

ERP- Oracle Apps

Oracle Application Server



Copyright © 2011 IGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of IGATE Corporation.

IGATE Corporation considers information included in this document to be Confidential and Proprietary.

Document History

Add info here.

Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
2-Feb-2012	1.0		Amit Sali	Content Creation
2-Feb-2012			CLS Team	Review

Course Goals and Non Goals

- **Course Goals**
 - Understand concepts of ERP and It's Implementation in Oracle Application
- **Course Non Goals**
 - DBA Activities of Oracle Application



Pre-requisites

- **Knowledge of SQL/PLSQL**
- **Knowledge of Unix**
- **Basic knowledge of D2K (forms and Reports)**

Intended Audience

- Programmers and Analysts



Day Wise Schedule

- **Day 1**

- Lesson 1: ERP Overview
- Lesson 2: Introduction to Oracle Apps
- Lesson 3: Oracle Applications File System

- **Day2**

- Lesson 4: Multiple Organisations in Oracle Applications
- Lesson 5: User profiles

- **Day3**

- Lesson 6: Value Set
- Lesson 7: Flexfields
- Lesson 8: Key Flexfields
- Lesson 9: Defining Descriptive Flexfields

Day Wise Schedule

- **Day 4**

Lesson 10: Form Registration

- **Day 5**

Lesson 11: Managing Concurrent Programs and Reports

Lesson 12: Registering Concurrent Programs in Oracle Applications

Lesson 13: Oracle Application 11i Interface

- **Day 6**

Lesson 14: SQL Loader

Lesson 15: Object Standards

Lesson 16: Coding Standards

Lesson 17: Best Practices - Aim Methodology



Day Wise Schedule

- **Day 7**

Lesson 18: Customization using Custom.pll

Lesson 19: Personalization of Forms

- **Day 8**

Lesson 20: Workflow

- **Day 9**

Lesson 21: Introduction to R12

- **Day 10**

Lesson 21: Introduction to R12



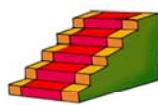
References

- **Student Guide**
 - All slides presented during lecture along with explanation
- **Lab Guide**
 - Hands-on lab exercises with sample solutions
- **Internet**
 - <http://apps2fusion.com/>
 - E-TRM



Next Step Courses (if applicable)

- NA



Other Parallel Technology Areas

- SAP as ERP

ERP- Oracle Apps

Lesson 1: ERP Overview

August 14, 2014 | Proprietary and Confidential | 1*



Lesson Objectives

➤ To understand the following topics:

- Enterprise?
- ERP
- Evolution of ERP
- Architecture and Module



What is an Enterprise?

- Is a collection of organizations and people formed to create and deliver products to customers.



August 14, 2014 | Proprietary and Confidential | + 3 +

Enterprise

An economic system in which goods and services are exchanged for one another or money, on the basis of their perceived worth. Every business requires some form of investment and a sufficient number of customers to whom its output can be sold at profit on a consistent basis.

A business (also known as enterprise or firm) is an organization designed to provide goods, services, or both to consumers.

The activity of providing goods and services involving financial and commercial and industrial aspects.

The entire organization, including all of its subsidiaries. It implies a large corporation or government agency, but it may also refer to a company of any size with many systems and users to manage.

Definition of ERP

- Software solution that addresses the enterprise needs taking the process view of an organization to meet the organization goals tightly integrating all function of an enterprise.

August 14, 2014

Proprietary and Confidential

+ 4 +



Enterprise Resource Planning

ERP is software solution that takes the process view to meet organizational goals, tightly integrating all functions of the enterprise.

ERP facilitates integration of company-wide information systems with the potential to go across companies.

ERP systems typically include the following characteristics:
An integrated system that operates in real time (or next to real time), without relying on periodic updates.

A common database, which supports all applications.
A consistent look and feel throughout each module.

What does ERP do?

- Facilitates Company-wide integrated Information Systems covering all Functional areas.

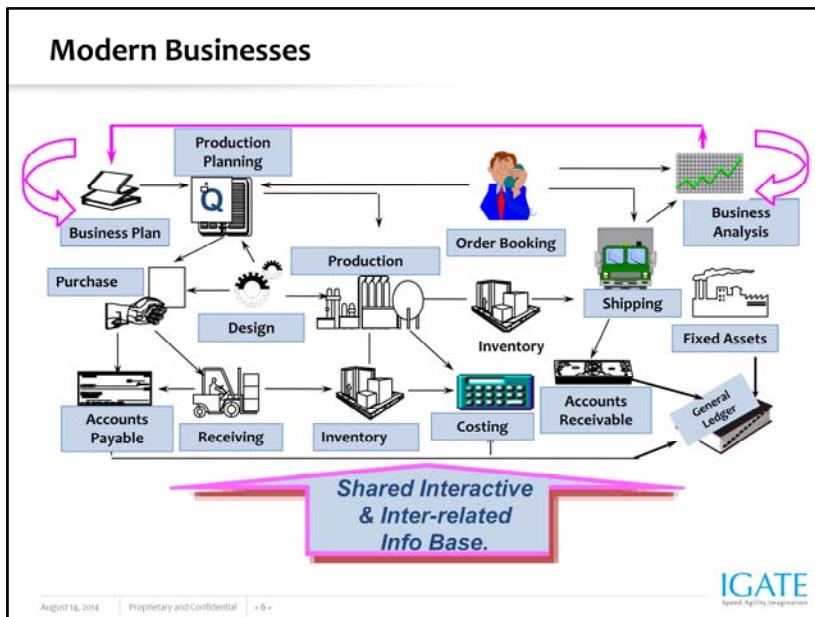
Enterprise resource planning (ERP)

ERP is a software architecture that facilitates the flow of information among the different functions within an enterprise. Similarly, ERP facilitates information sharing across organizational units and geographical locations.

It enables decision-makers to have an enterprise-wide view of the information they need in a timely, reliable and consistent fashion.

ERP provides the backbone for an enterprise-wide information system.

Enterprise resource planning (ERP) integrates internal and external management information across an entire organization, embracing finance/accounting, manufacturing, sales and service, CRM, etc. ERP systems automate this activity with an integrated software application. Its purpose is to facilitate the flow of information between all business functions inside the boundaries of the organization and manage the connections to outside stakeholders.



Basic Five Reasons for Having ERP

- Integrate Financial Information
- Integrate customer order information
- Standardize and speed up manufacturing processes
- Reduce inventory
- Standardize HR information

August 14, 2014 | Proprietary and Confidential | + 7 +



The Enterprise Resource Planning implementation streamlines the business processes of a company, bringing in efficiency in the operational areas. It provides the realistic, measurable and time benchmarked benefit.

These benefits can be broken down into

Better management of resources reducing the cost of operations and thus bringing down the number of order processing mistakes.

Allows a faster measurement of results.

Improved planning at functional and process level.

Customer satisfaction increases due to shorter delivery cycle.

Simultaneous activation of the decision centers because of instant inducement through triggers or updates.

Business operation transparency between business partners cutting down the execution time of critical business operations.

Down placement of decision making to lower levels, releasing the burden on middle management.

The strong interface capabilities of the system will provide for a better use of human resources due to the improved access to information.

The processes become faster due to work group technology and application of work flow automation.

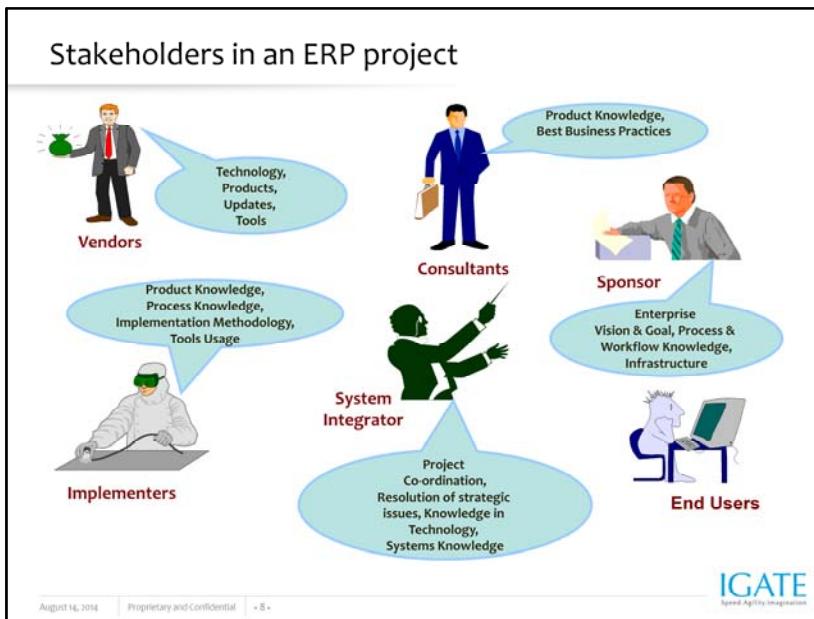
Will facilitate the future expansion of the business as the ERP system poses a scalable architecture.

The ERP implementation automatically leads to the usage of the best business procedures bringing consistency of operations.

With the use of data warehousing, management becomes more knowledge-driven and the organization will becomes a learned one.

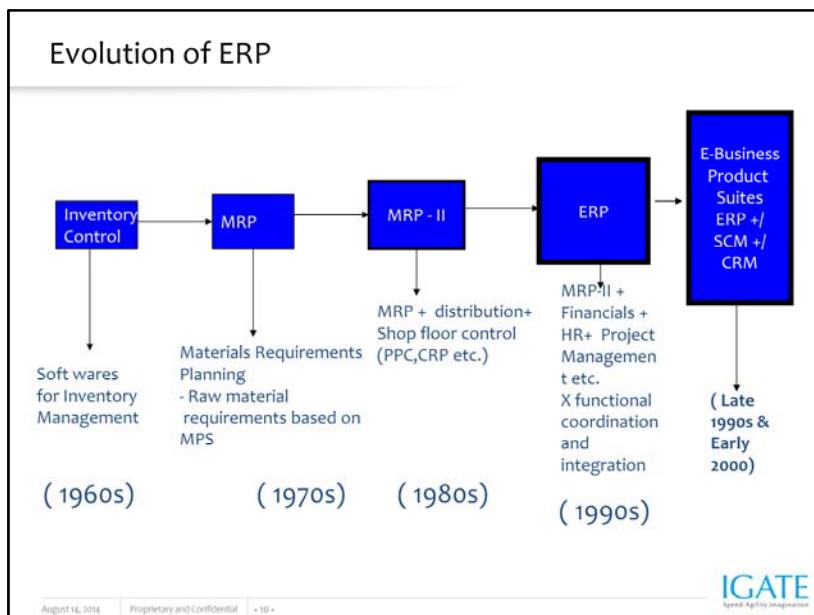
ERP Scope can be enlarged through the internet/intranet access; making the ERP sensitive to the latest events in the business, market and technology.

The quality of decision making improves as the users responsible for decision making are made alert and they are better informed.



Major ERP Products

- Oracle Applications
- SAP
- J.D. Edwards
- Baan
- PeopleSoft
- MAPICS
- MFG/PRO



Evolution of ERP

In 1990 [Gartner Group](#) first employed the [acronym](#) ERP as an extension of [material requirements planning](#) (MRP), later [manufacturing resource planning](#) and [computer-integrated manufacturing](#).

Without supplanting these terms, ERP came to represent a larger whole, reflecting the evolution of application integration beyond manufacturing. Not all ERP packages were developed from a manufacturing core.

Vendors variously began with accounting, maintenance and human resources. By the mid-1990s ERP systems addressed all core functions of an enterprise. Beyond corporations, governments and non-profit organizations also began to employ ERP systems.

ERP systems experienced rapid growth in the 1990s because the [year 2000](#) and the [Euro](#) disrupted legacy systems. Many companies took this opportunity to replace such systems with ERP. This rapid growth in sales was followed by a slump in 1999 after these issues had been addressed.

Evolution of ERP (Contd...)

➤ Oracle Applications R12 is an E business product suits developed by Oracle Corporation, USA.

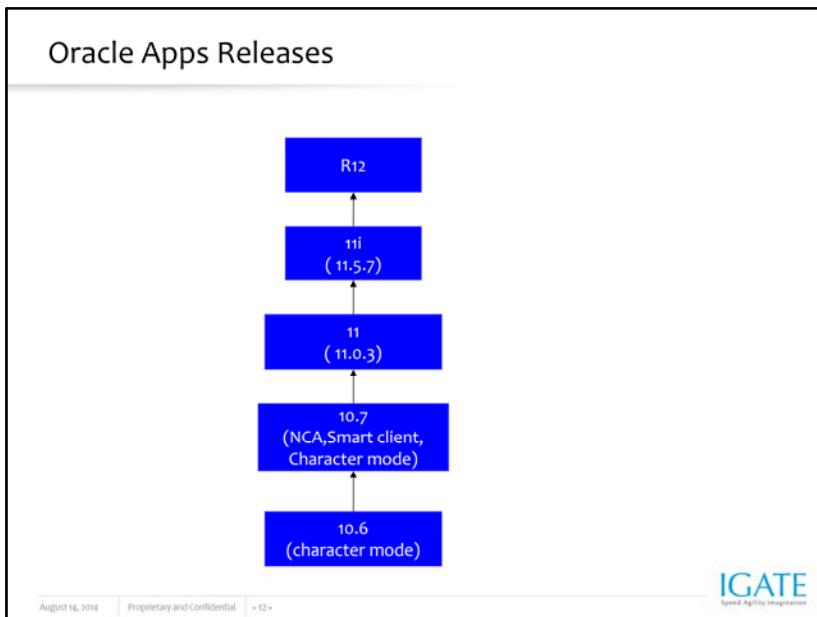
➤ **Product Suites**

- ERP (Back office functions)
 - Financials
 - Manufacturing and distribution
 - HR etc.
- CRM (front office functions)
 - Call center Management
 - Internet Sales and Marketing
 - E-commerce



ERP systems initially focused on automating back office functions that did not directly affect customers and the general public. Front office functions such as customer relationship management (CRM) dealt directly with customers, or e-business systems such as e-commerce, e-government, e-telecom, and e-finance, or supplier relationship management (SRM) became integrated later, when the Internet simplified communicating with external parties.

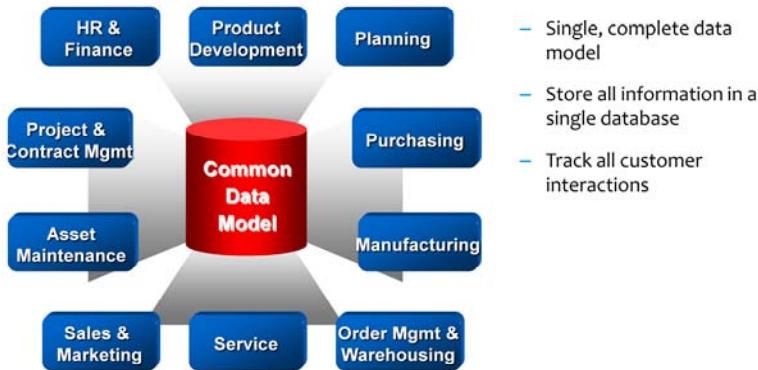
"ERP II" was coined in the early 2000s. It describes web-based software that allows both employees and partners (such as suppliers and customers) real-time access to the systems. "Enterprise application suite" is an alternate name for such systems

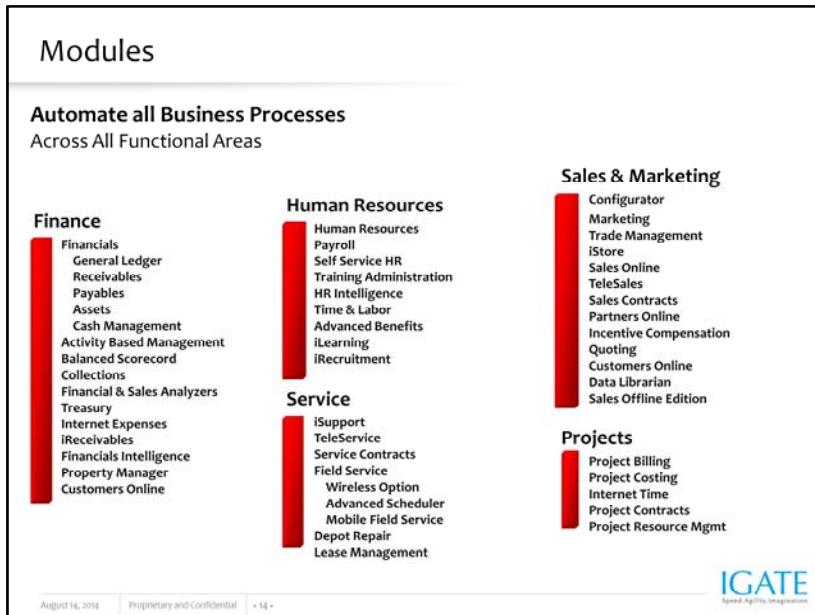


The Basic Architecture

Built on a Common Data Model

Provide Single Version of the Truth





ERP Modules

There are many ERP modules in ERP software solution. Each ERP software module corresponds to a major functional area of an organization. Generally it includes, modules for material purchasing, inventory control, product planning, product distribution, tracking the order, financial ERP module, accounting, marketing, and human resource. Organizations implement the modules in ERP that are both economically and technically feasible and profitable.

Finance

All kind of organizations small scale, large scale organizations benefit from the implementation of ERP financial module. The financial module is the core of many ERP software systems. It can gather financial data from various functional departments, and generates valuable financial reports such general ledger, trial balance, asbalance sheet and quarterly financial statements.

Human Resources

Human Resources is another widely implemented ERP module. HR module streamlines the management of human resources and human capitals. HR modules routinely maintain a complete employee database including contact information, salary details, attendance, performance evaluation and promotion of all employees. Advanced HR module is integrated with knowledge management systems to optimally utilize the expertise of all employees.

Inventory

Inventory module facilitates processes of maintaining the appropriate level of stock in a warehouse. The activities of inventory control involves in identifying inventory requirements, setting targets, providing replenishment techniques and options, monitoring item usages, reconciling the inventory balances, and reporting inventory status. Integration of inventory control module with sales, purchase, finance modules allows ERP systems to generate vigilant executive level reports.

Sales and Marketing

Sales module implements functions of order placement, order scheduling, shipping and invoicing. Sales module is closely integrated with organizations' ecommerce websites. Many ERP vendors offer online store front as part of the sales module. ERP marketing module along with CRP supports lead generation, direct mailing campaign and other marketing works. Scheduling of the promotion is possible using this.

Summary

➤ **In this lesson, you have learnt:**

- What is an Enterprise?
- Definition of ERP
- What does ERP do?
- Basic Five Reasons for Having ERP
- Major ERP Products
- Evolution of ERP
- Oracle Apps Releases
- The Basic Architecture



Review – Questions

- **Question 1:** Enterprise is a collection of ___ and ___ formed to create and deliver products to customers
- **Question 2: Major ERP Products are**
 - Oracle Applications
 - TAP
 - J.D. Edwards
 - Baan



ERP- Oracle Apps

Lesson 2: Introduction to Oracle Apps

August 14, 2014

Proprietary and Confidential

+ 1 +



Lesson Objectives

➤ **To understand the following topics:**

- Logon and off of Oracle Applications
 - Log on to Oracle Applications
 - Choose a responsibility
 - Use the Oracle Applications Navigator
 - Discuss the various components of an Oracle Applications form
 - Log off of Oracle Applications
- Using forms
- Using Reports





Starting Oracle Applications

The first step in starting Oracle Applications is to enter the appropriate URL for your site in an Oracle Applications certified browser.

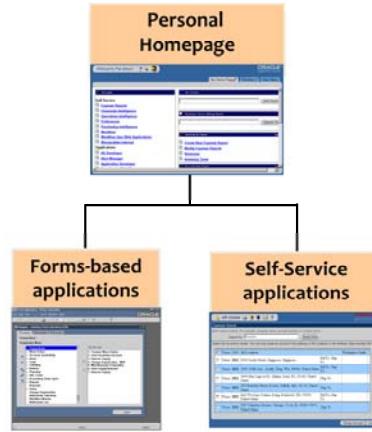
After starting Oracle Applications, the first window you see is the logon window.

You need an Oracle Applications user name and password, also known as an Oracle Applications user name, to log on to Oracle Applications. It is different from the user name and password you use to log on to your computer. If you are not sure of your Oracle Applications user name and password, consult your system administrator.

Oracle Applications security is based on your Oracle Applications user name. Your user name connects you to your responsibilities, which controls your access to applications, functions, reports, and data.

(Help) Oracle Applications User's Guide > Getting Started > Starting Oracle Applications > Starting and Logging On

Personal Homepage to Applications



IGATE
Speed. Agility. Innovation.

The Personal Homepage

After you log on to Oracle Applications, a Personal Homepage is displayed.

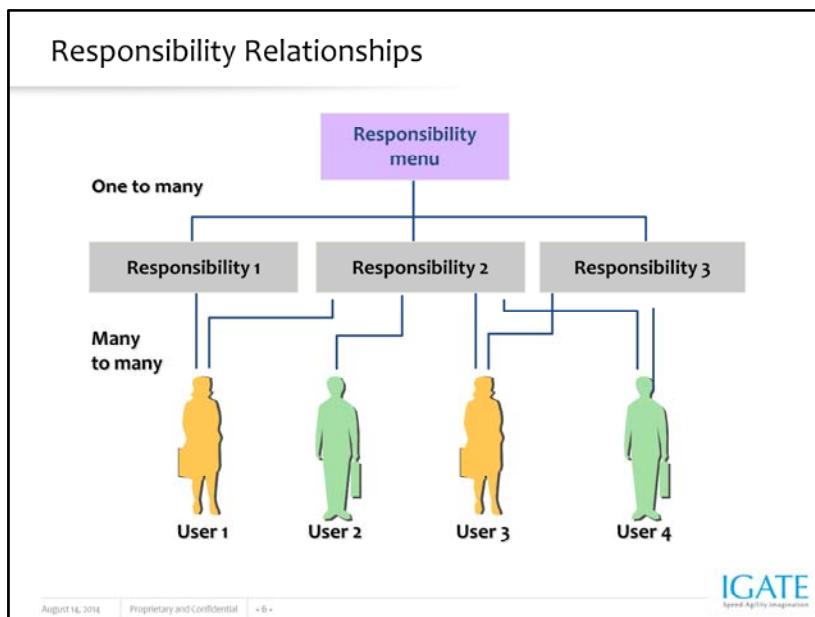
Responsibility Relationships: Many to One



IGATE
Spend. Agile. Imagine.

Selecting a Responsibility

Once you have used the logon form to begin the logon process, you must tell the system what type of access you will be using. A responsibility is a set of data, menus, and forms that defines your particular level of authority while using the system. For example, you would want the Accounts Payable department of your company to access the invoice forms of the system, but you would not want them to be able to access any payroll information. Another example is that the controller of a department would want to have access to all the data that his or her employees can use, so the controller would want access to both accounts payable and payroll information.



Responsibility Properties

The following is a list of the types of responsibilities and their particular properties that can be defined in Oracle Applications by your system administrator:

- A specific application (or applications), such as Oracle General Ledger.
- A set of books, such as Vision Operations or Vision Corporation or an organization, such as Vision Services or Vision Distribution.
- A restricted list of windows to which you can navigate. For example, a responsibility may allow certain Oracle Financials users to enter invoices, but not to enter suppliers (vendors) or customers.
- A restricted list of functions you can perform. For example, two responsibilities may have access to the same window, but the window of one responsibility may have additional functional buttons.
- Reports in a specific application. Your system administrator can assign groups of reports to one or more responsibilities, so the responsibility you choose determines the reports that you can submit.

Choosing a Responsibility

The screenshot shows the Oracle Applications Personal Homepage. On the left, there's a sidebar with 'Performance Measure' and 'AR of all Organizations'. The main content area has a 'Favorites' section with links like 'Oracle Home Page', 'Invoice and Payment Metrics', 'Travel and Ent. Metrics', and 'Cash Forecast'. Below that is an 'Ask Oracle' search bar. A blue callout box with a red arrow points from the text 'From the Personal Homepage click on a link to the responsibility' to the 'Oracle Home Page' link in the Favorites section.

From the Personal
Homepage
click on a
link to the
responsibility

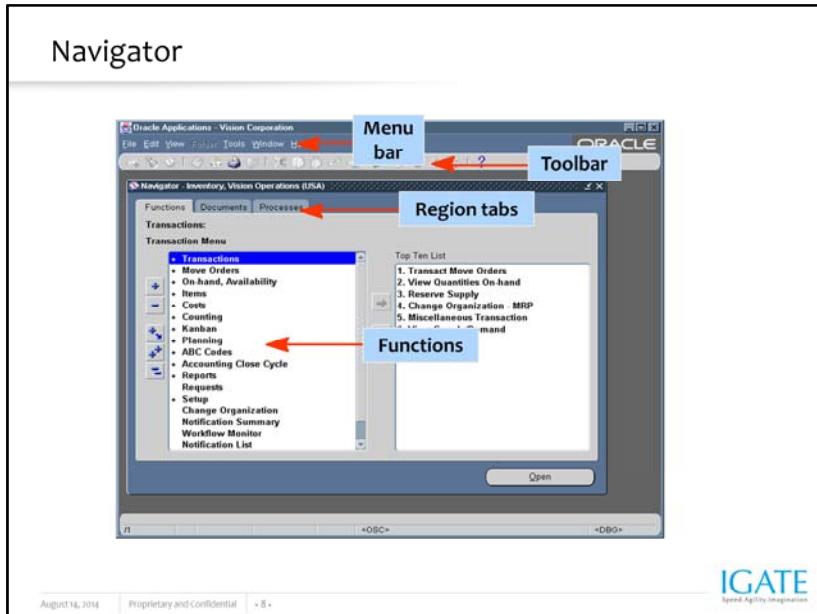
IGATE
Spend Agility Imagine

August 14, 2014 | Proprietary and Confidential | + 7 +

Responsibility Assignments

Each user has at least one responsibility and several users can share the same responsibility. Your system administrator can assign you any of the standard responsibilities provided with Oracle Applications, or create custom responsibilities for you. If you have only one responsibility with one function, and only one region on your Personal Homepage, you will go directly to that function.

Click on the underlined link in the Application section to select your responsibility.



Navigator Window

The Navigator window displays the name of the responsibility you select in the title bar. Use this window to navigate to a form so you can perform a specific business flow. You can navigate to the forms that are displayed in a navigation list on the left side of the Navigator window. You can click on the tabs to access different regions.

Navigator Region Tabs

Region Tab	Description
Functions	Displays the forms that you can navigate to in a navigation list on the left-hand side of the Navigate window.
Documents	Displays links to documents that you have created.
Processes	Displays a visual map of a business process.

August 14, 2014

Proprietary and Confidential

+ 9 +



Using Region Tabs

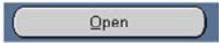
The Functions tab displays all of the applications functions that you can access for the responsibility that you selected.

If you have a document, such as a particular purchase order, invoice, or sales order that you want to access later, you can create a link to the document using the Navigator's Document feature. The Navigator's Document feature allows you to create as many links as you want and save them in the Documents region of the Navigator window. When you use a link to open a document, Oracle Applications opens the document in the appropriate form window. You can access the Document region using the tab control.

The Processes region of the Navigator (the “Process Navigator”) automates business flows across Oracle Applications forms. It allows you to model and execute complex business processes through an easy-to-use, graphical user interface. The business processes enabled through the Process Navigator can cross product boundaries and include complete business cycles.

The Process Navigator guides you step-by-step through each required function in a business process. In addition to providing a visual map of a business process, the Process Navigator can launch the appropriate Oracle Applications forms or standard reports at each step.

Expanding or Collapsing the Navigation List

- Choose one of the following methods to expand an item to its next sublevel form:
 - Double-click the item. 
 - Select the item and choose Open. 
 - Select the item and choose Expand. 
- To collapse an expanded item, select the item and choose Collapse.

August 14, 2014

Proprietary and Confidential

+ 10 +



Using the Navigation List

Each user can access the Oracle Applications forms in several ways so that they can use the system quickly, according to their own computer style. Use the various buttons on the Navigator to manipulate list items.

Expanding or Collapsing Several Items

- To expand or collapse several items at once, choose one of the following buttons:
- Expand All Children expands all the sublevels of the currently selected item.
 - Expand All expands all the sublevels of all expandable items in the navigation list.
 - Collapse All collapses all currently expanded items in the navigation list.



August 14, 2014 | Proprietary and Confidential | + 11 +

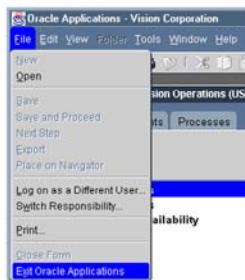
IGATE
Speed. Agility. Inspiration.

Using the Navigation List (continued)

Click any of these buttons to expand or collapse several items listed on the Navigator Menu. These functions are also available in the Tools pulldown menu.

Logging Off of Oracle Applications

- (M) File > Exit Oracle Applications
- Use this method so that your username is cleared from system access



IGATE
Speed Agility Inspiration

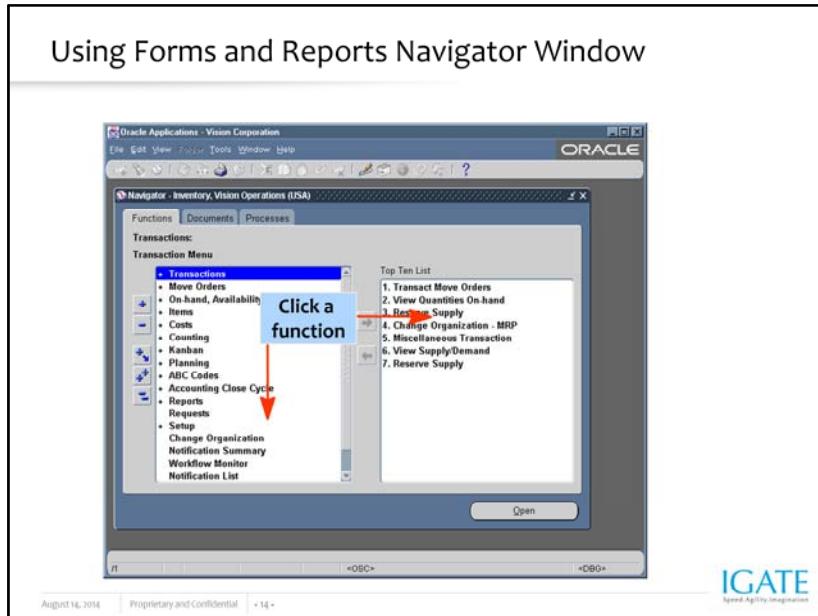
Exiting Oracle Applications

(M) File > Exit Oracle Applications, to log off the system. It is important to exit the system in this manner, rather than any other, as this is the only way to ensure that your user name is cleared from system access. You can also close the MDI window.

Demo : Login and Logoff

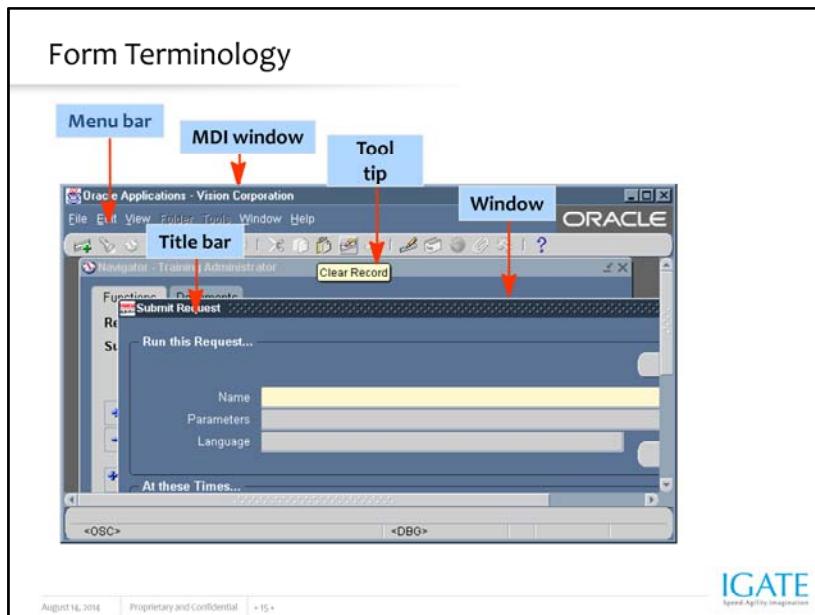
- Log on to Oracle Applications
- Choose the responsibility
- Exit Oracle Applications properly





The Navigator Window

Use the Navigator window to navigate to a form that lets you perform a specific business activity. The Navigator window is always present during your session of Oracle Applications and displays the name of your current responsibility in its window title.



Form Terminology

Oracle Applications Release 11i works specifically in a Web-enabled environment. It is important to understand the terminology of the components within an Oracle Applications form. Common terms used in Oracle Applications forms are listed below.

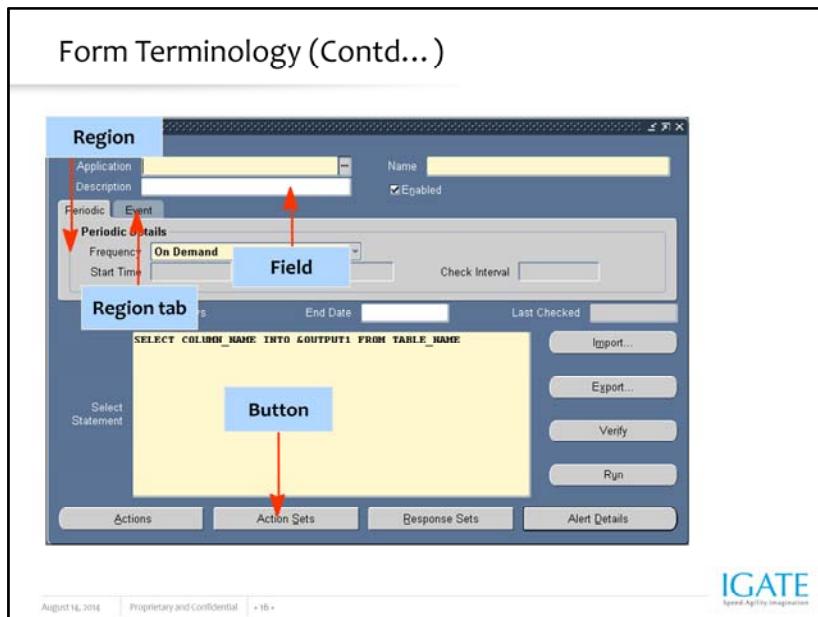
Menu bar—Use pull-down menus from this menu bar to navigate or perform actions within a form

Window—An area where the user interacts with an application (Many windows can be open at one time and you can access these “overlapping” windows to perform data entry or data search activities.)

Window title—Text in the title bar that indicates the name of the window, and usually, context information pertinent to the information in that window

MDI window—A master container window that houses all windows, toolbars, and application windows

Tool tips—Iconic bubble help that you can use to determine the function of a button on the toolbar



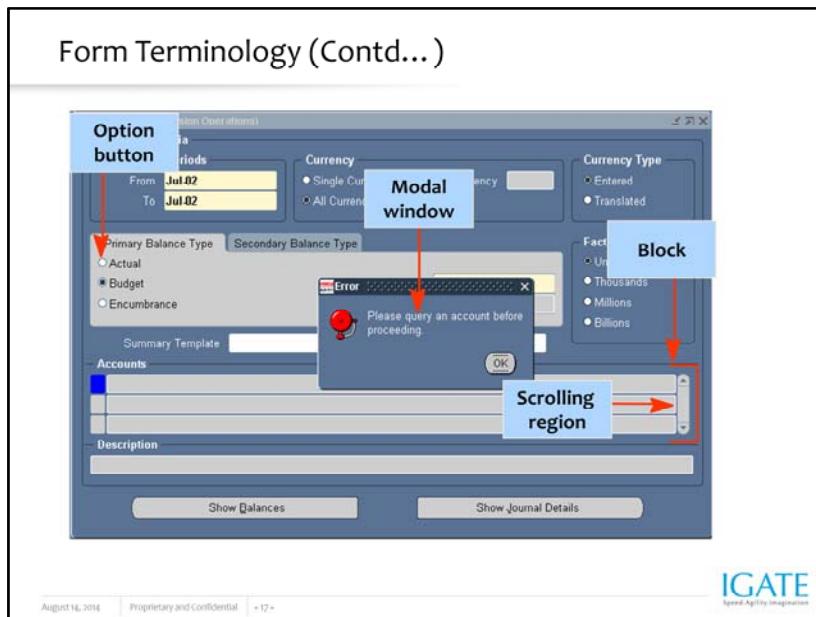
Form Terminology (continued)

Region—A logical grouping of fields set apart from other fields by a box outline

Region tab—A collection of regions that occupy the same space in a window where only one region can be displayed at a time

Field—An area in a window that displays data or enables you to enter data

Button—A graphic element that initiates a predefined action when you click it



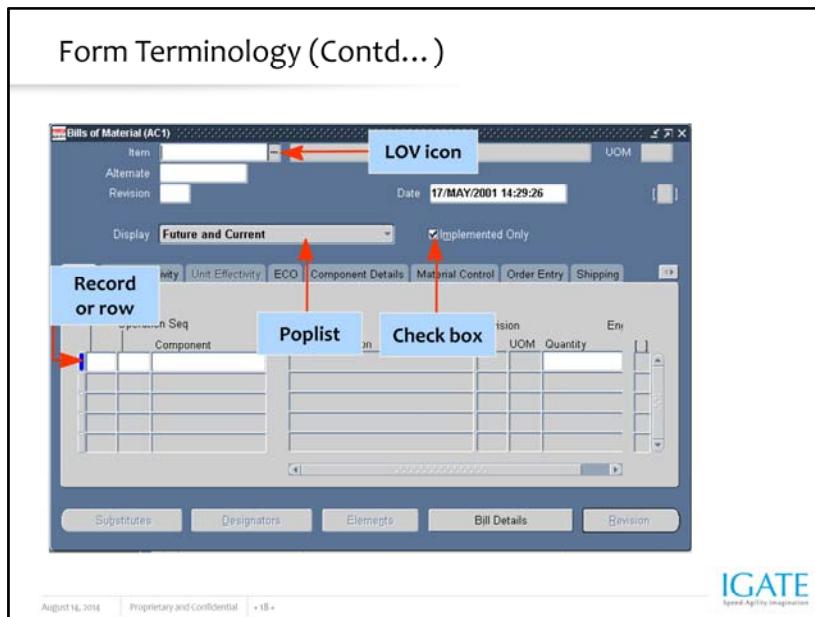
Form Terminology (continued)

Option button—A button that indicates an individual selection is available within an option group

Modal window—A window that requires you to act on its content before continuing

Scrolling region—A region, containing a scroll bar, in which to view other fields

Block—An area of information relative to a specific business function or entity



Form Terminology (continued)

Record or Row—A set of one or more related data items from a table or view that are grouped for processing

Check box—A box in which you can toggle between an on/off or yes/no state for a particular value

LOV icon—An icon that you can click to display a list of values (LOV) for the current field

Poplist—A poplist lets you choose a single value from a short list

Field Colors

Field Color	Description
White	Allow data entry
White with green text	Indicate drill-down capability
Yellow	Require data entry
Gray with black text	Are display only
Blue	Indicate fields to use in Query-Enter mode

What Field Colors Indicate

Each block contains fields you use to enter, view, update, or delete information. A field prompt describes each field by telling you what kind of information appears in the field or what kind of information you should enter in the field. Fields are color coded to indicate their type as follows:

White Fields—allow data entry

White Fields with Green Text—indicate drill-down capability

Yellow Fields—require data entry

Gray Fields with Black Text—are display-only

Blue Fields—indicate fields to use in Query-Enter mode

The term field generally refers to a text field, an area in a window that either displays data or allows you to enter data. However, a field can also include a button, check box, option group, or poplist.

The Toolbar

The diagram illustrates the Oracle Apps Toolbar with several groups of functions:

- Tool tip**: A callout box pointing to the "New" button on the toolbar.
- Save**
- Next Step**
- Print**
- Close Form**
- Edit Field**
- Zoom**
- Translations**
- Attachments**
- Folder Tools**
- New**
- Find**
- Show Navigator**
- Cut**
- Copy**
- Paste**
- Clear Record**
- Delete**
- Help**

Source: Oracle Applications

August 14, 2014 | Proprietary and Confidential | + 20 +

IGATE
Speed. Agility. Inspiration.

Using the Toolbar

The toolbar is a collection of iconic buttons, where each button performs a specific action when you choose it. Each toolbar button replicates a commonly-used menu bar item. Depending on the context of the current field or window, a toolbar button can be enabled or disabled. You can display help or a tool tip for an enabled toolbar button by holding your mouse over the button.

Creating New Records

New



(M) File > New

August 14, 2014

Proprietary and Confidential

+ 21 +

IGATE
Spend. Agile. Imagine.

Creating a New Record

When you add a new record to the database, Oracle Applications will move the current record down and insert a new blank row. You add a new record by entering information into this blank row. In most screens you are automatically on a new record when the form is opened, so you can just start entering the information.

After you finish entering the data for your new record, you must remember to click Save so that your newly added information will be written to the database. If you enter five new blank rows of information but do not click Save when you are finished, none of this newly added information will be updated in the database (the system will prompt you to save before you exit).

How to Create a New Record

(M) File > New. Or, click the New icon on the toolbar.

You can also press the down arrow on your keyboard to insert a blank row, if you are not in a table on the form.

Additionally, some applications have a button labeled New, and clicking it will insert a blank row.

Type the new information into the blank row.

(M) File > Save or click the Save icon to save the new record.

Deleting Records

Delete



(M) Edit > Delete

August 14, 2014

Proprietary and Confidential

+ 22 +

IGATE
Speed. Agility. Inspiration.

Deleting a Record from the Database

If you want to delete a record from the database, you must first display the record on your form and then click the Delete icon.

Note: This option is not always available. For example, in Oracle Applications you cannot delete a customer once they have been entered; you can only deactivate their status.

If you do not Save this new change to the database, the record is not truly deleted (the system will prompt you to save before you exit).

How to Delete a Record

(M) Edit > Delete, or click the Delete icon.

(M) File > Save or click the Save icon.

Concurrent Processing

- Every Oracle Applications product contains reports and programs that are specific to that product
- A report generates a summary or detail presentation of Oracle Applications information, whereas a program can perform a function
- The reports and programs you have access to are defined by the responsibility you use

August 14, 2014

Proprietary and Confidential

+ 23 +



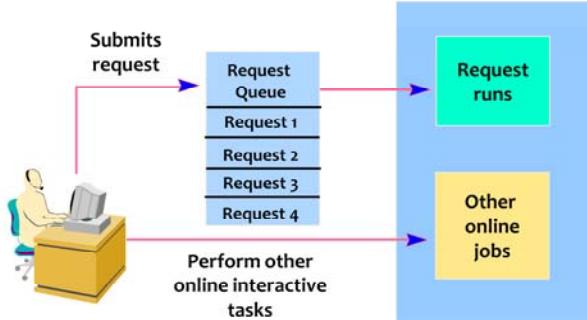
For example, as an Oracle Receivables user, you may be able to run a report to create invoices. Or in your Oracle General Ledger application, your responsibility may allow you to run a program to post journal entries.

Concurrent Processing (Contd...)

- Oracle Applications provides you with two features: Concurrent Processing and Standard Request Submission
- **Concurrent processing**
 - Is a feature that allows you to run a non-interactive, data-dependent function, such as a report or program, simultaneously with online operations
 - With concurrent processing, you can complete non-interactive tasks without interfering with the interactive work you perform at your computer

An example of concurrent processing occurs when you use the Post Journals window in your Oracle General Ledger application. Once you specify the journal batches to post and choose the Post button, your Oracle General Ledger application uses concurrent processing to post the journal batch entries without further involvement from you. Meanwhile, your computer is still available for you to continue doing other work in Oracle Applications. Oracle Applications runs all of its reports and programs as concurrent processes.

Concurrent Processing (Contd...)



Standard Request Submission

➤ Standard Request Submission

- Helps to run the reports and programs that are a part of your responsibility
- Is a feature that works with concurrent processing
- Provides a set of windows to run, submit and control over the output of reports and programs

August 14, 2014

Proprietary and Confidential

+ 26 +



Standard Request Submission

To help you run the reports and programs that are a part of your responsibility.

It is a feature that works with concurrent processing to provide a common interface for running your Oracle Applications reports and programs.

Standard Request Submission provides you with a set of windows for running reports and programs and a set of windows for creating groups of reports and programs to run together.

These windows give you control over the submission and output of your reports and programs.

Uses

- You can Concurrent processing helps you satisfy the following business needs
 - Keep working at your computer while running data-dependent reports and programs
 - Fully use the capacity of your hardware by executing many application tasks at once
- Standard Request Submission lets you satisfy a related set of business needs
 - You can use a standard interface to run your programs and reports
 - Control access to different programs and reports

August 14, 2014

Proprietary and Confidential

+ 27 +



Pass parameters from your environment to your reports and programs.

- View report output online.
- Create and run sets of reports and programs.
- Automatically run programs, reports, or request sets at specific time intervals.
- Specify whether reports and programs in a request set run sequentially or simultaneously.
- Specify whether to continue with a request set if a report or program in a sequential set fails.
- Specify alternative requests to run based on the completion status of previously run requests in a request set.
- View a log file that summarizes the completion information about all the reports and programs in a request set.

Basic Definitions

➤ Concurrent Program

- A concurrent program is a program that does not require continued interaction on your part to perform a specific task. For example, a concurrent program may be a program written to create a report, or to post a batch of general ledger journal entries

Basic Definitions (Contd...)

➤ Concurrent Process

- A concurrent process is an instance of a running concurrent program
- Each time a concurrent manager receives a request and runs a concurrent program, it creates a new concurrent process. A concurrent process can run simultaneously with other concurrent processes (and other activities on your computer)

Basic Definitions (Contd...)

➤ Concurrent Request

- A concurrent request is a request that you submit to run a concurrent program as a concurrent process. You issue a concurrent request when you submit a report or program to run using Standard Request Submission or when you choose an action button in a product-specific submission window

➤ Concurrent Manager

- A concurrent manager is a component of concurrent processing that monitors and runs tasks without tying up your computer

Basic Definitions (Contd...)

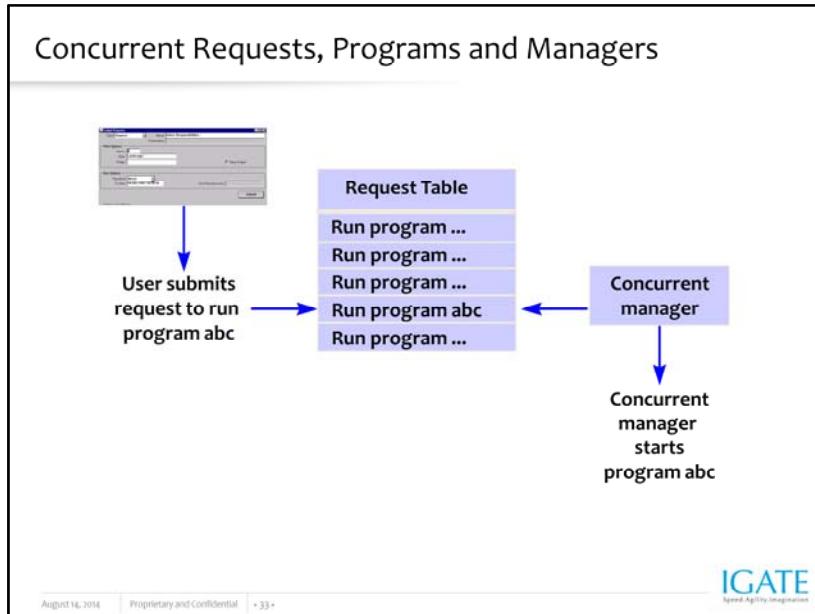
➤ Report

- A report is an organized presentation of specific Oracle Applications information. You can view a report online, or send it to a printer. The content of a report can range from summary information to a complete listing of values. Reports run as concurrent programs in Oracle Applications

Basic Definitions (Contd...)

➤ Parameter

- In Standard Request Submission, a parameter is a report variable whose value you can change each time you run a report. For example, you might run an audit report that requires you to enter an audit date each time you run the report. The audit date is a parameter for the report

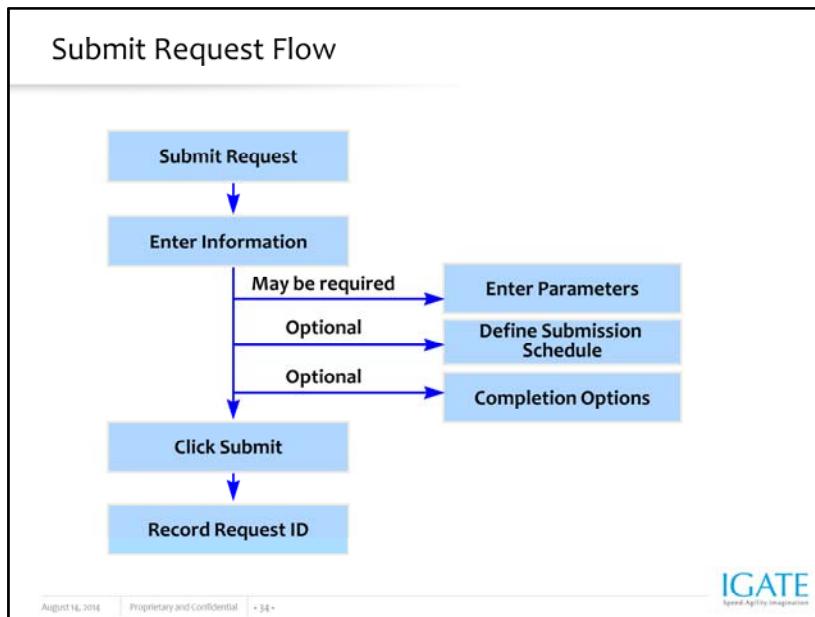


Concurrent processing allows long-running, data-intensive programs to run simultaneously with online operations.

Oracle Applications programs can run concurrently with each other as well as with other programs; they are referred to as concurrent programs.

Requests to run Oracle Applications programs—for example, to run an Oracle General Ledger report—are concurrent requests. Each concurrent request inserts a row into a database table maintained by the Oracle Application Object Library.

Concurrent managers read requests from the requests table and start concurrent programs.



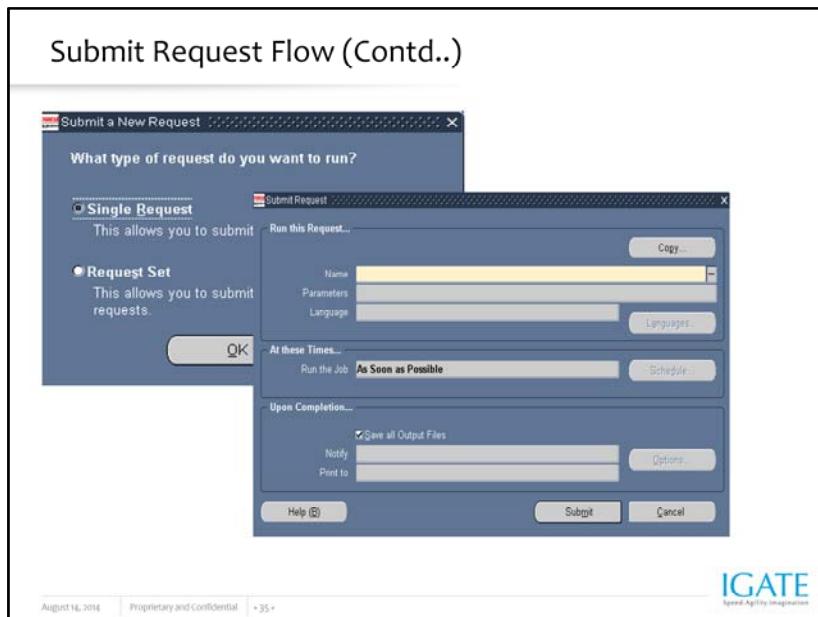
Using Standard Request Submission (SRS)

Using Standard Request Submission gives you control over how you can run your requests and request sets.

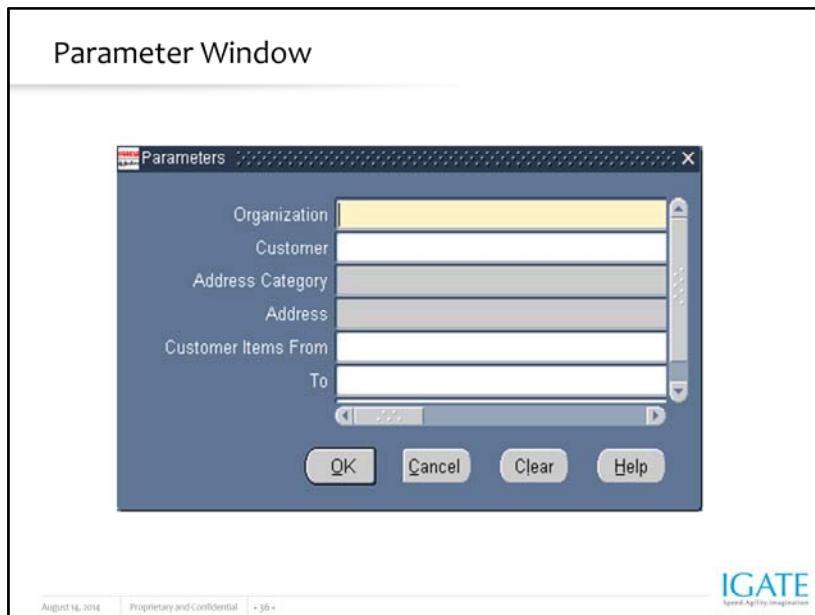
There are three elements involved in submitting a request: selecting the request or request set to be submitted, defining a submission schedule, and providing completion options.

Defining a schedule can be as simple as submitting As Soon as Possible, or it can involve using a more complex schedule that you define when you first submit your request. This schedule can then be used for other requests in the future. Completion options enable you to deliver notification to others using Oracle Workflow, as well as specifying which printers and how many copies of the output you want to produce for each request.

You can submit as many requests as you like from the Submit Request window. You can even submit a request more than once if you want to run the same request with different parameter values.



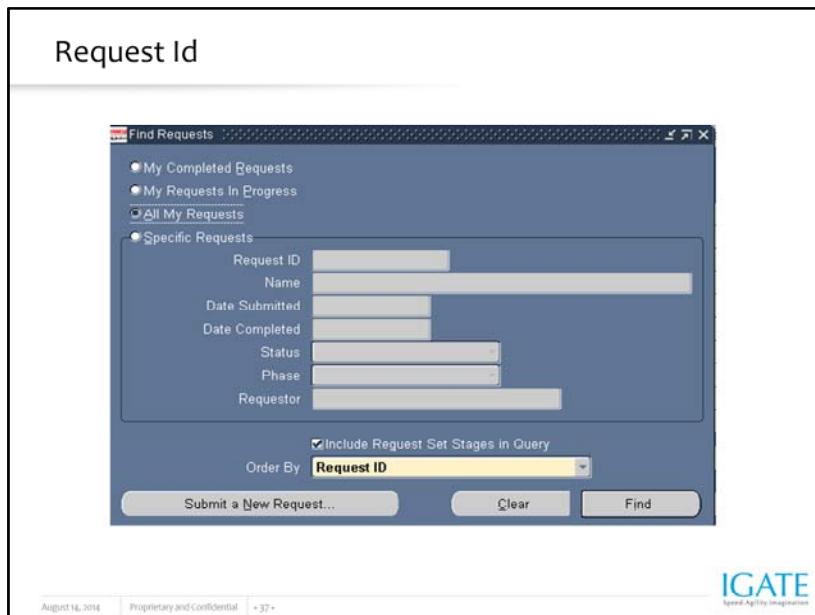
Navigate to the Submit a New Request Window:
Check the option for Single Request or Request Set.
Click OK.
Use the Copy a Prior Request button to use a previously entered request submission Or Select the name of the request that you want to run from the list of values.



A Parameters window automatically appears if you select a request that requires parameter values.

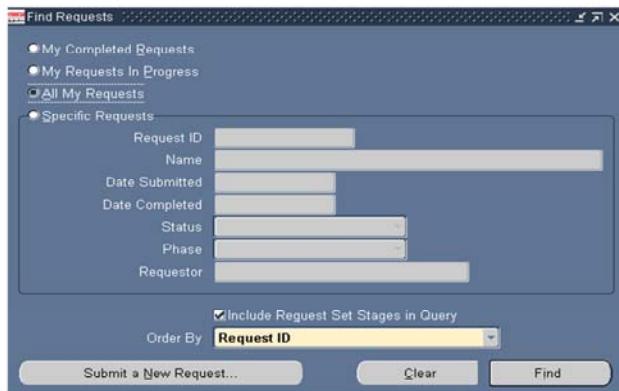
The Prompts in the Parameters window are specific to the request that you select.

The parameters you enter are concatenated and displayed in the Parameters field of the Submit Requests window.



Oracle Applications assigns a request ID to each request submission so that you can identify your request.
Use the request ID to query for your request output in the Requests window.
Oracle Applications assigns a new request ID to each resubmission of a request and displays the request ID of the previous request in the log file.

Use the Requests Window to View and Change Requests



August 14, 2014

Proprietary and Confidential | +38 +

IGATE
Speed. Agility. Inspiration.

Use the Requests window to perform the following functions:

- View all submitted concurrent requests
- Check the status of requests
- Change aspects of a request's processing options
- Diagnose Errors
- Find the position of a request in the queues of available concurrent managers

Using the Requests Window



IGATE
Spend Agility Imagine

Use the various buttons to perform tasks related to concurrent processing:

Refresh Data - Requeries the lines in the request table.

Find Requests - Displays the Find Request window to perform a search.

Submit a New Request... - Displays the Submit a New Request window.

Hold Request - Puts a request on hold if the request has not started running.

Cancel Request - Cancels a request

Demo

- Open/Close and save form
- Submit a concurrent request and see the output (Report)



Summary

- In this lesson, you should have learned how to:
 - Log on to Oracle Applications
 - Log off of Oracle Applications
 - Opening a form
 - Form terminology and characteristics
 - Running concurrent program and get output



Review – Questions

- **Question 1:** Parameter window is always mandatory while submitting concurrent program?
 - True/False
- **Question 2:** Oracle Applications assigns a _____ to each request submission so that you can identify your request.



Review – Questions

The screenshot shows the Oracle Apps Bills of Material (BOM) screen. The interface includes fields for Item, Alternate, Revision, and UOM. A date stamp indicates 17/MAY/2001 14:29:26. A checkbox labeled 'Implemented Only' is checked. The 'Display' dropdown is set to 'Future and Current'. Below the display area, there are tabs for Effectivity, ECO, Component Details (which is selected), Material Control, Order Entry, and Shipping. A large grid table is present, with its first column labeled 'Component'. To the right of the grid are columns for Revision, UOM, and Quantity. At the bottom of the screen are buttons for Substitutes, Designators, Elements, Bill Details, and Version.

