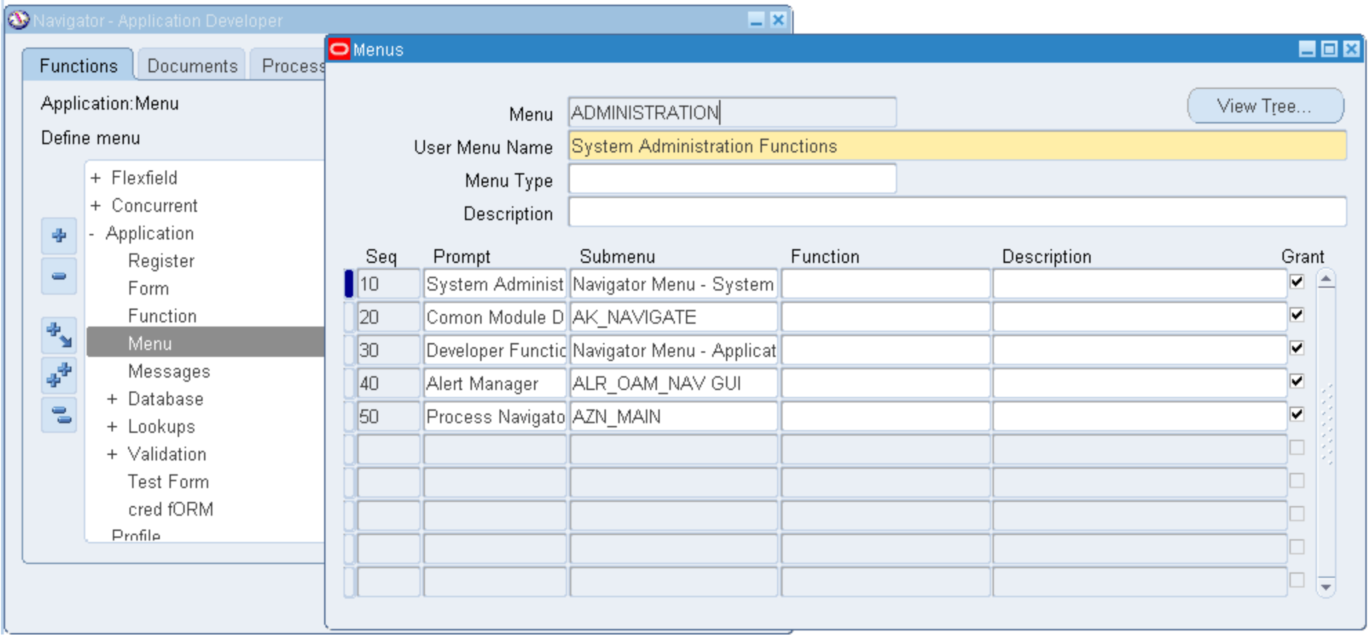
Since the Basepath of this application is FND_TOP, the form executable **FNDATCAT.fmx** needs to be in the directory as shown in the screenshot of the form server below:



```
oraapp12@tcs01hw325310-vm03:/oracle/R1212/apps/apps_st/appl/fnd/12.0.0/forms/US
[oraapp12@tcs01hw325310-vm03 US]$ cd $FND_TOP/forms/US
[oraapp12@tcs01hw325310-vm03 US]$ ll FNDATCAT*
-rwxrwxr-x 1 oraapp12 dba 355016 Oct 20  2010 FNDATCAT.fmx
[oraapp12@tcs01hw325310-vm03 US]$
```

6. Creating Custom Menu

Change Responsibility to **'Application Developer'**
Navigate to Application > Menu



Menu: ADMINISTRATION
User Menu Name: System Administration Functions
Menu Type:
Description:

Seq	Prompt	Submenu	Function	Description	Grant
10	System Administ	Navigator Menu - System			☑
20	Comon Module D	AK_NAVIGATE			☑
30	Developer Functic	Navigator Menu - Applicat			☑
40	Alert Manager	ALR_OAM_NAV GUI			☑
50	Process Navigato	AZN_MAIN			☑

Step1: Give a name and **User Menu Name** to the menu, the **User Menu Name** field will be used link the menu to the Responsibility.

Step2: Enter the menu items, and the runtime prompt. You can wither assign a function or a submenu to a menu item. Look at the function definition in earlier section.

Step3: Compile the menu – this is done automatically when you save the menu.

7. Alerts

Introduction:

Oracle Alert is a utility to create different kind of alert events, which can trigger, depending on certain condition.

This is a powerful tool to get alert messages, when something goes wrong in the system. For Example, we want the system to send a mail to the Stores manager, when a particular item in demand runs low in the inventory, say goes to 20% of the required stock level.

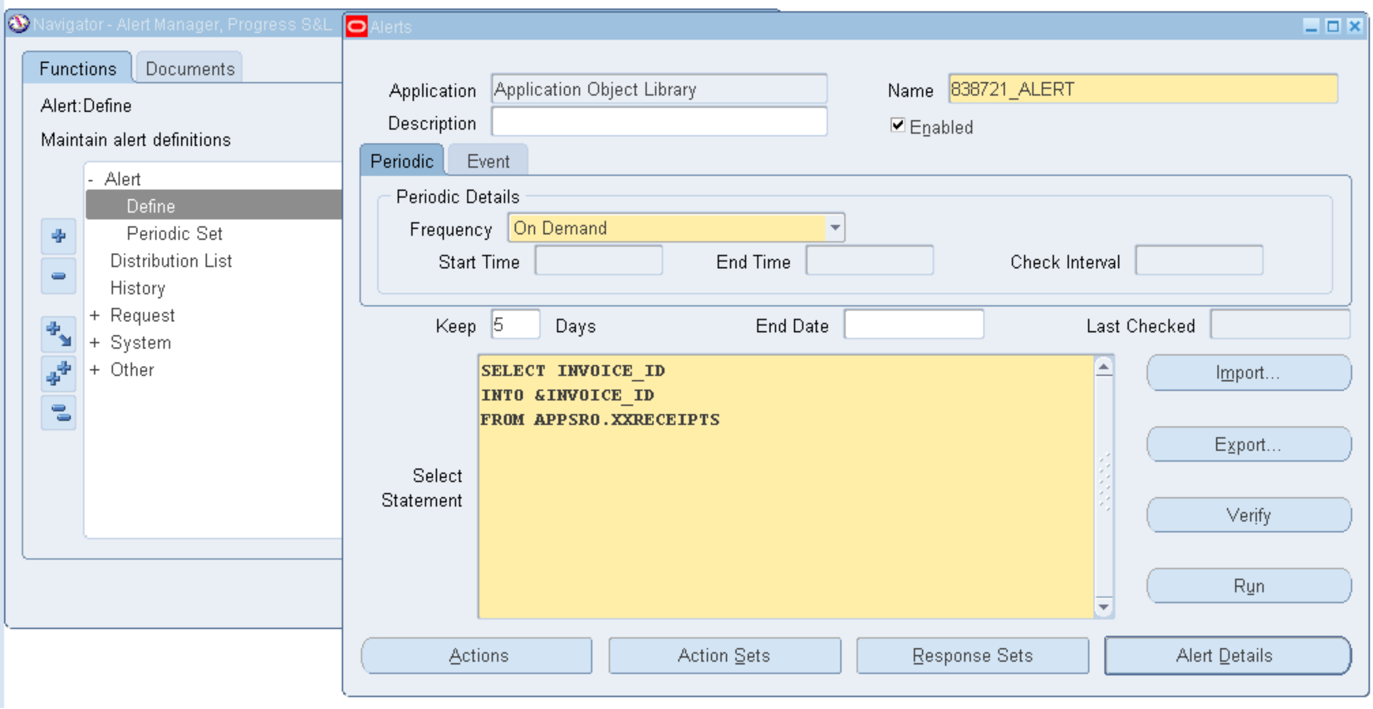
Since all the data in an ERP system is stored in oracle database, we can use this tool to generate a variety of alerts which will keep informing us well in time, before things really get wrong.

Another example of setting up alerts will be for the Database Administrators, which will alert them when space in the database is running low.

Creating an Alert:

Change Responsibility to **‘Alert Manager**

Navigate to Alert > Define



Application: Application Object Library

Description:

Name: 838721_ALERT

☑ Enabled

Frequency: On Demand

Start Time: End Time: Check Interval:

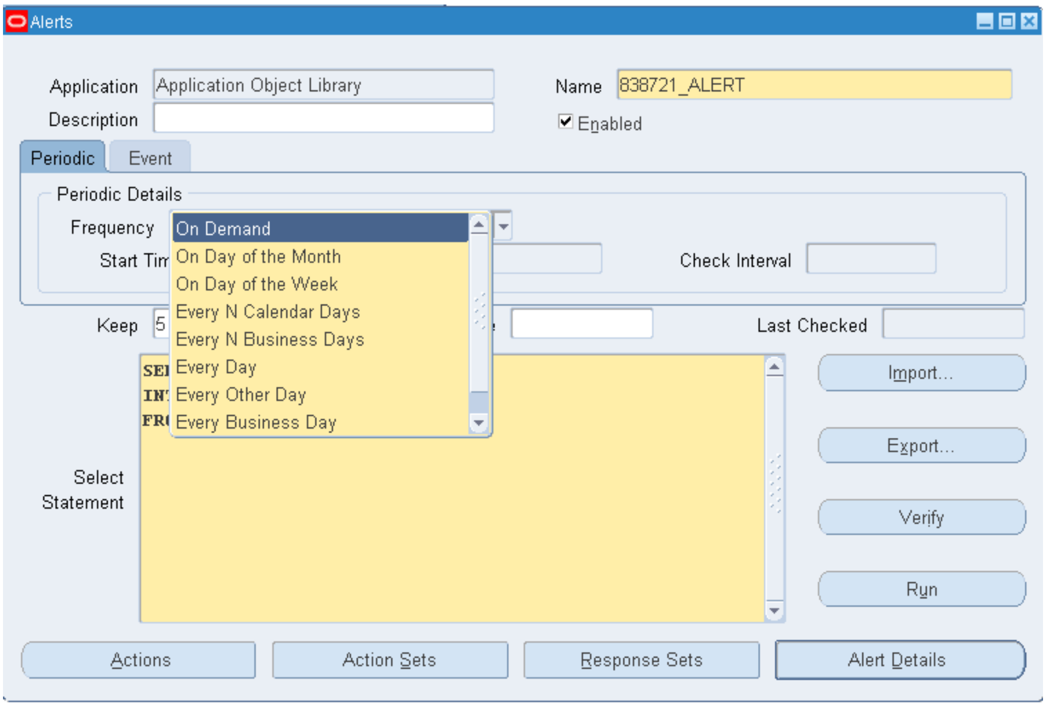
Keep 5 Days End Date: Last Checked:

Select Statement:

```
SELECT INVOICE_ID
INTO &INVOICE_ID
FROM APPSRO.XXRECEIPTS
```

An alert can be **Periodic** or **Event** type

A periodic alert is fired at a particular frequency, the frequency can be one these shown in the diagram below:



Alerts

Application: Application Object Library

Name: 838721_ALERT

Description:

☑ Enabled

Periodic | Event

Periodic Details

Frequency: On Demand

- On Demand
- On Day of the Month
- On Day of the Week
- Every N Calendar Days
- Every N Business Days
- Every Day
- Every Other Day
- Every Business Day

Start Tim

Check Interval

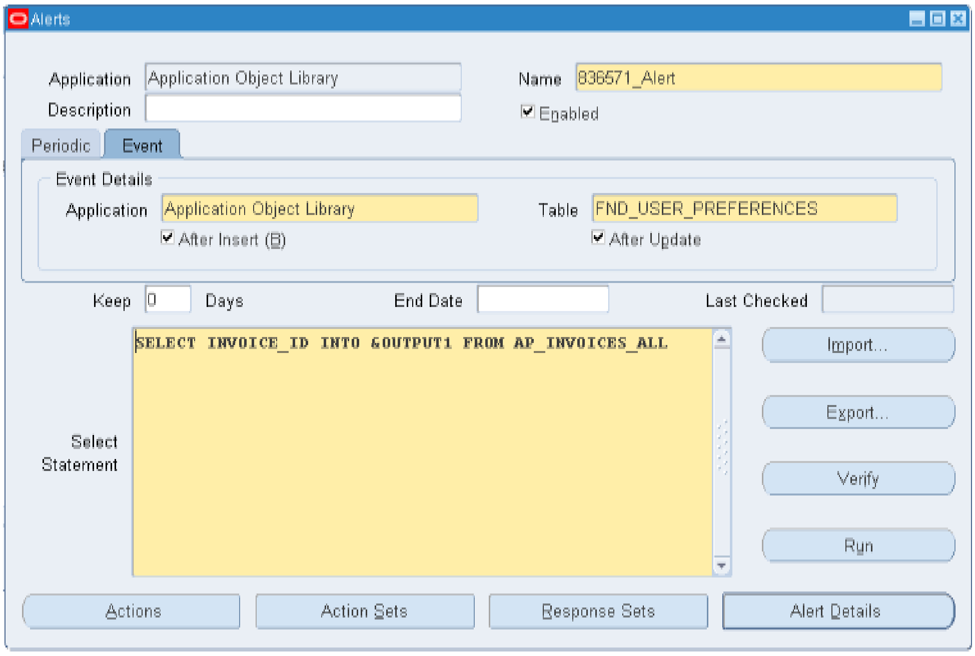
Keep 5

Last Checked

Select Statement

```
SEI
INT
FRO
```

An event Alert will fire if an insert or update is made on a table, in the below example, we have used the table FND_USER_PREFERENCES.



Application: Application Object Library
Name: 836571_Alert
Description:
☑ Enabled

Event Details
Application: Application Object Library
Table: FND_USER_PREFERENCES
☑ After Insert (B)
☑ After Update

Keep: 0 Days
End Date:
Last Checked:

Select Statement:

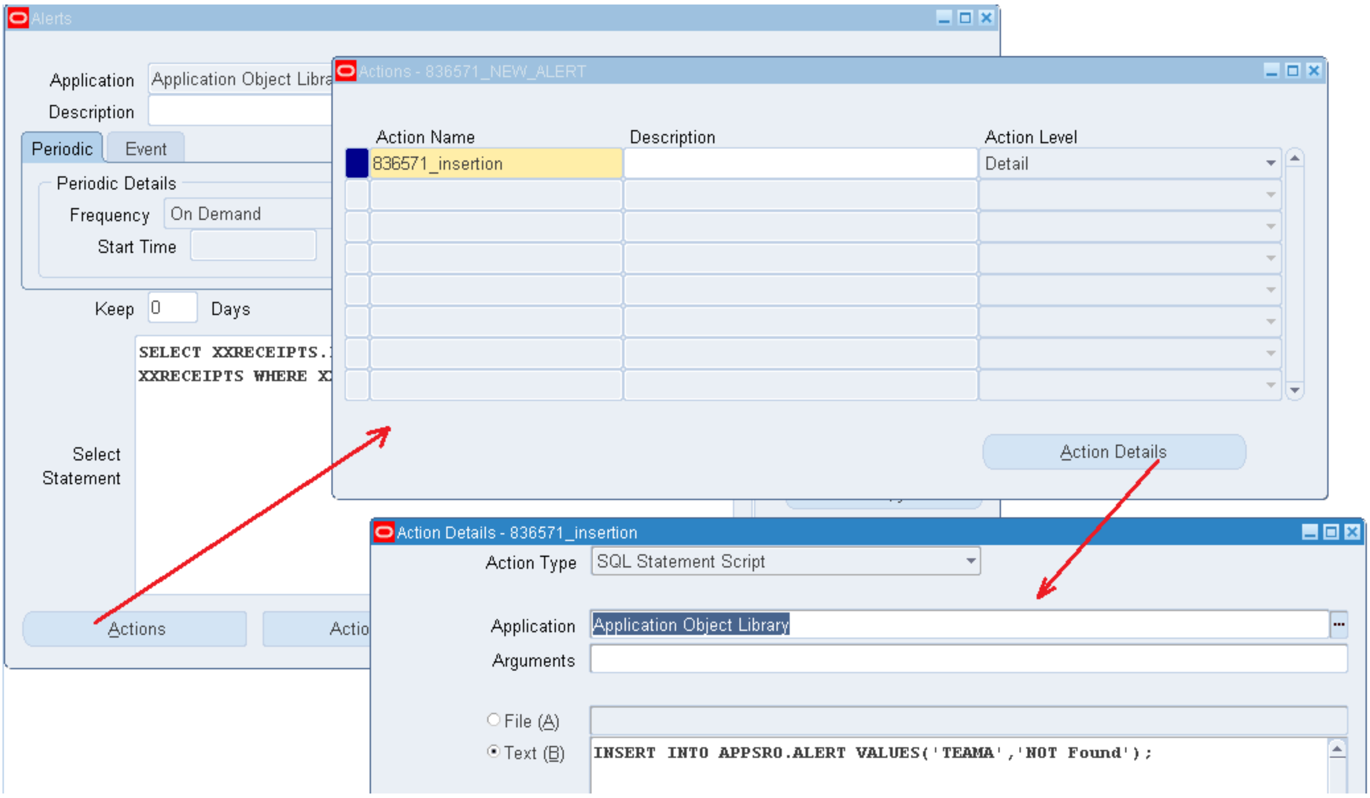
```
SELECT INVOICE_ID INTO &OUTPUT1 FROM AP_INVOICES_ALL
```

When this alert is fired, the sql statement mentioned in the '**Select Statement**' section gets executed.

What happens when an alert fires:

When an alert fires, the Actions or Action Sets defined with the alert gets executed. This is done by clicking the action button and then clicking the Action Details Button

The below example is a Periodic alert, inserts a row into table ALERT_VALUES when the alert fires.



Application: Application Object Libra

Description:

Periodic Details

Frequency: On Demand

Start Time:

Keep: 0 Days

Select Statement:

```
SELECT XXRECEIPTS.I
XXRECEIPTS WHERE X
```

Action Name	Description	Action Level
836571_insertion		Detail

Action Type: SQL Statement Script

Application: Application Object Library

Arguments:

○ File (A)

● Text (B)

```
INSERT INTO APPSRO.ALERT VALUES('TEAMA','NOT Found');
```

Example 1: Creating an Alert

Problem Description: Write a periodic alert with frequency = ‘On Demand’, which will check if the table has any data for invoice_id=840027. If it finds any such row, it will insert a row into the table ALERT, quoting the number of such invoice found. The detail of these tables are given below:

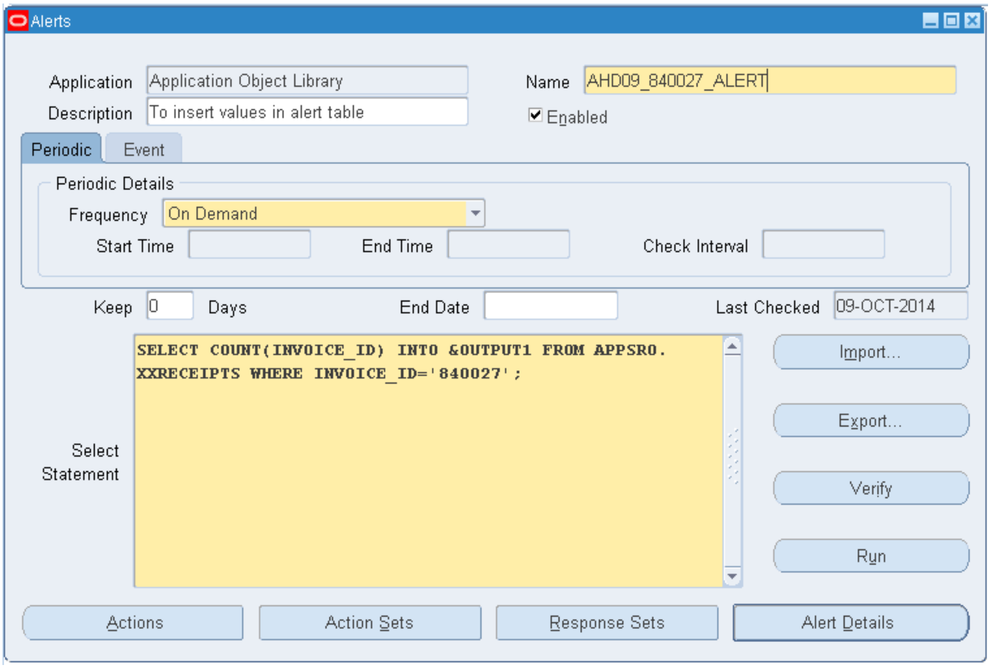
```
SQL> desc appsro.xxreceipts

Name                                      Null?    Type
 ----------------------------------------- -------- ---------------------
 INVOICE_ID                                         VARCHAR2(10)
 VENDOR_ID                                          CHAR(2)
 RECEIVED_ITEM_QTY                                  NUMBER
 RECEIVED_ITEM_NAME                                 VARCHAR2(30)


SQL> desc appsro.alert
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 TEAM_NAME                                          VARCHAR2(10)
 ALERT_DETAIL                                       VARCHAR2(100)
```