1. Item
2. Display
3. Component Details tab
4. Grid table
5. Implemented Only checkbox

Knowledge Check

IGATE
Speed. Agility. Imagination

August 14, 2014 Proprietary and Confidential • 43 •

Practice Instructions
Label the numbered boxes in the slide.

Review – Questions

The screenshot shows the Oracle Bills of Material (AC1) application window. Several UI components are highlighted with numbered labels:

- 1. LOV**: Located at the top left, above the item number input field.
- 2. Record or row**: Located above the main grid area.
- 3. Poplist**: Located in the main grid area, pointing to a dropdown menu.
- 4. Tabbed region**: Located in the main grid area, pointing to a tabbed interface.
- 5. Block**: Located at the top right, above the date and revision fields.

On the right side of the window, there is a yellow box containing a clipboard icon with a question mark, a checkmark, and a red X, labeled "Knowledge Check".

At the bottom of the window, there are several buttons: Substitutes, Designators, Elements, Bill Details, and Revision. The date "August 14, 2014" and the text "Proprietary and Confidential" are also visible at the bottom.

IGATE
Speed. Agility. Inspiration.

Instructor Note

Review the answers to the labeling practice

ERP- Oracle Apps

Lesson 3: Oracle Applications File System

August 16, 2014

Proprietary and Confidential

- 1 -

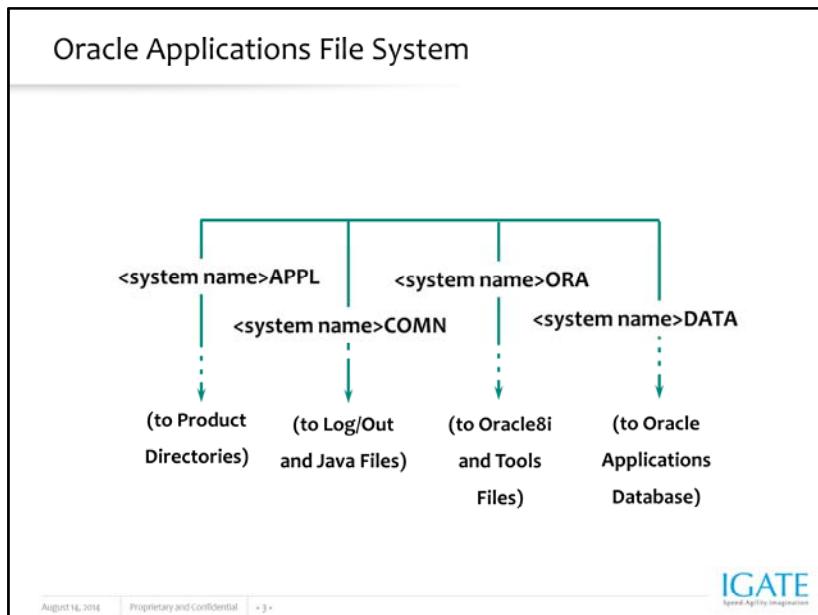


Lesson Objectives

➤ **To understand the following topics:**

- Describe the Oracle Applications file system.
- Identify the APPL_TOP, COMN_TOP, ORA_TOP, and DATA_TOP directories.
- Describe the file types available on the file system.
- Describe the product subdirectory structure.





Oracle Applications File System

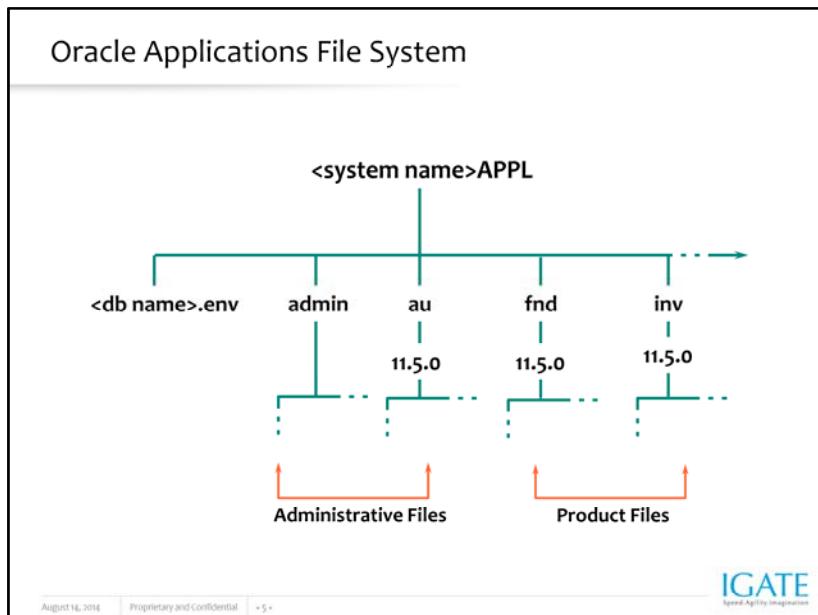
The Oracle Applications 11i system uses components from many Oracle products. It stores these product files within several different top level directories:

- <system name>APPL or APPL_TOP: contains the product directories and files for Oracle Applications.
- <system name>COMN or COMN_TOP: Contains directories and files used across products. <system name>COMN is also known as COMMON_TOP.
- <system name>ORA or ORA_TOP: Contains ORACLE_HOMEs for the technology stack components.
- <system name>DATA or DATA_TOP: Contains the Oracle Applications database files.

Note: <system name> is the name of your system determined through Rapid Install at the time of installation. For example, PROD.

Application Directory Structure

- The top directory in Oracle Applications is \$APPL_TOP
- \$APPL_TOP is an environment variable
- An environment variable is an operating system variable that describes an aspect of the environment in which the application runs. You can define an environment variable to specify a directory path. For example \$APPL_TOP maps to /ccnv01/oracle/ccnvappl



Oracle Applications File System

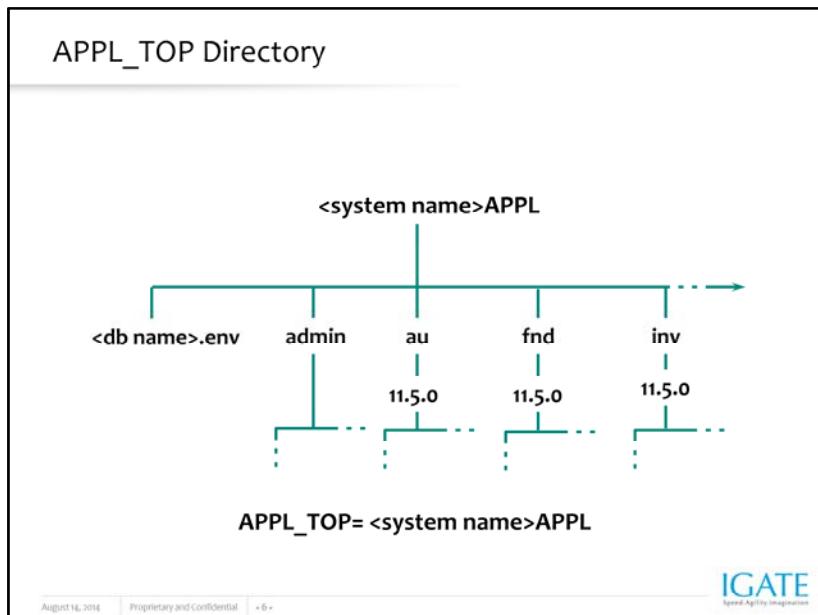
The Oracle Applications file system contains the product directories for Oracle Applications. The Oracle Applications file system contains:

The Oracle Applications environment files. The default name of the main Applications environment file is <db name>.env, where <db name> is the name of the database.

A directory for each of the products, licensed or not, that has been installed under the <system name>APPL, or APPL_TOP directory. Product directories use the standard product abbreviations.

Additional directories for administration and maintenance, such as the admin and au directories. AU is a product and is used for the maintenance of Oracle Applications.

Note: For Release 11i, all Oracle Applications products, regardless of license status, are installed in the database and the file system. Do not attempt to manually remove files for unlicensed products.



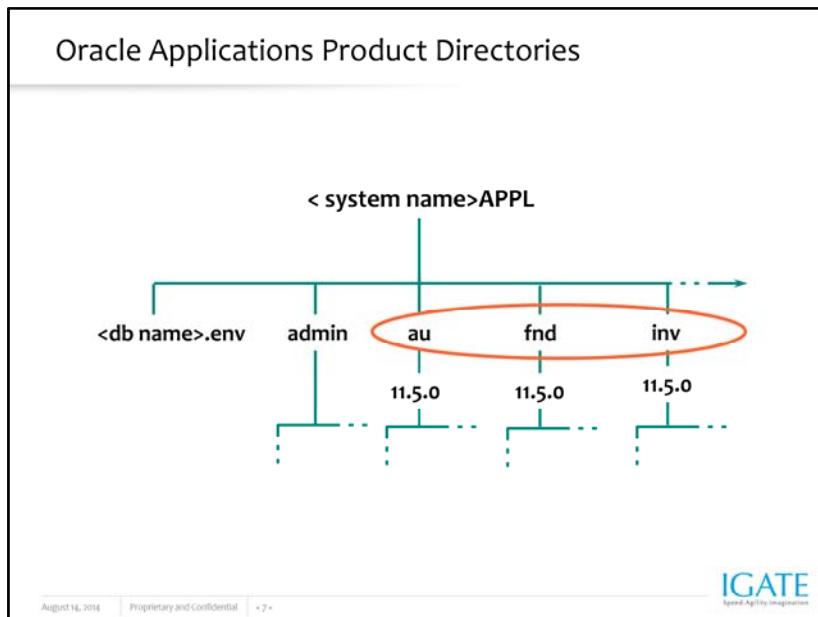
APPL_TOP Directory

The Oracle Applications top level directory path is defined in the environment variable APPL_TOP.

For both an installation and upgrade of Oracle Applications, Rapid Install creates the new APPL_TOP, or top Applications directory and defaults the APPL_TOP directory value to <system name>APPL.

This variable is used in subsequent directory definitions, and is also used as a term to refer to a particular Oracle Applications file system.

Technical note: The <db name>.env file is a very important file containing parameters defining the Oracle Applications environment. Typically, Rapid Install creates the <db name>.env file during the installation. Many of the parameters located in the <db name>.env file define important directories within the Oracle Applications file structure. For example, the APPL_TOP directory is identified in the environment parameter APPL_TOP. Additional parameters point to product top directories.



Oracle Applications Product Directories

Each product has its own subdirectory under APPL_TOP and the Oracle Applications base release is typically reflected in the subdirectory name. For Release 11i, the base release is 11.5.0. Multiple releases and product versions must not exist in a single APPL_TOP directory.

The product subdirectory name is defined in an environment variable <PROD>_TOP, where < PROD> is the product short name.

For example, the slide shows three product directories. The paths to these directories would be defined in the following declarations in the <db name>.env file:

```
APPL_TOP=/d01/prodappl
AU_TOP= /d01/prodappl /au/11.5.0
FND_TOP = /d01/prodappl /fnd/11.5.0
INV_TOP= /d01/prodappl /inv/11.5.0
```

Globalizations

The image shows a world map with four regions highlighted in green: North America (JL), Europe (JE), Asia/Pacific (JG), and Latin America (JA). The map is centered on the Atlantic Ocean.

IGATE
Empowering Imagination

August 16, 2014 | Proprietary and Confidential | 8 *

Globalizations

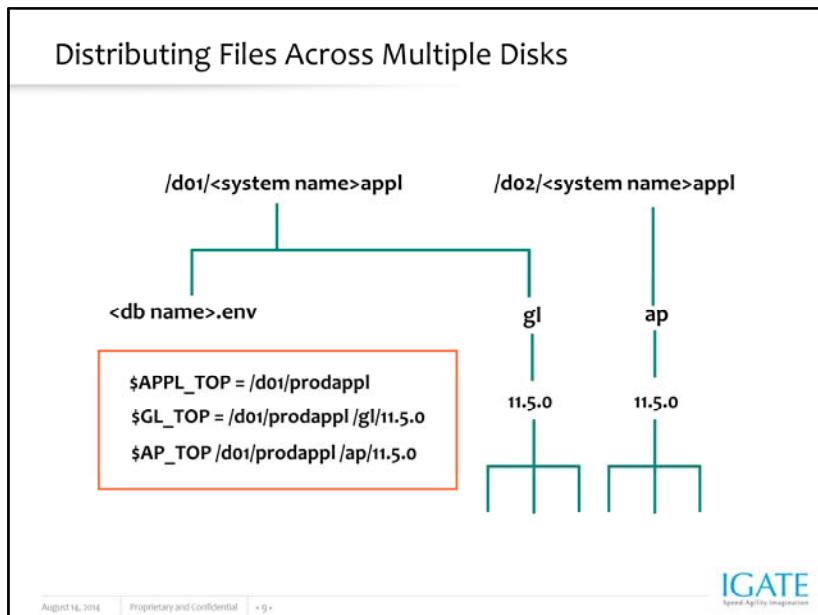
With Release 11i all Globalizations products (known as Localizations in releases prior to 11i) are installed in both the file system and the database. They have subdirectories under APPL_TOP similar to other Oracle Applications products.

Globalizations are Oracle Applications components that provide additional features for processing in a particular country or region. For example, Oracle Applications may extend the payment processing features of Oracle Payables to provide a feature needed for banks in France. Or Globalizations may be created to meet specific government requirements.

A Globalizations product may require additional:

- Forms
- Reports
- Seed data in the base product tables
- Database tables or other database objects

If you require the use of additional Globalizations products after the initial installation or upgrade, they can be licensed through the License Manager.

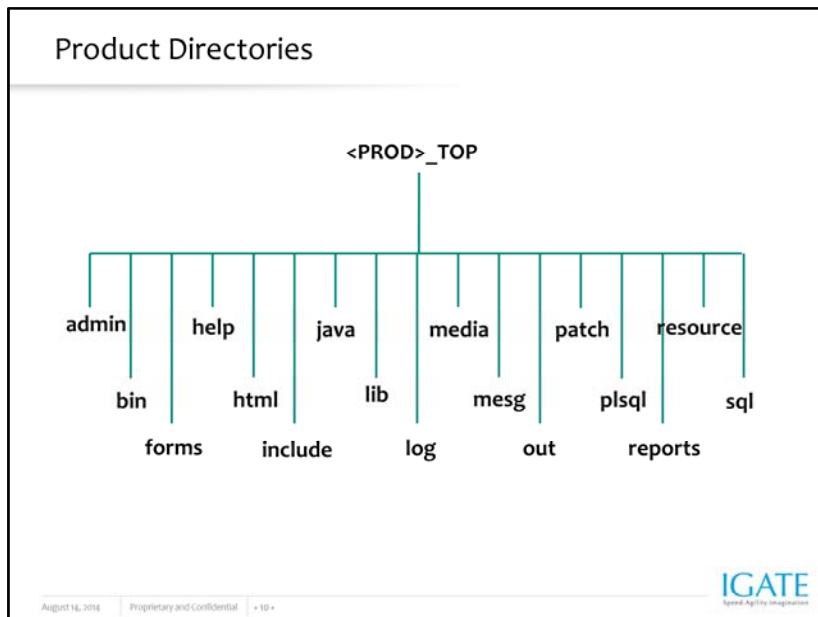


Distributing Files Across Multiple Disks

Oracle Applications files require a significant amount of space. All files may not fit on a single disk. You can distribute product directories across several disks if space is an issue.

When you install Oracle Applications, you can choose to put product files on disks other than the main disk. Rapid Install allows you to distribute the APPL_TOP across four mount points. If a product's directory is not located on the same disk, Rapid Install defines the full path to the directory in the <PROD>_TOP parameter in the <db name>.env file.

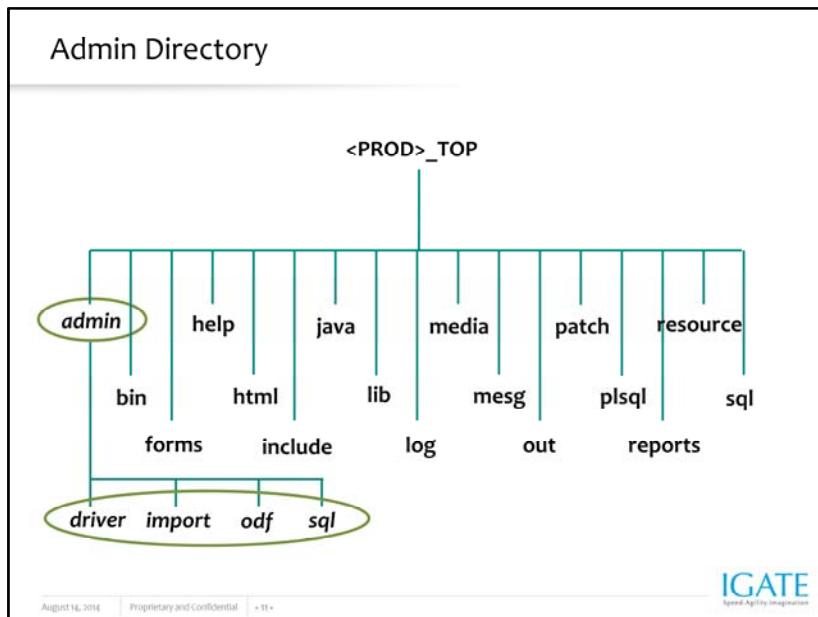
Technical note: Use Optimal Flexible Architecture (OFA) compliant mount point designations. OFA is a set of file naming and placement guidelines for Oracle software and databases. OFA helps users avoid problems by optimizing Oracle's relationship with its host operating system. One OFA rule is to name mount points using the format /x..xn..n, where x..x is a character string and n..n is a number (use zeros to pad fixed length numbers). The slide shows examples of this naming convention.



Product Directories

The Oracle Applications product directories contain many subdirectories that group the different files for a given product. A typical product directory has the subdirectories shown, however, there may be differences based upon configuration.

Note: Some products do not have all of the subdirectories you see on this slide.



Admin Directory

The admin subdirectory contains files used by AutoUpgrade to upgrade products to the current release.

The admin subdirectory has several subdirectories of its own:

- driver**: Contains the upgrade driver files (.drv).
- import**: Contains the import files used to upgrade seed data.
- odf**: Contains the object description files used to create and maintain tables, indexes, sequences, and views.
- sql**: Contains SQL scripts and PL/SQL scripts used to upgrade data and PL/SQL package creation scripts.

Bin Directory

The concurrent programs, other C language programs and operating system shell scripts for each product are stored in its respective bin directory. Of particular importance to Oracle Applications are the FND_TOP/bin and AD_TOP/bin directories. Some of the important programs in these directories include:

- f60webmx: the Applications Forms processor (in FND_TOP/bin)
- FNDLIBR: the concurrent manager (in FND_TOP/bin)

Forms Directory

Oracle Forms files include portable source files (.fmb files) and generated runtime files (.fmx files). Forms files are generated by converting the .fmb source file to .fmx runtime files. The forms directory contains Oracle Forms runtime files. The source files are stored in AU_TOP/forms so that runtime files can be generated more easily.

A subdirectory exists for the language(s) installed. This subdirectory is named according to the language.

for example:

- US for American English forms
- D for German forms
- F for French forms.

Help Directory

The help directory contains online help source files. These files are imported into the database to optimize the performance of online help.

HTML Directory

The html subdirectory contains HTML, Javascript, and Java Server Page files used by various products.

Include Directory

The include directory contains header (.h) files for custom development.

Java Directory

This directory is used to store Java files. During the installation or the upgrade, the files are copied to the directories identified in the JAVA_TOP environment variable (these directories are defined in the environment file created during the installation process). This is done to optimize processing.

Lib Directory

At some time, you may need to relink Oracle Applications programs, for example if you upgrade the Oracle8i server. The lib subdirectory contains files pertinent to the process of relinking Oracle Applications programs:

object files (.o files): There is one for each C program to relink.

library file (.a file): Is the compiled C code common to that product's programs.

Log and Out Directories

When the concurrent managers run Oracle Applications reports or data update programs, they write output files as well as diagnostic log files and temporary files to directories defined during the installation process. There are two methods for storing log and output files:

The log directory holds concurrent log files from each concurrent request. The concurrent manager log files are stored in FND_TOP/log.

The out directory holds the concurrent report output files.

Note: The log and out directories should be monitored for disk space usage and purged periodically.

Media Directory

The Oracle Applications Forms client applets display text and graphics in the form of .gif files. The media directory contains all product specific .gif files.

Mesg Directory

Oracle Applications forms display messages at the bottom of the screen and in pop-up boxes. Oracle Applications concurrent programs also print messages in their log and output files. Messages may be translated into different languages. Translated messages are stored in message files separate from forms and programs.

Patch Directory

Any updates to Oracle Applications data or data model use a directory named patch to store the patch files. Patch files are grouped by release within the following subdirectories:

driver: contains the driver files (.drv). Typically named d<patchnum>.drv, where <patchnum> is the patch number.

sql: contains sql (.sql) and PL/SQL (.pls) scripts used to patch the database.

odf: contains object description files (.odf) to patch the data model.

import: contains lct, ldt and slt files to update the seed data through loaders such as FNDLOAD and AKLOAD.

PL/SQL and Resource Directories

These directories are used for unloading PL/SQL libraries used by Oracle Applications reports and forms.

The files in the plsql subdirectory (.pll files) are used by Oracle Reports.

The files in the resource subdirectory (.pll and .plx files) are used by Oracle Forms.

Resource libraries that require language translation are stored in their own language-specific directory under the resource directory.

After these files are unloaded, they are copied to equivalent subdirectories under the AU_TOP directory.

Note that not all products have PL/SQL libraries.

Reports Directory

This directory contains the reports files for this product. For each report there is a portable binary .rdf file.

The AD Administration utility is used to regenerate reports. Generation of reports is usually recommended so the PL/SQL is optimally compiled for the platform.

Reports are stored in their own language-specific directory under the reports directory.

Sql Directory

There are many SQL scripts used by Oracle Applications for concurrent processing. These scripts typically produce reports or perform concurrent processing and are stored as .sql files in this subdirectory.

File Types and Extensions

Extension	Description
.a	Library files C code
.c	C source
.ctl	DataMerge control
.dat	DataMerge import/export
.drv	Driver
.env	UNIX environment
.exp	DataMerge export
.fmb	Binary forms

August 16, 2014

Proprietary and Confidential

+ 15 +



File Types and Extensions

These tables describe some of the file types and file extensions in the Oracle Applications file system:

- .a: library files
- .c: C source files
- .ctl: DataMerge control files
- .dat: DataMerge import/export files
- .drv: driver files (for AutoUpgrade and AutoPatch)
- .env: environment files
- .exp: DataMerge export files
- .fmb: binary Forms files

File Types and Extensions

Extension	Description
.fmx	Executable forms
.h	C header
.jar	Java archive
.lc	C source to be archived
.lct	Data loader control
.ldt	Data loader datafile
.log	Concurrent request log
.lpc	PRO*C source to be archived

August 16, 2014

Proprietary and Confidential

+ 16 +



File Types and Extensions (cont.)

- .fmx: Forms executable files
- .h: C header files
- .jar: Java Archive files
- .lc: C source to be archived
- .lct: Data loader control files
- .ldt: Data loader datafiles
- .log: Concurrent request log files
- .lpc: Pro*C source to be archived

File Types and Extensions

Extension	Description
.msb	Binary message
.msg	Readable message
.o	C object module
.odf	Object description
.out	Concurrent request output
.plb	PL/SQL package body
.pll	PL/SQL shared library (reports)
.pls	PL/SQL package specs

File Types and Extensions (cont.)

- .msb: binary message files
- .msg: readable message files
- .o: C object module
- .odf: object description files
- .out: Concurrent request output files
- .plb: PL/SQL package body files
- .pll: PL/SQL shared library files (for reports)
- .pls: PL/SQL package specification files

File Types and Extensions

Extension	Description
.rdf	Oracle Reports
.req	Log of concurrent request
.rex	Oracle Reports exec.
.sql	SQL*Plus scripts

August 14, 2014 Proprietary and Confidential + 18 -

IGATE
Speed. Agility. Imagination.

File Types and Extensions (cont.)

.rdf: Oracle Reports files

.req: concurrent request log files

.rex: Oracle Reports executable files

.sql: SQL*Plus scripts

Language Files

➤ Language Files

- When you install Oracle Applications in a language other than American English, each product tree includes directories that use the NLS language code
- These directories hold translated data, forms, message, and reports files

August 16, 2014

Proprietary and Confidential

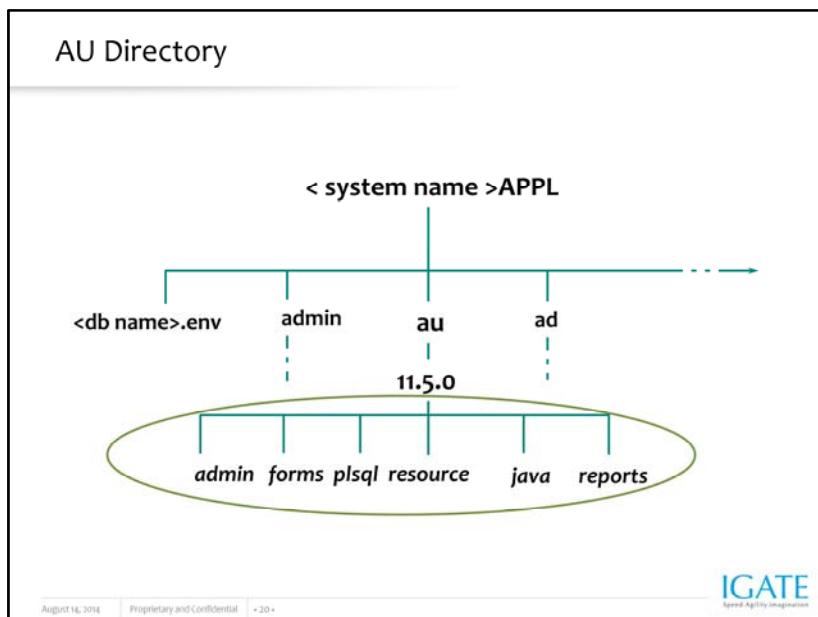
+ 19 +



Language Files

For example, the language directory named D designates German. The data loader files in the D subdirectory of admin contain the German translation of the product seed data. The D subdirectory of reports holds Oracle Reports files translated into German.

The US subdirectory in the forms directory holds Oracle Forms forms in American English. The D directory in the forms directory holds the same forms translated into German.



AU Directory

AU stands for Application Utilities. The AU_TOP directory contains product files that are consolidated in a single location for optimal processing. These files include:

PL/SQL libraries used by Oracle Reports, in the plsql subdirectory.

PL/SQL libraries used by Oracle Forms, in the resource subdirectory.

Oracle Forms source files, in the forms subdirectory.

A copy of all Java files used by JInitiator when regenerating the desktop client jar files, in the java subdirectory.

Certain reports needed by Discoverer or BIS, in the reports subdirectory.

Note: The public copy of all Java files are stored in JAVA_TOP.

Demo

- Navigate through the file system and review the contents of key Oracle Applications directories.



Summary

- **In this lesson, you should have learned how to do the following:**
 - Describe the Oracle Applications file system.
 - Describe the product subdirectory structure.
 - Describe the file types available on the file system.
 - Identify the APPL_TOP, COMN_TOP, ORA_TOP, and DATA_TOP directories.



Review Questions

- Question 1: What is the purpose of the APPL_TOP directory?
- Question 2: Identify and describe the directories present in CUSTOM_TOP.



ERP- Oracle Apps

Lesson 4: Multiple Organization in Oracle Applications

Lesson Objectives

- To understand the following
 - Multi-org
 - Types of organization
 - Selling, shipping ,purchasing and receiving
 - Identify the setup steps and considerations that are required for Multi-Org setup.



Multi - Org

- MultiOrg feature enables you to model a global enterprise with one installation of the Oracle Applications products.

August 14, 2014

Proprietary and Confidential

+ 3 +



Multi-org

MultiOrg feature enables you to model a global enterprise with one installation of the Oracle Applications products. This greatly reduces maintenance and setup effort, and allows more global access to some shared data. Prior to the Multiple Organization Support feature, customers could secure data by implementing separate installations of Oracle Applications. In particular, separate installations of the sub-ledger products (Oracle Order Entry, Oracle Payables, Oracle Purchasing, Oracle Receivables) were required. The Multiple Organization Support feature, which is supported by all Oracle Applications products, provides a more pragmatic solution to data security by allowing customers to secure data by business unit within one installation of all Oracle Applications.

Multi-Org is an Oracle Financials feature that lets you identify specific data and financial transactions as belonging to a single organization - classified as an 'Operating Unit' within your enterprise. This is essential for large customers with multiple lines of business or divisions where you want to secure access to information and simplify processing and reporting. In many ways Operating Unit security on data and processes is similar to the Business Group capability of Oracle HRMS and this has led to a lot of the confusion between the two.

With Multi-Org you can define an owning Operating Unit and associate this with a user profile option - MO:Operating Unit. When you define data or execute a business process then the system will automatically attach the id of your Operating Unit organization to the data or transaction. When you query data then the system will use the Operating Unit organization id from your user profile to filter the query.

Multi-org (Contd...)

This contrasts with the Business Group security model in HRMS where you can secure data and processes by id of the Business Group. However, in HRMS the Business Group id is closely related to the country of operation, and not simply a division or line of business. While there are similarities between the two, it is important to realize that these are distinct and different mechanisms for two different functional areas. This is especially true for any global implementation of HRMS where the data for each country exists within a single business group but Operating Units exist across many countries, and business groups.

Important Note: You can classify any organization in any business group as an Operating Unit and you can use this Operating Unit organization id to secure data in any business group.

WHY IS MULTI-ORG STRONGLY RECOMMENDED?

Multi-Org is strongly recommended in Release 11i in order to prepare for performance improvements and Multi-Org Access Control.

Performance Improvement

The Multi-Org functionality is being improved to take advantage of the new core database features of Oracle Release 9i, which improves the overall performance of the E-Business Suite. In future releases, the existing Multi- Org view layer will be replaced with the Oracle 9i Virtual Private Database technology.

Multi-Org Access Control

Multi-Org Access Control is an upcoming feature of the E-Business Suite that will allow a user to process and report on data that resides in an unlimited number of Operating Units from within a single applications responsibility.

Multi - Org (Contd...)

➤ Major features

- Multiple Organizations in a Single Installation
- Secure Access
- Sell And Ship Products From Different Legal Entities
- Receive Goods Into Any Inventory Organization
- Automatic Accounting for Internal Requisitions
- Multiple Organizations Reporting
- can support multiple organizations running any Oracle Applications product with a single installation

Multi - Org (Contd...)

➤ Database Architecture

- The Multiple Organization Support feature uses native database features introduced in Oracle 7.3 to build a security layer on top of a single installation of Oracle Applications. This layer of security provides the necessary data partitioning, while at the same time minimizes the number of potentially destabilizing changes to the application code itself. The security layer is provided using database views which allow access to the partitioned data without any changes to the applications code.

Multi - Org (Contd...)

➤ Oracle Database Schemas

- Beginning with Release 10.6, there is one Applications All Objects Oracle Schema, referred to as the APPS schema. This schema is maintained by AutoInstall. The schema contains synonyms to all tables and sequences as well as all server-side code (stored procedures, views, and database triggers).

August 14, 2014

Proprietary and Confidential

+ 7 +



Oracle Database Schemas

Beginning with Release 10.6, there is one Applications All Objects Oracle Schema, referred to as the APPS schema. This schema is maintained by AutoInstall.

The schema contains synonyms to all tables and sequences as well as all server-side code (stored procedures, views, and database triggers).

Data partitioning is performed by database views. These views reside in the APPS Oracle schema and derive the appropriate operating unit context from an RDBMS variable introduced in Release 10.7.

A user logging into Oracle Applications then, in effect, sees data that has been filtered through the Multiple Organization Support view layer. This way, a particular user is sure to see only the data relevant to his/her Oracle responsibility.

The Multi-Org views reside in the APPS schema, which has access to the complete E-Business Suite data model, and contains synonyms to all tables and sequences, stored procedures, views, and database triggers. All applications code runs against the APPS schema.

The Multi-Org views are based on the Multi-Org tables that are partitioned by Operating Unit (ORG_ID). These tables reside in the product schema (for example, AP). A Multi-Org table is named with _ALL suffix.

Oracle Database Schemas (Contd...)

- In the APPS schema, the corresponding Multi-Org view is created with the same table name without the _ALL suffix. The Multi-Org views derive the Operating Unit context from a global database variable "CLIENT_INFO" included in the view definition.
- The CLIENT_INFO variable is a global database variable.
- Upon session initialization, AOL initialization (FND_GLOBAL.initialize) populates the org context in the CLIENT_INFO global database variable, with the value from the profile option "MO: Operating Unit". When you access a form from Applications, the FND_GLOBAL.initialize() API sets the org context in the CLIENT_INFO variable.
- This API accepts the user_id, responsibility_id, and resp_appl_id for the session as input parameters, and performs the following activities:
 1. Reads the profiles, and caches the values.
 2. Sets the org context, and the MRC context in the CLIENT_INFO variable
 3. Sets the Security Group context.
 4. Executes product specific, and custom initialization

Multi - Org: Types of Organizations

- Six Types of Organizations
- Set of Books
- Business Group
- Legal Entity
- Balancing Entity
- Operating Unit
- Inventory Organization

Multi - Org: Types of Organizations (Contd...)

➤ Set of Books

- A financial reporting entity that uses a particular chart of accounts, functional currency, and accounting calendar

➤ Business Group

- Represents the highest level in the organization structure, such as the consolidated enterprise, a major division, or an operation company

➤ Legal Entity

- A legal company for which you prepare fiscal or tax reports.

August 14, 2016

Proprietary and Confidential

+ 10 +



Business group

A business group is the largest organizational unit you define to represent your enterprise as an employer. A business group may correspond to a company or corporation, or in large enterprises, to a holding or parent company or corporation.

Oracle Human Resources includes a predefined organization named Setup Business Group. If you plan to use only one business group, we recommend that you modify the definition of this predefined business group rather than defining a new one.

A business group is a special classification of an organization, so you also need to specify its location and organization type and identify it as an internal organization. It is also essential to select the correct legislation code for a business group for correct functioning of Oracle Human Resources. You cannot change the legislation code after entering employees in a business group.

Employees, organizations, and other entities are partitioned by business group. If you set up more than one business group, your data will be partitioned accordingly. In addition, classifying an organization as a business group is not reversible. Be sure to plan your business group setup carefully.

Multi - Org: Types of Organizations (Contd...)

➤ **Balancing Entity**

- Represents an accounting entity for which you prepare financial statements

➤ **Operating Unit**

- Uses Oracle Cash Management, Order Management and Shipping Execution, Oracle Payables, Oracle Purchasing, and Oracle Receivables

➤ **Inventory Organization**

- An organization for which you track inventory transactions and balances, and/or an organization that manufactures or distributes products.

Multi - Org: Types of Organizations (Contd...)

- **HR Organization**
 - Represent the basic work structure of any enterprise
- **Organizations in Oracle Projects**
 - To define organization hierarchies to reflect your company's organizations structure
- **Asset Organizations**
 - Allows you to perform asset-related activities for a specific Oracle Assets corporate depreciation book.

Multi - Org Organization Model

- With Oracle Applications accounting, distribution, and materials management functions, you define the relationships among inventory organizations, operating units, legal entities, and sets of books to create a multilevel company structure, as shown in the figure below

August 14, 2014

Proprietary and Confidential

+ 13 +



Oracle Applications organization models

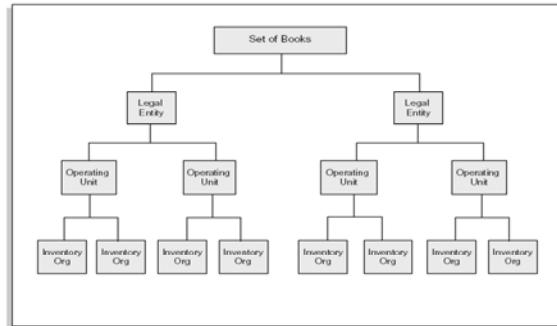
Define organizations and the relationships among them in arbitrarily complex enterprises

Dictates how transactions flow through different organizations and how those organizations interact with each other

Satisfies basic business needs

Complex enterprise has several organization structures

Human Resources Organization Model



August 14, 2014

Proprietary and Confidential

+ 14 +

IGATE
Speed. Agility. Imagination.

Responsibility Determines Operating Unit

Your responsibility determines the operating unit that you access when you use Oracle Applications. When you use Oracle Payables, Receivables, Order Entry, Purchasing, Projects, and Sales Compensation you see information that is relevant to your operating unit. All transactions that you create are automatically assigned to your operating unit.

Organization Model

- Legal Entities Post to a Set of Books
- Operating Units Are Part of a Legal Entity
- Inventory Organizations are Part of an Operating Unit
- Inventory Organization Determines Items Available to Order Management
- Inventory Organization Determines Items Available to Purchasing

Controlling Secure Access

- Data Security
- Inventory Organization Security by Responsibility
- Responsibility Determines Operating Unit

Sell And Ship Products From Different Legal Entities

- Sell from One Legal Entity, Ship from Another Legal Entity
- Use the Choose Organization Window to Choose Inventory Organization

Intercompany Accounting

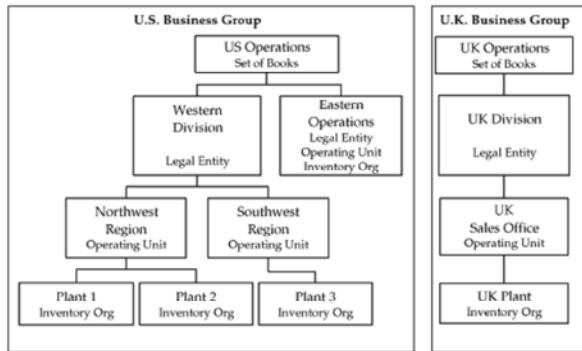
- Automatic Intercompany Sales Recognition
- Posting Intercompany Invoices to Different Accounts
- Report at the Legal Entity or Set of Books Level
- Multiple Organizations in European Implementations

Overview

- Developing organizational structure
- Defining organizations and relationships
- Implementing Application products
- Adding a new operating unit
- Using Multiple Organizations Validation Report

Steps for setting up a Multiple Organization Enterprise

➤ 1. Develop the Organization Structure



Steps

➤ 2. Define Sets of Books

Set of Books Name	SOB Short Name	Functional Currency	Accounting Flexfield Structure	Calendar
U.S. Operations	US_Op	USD	Standard	Standard
U.K. Operations	UK_Op	GBP	Standard	Standard

Steps (Contd...)

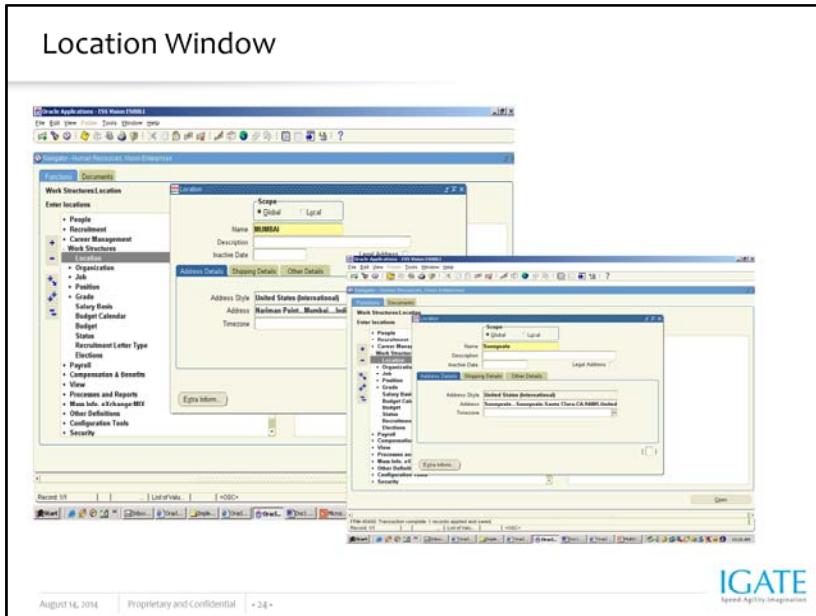
- 3. Define Locations
- 4. Define Business Groups (optional)
- 5. Associate Responsibilities with Business Groups
- 6. Define Organizations

Organization Name	Legal Entity?	Oper. Unit	Inv. Org?
Western Division	Yes		
Eastern Operations	Yes	Yes	Yes
U.K. Division	Yes		
Northwest Region		Yes	
Southwest Region		Yes	
U.K. Sales Office		Yes	
Plant 1			Yes
Plant 2			Yes
Plant 3			Yes
U.K. Plant			Yes

Steps (Contd...)

Classification	Information Type	Required Attributes
Business Group	Business group information if Human Resources is installed. Otherwise, None.	Short Name, Employee/Applicant Numbering, Key Flexfield Structures, Default Legislation Code, and Currency
GRE/Legal Entity	Legal Entity Accounting	Set of Books, Location
Operating Unit	Operating Unit Information	Legal Entity, Set of Books
Inventory Organization	Accounting Information	Set of Books, Legal Entity, Operating Unit

Sequence	Additional Organization Information	Required Information
1	Accounting information	Set of books, legal entity, operating unit
2	Receiving information	Three-character organization code, prefix serial (no dummy)
3	Inventory information	All required accounts



Location Window

You can enter addresses in the Location window.

Locations are shared across Business Groups in HRMS and with two other Oracle applications: Inventory and Purchasing. HRMS does not use some of the fields in the Location window. These fields are disabled for HRMS users.

1. Enter the name of the location, and a description if required.
2. Uncheck the Global check box if you want the location to only be available within the default Business Group of your current responsibility. Accept the default if you want the location to be a global location and therefore available to all Business Groups.

If you are setting up a global location, the location name must be unique across all Business Groups.

If you are setting up a location for one Business Group, the location name must be unique within that Business Group and all global locations, but does not have to be unique across all Business Groups.

Note: You cannot amend the Global check box once you have set up your location.

Select a national address style from the list. If a local address style exists for your country, it is displayed as the default. Otherwise, the international style is displayed.

Fill in information in the Shipping Details tabbed region.

Location Window (Contd...)

Note: Selecting the check boxes means that you are making the location Name a valid location in the list of values on a Purchasing document. For example, selecting Ship-To Site makes the location Name a valid choice in the list of values for the Ship-To field on a purchase order header. Note that if you define a default Ship-To or Bill-To Location in the Supplier-Purchasing region of the Financials Options window for your organization, that is the location that *defaults* onto your purchase orders. But if you wanted to change that default, the location Name that you define here and enable as a Ship-To or Bill-To site is *available* in the list of values for you to choose from in the Ship-To or Bill-To fields.

Contact: Optional contact name for the location Name.

Ship-To Location: Usually the same as the location Name. You could select a separate, previously defined Ship-To Location—for example, if you wanted to create a location Name, Office A, and specify Receiving Dock A as its Ship-To Location. Note, however, that once you specify a separate Ship-To Location, you can no longer enable the location Name as a Ship-To Site. In this example, Receiving Dock A is the ship-to site for Office A; therefore, Office A itself cannot also be the ship-to site.

Ship-To Site: Select this option to make the location Name a valid ship-to organization on a purchase order or requisition.

Receiving Site: Select this option to make the location a valid receiving Location when creating a receipt or receiving transaction.

Office Site: Select this option to indicate that this location Name is an office site, such as a field office.

Bill-To Site: Select this option to make the location Name a valid bill-to site. The Bill-To Site, which is used by Payables, is specified on a purchase order header.

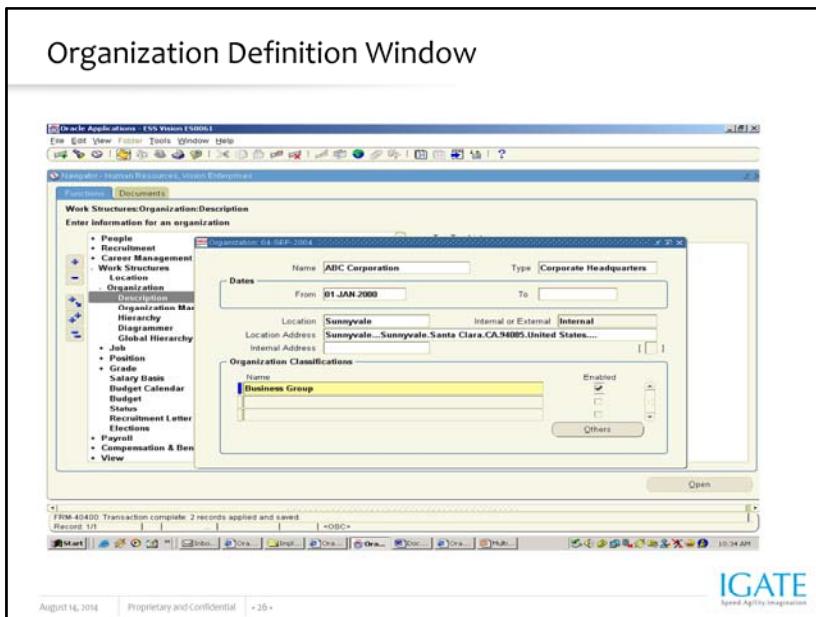
Internal Site: Select this option to make the location a valid internal ship-to location when creating an internal requisition.

4. Optionally fill in information in the Other Details tabbed region.

Inventory Organization: Select an inventory organization within which this location will be available in the list of values on a Purchasing document. By selecting no inventory organization, this location becomes available on Purchasing documents in all organizations.

Tax Name: Select a tax code to associate with the ship-to location. If you check the Ship-To Location option in the Purchasing tax default hierarchy, this is the tax code that defaults onto your requisitions and purchase orders, depending on your hierarchy.

EDI Location: If you use Electronic Data Interchange (EDI) to receive Advance Shipment Notices (ASNs) or ASNs with billing information (ASBNs), enter a defined location. This location should match the ship-to location specified on an ASN or ASBN.

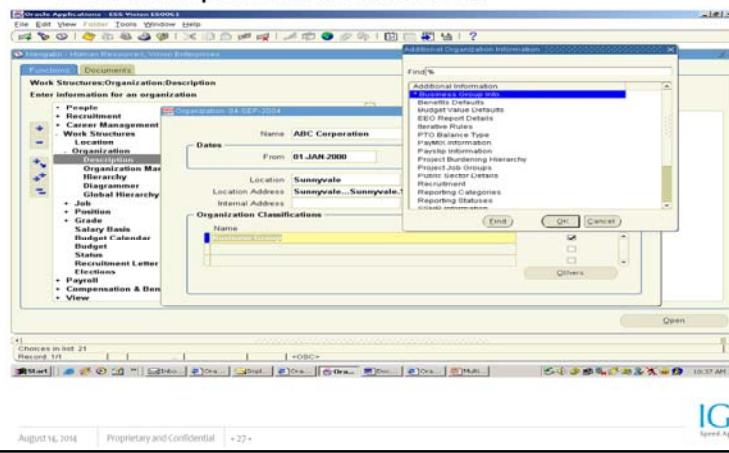


Responsibility : Human Resource, Vision Enterprise
Work Structure>Organization>Description

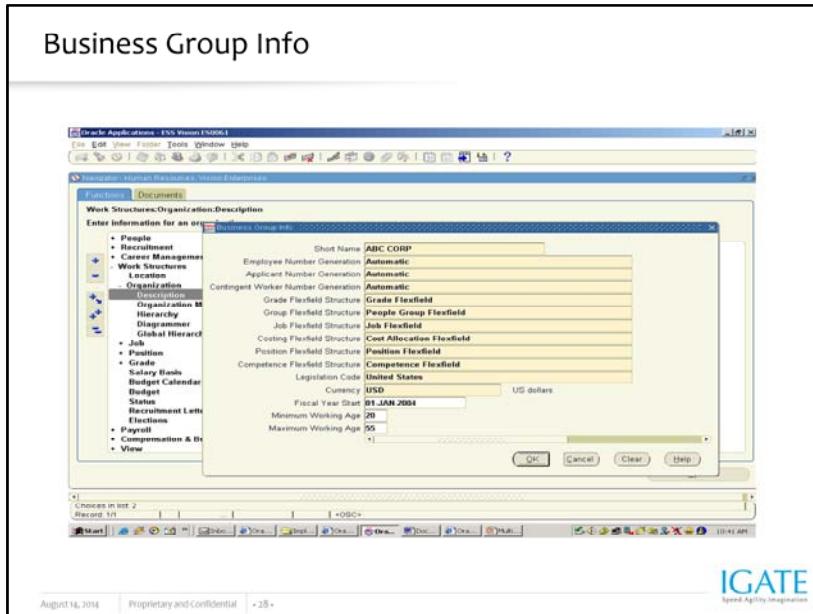
IGATE
Speed. Agility. Imagination

Organization Definition Window (Contd...)

- Select the 'Other' tab and for Additional Information select *Business Group Info from the list of values.



IGATE
Speed. Agility. Imagination



Business Group Info

To enter Business Group information:

1. In the Organization window, query the Business Group if it does not already appear there. In the Organization Classifications region, select Business Group, choose the Others button, and select Business Group Information.
2. Click in a field of the Additional Organization Information window to open the Business Group Information window.
3. You can enter a short name for the Business Group. This name is no longer used in Oracle HRMS It is provided for compatibility with earlier releases, where it appeared in the header line of each form. Select the method of creating identifying numbers for employees and applicants. The choices are:

Automatic number generation

Manual entry

Automatic use of the national identifier (for example, the social security number in the US, and the NI number in the UK). This option is available for employees only.

Attention: Once you save your method, you cannot later change to either of the automatic options. You can only change to manual entry.

Business Group Info (Contd...)

5. Select the names of the key flexfield structures you want to use in this Business Group.
6. Select the appropriate Legislation Code and default currency. The Legislation Code determines the startup data you can access and the contents of some legislation-specific windows.

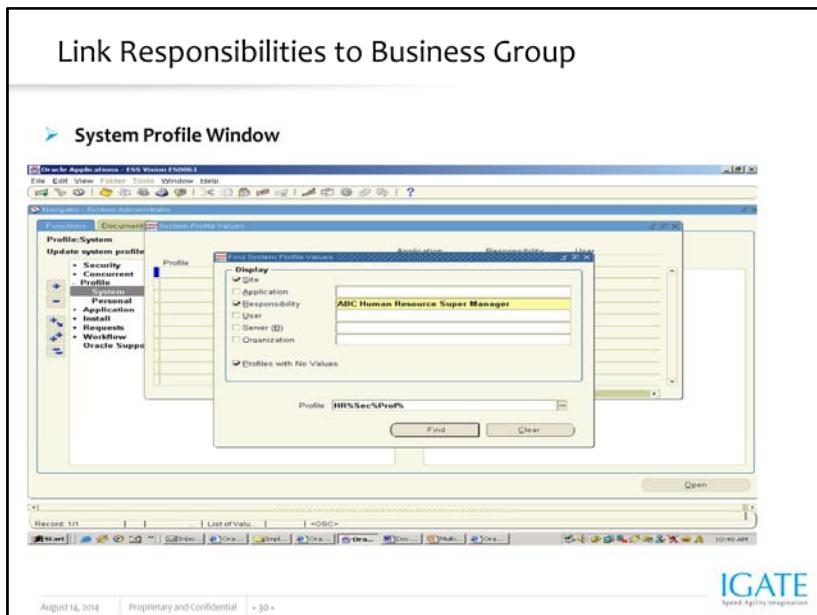
Attention: Selecting the correct legislation code is essential for the correct functioning of Oracle HRMS. You cannot change the legislation code after entering employees against the Business Group.

7. **US users:** To maintain fiscal year balances in Oracle Payroll if your fiscal year is different from the calendar year, enter the fiscal year start date.

South African users: You must enter the fiscal year start date for your Business Group, or you will not be able to define payrolls later.

8. You can enter a Minimum and Maximum Working Age for the Business Group. When you enter or hire employees, you receive a warning if the person's age is outside this range.

Also classify your Business Group as a HR Organization.



**Responsibility : System Administrator
Profile>System**

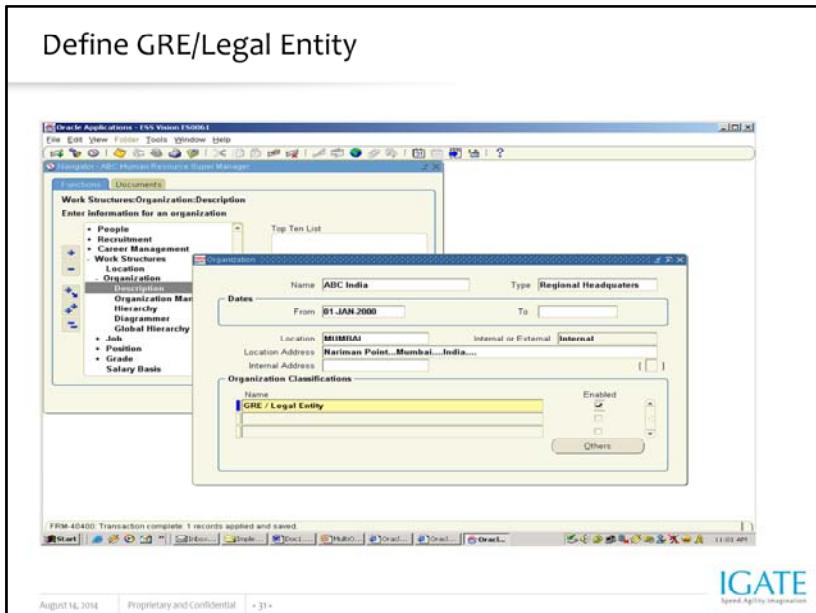
Responsibility : ABC Human Resource Super Manager, ABC India (Mumbai) Purchasing, ABC India (Mumbai) Payables, ABC India (Mumbai) Inventory, ABC India (Mumbai) Cash Management

Profile : HR Security Profile

Whenever a new Business Group is created, a Security Profile with identical name is automatically created.

If HR : Security Profile is set up, HR : Business Group is automatically defaulted with the same set-up, but not otherwise.

Set up for all responsibilities



Responsibility : ABC Human Resource Super Manager
Work Structure > Organization > Description

After defining one or more Business Groups for your enterprise, you set up one or more Government Reporting Entities (GREs) within each Business Group. The GRE is the organization that federal, state and local governments recognize as the employer who:

Issues pay to employees.

Withholds taxes from employees' pay and is liable for employer taxes and tax reporting.

Provides other government-mandated reporting, such as EEO-1, OSHA, and New Hire reports.

Each GRE has a unique 9-digit number (sometimes called the employer identification number or taxpayer identification number) issued by the IRS. Your enterprise may have just one such number, in which case your Business Group and GRE are the same organization.

Large enterprises may include a number of different divisions or companies each with its own identification number from the IRS, in which case you set up a GRE for each. You can associate several GREs together as a Tax Group, for which one of the GREs can serve as the common paymaster.

Legal Entity Accounting Information

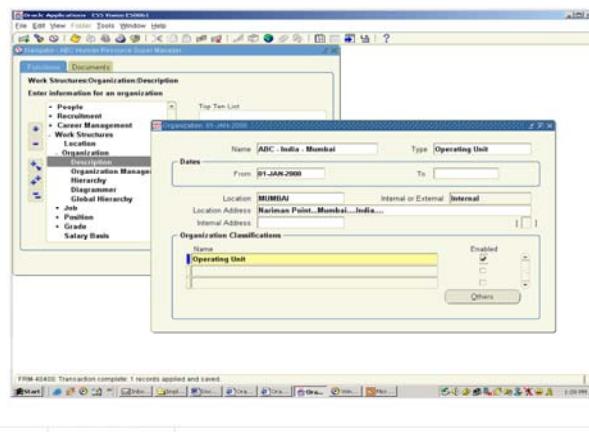
The screenshot displays several overlapping windows from the Oracle Applications interface:

- Main Window:** Shows a navigation bar with links like Home, People, Finance, General Ledger, Payroll, and Legal Entity Accounting.
- Central Workcenter:** A large window titled "Enter Information for an Organization". It includes sections for "Organization Classification" (with dropdowns for Name, Type, Location, and Internal Address), "Local Tax Rates", and "Financials".
- Sub-Windows:**
 - A modal window titled "Find/Select Organization Classification" with fields for Name, Type, and Location.
 - A window titled "Enter Information for an Organization" with tabs for "People", "Finance", "General Ledger", "Payroll", and "Legal Entity Accounting".
 - A window titled "Legal Entity Accounting" with fields for VAT Registration Number and other details.
- Bottom Bar:** Standard Oracle Applications navigation and search tools.

At the bottom of the screen, there is a watermark for IGATE.

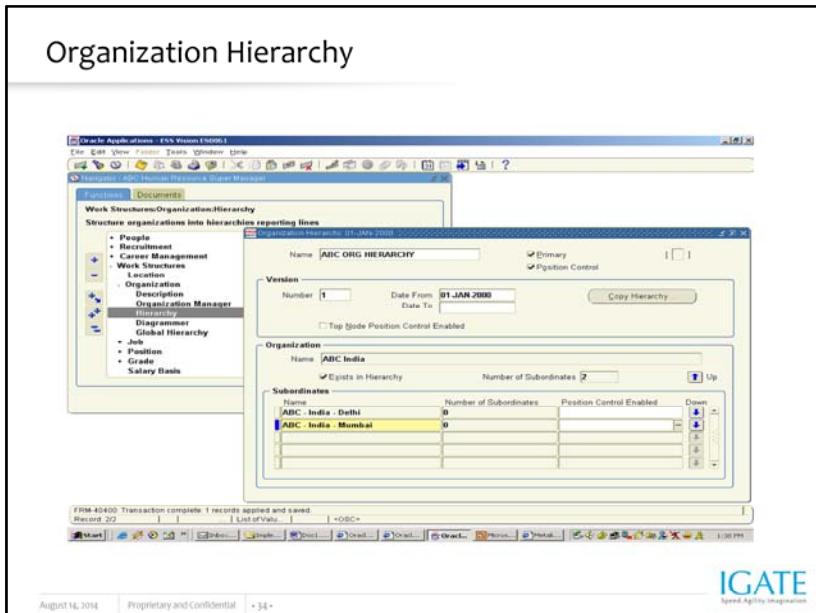
Define Operating Unit

➤ ABC India - Mumbai ABC India - Delhi



An organization that partitions data for sub ledger products (AP, AR, PO, OE).

IGATE
Speed. Agility. Imagination.



Responsibility:

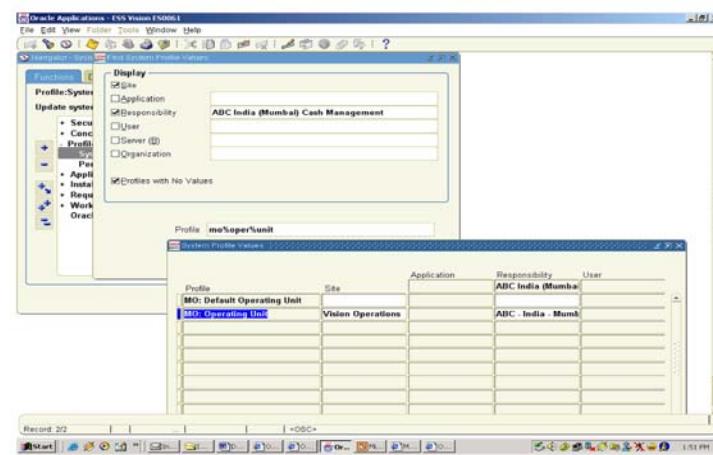
Work Structure > Organization > Hierarchy

Creating Organization Hierarchies

Create organization hierarchies to show reporting lines and other hierarchical relationships. If you want to include organizations from a single Business Group, use the Organization Hierarchy window, alternatively, use the Global Organization Hierarchy window to include organizations from any Business Group. Always define hierarchies from the top organization down.

You must define the top organization in the hierarchy, and at least one organization subordinate to it.

Set up MO: Operating Unit



IGATE
Speed. Agility. Imagination

Set up this profile option for all responsibilities which are associated with Applications that work at the Operating Unit level, e.g. Payables, Purchasing, Inventory, Cash Management etc.

Two profile options are introduced for Multi-Org Access Control:

MO: Security Profile

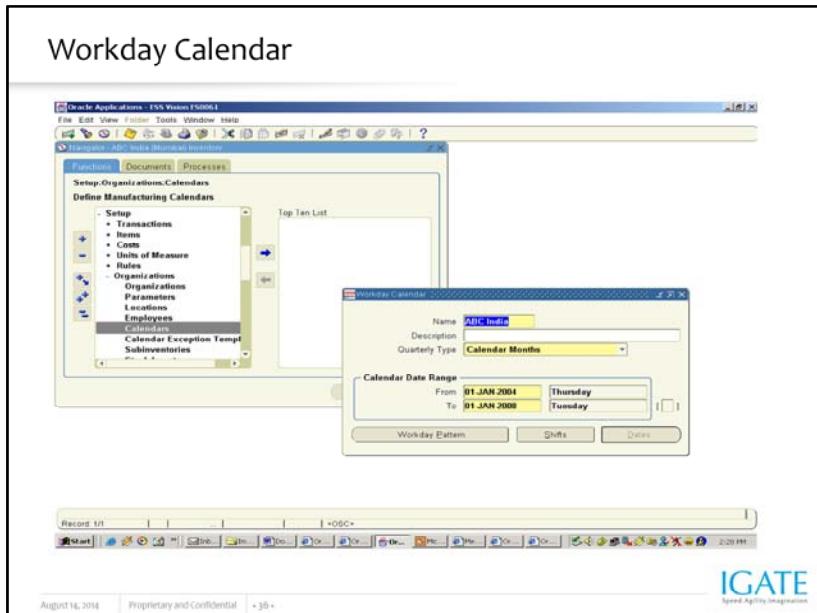
MO: Default Operating Unit

MO: Security Profile

Allows you to specify the Operating Units that a User has access to. The profile option can be set at Site, Responsibility, or User level. If you do not use the Multi-Org Access Control inquiry-only feature for JTF Territories, you should not set this profile option at any level. For JTF Territories, you may optionally set this profile option at the Responsibility level.

MO: Default Operating Unit

Allows you to specify which Operating Unit can be used as default for transaction entry, and reporting. This profile option can be set at Site, Responsibility, or User level. This profile option is reserved for the upcoming Multi-Org Access Control feature. Therefore, you should not set this profile option at any level.



Responsibility :
Setup > Organizations > Calendars

Creating a Workday Calendar

A workday calendar consists of a start date, and end date, and on and off days that follow one or more defined workday patterns. A workday calendar can also include shift information and exception dates, such as holidays or scheduled down time.

Use a workday calendar for forecasting and planning material requirements, use shift information for job scheduling, and capacity analysis. Workday exceptions identify deviations to the workday calendar. Shift exceptions identify deviations for a shift.

Attention: Shift exceptions take precedence over workday exceptions. (This is only relevant if a workday exception and a shift exception overlap.)

To assign workday exceptions to a workday calendar or calendar shift, either select them intuitively from the Calendar Dates (or Shift Calendar Dates) window, or apply them from an existing template, calendar, or shift.

To create a workday calendar:

1. Navigate to the Workday Calendar window.
2. Enter a name for the workday calendar.
3. Select a quarterly calendar type:

4/4/5 Week Pattern: Two four-week periods followed by one five-week period.

5/4/4 Week Pattern: One five-week period followed by two four-week periods.

Calendar Months: Twelve periods per year based on calendar months.

13 Periods: Thirteen four-week periods per year.

Note: If you use Oracle Master Scheduling/MRP, choose the 4/4/5 Weekly Pattern or the 5/4/4 Weekly Pattern to report MRP information in weeks and months. Otherwise, if you choose the 13 Periods calendar or Calendar Months, you can only report MRP information in monthly buckets.

4. Enter a date range for the workday calendar. The default end date is four years from the start date.

Note: Days on and off are calculated based on the start date and the day of the week. For example, if you want a standard five day workweek to start on 01-JAN, you must enter the start date as the Monday before 01-JAN (unless 01-JAN falls on a Monday).

Note: Set profile option BOM: Week or Period Start Date, to determine the start date. This profile option works in conjunction with the Workday Calendar.

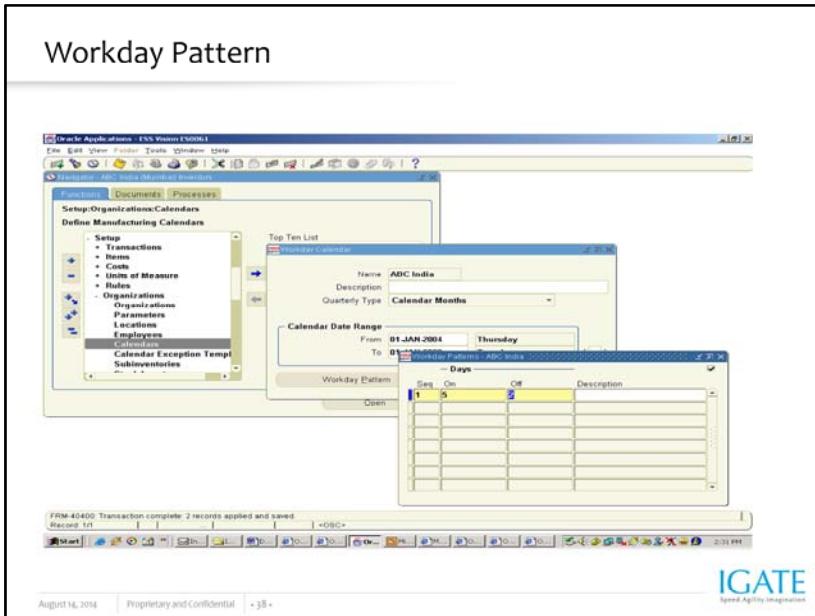
5. Continue creating the workday calendar by choosing one of the following buttons:

Workday Pattern: Set workdays on or off.

Shifts: Assign shifts to the workday calendar.

Dates: Review your work so far and its effect on the workday calendar. Do this before you build or rebuild the workday calendar you are now creating. (You can only do this after you have defined a workday pattern.)

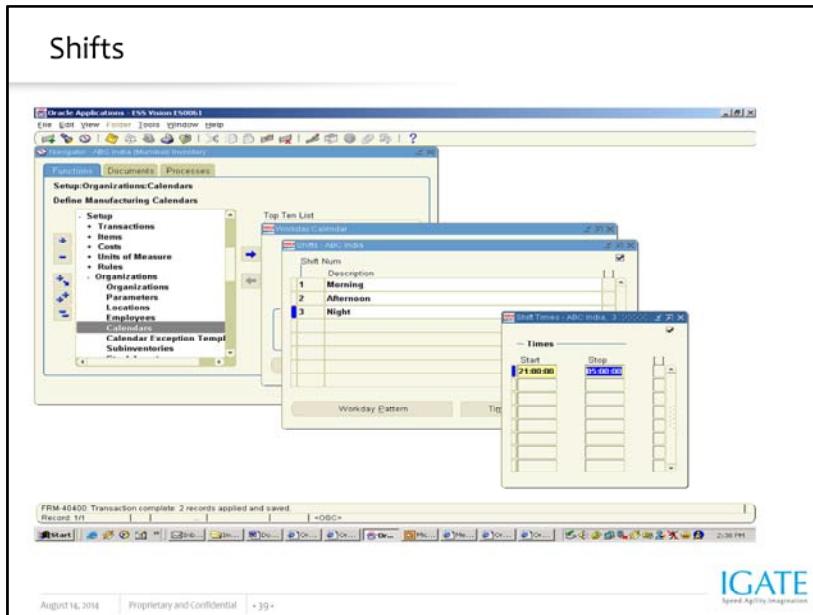
Once you have created a workday pattern, assigned shifts, assigned workday and shift exceptions, and reviewed your work, you must build the calendar and assign it to an organization.



Creating a Workday Pattern

To create a workday pattern:

1. Navigate to the Workday Patterns (or Shift Workday Patterns) window. You can do this by choosing the Workday Pattern button from either the Workday Calendar or Shifts window.
 2. Enter a sequence number in which the workday patterns are applied when the calendar is built.
 3. Enter the number for consecutive workdays on and off. For example, if you want Monday through Friday on and Saturday and Sunday off, enter 5 for Days On and 2 for Days Off.
 4. Enter a description for the sequence.
- Repeat the previous three steps for each workday pattern to create. All defined sequences are repeated for the duration of the workday calendar. If there are two sequences, sequence 1 will be in effect first, then sequence 2, then 1 again, and so on. For example, assume that sequence 1 is 5 on and 2 off, and sequence 2 is 4 on and 3 off. The calendar shows 5 on, 2 off, 4 on, and 3 off.
5. Save your work. Doing so ensures that you can view calendar dates.



Creating Shifts and Shift Times

To create shifts and shift times:

1. Navigate to the Shifts window. You can do this by choosing the Shifts button from the Workday Calendar window.
2. Enter a shift number and a description.
3. Choose the Times button to open the Shift Times window.
4. Enter the start and stop times for each shift.

You can enter multiple start and stop times, but the start and stop times for a shift cannot overlap.

If you create a shift without first creating a workday pattern, a default workday pattern of 5 days on and 2 days off is created. You can then update that workday pattern.

Select Tools > Build to build the calendar

Define Inventory Organization

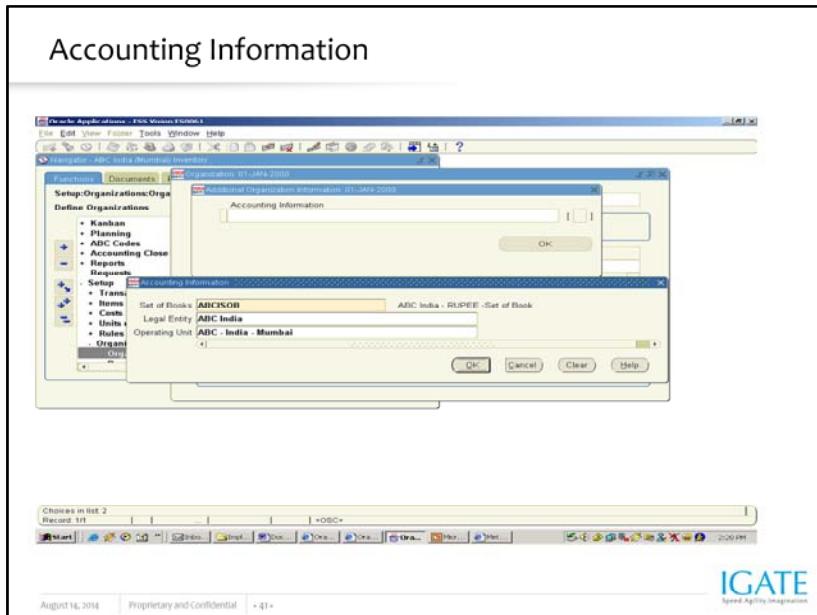
August 14, 2014

Proprietary and Confidential

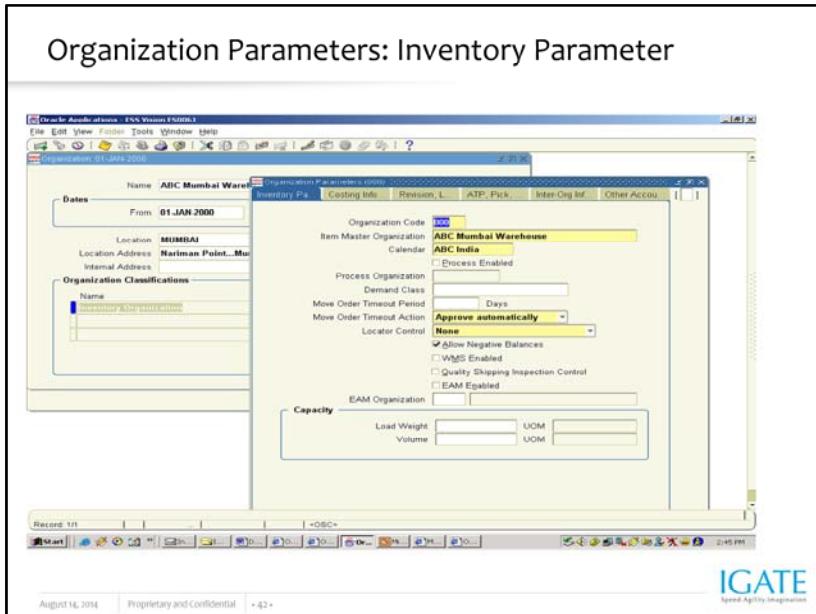
+ 40 +



This Organization is defined from Purchasing or Inventory Responsibility
Responsibility : ABC India (Mumbai) Inventory
Setup > Organizations > Organizations



Incorporate for the Inventory Organization Set of Book name, Legal Entity and Operating Unit.



Defining Default Inventory Parameters

To define inventory parameters:

1. Navigate to the Organization Parameters window.
 2. Select the Inventory Parameters tabbed region.
 3. Enter an organization code.
 4. Select an Item Master organization.
 5. Select a workday calendar. This is required when Oracle Master Scheduling/MRP and Oracle Supply Chain Planning is installed.
 6. Check the Process Enabled checkbox if the organization is a Process Manufacturing organization.
 7. Select a process organization to which you want to relate this inventory organization.
 8. Optionally, select a demand class.
- Demand classes segregate scheduled demand and production into groups, allowing you to track and consume those groups independently. Oracle Master Scheduling/MRP and Oracle Supply Chain Planning uses this demand class during forecast consumption, and shipment and production relief.
9. In the Move Order Timeout Period field, enter the number of days a move order requisition can wait for approval.

The workflow approval process sends a notification to the item planner when a move order requisition requires approval. After the first timeout period, if the recipient has not approved or rejected the order, a reminder notice is sent. After the second timeout period, the order is automatically approved or rejected, depending on whether you select *Approve automatically* or *Reject automatically* in the Move Order Timeout Action field. If you want to bypass the move order approval process and automatically approve move order requisitions, enter 0 days in the Move Order Timeout Period field and select *Approve automatically* in the Move Order Timeout Action field.

10. Select a move order timeout action:

Approve automatically: After the second timeout period, move order requisitions are automatically approved. Select this option and set the Move Order Timeout Period to 0 if you want to bypass the move order approval process and automatically approve move order requisitions.

Reject automatically: After the second timeout period, move order requisitions are automatically rejected.

11. Select a locator control option:

None: Inventory transactions within this organization do not require locator information.

Prespecified only: Inventory transactions within this organization require a valid, predefined locator for each item.

Dynamic entry allowed: Inventory transactions within this organization require a locator for each item. You can choose a valid, predefined locator, or define a locator dynamically at the time of transaction.

Determined at subinventory level: Inventory transactions use locator control information that you define at the subinventory level.

12. Indicate whether to allow negative balances.

Determines whether inventory transactions can drive the inventory balance of an item negative.

Attention: If insufficient quantity on hand exists in a supply subinventory to satisfy backflush demand, Oracle Work in Process forces the supply subinventory balance negative, ignoring this option setting.

13. Indicate whether this organization is an Oracle Warehouse Management enabled organization. You can use WMS features such as LPNs, task management, warehouse execution rules and cost groups in this organization.

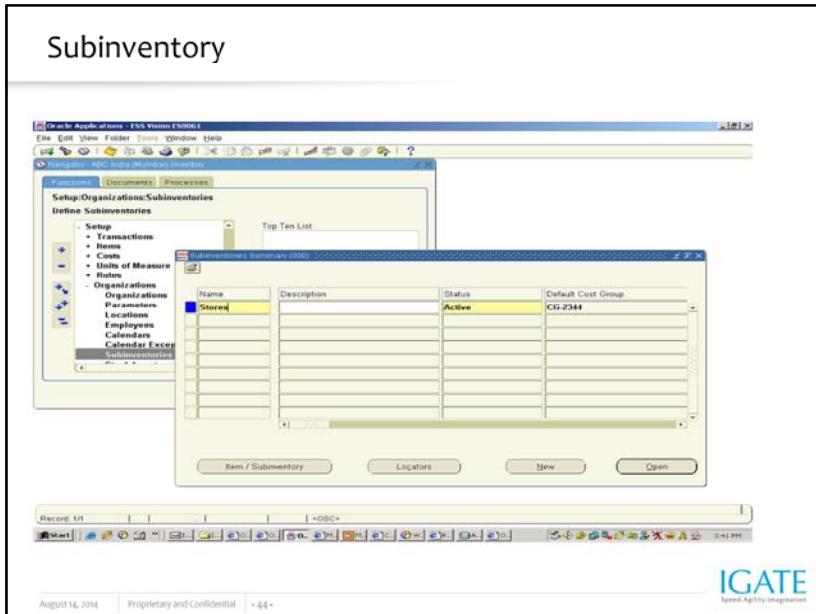
Locator control must be enabled in order to enable WMS. Once this has been enabled and transactions have been entered in this organization, this box cannot be unchecked.

14. Indicate whether this organization is enabled for Quality inspection.

15. Indicate whether this organization is an Oracle Enterprise Asset Management enabled Organization.

16. Enter a total load weight and unit of measure for this organization.

17. Enter a total volume and unit of measure for this organization.



Responsibility

Setup > Organizations > Subinventories

Defining Subinventories

Subinventories are unique physical or logical separations of material inventory, such as raw inventory, finished goods, or defective material. All material within an organization is held in a subinventory therefore, you must define at least one subinventory.

To define subinventory header information:

1. Navigate to the Subinventories Summary folder window.
2. Choose New. The Subinventories window appears.
3. Enter a unique alphanumeric name.
4. Indicate the material status of this subinventory, which controls the enabled transactions for all material in this subinventory. The status is not overridden by the status of any locator, lot or serial, within this subinventory. The statuses of those objects will be considered when determining transactions that are not enabled. This field is used if you have Oracle Warehouse Management installed.
5. Indicate the default cost group for this subinventory. If the cost group assignment rules fail to identify a cost group for newly received material, this cost group will be assigned. This cost group will remain with the material, even through subinventory transfers, until you perform a cost group change transaction. This feature is available if you have Oracle Warehouse Management installed, and you are working with a WMS enabled organization.

Summary

➤ **In this lesson we covered:**

- Multi-org Definition
- Major Features
- Types of organization
- Controlling access
- Selling, shipping ,purchasing and receiving
- Reporting at the legal entity of set of books level
- The setup steps and considerations that are required for Multi-Org setup.



Review Question

➤ **Question 1:** _____ is a financial reporting entity that uses a particular chart of accounts, functional currency, and accounting calendar.

- Business Group
- Set of books
- Legal Entity
- Operating unit



➤ **Question 2:** Operating Units are part of a

- Legal Entity
- Organization
- Set of Books
- None of the Above

Review Questions

- Question 1: What is the purpose of the APPL_TOP directory?
- Question 2: Identify and describe the directories present in CUSTOM_TOP.



ERP- Oracle Apps

Lesson 5: User profiles

August 14, 2014

Proprietary and Confidential

+ 1 +



Lesson Objectives

- Overview.
- Steps to create user profile.
- Setting Profile Option Values.
- Profile Levels.
- User Profile APIs
- Summary.
- Review Questions.



Overview of User Profiles

- A user profile is a set of changeable options that affects the way your applications run.
- Each profile option has an associated value
- Your user can change the value of profile options at any time
- AOL provides many predefined profile options that users can set
- Developers can also define their own profile options

August 14, 2014

Proprietary and Confidential

+ 3 +



When we buy a product, we expect some flexibility in using the product.

For example we do not want to buy a separate iron box for each type of cloth. We would like the same box to operate differently with each type of cloth, by using some switches/settings.

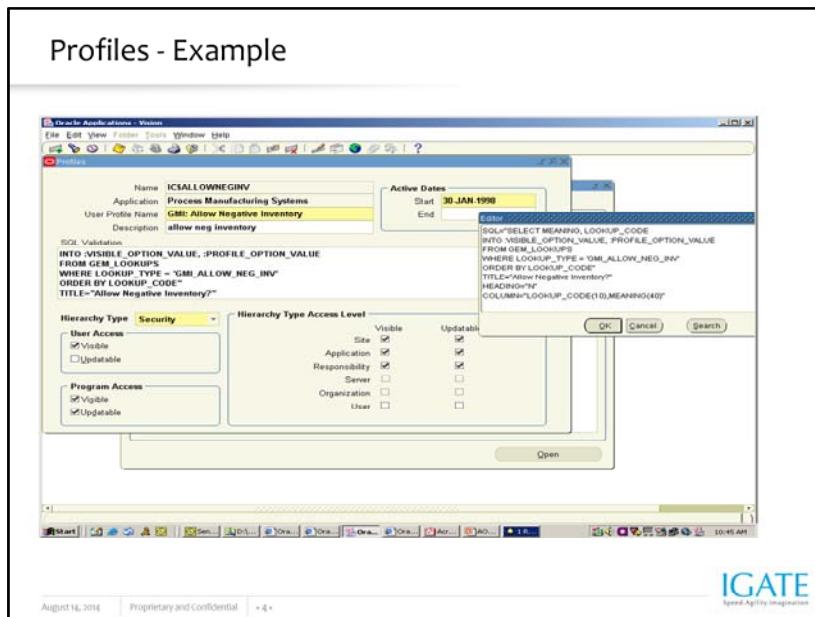
ERPs are designed to be used by various industries/customers. Each customer might want a specific functionality to work in a different way.

There might be some settings which are specific to a user or group of users. To address most of these requirements, Oracle Applications provides 'Profiles'.

A user profile is a set of changeable options that affects the way your applications run. Oracle Application Object Library establishes a value for each option in a user's profile when the user logs on or changes responsibility.

Users can change the value of profile options at any time. Oracle Application Object Library provides many predefined options that users can set to alter the user interface of applications to satisfy individual preferences.

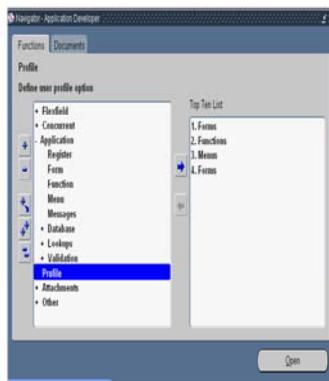
user profile options you define are indistinguishable from those Oracle Application Object Library provides.



Shows the entire user profile level options.

Steps to create a User Profile

- Switch to Application Developer
- Select Profile



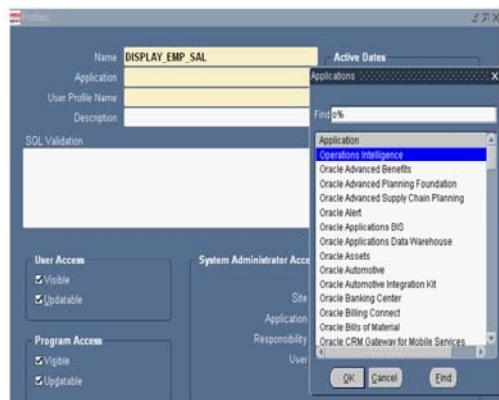
August 14, 2014 Proprietary and Confidential + 5 +

IGATE
Speed Agility Inspiration

In the 1st step switch to application developer and select the profile option.

Steps to create a User Profile (Contd.)

- Create a new profile
- Select the application
- Set values for the other options



IGATE
Agile. Agile by imagination

In this step create new user profile and also select the application name .The application name will come in lov choose in that which the user need.if the user wants to add description for reference he can add it.

You identify a profile option by application name and profile option name.

Name

The profile option name must be unique so that different profile options do not conflict. This is the name you use when you access your profile option using the Oracle Application Object Library profile option routines.

Application

Normally, you enter the name of the application you are building.

User Profile Name

This is the name your users see as their profile option, so it should be short but descriptive.

Description

Provide a better explanation of the content or purpose of this profile option. This description appears in a window with User Profile Name when a user or system administrator chooses profile options to set Values.

Steps to create a User Profile Contd.. Profile Option with LOV

If you want your profile option to provide a list of values (LOV) when the system administrator or user sets profile options, you must use the SQL Validation field.



August 14, 2014 Proprietary and Confidential + 2 +

IGATE
Speed. Agility. Innovation

Start Date/End Date

Enter the dates on which the profile option becomes active/inactive. The start date defaults to the current date, and if the end date is not entered, the option will be effective indefinitely. You cannot delete a user profile option, but you can disable it. Enter the current date if you want to disable the user profile option. If you wish to reactivate a disabled profile option, change the End Date to a date after the current date.

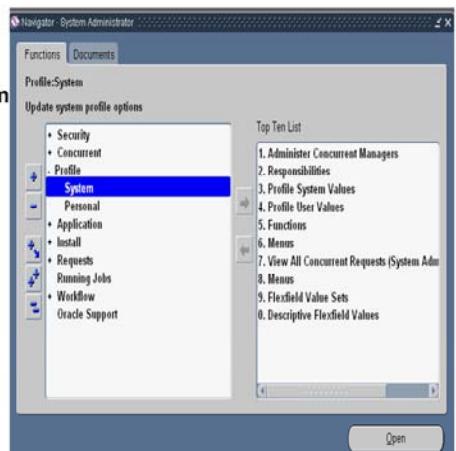
SQL Validation

If you want your profile option to provide a list of values (LOV) when the system administrator or user sets profile options, you must use A SELECT st. in the SQL Validation field.

A SELECT statement selects the rows to display in your LOV. In the SQL statement you can specify column aliases, use an INTO clause to put values into form fields, display database values without selecting them into form fields (by selecting values INTO NULL), and mix values to put into form fields with display only values in the same INTO clause.

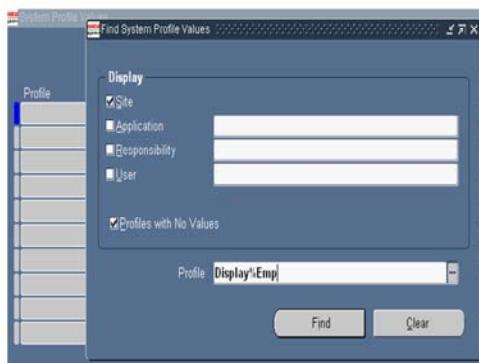
Steps to create a User Profile (Contd.)

- Switch to System Administrator
- Select Profile -> System



Steps to create a User Profile (Contd.)

- Query the profile just created in the Find System Profile Values
- The Site option is checked by default
- Click on Find



August 16, 2014 Proprietary and Confidential + 9 +

IGATE
Speed. Agility. Inspiration.

Query the profile just created in the Find System Profile Values
The Site option is checked by default
Click on Find

You can define options at one or more of four levels:

Site,
Application,
Responsibility
User.

For example, you have a site-level Printer option, an application-level Printer option, etc.

Site Level

Site is the lowest user profile level. Site-level option values affect the way all applications run at a given installation.

Application Level

Application is the user profile level immediately above Site. Application-level option values affect the way a particular application runs. For example, you might define a user profile option to determine which subset of the organization's data your end user sees.

Responsibility Level

Responsibility is the user profile level immediately above Application. Responsibility-level option values affect the way applications run for all users of a responsibility.

User Level

User is the highest user profile level and is immediately above Responsibility. User-level option values affect the way applications run for an application user.

For example, for a given end user, assume the Printer option is set only at the Site and Responsibility levels. When the end user logs on, the Printer option assumes the value set at the Responsibility level, since it is the highest-level setting for the option.

Steps to create a User Profile (Contd.)

- Enter Profile
Option values for
Site level

System Profile Values				
Profile	Site	Application	Responsibility	User
Display Employee Salary	N			

August 16, 2014

Proprietary and Confidential

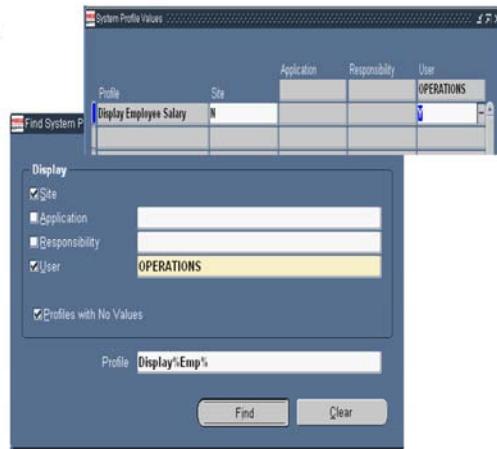
+ 11 +

IGATE
Speed. Agility. Imagination.

Enter Profile Option values for Site level which will take as N for site level.

Steps to create a User Profile (Contd.)

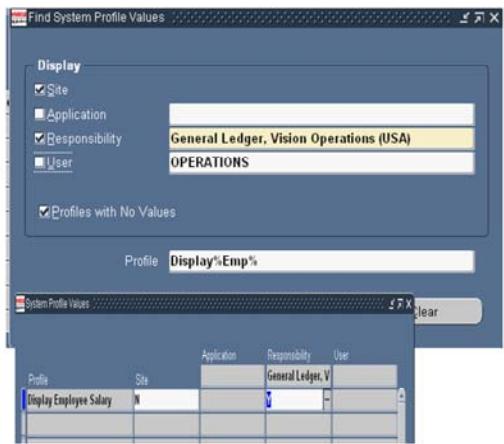
- If Profile Option values are to be set at User level, query on the profile created and select the User option and enter the name of the user as shown



If Profile Option values are to be set at User level, query on the profile created and select the User option and enter the name of the user as shown.

Steps to create a User Profile (Contd.)

- If Profile Option values are to be set at Responsibility level, query on the profile created and select the Responsibility option and enter the name of the Responsibility as shown



If Profile Option values are to be set at Responsibility level, query on the profile created and select the Responsibility option and enter the name of the Responsibility as shown .

Edit - Profiles

To view all profiles, select Edit -> Profiles

Profile Name	Default Value	User Value
DISPLAY_EMPCODE		Y
DISPLAY_MITI_OCCURS_FIELD		Y
DISPLAY_FREQUENCY	YES	YES
DISPLAY_FREQUENCY_36698		Y
DISPLAY_SAVIEMPCODE		Y
Display Employee Salary	Yes	Yes

IGATE
Agile. Agile by innovation.

To see all the profiles go to edit in that see preferences u will find the profile.

Profile Levels

- Site Level - Site is the lowest user profile level. Site-level option values affect the way all applications run at a given installation.
- Server Level -This is used when the system needs to determine the server on which the user's session is running. This is immediately above the site level in Server hierarchy.
- Organization Level - Organization refers to operating unit. All users within that organization will use the profile option value set once at the organization level. This is immediately above the site level in Organization hierarchy

Profile Levels

- Application Level - Application is the user profile level immediately above Site in security hierarchy. Application-level option values affect the way a particular application runs.
- Responsibility Level - Responsibility is the user profile level immediately above Application in Security hierarchy. This option values affect the way applications run for all users of a responsibility.
- User Level - User is the highest user profile level in all hierarchy types. This option values affect the way applications run for an application user.

Predefined Profile Levels

- Oracle Application Object Library also provides a set of profile options that you can access via the user profile routines.
- You can retrieve values for these profile options in your forms and programs.
EX: user_id, org_id
Refer to developers guide for complete list of the predefined profile options.

User Profile APIs

- You can use user profile routines to manipulate the option values stored in client and server user profile caches
- On the client, a single user profile cache is shared by multiple form sessions. Thus, when Form A and Form B are both running on a single client, any changes Form A makes to the client's user profile cache affect Form B's run-time environment, and vice versa
- On the server, each form session has its own user profile cache. Thus, even if Form A and Form B are running on the same client, they have separate server profile caches. Server profile values changed from Form A's session do not affect Form B's session, and vice versa

Any changes you make to profile option values using these routines affect only the run-time environment. The effect of these settings ends when the program ends, because the database session (which holds the profile cache) is terminated. To change the default profile option values stored in the database, you must use the User Profiles form.

User Profile APIs (Contd.)

```

FND_PROFILE.GET
procedure FND_PROFILE.GET
    (name IN varchar2,
     value OUT varchar2);
Example :
FND_PROFILE.GET
    ('USER_ID', user_id);

function FND_PROFILE.VALUE
    (name IN varchar2) return
    varchar2

```

The PL/SQL Editor window shows a trigger named 'WHEN-NEW-FORM-INSTANCE'. The code within the trigger handles the APP_STANDARD_EVENT('WHEN-NEW-FORM-INSTANCE') and checks if the profile option 'DISPLAY_EMP_SAL' is set to 'Y'. If so, it sets the app_item_property for the specified item.

August 14, 2014 | Proprietary and Confidential | + 19 +

IGATE
Speed. Agility. Innovation.

FND_PROFILE.GET

FNDSQF library and database (stored procedure)

Gets the current value of the specified user profile option, or NULL if the profile does not exist

All GET operations are satisfied locally/ñêîn other words, a GET on the server is satisfied from the server-side cache, and a GET on the client is satisfied from the client-side cache.

FND_PROFILE.VALUE

FNDSQF library and database (stored function)

VALUE works exactly like GET, except it returns the value of the specified profile option as a function result.

Arguments (input)

Name : The (developer) name of the profile option whose value you want to retrieve.

Arguments (output)

Value : The current value of the specified user profile option as last set by PUT or as defaulted in the current user's profile.

Demo

- Create a User profile.



Summary

In this session we covered:

- Overview.
- Definition of user profile.
- Hierarchy type.
- Setting Profile Option Values.
- Profile Levels.
- User Profile APIs.
- Steps to create user profile



Review Question

- **Question 3: Responsibility determines the operating unit that you access when you use Oracle Applications.**
 - True/False

- **Question 4: The first step involved in setting up a Multiple Organization Enterprise is to**
 - Define set of books
 - Develop Organization Structure
 - Define organization Relationships
 - Define locations



ERP - Oracle Apps

Lesson 6: Value Set

August 14, 2014

Proprietary and Confidential

- 1 -



None

Lesson Objectives

- To understand the following topics:
 - What is Value Set
 - Types of Value Set
 - Validation Types of Value Set
 - NONE
 - INDEPENDENT
 - TABLE
 - DEPENDENT
 - SPECIAL
 - Storing Method of Value Set
 - Use of Value Set



Introduction

- Value set is primarily a container for your values
- You define your value set such that it can control the types of values that are allowed into the value set
- You can specify the format of your values

Types of Value Set

- Character
- Number
- Time
- Standard Date, Standard Date Time
- Date, Date Time

Value Set Validation Types

- NONE - Validation is minimal
- INDEPENDENT - Input must exist on previously defined list of values.
- TABLE- Input is checked against values in an application table
- DEPENDENT - Input is checked against a subset of values based on a prior value
- SPECIAL - Value set uses a flexfield itself

Value Set Validation Types

- PAIR - Two flexfields together specify a range of valid values
- Translatable Independent - Input must exist on previously defined list of values; translated value can be used.
- Translatable Dependent - Input is checked against a subset of values based on a prior value; translated value can be used.

Validation Type - None

➤ None

- Use 'NONE' value set when you want to allow users to enter any value as long as the value set meets the boundary conditions. (For example, data type and length.)
- When the actual values to be input cannot be anticipated in advance

Validation Type - Independent

➤ **Independent**

- An independent value set provides a pre defined list of values for a segment.
- User selects these values from a pop-up list.
- Values are defined using oracle applications form, using 'define segments values form'.

Validation Type - Dependent

➤ **Dependant**

- A dependent value set is similar to an independent value set, except that the available values in the lov depend on which independent value was selected in a prior segment of the flexfield structure.
- You can think of a dependent value set as a collection of little values in the corresponding independent value set.
- The segment or parameter using the dependent value set must appear after the segment or parameter using the independent value set

Validation Type - Dependent

Independent Value Set

Part Number Structure 1
Category COM Computer
Item 876 Monitor
Color LTN Light Tan

COM - COMPUTER
MACH - MACHINERY
FURN - FURNITURE

Dependent Value Sets

0 - Not applicable

876 - Monitor

877 - Floppy Drive

881 - CPU

0 - Not applicable

1003 - Pump

1004 - Press

1006 - Drill

0 - Not applicable

2373 - Table

2375 - Couch

2376 - Chair

August 14, 2014 Proprietary and Confidential + 10 +

Validation Type - Dependent

➤ **Dependant**

- A table validated value set provides a pre defined list of values like an independent set, but the values are taken from an application table.
- You can specify selection criteria in the form of an sql where clause if you need to restrict the values for some reason.
- Table-validated value sets are very useful when the set of approved values is very large, changes frequently, and is maintained by the application.

SEGMENT VALIDATION – Validation Method

➤ Pair & Special

- Special value sets can accept an entire key flexfield as a segment value in a descriptive flexfield or report parameter
- Pair value sets are a specialized form of special value sets.
 - These allow a mechanism to build a flexfield-within-a-flexfield.
 - These are not generally used for normal flexfield segments.

Validation Type – Translatable Independent

➤ Translatable Independant

- A Translatable Independent value set is similar to an Independent value set in that it provides a predefined list of values for a segment. Translatable Independent value set can contain display values that are translated into different languages

Validation Type – Translatable Dependent

➤ Translatable Dependant

- A Translatable Dependent value set is similar to a Dependent value set in that the available values in the list and the meaning of a given value depend on which independent value was selected in a prior segment of the flexfield structure.
- Translatable Dependent value set can contain display values that are translated into different languages
 - Note]: Translatable value sets enable you to use hidden values and displayed (translated) values in your value sets. In this way users can see a value in their preferred languages, yet the values will be validated against a hidden value that is not translated.

Storing Location of Value Set

Type	List of Values Available	Values Stored In
None	No	N/A
Independent	Yes	AOL* Table
Dependent	Yes	AOL* Table
Table	Yes	Application Table
Special/Pair	Depends on value set	Depends on value set
Translatable Independent	Yes	AOL* Table
Translatable Dependent	Yes	AOL* Table

*Application Object Library

Use of Value Sets

➤ Some of the scenarios where value-set is used

- In a concurrent program, we may want users to enter only number between 1 and 100 for a particular parameter.
- In a concurrent program, we may want users to enter only Yes or No for a particular parameter.
- Suppose a concurrent program has two parameters. First parameter is department and second parameter is employee name. On selecting a particular department, we want to show only those employee names which belongs to the selected department.
- In a particular form , we want to show only a designated list of values for selection by a user.
- In case of accounting reports, we may want users to enter a range of values.

Demo

- Create a value set of Type YES/NO.
- Create a Table Value set based on column of Table.



Review Question

- What are User Profiles?
- What are the user profile levels ?types ?



ERP - Oracle Apps

Lesson 7: Flexfields

August 14, 2014

Proprietary and Confidential

» 1 «



Lesson Objectives

- **At the end of the session you will understand:**
 - What is Flex Field
 - Types of Flex Fields
 - Benefits of Flex Fields



Introduction

- A flexfield is composed of segments. The segments of a flexfield contain the flexfield values. Flexfield segments are combined in a grouping called a structure. The segments that make up a particular structure are logically or functionally related. A single flexfield can have more than one structure.

August 16, 2016

Proprietary and Confidential

+ 3 +



A flexfield is a field made up of sub-fields, or segments. A flexfield appears on your form as a pop-up window that contains a prompt for each segment.

Example

Item Information Flexfield

Segments

Segments: (Category, Item, Color)

	Item Information	
Category	COM	Computer
Item	876	Monitor
Color	LTN	Light tan

Structure: (Category Segment + Item Segment + Color Segment)

o

August 16, 2016 | Proprietary and Confidential | + 46 +

IGATE
Speed. Agility. Imagination.

When you implement a flexfield, you decide how many segments you need for a particular structure and whether you need multiple structures. Being able to define different structures for the same flexfield allows you to tailor the same flexfield to the needs of different end users.

The example on the slide shows three segments:

Category

Item

Color

These three segments together form a structure.

Types

- **There are two types of flexfields:**
 - Key flexfields
 - Descriptive flexfields

August 16, 2016

Proprietary and Confidential

+ 5 +



A flexfield is a field made up of sub-fields, or segments. A flexfield appears on your form as a pop-up window that contains a prompt for each segment.

Storage

➤ Storing FlexData Internally...

The diagram illustrates the internal storage of FlexData. It shows two types of flexfields: 'Key flexfield' and 'Descriptive flexfield'. These are mapped to specific columns in 'Oracle Applications base tables'. The 'Key flexfield' maps to 'Segment1' and 'Segments' columns, while the 'Descriptive flexfield' maps to 'Attribute1' and 'Attribute2' columns.

August 16, 2016

Proprietary and Confidential

+ 8 +

IGATE
Speed. Agility. Imagination.

Storing Flexfield Values in Database Tables

Each segment's value is stored in a column in one of the base tables.

The column name reflects the type of flexfield data it holds. In general, key flexfields store their data in columns called SEGMENTn, where n is a number. In general, descriptive flexfields store their data in columns called ATTRIBUTEn, where n is a number.

The number of SEGMENT and ATTRIBUTE columns available for use by a flexfield varies between flexfields. Not all flexfields can have the same number of fields defined. When implementing a flexfield, determining the number of fields available for use is an important planning step.

When To Use Flexfields?

➤ Descriptive flexfields

- To let your users create new data entry fields without programming.
- To provide user customizable “expansion space” in your forms
- To have context sensitive field arrangements that cannot be foreseen in advance.

August 16, 2016

Proprietary and Confidential

+ 7 +



A flexfield is a field made up of sub-fields, or segments. A flexfield appears on your form as a pop-up window that contains a prompt for each segment.

When To Use Flexfields?

➤ Key flexfields

- To uniquely identify an application entity with an intelligent key, where the key can be multipart and each part having a meaning.

August 16, 2016

Proprietary and Confidential

v 8.0



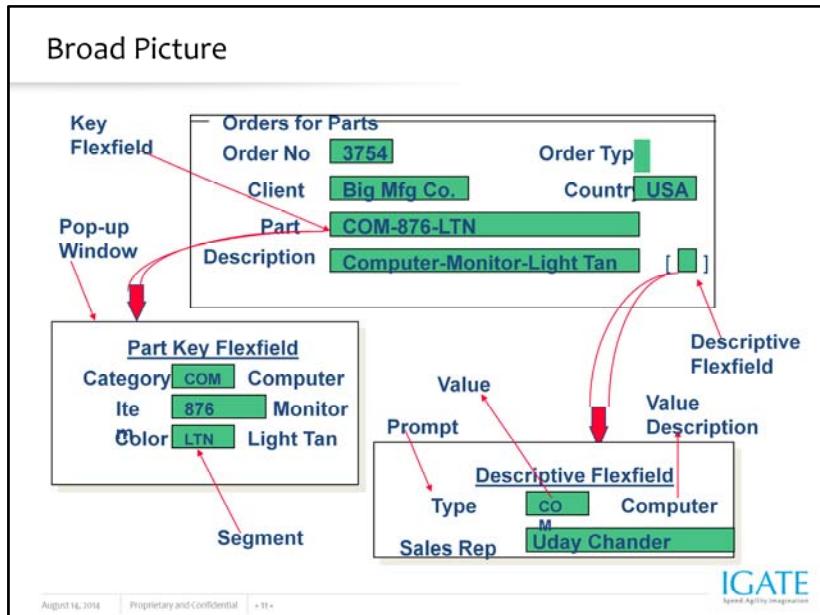
A flexfield is a field made up of sub-fields, or segments. A flexfield appears on your form as a pop-up window that contains a prompt for each segment.

Which flexfield to Use.....

- You use key flexfields to define your own structure for many of the identifiers required by Oracle Applications.
- You use descriptive flexfields to gather additional information about your business entities beyond the information required by Oracle Applications.

Benefits of Flexfields

- Customize applications to support your own accounting, product, and other codes.
- Enable the construction of intelligent keys.
- Customize applications to capture additional data.
- Use the application to validate values and value combinations entered by the user.
- Support multiple field structures depending on data context.



Summary

➤ In this lesson, you should have learned how to:

- What is Flex Field
- Types of Flex Fields
- Benefits of Flex Fields



Summary

➤ In this lesson, you should have learned how to:

- What is Value Set
- Types of Value Set
- Validation Types of Value Set
 - NONE
 - INDEPENDENT
 - TABLE
 - DEPENDENT
 - SPECIAL
- Storing Method of Value Set
- Use of Value Set



ERP - Oracle Apps

Lesson 8: Key Flexfields

August 14, 2014

Proprietary and Confidential

* 1 *



Lesson Objectives

- After completing this lesson, you should be able to do the following:
- Explain intelligent keys and provide examples
 - Identify key flexfields that are required by Oracle Applications
 - Explain the purpose of flexfield qualifiers and optional key flexfield features
 - Design a key flexfield structure
 - Define the key flexfield structure and segment attributes
 - Define flexfield qualifiers and segment qualifiers
 - Implement optional features as needed



Introduction

- Key flexfields are an integral part of a form.
- Each key flexfield comprises a number of segments. They appear on your form as a pop-up window that contains a prompt for each segment.
- Key flexfields serve as an intelligent primary key that uniquely identifies an application entity.

August 14, 2014

Proprietary and Confidential

+ 3 +



A key flexfield lets you provide your users with a flexible “code” data structure that users can setup, the way they like using key flexfield segments. That is, key flexfields let your users customize the application to design their codes the way they want.

Key flexfields let you satisfy different customers without having to reprogram your applications.

Use of Key Flexfields

- Collect information required by Oracle Applications
- Provide users with ability to customize structure and appearance

August 14, 2014

Proprietary and Confidential

+ 4 +



Requirements for Key Flexfields

Key flexfields have a dual purpose. They must collect information required by Oracle Applications and still allow for user customization.

Provide Required Information

Provide information needed for reports and processing. For some applications, particular items of information must be identified within the flexfield. For example, Oracle General Ledger requires the balancing segment of the Accounting Flexfield.

Build unique IDs for use by the applications while giving users meaningful codes.

Provide Customization Capability

Tailor the flexfield to the company's business practice instead of changing the practice to fit the flexfield.

Retain the information the company already keeps.

Oracle Applications Key Flexfields

Oracle Assets	Asset Category Location	Oracle Payroll	Bank Details Cost Allocation People Group
Oracle General Ledger	Accounting	Oracle Receivables	Sales Tax Location Territory
Oracle Inventory	Account Aliases Item Catalogs Item Categories Sales Orders Stock Locators System Items	Oracle Service	Oracle Service Item Grade Job Personal Analysis Position Soft Coded

August 14, 2014

Proprietary and Confidential

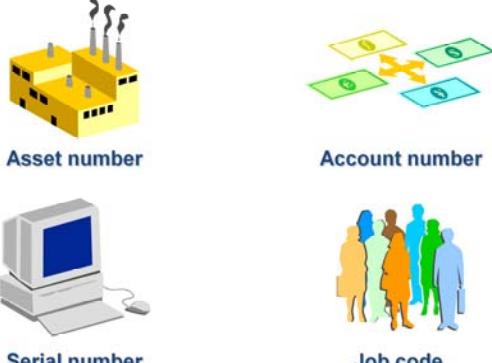
+ 5 +

IGATE
Speed. Agility. Imagination.

Key Flexfields Used by Oracle Applications

The slide shows the key flexfields used by Oracle Applications. The number of key flexfields is significantly smaller than the number of descriptive flexfields.

Intelligent Keys



The diagram illustrates four types of intelligent keys:

- Asset number**: Represented by a yellow building with multiple pipes or antennas.
- Account number**: Represented by a stack of money bills (green, yellow, blue) with a yellow 'X' drawn over them.
- Serial number**: Represented by a computer monitor and keyboard.
- Job code**: Represented by a group of five stylized human figures in orange, green, blue, and yellow.

August 14, 2014 | Proprietary and Confidential | + 6 +

IGATE
Speed. Agility. Imagination.

Key flexfields uniquely identify an important business entity. These often contain one or more values, (value sets), which may individually qualify to be a primary key of a table. Thus the combination, used to identify objects are called intelligent keys. These can be customized to meet individual needs without requiring programming skills. The structure of intelligent fields can be made to change depending upon the data in the form.

Building Intelligent Keys for Oracle Applications

Intelligent keys are multipart key values in which each part of the key contains meaningful information. You use key flexfields to build the intelligent keys required by Oracle Applications.

Because key flexfields are integrated with the internal processing of Oracle Applications, there are more requirements for the structure and content of key flexfields than was the case with descriptive flexfields.

Key Flexfield Structures

Business A

Account Entry	
Corporation	10
Subsidiary	203
Division	3003
Department	025
Account	203

Business B

Account Entry	
Company	10
Division	203
Department	3003
Account	025

August 14, 2014 | Proprietary and Confidential | + 7 +

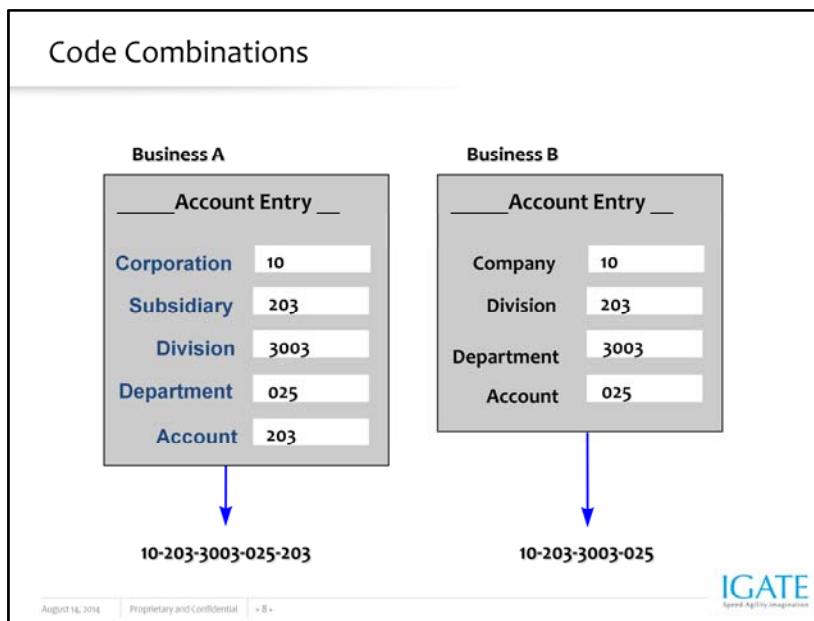


The slide displays two accounting flexfield structures side-by-side. Each structure has a header row 'Account Entry' followed by five levels of hierarchy. Business A's structure is: Corporation (10), Subsidiary (203), Division (3003), Department (025), Account (203). Business B's structure is: Company (10), Division (203), Department (3003), Account (025).

Key Flexfield Structures

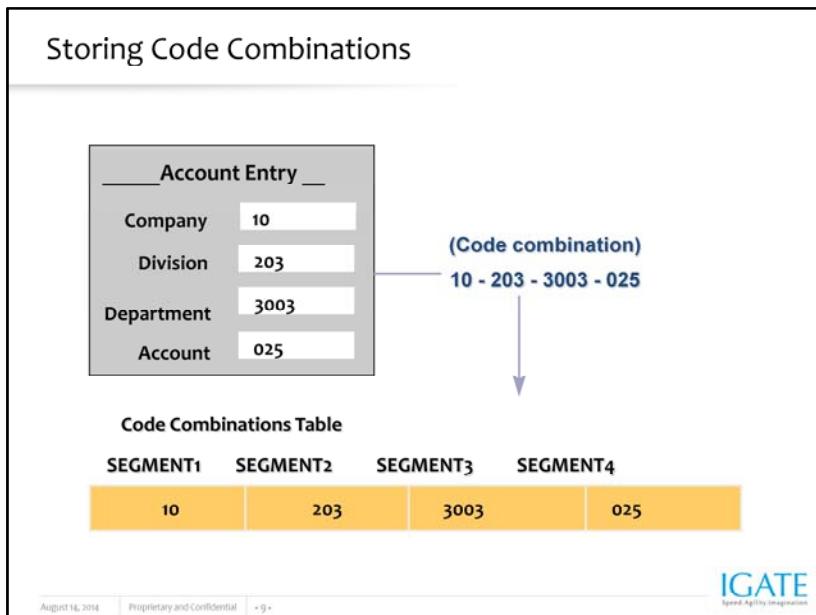
Although it is possible for a key flexfield to use multiple structures, it is much less typical than with descriptive flexfields. Most key flexfields use only one structure. However, key flexfields still allow the user to control the structure of the flexfield. For example, the slide shows two different accounting flexfield structures defined by two different businesses. Each business defines an accounting flexfield that reflects its operating structure.

While in many cases the user has control over which descriptive flexfield structure is used, with key flexfields the application usually determines the correct structure with which to function. For example, Oracle General Ledger determines which accounting flexfield structure to use from the profile option Set of Books.



Key Flexfield Code Combinations

Key flexfields typically consist of several segments. The values provided by these segments make up the code combinations that function as intelligent keys for use by Oracle Applications.



Storing Code Combinations

Each flexfield stores its code combinations in a database table called a code combinations table. In the combinations table, there is one column for every key flexfield segment. These columns are usually named SEGMENTn, where n is a number. There is a set number of SEGMENT columns available for each key flexfield. You assign a key flexfield segment to a particular SEGMENT column when you define the key flexfield.

Each row in the combinations table (that is, each unique combination of segment values) is identified by a unique ID value stored in a unique ID column. This column functions as the primary key for the combinations table. For key flexfields that have multiple structures, there is also a structure ID column.

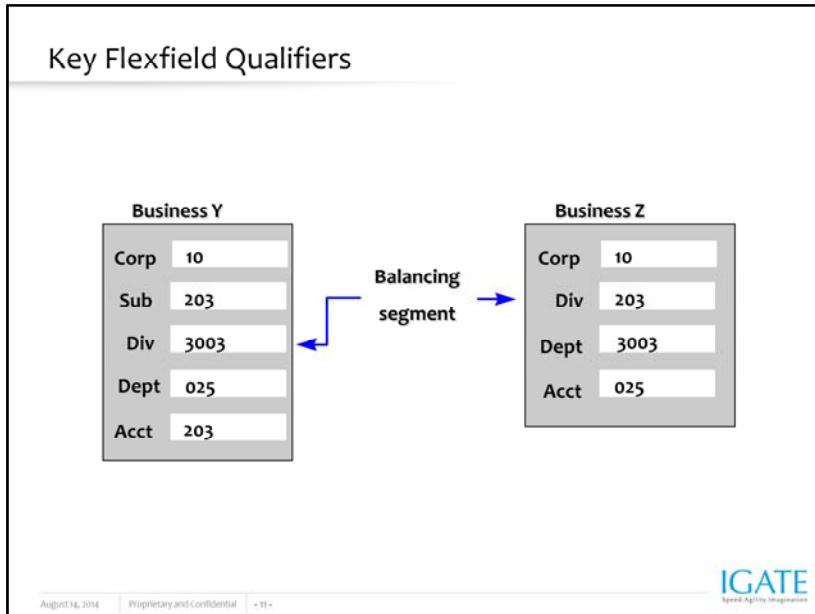
Key Flexfield Application Tables

```
select id_flex_name,  
       application_table_name  
  from apps.fnd_id_flexs  
order by application_id;
```

Key Flexfield Application Tables

This SQL*Plus query can be used to show Oracle Applications key flexfields and the tables in which they store their code combinations.

ID_FLEX_NAME	APPLICATION_TABLE_NAME
Accounting Flexfield	GL_CODE_COMBINATIONS
Category Flexfield	FA_CATEGORIES_B
Asset Key Flexfield	FA_ASSET_KEYWORDS
Location Flexfield	FA_LOCATIONS
Oracle Service Item Flexfield	MTL_SYSTEM_ITEMS_B
Territory Flexfield	RA_TERRITORIES
Sales Tax Location Flexfield	
AR_LOCATION_COMBINATIONS	
Item Categories	MTL_CATEGORIES_B
Account Aliases	MTL_GENERIC_DISPOSITIONS
Item Catalogs	MTL_ITEM_CATALOG_GROUPS
Sales Orders	MTL_SALES_ORDERS
System Items	MTL_SYSTEM_ITEMS_B
Stock Locators	MTL_ITEM_LOCATIONS
Grade Flexfield	PER_GRADE_DEFINITIONS



Key Flexfield Qualifiers

Both descriptive flexfields and key flexfields allow the user to design the flexfield structures and their segments. With descriptive flexfields, neither the information gathered nor the way the information is structured is used internally by Oracle Applications. Key flexfields, however, are different.

Oracle Applications use certain pieces of information collected by some key flexfield segments internally. For example, Oracle General Ledger needs to know which segment in the Accounting flexfield to use for balancing operations. But since the location of the balancing segment in the accounting flexfield can be customized, the application must have a way of locating the segment it needs within any accounting flexfield structure.

Being able to locate particular segments in a key flexfield structure is the purpose for qualifiers. A qualifier is a label attached to a particular key flexfield segment so it can be located by the application requiring its information.

Types of Qualifiers

Flexfield qualifiers:
identify a particular segment

10 203 3003 025

Segment qualifiers:
identify a particular value

IGATE
Speed. Agility. Imagination.

Types of Key Flexfield Qualifiers

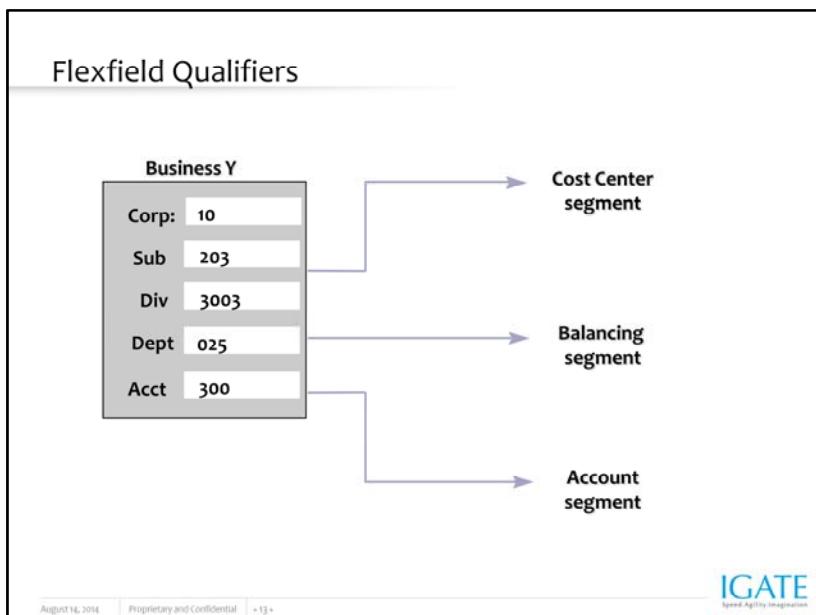
Qualifiers allow the user to retain the ability to customize the structure of the flexfield while still allowing the Oracle Application to find the information it needs to process.