Solution 1:

Step1: Create the alert – Alert > Define:



Application: Application Object Library

Description: To insert values in alert table

Name: AHD09_840027_ALERT

☑ Enabled

Periodic Details

Frequency: On Demand

Start Time:

End Time:

Check Interval:

Keep: 0 Days

End Date:

Last Checked: 09-OCT-2014

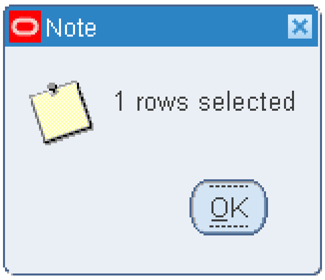
Select Statement:

```
SELECT COUNT(INVOICE_ID) INTO &OUTPUT1 FROM APPSRO.
XXRECEIPTS WHERE INVOICE_ID='840027';
```

```
SELECT COUNT(INVOICE_ID) INTO &OUTPUT1 FROM APPSRO.XXRECEIPTS WHERE
INVOICE_ID='840027';
```

Note that we are storing the result of the query into a variable named OUTPUT1. This variable can be used later on in the alert.

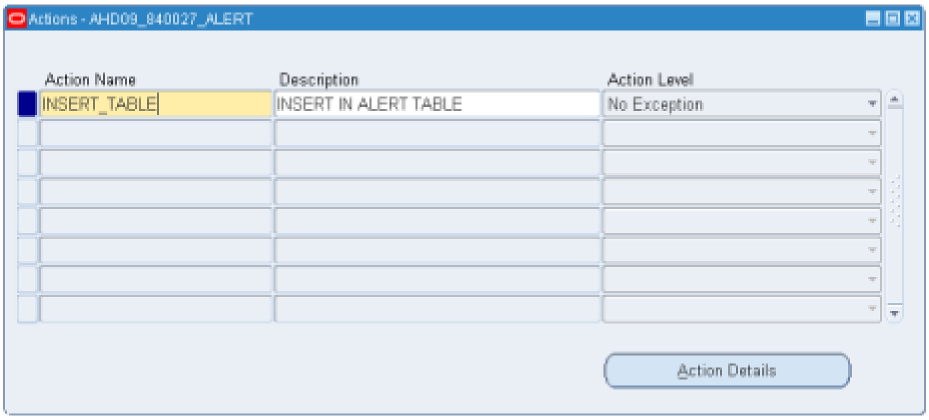
Step2: Hit the run button



1 rows selected

It will run the sql and tell you, if it finds any data.

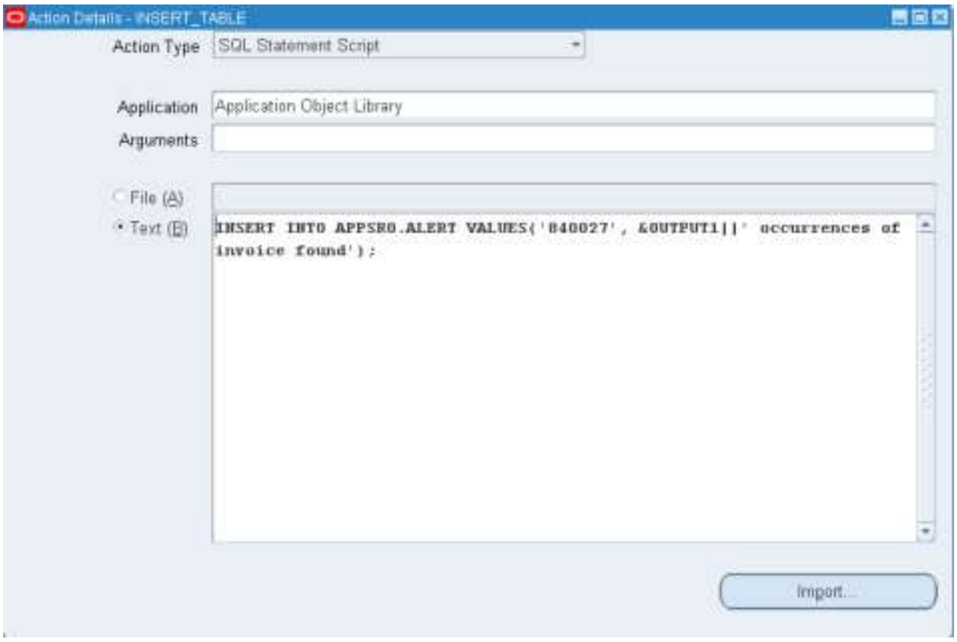
Step2: Hit the Actions button



Action Name	Description	Action Level
INSERT_TABLE	INSERT IN ALERT TABLE	No Exception

Enter the details as above.

Step2: Hit the 'Actions Details' button and type as below.



Action Type: SQL Statement Script

Application: Application Object Library

Arguments:

○ File (A)

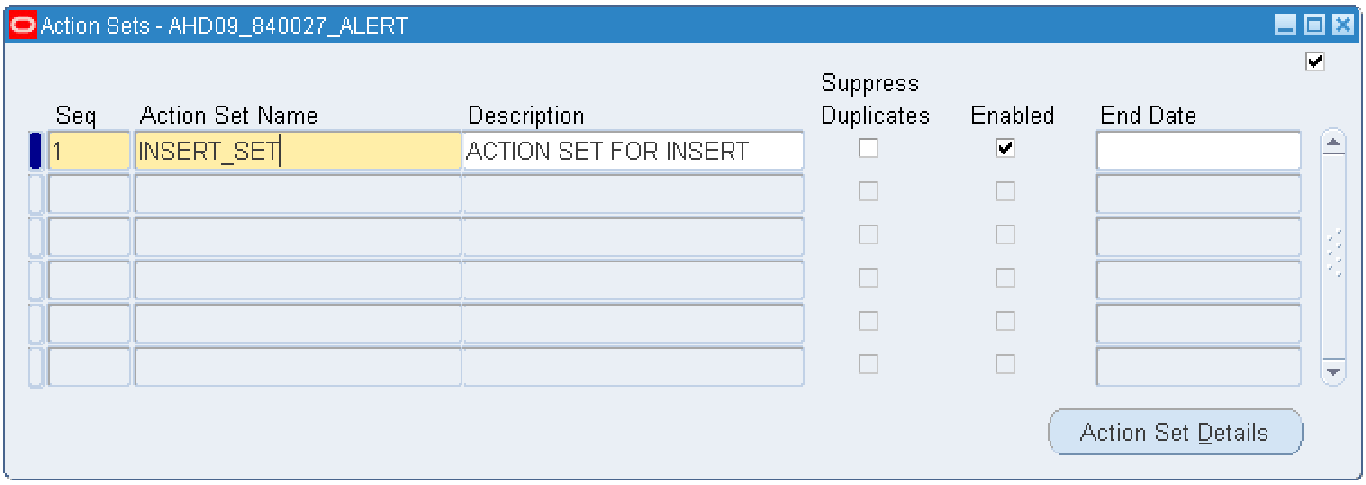
● Text (B)

```
INSERT INTO APPSRO.ALERT VALUES('840027', &OUTPUT1||' occurrences of
invoice found');
```

INSERT INTO APPSRO.ALERT VALUES('840027', &OUTPUT1||' occurrences of invoice found');

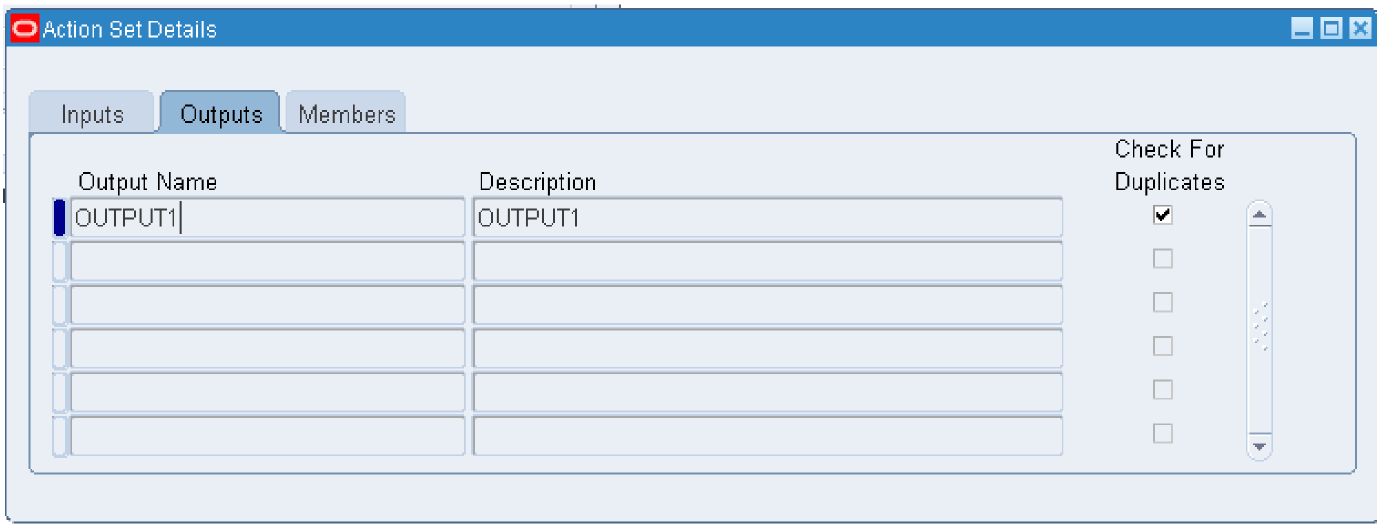
Now you have created the action. But the action will not fire unless it is attached to an action set.