There are two types of qualifiers:

Flexfield qualifiers identify a segment in a flexfield.

Segment qualifiers identify a value in a segment.

The slide shows both types of qualifiers assigned to an accounting flexfield combination.



Flexfield Qualifiers Identify Key Flexfield Segments

The flexfield asks each segment a yes/no question.

Flexfield qualifiers may be unique, global, and required:

Unique: “Is this the segment that this flexfield can have only one of?”

Required: “Is this the segment this flexfield must have to do its work?”

Global: “Is this a segment?” Global qualifiers exist as “carriers” for segment qualifiers.

Assigning Flexfield Qualifiers to Segments

Global qualifiers need not be assigned since they apply automatically to every segment in the flexfield.

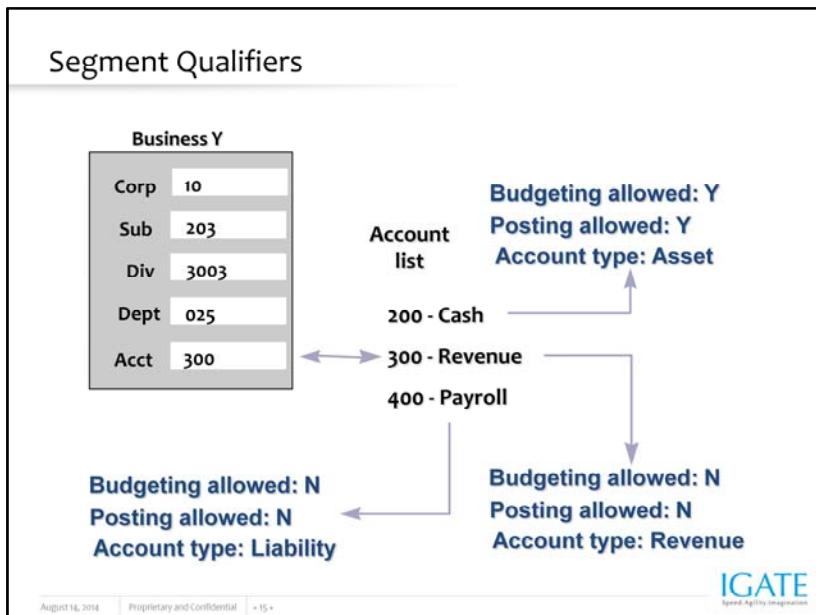
Assign flexfield qualifiers while defining segments.

Examples of Key Flexfields using Qualifiers

- **General Ledger**
 - Accounting flexfield
- **Oracle Assets**
 - Location flexfield
 - Asset Category flexfield
- **Oracle Human Resources**
 - SoftCoded Key flexfield
- **Oracle Payroll**
 - Cost Allocation flexfield
- **Oracle Accounts Receivable**
 - Sales Territory flexfield

Key Flexfields Using Qualifiers

The slide shows the key flexfields that use qualifiers and the Oracle application that uses each key flexfield.



Identifying Values in Flexfield Segments with Segment Qualifiers

A segment qualifier is similar to the segment asking each value the question, “What type of value are you?”

For example, the account number 300 may be used within a company as a revenue account.

Use the following segment qualifiers with the accounting flexfield:

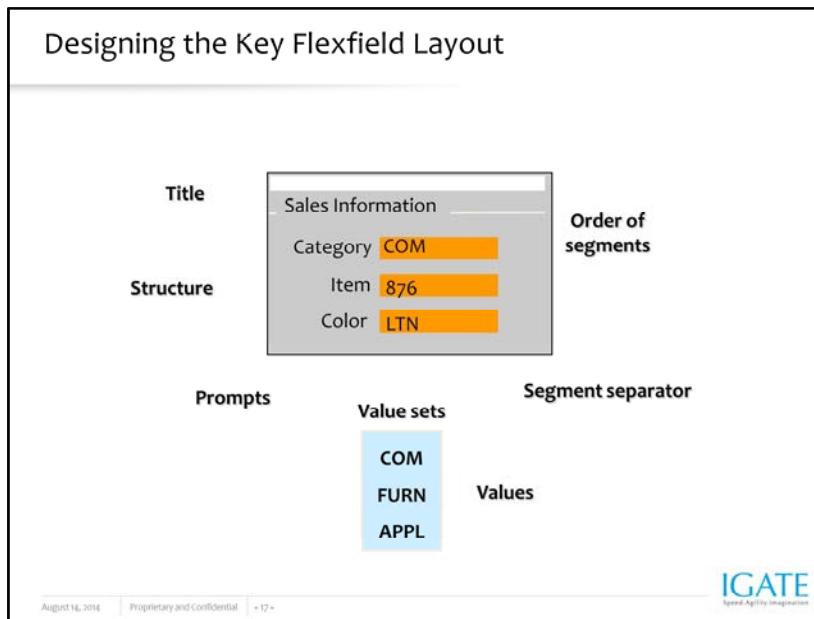
Allow Budgeting

Allow Posting

Account Type: Asset, Expense, Liability, Ownership/Stockholder's Equity, or Revenue

Planning Decisions

- Multiple structures?
- Resources available?
- Qualifiers required?
- Dynamic inserts?
- Cross validation?
- Shorthand aliases?
- Value checking?
- Value security?



Designing Key Flexfield Layout

\Design the structures needed and the segments for each structure:

- Identify the structure titles.
- Plan the number and order of segments.
- Identify the segment separator.
- Determine the value sets and values to be used.
- Plan the window prompts.

Designing Segments

- Enabled or Displayed
- Required
- Validated
- Secured

August 14, 2014

Proprietary and Confidential

+18+



Designing Key Flexfield Segments

Decide how each segment of the key flexfield should behave and what values to allow in each segment.

Enabled or Displayed - Can users see this segment? Disabled segments are not displayed. If the segment does not display, use a default value to populate it.

Required - Can users leave the segment without entering a value? Most key flexfield segments require a value.

Validation - Most key flexfield segments provide a list of values. Use a predefined value set, or design a new one for this segment. Not using a value set is equivalent to using a validation type of None, character format, width same as underlying segment column, uppercase allowed, and no right justification or zero fill.

Value Security - Should security rules for the value set apply to this segment?

Related Segments - Link segments with ranges of Low and High to enforce a relationship between them.

Specifying Default Values

Default Type	Default Value
Constant	Any literal value
Current Date	Current time
Current time	Current time or current date/time
Field	Default Value field value
Profile	Value of profile in Default Value
Segment	Value in prior segment
SQL Statement	Result of SQL query

August 14, 2014

Proprietary and Confidential

+ 19 +



Specifying Segment Defaults - Examples

Default Type: Constant

Default Value: The constant specified.

Example: USA

Result: USA

Default Type: Current Date

Default Value: The date at the time of entry.

Example:

Result: MAY 01, 2000

Default Type: Current Time

Default Value: The Date/Time at the time of entry.

Example:

Result: 14:30:00 MAY 01, 2000

Default Type: Field

Default Value: The value in the specified field. Use the format block:field

Example: ORDER:LINE

Result: 3

Default Type: Profile

Default Value: The value of the specified profile option. Use the application name of the profile option.

Example: GL_SET_OF_BOOKS_ID

Result: 101

Planning Key Flexfield Values

- Plan values for independent and dependent value sets.
- Group values by ranges to allow easier security and cross-validation.
- Identify values to be used with segment qualifiers.

August 14, 2014

Proprietary and Confidential

• 20 •



Planning Values for Use with Key Flexfields

Plan the values for the independent and dependent value sets created for this key flexfield. Choose values logically and systematically. Grouping values together logically makes defining security and validation rules much easier.

Oracle General Ledger Values Information

Certain applications (especially Oracle General Ledger) require special handling of values:

Oracle General Ledger applications require that segment qualifiers be assigned to some values used by the Accounting Flexfield.

Oracle General Ledger applications can create hierarchies of values using rollup groups and parent-child relationships for processing and reporting.

The Oracle General Ledger courses provide more details.

Definition Procedure

- Define new value sets if needed.
- Define the key flexfield structure.
- Define the structure segments, including qualifiers.
- Freeze and compile the flexfield definition.
- Define value set values, including qualifiers.

Defining Value Sets

- Use the Value Sets window to define a value set for each segment of the key flexfield.
- Create independent, dependent, or table-validated value sets for segments that should use a pop-up list of values.
- Define the maximum size to be no larger than the size of the underlying table segment column.

(N) Application—>Validation—>Sets

Accessing the Key Flexfield Definition

- Use the Key Flexfield Segments window to find the flexfield definition you want to modify.
- Before you can modify the definition, you must unfreeze it.

(N) Application—>Flexfield—>Key—>Segments

Specifying Flexfield Behavior

- Use the Key Flexfield Segments window to enter:
 - Enabled
 - Segment Separator
 - Cross-Validate Segments
 - Freeze Rollup Groups
 - Allow Dynamic Inserts

(N) Application—>Flexfield—>Key—>Segments

August 14, 2014

Proprietary and Confidential

• 24 •



Specifying the Key Flexfield Behavior

Enabled - This enables shorthand entry of frequently used value combinations.

Segment Separator - Specify a segment separator character. Segment separators are especially important for key flexfields since their values are often displayed concatenated.

Cross-Validate Segments - This enables cross-checking of segment value combinations.

Freeze Rollup Groups - Rollup groups are used by the Accounting flexfield.

Allow Dynamic Inserts - This allows new key value combinations to be dynamically created and inserted into the table.

When you have defined the flexfield level attributes, click the Segments button to continue defining individual segments for this structure.

Defining Segment Attributes

➤ Use the Segments Summary window to enter:

- Number
- Name
- Window Prompt
- Column
- Value Set
- Displayed
- Enabled

(N) Application—>Flexfield—>Key—>Segments (B) Segments

Defining Segment Attributes

Use the Segments Summary window to define most of the segment attributes.

Number - This specifies the sequence in which the fields will appear on the window.

Name - The name by which this segment is known within Oracle Applications. Name the segment intuitively. Other segments may refer to this one for validation information. Also, the view generated uses the segment names for its column names.

Window Prompt - The prompt that will appear on the window. The segment name is the default.

Column - Specify the SEGMENT column in the underlying base table that contains this segment's data. A pop-up list shows the SEGMENT columns that are still available for use.

Displayed - If you choose to not display a segment, specify a default to populate it.

Enabled - This flags the segment as available for use.

Defining Validation and Size Attributes

- Use the Segments window to enter options for:
 - Validation
 - Sizes
 - Prompts

(N) Application—>Flexfield—>Key—>Segments (B) Segments
(B) Open



Defining Validation and Size Attributes

Validation Information

Use the Validation block fields to specify value set information.

Choose a predefined value set with a list of values or design one for this particular segment with the Value Sets window.

Specify default information if you need to populate the segment with a default value.

Determine whether security rules should apply to this value set for this segment.

Size Specifications

Display Size determines the field size on the flexfield. Specify a display size the same as the maximum segment size to avoid scrolling.

Keep prompts small for neater reports.

Click the Flexfield Qualifiers button to determine whether this flexfield has any qualifiers to be assigned.

Defining Flexfield Qualifiers

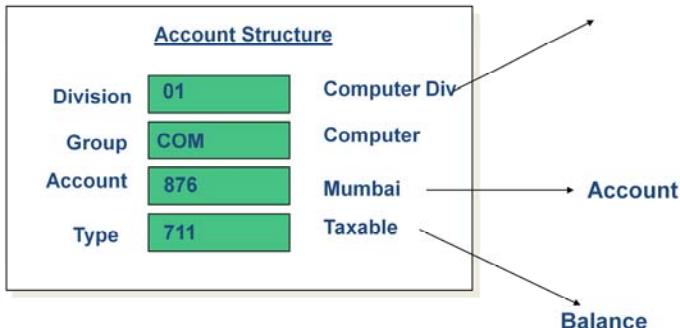
- Use the Flexfield Qualifiers window to assign qualifiers to segments as appropriate for this flexfield.
- Not all flexfields use qualifiers with segments.
- The Accounting Flexfield is an important user of flexfield qualifiers.

(N) Application—>Flexfield—>Key—>Segments (B) Segments

(B) Open (B) Flexfield Qualifiers



Flexfield Qualifiers



- Cost center, account and balance are Flexfield qualifiers. They tell you what

Freezing and Compiling the Definition

- Save after freezing to automatically compile the flexfield definition.
- Submit the request to build the structure view by freezing; submit the request to rebuild the flexfield view by closing the window.
- Freeze and compile after making any changes to the definition. Changes take place immediately.
- You see your changes immediately. Other users must exit the system or change responsibilities.

(N) Application—>Flexfield—>Key—>Segments

Defining Value Set Values

- Use the Segment Values window to create values for the independent and dependent value sets created for the new key flexfield structure.
- Access the value sets by specifying the flexfield segments using them.

(N) Application—>Validation—>Values

Defining Segment Qualifiers

- In the Values, Hierarchy, Qualifiers region of the Segment Values window, navigate to the Qualifiers field to open the Segment Qualifiers window.
- Specify segment qualifiers for values as appropriate for the key flexfield.

(N) Application—>Validation—>Values

August 14, 2014

Proprietary and Confidential

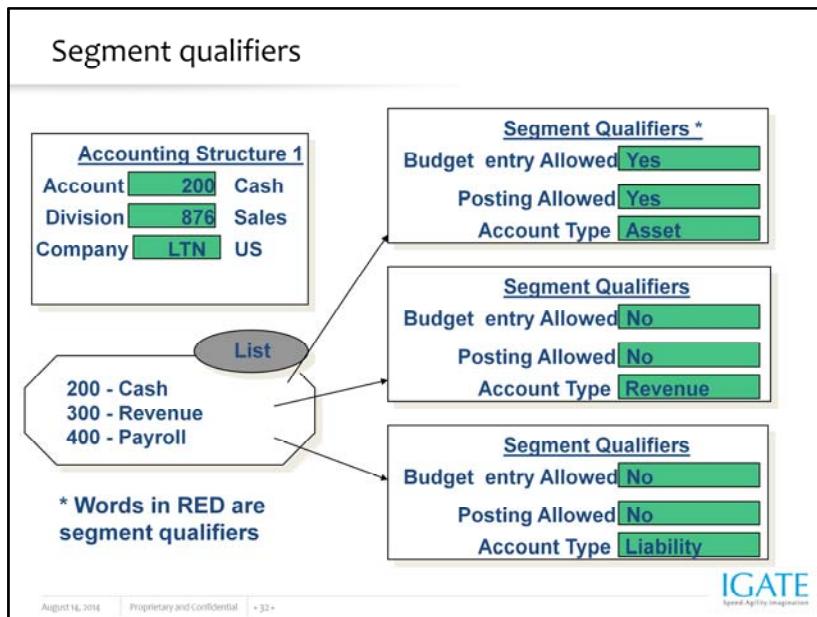
+ 31 +



Defining Segment Qualifiers

Specify additional qualifiers at the value level when appropriate. For example, Allow Budgeting is an Accounting Flexfield segment qualifier.

Whenever possible do not change a value, change its description. If reuse is not possible, disable unused values, do not delete them.



A segment qualifier describes characteristics of key segment values.

They hold additional information about the flexfield qualified segments.

Segment qualifiers are used to obtain information about segment values the end user enters while using the application.

A segment qualifier designates a particular type of value within a segment.

In oracle applications only 'accounting flexfield' uses segment qualifiers.

Key Flexfield Combinations Table

- Use a generic combinations table to accept new valid combinations which are not predefined by you or the end user. This facilitates dynamic insertions. However, generic insertions may be disallowed, even in this type of table.
- Use a specific combinations table to accept only those combinations which are pre-defined.

August 14, 2014

Proprietary and Confidential

• 33 •

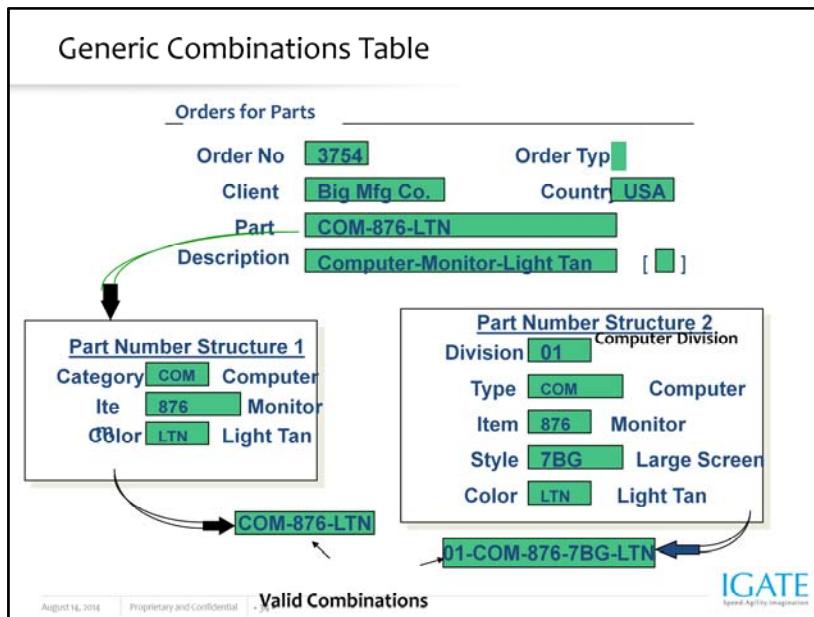


Key flexfields support a maximum of 70 segment columns in a combinations table.

One column is used to assign ‘unique id’ to each valid combination.

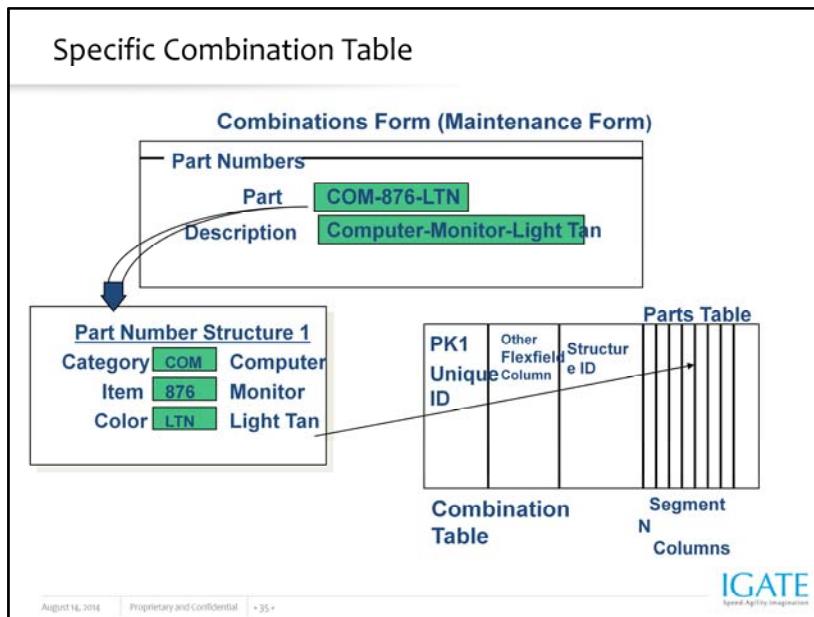
One column may be used for defining multiple structures by users.

The table should have a derived column for segment qualifiers, if they are used.



If the application uses a generic combination table, the end users can enter new valid combinations directly from the application form.

Before storing a new valid combination in a generic combinations table, the flexfield ensures that the new combination satisfies the cross-validation rules.



Define your combination table as specific if it contains any mandatory application specific columns, i.e., Columns which cannot be entered using flexfield popup window.

Therefore users cannot dynamically enter new valid combinations.

All the valid combinations are anticipated and identified using a combinations table maintenance form.

Demo

- Demo to define key flexfields



Summary

➤ **In this lesson, you have learnt:**

- Key flexfields are required to pass information to Oracle Applications.
- Some key flexfields have qualifiers that must be defined.
- Design the key flexfield structure, behavior, and appearance.
- Define your key flexfield according to your previously developed plan.
- Define flexfield qualifiers and segment qualifiers if required by the flexfield.
- Implement key flexfield options where appropriate.



IGATE
Speed. Agility. Imagination.

Review – Questions

➤ **Question 1: Key Flex fields are Unique in Combination**

- True/False



ERP - Oracle Apps

Lesson 9: Defining Descriptive Flexfields

August 16, 2014

Proprietary and Confidential

+1+



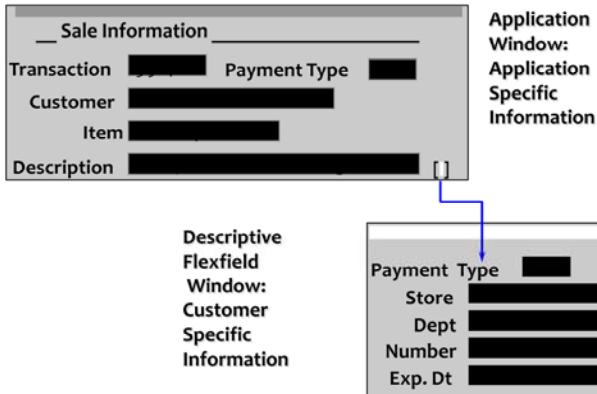
Lesson Objectives

➤ After completing this lesson, you should be able to do the following:

- Identify the descriptive flexfield to be used and the information to be gathered
- Organize the information according to usage
- Plan the layout of the descriptive flexfield
- Plan the behavior of the descriptive flexfield
- Define the descriptive flexfield structure
- Define global segments
- Define context-sensitive segments as appropriate



Using Descriptive Flexfields



Typical Descriptive Flexfields Information

Benefits information

Calendar information

Labor cost information

Lease information

Currency exchange information

Payment information

Credit information

Budget information

Distribution system information

Some Oracle Applications: Descriptive Flexfields

Oracle Assets	Bonus Rates Calendar Types Price Indexes	Oracle Payables	Bank Branch Payment Terms Site Address
Oracle General Ledger	Daily Rates	Oracle Receivables	Credit History Information
Oracle Bills of Material	Activity Information Item Cost Information Shift Time Information	Oracle Work in Progress	Employee Labor Rate Shop Floor Status WIP Parameters

Identifying a Descriptive Flexfields

- The presence of a descriptive flexfield on a form is indicated by a single-space field enclosed in brackets. Whenever you see this, there is a descriptive flexfield defined for use with that form.
- In some cases there may be multiple descriptive flexfields for use with the same form.

Descriptive Flexfield Components

- **Global segment**
 - Displays information common to all contexts
- **Context-sensitive segment**
 - Displays information appropriate only to a particular context
- **Reference field**
 - A field on the application window whose value is used to determine contexts
- **Context field**
 - A field in the structure whose value is used to determine contexts

Global Segments

**Global
Segments
(Common)**

Finance context

Store number

Credit card context

Store number

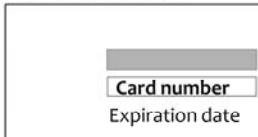
Check context

Store number

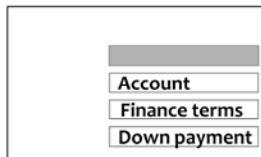
Context-Sensitive Segments

Context-Sensitive Segments
(Vary by context)

Credit card context



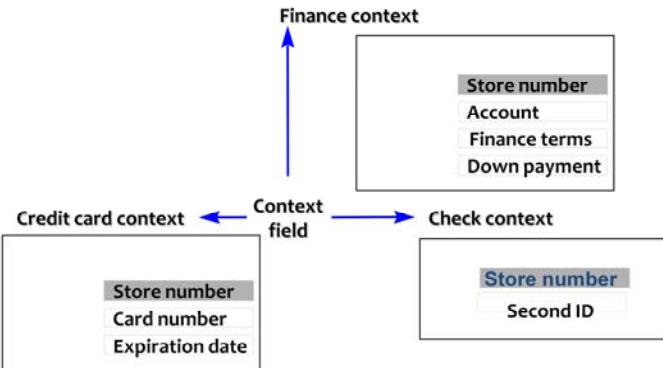
Finance context



Check context



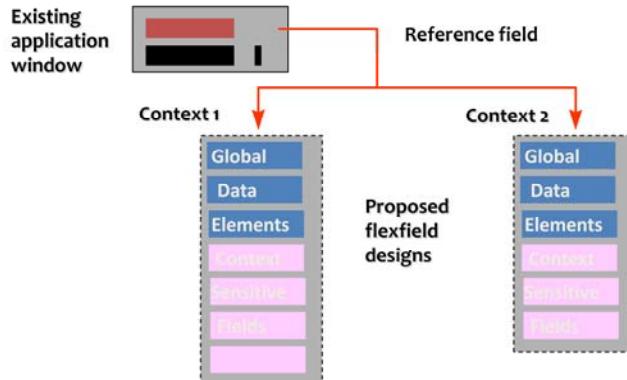
Distinguishing Between Contexts



Reference and Context Fields

- Reference Field: a field on the existing form whose value is used to automatically distinguish between contexts
- Context Field: a field created in the descriptive flexfield structure that is used to allow the user to manually select different contexts

Using Reference Fields



Using Context Fields

Existing application window

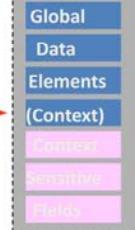


Context 1



Proposed flexfield designs

Context 2



(Context fields)



Demo

- Demo to define descriptive flexfields



Review – Questions

- **Question 1: Intelligent Keys are nothing but key Flexfields**
 - True / False

- **Question 2: ISBN number is a Key Flexfields**
 - True / False



ERP- Oracle Apps

Lesson 10: Form Registration

August 14, 2014

Proprietary and Confidential

+ 1 +



Lesson Objectives

- At the end of the session you will be able to:
 - Know about the Function Security
 - Define a New Menu Structure
 - Data groups
 - Integrate Applications with Custom Applications
 - Register the Forms, Functions and Create the Menu of Functions



Form Registration

- When you develop your application components, you must place them in the appropriate directories on the appropriate machines so that Oracle Application Object Library can find them
- Compile the form on the OraApps Server using predefined commands to create the fmx file

August 14, 2014

Proprietary and Confidential

+ 3 +



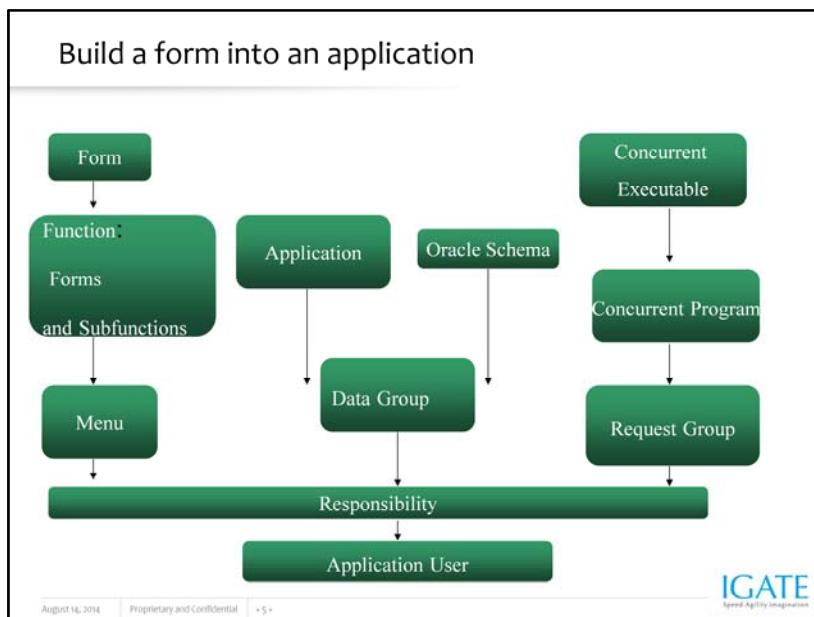
For example, reports written using Oracle Reports are typically placed in a subdirectory called reports on the concurrent processing server machine, while forms belong in separate subdirectories, depending on their territory and language (such as US for American English, D for German, and so on), on the forms server machine.

The fmx file of the form must be in all uppercase and must be located in forms/<language> subdirectory of your application directory structure.

For example : XXCOFI_TOP/forms/US where
\$XXCOFI_TOP is the product directory and maps to
/ccnv01/oracle/ccnvappl/xxcofi/1.0.0

Form Registration Contd...

- To register a form, the pre-requisite is to register your application with Oracle Application Object Library
- The combination of the application name and form name uniquely identifies your form
- Note: The form cannot be compiled and run on the client machine



One of the most important functions of a system administrator is to manage security for the Oracle Applications environment. In Oracle Applications security is implemented by creating user signons and relating them to a responsibility. The responsibility specifies the actual access authorizations. I.e Users then have access to all the functionality associated with that responsibility. You can also create any custom responsibilities you need apart from the standard responsibilities.

Application Security: Overview

User Security

You authorize a user to sign on to Oracle Applications by defining an application user. You then assign one or more responsibilities to the new user.

Responsibility Security

A responsibility is a collection of authorizations that allow access to:

- A specific application or applications
- A set of books

A restricted list of windows, functions, and reports

Each user has one or more responsibilities, and several users can share the same responsibility.

A system administrator can assign standard or with Oracle Applications or create new custom responsibilities as needed.

Defining a New User

Security > User > Define

Define an authorized user of Oracle Applications by specifying a username and password. Grant application privileges by assigning one or more responsibilities to the user.

When you install Oracle Applications, a standard Applications user called SYSADMIN is created for you. Several default responsibilities are also created. You can also create custom responsibilities.

What Is a Data Group?

A data group is a collection of pairings of an application with an Oracle ID. Data groups automatically support concurrent processing and cross-application reporting. They guarantee that an application connects to a unique application database account.

Note: The installation process automatically defines data groups for Oracle Applications, so you only need to define additional data groups based on your specific requirements.

Application-Oracle ID Pairs

An application can be listed only once in a data group.
An Oracle ID can be paired with more than one application.
A custom application registered with Oracle Applications can be included in a data group.

Data Groups and Application Object Library

Application Object Library owns the database tables referred to during concurrent processing and the standard submission of reports by any Oracle Application. Therefore all applications need access to the Application Object Library tables. When you are defining a data group, the application Application Object Library is automatically included. The Application Object Library's Oracle ID cannot be updated or deleted.

By defining a data group, you can determine which Oracle account (Oracle ID) an application's windows, reports, or concurrent programs connect to. Use data groups to grant application database account privileges to a responsibility and the requests that it submits

You can control the relationship among applications, forms, and concurrent programs by defining a data group.

Setting Up Function Security



August 14, 2014

Proprietary and Confidential

> 8 <

IGATE
Speed. Agility. Imagination.

Function security is the mechanism by which user access to applications functionality is controlled.

Oracle Applications GUI-based architecture aggregates several related business functions into a single form.

Because all users should not have access to every business function in a form, Oracle Applications provides the ability to identify pieces of applications logic as functions. When part of an application's functionality is identified as a function, it can be secured (i.e., included or excluded from a responsibility).

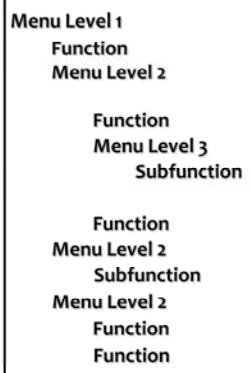
Application developers register functions when they develop forms. A System Administrator administers function security by creating responsibilities that include or exclude particular functions.

Terms

Function: A function is a part of an application's functionality that is registered under a unique name for the purpose of assigning it to, or excluding it from, a responsibility.

There are two types of functions: form functions, and non-form functions. For clarity, we refer to a form function as a form, and a non-form function as a subfunction, even though both are just instances of functions in the database.

Function Security Contd...



- **Function:** A set of executable code available as a menu option
- **Subfunction:** A subset of a form's functionality

August 16, 2014

Proprietary and Confidential

+ 9 +



Function Security

You can manage security by controlling access to individual functions through menu definitions.

About Functions

A function is a set of code in Oracle Applications that is executed only if the name of the function is present in a list maintained within a responsibility.

There are two types of functions: a form function and a nonform function or subfunction. A subfunction represents a securable subset of a form's functionality.

Adding Functions to or Removing Functions from a Responsibility

Maintain menu structures while eliminating specific functionality.

Exclude individual functions from a responsibility.

Adding or Removing Menus of Functions

Use menus to group functions together.

Exclude groups of functions by excluding a menu from a responsibility.

Navigator Menu Displays

Navigator displays only the menu items needed for navigation. Because you cannot choose subfunctions from a menu, they are not displayed. Submenus consisting only of subfunctions are also not displayed.

Menus

- A menu is a hierarchical arrangement of functions and menus of functions
- Each responsibility has a menu assigned to it.

August 14, 2014

Proprietary and Confidential

+ 10 +



Menu: A menu is a hierarchical arrangement of functions and menus of functions. Each responsibility has a menu assigned to it.

Menu Entry: A menu entry is a menu component that identifies a function or a menu of functions. In some cases, both a function and a menu of functions correspond to the same menu entry. For example, both a form and its menu of subfunctions can occupy the same menu entry.

Responsibility: A responsibility defines an application user's current privileges while working with Oracle Applications. When an application user signs on, they select a responsibility that grants certain privileges, specifically:

The functions that the user may access: Functions are determined by the menu assigned to the responsibility.

The concurrent programs, such as reports, that the user may run: The application database accounts that forms, concurrent programs, and reports connect to.

Defining a New Menu Structure



August 14, 2014

Proprietary and Confidential

+ 11 +



Exclusion of Functions and Menus

Use exclusion rules to customize a responsibility. You can exclude functions at any level.

When you exclude a menu item from a responsibility, all menus and functions that are nested in that menu are also excluded.

When you exclude a function from a responsibility, all occurrences of that function throughout the menu structure of a responsibility are excluded.

New Menu Structure

Use the Menus form to define menus pointing to functions that you want to make available to a new responsibility.

Make New Responsibilities, Not New Menus

If possible, apply exclusion rules to existing menus to customize a responsibility rather than constructing an entirely new menu structure.

Determine the Application Functionality Required

Different jobs require access to different function groups.

Identify predefined menus, forms, and form subfunctions to use as entries when defining a new menu.

Plan Your Menu Structure

Start with a blank Menus form (blank screen). Menus cannot be copied. A menu saved under a different name overwrites the original menu (there is no Save As feature).

Start with the lowest-level menus. A menu must be defined before it can be selected as an entry on another menu.

Assign menus and functions to higher-level menus.

Document your menu structure by printing the Function Security Menu Report.

Assign the menu structure to a new responsibility by using the responsibilities form.

Create a new responsibility.

Data Groups

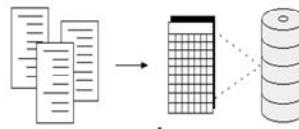


The screenshot shows a window titled "Data Groups" with a tab labeled "Standard". The description is "Standard Data Group". The table lists various applications and their Oracle IDs:

Application	Oracle ID
ADS Development	APPS
Activity Based Management	APPS
Advanced Benefits	APPS
Advanced Outbound Telephone	APPS
Advanced Planning Foundation	APPS
Advanced Pricing	APPS
Advanced Product Catalog	APPS
Advanced Supply Chain Plan	APPS
Alert	APPS
Application Implementation	APPS

Below the table is a button labeled "Go to Application Page".

Applications and ORACLE Usernames

APPLICATION	ORACLE USERNAME	DATABASE
Application owners: Concurrent Programs, which execute commands on the application's tables.	Database Privileges: <ul style="list-style-type: none"> - Tables accessed - SELECT rows - UPDATE rows - DELETE rows 	



IGATE
Speed. Agility. Integration.

Defining Data Groups

Data groups specify your applications database connections.

Introduction to Data Groups

A data group is a list of Oracle Applications and the Oracle username assigned to each application. Each application in a data group must have an Oracle username assigned to it. An application may be listed only once in a data group.

An Oracle username and password allow access to an application's tables in an Oracle database. Each Oracle username in a data group determines the database tables and table privileges accessible by the corresponding application or applications.

Data Group's Purpose

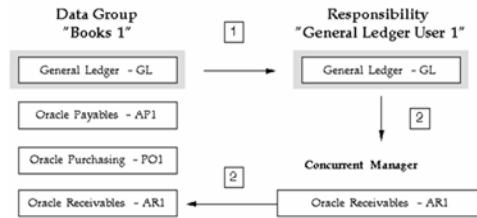
Each responsibility has a data group associated with it. A data group serves two purposes:

1. It identifies the Oracle username that forms connect to when you select the responsibility.
2. Concurrent managers use a data group to match the application that owns a report or concurrent program (submitted by a user of the responsibility) with an Oracle username.

Page-10-13

Data Groups Contd...

Using Data Groups with multiple product installations.



August 14, 2014 Proprietary and Confidential • 14 •

IGATE
Speed. Agility. Imagination.

Using Data Groups with multiple Sets of Books

Use data groups to support multiple installations of an Oracle Applications product (for example, Oracle Payables) that supports multiple sets of books, where a different application is associated with each set of books. See: Using Data Groups with multiple product installations.

For example, with two installations of Oracle Payables supporting two Sets of Books, use data groups to indicate which Oracle Payables Oracle username to access from a certain General Ledger responsibility.

Define a data group for each application installation (set of books).

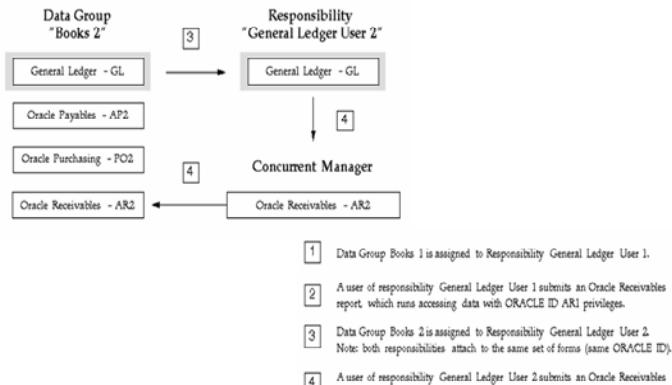
Define a responsibility for each application installation (set of books), and assign the appropriate data group to each responsibility.

Using Data Groups to include custom applications

Use data groups to include custom applications you develop using Oracle's Application Object Library. To integrate a custom application with Oracle Applications, you must register the application using the Applications window. See: Applications.

An example of using two Data Groups to support two installations of Oracle Payables, Oracle Purchasing, and Oracle Receivables is shown above.

Data Groups Contd...



Basic Definitions

- **Form**
 - An Oracle Forms .fmx file located in their application basepath/forms/US (or appropriate language directory)
- **Function**
 - A function is a part of an application's functionality, registered under a unique name that can be assigned to a responsibility
- **There are two types of functions:**
 - Form functions (forms)
 - Subfunctions

Basic Definitions Contd...

➤ **Form function**

- Invokes an Oracle Forms form
- A form has the unique property that users may navigate to it from the Navigate window

➤ **Subfunction**

- It is a subset of a form's functionality

➤ **Menu**

- A menu is a hierarchical arrangement of functions and menus of functions

Basic Definitions Contd...

➤ **Menu Entry**

- A menu entry is a menu component that identifies a function or a menu of functions

➤ **Responsibility**

- When application users sign on, they select a responsibility that determines, among other things, the functions they may access
- Available functions are determined by the menu assigned to the current responsibility

Register a Form with AOL

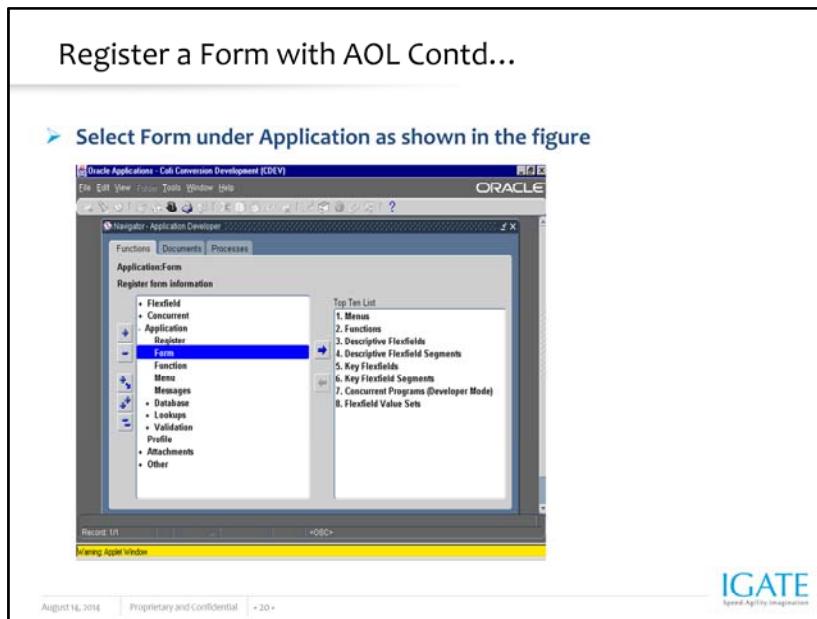
➤ Select the responsibility Application Developer



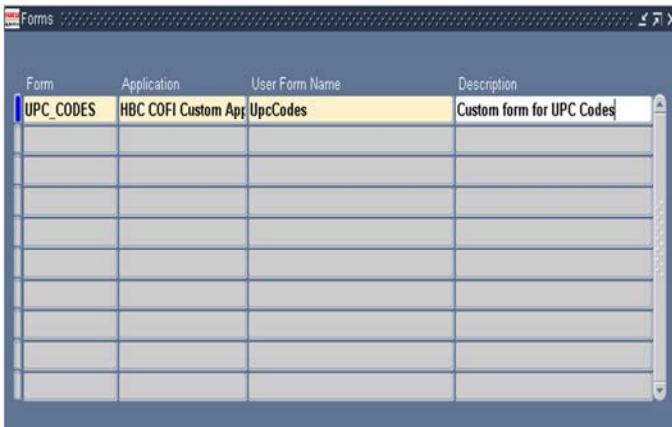
The screenshot shows a web-based Oracle Applications interface. At the top, there's a blue header bar with the text "Welcome OffShore" and some icons. On the right side of the header, it says "ORACLE Applications" and "Main Menu". Below the header is a sidebar titled "Navigate". The sidebar has two main sections: "Self Service" and "Applications". Under "Self Service", there are links for "Oracle Fulfillment Core User", "Oracle Fulfillment Super User", "Oracle Installed Base Customer", "Oracle Installed Base User", and "Workflow Administrator Web Applications". Under "Applications", there are links for "Alert Manager", "Application Developer", "Bills of Material", "COFI Customer Support", "COFI Installed Base Admin", and "COSEI Inventory". The link "Application Developer" is highlighted with a blue background, indicating it is the selected responsibility.

IGATE
Speed. Agility. Integration.

August 14, 2014 | Proprietary and Confidential | + 19 +



Register a Form with AOL Contd...



August 16, 2014

Proprietary and Confidential

+ 21 +

IGATE
Speed. Agility. Imagination.

Enter the filename of the form (without the extension)

The application is the one that owns your form. The application tells Oracle Application Object Library where to find your form file

This is the form name you see when selecting a form using the Functions window

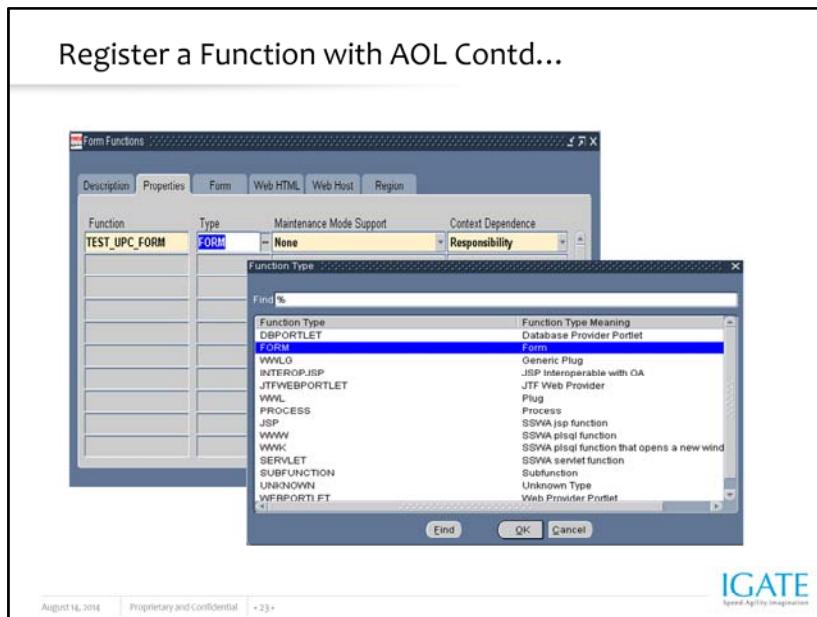
Register a Function with AOL Contd...

➤ Select function under Application

The screenshot shows the Oracle Application Navigator interface. The title bar reads "Navigator - Application Developer". Below it, there are three tabs: "Functions" (which is selected), "Documents", and "Processes". Under the "Functions" tab, the "Application Function" section is visible. On the left, a tree view titled "Define Form Functions" lists several categories: Flexfield, Concurrent, Application, Register, Form, Function (which is highlighted with a blue selection bar), Menu, Messages, Database, Lookups, Validation, Profile, Attachments, and Other. To the right of the tree view is a "Top Ten List" containing the following items:

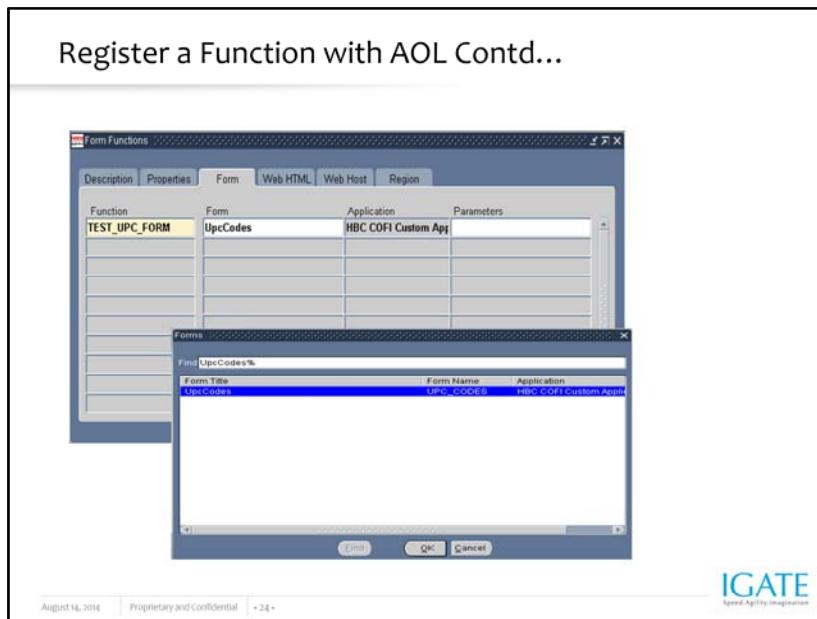
1. Menus
2. Functions
3. Descriptive Flexfields
4. Descriptive Flexfield Segments
5. Key Flexfields
6. Key Flexfield Segments
7. Concurrent Programs (Developer Mode)
8. Flexfield Value Sets

At the bottom of the window, there are footer links: "August 14, 2014", "Proprietary and Confidential", and "22". In the bottom right corner of the slide area, there is a logo for IGATE with the tagline "Speed. Agility. Integration".



Function: Users do not see this unique function name, but you use it in your code when starting a form using function security routines or testing for function availability.

The function is a form type function which invokes the Oracle Forms file



If you are defining a user form, select the user form name and application of your form

Create a Menu of Functions

➤ Select Menu under Application

The screenshot shows the Oracle Application Developer Navigator interface. The title bar says "Navigator - Application Developer". The main window has tabs for "Functions", "Documents", and "Processes", with "Functions" selected. Under the "Application:Menu" heading, there's a "Define menu" section with a tree view. The tree includes categories like Flexfield, Concurrent, Application, Register, Form, Function, and a selected "Menu" node. Other nodes like Database, Messages, Lookups, Validation, Profile, Attachments, and Other are also listed. To the right of the tree is a "Top Ten List" panel containing the following items:

1. Menus
2. Functions
3. Descriptive Flexfields
4. Descriptive Flexfield Segments
5. Key Flexfields
6. Key Flexfield Segments
7. Concurrent Programs (Developer Mode)
8. Flexfield Value Sets

At the bottom of the window, there are footer links: "August 14, 2014", "Proprietary and Confidential", and "25". In the bottom right corner, there is an "IGATE" logo with the tagline "Speed. Agility. Integration".

Create a Menu of Functions Contd...

The screenshot shows the Oracle Applications 'Menus' screen. The 'Menu' field is set to 'INV_ITEMS'. The 'User Menu Name' is 'INV_ITEMS', 'Menu Type' is 'Standard', and 'Description' is 'Inventory Items Menu'. The main area displays a table of menu items:

Seq	Prompt	Submenu	Function	Description	Grant
8	Item Search		Items Search	Item Search	<input checked="" type="checkbox"/>
9	Pending Status		Pending Status	Pending Status	<input checked="" type="checkbox"/>
10	Documents		Inventory Documents	Maintain Inventory Documents	<input checked="" type="checkbox"/>
11	Import	INV_ITEMS_OI		Item Open Interfaces Menu	<input checked="" type="checkbox"/>
12	Delete Items	INV_BOMFDEL_MENU	Inventory Delete Items	Delete Items	<input checked="" type="checkbox"/>
13	Copy Item Attrib		Copy Item Attributes	Copy Item Attributes	<input checked="" type="checkbox"/>
14	ICQ		ICQ		<input checked="" type="checkbox"/>
15	Test Folder		Test Folder		<input checked="" type="checkbox"/>
16	Test UPC		Test UPC		<input checked="" type="checkbox"/>
17	Test UPC Form		Test UPC Form		<input checked="" type="checkbox"/>

At the bottom left, it says 'August 16, 2014 Proprietary and Confidential • 26 •'. On the right, there is a blue 'IGATE Speed. Agility. Imagination.' logo.

Select Menu under Application

Press F11 to enter query mode, enter the menu name (wildcard allowed) for the Inventory responsibility and press ctrl-F11 to execute

Menu

Choose a name that describes the purpose of the menu

User Menu Name

A menu name that is used when a responsibility calls a main menu or when one menu calls the another

Sequence

Enter a sequence number to specify where a menu entry appears relative to other menu entries in a menu

Navigator Prompt

Enter a user-friendly, intuitive prompt that would be displayed by your menu in the Navigate window

Leave the prompt blank for subfunctions that should not appear in the Navigator menu listing even though they are on the menu

Submenu

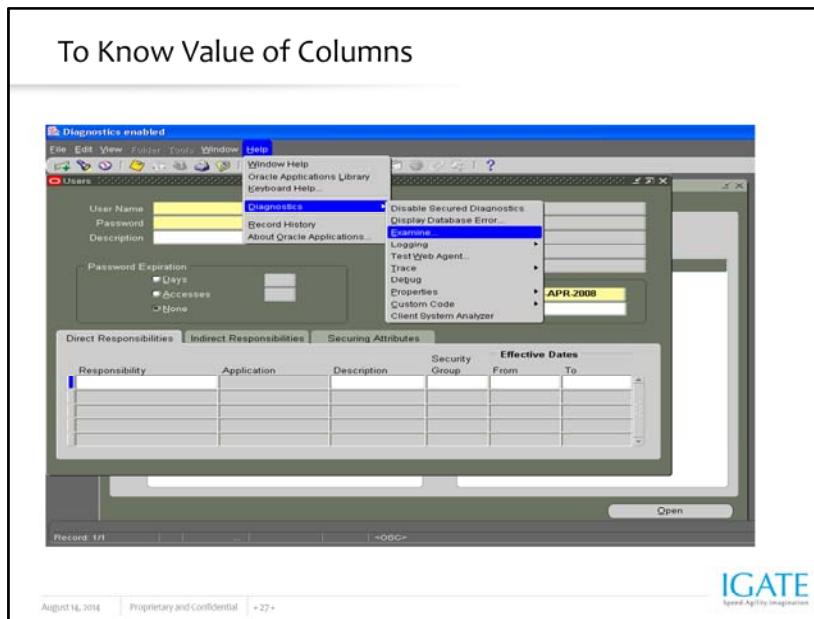
Calls another menu, from which you can select menu entries

Function

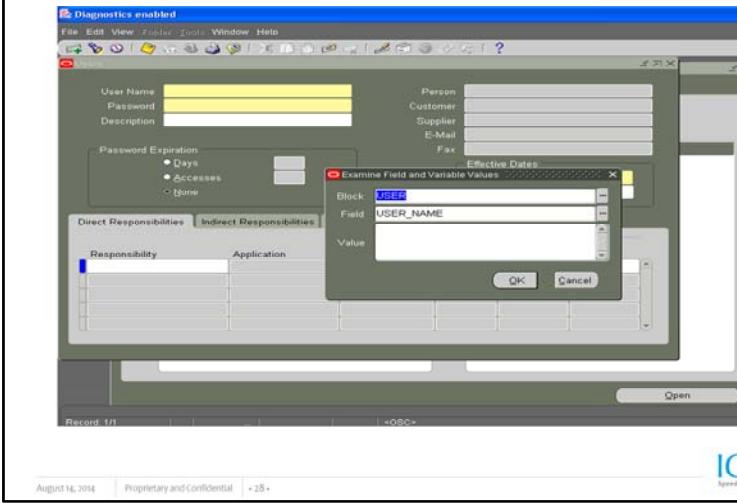
Call a function that is to be included in the menu. A form function (form) appears in the Navigate window and allows access to that form.

Description

Enter a description of the menu choice. This description appears in the Description field under the menu path in the Navigator

IGATE
Speed. Agility. Ingenuity.

To Know Value of Columns



IGATE
Speed. Agility. Ingenuity.

Instructor Demo

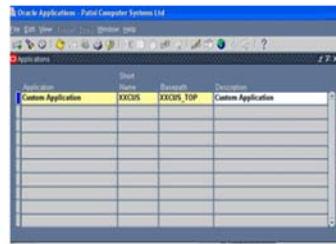
- Demo to view a form in Oracle Applications and Register a Form with AOL



IGATE
Speed. Agility. Imagination.

Integrate Applications with Custom Applications

- Register your custom application with the Application Object Library. It is recommended to use an XX as the preface to the custom schema short name so that it does not conflict with any future Oracle Application short names
 - Log into Applications as the System Administrator and navigate to:
Application -> Register



August 14, 2014

Proprietary and Confidential

- 30 -

IGATE
Speed. Agility. Imagination.

Integrate Applications with Custom Applications Contd...

- Create a custom directory tree for your custom schema as the APPLMGR user. Use the basepath parameter from the Application registration for the top level directory. This top level directory will reside just under APPL_TOP. The subdirectories under the custom directory may vary depending on the server type (forms server, concurrent processing server, etc....). Make sure that the rights/protections are open for the world (rwx)

- For example:

APPL_TOP (/appl/v1100000)		XXCUS_TOP
(/appl/v1100000/XXCUS) --> other product directories		

bin forms html lib log mesg out reports
US US



IGATE
Speed. Agility. Integration.

Integrate Applications with Custom Applications
Contd...

The screenshot shows a PuTTY terminal window titled "192.168.27.181 - PuTTY". The command entered is "ls" in the directory "/data/applications/visionapp1/XXCUS/11.5.0". The output lists several files and subdirectories:

```
f pwd
/datas/applications/visionapp1/XXCUS/11.5.0
f ls
emp.log      lib          out          xx.rdf
Converter.c   forms        log          reports
afiedt.buf    hist.txt    mesg        sql
bin          html        nyse_budgeting temp
f ls -t1
total 36
drwxrwxrwx  2 applvis  dba          512 Mar 29 11:05 out
-rw-r--r--  1 custvis other      39 Mar 24 10:53 afiedt.buf
drwxrwxrwx  3 applvis  dba        1536 Mar 24 10:11 bin
drwxrwxrwx  3 applvis  dba          512 Mar 21 18:27 forms
drwxr-xr-x  7 custvis other      512 Feb 20 14:23 BSE
-rw-r--r--  1 custvis other      0 Feb 15 09:46 xx.rdf
-rw-r--r--  1 custvis other     221 Feb  9 12:54 emp.log
drwxr-xr-x  2 custvis other      512 Jan 17 11:55 temp
-rw-r--r--  1 custvis other      4 Jan  5 10:36 Converter.c
-rw-r--r--  1 custvis other      0 Dec  9 13:12 hist.txt
drwxr-xr-x  4 custvis other      512 Sep 19 2005 nyse_budgeting
drwxrwxrwx  3 applvis  dba        512 Sep 16 2005 reports
drwxrwxrwx  2 applvis  dba        3072 Sep 14 2005 mesg
drwxrwxrwx  2 applvis  dba        512 Jun  3 2005 sql
drwxrwxrwx  3 applvis  dba        512 Apr 11 2005 html
```

August 14, 2014 | Proprietary and Confidential | +32+

IGATE
Speed. Agility. Integration.

Integrate Applications with Custom Applications Contd...

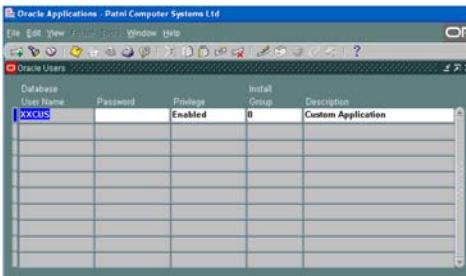
- Modify the applications environmental file (example: APPLSYS.env) to include the custom schema basepath as the APPLMGR user
- For example: XXCUS_TOP="/appl/v1100000/XXCUS" export XXCUS_TOP
- Register the custom schema as an Oracle user

Integrate Applications with Custom Applications Contd...

- Create the user in the RDBMS database using SQL*Plus under the system account. Give the user a default and temporary tablespace with quotas and then grant the CONNECT role
 - For example: Connect to Oracle as System user
- ```
SQL> create user XXCUS identified by CUST default tablespace USER_DATA
 temporary tablespace TEMP quota unlimited on USER_DATA quota
 unlimited on TEMP;
```
- SQL> grant connect to XXCUS identified by CUST;
- XXCUS is the product short name, CUST is the password for the custom schema, USER\_DATA and TEMP are existing tablespaces

## Integrate Applications with Custom Applications Contd...

- Register the user with the Application Object Library.
- Log into Applications as the System Administrator and navigate to:  
Security -> ORACLE -> Register. For example:



The screenshot shows a window titled "Oracle Applications - Patni Computer Systems Ltd". The menu bar includes File, Edit, View, Insert, Tools, Window, Help. The main area is titled "Oracle Users" and contains a table with columns: Database, User Name, Password, Privilege, Install Group, and Description. A single row is selected, showing "XXCUS" in the User Name column, "Enabled" in the Privilege column, "0" in the Install Group column, and "Custom Application" in the Description column. The bottom of the window shows the date "August 16, 2014", the status "Proprietary and Confidential", and the page number "• 35 •".

IGATE  
Speed. Agility. Imagination.

Register an ORACLE username with Oracle Applications. An ORACLE username grants access privileges to the ORACLE database. This activity is performed by the DBA.

The installation process always registers your ORACLE username, so you need not register it unless you create a custom application using Oracle Application Object Library, or if you wish to associate an additional ORACLE username with Oracle Applications.

If you register an ORACLE username as a "restricted" ORACLE username, you submit a concurrent request to set up read-only privileges to the Oracle Application Object Library tables. An "enabled" ORACLE username has all privileges to those tables. A "disabled" ORACLE username has no privileges to those tables.

If you do not 'register and enable' your ORACLE username or if you disable a registered ORACLE username, the user cannot use Oracle Application Object Library features such as menus and flexfields.

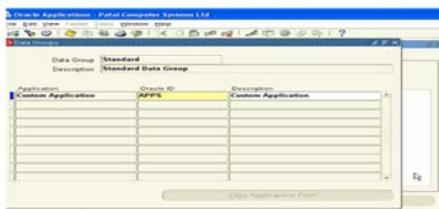
You should not change the registration of any ORACLE usernames that the installation process registers, other than changing the passwords.

#### Passwords for the APPS Accounts

The applsys password must be identical to the password for the APPS accounts (APPS, APPS2, APPS3). The uniform passwords enable the different sets of books to operate correctly.

## Integrate Applications with Custom Applications Contd...

- Add the custom schema to a data group. Log into Applications as the System Administrator and navigate to: Security -> ORACLE -> DataGroup
- It is recommended that you use the STANDARD datagroup and pair the custom schema with APPS or you can add a new data group. This depends upon the requirements

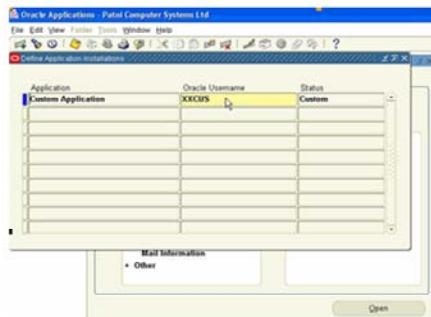


**IGATE**  
Speed. Agility. Imagination

Since the application will be accessing the database objects with user apps, there should be synonyms in apps for the custom objects in the custom schema xxkus, AND apps should be given the necessary privileges select/insert/update/delete/execute) on the objects in xxkus.

## Integrate Applications with Custom Applications Contd...

- Register your custom application with Alerts. Log into Applications as the Alert Manager and navigate to System -> Installations



**IGATE**  
Speed. Agility. Integration

## Integrate Applications with Custom Applications Contd...

- Create your custom tables, indexes, views and sequences. It is suggested that you add WHO columns to your custom tables so that Oracle Applications can keep track of customizations
- Register your custom schema's tables (including flexfields) with the PL/SQL package AD\_DD. You use the procedure AD\_DD.register\_table for the custom schema tables and AD\_DD.register\_column for the custom schema table columns
- Issue a commit after the pl/sql procedure is executed. The results can be seen in the application by navigating to Application→Database→Table using the Application Developer responsibility

August 14, 2014

Proprietary and Confidential

• 38 •



```
execute ad_dd.register_table(appl short name,table
name,table type, next extent, % free, % used)
```

For example:

```
$ sqlplus apps/apps
SQL> execute ad_dd.register_table ('XXCUS',
'CUST_TABLE', 'T', 8,10, 90)
appl short name=XXCUS, table name=CUST_TABLE, table
type=T, next extent=8, % free=10, % used=90
```

```
execute ad_dd.register_column (appl short name,table
name,column name, column seq, column type, column
width, null, translate)
```

For example:

```
$ sqlplus apps/apps
```

```
SQL> execute ad_dd.register_column ('XXCUS',
'CUST_TABLE', 'CUST_NO',1,NUMBER,5,'N','N')
```

Here appl short name=XXCUS, table name=CUST\_TABLE,  
column name=CUST\_NO, column seq=1, column  
type=NUMBER, column width=5, null=N, translate=N.  
Issue a commit after the pl/sql procedure is executed. The  
results can be seen in the application by navigating to  
Application→Database→Table using the Application  
Developer responsibility.

## Integrate Applications with Custom Applications Contd...

- Run the custom schema against the APPS\_DDL and APPS\_ARRAY\_DDL packages
- Run the scripts \$AD\_TOP/admin/sql/adaddls.pls, adaaddls.pls, adaddlb.pls and then adaaddlb.pls (in this order) under SQL\*Plus:

August 14, 2014

Proprietary and Confidential

• 39 •



For example :

```
$ sqlplus apps/apps
SQL> @$AD_TOP/admin/sql/adaddls.pls
system_pword custom_schema
custom_schema_pword
SQL> @$AD_TOP/admin/sql/adaaddls.pls
system_pword custom_schema
custom_schema_pword
SQL> @$AD_TOP/admin/sql/adaddlb.pls
system_pword custom_schema
custom_schema_pword
SQL> @$AD_TOP/admin/sql/adaaddlb.pls
system_pword custom_schema
custom_schema_pword
```

## Integrate Applications with Custom Applications Contd...

- Integrate your database objects with the APPS schema by granting APPS the access to your custom schema's objects
  - Grant all privileges from each custom data object to APPS
  - Create a synonym in APPS to each custom data object
  - Create custom code objects in APPS

August 14, 2014 | Proprietary and Confidential | + 40 +



For example: \$ sqlplus xxcus/cust

SQL> grant all on CUST\_TABLE to APPS

For example: \$ sqlplus apps/apps

SQL> create synonym APPS.CUST\_TABLE for  
XXCUS.CUST\_TABLE

\$ sqlplus apps/apps

SQL> create function CUST\_FUNCTION

## Summary

➤ In this session we covered about:

- Setting up the Function Security
- Defining a New Menu Structure
- Data groups
- Integrate Applications with Custom Applications
- Registering the Forms, Functions and Creating the Menu of Functions



## Summary

➤ In this lesson, you should have learned how to:

- Descriptive flexfields gather additional information.
- Design the descriptive flexfield to support the different needs of different users.
- Define flexfield level attributes.
- Define global segments for the Global Data Elements structure.
- Define a reference or context field if using different contexts.



## ERP- Oracle Apps

### Lesson 11: Managing Concurrent Programs and Reports

August 14, 2014

Proprietary and Confidential

+ 1 +



## Lesson Objectives

- After completing this lesson, you should be able to do the following:
- Use Standard Request Submission (SRS) to submit requests
- Monitor the processing of a request
- Define a request group to control user access to reports
- Customize the standard SRS processing of reports by defining a request group with a code
- Set default parameter values and share values among multiple reports by defining a request set

## Basic Definitions

### ➤ Concurrent Program

- A concurrent program is a program that does not require continued interaction on your part to perform a specific task. For example, a concurrent program may be a program written to create a report, or to post a batch of general ledger journal entries

### ➤ Concurrent Process

- A concurrent process is an instance of a running concurrent program
- Each time a concurrent manager receives a request and runs a concurrent program, it creates a new concurrent process. A concurrent process can run simultaneously with other concurrent processes (and other activities on your computer)

## Concurrent Processing

Every Oracle Applications product contains reports and programs that are specific to that product.

A report generates a summary or detail presentation of Oracle Applications information, whereas a program can perform a function.

The reports and programs you have access to are defined by the responsibility you use.

For example, as an Oracle Receivables user, you may be able to run a report to create invoices. Or in your Oracle General Ledger application, your responsibility may allow you to run a program to post journal entries.

## Basic Definitions (Contd.)

### ➤ Concurrent Request

- A concurrent request is a request that you submit to run a concurrent program as a concurrent process.

### ➤ Concurrent Manager

- A concurrent manager is a component of concurrent processing that monitors and runs tasks without tying up your computer.

### Concurrent Request:

You issue a concurrent request when you submit a report or program to run using Standard Request Submission or when you choose an action button in a product-specific submission window

## Basic Definitions (Contd.)

### ➤ Report

- A report is an organized presentation of specific Oracle Applications information.

### ➤ Request Set

- A request set is a collection of reports and/or programs that you group together

August 14, 2014

Proprietary and Confidential

+ 5 +



### Report:

You can view a report online, or send it to a printer. The content of a report can range from summary information to a complete listing of values. Reports run as concurrent programs in Oracle Applications.

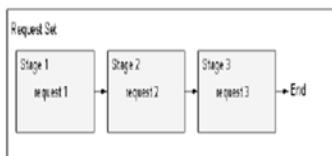
### Request Set:

You can submit the reports and/or programs in a request set all at once using a single transaction.

## Basic Definitions (Contd.)

### ➤ Stage

- A stage is a component of a request set used to group requests within the set
- All requests in a stage are run in parallel, while the stages themselves are run sequentially in the request set



Request sets are divided into one or more "stages" which are linked to determine the sequence in which your requests are run. Each stage consists of one or more requests that you want to run in parallel (at the same time in any order). For example, in the simplest request set structure, all requests are assigned to one stage. This allows all of the requests to run in parallel.

To run requests in sequence, you assign requests to different stages, and then link the stages in the order you want the requests to run. For example, if you had three requests: Request 1, Request 2, and Request 3, that needed to run in the sequence 1, 2, 3, then you would assign each request to a different stage and link the stages in that order.

The concurrent manager allows only one stage in a request set to run at a time. When one stage is complete the following stage is submitted.

A stage is not considered to be complete until all of the requests in the stage are complete. One advantage of using stages is the ability to run several requests in parallel and then move sequentially to the next stage. This allows for a more versatile and efficient request set.

## Basic Definitions (Contd.)

### ➤ Parameter

- In Standard Request Submission, a parameter is a report variable whose value you can change each time you run a report.

August 14, 2014

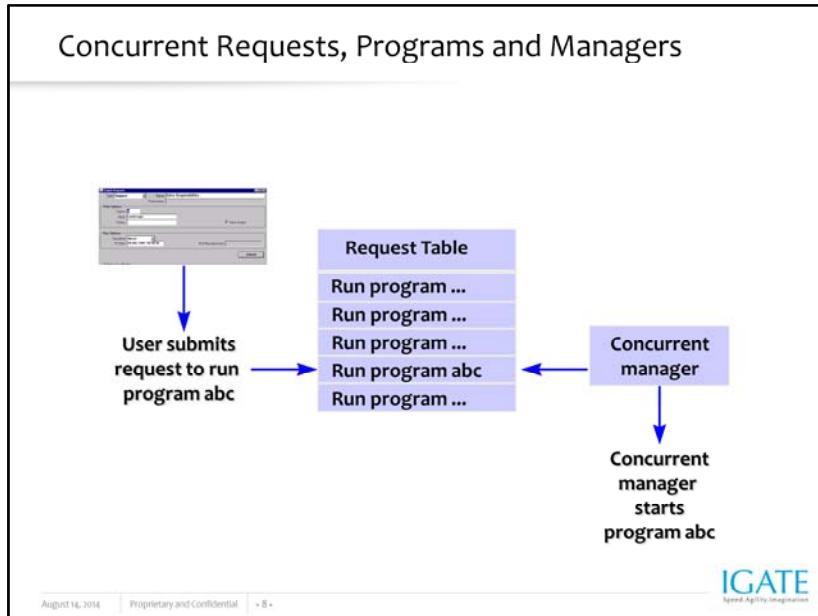
Proprietary and Confidential

+ 7 +



### Parameter:

For example, you might run an audit report that requires you to enter an audit date each time you run the report. The audit date is a parameter for the report.



#### Concurrent Requests, Programs, and Manager

Concurrent processing allows long-running, data-intensive programs to run simultaneously with online operations.

Oracle Applications programs can run concurrently with each other as well as with other programs; they are referred to as concurrent programs.

Requests to run Oracle Applications programs—for example, to run an Oracle General Ledger report—are concurrent requests. Each concurrent request inserts a row into a database table maintained by the Oracle Application Object Library.

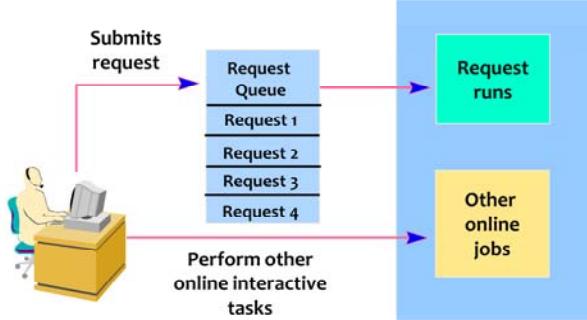
Concurrent managers read requests from the requests table and start concurrent programs.

## Concurrent Requests, Programs and Managers (Contd.)

- Oracle Applications provides you with two features: Concurrent Processing and Standard Request Submission
- **Concurrent processing**
  - Is a feature that allows you to run a non-interactive, data-dependent function, such as a report or program, simultaneously with online operations
  - With concurrent processing, you can complete non-interactive tasks without interfering with the interactive work you perform at your computer

An example of concurrent processing occurs when you use the Post Journals window in your Oracle General Ledger application. Once you specify the journal batches to post and choose the Post button, your Oracle General Ledger application uses concurrent processing to post the journal batch entries without further involvement from you. Meanwhile, your computer is still available for you to continue doing other work in Oracle Applications. Oracle Applications runs all of its reports and programs as concurrent processes.

## Concurrent Requests, Programs and Managers (Contd.)



## Standard Request Submission

### ➤ Standard Request Submission

- Helps to run the reports and programs that are a part of your responsibility
- Is a feature that works with concurrent processing
- Provides a set of windows to run, submit and control over the output of reports and programs

August 14, 2014

Proprietary and Confidential

+ 11 +



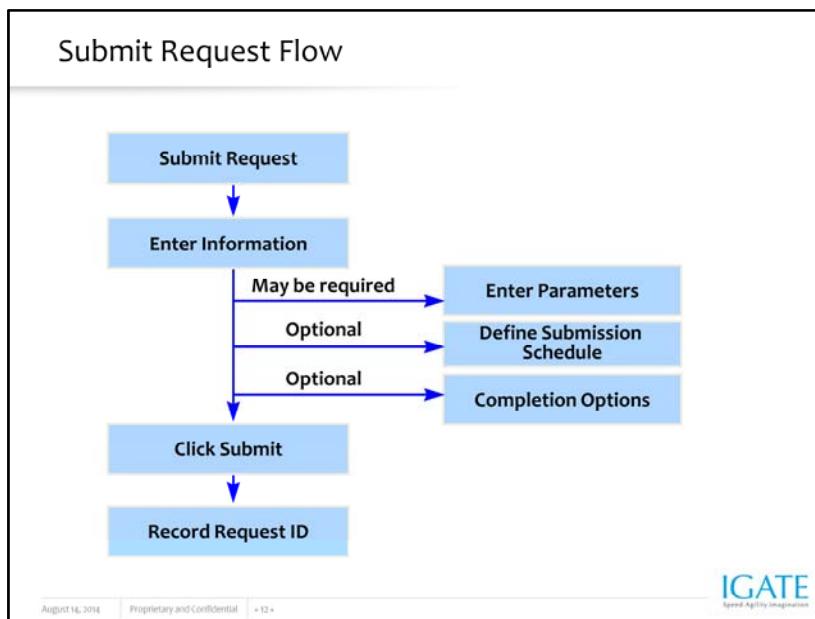
## Standard Request Submission

To help you run the reports and programs that are a part of your responsibility.

It is a feature that works with concurrent processing to provide a common interface for running your Oracle Applications reports and programs.

Standard Request Submission provides you with a set of windows for running reports and programs and a set of windows for creating groups of reports and programs to run together.

These windows give you control over the submission and output of your reports and programs.



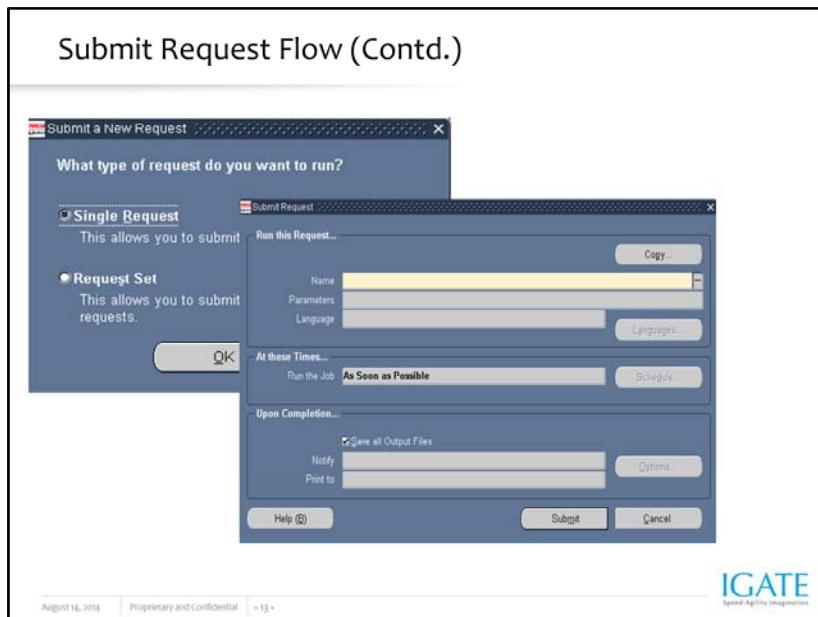
### Using Standard Request Submission (SRS)

Using Standard Request Submission gives you control over how you can run your requests and request sets.

There are three elements involved in submitting a request: selecting the request or request set to be submitted, defining a submission schedule, and providing completion options.

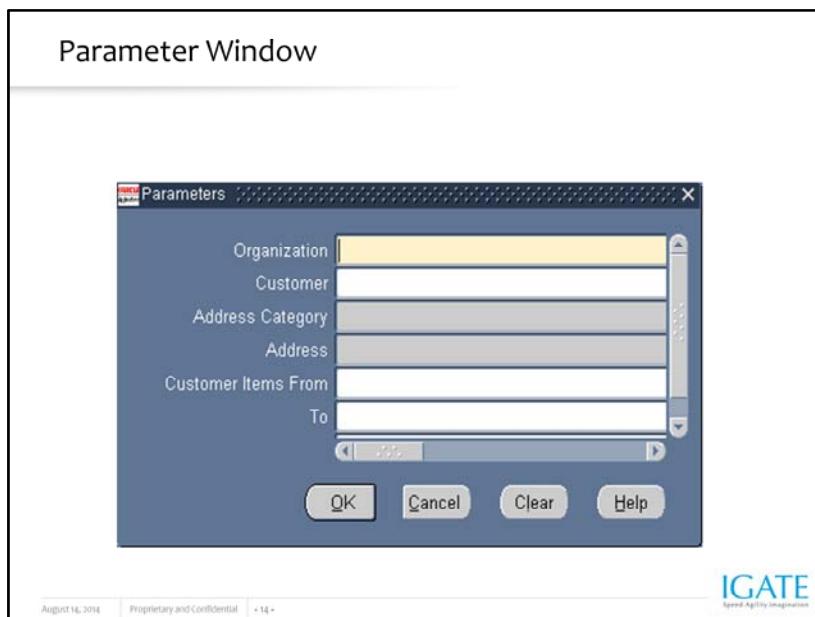
Defining a schedule can be as simple as submitting As Soon as Possible, or it can involve using a more complex schedule that you define when you first submit your request. This schedule can then be used for other requests in the future. Completion options enable you to deliver notification to others using Oracle Workflow, as well as specifying which printers and how many copies of the output you want to produce for each request.

You can submit as many requests as you like from the Submit Request window. You can even submit a request more than once if you want to run the same request with different parameter values.



1. Navigate to the Submit a New Request Window: (N)  
Requests > Run
2. Check the option for Single Request or Request Set.
3. Click OK.
4. Use the Copy a Prior Request button to use a previously entered request submission or Select the name of the request that you want to run from the list of values.

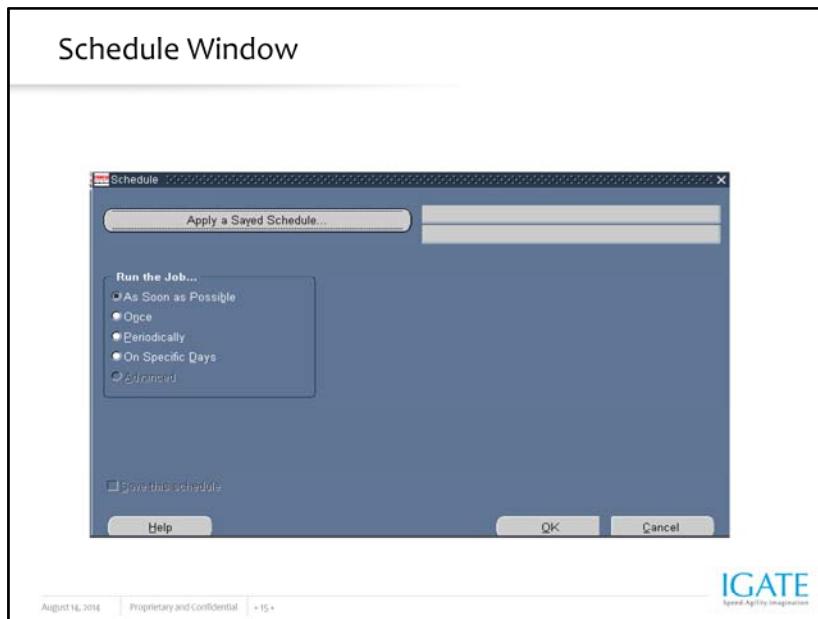
The responsibility that you are using determines the request group and the requests that will appear in the list of available requests.



A Parameters window automatically appears if you select a request that requires parameter values.

The Prompts in the Parameters window are specific to the request that you select.

The parameters you enter are concatenated and displayed in the Parameters field of the Submit Requests window.



From the Submit Request window, click Schedule...

In the Schedule window you can either Apply a Saved Schedule or establish a schedule by choosing one of the scheduling options.

To apply a saved schedule, click the button to display the Predefined Schedules find window.

Find the schedule you want to apply and click OK.

A message describing the schedule or further scheduling options for you to define will appear in the window.

To define your own schedule, choose one of the Run the Job... options.

The option you choose determines the type of calendar that appears for you to define your schedule.

If you wish to save your schedule for future use, click the "Save this schedule" check box.

The Save Schedule window appears. Enter a name and description for your schedule.

## Defining a Submission Schedule

### ➤ Submitting Requests

- (N) Concurrent > Requests > Submit a New Request > (B) Schedule...

August 14, 2014

Proprietary and Confidential

+ 16 +

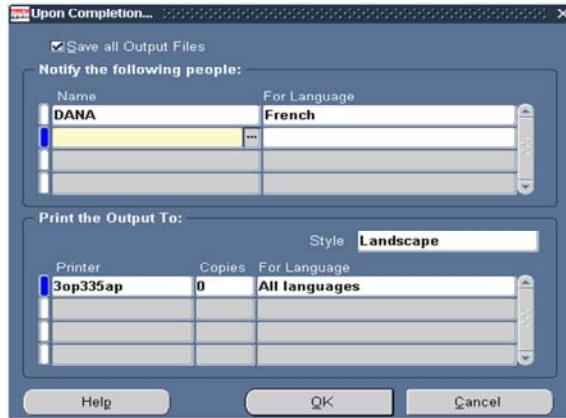


### The Schedule Window

The scheduling window provides you with several scheduling options. You can choose to reuse a schedule that you previously defined and saved, or define a new schedule. You can define your schedule to run a request as soon as possible, at a specific time, or repeatedly at specific intervals, or on specific days of the week or month.

When saving your schedule you must provide a unique name. You can also provide additional information in the Description field.

## Defining Completion Options



**IGATE**  
Speed. Agility. Inspiration.

Check the Save all Output Files check box to write your request to a file. If you want to view your report online, you must enable this box.

Click the Options... button.

Specify additional people to notify using Oracle Workflow, upon completion of this report.

Select a print style, a printer, and a number of copies.

Click OK.

Click Submit Request to submit your request.



Oracle Applications assigns a request ID to each request submission so that you can identify your request.

Use the request ID to query for your request output in the Requests window.

Oracle Applications assigns a new request ID to each resubmission of a request and displays the request ID of the previous request in the log file.

## Reprinting a Report

1. Navigate to the Requests window:  
**(N) Requests > View**
2. Query your request so that it is displayed as a record in the Requests window.
3. Select your request.
4. From the Tools menu select Reprint...



Use the Requests window to perform the following functions:

- View all submitted concurrent requests
- Check the status of requests
- Change aspects of a request's processing options
- Diagnose Errors
- Find the position of a request in the queues of available concurrent managers

## How to Use the Requests Window

- 1. Navigate to the Find Requests window:**  
(N) Requests > View
- 2. Enter specific criteria in the Find Requests window**  
**Or Click Find to display all your submitted requests.**

## Using the Requests Window

The screenshot shows the Oracle Requests window. At the top, there are three buttons: 'Refresh Data', 'Find Requests', and 'Submit a New Request...'. Below this is a table with columns: Request ID, Name, Parent, Phase, Status, and Parameters. A single row is selected, showing Request ID 279391, Name 'Customer Items Report', Parent blank, Phase 'Completed', Status 'Normal', and Parameters '511,.....N'. At the bottom of the window are several buttons: 'Hold Request', 'View Details...', 'View Output', 'Cancel Request', 'Diagnostics', and 'View Log...'. The footer of the window includes the date 'August 14, 2014', the text 'Proprietary and Confidential', and a page number '+ 22 +'.

Use the various buttons to perform tasks related to concurrent processing:

Refresh Data – Re queries the lines in the request table.

Find Requests - Displays the Find Request window to perform a search.

Submit a New Request... - Displays the Submit a New Request window.

Hold Request - Puts a request on hold if the request has not started running.

Cancel Request - Cancels a request

View Details... - Displays the Request Detail window. If the request has not already run, you can change selected fields.

Diagnostics - Displays diagnostic information about a request.

View Output - Displays an online format of the report.

View Log... - Displays information about the request such as arguments used and other technical information.



August 14, 2014

Proprietary and Confidential

+ 23 +



### Using the Requests Window (continued)

In order to use the View Output button, your program must have paper output. If the report file format you selected for your request has multiple MIME types associated with it, clicking the View Output button will display a window prompting you to select the MIME type you wish to use to view your output.

For more information on associating file formats with MIME types see:

(Help) Applied Technology > Oracle Applications System Administration >

How to View Request Status and Output > Defining the Reports Viewer.

## Viewer Options Window

- Use this window to define the MIME types for the output formats of your concurrent requests.
- These MIME types are used in viewing the reports.
- For each file format, you can associate one or more MIME types.

August 14, 2014

Proprietary and Confidential

+ 24 +



### Use the Viewer Options Window to Define MIME Types

#### (N) Install > Viewer Options

You can use one MIME type to view reports of a certain format. For example, you can view all text format reports in Microsoft Word. MIME types for supported formats for a particular user are set by several profile options. Seeded MIME types are:

Viewer: Application for HTML

Viewer: Application for PCL

Viewer: Application for PDF

Viewer: Application for PostScript

Viewer: Application for Text

This MIME type is sent to a browser window when the user views a report of that file format.

## Cancelling a Request

- 1. Navigate to the Find Requests window:**  
(N) Requests > View > (B) Find
- 2. Select the request you want to cancel and click Cancel Request.**
- 3. A decision window will prompt you to verify your action. Click Yes.**  
The status of the request will change immediately to Cancelled.

August 14, 2014

Proprietary and Confidential

+ 25 +



### Cancelling a Request That Has Not Yet Completed

If you cancel a request set, then Oracle Applications automatically cancels all requests in the set.

•

## Holding a Request

- 1. Navigate to the Find Requests window:**  
(N) Requests > View > (B) Find
- 2. Select the request you want to put on hold and click Hold Request.**
- 3. The button will change to a “Remove Hold” button and the status of the request will change to On hold.**
- 4. To remove the hold, select the request and click Remove Hold.**

**Holding a Request That Has Not Started Running**  
Only requests that have not started running can be put on hold.

## Changing Request Options

If your request has not started running, you can change how it runs and prints its output by using the Requests window.

August 14, 2014

Proprietary and Confidential

+ 27 +



1. Navigate to the Find Requests window:  
(N) Requests > View > (B) Find
2. Put your request on Hold.
3. Click View Details to display the Request Details window.
4. Change the desired options and click OK.

## Submitting Requests : Responsibility

- + Security
- + Concurrent
- + Profile
- + Application  
Load/Copy/Merge
- + Install
- Requests
  - Run
  - View
  - Set

Normal report submission uses request security group.

August 14, 2014

Proprietary and Confidential

+ 28 +

**IGATE**  
Spend Agility Imagine

### Responsibility-Based Access

This is the typical way a user submits a report. The menu prompt Run does not pass any arguments to the Submit Requests form when the prompt is chosen.

The list of values includes all the programs in the responsibility's request security group.

## Business Needs

- **Concurrent processing helps you satisfy the following business needs**
  - Keep working at your computer while running data-dependent reports and programs
  - Fully use the capacity of your hardware by executing many application tasks at once
- **Standard Request Submission lets you satisfy a related set of business needs**
  - You can use a standard interface to run your programs and reports
  - Control access to different programs and reports

August 14, 2014

Proprietary and Confidential

+ 29 +



Pass parameters from your environment to your reports and programs.

- View report output online.
- Create and run sets of reports and programs.
- Automatically run programs, reports, or request sets at specific time intervals.
- Specify whether reports and programs in a request set run sequentially or simultaneously.
- Specify whether to continue with a request set if a report or program in a sequential set fails.
- Specify alternative requests to run based on the completion status of previously run requests in a request set.
- View a log file that summarizes the completion information about all the reports and programs in a request set.

## Business Needs (Contd.)

➤ **Standard Request Submission enables you to:**

- Use a standard interface to run your programs and reports
- Control access to different reports and programs
- View report output online
- Automatically run programs, reports, or request sets at specific time intervals
- View a log file that summarizes the completion information about all the reports and programs in a request set

## Instructor Demo

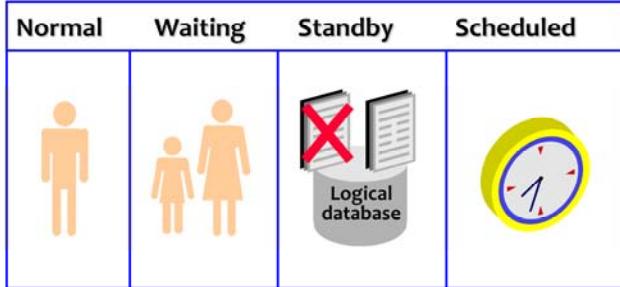
- Demo to submit a request



## Phases of Concurrent Request

- **Four Phases of a Concurrent Request**
  - The system administrator must periodically review the status of Oracle Applications programs to determine what a particular program's status is in the lifecycle of a request
- **A concurrent request has a lifecycle of either three or possibly four phases:**
  - Pending: The request is waiting to be run
  - Running: The request is running
  - Completed: The request has finished execution
  - Inactive: The request cannot yet be run

## Pending Phase



August 14, 2014

Proprietary and Confidential

+ 33 +

**IGATE**  
Speed. Agility. Inspiration.

## Pending Phase

A program in the Pending phase can be in one of four statuses:

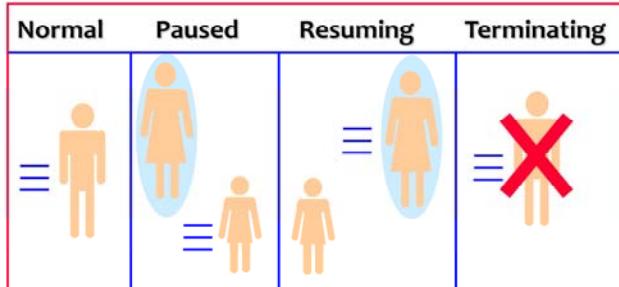
**Normal:** The program is waiting for an available manager.

**Waiting:** A child program is waiting for a parent to mark it ready to run.

**Standby:** A program is waiting for another incompatible program in the same logical database to complete.

**Scheduled:** A program's scheduled start time has not yet elapsed.

## Running Phase



August 14, 2014

Proprietary and Confidential

+ 34 +

**IGATE**  
Speed. Agility. Inspiration.

## Running Phase

A program in the Running phase can be in one of four statuses:

**Normal:** Program is in progress.

**Paused:** A parent program is waiting for one or more child programs to complete.

**Resuming:** A parent program is continuing after the completion of one or more child programs.

**Terminating:** The program is being terminated.

## Completed Phase

| Normal                                                                            | Warning                                                                           | Error                                                                             | Terminated                                                                        | Canceled                                                                          |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
|  |  |  |  |  |

August 14, 2014

Proprietary and Confidential

+ 35 +

 Spend Agility Imagine

## Completed Phase

A program in the Completed phase can be in one of five statuses:

Normal: Program completed successfully.

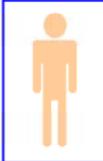
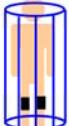
Warning: Program completed successfully but with warning messages.

Error: Program failed to complete successfully.

Terminated: A running program was terminated.

Canceled: A pending or inactive program was canceled before it started.

### Inactive Phase

| Disabled                                                                          | On Hold                                                                           | No Manager |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------|
|  |  |            |

August 14, 2014

Proprietary and Confidential

+ 36 +

**IGATE**  
Spend Agility Imagine

### Inactive Phase

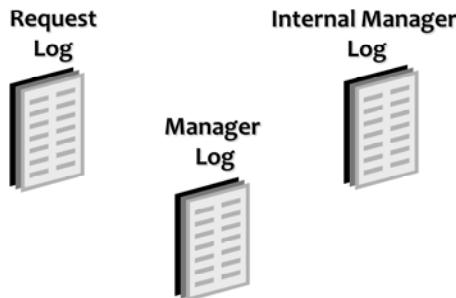
A program in the Inactive phase can be in one of three statuses:

Disabled: The requested program has not been enabled for execution.

On Hold: The requested program has been placed on hold.

No Manager: There is no manager defined to run this type of request.

## Review Log Files



August 14, 2014

Proprietary and Confidential

+ 37 +

**IGATE**  
Speed. Agility. Inspiration.

### Log File Descriptions

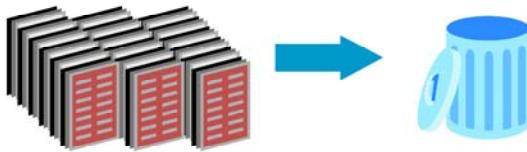
**Request Log** When a user submits a request in Oracle Applications, a concurrent manager processes the request and creates a diagnostic log file. Request Log files document the execution of a concurrent program running as the result of a concurrent request. The file contains the program parameters, the start and completion times, and any error messages. Both the user and the system administrator can access this file.

**Manager Log** Manager Log files document the performance of a concurrent manager that is running a request. The Manager Log lists each request processed by this manager in descending order by start date and time. This file is accessible by both the user and the system administrator.

**Internal Manager Log File** This file documents the performance of the Internal Concurrent Manager. It displays parameter values that are loaded when the Internal Concurrent Manager is started (STARTMGR command) and records the time that each concurrent manager is started and when each process monitor session (or PMON) cycle is initiated. During each PMON cycle, the Internal Concurrent Manager verifies the correct operation of each defined concurrent manager. Only the System Administrator can access the Internal Concurrent Manager Log file.

## Managing Log Files and Tables

- To conserve space you should periodically delete log and output files.



Use the Purge Concurrent Request and/or the Manager Data program to purge Request Log files, Concurrent Manager Log Files, and report output files.

August 14, 2014

Proprietary and Confidential

+ 38 +

**IGATE**  
Spend Agility Imagine

### Managing Log Files and Tables

**Purging Log Data** Eventually the tables holding log information consume disk space with data that may no longer be of use to you. To conserve disk space, you should periodically delete Oracle Applications log files and output files. Your company's MIS department and application users should agree on an archiving and file retention policy that is appropriate for your organization. Use the Purge Concurrent Request and/or Manager Data program to purge Request Log files, Concurrent Manager Log files, and report output files from your product directories maintained by the operating system, as well as records from Application Object Library tables that contain history information about concurrent requests and concurrent manager processes.

**Scheduling Purge Submissions** Run the Purge Concurrent Request and/or Manager Data program once and automatically resubmit the program to run at specific time intervals. Use the Parameters window to specify various criteria with which you can control the timing and frequency of program execution.

**Loss of Audit Data** Be aware that purging concurrent request information loses audit details used by the Sign-on Audit Concurrent Requests Report.

## Request Set

### Request Set

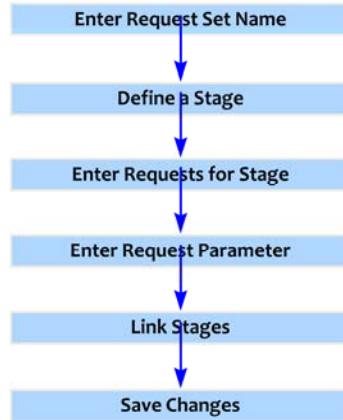
- Run option
- Run option
- Print option
- Report parameter
- Print option
- Report parameter

Programs and reports  
available to a user

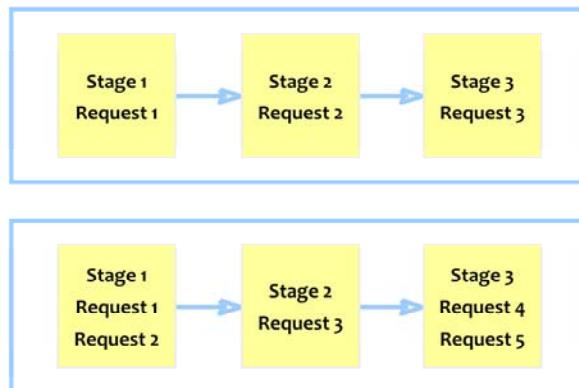
### Request Sets

A request set is a collection of concurrent programs. Request sets can also have run and/or print options, which apply to every member of the set. Programs in a request set can share parameters; therefore a parameter value needs to be entered only once for multiple programs. Any user can create a request set.

## Defining a Request Set



## Request Set Stages



August 14, 2014

Proprietary and Confidential

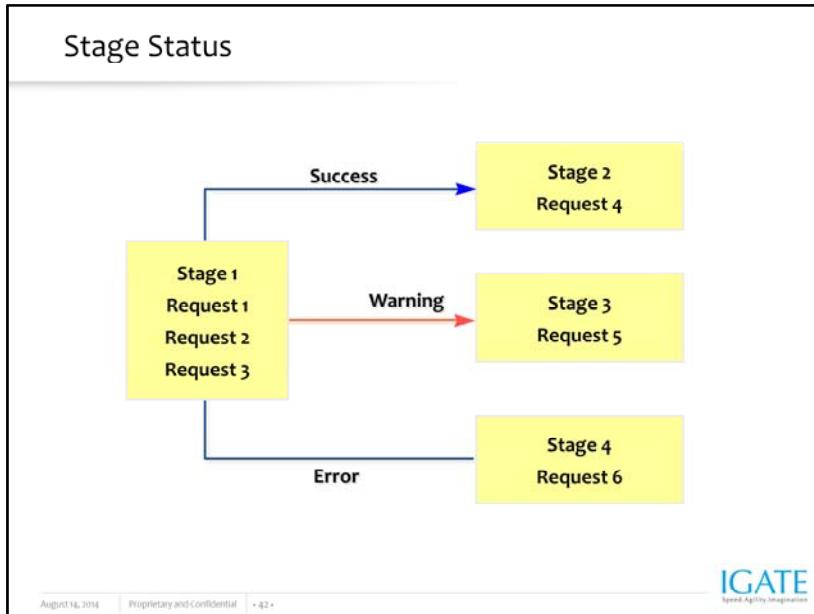
IGATE  
Agile. Agile by imagination.

## Organizing Requests with Stages

Request sets are divided into one or more stages, which are linked to determine the sequence in which your requests are run. Each stage consists of one or more requests that you want to run in parallel (at the same time in any order). For example, in the simplest request set structure, all requests are assigned to one stage. This allows all requests to run in parallel.

To run requests in sequence, assign requests to different stages and then link the stages in the order that you want the requests to run.

The concurrent manager allows only one stage in a request to run at a time. When one stage is complete, the next stage is submitted. A stage is not considered complete until all of the requests in the stage are complete. One advantage of using stages is the ability to run several requests in parallel and then move sequentially to the next stage. This enables you to create more versatile and efficient request sets.



### Linking of Stages

No restrictions on linking stages within a set

Links can point to any other stage in the set

Two or more links can point to the same stage

Request set is completed when a stage ends with no further links to be followed

## Defining Request Sets

### Step 1: Enter Request Set Name

- Navigate to the Submit a New Request window:  
(N) Requests > Set
- Enter the name of the request set.
- Enter a unique Set Code for your request set.
- Choose the application with which to associate your request set from the list of values.
- Enter a Description for your request set.

## Defining A Request Set

The Owner field defaults to your user name and can be changed only by your system administrator.

## Defining Request Sets

### Step 1: Enter Request Set Name

- Enter Active Dates From and To fields to define an effective period.
- Enable or disable Print Together as appropriate.
- Enable Allow Incompatibility as appropriate.
- Click Define Stages to define the stages for your request set.

August 14, 2014

Proprietary and Confidential

+ 44 +



### Defining A Request Set (continued)

If the current date is outside the Active Dates From and To range you define, the request set will not be available in the Submit Requests window.

Selecting the Print Together check box will send all your requests to the printer together when they complete.

Selecting the Allow Incompatibility check box allows your system administrator to specify programs that are incompatible with this request and may not run with it.

## Defining Request Sets

### Step 2: Define a Stage

- Enter a name for your stage.
- Enter a description for your stage.
- Enter a short code for the stage.
- Use the LOV in the function field to select a function.
- Enable the Return Value of this Stage Affects the Set Outcome check box as appropriate.
- Enable the Allow Incompatibility check box as appropriate.
- Click Requests to display the Stage Requests window.

August 14, 2014

Proprietary and Confidential

+ 45 +



## Stages

(N) Requests > Set > (B) Define Stages

### Defining Stages

The value for the Display Sequence field is defaulted in sequence as you enter your stages. You may change the display order of the stages by modifying the field.

The default value for the Function field is Standard Evaluation that bases its completion status on the normal completion status of the requests it contains. If you select the Return Value check box for more than one stage, the completion status of the request set will equal the completion status of the last of these stages to run.

## Defining Request Sets

### Step 3: Enter Requests for Stage

- Select the report or program that you want to include in this stage.
- Specify the number of copies of output to print, the style to print, and the printer to print to.
- Enable the Save check box as appropriate.
- Enable the Allow Stage Function to Use This Program's Results check box as appropriate.
- Click Parameters to display the Request Parameters window.

## Defining Stages (continued)

(N) Requests > Set > (B) Define Stages > (B) Requests

## Defining Request Sets

### Step 4: Enter Request Parameter

- Select the Display check box as appropriate.
- Select the Modify check box as appropriate.
- Use the Shared Parameter field to set a default value for a parameter that occurs in more than one report or program of a request set.
- Save your work.

August 14, 2014

Proprietary and Confidential

+ 47 +



#### Request Parameters

((N) Requests > Set > (B) Define Stages > (B) Requests > (B) Parameters

#### Request Parameters Window

Use the Request Parameters window to customize the parameter values of a specific request in a request set. The fields at the top of the Request Parameters window list general information about the current request set for which you can customize the parameter values. The multirow portion of the window lists the parameters for that request.

The Sequence and Prompt fields are display only. Selecting the Display check box specifies that you can see a request parameter at submission time. Selecting the Modify check box to specifies that you can insert or change the value for a request parameter at submission time.

Using the Shared Parameter field sets a default value for a parameter that occurs in more than one report or program of a request set. Use the shared parameter label to set an initial default value for all occurrences of the same parameter so that you can avoid typing the same value all over again for every occurrence of the parameter.

## Defining Request Sets

### Step 5: Link Stages

- Navigate to the Link Stages window:  
(N) Requests > Set > (B) Link Stages
- Choose the start stage from the LOV of the stages you defined for your set.
- Enter the stages that you want to run in the Success, Warning, and Error columns.
- Click Done when you are finished.

August 14, 2014

Proprietary and Confidential

+ 48 +



### Linking Stages in a Request Set

Success, Warning, and Error columns - To ensure that a particular stage follows the preceding stage regardless of the completion status, enter the desired stage in all three columns. To stop the request set if a stage ends in Error, leave the Error column blank. Any time you do not specifically indicate which stage should follow for a completion status, the request set will exit on that completion status.

## Submitting a Request Set

- Navigate to the Submit Request Set window:
- Follow the instructions for Submitting Requests presented earlier

August 14, 2014

Proprietary and Confidential

+ 49 +



### Who Can Use a Request Set

After you define a request set, it becomes your private request set. You can run it as long as you have access to a standard Submit Request window that does not limit access to specific requests.

Other users can run the request set only if your system administrator assigns the request set to their responsibility's request group. It is possible to have a request set in your request group that contains individual requests that are not in your request group, but you can only edit request sets that you own. You can add any requests in your request group to the request set. You can delete any request from the request set, regardless of whether that request is in your request group. To update information in about an individual request in the request set, however, the request set must be in your request group.

## Instructor Demo

- Demo to submit a request set



## Request Group

### Request Group

Report  
Report  
Program  
Report  
Program  
Report

Programs and reports  
available to a responsibility

#### ➤ Request Groups

A request group is a collection of reports and other concurrent programs. You use request groups to implement security at the responsibility level. Request groups are normally associated with a responsibility, in which case they are referred to as request security groups. Any user of a responsibility has access to the reports in that responsibility's request security group.

Additionally, you can define a request group to have an access code. Users must supply this code to access the reports in the coded request group.

## Grouping Concurrent Programs and Requests

### Request Group

Report  
Report  
Program  
Report  
Program  
Report

Programs and reports available to a responsibility

### Request Set

Run option  
Run option  
Print option  
Report parameter  
Print option  
Report parameter

Programs and reports available to a user

**IGATE**  
Speed. Agility. Inspiration.

## Grouping Concurrent Programs and Requests

Grouping concurrent programs and other requests together allows you to control access and streamline processing. In Oracle Applications, you group programs and requests into request groups and request sets.

### Request Groups

A request group is a collection of reports and other concurrent programs. You use request groups to implement security at the responsibility level. Request groups are normally associated with a responsibility, in which case they are referred to as request security groups. Any user of a responsibility has access to the reports in that responsibility's request security group.

Additionally, you can define a request group to have an access code. Users must supply this code to access the reports in the coded request group.

### Request Sets

Like a request group, a request set is a collection of concurrent programs. Request sets can also have run and/or print options, which apply to every member of the set. Programs in a request set can share parameters; therefore a parameter value needs to be entered only once for multiple programs. Any user can create a request set.

## Request Group

➤ Request groups can include:

- All the reports and concurrent programs owned by an application
- Individual concurrent requests
- Request sets
- Stage Functions

August 14, 2014

Proprietary and Confidential

+ 53 +



### Request Group Creation

(N) Security > Responsibility > Request

An application name is required when defining the request set. This application name and the request group name uniquely identify this request set. The application name does not prevent you from assigning reports and report sets from other applications to this group.

For more information see:

(Help) Applied Technology > Oracle Applications System Administration > Request Groups Window.

## Summary

- Use Standard Request Submission (SRS) to submit requests
- Monitor the processing of a request
- Define a request group to control user access to reports
- Customize the standard SRS processing of reports by defining a request group with a code
- Set default parameter values and share values among multiple reports by defining a request set
- Control the behavior and update of report parameters by defining a request set
- Control user access to reports and programs by specifying the ownership of a request set



### Review Question

- 1. A \_\_\_\_\_ generates a summary or detail presentation of Oracle Applications information.
- 2. A \_\_\_\_\_ is a collection of concurrent programs.



## Review Questions

- The form can be compiled and run on the client machine (True/False).
- A \_\_\_\_\_ is a list of Oracle Applications and the Oracle username assigned to each application.



## ERP- Oracle Apps

### Lesson 12: Registering Concurrent Programs in Oracle Applications

August 16, 2014

Proprietary and Confidential

+ 1 +



## Lesson Objectives

➤ **To understand the following**

- Concurrent Programs
- Steps Involved in registering a Concurrent Program
  - Executable Registration
  - Attaching Executable to the Concurrent Program
  - Attaching Concurrent Program to the Request Group



## Concurrent Programs

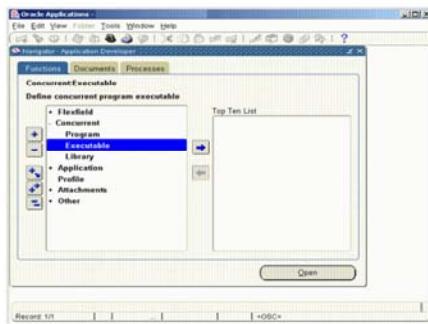
- A concurrent program is an instance of an execution file, along with parameter definitions and incompatibilities.
- Concurrent programs use concurrent program executables to locate the correct execution file.
- Several concurrent programs may use the same execution file to perform their specific tasks, each having different parameter defaults and incompatibilities.

## Concurrent Program Executable

- A concurrent program executable links an execution file or and the method used to execute it with a defined concurrent program. Under Concurrent Processing, an execution method may be a program written in a standard language, a reporting tool, or an operating system language.
- An execution method can be a PL/SQL Stored Procedure, an Oracle Tool such as Oracle Reports or SQL\*Plus, a spawned process, or an operating system host language.

## Steps to Create a Concurrent Program

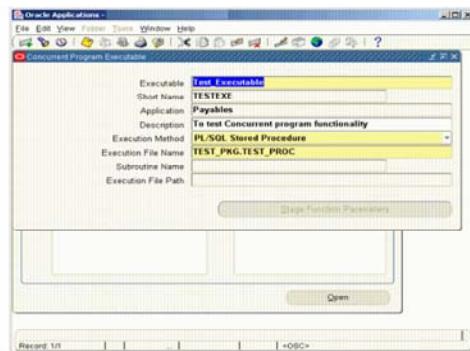
- **Executable:** To open the Executable form, follow the below mentioned navigation path:
- Application Developer > Concurrent > Executable



**IGATE**  
Speed. Agility. Imagination.

## Steps to Create a Concurrent Program

- Executable form: To define an executable select Program under concurrent from Application Developer responsibility



**IGATE**  
Speed. Agility. Ingenuity.

### Field and Description

Executable - Enter executable name here. Enter any user friendly name.

Short Name - Enter short name for your executable. This is used for mapping the executable with the concurrent program. Usually executable name are of 8 characters.

Application - Give the application to which the executable belongs to. E.g. you want some program to run from oracle payables then account payables should be entered as application name.

Description - Give the brief description of the executable in this field.

Execution Method - Here you need to choose the appropriate execution method for your executable.

Oracle Reports – used for the RDF reports

Host – used for shell scripts, basically the language of the host operating system

PL/SQL Stored procedure – used to run the stored procedure through oracle applications

SQL\*Loader – used to run the SQL loader programs

SQL\*Plus - used to run the anonymous PL/SQL blocks. It will get executed in the same fashion as you are running on SQL Plus.

Java Stored Procedure – The execution file is a Java stored procedure.

Java Concurrent Program – Used for program written in Java.

**Spawned** – used for c or pro\*c Program. Mainly used by standard oracle interfaces.

**Perl Concurrent Program** – used for programs written in CGI Perl.

**Request Set Stage Function** – PL/SQL stored function that can be used to calculate the completion statuses of request set stages.

**Immediate** – execution file is a program written to run as subroutine of the concurrent manager. Oracle doesn't recommend use of this executable type.

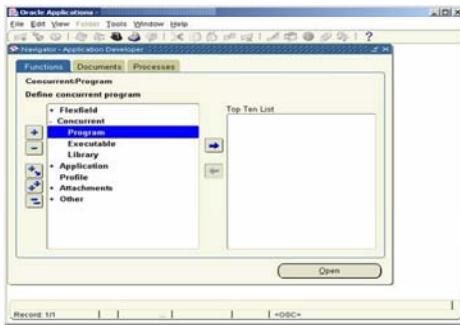
**Executable File name** – This should contain the name of the executable file. In case of PL/SQL Stored procedure or Java Stored Procedure it should be the fully defined name of the stored procedure.

**Subroutine Name** – This field is only used when executable type is spawned or immediate.

**Multi-Language function** – execution file is an MLS function that supports running concurrent program in multiple languages.

## Steps to Create a Concurrent Program

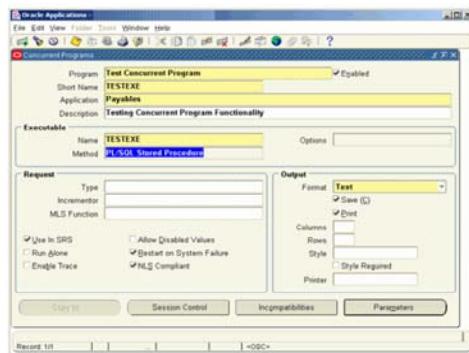
- **Concurrent Program: Select Program under Concurrent from Application Developer responsibility using the navigation: Application Developer > Concurrent > Program**



**IGATE**  
Speed. Agility. Ingenuity.

## Steps to Create a Concurrent Program

- Concurrent Program Form: Enter a program name and add the name of the executable. The executable should already be defined.



**IGATE**  
Speed. Agility. Ingenuity.

### Field and Description

**Program** – Give user friendly name for your concurrent program. This name will be displayed in Requests submission screen while submitting the requests.

**Short Name** – Give short name for concurrent program. This is used within the database tables in oracle applications. It's a common practice to have the Short name same for the executable and the concurrent program.

**Application** – Give the application to which the concurrent program belongs to. E.g you want some program to run from oracle payables then account payables should be entered as application name.

**Description** - Give the brief description of the concurrent program.

**Executable** - Enter the short name of the executable (Defined in Executables Screen) you want to attach to this concurrent program.

**Method** - will be defaulted once you enter executable short name

**Options** - will be defaulted once you enter executable short name Normally default values are given for the other fields. Following is the significance of these fields.

**Request Type** – Concurrent program can be associated to a predefined request type so that only few concurrent managers can run the program.

**Incrementor** – To be used by Oracle only.

**MLS Function** - This feature allows the program to be submitted once by the user but runs it in the multiple languages.

**Use In SRS** – Only when this check box is checked the Concurrent Program would be available in Standard Request Submission (SRS) through the request group.

**Allow Disable Value** – This will allow the disabled values in the value sets to be used while entering the values of the parameters in the Concurrent Program.

**Run Alone** – Indicates that program is incompatible with all other concurrent programs and should be run alone.

**Enable Trace** – This will enable the SQL trace for the concurrent program and will generate the trace file when concurrent program is run. Only used in development environments to check the performance of the concurrent program.

**Restart on system Failure** – This option is used to indicate that concurrent program should automatically be started when concurrent manager is restored after the system failure.

**NLS Compliant** – This box is checked if the program allows for a user to submit request of the program that will reflect a language and territory that are different from the language and territory that the users are operating in.

**Output Format** – Format in which output should be printed. Possible format values are

HTML

PDF

TEXT

PS (Post Script)

PCL(HP's Printer Control Language)

**SAVE** – Check to indicate that output should be automatically saved in an operating system file.

**PRINT** – Whether you want the output to be sent to printer for printing.

**Column / Rows** – Column and Row length of the concurrent program output. Oracle Applications uses this information to decide the print style.

**Style Required** – Print Style

**Printer** – A particular printer on which output should be sent.

## Steps to Create a Concurrent Program

- Concurrent Parameters

IGATE  
Speed. Agility. Ingenuity.

### Field and Description

Program – will be defaulted from Concurrent Program

Application – will be defaulted from Concurrent Program

Conflicts Domain - Enter the parameter which will hold the value of the conflict domain of the program.

Security Group - This field is for HRMS security only.

Seq – Enter sequence number for the parameter

Parameter – name of the parameter, will be displayed in parameter entry screen

Description – description about the usage of the parameter

Enabled – check box to enable or disable the parameter. Disable a parameter when you don't want to use it.

Value set – enter the name of the value set which you want to use to validate the value enter in the parameter field.

Description – will be defaulted from value set definition

Default Type – choose the default type for the default value of the parameter.

Possible default types are

Constant : The default value can be any literal value.

Profile : The default value is the current value in the user profile option defined in the Default Value field. Use the profile option name, not the end-user name. You do not need to include \$PROFILE\$.

SQL Statement : The default value is determined by the SQL statement you defined in the Default Value field.

Segment : The default value is the value entered in a prior segment of the same parameter window.

Display size – Enter the field length in characters for this parameter. The user sees and fills in the field in the Parameters window of the Submit Requests window.

Token - For a parameter in an Oracle Reports program, the keyword or parameter appears here. The value is case insensitive. For other types of programs, you can skip this field.

## Steps to Create a Concurrent Program

- Incompatibilities Form

IGATE  
Speed. Agility. Ingenuity.

### Field and Description

Program – Defaulted from Concurrent Programs Window

Application – Defaulted from Concurrent Programs Window

Application – Application of the concurrent program which is incompatible to the defined concurrent program.

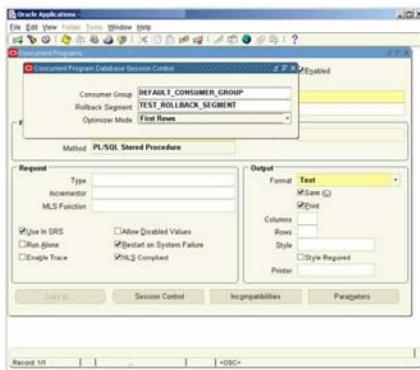
Name – Name of the concurrent program which is incompatible to the defined concurrent program.

Scope – Used to identify if concurrent program is incompatible if the program or also with its child requests.

Type - Enter Domain or Global. If you choose Domain, the incompatibility is resolved at a domain-specific level. If you choose Global, then this concurrent program will be considered globally incompatible with your concurrent program, regardless of which domain it is running in

## Steps to Create a Concurrent Program

### Session Control Form



**IGATE**  
Speed. Agility. Ingenuity.

### Field and Description

**Consumer Group** – resource consumer group of the concurrent program can be specified. A resource consumer group defines a set of users who have similar resource usage requirements. An overall resource plan specifies how resources are distributed among the different resource consumer groups. Resource consumer groups and resource plans provide a method for specifying how to partition processing resources among different users.

**Rollback Segment** – Rollback segment specified here would be used instead of the default rollback segment. If you specify a rollback segment here, your concurrent program must use the APIs FND\_CONCURRENT.AF\_COMMIT and FND\_CONCURRENT.AF\_ROLLBACK to use the specified rollback segment.

**Optimizer mode** - Optionally specify an optimizer mode. You can choose ALL\_ROWS, FIRST\_ROWS, Rules, or Choose. You would specify an optimizer mode only for a custom program that may not perform well with the default cost-based optimizer (CBO) and needs tuning. You can use a different optimizer mode until your program is tuned for CBO.

## Steps to Create a Concurrent Program

### – Copy To Form



**IGATE**  
Speed. Agility. Ingenuity.

### Field and Description

Program – Enter the program name to be copied

Short Name – Short Name of the program to be copied

Application – Application name of the concurrent program to be copied

Include Incompatible programs - Check this box if you want incompatibilities to be copied

Include Parameters – Check this box if you want parameters to be copied.