Step2: Hit the 'Actions Set' button, give a name to the action set as below.



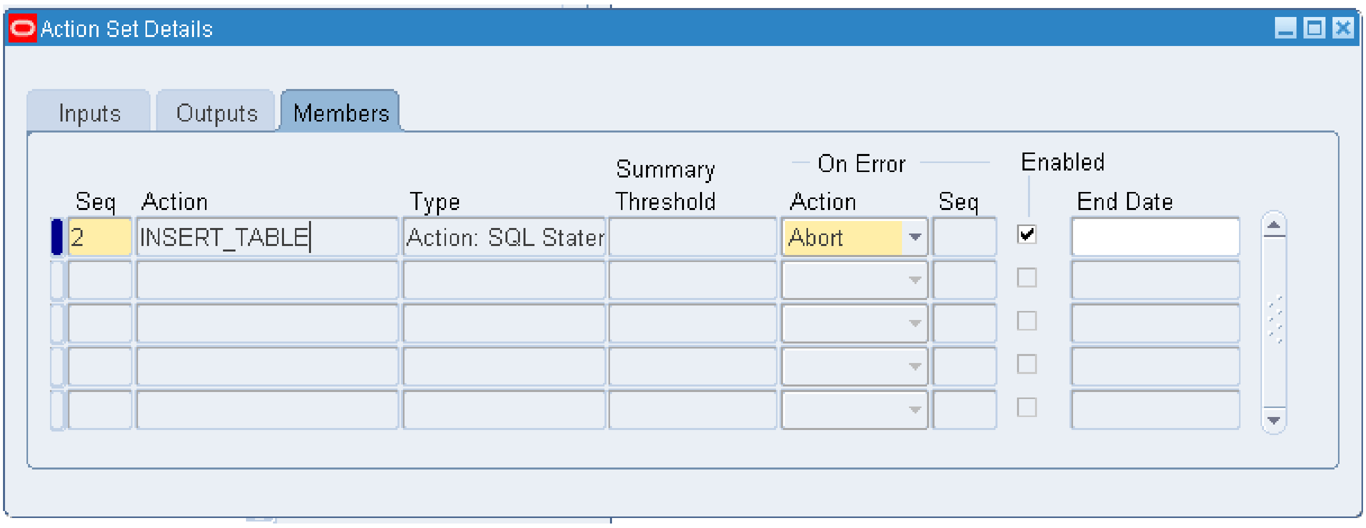
Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
1	INSERT_SET	ACTION SET FOR INSERT	□	☑	

Step2: Hit the 'Actions Set Details' button, and click on the output tab, you can see the OUTPUT1 variable, which we have create in the main select statement.



Output Name	Description	Check For Duplicates
OUTPUT1	OUTPUT1	☑

Now click on the Members Tab, here you can see all the actions you have defined earlier. In this case , we will see only one action called INSERT_TABLE. Select that action.

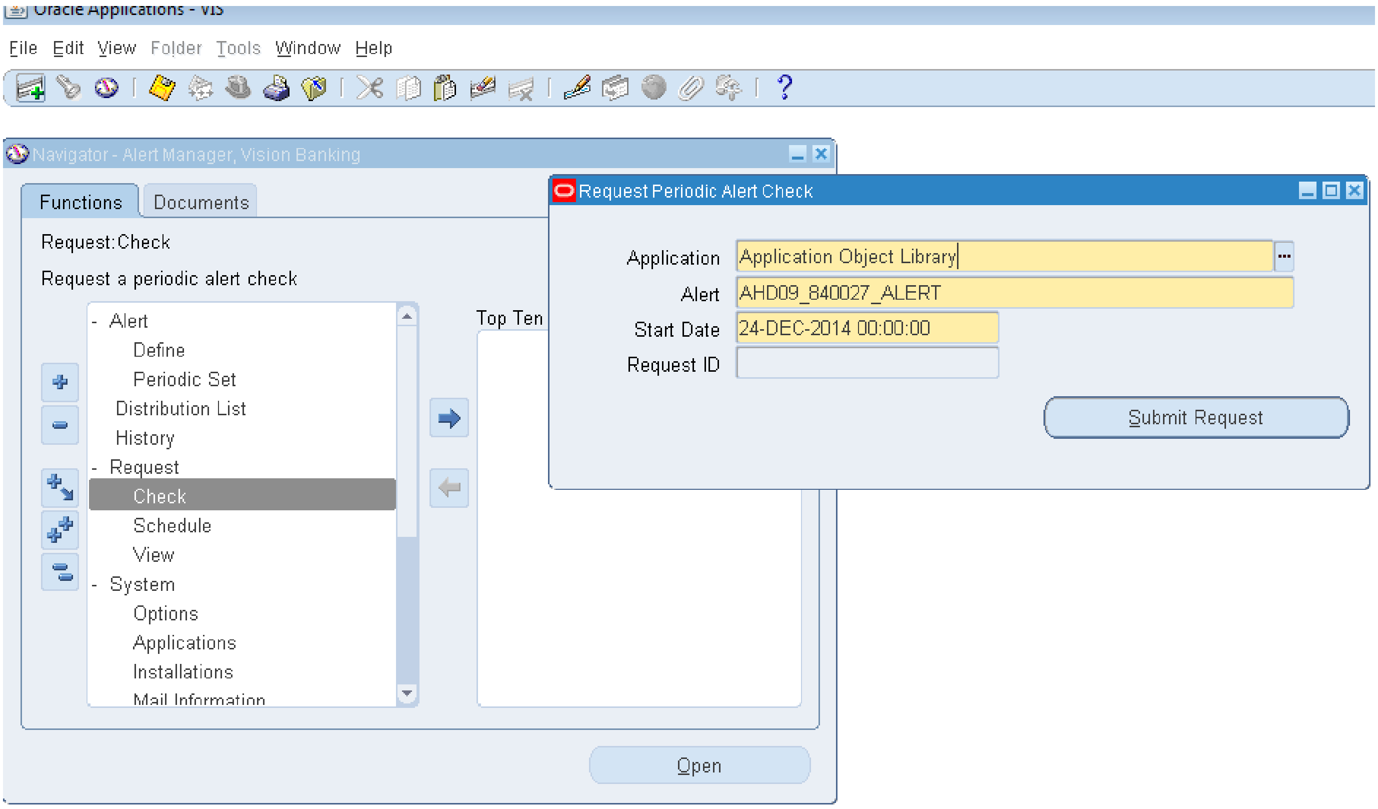


Seq	Action	Type	Summary Threshold	On Error Action	On Error Seq	Enabled	End Date
2	INSERT_TABLE	Action: SQL Stater		Abort		☑	
						□	
						□	
						□	
						□	

Step2: Run the alert:

Navigation – Request > Check

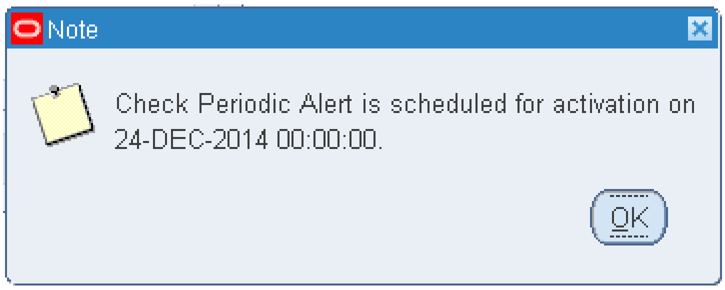
Select the application and Alert Name and hit 'Submit Request'



Request:Check

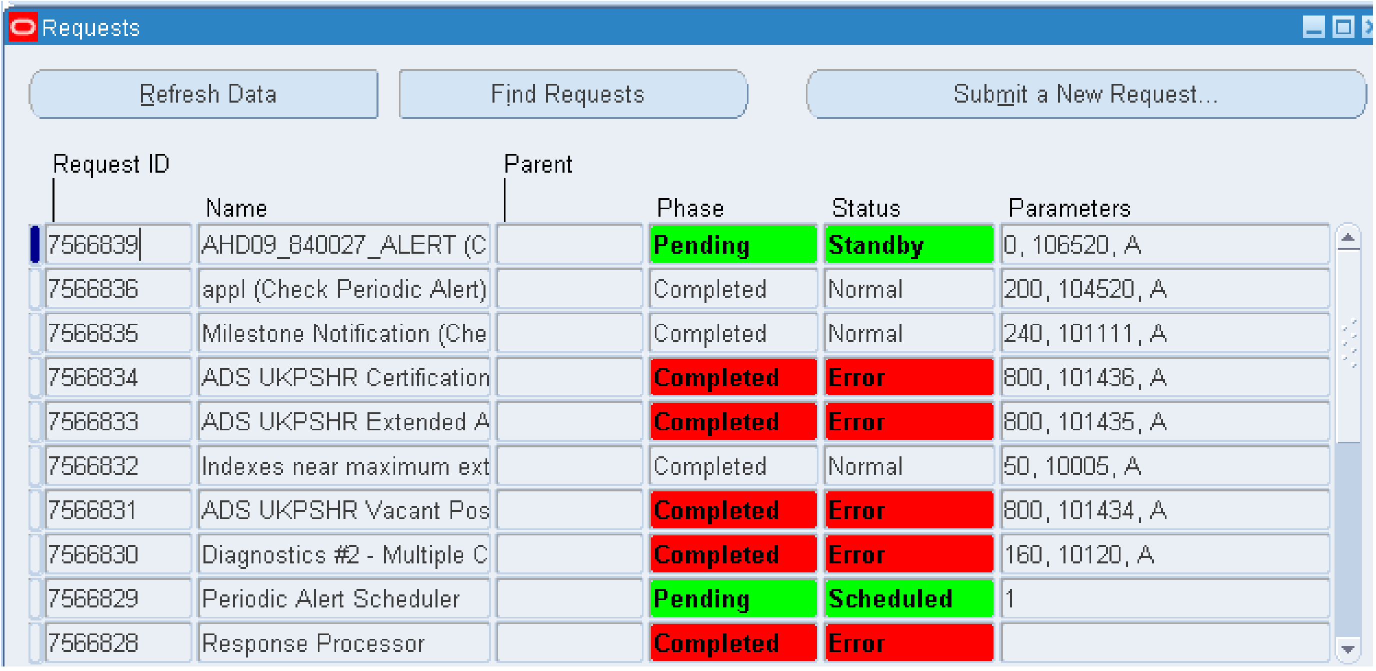
Request a periodic alert check

Application	Application Object Library
Alert	AHD09_840027_ALERT
Start Date	24-DEC-2014 00:00:00
Request ID	



Check Periodic Alert is scheduled for activation on 24-DEC-2014 00:00:00.