## Steps to Create a Concurrent Program

### ➤ Request Group

- A request group is a collection of reports and other concurrent programs. You use request groups to implement security at the responsibility level. Request groups are normally associated with a responsibility, in which case they are referred to as request security groups. Any user of a responsibility has access to the reports in that responsibility's request security group.
- Additionally, you can define a request group to have an access code. Users must supply this code to access the reports in the coded request group.

Request Group

Report  
Report  
Program  
Report  
Program  
Report

Programs and reports  
available to a responsibility

## Request Group

➤ **Request groups can include:**

- All the reports and concurrent programs owned by an application
- Individual concurrent requests
- Request sets
- Stage Functions

August 16, 2014

Proprietary and Confidential

+ 16 +



### Request Group Creation

(N) Security > Responsibility > Request

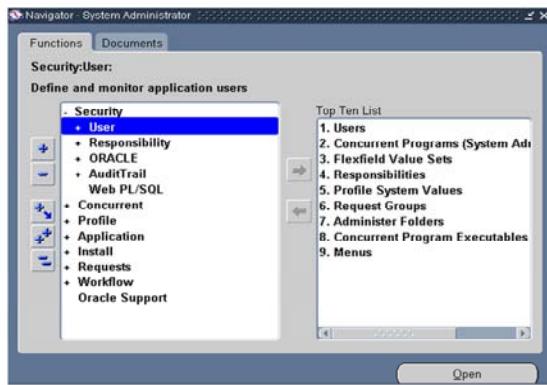
An application name is required when defining the request set. This application name and the request group name uniquely identify this request set. The application name does not prevent you from assigning reports and report sets from other applications to this group.

For more information see:

(Help) Applied Technology > Oracle Applications System Administration > Request Groups Window.

## Registering a Concurrent Program

### ➤ Select Responsibility under Security



August 16, 2014

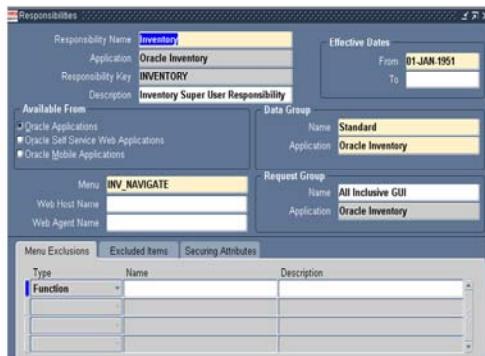
Proprietary and Confidential

+ 17 +

**IGATE**  
Speed. Agility. Ingenuity.

## Registering a Concurrent Program

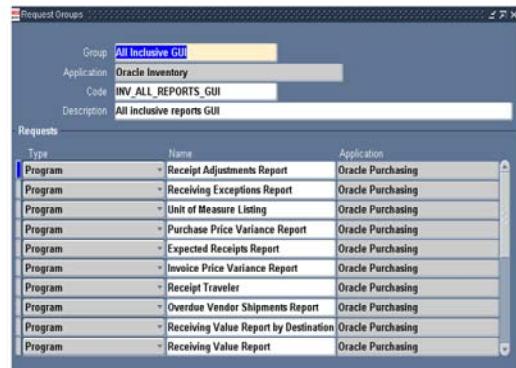
- Find the request group for the required responsibility



**IGATE**  
Speed. Agility. Imagination.

## Registering a Concurrent Program

### ➤ Add your program to the Request Group



August 14, 2014

Proprietary and Confidential

+ 19 +

**IGATE**  
Speed. Agility. Ingenuity.

## Submitting Requests : Responsibility

- Normal report submission uses request security group.

System Administrator > Functions > Security > Responsibility >  
Request > Run



### Responsibility-Based Access

This is the typical way a user submits a report. The menu prompt Run does not pass any arguments to the Submit Requests form when the prompt is chosen.

The list of values includes all the programs in the responsibility's request security group.

## Demo

- Demo to register a stored procedure



## Summary

➤ **In this lesson, we covered:**

- Concurrent Programs
- Steps Involved in registering a Concurrent Program
  - Executable Registration
  - Attaching Executable to the Concurrent Program
  - Attaching Concurrent Program to the Request Group



## Review Question

- 3. \_\_\_\_\_ Is a feature that allows you to run a non-interactive, data-dependent function, such as a report or program, simultaneously with online operations.
- 4. List the 4 phases of a concurrent request.
- 5. Programs in a request set can share \_\_\_\_\_ therefore a \_\_\_\_\_ needs to be entered only once for multiple programs.



## ERP - Oracle Apps

### Lesson 13: Oracle Application 11i Interface

August 16, 2014

Proprietary and Confidential

- 1 -



## Lesson Objectives

- What is an Interface?
- Interface Types
- Importing Information into Oracle Financials Applications
- Open interfaces
- API Interfaces
- Integration
- The benefits of using open interfaces
- The Oracle Applications Open Interface Model
- How to manage your open interface processing



## Introduction



### What is an Interface?

- Integrating one application to other applications.
- It involves data file, insertion scripts, loader scripts, temp tables. Data file is a flat file having contents with delimiter insertion scripts are used to insert data into product tables from temp table.
- Loader scripts are also called as control files which are used insert data from data file to temp table.
- Temp tables are used to transfer data from one module to another or application.

## Supported Interfaces

### ➤ Table-based Open Interfaces

- Documented in Open Interface manuals
- Not every table ending with 'INTERFACE' is supported as an open interface

### ➤ PL/SQL Object APIs

- There are also many low-level PL/SQL 'table handlers', but most are not open interfaces

### ➤ ADI/WebADI

### ➤ FNDLOAD (AOL setups)

- Be careful with Profile Options, some store sequence-generated Ids

## Migration to Oracle

➤ **Supported:**

- Entering data via the UI
- Using Open Interfaces / API's
- Using Oracle data migration utilities

➤ **Not supported:**

- Direct manipulation of the data in the database.

## Migration to Oracle

- **Setup Data (Application Object Library (AOL) data)**
  - Profiles
  - Flexfield definitions
- **Master Data**
  - Items
  - Customers
- **Transaction Data**
  - Purchase Orders
  - Invoices

## Types of Migration

- Migrate from legacy system to Oracle
- Migrate to new hardware platforms
- Migrate from older Oracle release to newer Oracle release
- Migrate multiple installs to multi-org single instance
- Migrate between instances (Test to Prod)
- Propagation of master reference data

## Interface Success Factors

- Adequate documentation of source data
- Quality of source data
- Business expertise to validate loaded data
- Understanding of current and future business needs
- Understanding of audit/reconciliation requirements
- Complete documentation for application configuration
- Carefully planned migration execution and transition
- Documented data exception handling procedures
- Realistic expectations for migration cutover time window

## Types

### ➤ Types Of Interface

- Inbound Interface
- Outbound Interface

## Interfaces



### How to get the data into the Application?

There are three alternatives to getting data into Oracle Applications :

- The data can be entered using the Applications screens
- The data can be entered using Oracle's Open Systems Interfaces
- The data can be stored in the database tables directly

## Interfaces

### ➤ Entering data using Applications screens

Entering data into the screens is the safest, but most time consuming method for getting data into the Apps. It is also prone to quality problems, if people are manually typing data each time an instance of the apps is populated. The potential for inconsistency through human error is very high. There are ways of using Excel Macros to take data from an Excel spreadsheet and stuff them into the Oracle Applications screens. The screens that can be stuffed are quite few, however. This technique works best for elemental items, like freight terms, employees, and simple lists of values.

## Interfaces

### ➤ Entering data using the Open Systems Interface

It is a good idea to use Oracle's Open System Interfaces (OSI) whenever possible. It is also a good idea to write one piece of software, however, that can function as both a conversion and an interface. Some of the OSIs require a bit of work to figure out. This is because the behavior of the OSI varies according to the unique setups entered by the user. Therefore, the behavior of the Open Item Interface is likely to be different from one Oracle Applications customer to another.

## Interfaces

### ➤ Entering data into the tables directly

This is usually not advised. It carries with it the tremendous risk of missing a referential integrity nuance or the validation of some field. If the database is populated with technically bad data, returning the system to a functional state may not be possible. This is one reason why Oracle does not advise customers updated database tables directly. Having stated that, there will be occasions where you must update the tables directly.

## Interfaces

### ➤ Designing and Developing the Interface Process

The design of an interface should permit ease of troubleshooting and maintenance. It is usually best to divide the work into distinct modules, each of which can be more easily tested. The modular design also makes changes easier and faster to implement. Below figure contains a general flow for an inbound interface.

The greatest benefits are derived from developing a standard model for all conversions and interfaces. By following a standard of developing modular code, much code can be reused. One of the greatest benefits accrue to the support of the interface once it is placed in Production. Because the design is known, troubleshooting and repair is greatly simplified. A simple standard for interface design is shown below:

## Using open interface / API

- Sample usage of open interfaces for conversion purpose:
- 3-step process:
  - Load: Get the data from the flat file into the staging table
  - Validate: Assess the quality of the data by calculating and evaluating metrics.
  - Transfer: Insert data into interface table / call APIs with values in staging table
- Run the conversion programs in the E-Business Suite for security / accessibility / traceability.

## Using open interface / API

- Load program: generic shell script taking name of SQL\*Loader control file as parameter.
- Validate program: 1 per interface.
- Transfer program: 1 per interface.

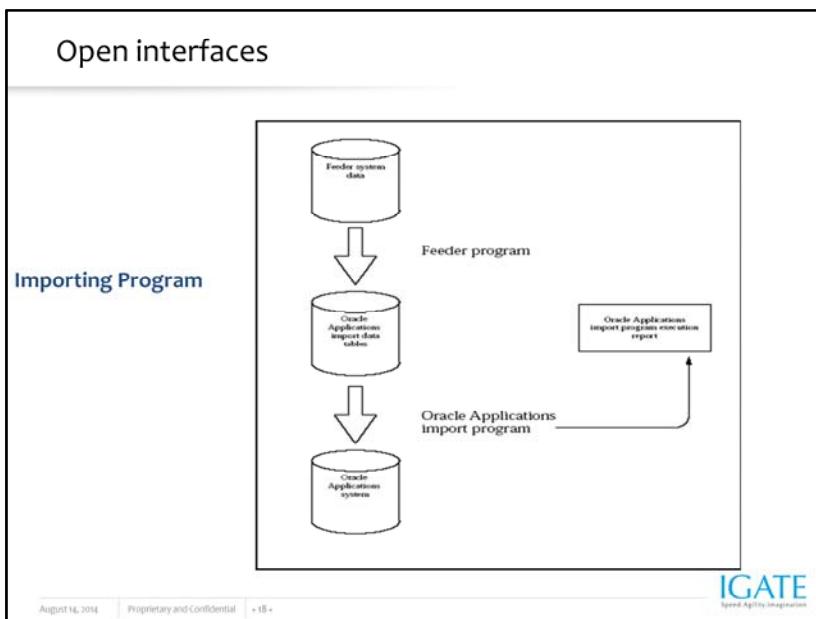


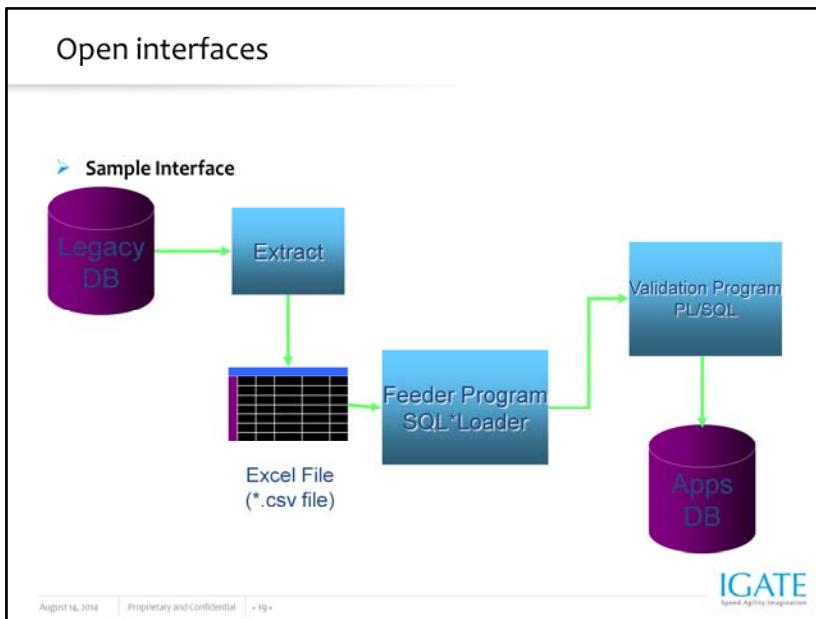
## Open interface

### Objective

The goal of your import program is to convert data from your feeder system into a standard data format that your Oracle applications can read and then convert for further modification or processing in your Oracle application.







## Open interfaces

### ➤ Feeder Program

- A custom program you use to import your detail accounting transactions from an external or feeder system into your Oracle Financials application. The type of feeder program you write depends on the environment from which you are importing data.

### ➤ Not Null Columns

- A column in which you must enter information. In other words, a column in which the value may not be null.

## Open interfaces

### ➤ Writing a Feeder Program

- The type of environment from which you want to import data determines the type of feeder program you need to write. For example, you can use SQL\*Loader to write an import program to feed data from a non-Oracle system. Or, you can write a feeder program to import historical data from your previous accounting system. Regardless of the type of feeder program you write, the output should be in standard data format that an Oracle Applications import program can use to convert your import data into your Oracle Applications system.

## Open interfaces

- SQL\*Loader is a powerful and easy-to-use tool you can to write a feeder program. SQL\*Loader lets you map elements of a regularly formatted file, such as a listing or flat file, and specify which columns of which tables to populate. Chances are, SQL\*Loader is a powerful enough tool to use for your feeder program.

## General Tips

- Clean up data before migrating
- Back up database at critical milestones
- Don't underestimate time for analysis, design, development, testing, debugging, and retesting

August 16, 2014

Proprietary and Confidential

+ 23 +



S

## Review Questions

- **Question 1:** A \_\_\_\_\_ is a collection of concurrent programs.
  
- **Question 2:** An execution method can be a \_\_\_\_\_.



## ERP- Oracle Apps

### Lesson 14: SQL Loader

August 14, 2014

Proprietary and Confidential

- 1 -



## Lesson Objectives

➤ To understand the following:

- Sql Loader
- Control File
- Control file syntax
- Log File
- Discard and Bad file



## An Introduction to SQL Loader

- SQL\*Loader is an Oracle-supplied utility that allows you to load data from a flat file into one or more database tables.
- SQL\*Loader accepts input data and processes it according to field specifications and instructions contained in the SQL\*Loader control file.
- It then rejects any data records that cause errors and discards records that do not match specified selection criteria.

SQL\*Loader allows the insertion, appendage, or replacement of records from a sequential text file

SQL\*Loader is executed from the operating system prompt, not the SQL prompt.

There must be at least one input file specified for

SQL\*Loader: The control file

Optionally the raw data may be placed in a separate file.

Output files include:

Error Log File

Bad File (Rejected records - Optional)

Discard File (Does not meet when clause Optional)

Accessing SQL\*Loader

```
§ sqlldr keyword=value [, keyword=value]
```

Warning: SID is a reserved word in the SQL\*Loader

## An Introduction to SQL Loader (contd..)

- SQL\*Loader generates a log file that documents its actions. It then passes the prepared input data to the Oracle server.
- The Oracle server rejects any data that causes Oracle errors and inserts the rest into the specified table or tables within the database.
- The basis for almost everything you do with SQL\*Loader is a file known as the control file.

## Keywords

- **userid** - username/password
- **control** - control file name
- **bad** - bad data file name
- **data** - datafile name
- **discard** - discard file name
- **discardmax** - number of discards to allow (all)
- **skip** - number of logical records to skip (0)

If the values are specified in the above order, the keywords are not necessary

If the values are specified out of sequence, the keywords must be used.

Delimiters are either commas or spaces

Any values not specified in the control line will either use the default value or  
be prompted for

Examples

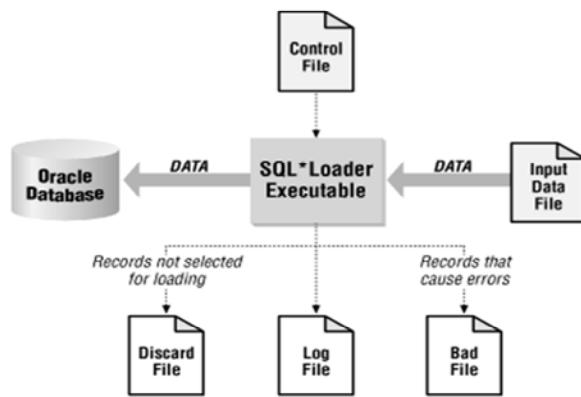
Orca: sqldr scott/tiger c.ctl l.log b.bad d.dat d.dis  
5 0 2000 999

Orca: sqldr skip=1000,userid=scott/tiger,log=l.log,  
control=c.ctl,bad=b.dat

## Keywords (contd..)

- **load** - number of logical records to load (all)
- **errors**- number of errors to allow (50)
- **rows** - number of rows to bind array (64)
- **bindsize** - size of bind array (nnnnn)
- **silent** - suppress messages during run

## The SQL\*Loader environment



## The SQL \* Loader environment

- The SQL\*Loader executable does the work of reading the input file and loading the data.
- The input file contains the data to be loaded, and the database receives the data.
- Output of the SQL\*Loader is an Oracle database (where the data is loaded), a log file, a bad file, and potentially a discard file.

## The SQL \* Loader control file

- The SQL\*Loader control file is the key to any load process.
- The control file provides the following information to SQL\*Loader:
  - The name and location of the input data file
  - The format of the records in the input data file
  - The name of the table or tables to be loaded
  - The correspondence between the fields in the input record and the columns in the database tables being loaded.
  - Selection criteria defining which records from the input file contain data to be inserted into the destination database tables.
  - The names and locations of the bad file and the discard file

Control files are created via the system text editor.

Control file commands are “free field”

Control file commands may be in upper or lowercase

Comments can be inserted by preceding them with a double dash (--)

SQL\*Loader keywords can be enclosed in quotes if you want to use them as table or column names.

OPTIONS (options)

LOAD [DATA]

[{INFILE | INDDN} {filename | \*} ]  
[STREAM | RECORD | FIXED len |  
BLOCKSIZE size] | VARIABLE [len] ]  
[{BADFILE | BADDN } filename]  
[{DISCARDFILE | DISCARDDN } filename]  
[{DISCARDS | DISCARDMAX} n]

[{INDDN | INFILE} . . . ] [APPEND | REPLACE | INSERT]  
[RECLEN n]  
[{CONCATENATE n |  
CONTINUEIF {[THIS | NEXT] (start [:end])}  
| LAST}  
operator {'char-str' | x'hex-str'} } ]

```
INTO TABLE tablename
[APPEND | REPLACE | INSERT]
[WHEN field-condition [AND
field-condition ...]]
[FIELDS [delimiter-spec]]
(column-name
{RECNUM | CONSTANT value |
SEQUENCE ({n | MAX | COUNT}
[, increment]) |
[POSITION ({start[:end] | * [+n]})]
datatype-spec
[NULLIF field condition]
[DEFAULTIF field-condition]}
[...])
[INTO TABLE] [BEGINDATA data]
```

## The SQL \* Loader control file (contd..)

- It's also possible for the control file to contain the actual data to be loaded. This is sometimes done when small amounts of data need to be distributed to many sites, because it reduces (to just one file) the number of files that need to be passed around. If the data to be loaded is contained in the control file, then there is no need for a separate data file.

```
Ex:- LOAD DATA INFILE * INTO TABLE dept FIELDS
TERMINATED BY ','
OPTIONALLY ENCLOSED BY """
(deptno, dname, loc)
BEGINDATA
12,RESEARCH,"SARATOGA"
10,ACCOUNTING,"CLEVELAND"
11,"ART","SALEM"
13,FINANCE,"BOSTON"
```

Options: LOAD, SKIP, ERRORS, BINSIZE, SILENT  
Example OPTIONS (SKIP=100, LOAD=200)

Field Condition:  
{(start [:end]) | column name } operator  
{'char string' | X'hex byte string'}  
Example: NAME = 'Bill Collector'

Datatype Spec:

CHAR [(length)] [delimiter spec]

DATE [(length)] [mask] [delimiter spec]

VARCHAR [(length)]

SMALLINT

DOUBLE

GRAPHIC [EXTERNAL] [(length)]

{INTEGER | FLOAT | DECIMAL} EXTERNAL [(length)]

INTEGER [(length)]

FLOAT [(length)]

DECIMAL (digits [,precision])

VARGRAPHIC [(length)]

Delimiter Spec:

[TERMINATED [BY] {WHITESPACE | [X] ‘char’}]

[[OPTIONALLY] ENCLOSED [BY] [X] ‘char’]

## Control File Syntax Notes

- Specifying Data Files
- Must use the filename statement optionally followed by the File Read Mode
- Filename: To specify the name of the file containing the raw data.
- {INFILE | INDDN} {filename | \*} filename: The name of the file containing the data.
- \*: The data is in the control file

## Control File Syntax Notes (contd..)

### ➤ **Read File Mode:**

- Tells SQL\*Loader how to open the data file and retrieve physical records from it. This provide a great deal of flexibility on a wide variety of Operating Systems

### ➤ **STREAM | RECORD |**

**FIXED len [BLOCKSIZE size] VARIABLE [len]**

### **Specifying Multiple Data Files:**

- INFILE DATA1.DAT
- INFILE DATA2.DAT

## Control File Syntax Notes (contd..)

- Specifying Bad Files: (records with formatting errors or ORACLE errors)
- [BADFILE | BADDN] filespec
- Specifying Discard Files: (Records which meet none of the filtering requirements)
- [DISCARDFILE | DISCARDDN] filespec

## Control File Syntax Notes (contd..)

- Specifying The Position of a Data Field
- POSITION ({start[:end] | \* | [+n] })
- Start: Starting column number  
End: Ending column number. If omitted the length of the datafield is determined from the datatype in the datafile.
- Maybe specified: start:end or start-end

## Specifying Control File Syntax Notes (contd..)

- \* : Indicates that the data field follows immediately after the previous field. If an \* is used for the first data field, that data field is assumed to begin in datatype format.
- +n : An offset, specified as +n, may be used with \* to offset this field from the previous one.
- POSITION may be omitted entirely. If so, the position specification for the data field is the same as if POSITION (\*) had been used.

### Examples:

```
SITEID POSITION(*) SMALLINT
SITELOC POSITION(*) INTEGER
ENAME POSITION (1:20) CHAR
EMPNO POSITION (22-26) INTEGER EXTERNAL
ALLOW POSITION (*+2) INTEGER EXTERNAL TERMINATED
BY “/”
```

## The Log file

- **The log file is a record of SQL\*Loader's activities during a load session. It contains information such as the following:**
  - The names of the control file, log file, bad file, discard file, and data file.
  - The values of several command-line parameters
  - A detailed breakdown of the fields and datatypes in the data file that was loaded.
  - Error messages for records that cause errors.
  - Messages indicating when records have been discarded.
  - A summary of the load that includes the number of logical records read from the data file, the number of rows rejected because of errors, the number of rows discarded because of selection criteria, and the elapsed time of the load.

## The Bad File and the Discard File

- Whenever you insert data into a database, you run the risk of that insert failing because of some type of error such as violating the Integrity Constraint.
- Whenever SQL\*Loader encounters a database error while trying to load a record, it writes that record to a file known as the *bad* file.
- Discard files, on the other hand, are used to hold records that do not meet selection criteria specified in the SQL\*Loader control file.
- Records that do not meet the specified criteria are not loaded, and are instead written to a file known as the *discard* file.

## Conventional Path Load VS Direct Path Load

- SQL\*Loader provides two methods to load data: **Conventional Path**, which uses a SQL INSERT statement with a bind array, and **Direct Path**, which loads data directly into a database. Direct path load is much faster than conventional path load.

## Summary

➤ **In this lesson we have covered:**

- Sql Loader
- Control File
- Control File syntax
- Log File
- Discard and Bad file



## Review Question

- **Question 1:** SQL\*Loader is an Oracle-supplied utility that allows you to load data from a \_\_\_\_\_ into one or more database tables.
- Option 1: Control file
  - Option 2: Flat file
  - Option 3: Log file
  - Option 4: Discard file
- **Question 2.** The basis for almost everything you do with SQL\*Loader is a file known as the \_\_\_\_\_ .
- Option 1: Control file
  - Option 2: Flat file
  - Option 3: Log file
  - Option 4: Discard file



## Summary

In this lesson, you should have learned how to:

- What is an Interface?
- Interface Types
- Importing Information into Oracle Financials Applications
- Open interfaces
- API Interfaces
- Integration
- The benefits of using open interfaces
- The Oracle Applications Open Interface Model
- How to manage your open interface processing



## ERP- Oracle Apps

### Lesson 15: Object Standards

## Lesson Objectives

➤ **At the end of the session you will be able to:**

- Understand the correct usage of data types
- Work with Oracle Applications Views
- Sequences
- Understand Table Registration
- Know What are WHO columns



## Data Types to be avoided

- Avoid creating tables with the LONG, LONG RAW, or RAW datatypes
- Within Oracle Forms, you cannot search using wildcards on any column of these types. Use VARCHAR2(2000) columns instead

## Examples of wrong data type usage

- Setting a value 'Generator' to a variable/field whose field size is less than 9 would raise an exception
- Wrong format specification in the type conversion functions like to\_date or to\_char would raise an exception
- Storing a value 123.4567 in a field having two decimal precision and using it in the further calculations would result in unexpected result

## Views

- Complex blocks are based on views while simple setup blocks are based on tables
- You do not need to code any POST-QUERY logic to populate non-database fields
- You do not need to code PRE-QUERY logic to implement query-by-example for non-database fields
- This allows you to centralize and share LOV definitions. An LOV view is usually simpler than a block view, since it includes fewer denormalized columns, and contains only valid rows of data

## Views (Contd... )

- You should also base your Lists of Values (LOVs) on views
- Whenever performance is an issue and your table has foreign keys, you should define a view to improve performance
- Views allow a single SQL statement to process the foreign keys, reducing parses by the server, and reducing network traffic

## Views: Advantages

➤ **Views are extremely desirable because:**

- They speed development, as developers can build on logic they already encapsulated
- They modularize code, often meaning that a correction or enhancement can be made in a single location
- They reduce network traffic
- They are often useful for reporting or other activities
- They can be easily and centrally patched at a customer site

## Views: Restrictions

- Avoid creating views that are used by only one SQL statement
- Creating a view that is only used by a single procedure increases maintenance load because both the code containing the SQL statement and the view must be maintained

## Triggers on Views

- When basing a block on a view, you must code ON-INSERT, ON-UPDATE, ON-DELETE, and ON-LOCK triggers to insert, update, delete, and lock the root table instead of the view
- Single table views do not require triggers for inserting, updating, deleting and locking. Set the block Key Mode to Unique
- Single table views do not require a ROW\_ID column

August 14, 2014

Proprietary and Confidential

+ 9 +



Your view should then include all of the columns in the root table, including the WHO columns, and denormalized foreign key information.

Suggestion: You only need to include the ROWID column if an Oracle Forms block is based on this view. The Oracle Forms field corresponding to the ROW\_ID pseudo-column should use the ROW\_ID property class.

### Change Block Key Mode

In Oracle Forms, you need to change the block Key Mode property to Non-Updateable to turn off Oracle Forms default ROWID references for blocks based on views. Specify the primary keys for your view by setting the item level property Primary Key to True.

For example, a view based on the EMP table has the columns ROW\_ID, EMPNO, ENAME, DEPTNO, and DNAME. Set the Key Mode property of block EMP\_V to Non-Updateable, and set the Primary Key property of EMPNO to True.

If your block is based on a table, the block Key Mode should be Unique.

## Sequences

- Use a NUMBER datatype to store sequence values within PL/SQL
- Use each sequence to supply unique ID values for one column of one table
- The maximum value for an ascending sequence is  $10^{27}$

August 14, 2014

Proprietary and Confidential

+ 10 +



Sequences can be created through front end of Apps or at the database level. Sequences created using the front end can be assigned to a particular field on the form. Seeded sequences are also available.

The purpose of backend sequences would be to provide unique ids to certain fields, if they are not part of the source data.

For more details, refer to “Sequences” (See Page 3-10) from Oracle Applications Developer’s Guide.

## Don'ts on Sequences

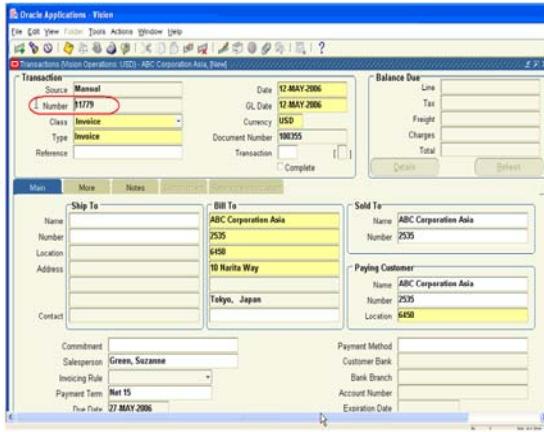
- Do Not Limit the Range of Your Sequences
- Do Not Use the FND\_UNIQUE\_IDENTIFIER\_CONTROL Table

Do not create sequences that wrap using the CYCLE option or that have a specified MAXVALUE. The total range of sequences is so great that the upper limits realistically are never encountered. In general, do not design sequences that wrap or have limited ranges.

Do not rely on the FND\_UNIQUE\_IDENTIFIER\_CONTROL table to supply sequential values. Use a sequence or the sequential numbering package instead. The FND\_UNIQUE\_IDENTIFIER\_CONTROL table is obsolete and should not have any rows for objects in your product. Additionally, do not create application-specific versions of the FND table to replace the FND\_UNIQUE\_IDENTIFIER\_CONTROL table.

### Create a Sequence

➤ The Transaction number is generated through a sequence, when the record is saved



The screenshot shows the Oracle Applications Vision Transaction screen. The transaction number field, labeled 'L Number' with value '31779', is circled in red. Other fields visible include 'Source' (Manual), 'Date' (12-MAY-2006), 'GL Date' (12 MAY 2006), 'Currency' (USD), 'Document Number' (10005), and 'Reference'. The 'Main' tab is selected. The 'Bill To' section shows ABC Corporation Asia with address 2535 6450 10 Narita Way Tokyo, Japan. The 'Sold To' section shows ABC Corporation Asia with number 2535. The 'Paying Customer' section also shows ABC Corporation Asia with number 2535. The 'Payment Method' section includes fields for Customer Bank, Bank Branch, Account Number, and Expiration Date.

IGATE  
Speed. Agility. Imagination

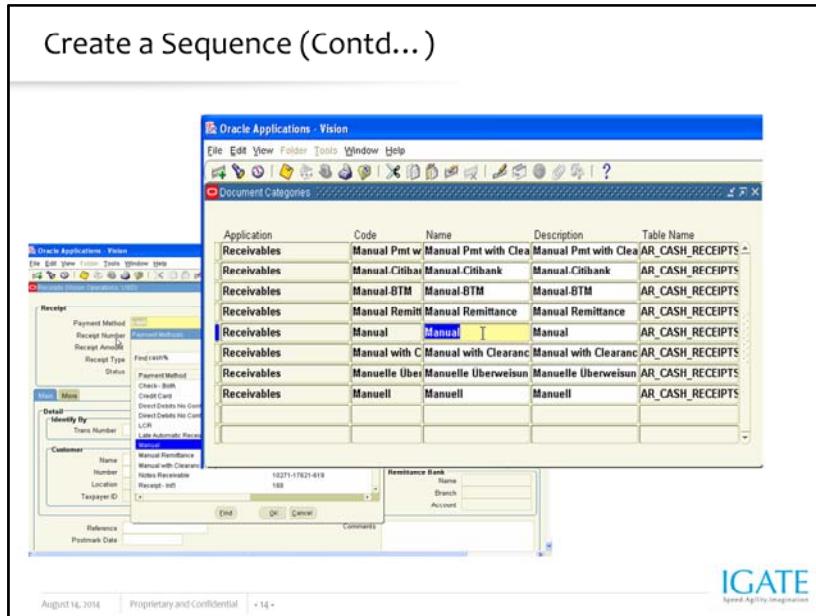
To navigate to the form shown above, select Receivables, Vision Operations (USA) -> Transactions -> Transactions. The Transaction number is generated through a sequence.

## Create a Sequence (Contd...)

- Shown in the figure are existing sequences.
- You can also create a new sequence
- Navigate to System Administrator -> Document-> Define

| Name            | Application | From        | To          | Type      | Message                  | Initial Value |
|-----------------|-------------|-------------|-------------|-----------|--------------------------|---------------|
| ACHDE           | Receivables | 01-JAN-1990 |             | Automatic | <input type="checkbox"/> | 1             |
| ACHNL           | Receivables | 01-JAN-1990 |             | Automatic | <input type="checkbox"/> | 1             |
| Automatic Rece  | Receivables | 01-JAN-2001 | 01-JAN-2001 | Automatic | <input type="checkbox"/> | 20000         |
| Automatic Rece  | Receivables | 01-JAN-1990 |             | Automatic | <input type="checkbox"/> | 20000         |
| BE AR ADJ       | Receivables | 01-JAN-1998 |             | Automatic | <input type="checkbox"/> | 1             |
| BE AR INV       | Receivables | 01-JAN-1998 |             | Automatic | <input type="checkbox"/> | 1             |
| BE RECEIPT      | Receivables | 01-JAN-1998 |             | Automatic | <input type="checkbox"/> | 1             |
| BE Receipt - US | Receivables | 01-JAN-1990 |             | Automatic | <input type="checkbox"/> | 1000          |
| BR_Adjust_NU    | Receivables | 01-JAN-2002 |             | Automatic | <input type="checkbox"/> | 1             |
| Test            | Receivables | 16-MAY-2006 |             | Gapless   | <input type="checkbox"/> | 1             |

A new sequence Test is created and its type is set to Gapless, i.e. which indicates that there would be no gaps in the sequence numbers.

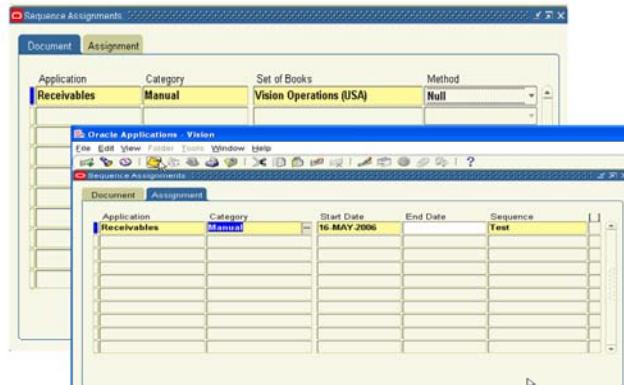


The window on the top can be accessed by selecting System Administrator -> Document-> Categories. This window displays the categories. Manual payment is one such category, associated with Application : Receivables and Table name : AR\_CASH\_RECEIPTS\_ALL

Various payment Methods (Categories) are as shown in the Receipts form on the left. The Receipts form can be accessed by selecting : Receivables, Vision Operations (USA) -> Receipts.

The sequence created is created for manual type payments.

## Create a Sequence (Contd...)



**IGATE**  
Speed. Agility. Imagination

The sequence is assigned to the Manual category in the Sequence Assignments form. This can be accessed by selecting System Administrator -> Document-> Sequence Assignments.

Note that initially the value under Method is set to Null, indicating that no sequence is yet assigned. To assign a sequence, click on the Assignment

### Create a Sequence (Contd...)

The screenshot shows the Oracle Applications - Voices Receipts screen. The 'Receipt' section contains fields for Payment Method (Manual), Receipt Number (123), Receipt Amount (USD 123.00), Receipt Type (Standard), and Status (Cleared). The 'Balances' section shows a table with columns for Unapplied, Applied, On Account, Unapplied, Cash Claims, and Prepayments, all with values of \$123.00. The 'Main' tab is selected, showing a 'Document Number' field with '123' highlighted and circled in red. The 'Tax Code' field is empty. The 'Reversal' section includes fields for Category and Reason. The 'Remittance' section includes fields for Bills Receivable, Deposit Date (16-MAY-2006), Batch, Override (Don't Allow), and Bank Currency (USD). The 'Notes Receivable' section includes fields for Issuer Name, Issue Date, Issuer Bank, and Bank Branch. At the bottom left, it says 'August 14, 2014 Proprietary and Confidential 16+' and at the bottom right is the IGATE logo.

In the Receipts form, when a record is created for Manual type of payment, the Document Number is automatically generated through the sequence.

## Table Registration

- You register your custom application tables using a PL/SQL routine in the AD\_DD package
- Flexfields and Oracle Alert are the only features or products that depend on this information
- Therefore you only need to register those tables (and all of their columns) that will be used with flexfields or Oracle Alert
- You can also use the AD\_DD API to delete the registrations of tables and columns from Oracle Application Object Library tables should you later modify your tables



August 14, 2014 | Proprietary and Confidential | +17+

### Sample Procedures in the Package

```
procedure register_table (p_appl_short_name in varchar2,
p_tab_name in varchar2,
p_tab_type in varchar2,
p_next_extent in number default 512,
p_pct_free in number default 10,
p_pct_used in number default 70);
```

```
procedure register_column (p_appl_short_name in
varchar2,
p_tab_name in varchar2,
p_col_name in varchar2,
p_col_seq in number,
p_col_type in varchar2,
p_col_width in number,
p_nullable in varchar2,
p_translate in varchar2);
```

```
EXECUTE ad_dd.register_table('FND',
'CUST_FLEX_TEST', 'T', 8, 10, 90);
```

Here is an example of using the AD\_DD package to register a flexfield table and its columns:

## WHO Columns

- The Record History (WHO) feature reports information about who created or updated rows in Oracle Applications tables
- If you add special WHO columns to your tables and WHO logic to your forms and stored procedures, your users can track changes made to their data
- By looking at WHO columns, users can differentiate between changes made by forms and changes made by concurrent programs

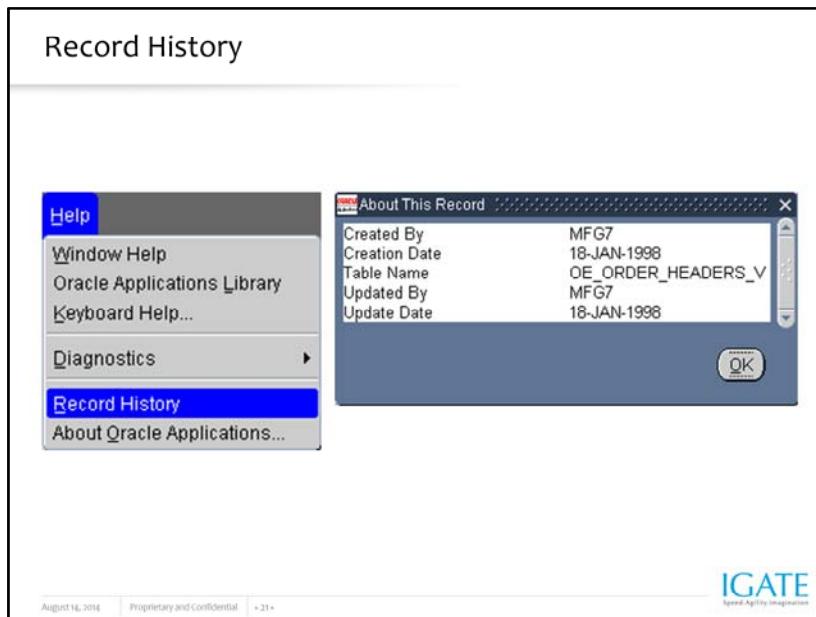
## WHO Columns (Contd...)

- You represent each of the WHO columns as hidden fields in each block of your form (corresponding to the WHO columns in each underlying table)
- Call FND\_STANDARD.SET\_WHO in PRE-UPDATE and PRE-INSERT to populate these fields
- Set the CREATED\_BY and CREATION\_DATE columns only when you insert a row (using FND\_STANDARD.SET\_WHO for a form)

### WHO Columns (Contd...)

- Apply the CREATION\_OR\_LAST\_UPDATE\_DATE property class to the form fields CREATION\_DATE and LAST\_UPDATE\_DATE
- This property classes sets the correct attributes for these fields, including the data type and width

For more details, refer to “Tracking Data Changes with Record History (WHO)” (See Page 3-2) from Oracle Applications Developer’s Guide.

IGATE  
Speed. Agility. Imagination.

## Tables Without Record History Information

- For blocks that are based on a table, but do not have Record History information, disable the menu entry HELP -> ABOUT\_THIS\_RECORD
- Code a block-level WHEN-NEW-BLOCK-INSTANCE trigger (style "Override") with these lines: app\_standard.event('WHEN-NEW-BLOCK-INSTANCE'); app\_special.enable('ABOUT', PROPERTY\_OFF);

August 14, 2014 | Proprietary and Confidential | +22+



Use the APP\_SPECIAL package to enable and customize menu entries and buttons on the toolbar. You can use the APP\_SPECIAL.ENABLE procedure to dynamically control menu items, if the behavior you need is not provided automatically. First, determine if the default menu control handles the menu item in question, and ensure that there really is a need to override the default behaviors.

If the menu item is not controlled by the default menu control, use any appropriate trigger (typically PRE-BLOCK or WHEN-NEW-BLOCK-INSTANCE), adding the code:

Turn the menu item back on when you leave (typically POST-BLOCK) by calling:

```
app_special.enable('the menu item', PROPERTY_OFF|ON);
```

Include the full name of the menu item in this call, for example:

```
app_special.enable('the menu item', PROPERTY_ON|OFF);
```

You can determine the full names of the menu items by copying FNDMENU from the AU\_TOP/resource/<language> area and opening the copy to examine the menu items.

```
app_special.enable('CLEAR.FIELD', PROPERTY_OFF);
```

If the menu item is controlled by the default menu control and you want to modify its behavior (enable or disable it), create the field- or block-level trigger listed (either WHEN-NEW-BLOCK-INSTANCE, WHEN-NEW-RECORD-INSTANCE, or WHEN-NEW-ITEM-INSTANCE). Set the trigger Execution Hierarchy to "Override" and add the following code:

```
app_standard.event('TRIGGER_NAME');
app_special.enable('Menu_item',
PROPERTY_OFF|ON);
```

The item will be correctly reset in other blocks by the default menu control, so it is not necessary to reset it when leaving the block, record, or item.

## Summary

➤ **In this session we covered:**

- The Correct Usage of Data Types
- Oracle Applications Views
- Sequences
- Table Registration Process
- What are WHO columns



## Review Question

- **Question 3:** Whenever SQL\*Loader encounters a database error while trying to load a record, it writes that record to a file known as the \_\_\_\_\_
  - Option 1: Control file
  - Option 2: Flat file
  - Option 3: Bad file
  - Option 4: Discard file
  
- **Question 4:** Records that do not meet the specified criteria are not loaded and are instead written to a file known as \_\_\_\_\_
  - Option 1: Control file
  - Option 2: Flat file
  - Option 3: Bad file
  - Option 4: Discard file



# ERP – Oracle Apps

## Lesson 16: Coding Standards

August 14, 2014

Proprietary and Confidential

• 1 •



## Lesson Objectives

➤ To understand the coding standards of following topics:

- Basic of Coding Standards
- Handlers
- Triggers
- SQL
- PL/SQL Code



## Coding Standards

- Oracle Applications is built by Oracle Corporation developers using the standards given in:
  - Oracle Applications Developer's Guide
  - Oracle Applications User Interface Standards for Forms-Based Products
- Follow the standards to build custom application code that integrates with and has the same look and feel as Oracle Applications
- The libraries and procedures that are packaged with Oracle Applications all assume adherence to these standards

### Importance of these Standards

The coding standards described in Oracle Applications

Developer's Guide, together with the user interface standards described in the Oracle Applications User Interface Standards for Forms-Based Products, are used by Oracle Corporation developers to build Oracle Applications. If you want to build custom application code that integrates with and has the same look and feel as Oracle Applications, you must follow these standards. If you do not follow these standards exactly as they are presented, you may not achieve an acceptable result.

The libraries and procedures that are packaged with Oracle Applications all assume adherence to these standards. In fact, since the behavior of Oracle Forms, the Oracle Applications standard libraries, and the standards are so tightly linked, a deviation from standards that appears to be minor may in fact have far-reaching and unpredictable results. Therefore, its recommend that when you develop custom application code, you follow the standards exactly as they are described in these manuals.

## Points for developer

- Oracle Applications coding standards are guided by the following principles:
  - Code must be readable to be maintained
  - Tools such as Oracle Forms and PL/SQL are used whenever possible
  - Fast performance over the World Wide Web is critical
  - Platform-specific code should be avoided except where absolutely necessary
  - Reusable objects should be employed wherever possible

Oracle Applications coding standards are guided by the following principles:

- Code must be readable to be maintained
- Tools such as Oracle Forms and PL/SQL are used whenever possible (avoid complex user exits using other coding languages)
- Fast performance over the World Wide Web (the web) is critical
- Platform-specific code should be avoided except where absolutely necessary
- Reusable objects should be employed wherever possible

## Coding with Handlers

- Handlers are groups of packaged procedures
- Easier to develop, maintain, and debug
- Call the handlers from the triggers by passing the name of the trigger as an argument
- Types of procedures are:
  - Item handlers
  - Event handlers
  - Table handlers
  - Business rules
- Can reside in program units, libraries or stored packages

August 14, 2014

Proprietary and Confidential

+ 5 +



Oracle Applications uses groups of packaged procedures, called handlers, to organize PL/SQL code in forms so that it is easier to develop, maintain, and debug.

In Oracle Forms, code is placed in triggers, which execute the code when that trigger event occurs. Implementing complex logic may require scattering its code across multiple triggers. Because code in triggers is not located in one place, it cannot be written or reviewed comprehensively, making development, maintenance, and debugging more difficult. To determine what code and events affect a particular item, a developer must scan many triggers throughout the form. Code that affects multiple items can be extremely difficult to trace.

To centralize the code so it is easier to develop, maintain, and debug, place the code in packaged procedures and call those procedures from the triggers. Pass the name of the trigger as an argument for the procedure to process. This scheme allows the code for a single business rule to be associated with multiple trigger points, but to reside in a single location.

There are different kinds of procedures for the different kinds of code you write: item handlers, event handlers, table handlers, and business rules. Code resides in these procedures; do not put any code in the triggers other than calls to the procedures.

Handlers may reside in program units in the form itself, in form libraries, or in stored packages in the database as appropriate.

## Coding with Handlers

- **Item Handlers**
  - Is a PL/SQL procedure that encapsulates all of the code that acts upon an item
- **Event Handlers**
  - Is a PL/SQL procedure that encapsulates all of the code that acts upon an event
- **Table Handlers**
  - Encapsulates all of the code that manages interactions between a block and its base table
- **Business Rules**
  - A business rule describes complex data behavior

August 14, 2014

Proprietary and Confidential

+ 6 +



### Item Handlers

An item handler is a PL/SQL procedure that encapsulates all of the code that acts upon an item. Most of the validation, defaulting, and behavior logic for an item is typically in an item handler.

### Event Handlers

An event handler is a PL/SQL procedure that encapsulates all of the code that acts upon an event. Usually event handlers exist to satisfy requirements of either Oracle Forms or the Oracle Applications User Interface Standards for Forms-Based Products, as opposed to particular business requirements for a product.

### Table Handlers

A table handler encapsulates all of the code that manages interactions between a block and its base table. When an updatable block is based on a view, supply procedures to manage the insert, update, lock and delete. Referential integrity checks often require additional procedures. Table handlers typically reside in the database but may also reside on the forms server depending on size and the amount of interaction with the database.

### Business Rules

A business rule describes complex data behavior. For example, one business rule is: "A discount cannot be greater than 10% if the current credit rating of the buyer is less than 'Good'." Another business rule is: "A Need-By Date is required if a requisition is made for an inventory item."

## Coding for Performance

- Applications must avoid overloading the network that connects desktop client, application server and database servers
- Oracle Applications are designed by employing the following coding standards:
  - Use database stored procedures when extensive SQL is required
  - Code all non-SQL logic on the client side where possible
  - Cache data on the client side where practical
  - Base blocks on views that denormalize foreign key information where practical

August 14, 2014

Proprietary and Confidential

+ 7 +



### Performance

Performance is a critical issue in any application. Applications must avoid overloading the network that connects desktop client, application server, and database servers, since often it is network performance that most influences users' perceptions of application performance.

Oracle Applications are designed to minimize network traffic on all tiers. For example, they try to limit network round trips to one per user-distinguishable event by employing the following coding standards:

Use database stored procedures when extensive SQL is required

Code all non-SQL logic on the client side where possible

Cache data on the client side where practical

Base blocks on views that denormalize foreign key information where practical

## Coding for Web Compatibility

- **Avoid using the following features in forms as they are not applicable in Oracle Applications architecture:**
  - ActiveX, VBX, OCX, OLE, DDE
  - Timers other than one-millisecond timers
  - WHEN-MOUSE-MOVE, WHEN-MOUSE-ENTER/LEAVE and WHEN-WINDOW-ACTIVATED/DEACTIVATED triggers
  - Open File dialog box
  - Combo boxes
  - Text\_IO and HOST built-in routines

August 14, 2014

Proprietary and Confidential

+ 8 +



## Coding for Web Compatibility

Following Oracle Applications standards carefully will help ensure that your forms can be deployed on the Web.

You should avoid using the following features in your forms, as they are not applicable in this architecture:

ActiveX, VBX, OCX, OLE, DDE (Microsoft Windows-specific features that would not be available for a browser running on a Macintosh, for example, and cannot be displayed to users from within the browser)

Timers other than one-millisecond timers (one-millisecond timers are treated as timers that fire immediately)

WHEN-MOUSE-MOVE, WHEN-MOUSE-ENTER/LEAVE and WHEN-WINDOW-ACTIVATED/DEACTIVATED triggers

Open File dialog box – It would open a file on the applications server, rather than on the client machine (where the browser is) as a user might expect

Combo boxes – Our standards do not use combo boxes anyhow

Text\_IO and HOST built-in routines – These would take place on the applications server, rather than on the client machine (where the browser is) as a user might expect

## Settings for Form Generation

### ➤ Mandatory Settings for Form Generation

- NLS\_LANG variable is set to the required language
- Set FORMS6o\_PATH environment variable to include any directory that contains forms, files, or libraries
- Ensure to use the character set used by Oracle Applications installation

### ➤ Recommended Setting for Form Development

- Set the environment variable ORACLE\_APPLICATIONS to TRUE before starting Oracle Forms Developer



### Mandatory Settings for Form Generation

At form generation time, make sure that NLS\_LANG variable is set to the required language in the Windows NT registry or environment file (for Unix). Also ensure that the character set specified is the character set being used for Oracle Applications installation.

You must also set the value of FORMS6o\_PATH environment variable in environment file (Windows registry) to include any directory that contains forms, files, or libraries to be used to develop and generate forms.

Specifically, include a path to the <\$AU\_TOP>/forms/US directory to be able to find all referenced forms, and a path to the <\$AU\_TOP>/resource directory to be able to find the Oracle Applications library files you need.

### Recommended Setting for Form Development

Oracle Forms Developer allows referenced objects to be overridden in the local form. Oracle Forms Developer also does not normally provide any indication that an object is referenced unless you set a special environment variable. Set the environment variable ORACLE\_APPLICATIONS to TRUE before starting Oracle Forms Developer. This setting allows you to see the reference markers (little flags with an "R" in them) on referenced objects so you can avoid changing referenced objects unintentionally. Any object referenced from the APPSTAND form must never be changed.

## Setting Object Characteristics

➤ **The characteristics of most form objects may be set in the following ways:**

- Inherited through property classes
- At the discretion of the developer during form design
- At runtime, by calling standard library routines

## Setting Object Characteristics

The characteristics of most form objects, including modules, windows, canvases, blocks, regions, and items may be set in the following ways:

- Inherited through property classes, which cause certain properties to be identical in all forms (such as canvas visual attributes)
- At the discretion of the developer during form design (such as window sizes)
- At runtime, by calling standard library routines (such as window positions)

## Using PL/SQL in Applications

➤ You use PL/SQL to:

- Develop procedural extensions to your forms and reports quickly and easily.
- Modularize your application code.
- Optimize your application code.

August 14, 2014

Proprietary and Confidential

+ 11 +



You can use PL/SQL procedures as part of an application that you build around Oracle Applications. By following the coding standards, you can create a PL/SQL procedure that integrates seamlessly with your application and with Oracle Applications.

You use PL/SQL to:

Develop procedural extensions to your forms and reports quickly and easily

Modularize your application code to speed development and improve maintainability

Optimize your application code to reduce network traffic and improve overall performance

You can use PL/SQL to develop procedural extensions to custom forms and reports you create with Oracle tools. For example, to develop a form that follows Oracle Applications standards, organize form code into PL/SQL business rule procedures, item handlers, event handlers, and table handlers. Put very little PL/SQL code directly into form triggers because those triggers do not represent a logical model; they are simply event points that Oracle Forms provides for invoking procedural code. If you put most of the code in packaged PL/SQL procedures, and then call those procedures from triggers, you will have modular form code that is easy to develop and maintain.

You can also use PL/SQL to develop concurrent programs or stored procedures that are called from concurrent programs.

## PL/SQL Coding Standards

- Always Use Packages.
- Package size must not exceed 10K.
- New Procedures or Functions should be added to the end of an Existing Package.
- Always specify field names completely by including the block name i.e., BLOCK.FIELD\_NAME instead of just FIELD\_NAME

August 14, 2014

Proprietary and Confidential

+12+



### Always Use Packages

PL/SQL procedures should always be defined within packages. Create a package for each block of a form, or other logical grouping of code. A client-side (Oracle Forms) PL/SQL program unit's (package specification or body or stand-alone procedure.) source code and compiled code together must be less than 64K. This implies that the source code for a program unit cannot exceed 10K.

### Adding New Procedures to Existing Packages

When you add new procedures or functions to existing packages (either stored in the database or in Oracle Forms libraries), you should usually add them to the end of the package (and package specification). If you add new procedures to the middle of the package specification and package, you must regenerate every form that references the package, or those forms may get ORA-4062 errors.

### Using Field Names in Client-Side PL/SQL Packages

Always specify field names completely by including the block name i.e., BLOCK.FIELD\_NAME instead of just FIELD\_NAME. Just specifying the field name makes Oracle Forms to scan through the entire list of fields for each block in the form to locate the specified field and check if its name is ambiguous, potentially degrading your form performance. If the block name is included, Oracle Forms searches only the fields in that block and stops when it finds a match. Moreover, if you ever add more blocks, existing code continues to work since you specified field names unambiguously.

## PL/SQL Coding Standards

- Pass field names to procedures and use COPY to update field values instead of using IN OUT or OUT parameters.
- Use DEFAULT instead of ":=" when declaring default values for parameters.
- Use ":=" instead of DEFAULT when declaring values for constant variables.

August 14, 2014

Proprietary and Confidential

+13+



### Field Names in Procedure Parameters

Pass field names to procedures and use COPY to update field values instead of using IN OUT or OUT parameters. This method prevents a field from being marked as changed whether or not you actually modify it in your procedure. Any parameter declared as OUT is always written to when the procedure exits normally.

For example, declare a procedure as

`test(my_var VARCHAR2 IN)`

and call it as `test('block.field')` instead of declaring the procedure as

`test(my_var VARCHAR2 IN OUT)`

and calling it as `test(:block.field)`.

### Using DEFAULT

Use DEFAULT instead of ":=" when declaring default values for your parameters. DEFAULT is more precise because you are defaulting the values; the calling procedure can override the values.

Conversely, use ":=" instead of DEFAULT when declaring values for your constant variables. Using ":" is more precise because you are assigning the values, not defaulting them; the values cannot be overridden.

## PL/SQL Coding Standards

- **SET\_<OBJECT>\_PROPERTY** built-in (or AOL equivalent) should use object IDs.
- To check if a value is equal to NULL, use the operator "is" instead of "=".

August 14, 2014

Proprietary and Confidential

+ 14 +



### Use Object IDs

Any code that changes multiple properties of an object using the **SET\_<OBJECT>\_PROPERTY** built-in (or the Oracle Application Object Library equivalent) should use object IDs. First use the appropriate **FIND\_<OBJECT>** built-in to get the ID, then pass the ID to the **SET\_<OBJECT>\_PROPERTY** built-in. Consider storing the ID in a package global so that its retrieved only once while the form is running.

### Handling NULL Value Equivalence

Handle NULL values very carefully in PL/SQL. For example, if a := NULL and b := NULL, the expression (a = b) evaluates to FALSE. In any "=" expression where one of the terms is NULL, the whole expression will resolve to FALSE.

For this reason, to check if a value is equal to NULL, you must use the operator "is" instead. If you're comparing two values where either of the values could be equal to NULL, you should write the expression like this: ((a = b) or ((a is null) and (b is null)))

## PL/SQL Coding Standards

| Behavior                                    | Oracle Forms Global | PL/SQL Package Global | Oracle Forms Parameter |
|---------------------------------------------|---------------------|-----------------------|------------------------|
| Can be created at Design time               |                     | Y                     | Y                      |
| Can be created at runtime                   | Y                   |                       |                        |
| Accessible across all forms                 | Y                   |                       |                        |
| Accessible from attached libraries          | Y                   | *                     | Y                      |
| Support specific datatypes                  | **                  | Y                     | Y                      |
| Have declarative defaults                   |                     |                       | Y                      |
| Can be referenced indirectly                | Y                   |                       | Y                      |
| Can be specified on command line            |                     |                       | Y                      |
| Must be erased to recover memory            | Y                   |                       |                        |
| <b>Can be used in any Oracle Forms code</b> | Y                   |                       | Y                      |

August 14, 2014

Proprietary and Confidential

+ 15 +



### Global Variables

Oracle Forms Developer and PL/SQL support different types of global variables:

Oracle Forms Global: a variable in the "global" pseudo-block of a form

PL/SQL Package Global: a global defined in the specification of a package

Oracle Forms Parameter: a variable created within the Oracle Forms Designer as a Parameter

The above table lists the characteristics of each type of variable, and enables you to select the type most appropriate for your code.

\* A package variable defined in a form is not visible to any attached library; a variable defined in an attached library is visible to the form. (An Oracle Forms Global is visible to an attached library)

\*\* Always CHAR(255)

## PL/SQL Coding Standards

- Minimizing the number of round trips is key to ensuring good performance
- Decide whether PL/SQL procedures should reside on the server or on the client based on whichever results in the fewest number of network round trips

August 14, 2014

Proprietary and Confidential

+ 16 +



### Database Server Side versus Client Side

Performance is a critical aspect of any application. Because network round trips are very costly in a typical internet environment, minimizing the number of round trips is key to ensuring good performance.

You should decide whether your PL/SQL procedures reside on the server or on the client based on whichever results in the fewest number of network round trips. Here are some guidelines:

Procedures that call Oracle Forms built-ins (more generally, client built-ins) must reside on the client

Procedures that reference fields directly, either as :block.field or via NAME\_IN/COPY, must reside on the client.