Sine this is an ‘On Demand’ type, a concurrent program will fire immediately, in this case, request ID – 7556839.



Request ID	Name	Parent	Phase	Status	Parameters
7566839	AHD09_840027_ALERT (C		Pending	Standby	0, 106520, A
7566836	appl (Check Periodic Alert)		Completed	Normal	200, 104520, A
7566835	Milestone Notification (Che		Completed	Normal	240, 101111, A
7566834	ADS UKPSHR Certification		Completed	Error	800, 101436, A
7566833	ADS UKPSHR Extended A		Completed	Error	800, 101435, A
7566832	Indexes near maximum ext		Completed	Normal	50, 10005, A
7566831	ADS UKPSHR Vacant Pos		Completed	Error	800, 101434, A
7566830	Diagnostics #2 - Multiple C		Completed	Error	160, 10120, A
7566829	Periodic Alert Scheduler		Pending	Scheduled	1
7566828	Response Processor		Completed	Error	

Upon completion of the request, you can see the output file , in this case it is as below:

```
View
**************************************************************************
Performing alert "AHD09_840027_ALERT"
Connecting to "APPS"
  Performing action set "INSERT_SET"
Oracle Alert did not perform the No exception action "INSERT_TABLE" because there
were exceptions returned for this action set.
```

TASK1: Find out why the program has failed.

8. Alert Concurrent Programs

Concurrent managers are components of the Oracle Alert concurrent processing facility that monitor and run time-consuming, non-interactive tasks without tying up your terminal. Whenever you submit a concurrent request in Oracle Alert, such as checking an on-demand alert, a concurrent manager processes that request in the background, letting you perform an unlimited number of tasks simultaneously.

The concurrent manager processes the following concurrent programs for Oracle Alert:

- **Periodic Alert Scheduler** - Resides in the concurrent queue and runs every 24 hours at 12 AM, when it submits requests for all periodic alerts that are scheduled to run during the next 24 hours
- **Check Periodic Alert** - Submits a request to the concurrent manager when you check an on demand alert
- **Check Event Alert** - Submits a request to the concurrent manager when an insert or an update to an event table occurs
- **Alert Action Processor** - Performs response actions when Oracle Alert receives an e-mail response to an alert message
- **No Response Action Processor** - Checks for alert messages whose specified response days have passed without receiving a valid response and performs the no response follow-up actions

If for some reason you need to inactivate or activate the concurrent manager, you can use the Schedule Alert Programs form.

Before you use the Schedule Alert Programs form for the first time, make sure you specify the following in the More Options tabbed region of the Oracle Alert Options form:

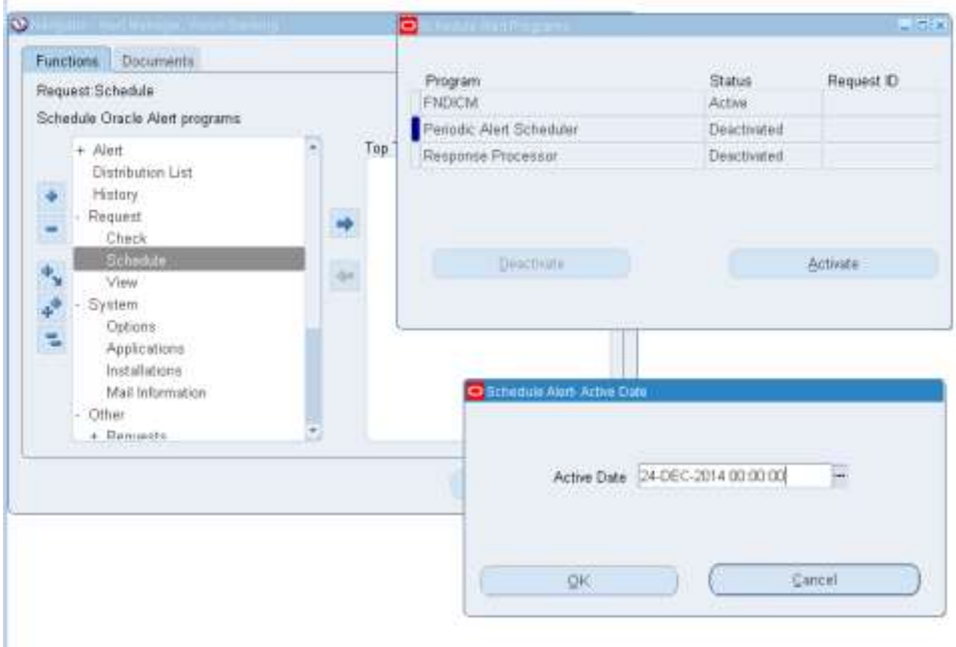
- Concurrent Manager Name (default is STD)
- Concurrent Manager Startup Command (default is STARTMGR)

To activate or deactivate the concurrent manager

1. Navigate to the Schedule Alert Programs form. The Status field shows whether the concurrent manager is active or not.
2. Select Internal Manager.
3. If the status is deactivated, choose Activate to start the concurrent manager immediately.
4. If the status is Active, choose Deactivate to terminate any running requests and deactivate the concurrent manager immediately.

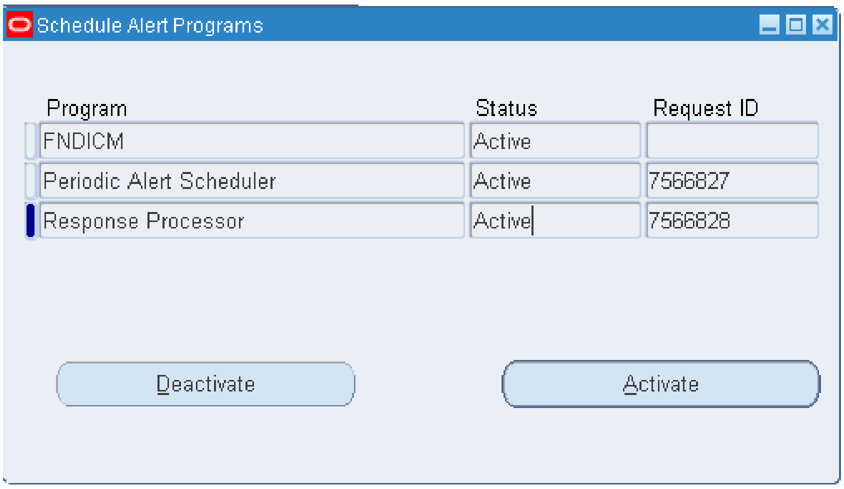
Note: You can activate or deactivate the concurrent manager on the current date you can activate or deactivate the concurrent manager on the current date only.

The below screenshot illustrates how to activate and deactivate these programs:



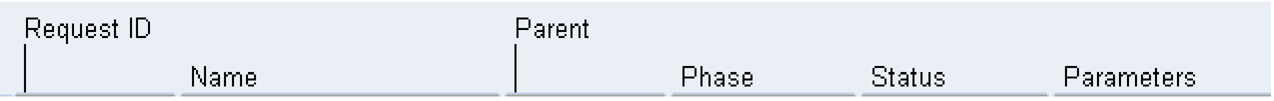
Program	Status	Request ID
FNDICM	Active	
Periodic Alert Scheduler	Deactivated	
Response Processor	Deactivated	

Active Date 24-DEC-2014 00:00:00

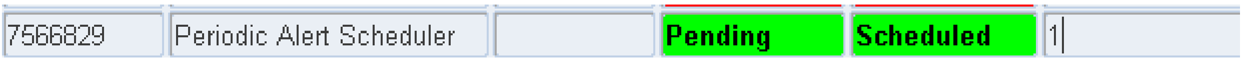


Program	Status	Request ID
FNDICM	Active	
Periodic Alert Scheduler	Active	7566827
Response Processor	Active	7566828

Upon activation, a request ID will be generated. Note that these programs will always be in Pending/Schedules state, meaning they keep on running all the time, waiting for the alert requests.