You can avoid referencing fields directly by accepting field values or names as parameters to your PL/SQL procedures, which also improves your code's modularity

If a procedure contains three or more SQL statements, or becomes very complicated, the procedure usually belongs on the server

Procedures that perform no SQL and that need no database access should reside wherever they are needed

If a procedure is called from the server, it must reside on the server. If a procedure is called from both client and server, it should be defined in both places, unless the procedure is very complicated and double maintenance is too costly. In the latter case, the procedure should reside on the server.

## Formatting PL/SQL Code

- Within a package, define private variables first, then private procedures, and finally public procedures.
- Always end procedures and packages by following the "end" statement with the procedure or package name.
- Indent code logically using increments of two spaces; indent comments to align with the code.
- Use uppercase and lowercase to improve the readability of code.
- Use uppercase for reserved words and lowercase for everything else.
- Use --for commenting single line /\* ... \*/ for commenting multiple lines
- Avoid deeply nested IF-THEN-ELSE condition control.
- Create nested PL/SQL blocks within a procedure only when there is specific exception handling to be trapped.



## Formatting PL/SQL Code

This section contains recommendations for formatting PL/SQL code.

Within a package, define private variables first, then private procedures, and finally public procedures.

Always end procedures and packages by following the "end" statement with the procedure or package name to help delineate procedures.

Indent code logically. Using increments of two spaces provides an easy way to track your nested cases.

Indent SQL statements as follows:

### Example

```
DECLARE
 CURSOR employees IS
 SELECT empno
 FROM emp
 WHERE deptno = 10
 AND ename IN ('WASHINGTON', 'MONROE')
 AND mgr = 2701;
```

## Formatting PL/SQL Code

- Use uppercase for reserved words and lowercase for everything else.
- Use --for commenting single line /\* .. \*/ for commenting multiple lines
- Avoid deeply nested IF-THEN-ELSE condition control.
- Create nested PL/SQL blocks within a procedure only when there is specific exception handling to be trapped.

August 14, 2014

Proprietary and Confidential

+18+



Use "--" to start comments so that you can easily comment out large portions of code during debugging with /\* ... \*/

Indent comments to align with the code being commented. When commenting out code, start the comment delimiter in the leftmost column. When the code is clearly no longer needed, remove it entirely.

Use uppercase and lowercase to improve the readability of your code (PL/SQL is case-insensitive). As a guideline, use uppercase for reserved words and lowercase for everything else.

Avoid deeply nested IF-THEN-ELSE condition control. Use IF-THEN-ELSIF instead.

### Example of Bad Style

```
IF ... THEN ... ELSE
 IF ... THEN ... ELSE
 IF ... THEN ... ELSE
 END IF
 END IF;
 END IF;
```

### Example of Good Style

Only create nested PL/SQL blocks (BEGIN/END pairs) within a procedure when there is specific exception handling you need to trap

```
IF ... THEN ...
ELSIF ... THEN ...
ELSIF ... THEN ...
ELSIF ... THEN ...
ELSE ...
END IF;
```

## Exception Handling

- Use FND\_MESSAGE to display an error message, then RAISE FORM\_TRIGGER\_FAILURE to stop processing.
- Use the package procedures FND\_MESSAGE.SET\_NAME to set a message, and APP\_EXCEPTION.RAISE\_EXCEPTION to stop processing
- When testing FORM\_SUCCESS, FORM\_FAILURE, or FORM\_FATAL note that their values may be changed by a built-in in another trigger that is fired as a result of your built-in

### Errors in Oracle Forms PL/SQL

If a failure occurs in Oracle Forms PL/SQL and you want to stop further processing, use FND\_MESSAGE to display an error message, then RAISE FORM\_TRIGGER\_FAILURE to stop processing:

```
IF (error_condition) THEN
 fnd_message.set_name(appl_short_name,
 message_name);
 fnd_message.error;
 RAISE FORM_TRIGGER_FAILURE;
END IF;
```

Note: RAISE FORM\_TRIGGER\_FAILURE causes processing to stop quietly without error notification, you must display any messages yourself using FND\_MESSAGE before raising the exception.

## SQL Coding Guidelines

- All select statements should use an explicit cursor .
- Write an exception handler to handle NO\_DATA\_FOUND exception.
- Do the check in PL/SQL code not in WHERE clause.
- Explicitly check the value of SQL%NOTFOUND.

August 14, 2014

Proprietary and Confidential

+ 20 +



Follow these guidelines for all SQL that you code:

Use "select from DUAL" instead of "select from SYS.DUAL". Do not use SYSTEM.DUAL

All SELECT statements should use an explicit cursor. Implicit SELECT statements actually cause 2 fetches to execute: one to get the data, and one to check for the TOO\_MANY\_ROWS exception. Avoid this by FETCHing just a single record from an explicit cursor

To SELECT into a procedure parameter, declare the parameter as IN OUT, whether or not the parameter value is referenced, unless the parameter is a field

A single-row SELECT that returns no rows raises the exception NO\_DATA\_FOUND. An INSERT, UPDATE, or DELETE that affects no rows does not raise an exception. Explicitly check the value of SQL%NOTFOUND if no rows is an error.

To handle NO\_DATA\_FOUND exceptions, write an exception handler. Do not code COUNT statements to detect the existence of rows unless that is your only concern.

When checking the value of a field or PL/SQL variable against a literal, do the check in PL/SQL code, not in a WHERE clause.

Do not check for errors due to database integrity problems. For example, if a correct database would have a table SYS.DUAL with exactly one row in it, you do not need to check if SYS.DUAL has zero or more than one row or if SYS.DUAL exists

## Triggers in Forms

- Execution style for all block or field level should either be Override or before.
- Set Show Keys property to true for all key-triggers.
- Set Show keys description property to null.
- Avoid using these triggers in the Oracle forms.
  - WHEN-MOUSE-MOVE, WHEN-MOUSE-ENTER/LEAVE and WHEN-WINDOW-ACTIVATED/DEACTIVATED triggers



August 14, 2014 | Proprietary and Confidential | + 21 +

Follow these general rules for triggers in the forms.

### Execution Style

The 'Execution Style' for all block or field level triggers should either be Override or Before. In general, use style Before, since usually the form-level version of the trigger should also fire. The exception is if there is a flexfield call in the form-level POST-QUERY trigger, but you reset the query status of the block in the block level POST-QUERY. In that case, the block-level POST-QUERY should use Execution Style After.

### KEY-Trigger Properties

Set the "Show Keys" property to True for all KEY- triggers you code, except those that you are disabling (which should have "Show Keys" set to False). Always set the "Show Keys Description" property to NULL.

### WHEN-CREATE-RECORD in Dynamic Query-Only Mode

The WHEN-CREATE-RECORD trigger fires even when the block does not allow inserts. Check if the block allows insert if you have logic in this trigger and your block may dynamically have insert-allowed "FALSE":

```
IF GET_ITEM_PROPERTY('<BLOCK>',
INSERT_ALLOWED) = FALSE THEN
 null;
ELSE
 <your logic here>;
END IF;
```

## Replacements for Oracle Forms Built-ins

- Use DO\_KEY built-in to invoke the key trigger logic.
- Do Not Use CALL\_FORM.
- The following Oracle Forms built-ins have equivalent APPCORE routines that provide additional functionality.
  - OPEN\_FORM - Do\_Key('OPEN\_FORM');
  - SET\_ITEM\_PROPERTY - APP\_ITEM\_PROPERTY.SET\_PROPERTY.
  - GET\_ITEM\_PROPERTY - APP\_ITEM\_PROPERTY.GET\_PROPERTY.
  - VALIDATE - APP\_STANDARD.APP\_VALIDATE

These standards require that certain built-ins be avoided entirely, or "wrapper" routines be called in their place. For many built-ins, there are multiple methods of invocation. Call the built-in directly, to give the standard forms behavior. For some built-ins, there are standard Oracle Applications behaviors, which are invoked by calling APP\_STANDARD.EVENT.

Many of these built-ins have a key and a KEY- trigger associated with them. If there is any additional logic which has been added to the KEY- trigger that you want to take advantage of, you can invoke the trigger by using the DO\_KEY built-in. This is the same result you would get if the user pressed the associated key.

Do not use CALL\_FORM Oracle Forms built-in.

This built-in is incompatible with OPEN\_FORM, which is used by Oracle Applications routines. Use FND\_FUNCTION.EXECUTE instead of either CALL\_FORM or OPEN\_FORM whenever a form to be opened programmatically. Using FND\_FUNCTION.EXECUTE allow the form to open without bypassing Oracle Applications security, and takes care of finding the correct directory path for the form.

## Review Question

- Which all are the data types to be avoided?
- What are views? Why is an LOV view simpler than a block view?
- Give some advantages of views.
- What are WHO columns?



## **ERP – Oracle Apps**

**Lesson 17: Best Practices - Aim Methodology**

## Lesson Objectives

- What is AIM?
- Activities Supported – AIM
- AIM - Phases
- AIM – Data Conversion (CV) Methodology
- Data Conversion (CV) – Objective & Steps
- Documentations – Data Conversions
- Data Conversion – Processes
- Usage & Deliverables – Each Conversion Documents
- Customization Process - Documents



## What is AIM?

- **AIM – Application Implementation Method**
- **Oracle Full life-cycle approach for implementing Oracle Apps package.**
  - AIM is comprised of well-defined processes that can be managed in several ways to guide you through an application implementation project
  - AIM provides the tools needed to effectively and efficiently plan, conduct, and control project steps to successfully implement new business systems

August 16, 2014

Proprietary and Confidential



Oracle's Application Implementation Method (AIM) is a proven approach for implementing Oracle Applications.

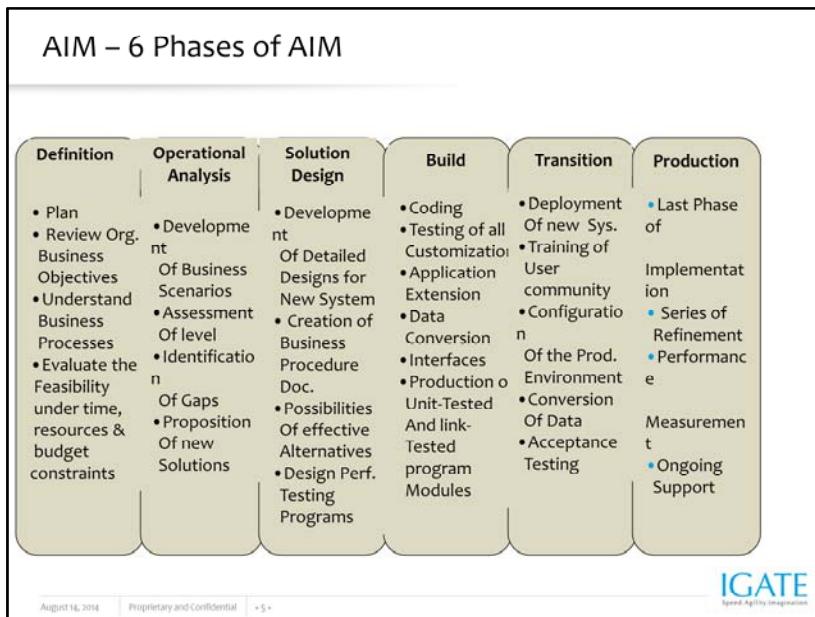
AIM is comprised of well-defined processes that can be managed in several ways to guide you through an application implementation project. AIM provides the tools needed to effectively and efficiently plan, conduct, and control project steps to successfully implement new business systems.

AIM is very broad in its support of all of the activities within your implementation project including:

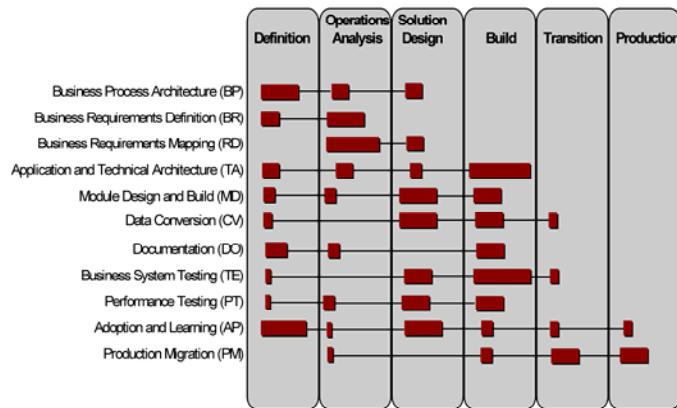
- .Planning
- .requirements definition
- .business process alignment and modeling
- .customization
- .interfaces and integration between systems
- .data conversion
- .organization change management including specific support for executive, management, and user groups
- .application and technical architecture including network and server design
- .reporting and information access systems
- .security and access control

## AIM Processes

- Planning
- Requirement Definition
- Business Process alignment and modeling
- Customization
- Interfaces and integration between systems
- Data Conversion
- Organization Change Management
- Application and technical architecture
- Reporting and information access systems
- Security and access control

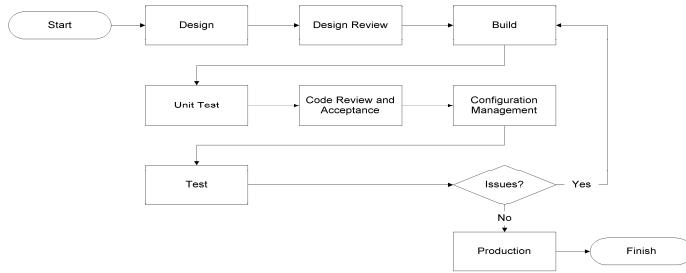


## Phases and Processes Relationship



## Data Conversions Approach

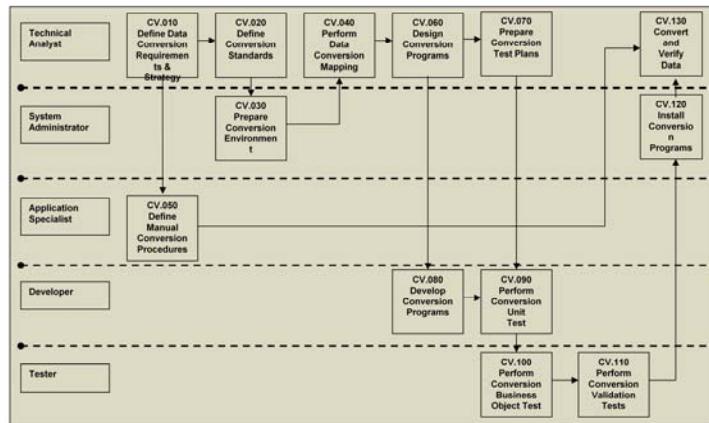
- Oracle Conversion Projects mainly involves migration of data from Legacy system to Oracle Applications or from lower version of Ora Apps to higher version. This diagram describes the project execution model.



## Documentations – Data Conversion

- CV.010 - Define Data Conversion Requirements and Strategy
- CV.020 - Define Conversion Standards
- CV.030 - Prepare Conversion Environment
- CV.040 - Perform Conversion Data Mapping
- CV.050 - Define Manual Conversion Procedures
- CV.060 - Design Conversion Programs
- CV.070 - Prepare Conversion Test Plans
- CV.080 - Develop Conversion Programs
- CV.090 - Perform Conversion Unit Tests
- CV.100 - Perform Conversion Business Object Tests
- CV.110 - Perform Conversion Validation Tests
- CV.120 - Install Conversion Programs
- CV.130 - Convert and Verify Data

## Data Conversion (CV) - Processes



## Terms Used for the Key Responsibilities

- **Technical Analyst :** Performs conversion mapping and design conversion modules
- **System Administrator:** Support the data conversion and provide assistance as needed for accessing the legacy systems.
- **Application Specialist:** Provides Knowledge and guidance regarding application functionality
- **Developer:** Code Conversion Modules. Develop Conversion unit test specifications and perform the conversion unit test. Perform the data conversion, if required.
- **Tester:** Help with conversion business object and validation testing to verify data correctness.
- **Business Object:** Physical or logical object of significance to a business e.g. vendors, customers, people, assignments etc.

## CV.010 – Define Data Conversion Requirements and Strategy

- Define the scope of the conversion project
- Conversion Objective and approach
- Prepare a strategy for converting information from the legacy systems to the new application environment
- The task provides a roadmap for performing the conversion of data from the legacy system to the new Oracle System
- Deliverable  
Data Conversion Requirements and Strategy

## CV.020 – Define Conversion Standards

- **Creation of the Conversion Standards** the Conversion Team needs to follow when performing conversion tasks.
- Documents the file structure and naming conventions for the legacy and target systems as well as for any automated tools which would help in facilitate the conversion process.
- **Deliverable**
  - Conversion Standards

## CV.030 – Prepare Conversion Environment

- **Creation of the Conversion Environment** which includes the hardware platforms, servers and other software which are required to support the design and build activities of conversion.
- **Deliverable**
  - Conversion Environment

## CV.040 – Perform Conversion Data Mapping

- **Mapping the data elements from the legacy systems to the target Oracle Applications**
- **Identification – Data Sources and target tables and columns**
- **Identification – Data Defaults, Validations, processing, translation, filters and foreign key rules.**
  
- **Deliverable**  
Conversion Data Mapping
  - provides the detailed data mapping and extract file layout for each legacy data elements one plans to convert to Oracle.

## CV.050 – Define Manual Conversion Procedures

- Define the plan to convert the business objects which require manual conversion.

- **Deliverable**

Manual Conversion Procedures

- Defines how to manually convert a business object
- Discuss the impact if one does not successfully convert the business objects in the required time frame.

## CV.060 – Define Data Conversion Requirements and Strategy

### ➤ Design and Document the Conversion Programs

### ➤ Deliverable

#### Conversion Program Designs

- Defines the key assumptions, rules and logic needed to create the conversion Modules.

## CV.070 – Develop Conversion Programs

- Outlines the testing plans for the unit, business object and validation testing for conversion.

- **Deliverable**

Conversion Test Plans

- Provides plans for:
  - Conversion Unit
  - Business Object
  - Validation Tests

## CV.080 – Develop Conversion Programs

- Creation of the conversion programs that performs all of the functions required to convert legacy business objects to the target Oracle Applications.
- Conversion Programs of each Business Object consist:
  - Download Program
  - 1. Interface Table Creation Program
  - 2. Upload Program
  - 3. Translation Program
  - 4. Interface and Validation Program
- Deliverable  
Conversion Program

## CV.090 – Perform Conversion Unit Tests

- **Purpose** – to test the conversion programs to verify that all programs work without errors and according to the conversion testing specifications pre-defined in the CV.070.
- **Deliverable**  
Unit-Tested Conversion Programs.
  - It includes conversion program source code which has been tested to verify the processing logic of each program module functions without errors.

## CV.100 – Perform Conversion Business Object Tests

- **Testing the complete conversion of each business object by executing all conversion modules for the business object in the appropriate sequence and thus verifying so that the resulting data is correct.**
- **Deliverable**  
Business Object-Tested Conversion Programs

## CV.110 – Perform Conversion Validation Tests

- **Validation of the target applications function correctly with the converted business objects**

- **Deliverable**

- Validation-Tested Conversion Programs

- It includes conversion programs that have been tested to verify that the resulting converted legacy data performs correctly in the entire suite of Oracle Applications.

## CV.120 – Install Conversion Programs

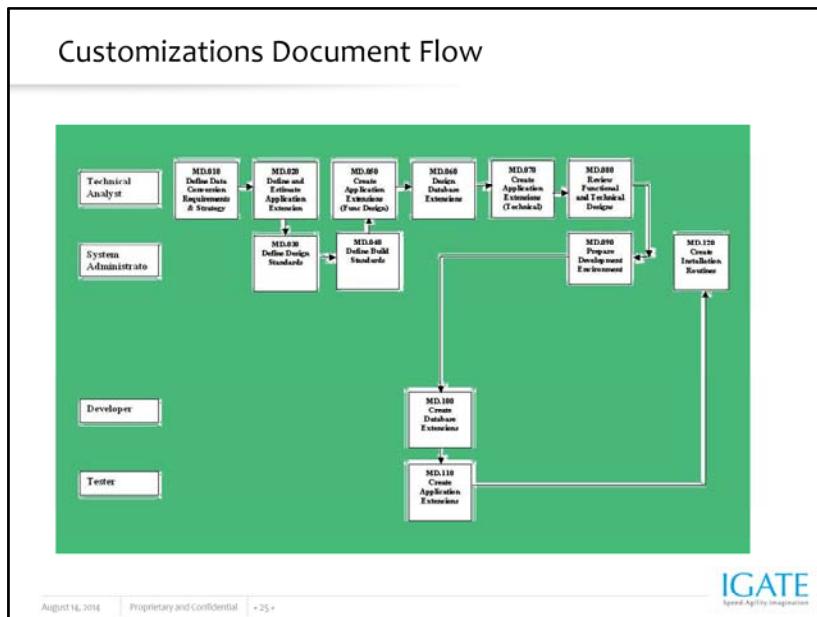
- Perform and document the installation of the conversion programs in the production environment.
- **Deliverable**  
Installed Conversion Programs

## CV.130 – Convert and Verify Data

- Conversion and Migration of the production data from the old system to the new Oracle Production Environment
- Completion of task → data is ready for production use.
- Deliverable  
Converted and Verified Data

## Customization Documents

- MD.010 - Define Application Extension Strategy
- MD.020 - Define and Estimate Application Extensions
- MD.030 - Define Design Standards
- MD.040 - Define Build Standards
- MD.050 - Create Application Extensions Functional Design
- MD.060 - Design Database Extensions
- MD.070 - Create Application Extensions Technical Design
- MD.080 - Review Functional and Technical Designs
- MD.090 - Prepare Development Environment
- MD.100 - Create Database Extensions
- MD.110 - Create Application Extension Modules
- MD.120 - Create Installation Routines



## Summary

In this lesson, you should have learned how to:

- What is AIM?
- Activities Supported – AIM
- AIM - Phases
- AIM – Data Conversion (CV) Methodology
- Data Conversion (CV) – Objective & Steps
- Documentations – Data Conversions
- Data Conversion – Processes
- Usage & Deliverables – Each Conversion Documents
- Customization Process - Documents



## Summary

In this lesson, you should have learned how to:

- Basic of Coding Standards
- Handlers
- Triggers
- SQL
- PL/SQL Code



## ERP- Oracle Apps

### Lesson 18: Customization using Custom.pll

August 16, 2014

Proprietary and Confidential

+1+



## Lesson Objectives

- List the coverage for this lesson
- Custom Library Overview.
- Architecture of the CUSTOM Library.
- Routines in the CUSTOM library.
- Adding packages to the CUSTOM library.
- Coding Restrictions.
- Events passed to the CUSTOM library.
- Examples.
- Zoom LOV.
- Summary.
- Review Question



## Custom Library Overview

- An oracle forms allow PL/SQL library.
- Allow the extension of oracle applications without modifying oracle applications code.
- Receives events from forms in oracle applications. Can be used to:
  - Code zoom
  - Enforce business rules.
  - Disable fields.
  - Modify form appearance.

August 16, 2014

Proprietary and Confidential

+ 3 +



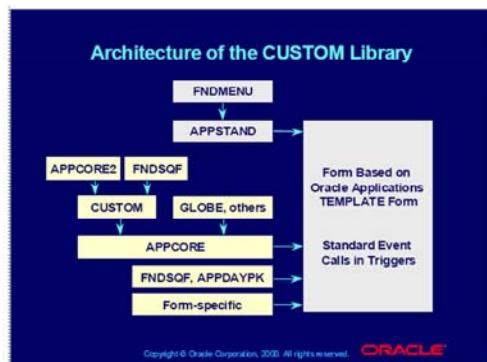
CUSTOM.dll is used to add extensions to Oracle's form Functionality. Some of the common scenarios where CUSTOM.dll can be used are:-

1. Enabling/Disabling the fields
2. Changing the List of Values in a LOV field at runtime
3. Defaulting values
4. Additional record level validations
5. Navigation to other screens
6. Enabling Special Menu

The CUSTOM.dll 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.

## Architecture of the CUSTOM Library



August 16, 2014

Proprietary and Confidential

+ 4 +

**IGATE**  
*Speed. Agility. Imagination.*

All the forms that are based on the oracle applications TEMPLATE from reference objects from the APPSTAND.fmb form.libraries attached directly to the form include APPCORE,APPYPK,FNDSQF and any other form specific libraries.

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 (CUSTOMplx) 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.

## Architecture of the CUSTOM Library

- CUSTOM is called from events common to all in all oracle application.
- Event triggers in the TEMPLATE from call APPCORE routines which in turn call the CUSTOM library.
- CUSTOM contains 3 routines:
  - ZOOM\_AVIALABLE
  - STYLE
  - EVENT

The CUSTOM library allows extension of Oracle Applications without modification of Oracle Applications code. You can use the CUSTOM library for customizations such as Zoom (such as moving to another form and querying up specific records), enforcing business rules (for example, vendor name must be in uppercase letters), and disabling fields that do not apply for your site.

You write code in the CUSTOM library, within the procedure shells that are provided. All logic must branch based on the form and block for which you want it to run. Oracle Applications sends events to the CUSTOM library. Your custom code can take effect based on these events.

## Routines in the CUSTOM Library

### ➤ ZOOM\_AVIALABLE:

Code whether zoom is available a particular block on a form

### ➤ STYLE:

Choose to have your code executable before , after,or in place of the code provided. STYLE is available only for product specific events that support custom execution styles.

### ➤ EVENT:

Code specific business logic which is based on events occurring in form.

## ZOOM\_AVIALABLE

This function allows you to specify if zooms exist for the current context. If zooms are available for this block, then return TRUE else return FALSE. This routine is called on a per-block basis within every Applications form from the WHEN-NEW-BLOCK-INSTANCE trigger. Therefore, any code that will enable Zoom must test the current form and block from which the call is being made. By default this routine must return FALSE.

## Appcore2 Library

- APPCORE cannot be attached to CUSTOM because CUSTOM is attached to APPCORE.
- APPCORE2 library duplicates most APPCORE routines in
  - APP\_ITEM\_PROPERTY2
  - APP\_DATE2
  - APP\_SPECIAL2
- Procedures and functions are the same, just the package names are different (add a 2)
- You must mutually attach APPCORE2 to CUSTOM, if you need any of this packages.

August 16, 2014

Proprietary and Confidential

+ 2 +



### STYLE

This function returns a integer value. This function allows to override the execution style of Product specific events, but it doesn't effect generic events like when-new-form-instance. Possible return values are:

1. custom.before
2. custom.after
3. custom.override
4. custom.standard

By default it returns custom.standard.

### EVENT

This procedure allows you to execute your code at specific events including:

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

By default this routine must perform 'null;'

## Adding packages to the custom library

- You cannot change the speciation of the CUSTOM package in any way.
- You can place your code within the existing package body , or add your own packages.
- You may attach other libraries to the CUSTOM library.
  - FNDSQF already attached.
  - Attach APPCORE2 to instead of APPCORE.

The CUSTOM library is shipped with the FNDSQF library already attached however, early versions of release 10SC did not include the attachment.

## Coding Restrictions

- You cannot use any SQL in the CUSTOM library or any library attached to it.
- Use record groups for SELECT statements, call stored procedures for other DML operations.
- The CUSTOM library is global to all oracle applications.
- There is single instance of the package variables regardless of the no of forms being in that session.

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.

Within the CUSTOM library, you are free to write almost any code supported by the Developer/2000 toolset, so long as you follow all Oracle Applications coding standards. However, there is an exception: you cannot call any APPCORE routines from CUSTOM. Almost all APPCORE routines start with the prefix "APP".

If you use Zoom or the CUSTOM library to invoke forms that you have developed, those forms must adhere completely to all of the Oracle Applications coding standards.

Attention: To invoke another form, use the function security routines. Do not use CALL\_FORM since the Oracle Applications libraries do not support it.

After you write code in the CUSTOM procedures, compile and generate the library using Oracle Forms. Then place this library into \$AU\_TOP/res/plsql directory (or platform equivalent). Subsequent invocations of Oracle Applications will then run this new code.

## Events passed to the CUSTOM library

- Zoom.
- WHEN\_NEW\_FORM\_INSTANCE.
- WHEN\_FORM\_NEVIGAT.
- WHEN\_NEW\_BLOCK\_INSTANCE.
- WHEN\_NEW\_RECORD\_INSTANCE.
- WHEN\_VALIDATE\_RECORD.
- WHEN\_NEW\_ITEM\_INSTANCE.
- SPECIAL n and KEY Fn.
- EXPORT.

## Events passed to the CUSTOM library (contd..)

- **AOV product -specific events.**
  - WHEN\_LOGON\_CHANGED.
  - WHEN\_PASSWORD\_CHANGED.
  - WHEN\_RESPONSIBILITY\_CHANGED.
- **Other product-specific events.**
  - Check the documentation for your product to see a list of events passed to the CUSTOM.
  - product documentation also contains list of forms that accept parameters for querying (used by zoom).

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

### Example: Force a field to uppercase

```
begin
 If(event_name='WHEN_NEW_FORM_INSTANCE')
 then
 If (form_name='APXVENDR') then
 app_item_property2.set_property('VENODR.NAME'.CASE_RESTRICTI
 ON,UPPERCASE);
 end if;
 end if;
 end if;
```

### Example: Change a button label

```
Begin
If(event_name='WHEN_NEW_FORM_INSTANCE')
then
If (form_name='APIVENDR') then
app_item)property2.set_property('vindr.details',LABEL,'More
Details');
end if;
end if;
end if;
```

### Example: Hide a Field

```
begin
 If(event_name='WHEN_NEW_FORM_INSTANCE')
 then
 If(form_name='APIVENDR') then
 app_item)property2.set_property('venodr.note',DISPL
 AYED, PROPERTY_OFF);
 end if;
 end if;
 end if;
```

### Example: Change a Field prompt

```
begin
 If(event_name='WHEN_NEW_FORM_INSTANCE')
 then
 If (form_name='APXVENDR') then
 app_item_property2.set_property('vendor.name'.PRO
 MPT_TEXT, 'supplier name');
 end if;
 end if;
 end if;
```

### Example: Prevent inserts/updates to a block

```
begin
If(event_name='WHEN_NEW_FORM_INSTANCE')
then
If (form_name='APIVENDR' and block_name='VENDOR') then
Set_block_property(block_name,insert_allowed,property_false);
Set_block_property(block_name,update_allowed,property_false);
End if;
End if;
End if;
```

## Special Menu Example

- (display the special15 entry on the tools menu)

```
begin
If(event_name='WHEN_NEW_FORM_INSTANCE')
If(form_name='DEMXXOER')then
App_special2.instantiate('SPECILA15','print order
again',",TRUE,'LINE');
End if;
End if;
End if;
```

## Special Menu Example

- (toggle display the special15 tools menu entry.)

```
If(event_name='WHEN_NEW_FORM_INSTANCE')
If(form_name='DEMXXOER' and block_name='ORDERS')then) then
App_special2.enable('SPECILA15',PROPERTY_ON);
Elseif(form_name= 'DEMXXOER' and block_name='LINES')then
App_special2.enable('SPECILA15',PROPERTY_OFF);
End if;
End if;
End if;
```



## Special Menu Example

- (code the logic for the special15 menu entry)

```
If(event_name='SPECIALS')then
If(form_name='DEMXXEOR' and block_name='ORDERS') then
/*add your print order logic here*/
Raise FORM_TRIGGER_FAILURE;
End if;
End if;
End if;
```

## Coding CUSTOM.ZOOM\_Available

- (If zoom is available for this block then TRUE; otherwise return FALSE.)

```
Function zoom_available return boolean is form_name
varchar2(30):=name_in('system.current_form');
Block_name varchar2(30):=name_in('system.cursor_block');
Begin
If(form_name='FNDSCAUS' and block_name='USER') then
Return TRUE;
else return FALSE;
End if;end zoom_available;
```

## CUSTOM.EVENT for ZOOM

```
begin
If (event_name='ZOOM')then
If (form_name='FNDSCAUS' and block_name='USER') then
Param_to_pass:=name_in('user.description');
Fnd_function.execute(FUNCTION_NAME=>'DEM
DEMXXOER',OPEN_FLAG=>'Y',SESSION_FLAG=>'Y',OTHER PARA
MS=>'DESCRIPTION="||param_to_pass||" ');
End if;else
Null;end if;end;
```

## Support

- There is a new LOV and parameter in all 11i forms:
- APPCORE\_ZOOM(LOV)
- APPCORE\_ZOOM\_VALUE (parametre)
- Use these when you have more than one zoom from a particular block.

## Zoom LOV

- Create a record group in CUSTOM library and populate it with the names of available zooms for the block.
- Attach the record group to the APPCORE\_ZOOM LOV.
- Call show\_lov to show the APPCORE\_ZOOM LOV.
- When the user picks a zoom, a value is returned into the APPCORE\_ZOOM\_VALUE parameter in the form.
- Retrieve the value and branch your zoom code accordingly.

## Summary

- Custom Library Overview.
- Architecture of the CUSTOM Library.
- Routines in the CUSTOM library.
- Adding packages to the CUSTOM library.
- Coding Restrictions.
- Events passed to the CUSTOM library.
- Examples.
- Zoom LOV.



August 16, 2014

Proprietary and Confidential

• 24 •

**IGATE**  
Speed. Agility. Imagination.

Add the notes here.

## Review Question

➤ **Question 1: CUSTOM contains how many routines?**

- Option 1: 1
- Option 2: 2
- Option 3: 3



➤ **Question 2: APPCORE can be attached to CUSTOM library**

- True/False

➤ **Question 3: Where CUSTOM.dll is located?**

- Option 1: \$AU\_TOP/resource Directory.
- Option 2: Fnd\_top.
- Option 3: Custom\_top.

August 16, 2014

Proprietary and Confidential

+ 25 +

**IGATE**  
Speed. Agility. Imagination.

Add the notes here.

## **ERP - Oracle Apps**

### **Lesson 19: Personalization of Forms**

August 14, 2014

Proprietary and Confidential

- 1 -



## Lesson Objectives

- Overview
- Terminology
- Personalizing a Form
- Defining Rules
  - Rules
  - Conditions
  - Scope
  - Actions
- Evaluation of Strings
- Relationship to Custom Library
- Limitations



## Overview

- The Form Personalization feature is introduced in 11.5.10
- This allows you to declaratively alter the behavior of Forms-based screens, including changing properties, executing builtins, displaying messages, and adding menu entries.
- 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.
- Once Rules are defined, when the target function is run then the Rules are automatically applied as events occur within that form.

## Terminology

### ➤ Event

- An Event is a trigger point within a form, such as WHEN-NEW-FORM-INSTANCE, WHEN-NEW-RECORD-INSTANCE.
- There are standard events that almost every form sends, and certain forms send additional product-specific events.

### ➤ Scope

- The Scope is evaluated based on the current runtime context to determine if a Rule should be processed or not.
- The Scope can be at the Site, Responsibility, User, or Industry level. Each Rule can have one or more Scopes associated with it.

## Terminology

### ➤ Condition

- The Condition is an optional SQL code fragment that is evaluated when the Event occurs; if it evaluates to TRUE then the Actions are processed.

### ➤ Action

- Each Action consists of one of the following:
  - setting a Property, such as making a field Required or hiding a Tab page
  - executing a Builtin, such as GO\_BLOCK, DO\_KEY or FND\_FUNCTION.EXECUTE
  - displaying a Message
  - enabling a Special menu entry

## Personalizations Form

The screenshot shows the Oracle Application Express Personalization Form interface. The title bar reads "Personalizations Form". The main area displays a list of actions for a form named "ICITMED\_F". The actions are:

- 10 Make desc mandatory
- 20 Default Item
- 30 Generate item number

The "Enabled" column for all actions has checkmarks. Below this, there are tabs for "Condition" and "Actions". The "Actions" tab is selected, showing a trigger event "WHEN NEW FORM INSTANCE" and a trigger object "Condition". A checkbox "Fire in Enter-Query Mode" is unchecked. At the bottom, there are buttons for "Add Parameter", "Add Block", "Add Item", "Validate", and "Apply Filter".

August 14, 2014 Proprietary and Confidential + 6 +

IGATE Speed. Agility. Imagination

## Personalizing a Form

- To create personalizations for a particular function, first invoke that function from the Navigation menu.
- While in the form, choose Help->Diagnostics->Custom Code-> Personalize from the pulldown menu.
- The Personalization form will open and automatically query existing Rules for that function.
- After making changes, Save them then close and re-run the function to have them take effect.
- You can also Validate or Apply certain changes immediately to test them without having to re-run the target form by pressing the 'Validate' or 'Apply Now' buttons.

## Personalizing a Form

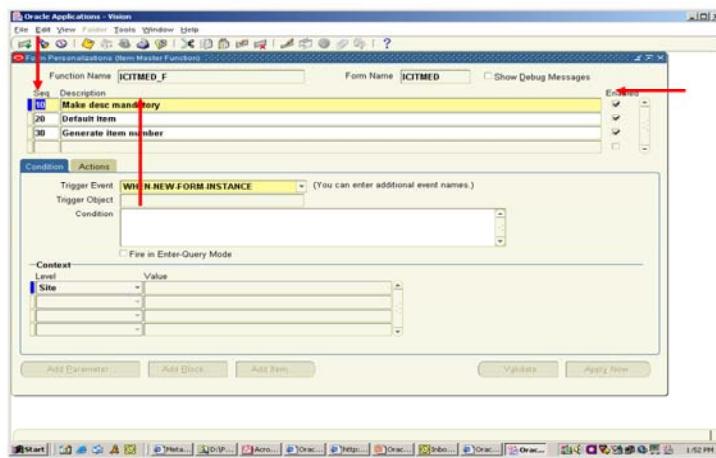
- This menu entry is secured by the 'Hide Diagnostics' menu entry and 'Utilities:Diagnostics' profiles, as are most other entries on the Diagnostics menu.

## Defining Rules

➤ **Each Rule consists of the following fields:**

- **Seq:** The sequence in which rules will be processed. This is a value between 1 and 100, with 1 being processed first. The sequence of rules does not have to be unique. Note that there is an interaction with the Trigger Event field, described below.
- **Description:** Use this field to document the personalization you are making.
- **Enabled:** Uncheck this checkbox to temporarily disable processing of a Rule.

## Defining Rules



**IGATE**  
Speed. Agility. Imagination

## Defining Rules - Conditions

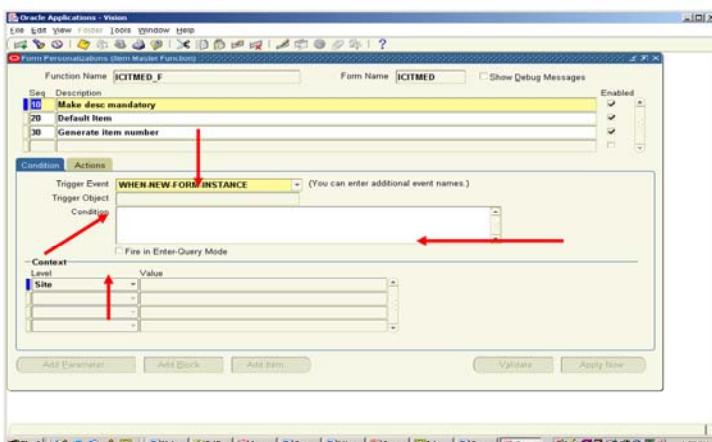
### ➤ Condition tab:

- **Trigger Event:** Select the event at which you want the Rule to be processed.  
Rules are processed first by matching the Event, then by their Sequence number.
- **Trigger Object:** Depending on the Trigger Event, this field may be Disabled, or Enabled and Required in which case it will validate against a List of Values.  
For example, if Trigger Event WHEN-NEW-ITEM-INSTANCE is selected, then you must enter a specific block.field for that trigger to be processed.
- **Condition:** This is an optional SQL code fragment that is evaluated when the Event occurs; if it evaluates to TRUE then the Actions are processed.

## Defining Rules - Conditions

- **The condition can contain any of the following:**
  - SQL functions and operators, such as AND, OR, TO\_CHAR, DECODE, and NVL
  - References to bind variables (:block.field), including :system, :global and :parameter values.
    - Use the 'Add Item...' button to assist with item names.
  - Calls to server-side functions that do not have OUT parameters  
The syntax can be tested with the 'Validate' button, which will evaluate it in the current context of the target form.
  - **Fire in Enter-Query Mode:** This checkbox controls whether the Rule should be processed if the event occurs during enter-query mode processing.

## Defining Rules - Conditions



**IGATE**  
Speed. Agility. Imagination.

## Defining Rules - Scope

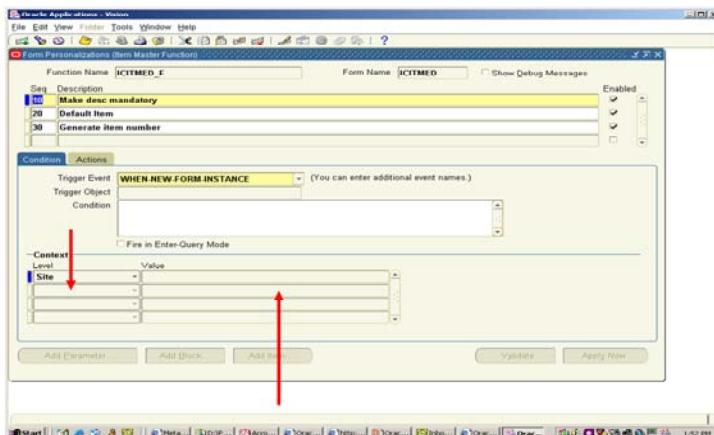
➤ **The following Scope fields appear in the Context region of the Condition tab:**

- **Level:** Select the level at which you want the rule to be applied, either Site, Responsibility, User, or Industry.
- **Value:** Based on the Level, either Disabled, or Enabled and Required in which case it will validate against a List of Values.

If a Rule has no Scope rows or Action rows, it is not processed.

Upon saving a Rule, if no Scope rows have been entered the form will automatically create a row at the Site level.

## Defining Rules - Scope



**IGATE**  
Speed. Agility. Imagination

## Defining Rules - Actions

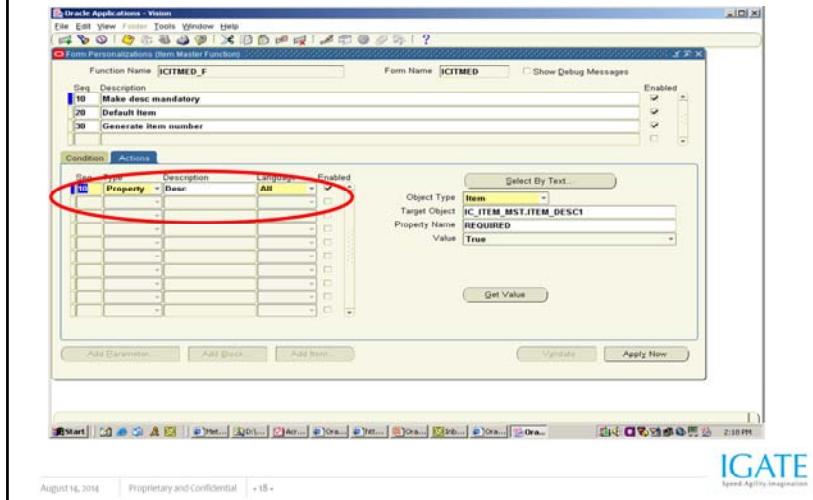
### ➤ Actions tab:

- Seq: The sequence in which actions will be processed within that Rule. This is a value between 1 and 100, with 1 being processed first. The sequence does not have to be unique.
- Type: The type of action to take
  - Property: Allows you to select a specific object, a property of that object, and specify a new value for that property
  - Builtin: Allows execution of a standard Forms Builtin, such as GO\_BLOCK or DO\_KEY
  - Message: Displays a message in one of several styles
  - Special: Enables a special menu entry, defining its label, icon name and which blocks it applies to.

## Defining Rules - Actions

- Description: Use this field to document the personalization action you are making.
- Language: Specify 'All' to have the action processed for any language, or select a specific language. Typically text-related personalizations would be applied for a specific language.
- Enabled: Uncheck this checkbox to temporarily disable processing of the action.
- Apply Now: For several Types, this button will be enabled. It allows you to apply the change immediately to the target form to test its effect.

## Defining Rules - Actions



**IGATE**  
Speed. Agility. Imagination

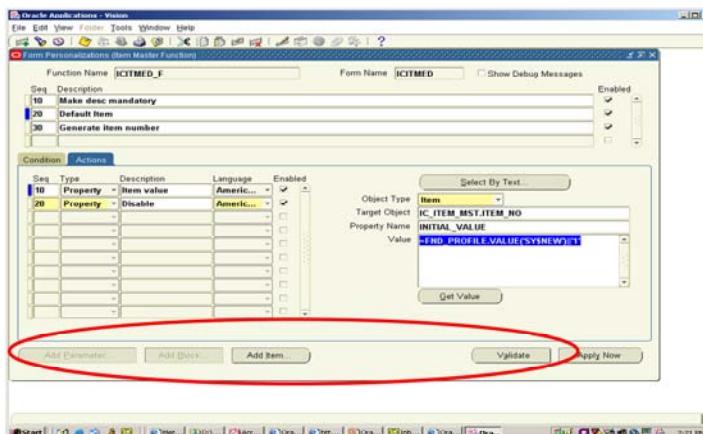
## Define Rules - Actions

➤ The following buttons are enabled conditionally based on the Type field:

- Add Parameter: List of Values that displays currently used parameters. Applies to the builtin FND\_FUNCTION.EXECUTE only.
- Add Block: List of Values that displays block names.
- Add Item: List of Values that displays item names.
- Validate: Used to test if the syntax of your string is valid.

If the evaluation fails, the processing engine will return an ORA error. Otherwise, it will display the text exactly as it would appear at runtime in the current context.

## Define Rules - Actions



August 14, 2014 Proprietary and Confidential • 20 •

**IGATE**  
Speed. Agility. Imagination

## Defining Rules - Actions

- Some fields appear conditionally based on the Type field.

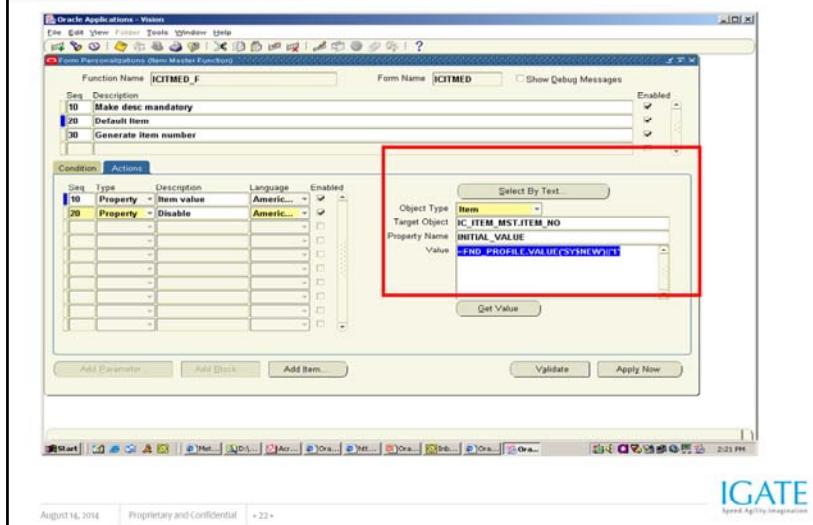
- Type: Property

- **Select By Text:** Allows you to select an object based on text appearing on the screen at the point in time that you invoke the Personalization form, including any changes that current rules might have performed.

Selecting it will automatically fill in the Object Type and Target Object fields.

- **Object Type:** The type of object, including Item, Window, Block, Tab, Canvas, Radio button, View, GLOBAL, or PARAMETER.
    - **Target object:** The internal name of the object. For Object Types of GLOBAL and PARAMETER, the Target Object name must not include those keywords.
    - **Property Name:** The properties that can be personalized.
    - **Value:** The new value.
    - **Get Value:** This button gets the current property value of the object.

## Defining Rules - Actions

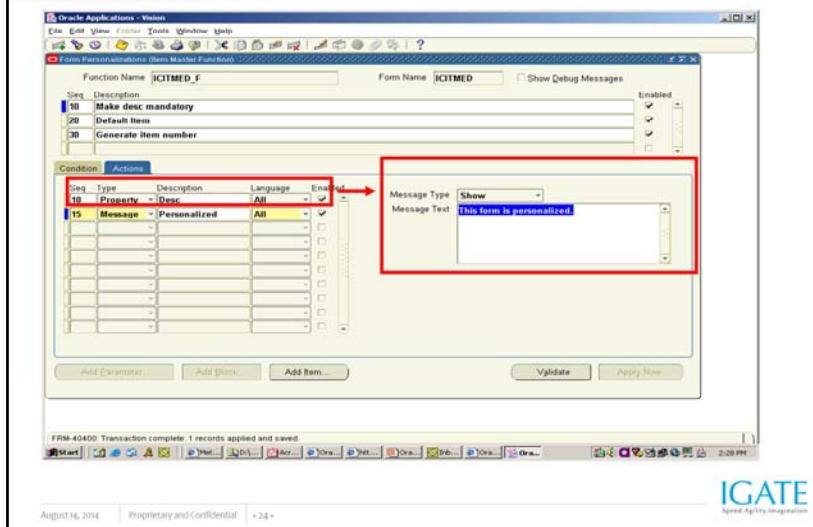


## Defining Rules - Actions

### Type: Message

- **Message Type:** Either 'Show', 'Hint', 'Warn', 'Error', or 'Debug'.  
'Error' and 'Warn' if the user selects the 'Cancel' button will raise a form\_trigger\_failure after executing, and stop all further processing.  
Messages of type 'Debug' will only be displayed if the 'Show Debug Messages' checkbox is checked.
- **Message Text:** The text you want to display in the message.

## Defining Rules - Actions



## Defining Rules - Actions

### Type: Built-in

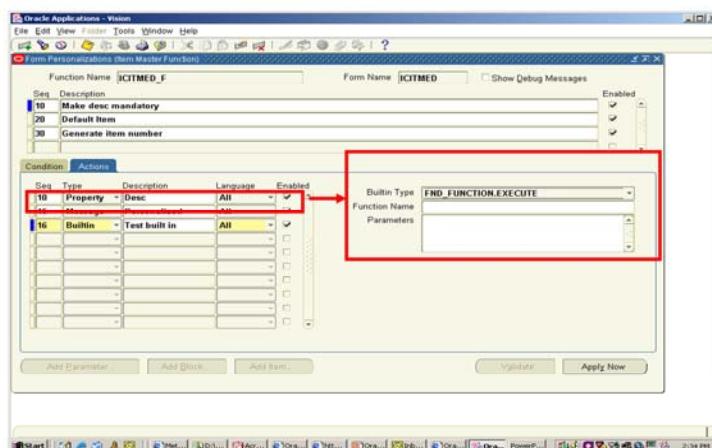
- **Builtin Type:** The name of the builtin, such as
  - GO\_ITEM
  - DO\_KEY
  - GO\_BLOCK
  - RAISE FORM\_TRIGGER\_FAILURE
  - FORMS\_DDL
  - FND\_UTILITIES.OPEN\_URL
  - FND\_FUNCTION.EXECUTE
- **Argument:** The argument for the currently selected builtin, if applicable.

## Defining Rules - Actions

➤ Depending on the specific builtin, other argument fields may appear.

- **Function Name:** The name of the function that should be executed.
- **Parameters:** You can manually enter parameters or use the 'Add Parameter...' button.

## Defining Rules - Actions



## Defining Rules - Actions

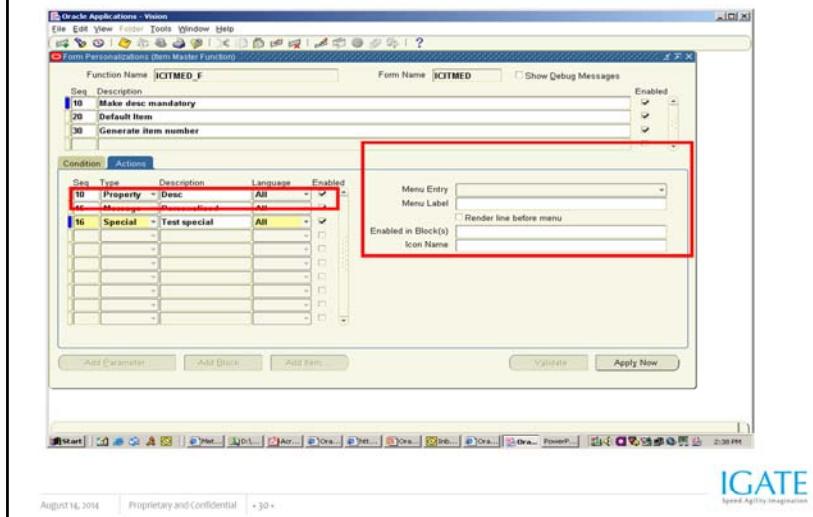
### — Type: Special

- **Menu Entry:** One of 45 menu entries that you can activate.
- **Menu Label:** The textual label that you want on the menu entry.
- **Render line before menu:** If checked, will render a line above the menu entry to visually separate it from prior entries.
- **Enabled in Block(s):** Specify the blocks that you want the menu entry enabled in; specify more than one block by separating the names with a comma.  
If no blocks are specified, the entry will be enabled in all blocks.
- **Icon Name:** Specify an optional icon name that you want added to the Toolbar to achieve the same functionality as in the special pulldown menu entry.

## Defining Rules - Actions

- Specifying an action of 'Special' merely activates the appropriate menu entry.
- When the user selects the entry, it will fire the corresponding SPECIAL# trigger.
- You must also create another rule that traps this Trigger Event and performs the desired functionality.

## Defining Rules - Actions



## Evaluation of Strings

- Every property that takes a string can either be processed literally or evaluated at runtime.
- If you type a string in that does not start with '=', then the exact value you type in will be used at runtime.
- If the string you type starts with '=', then the text immediately after that character will be evaluated at runtime.
- This allows you to write complex logic that can include references such as:
  - SQL operators, such as ||, TO\_CHAR, DECODE, and NVL
  - bind variables (:block.field), including :system, :global and :parameter values.
  - Calls to server-side functions that do not have OUT parameters.
  - SELECT statements.

## Evaluation of Strings

➤ To use the SELECT form, you must follow these rules:

- The text must start with '=SELECT'
- The column being selected must evaluate to a CHAR, with a length no longer than 2000 bytes.
- You must alias the column being selected to 'A'.
- Your SELECT statement should only return one row, but if more than one is returned only the value of the first row will be used.

## Relationship to CUSTOM Library

- Form Personalization allows personalizations that could be made in the CUSTOM library.
- It does not require that you use the Oracle Forms Builder to edit and compile the CUSTOM file.
- The CUSTOM library is able to support more complex personalizations because it gives you access to:
  - all of the capabilities of the PL/SQL programming language
  - client-side program units
  - all Oracle Forms builtins
  - issuing any SQL.

## Relationship to CUSTOM Library

- Both Form Personalization and the CUSTOM library drive off the exact same events.
- The Form Personalization feature receives and processes them first, then passes them to the CUSTOM library, thus you can use both mechanisms simultaneously.
- Both features also respond identically to the Custom Code events of 'Normal', 'Off' and 'Core Code Only'.
- In general, Oracle recommends that you use the Form Personalization feature whenever possible, and only use the CUSTOM library when significantly more complex processing is required.

## Limitations

- Form Personalization can only respond to events that are centrally processed and dispatched by APPCORE. These are limited to:
  - WHEN-NEW-FORM-INSTANCE
  - WHEN-NEW-BLOCK-INSTANCE
  - WHENNEW-RECORD-INSTANCE
  - WHEN-NEW-ITEM-INSTANCE.
  - WHEN-VALIDATE-RECORD
  - SPECIAL1 through SPECIAL45.
  - Product-specific events. These are typically documented in implementation manuals.

## Limitations

- You can only change what Oracle Forms allows at runtime. For example, the following cannot be changed:
  - You cannot create new items
  - You cannot move items between canvases
  - You cannot display an item which is not on a canvas (thus, individual flexfield segments cannot be displayed)
  - You cannot set certain properties such as the Datatype of an Item.
  - You cannot change frames, graphics, or boilerplate
  - You cannot hide the item that currently has focus

## Demo

- Disable the Submit button of order entry form using form personalization.



## Summary

➤ In this lesson, you should have learned how to do the following:

- Overview
- Terminology
- Personalizing a Form
- Defining Rules
- Evaluation of Strings
- Relationship to Custom Library
- Limitations



## Review Questions

- **Question 1: Which of the following activity cannot be done using Form Personalization :**
- Create new items
  - Move items between canvases
  - Disable Items
  - Hide the item that currently has focus



## ERP- Oracle Apps

### Lesson 20: Workflow

August 14, 2014

Proprietary and Confidential

+ 1 +



## Lesson Objectives

To understand the following topics:

- **Workflow**
  - Overview of Oracle Workflow
- **Components**
  - Workflow Engine
  - Workflow Monitor
  - Workflow Builder
  - Notification Mailer
  - Directory Services
- **Workflow Examples (Using Workflow Builder)**
  - Creation Of new Workflow
  - Approaches for tracking Errors
- **Workflow Related screens/Programs in APPS**



## What is Oracle Workflow

- Graphically represents a business process using drag and drop process designer.
- Oracle Workflow comprises of a set of tables , PL/SQL packages, Procedures ,front-end tools and back-end processes that are initiated automatically in the background as and when required according to the business rules modeled.

## Why Workflow ?

- Flexibility in modeling, automation, and continuous improvement of business processes, routing information of any type according to user-defined business rules.
- Benefits from Oracle Workflow
  - Routing Information
  - Defining and Modifying Business Rules
  - Delivering E-Notifications
  - Integrating Systems by using Business Event Manager.

## Oracle Workflow Overview

- Oracle Workflow accomplishes three important business requirements :

- Routes information
- Defines any business rule
- Delivers electronic notification

## Workflow Components

- Workflow Engine
- Workflow Monitor
- Workflow Builder
- Notification Mailer
- Directory Services

## Oracle Workflow Engine

- The Workflow Engine embedded in the Oracle database server, implements process definitions at runtime.
- The Workflow Engine monitors workflow states and coordinates the routing of activities for the process.
- Changes in workflow state, such as the completion of Workflow activities, are signaled to the engine via a PL/SQL API or a Java API.

## Oracle Workflow Engine Features

- Based on flexibly-defined workflow rules, the engine determines which activities are eligible to run, and then runs them.
- The Workflow Engine
  - supports sophisticated workflow rules, including looping, branching, parallel flows, and sub flows.
  - Can defer costly function activities to background engines for processing
  - Maintains a history of completed activities
  - Detects error conditions and runs error processes

## Initiating a Workflow Process

➤ **To start a workflow process:**

- Your application must execute a procedure that calls the following Workflow Engine API's
  - WF\_ENGINE.CreateProcess
  - WF\_ENGINE.StartProcess
- The procedure must identify the value of the process item type and item key for these API's
- These API's can be called from Forms,PLL's back-end procedures ,functions,triggers and packages

## Workflow Engine Processing

- Upon starting a process , the workflow engine:
  - Identifies and executes the Start activity
  - Determines the next activity to transition to after completing the prerequisite activity or activities
  - Drives through the process, automatically executing all function activities, until it comes to a notification activity or blocking activity
  - Calls the notification system to notify the performer
  - Continues driving through the process until it encounters an End activity.