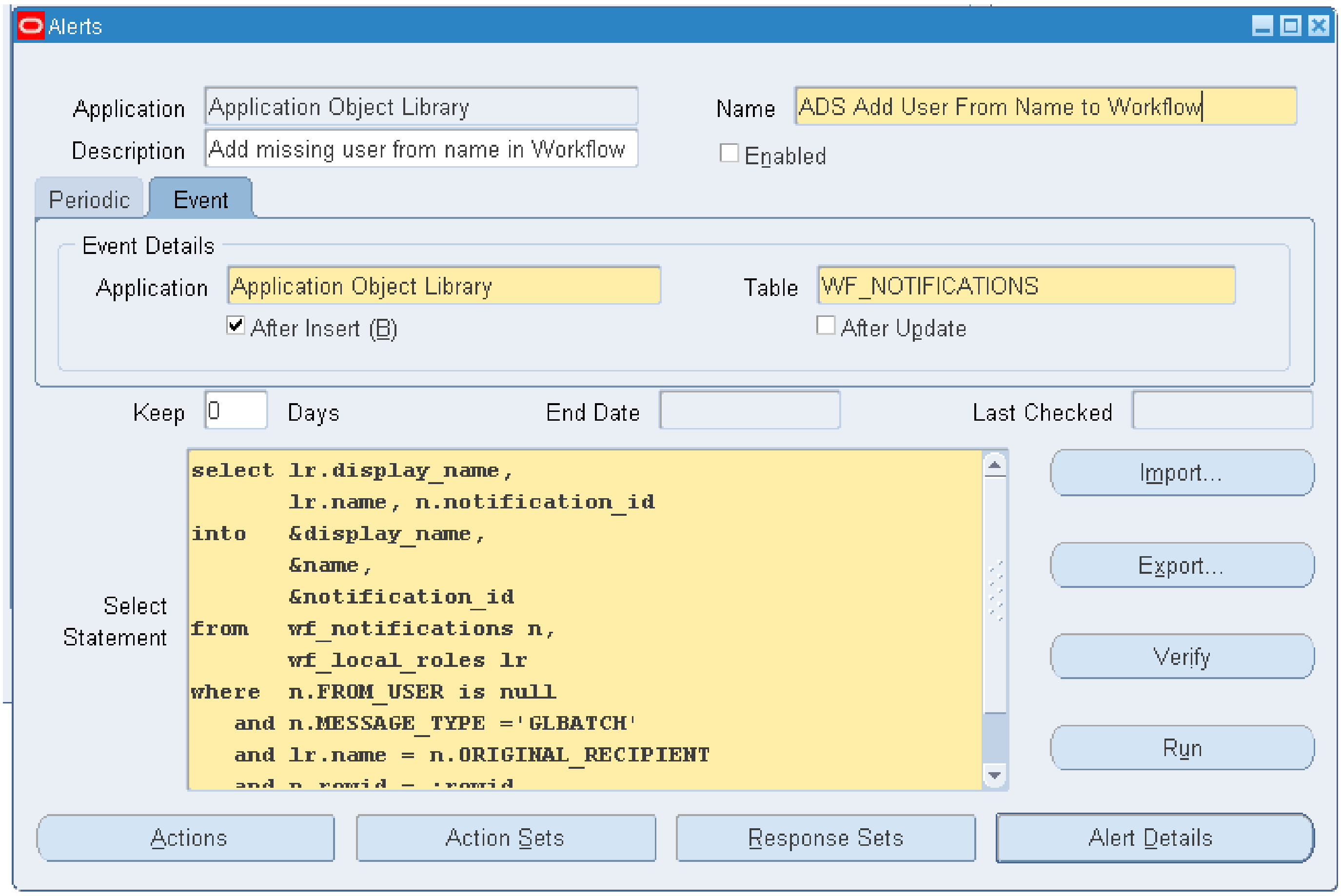
Request ID	Name	Parent	Phase	Status	Parameters



7566829	Periodic Alert Scheduler		Pending	Scheduled	1

Here is an example of an alert program which comes by default with Oracle Applications:

Alert:



Application: Application Object Library

Name: ADS Add User From Name to Workflow

Description: Add missing user from name in Workflow

□ Enabled

Event Details

Application: Application Object Library

Table: WF_NOTIFICATIONS

☑ After Insert (B)

□ After Update

Keep: 0 Days

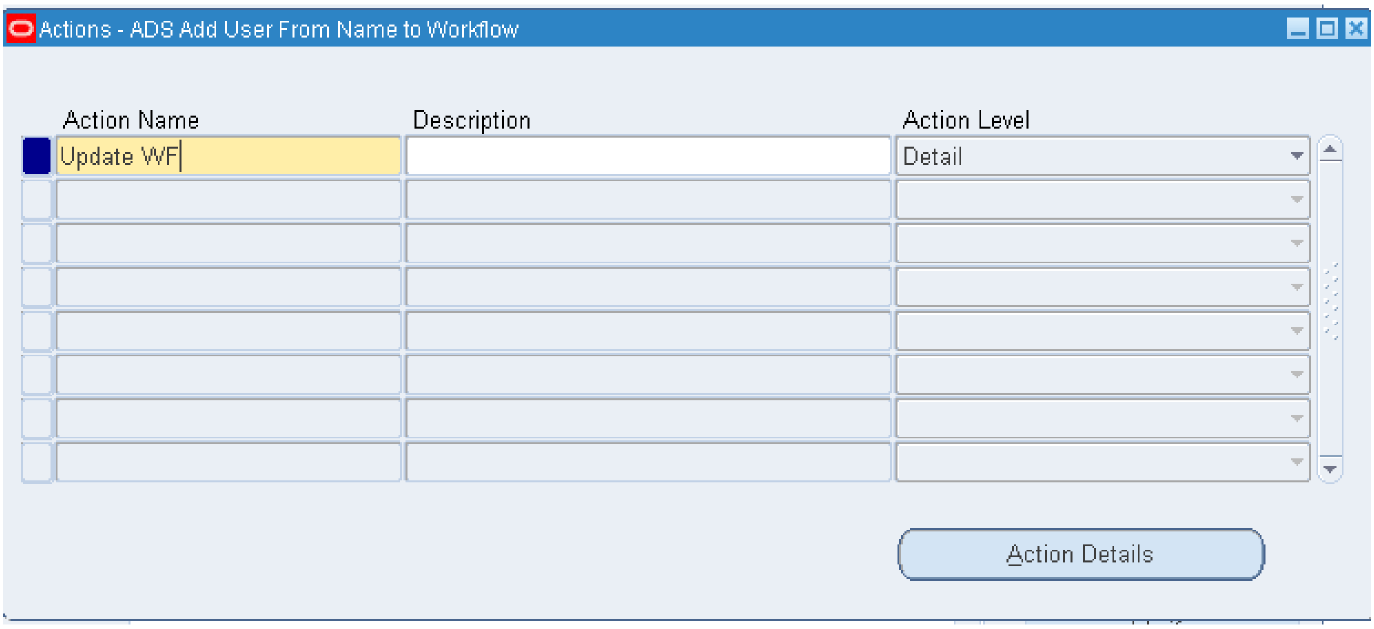
End Date:

Last Checked:

Select Statement:

```
select lr.display_name,
       lr.name, n.notification_id
into   &display_name,
       &name,
       &notification_id
from   wf_notifications n,
       wf_local_roles lr
where  n.FROM_USER is null
   and n.MESSAGE_TYPE ='GLBATCH'
   and lr.name = n.ORIGINAL_RECIPIENT
   and n.rowid = :rowid
```

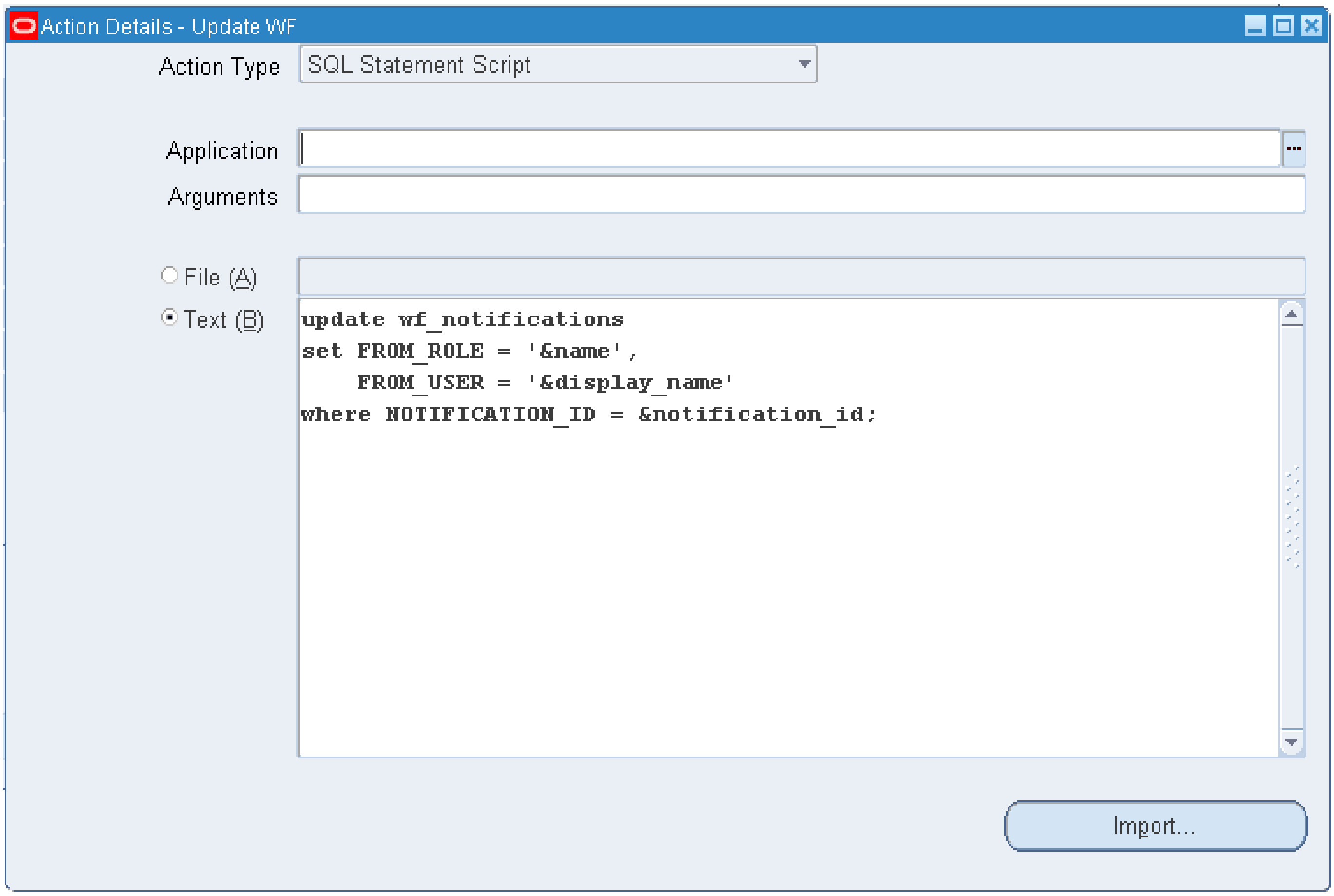
Action:



Actions - ADS Add User From Name to Workflow

Action Name	Description	Action Level
Update WF		Detail

Action Detail:



Action Type: SQL Statement Script

Application:

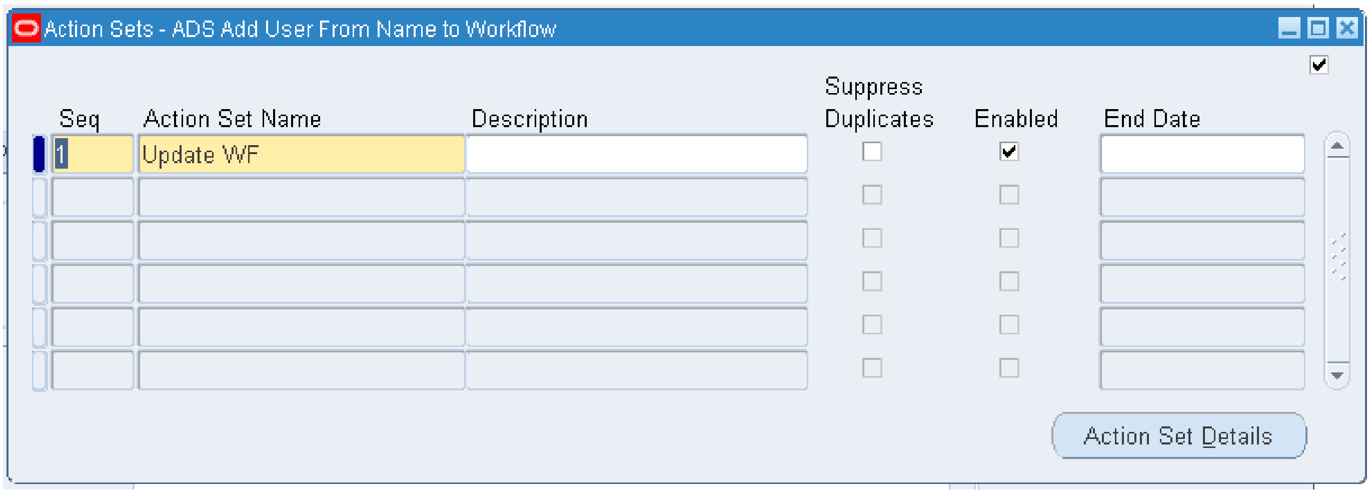
Arguments:

○ File (A)

● Text (B)

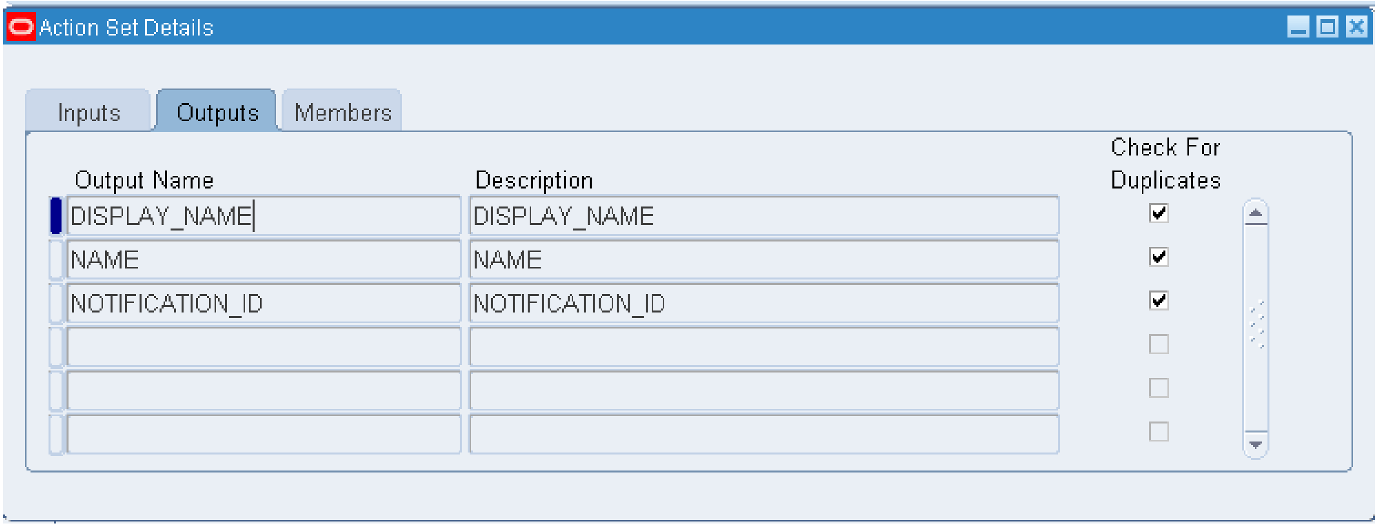
```
update wf_notifications
set FROM_ROLE = '&name',
    FROM_USER = '&display_name'
where NOTIFICATION_ID = &notification_id;
```

Action Set:

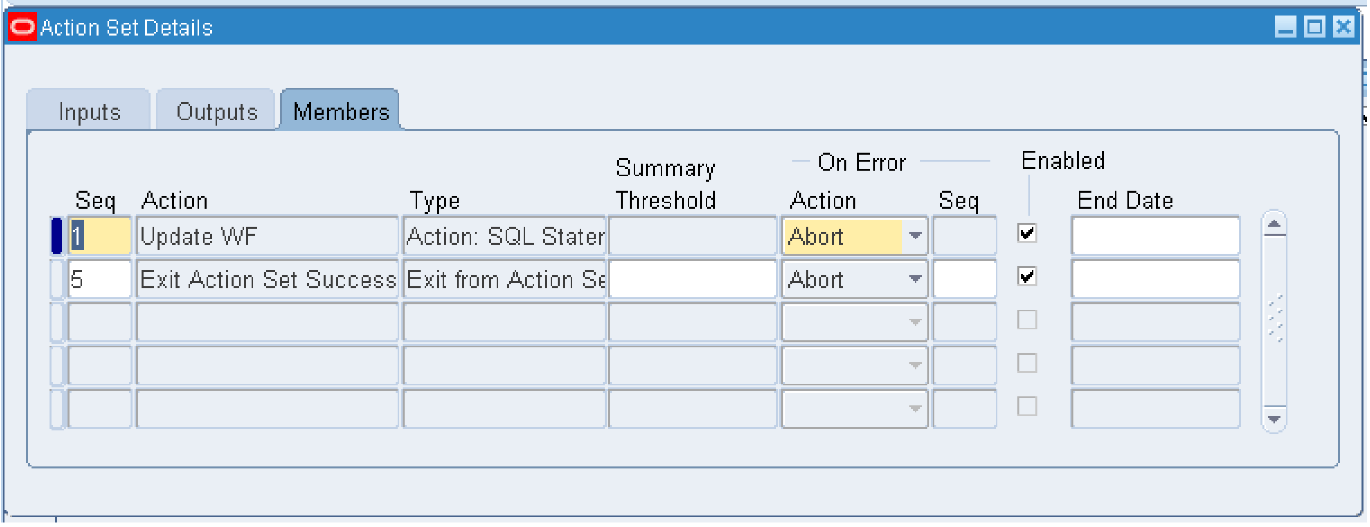


Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
1	Update WF		☐	☑	
			☐	☐	
			☐	☐	
			☐	☐	
			☐	☐	
			☐	☐	

Action Set Details:



Output Name	Description	Check For Duplicates
DISPLAY_NAME	DISPLAY_NAME	☑
NAME	NAME	☑
NOTIFICATION_ID	NOTIFICATION_ID	☑
		□
		□
		□



				On Error		Enabled	
Seq	Action	Type	Summary Threshold	Action	Seq		End Date
1	Update WF	Action: SQL Stater		Abort		☑	
5	Exit Action Set Success	Exit from Action Se		Abort		☑	
						□	
						□	
						□	

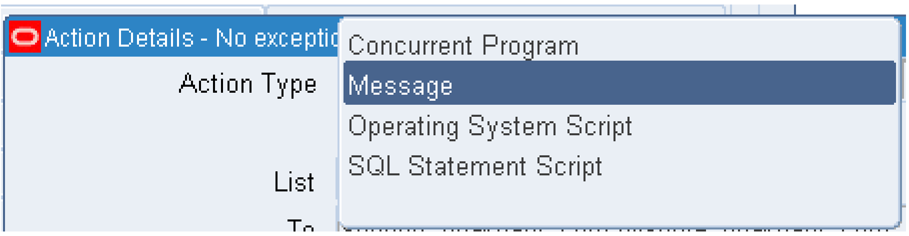
TASK 2: Login to Oracle Applications, review this alert and write down what this alert is trying to achieve.

9. Distribution Set

So far, we have seen that an alert can execute a sql statement as an action, however, it can also perform the following:

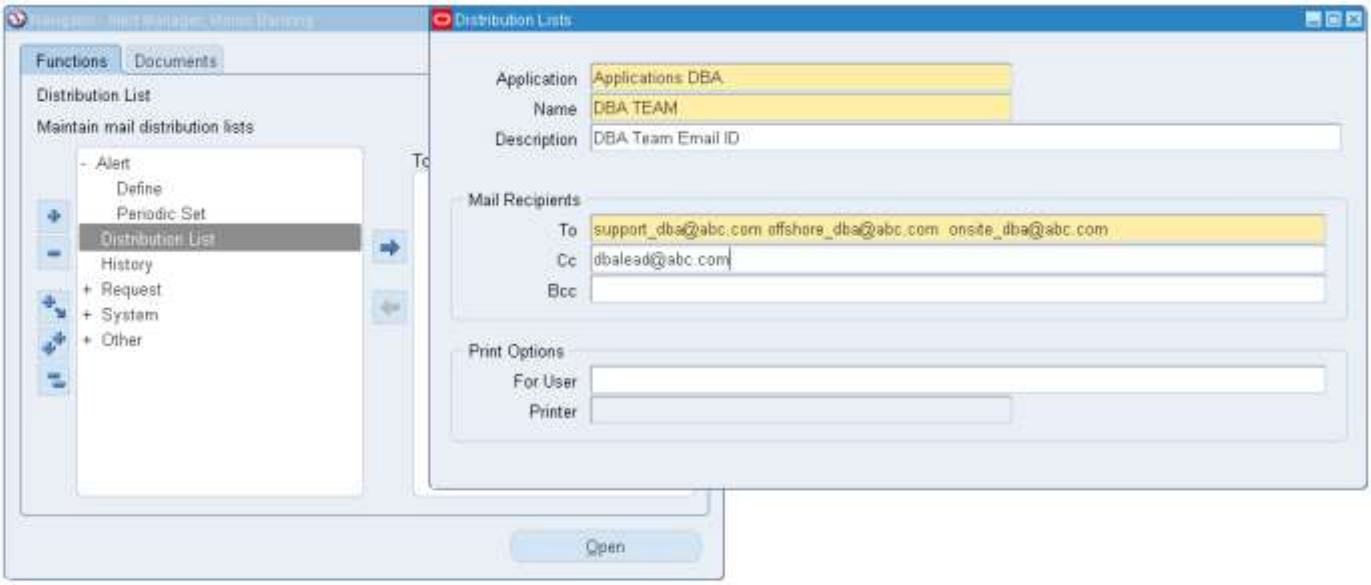
1. Run a Concurrent Program
2. Send an email message
3. Run an Operating System Script

This can be done by selecting the Action Type in the Action windows:



When we define the Action Type as Message, either we can type the email addresses manually, or create a **'Distribution Set'** . A distribution set can contain a set of email addresses.

The below screenshot creates a distributuin set (Navigation: Alert > Distribution Set)

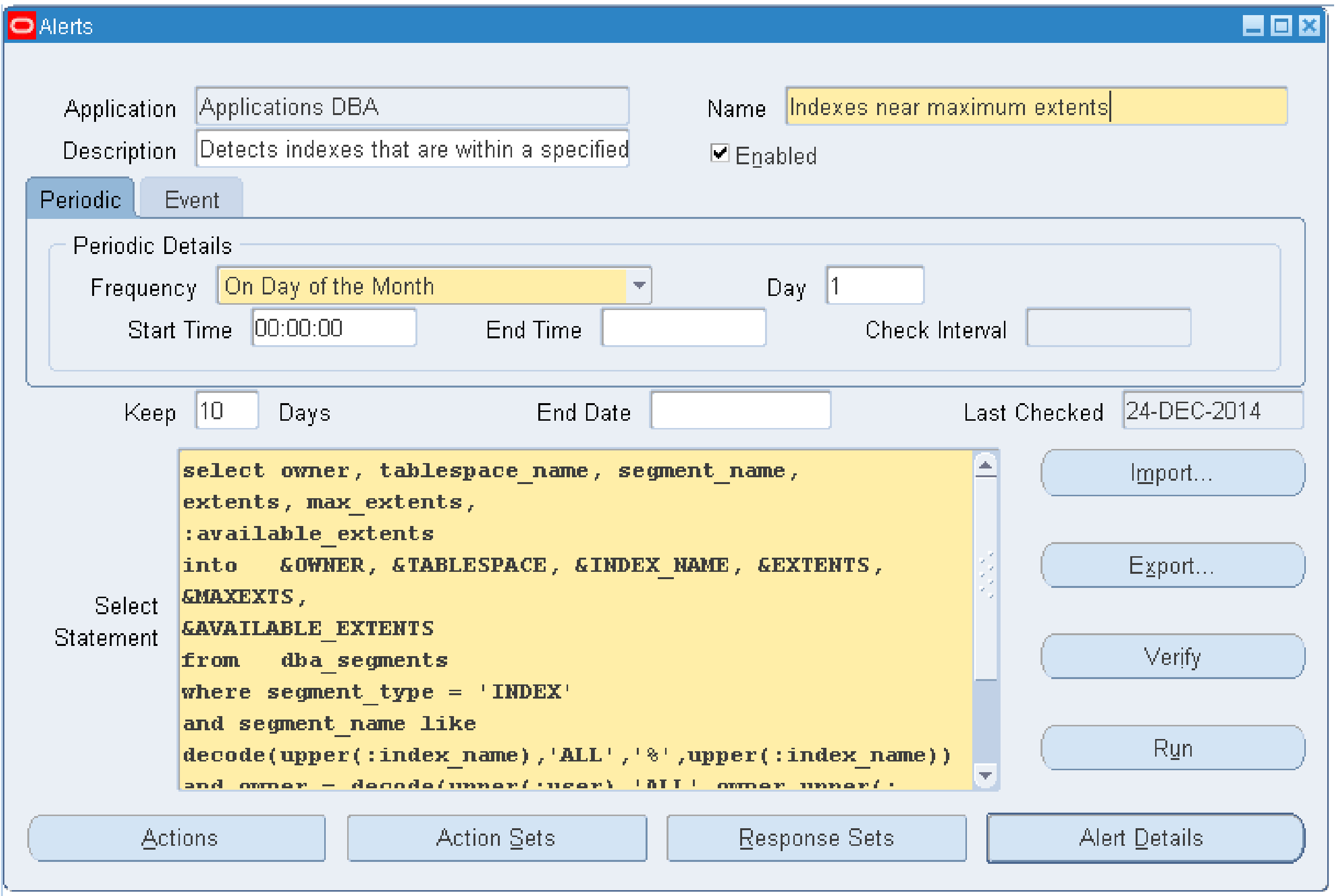


Application: Applications DBA
Name: DBA TEAM
Description: DBA Team Email ID

Mail Recipients
To: support_dba@abc.com offshore_dba@abc.com onsite_dba@abc.com
Cc: dbalead@abc.com
Bcc:

Print Options
For User:
Printer:

This distribution set can be selected while creating alert, shown in the below example.



Application: Applications DBA
Name: Indexes near maximum extents
Description: Detects indexes that are within a specified
☑ Enabled

Periodic Details
Frequency: On Day of the Month
Day: 1
Start Time: 00:00:00
End Time:
Check Interval:

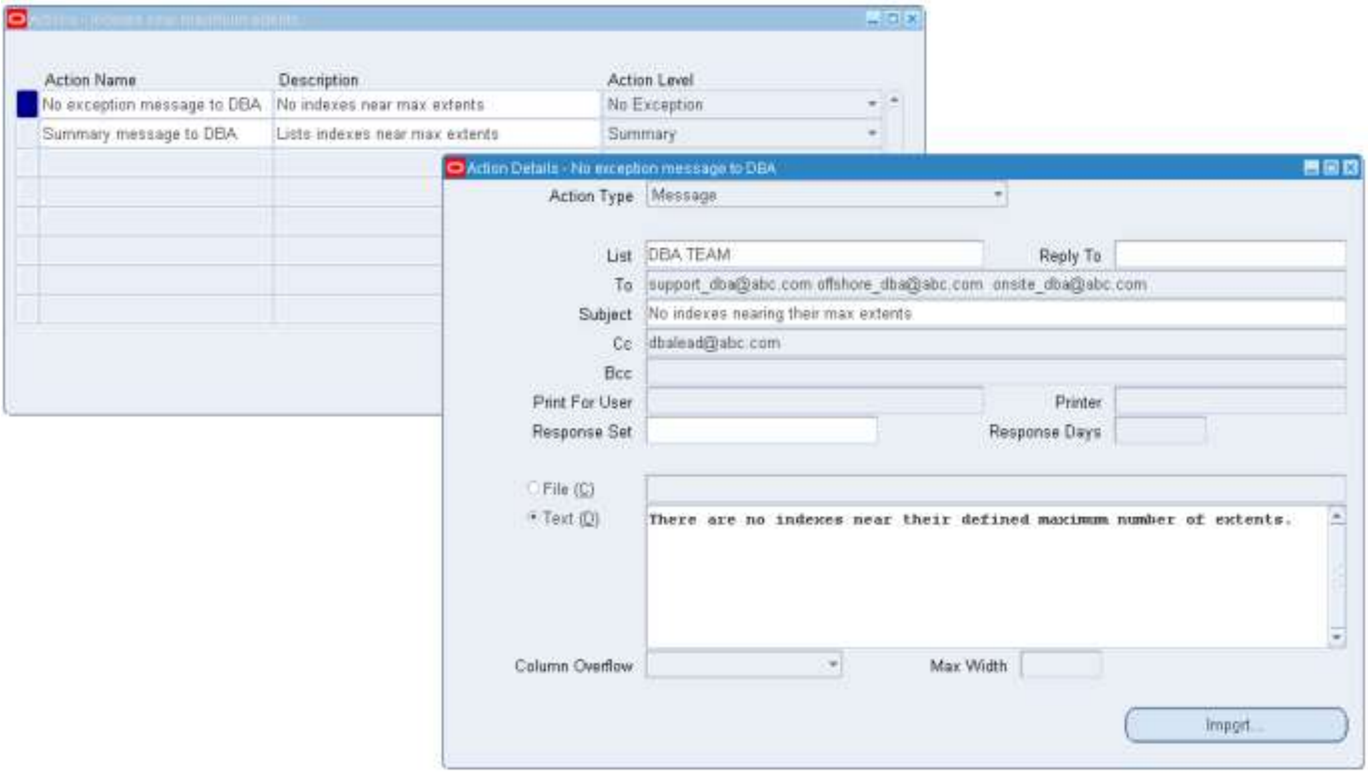
Keep: 10 Days
End Date:
Last Checked: 24-DEC-2014

Select Statement:

```
select owner, tablespace_name, segment_name,
extents, max_extents,
:available_extents
into   &OWNER, &TABLESPACE, &INDEX_NAME, &EXTENTS,
&MAXEXTS,
&AVAILABLE_EXTENTS
from   dba_segments
where segment_type = 'INDEX'
and segment_name like
decode(upper(:index_name),'ALL','%',upper(:index_name))
and owner = decode(upper(:user),'ALL',owner,upper(:
```

Select owner, tablespace_name, segment_name,

```
extents, max_extents,
:available_extents
into   &OWNER, &TABLESPACE, &INDEX_NAME, &EXTENTS, &MAXEXTS,
&AVAILABLE_EXTENTS
from   dba_segments
where segment_type = 'INDEX'
and segment_name like
decode(upper(:index_name),'ALL','%',upper(:index_name))
and owner = decode(upper(:user),'ALL',owner,upper(:user))
and max_extents - extents < :available_extents
order by owner, tablespace_name, segment_name
```



Action Name	Description	Action Level
No exception message to DBA	No indexes near max extents	No Exception
Summary message to DBA	Lists indexes near max extents	Summary

Action Details - No exception message to DBA

Action Type: Message

List: DBA TEAM

Reply To:

To: support_dba@abc.com offshore_dba@abc.com onsite_dba@abc.com

Subject: No indexes nearing their max extents

Cc: dbalead@abc.com

Bcc:

Print For User:

Printer:

Response Set:

Response Days:

○ File (C)

● Text (Q)

```
There are no indexes near their defined maximum number of extents.
```

Column Overflow:

Max Width:

TASK 3: Create an alert which will send mail to a distribution list every hour and send the list of tablespaces which are reaching 80% of its allocated space.