## Example: Initiating a Workflow Process

```
procedure StartProcess (RequisitionNumber in varchar2,
 RequisitionDesc in varchar2,
 RequisitionAmount in number,
 RequestorUsername in varchar2,
 ProcessOwner in varchar2,
 Workflowprocess in varchar2 default
null,
 item_type in varchar2 default null) is

 ItemType varchar2(30) := nvl(item_type,'WFDEMO');
 ItemKey varchar2(30) := RequisitionNumber;
 ItemUserKey varchar2(80) := RequisitionDesc;
```

### Example: Initiating a Workflow Process (Contd... )

```
begin
 wf_engine.CreateProcess(...);
 wf_engine.SetItemUserKey (...);
 wf_engine.SetItemAttrText (...);
 wf_engine.StartProcess (...);

exception
 when others then
 wf_core.context (...);
 raise;
end StartProcess;
```

## Background Engine

- **The Background Engine is a PL/SQL procedure which executes the deferred activities.**
  - The Background Engine checks for and executes any deferred or timed out activities that satisfy the arguments of the procedure at the time the procedure is invoked.
  - The procedure ends once all matching activities are executed.
  - If new activities are deferred or timed out after the initiation of the current Background Engine procedure, they will be processed when the next Background Engine procedure is initiated.

## Background Engine (Contd...)

### ➤ Background Engine API

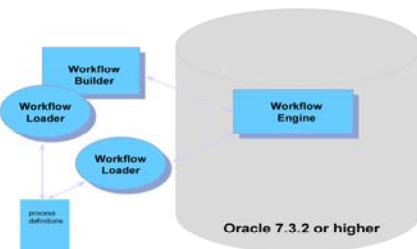
```
WF_ENGINE.BACKGROUND(
itemtype in varchar2,
minthreshold in number default null,
maxthreshold in number default null,
process_deferred in boolean default TRUE,
process_timeout in boolean default TRUE);
```

## Deferred Processing

- Set the Workflow Engine threshold cost to control which activities get deferred.
- The default threshold cost of the workflow engine is set to 50.
- The workflow engine threshold is an externalized constant.
- Add the command to a PL/SQL stored procedure or execute the command in sql\*plus to change the threshold : WF\_ENGINE.THRESHOLD := n;

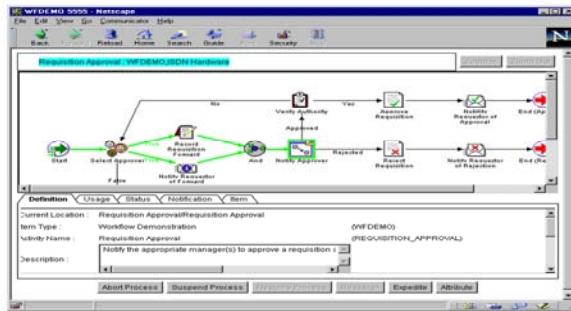
## Workflow Definitions Loader

- The Workflow Definitions Loader is a utility program that moves workflow definitions between database and corresponding flat file representations.



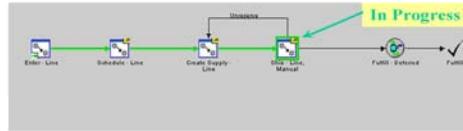
## Workflow Monitor

- The Workflow Monitor displays an annotated view of the process diagram for a particular instance of a workflow process, so that users can get a graphical depiction of their work item status.



IGATE  
Speed Agility Inspiration

## Workflow Monitor: Process Diagram



| Color  | Status      |
|--------|-------------|
| Green  | In Progress |
| Red    | Error       |
| Yellow | Hold        |
| <None> | Complete    |

## Workflow: Notifications

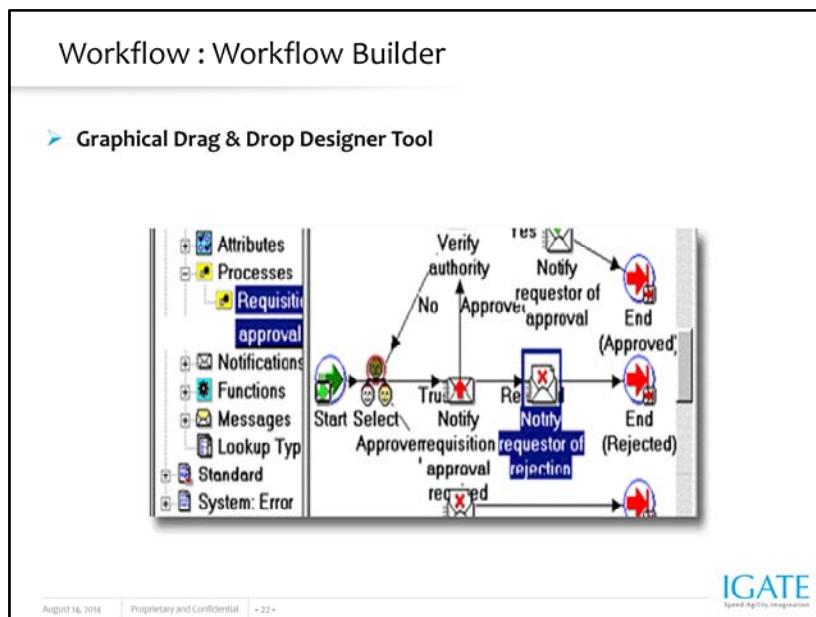
- **Automatically Notify Users**
  - Alert users of exception conditions
  - Respond to events that require human judgment
- **Send Notifications to individuals or roles**
  - Change the participants in a role without changing the process
- **Automatic Notification Forwarding**
  - When user is unavailable (vacation, sick leave)
  - When user wants to temporarily delegate a task

## Overview of the Workflow Directory Service

- Workflow uses Directory Service to resolve the performers and email addresses for sending the notifications
- Oracle Workflow ships a unified directory service for Oracle Applications-wfdirhrv.sql. The unified directory service maps:
  - WF\_USERS to Oracle HRMS employees, Oracle Application/Oracle Self-service application users, oracle receivables customer contacts and WF\_LOCAL\_USERS.
  - WF\_ROLES to users in WF\_USERS , Oracle HRMS positions, oracle applications responsibilities, oracle engineering approval lists and WF\_LOCAL\_ROLES

## Directory Service Views And Local Tables

- WF\_USERS
- WF\_ROLES
- WF\_USER\_ROLES
- WF\_LOCAL\_USERS
- WF\_LOCAL\_ROLES
- WF\_LOCAL\_USER\_ROLES



## Notifications

- A notification activity sends a message to a user or role.
- Oracle Workflow sends a notification to a role when the Workflow Engine executes a notification activity in a workflow process. The notification activity may designate the role as being responsible for performing some human action or may simply relay process-related information to the role.
- Messages lists the messages that a notification activity is associated with. the current item type can send to a user or role. A message can have message attributes associated with it.

## Summary

- Oracle Workflow manages business processes according to rules that you define
- Define your business rules using Oracle Workflow Builder and PL/SQL
- The Workflow Engine and the Notification System carry out your business rules
- Any Oracle Applications user and anyone with access to the internet can participate in the workflow.
- Use the Workflow Monitor to check and/or administer the status of your workflow processes.



## Review Questions

- **Question 1: Which of the following activity cannot be done using Form Personalization :**
- Create new items
  - Move items between canvases
  - Disable Items
  - Hide the item that currently has focus



## Review – Questions

- **Question 1: Which of the following is NOT the component of workflow?**
- Workflow Engine
  - Workflow Monitor
  - Workflow Tracker
  - Notification Mailer



# EAS Oracle Apps

## Lab Book

---

Copyright © 2011 iGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of iGATE Corporation.

iGATE Corporation considers information included in this document to be Confidential and Proprietary.

## Document Revision History

---

| Date        | Revision No. | Author    | Summary of Changes |
|-------------|--------------|-----------|--------------------|
| 11-Aug-2011 | 1.0          | Amit Sali | Content Creation   |
|             |              |           |                    |
|             |              |           |                    |

## Table of Contents

---

|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| Document Revision History .....                                                | 2  |
| Table of Contents .....                                                        | 3  |
| Getting Started.....                                                           | 4  |
| Overview .....                                                                 | 4  |
| Setup Checklist for EAS Oracle Application .....                               | 4  |
| Instructions .....                                                             | 4  |
| Lab 1. Creating report and uploading report in Oracle Application server ..... | 5  |
| 1.1: Create a report.....                                                      | 5  |
| Lab 2. Registering report in Oracle Application.....                           | 9  |
| 2.1: Registering report .....                                                  | 9  |
| Lab 3. Running report in Oracle Application.....                               | 12 |
| 3.1: Run Report .....                                                          | 12 |
| Lab 4. Registration and Running a PL/SQL Program in Oracle Application .....   | 14 |
| 4.1: Registration and Running a PL/SQL Program in Oracle Application .....     | 14 |
| Lab 5. Generating executable and uploading form in Oracle Application.....     | 16 |
| 5.1: Create a form using templet.fmb.....                                      | 16 |
| 5.2: Upload it to CUSTOM_TOP.....                                              | 18 |
| Lab 6. Registering form in Oracle Application .....                            | 22 |
| 6.1: Registering form .....                                                    | 22 |
| Appendices .....                                                               | 25 |
| Appendix A: Table of Figures .....                                             | 25 |
| Appendix B: Table of Examples.....                                             | 25 |

## Getting Started

---

### Overview

This lab book is a guided tour for learning EAS oracle application. It comprises ‘To Do’ assignments. Follow the steps provided and work out the ‘To Do’ assignments.

### Setup Checklist for EAS Oracle Application

Here is what is expected on your machine in order for the lab to work.

#### Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)
- Internet Explorer 6.0 or higher sas

#### Please ensure that the following is done:

- NA

### Instructions

- For all coding standards refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory <eas orapps>\_assgn. For each lab exercise create a directory as lab <lab number>.

## Lab 1. Creating report and uploading report in Oracle Application server

|              |                                                               |
|--------------|---------------------------------------------------------------|
| <b>Goals</b> | • To create report and upload it in Oracle Application server |
| <b>Time</b>  |                                                               |

### 1.1: Create a report

**Solution:**

**Step 1: Start Report Builder** Open reports6i Application, by selecting  
**Start → Programs → Oracle Reports 6i – OraHome62 → Report Builder.**

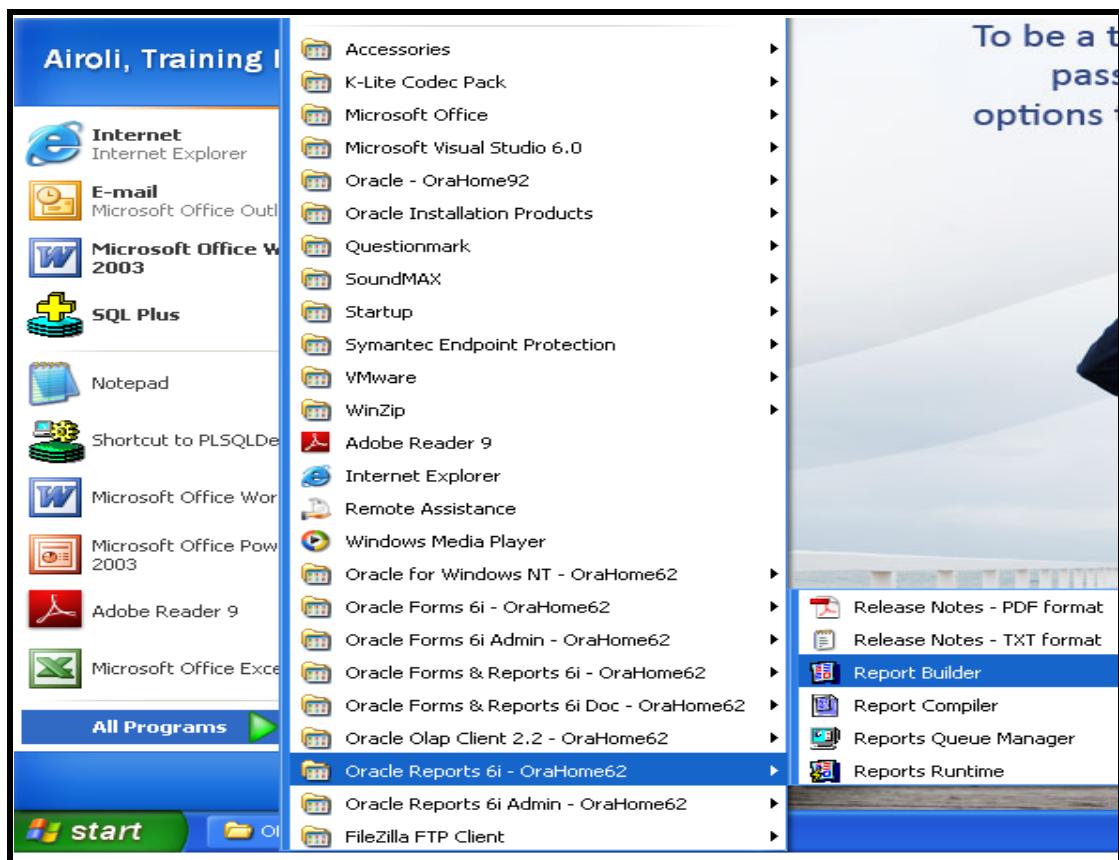


Figure 1: Start Report Builder

**Step 2:** Develop the report as per requirement. Save this form in a proper location.

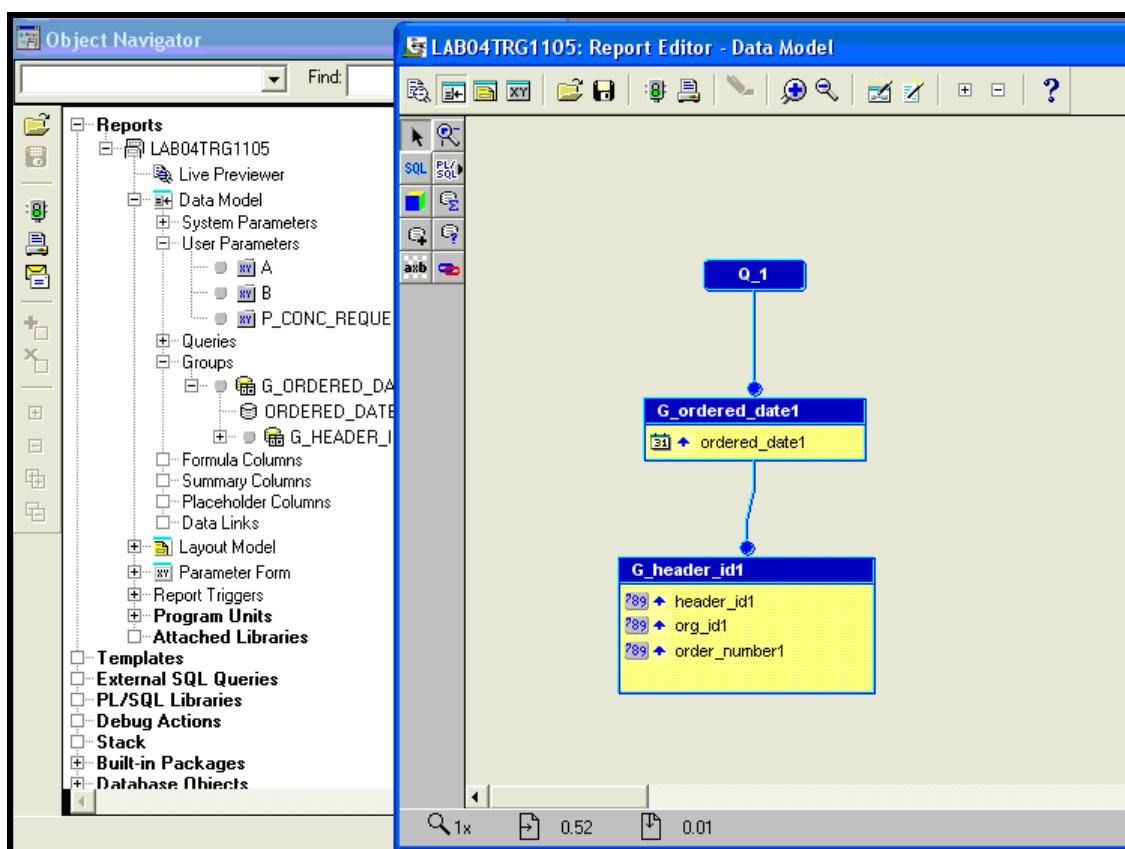
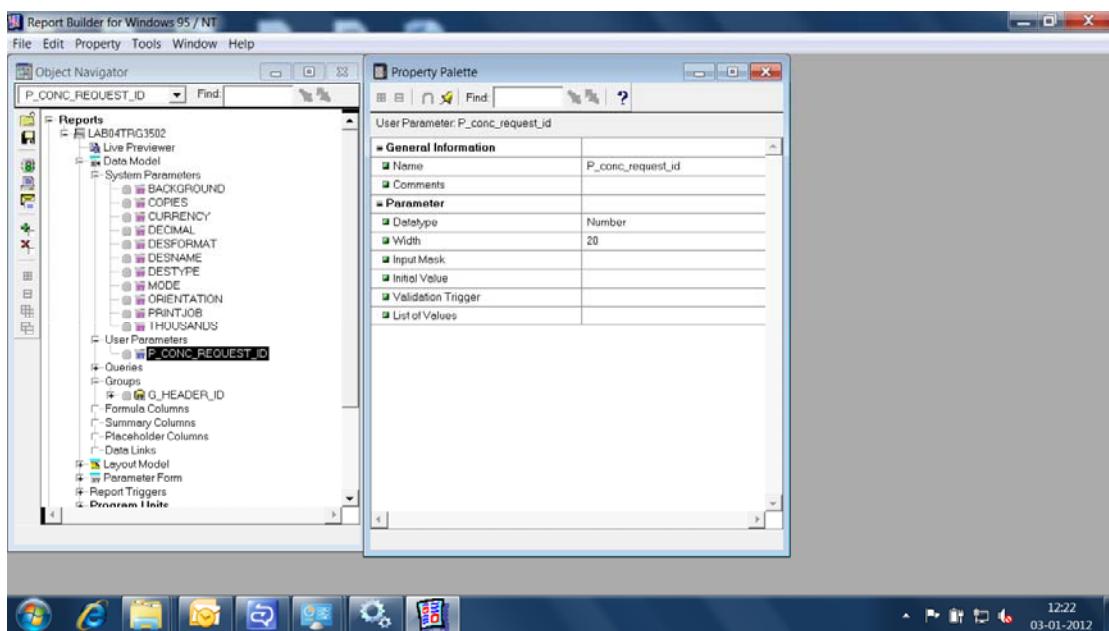
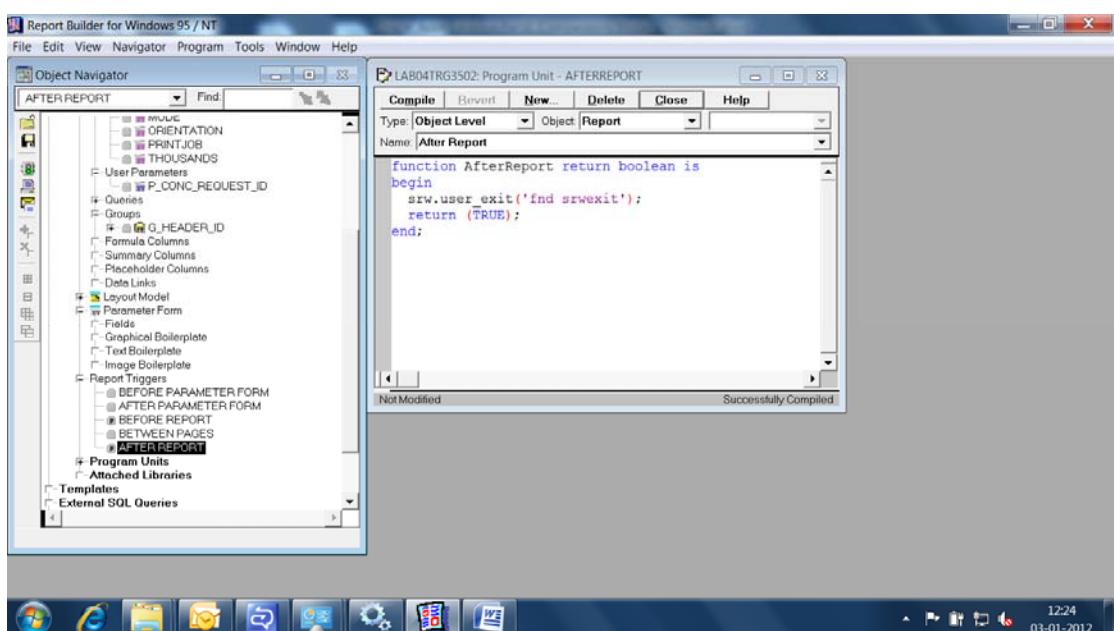
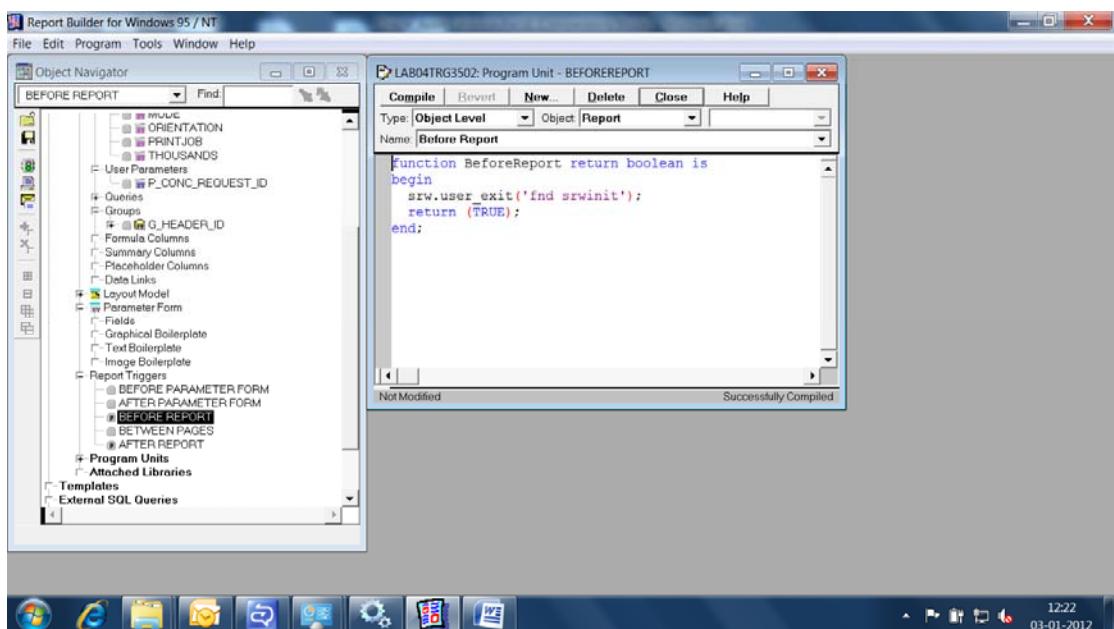


Figure 2: Design the report





**Step 3:** Upload the file to CUSTOM\_TOP

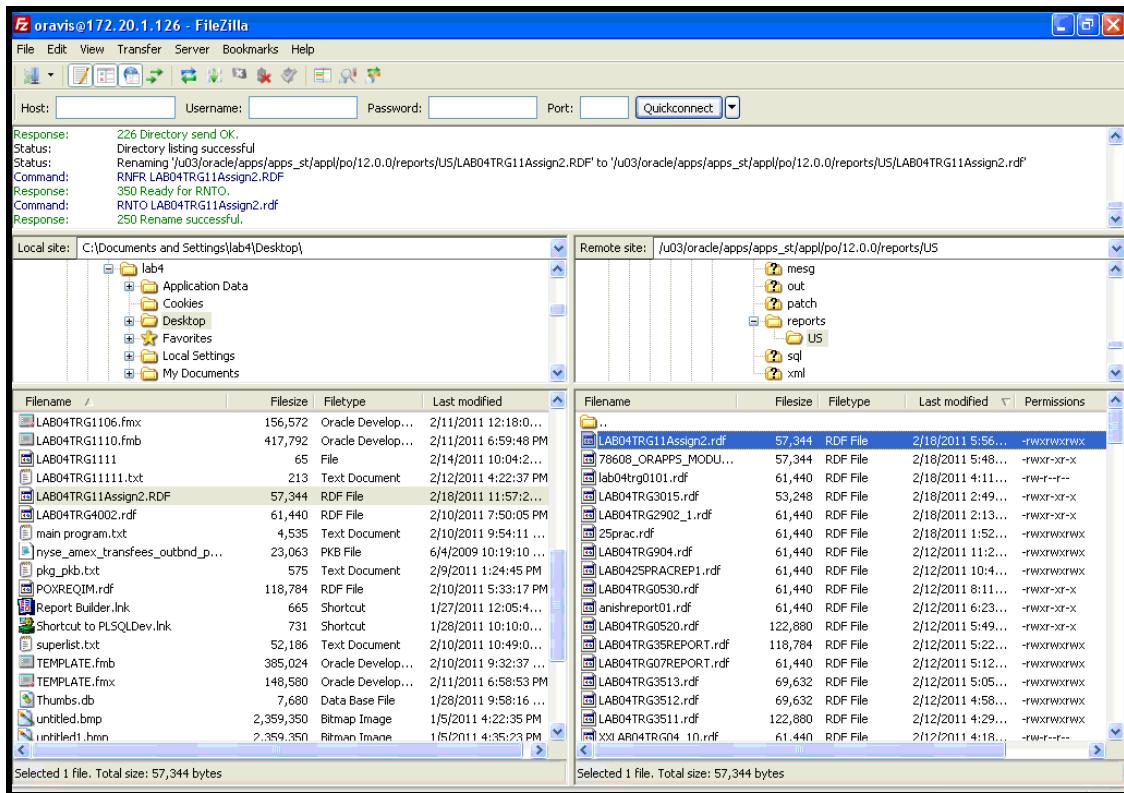


Figure 3: Upload the report on custom top

## Lab 2. Registering report in Oracle Application

|              |                                             |
|--------------|---------------------------------------------|
| <b>Goals</b> | • To register report in Oracle Application. |
| <b>Time</b>  |                                             |

### 2.1: Registering report

#### Solution:

**Step 1:** Now Login to Oracle Application.

Go to Responsibility “Application developer” → Concurrent → Program

Register an executable in oracle applications, as follows

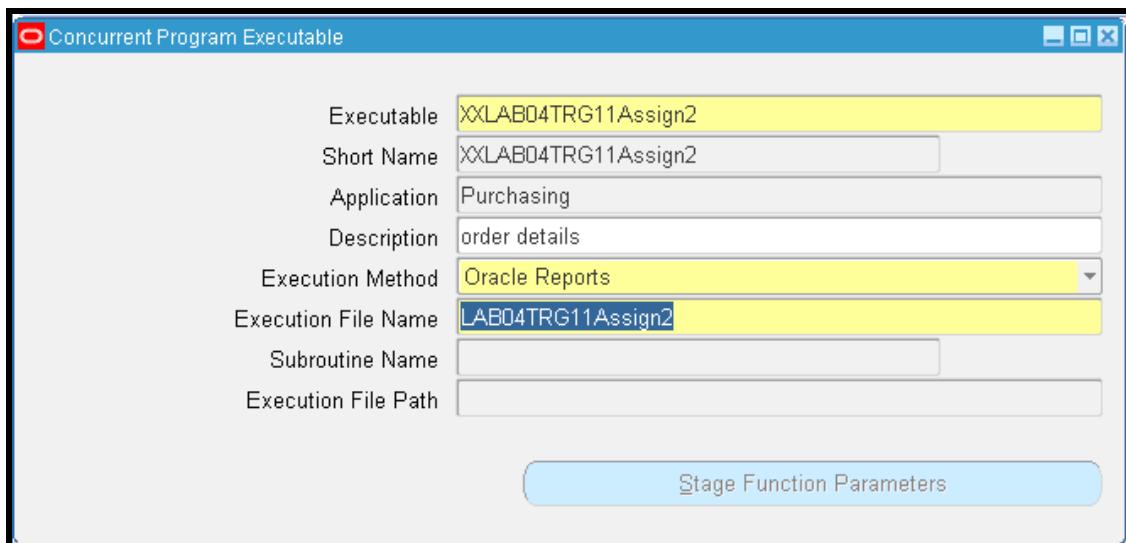


Figure 4: Executable Registration

**Step 2:** Make a program and attach executable

Go to Responsibility “Application developer” → Concurrent → Executable

Fill the details and save.

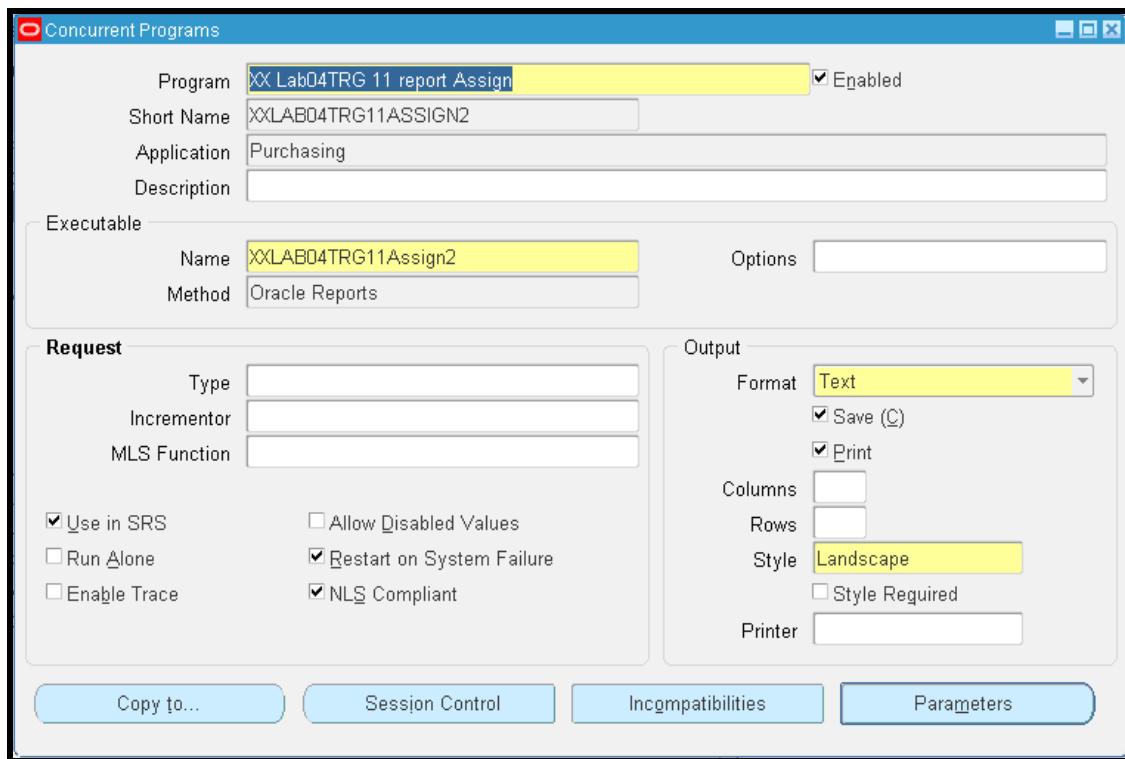


Figure 5: Program Registration

**Step 3:** Click on parameter Button

Create parameter and give the respective parameter in token field

Concurrent Program Parameters

Program: XX Lab04TRG 11 report Assign

Application: Purchasing

Conflicts Domain:

Security Group:

| Seq | Parameter         | Description | Enabled                             |
|-----|-------------------|-------------|-------------------------------------|
| 10  | P_CONC_REQUEST_ID | request Id  | <input checked="" type="checkbox"/> |
| 20  | A                 | from date   | <input checked="" type="checkbox"/> |
| 30  | B                 | to date     | <input checked="" type="checkbox"/> |

Validation

Value Set: DATE\_LIST\_STANDARD

Default Type: Constant

Required    Enable Security

Description:

Default Value: 01-JAN-2000

Range:

**Display**

Display Size: 11

Concatenated Description Size: 25

Description Size: 50

Prompt: Order Date From

Token: A

Figure 6: Parameter Registration

**Step 4:** Attach the program name to a responsibility

Go to Responsibility “System Administrator”→Security→Responsibility→ Request

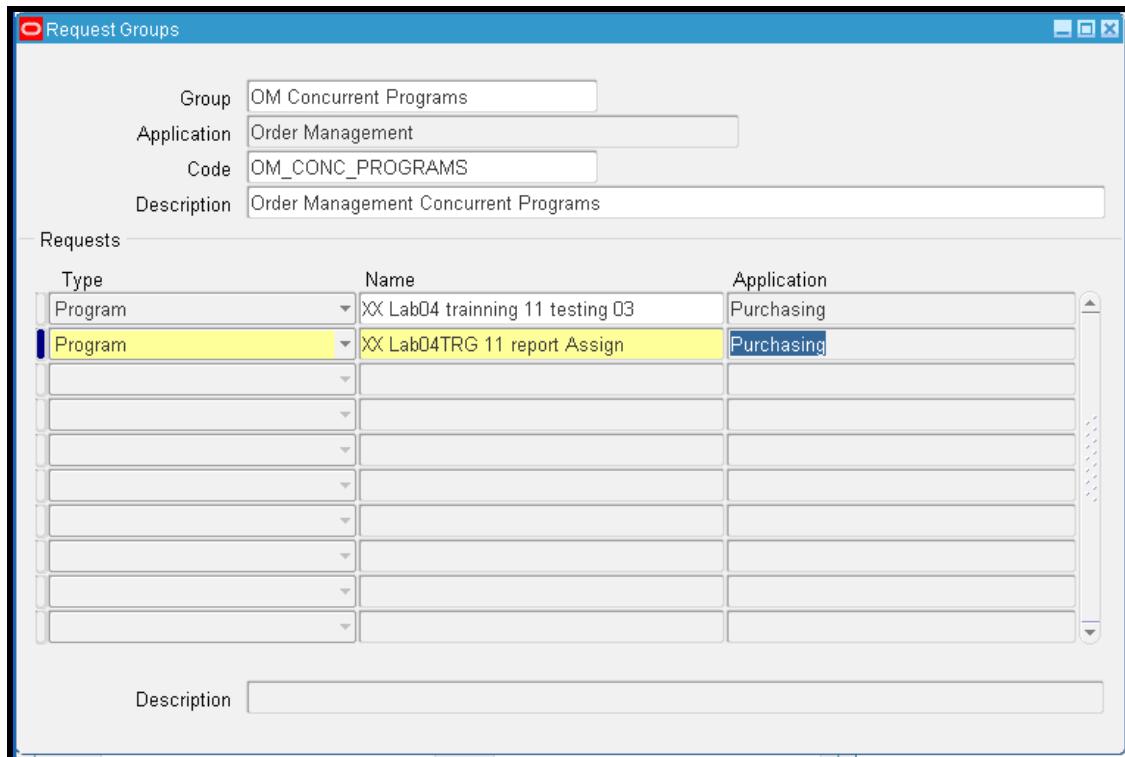


Figure 7: Attaching program to request group

## Lab 3. Running report in Oracle Application

|              |                                       |
|--------------|---------------------------------------|
| <b>Goals</b> | • To run report in Oracle Application |
| <b>Time</b>  |                                       |

### 3.1: Run Report

#### Solution:

**Step 1:** Now Login to Oracle Application.

#### Run a request

Go to Respective Responsibility →(M) view → submit a new request

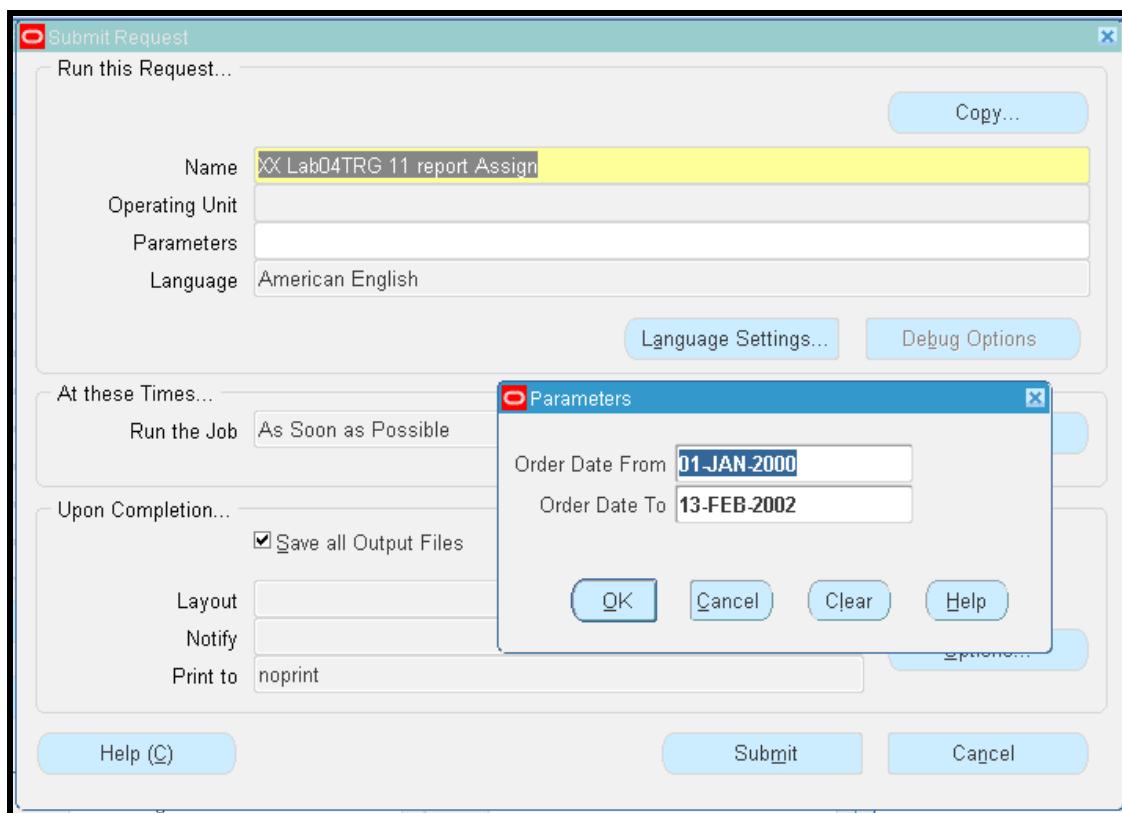
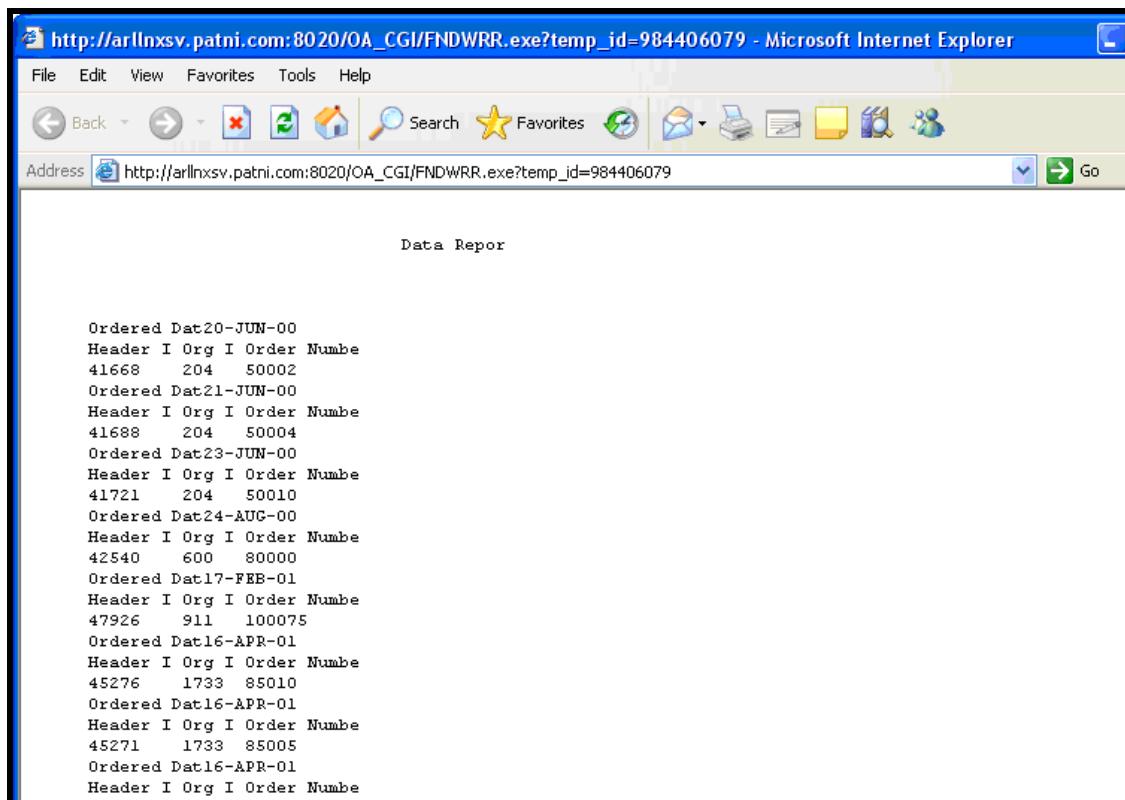


Figure 8: Running a program

**Step 2:**

Give the parameter and submit request and find the request id and see the output



The screenshot shows a Microsoft Internet Explorer window with the URL [http://arllnxsv.patni.com:8020/OA\\_CGI/FNDWRR.exe?temp\\_id=984406079](http://arllnxsv.patni.com:8020/OA_CGI/FNDWRR.exe?temp_id=984406079). The page title is "Data Repor". The content of the page displays a series of ordered data entries, each consisting of a date header followed by two numerical values. The data is as follows:

```
Ordered Dat20-JUN-00
Header I Org I Order Numbe
41668 204 50002
Ordered Dat21-JUN-00
Header I Org I Order Numbe
41688 204 50004
Ordered Dat23-JUN-00
Header I Org I Order Numbe
41721 204 50010
Ordered Dat24-AUG-00
Header I Org I Order Numbe
42540 600 80000
Ordered Dat17-FEB-01
Header I Org I Order Numbe
47926 911 100075
Ordered Dat16-APR-01
Header I Org I Order Numbe
45276 1733 85010
Ordered Dat16-APR-01
Header I Org I Order Numbe
45271 1733 85005
Ordered Dat16-APR-01
Header I Org I Order Numbe
```

**Figure 9: Output of the report**

## Lab 4. Registration and Running a PL/SQL Program in Oracle Application

|              |                                                                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Goals</b> | <ul style="list-style-type: none"> <li>To run report in Oracle Application. Registration and Running a PL/SQL Program in Oracle Application</li> </ul> |
| <b>Time</b>  |                                                                                                                                                        |

### 4.1: Registration and Running a PL/SQL Program in Oracle Application

#### Solution:

**Step 1:** Login to the Oracle Apps Database. And compile the package mention below

#### CODING

```

CREATE OR REPLACE PACKAGE labo4test_pkg
AS
 PROCEDURE labo4training(
 errbuf OUT VARCHAR2,
 retcode OUT NUMBER);

END labo4test_pkg;
/

CREATE OR REPLACE PACKAGE BODY labo4test_pkg
AS
 PROCEDURE labo4training(
 errbuf OUT VARCHAR2,
 retcode OUT NUMBER)
 AS

 BEGIN
 fnd_file.put_line (fnd_file.output,
 'in output file');
 fnd_file.put_line (fnd_file.log, 'in log file');

 END labo4training;
END labo4test_pkg;
/

```

Example 1: Sample Code

#### Step 2:

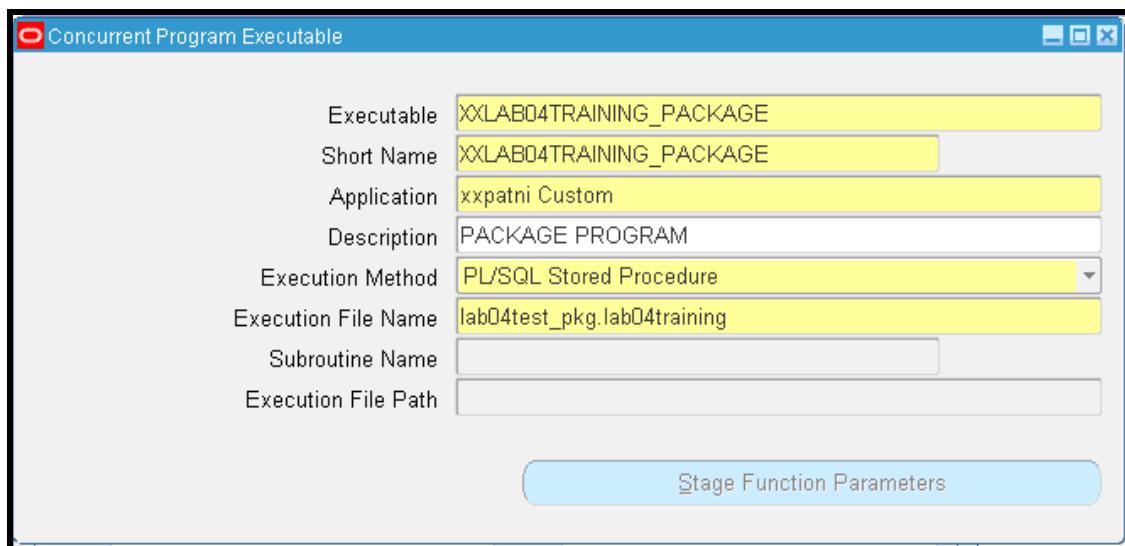
Make a program and attach executable

Now Login to Oracle Application.

Go to Responsibility “Application developer”→Concurrent→Executable

Fill the details and save.

Register an executable in oracle applications, as follows



**Figure 10: Executable Registration**

**Step 3:**

Now Follow lab 2 (Steps 2 to 4), and to run the program follow steps mention in Lab 3 and see/analyze output.

## Lab 5. Generating executable and uploading form in Oracle Application

|       |                                                                                                                |
|-------|----------------------------------------------------------------------------------------------------------------|
| Goals | <ul style="list-style-type: none"> <li>To generate executable and upload form in Oracle Application</li> </ul> |
| Time  |                                                                                                                |

### 5.1: Create a form using templet.fmb

#### Solution:

##### Step 1: Creating form:

Open form 6i Application, by selecting **Start → Programs → Oracle Forms 6i – OraHome → Form Builder.**

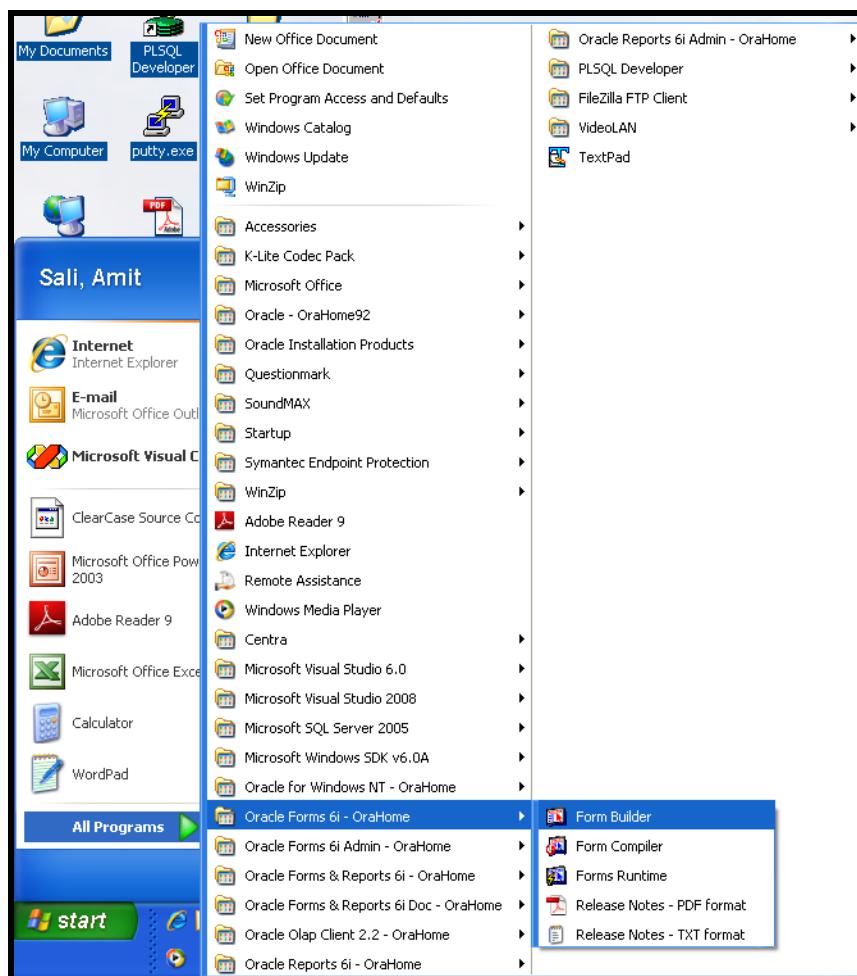


Figure 11: All Programs menu

**Step 2:** In forms builder, open the template form:

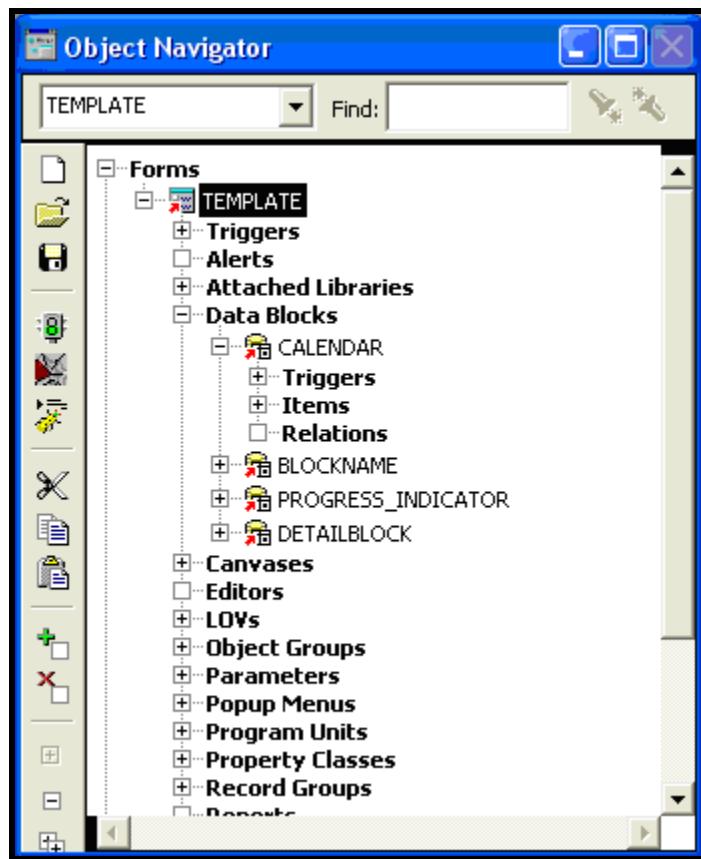


Figure 12: Creating from using template.fmb

**Step 3:** In the template form, change the layout as per your requirements. Save this form in a proper location.

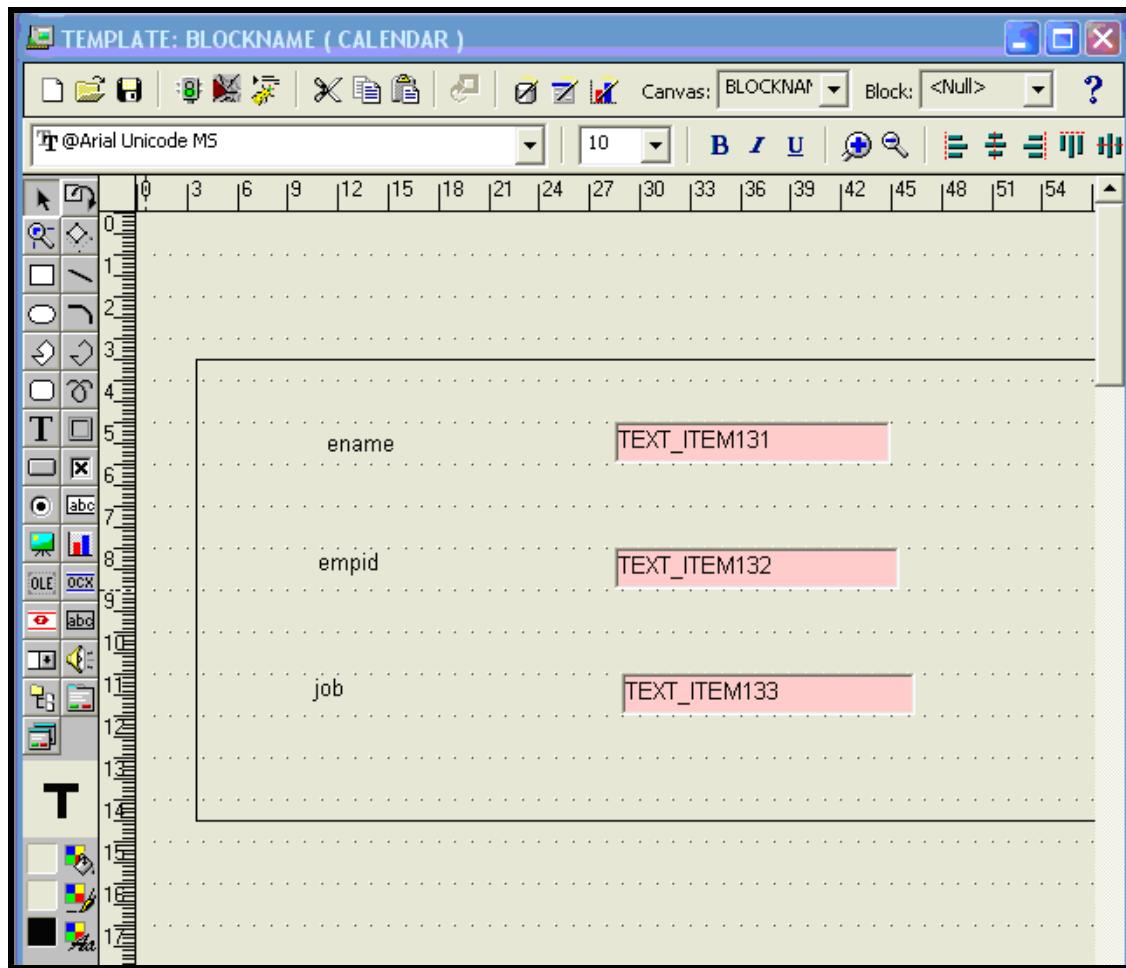


Figure 13: Sample Form

#### 5.2: Upload it to CUSTOM\_TOP

**Solution:**

**Step 1:** Open FILEZILLA and login and go to the **au top-forms**.

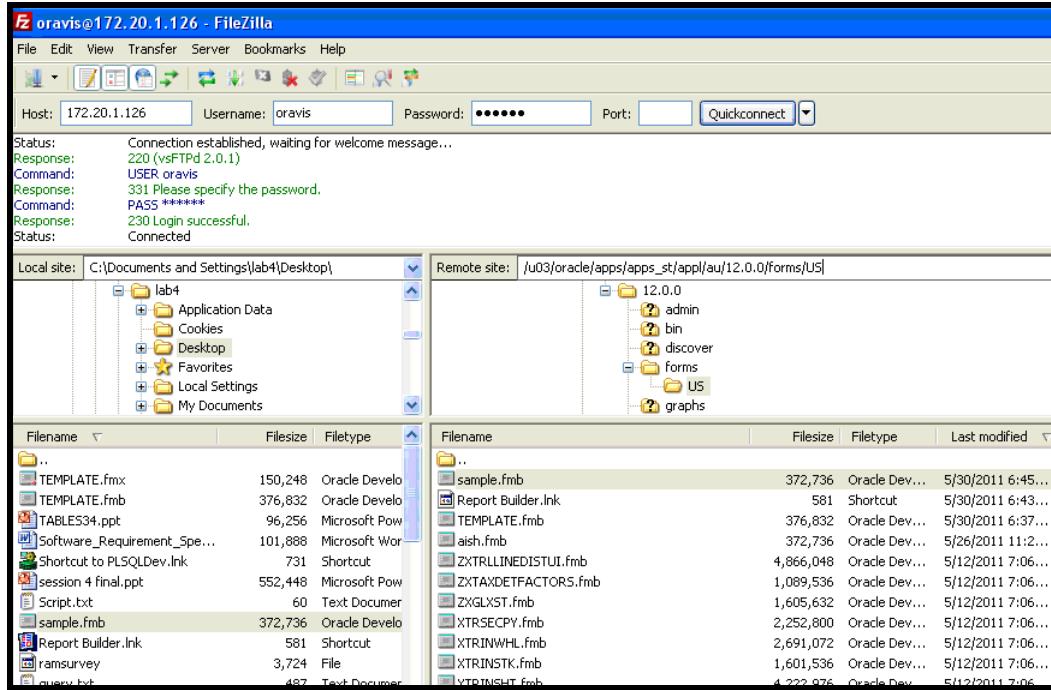


Figure 14: Form Upload step 1

**Step 2:** Upload the file to AU\_TOP and Set the file permission to 755.

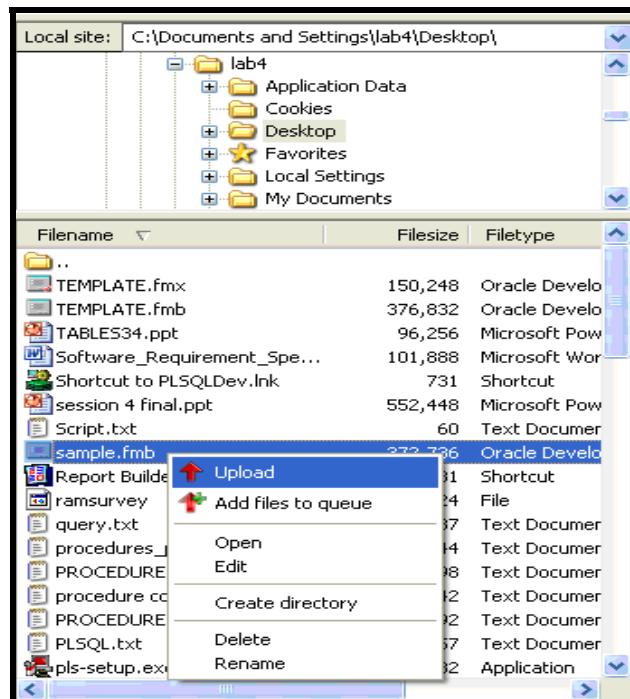


Figure 15: Form Upload step 2

**Step 3:** Change the file permissions for that form. Give it as 755.

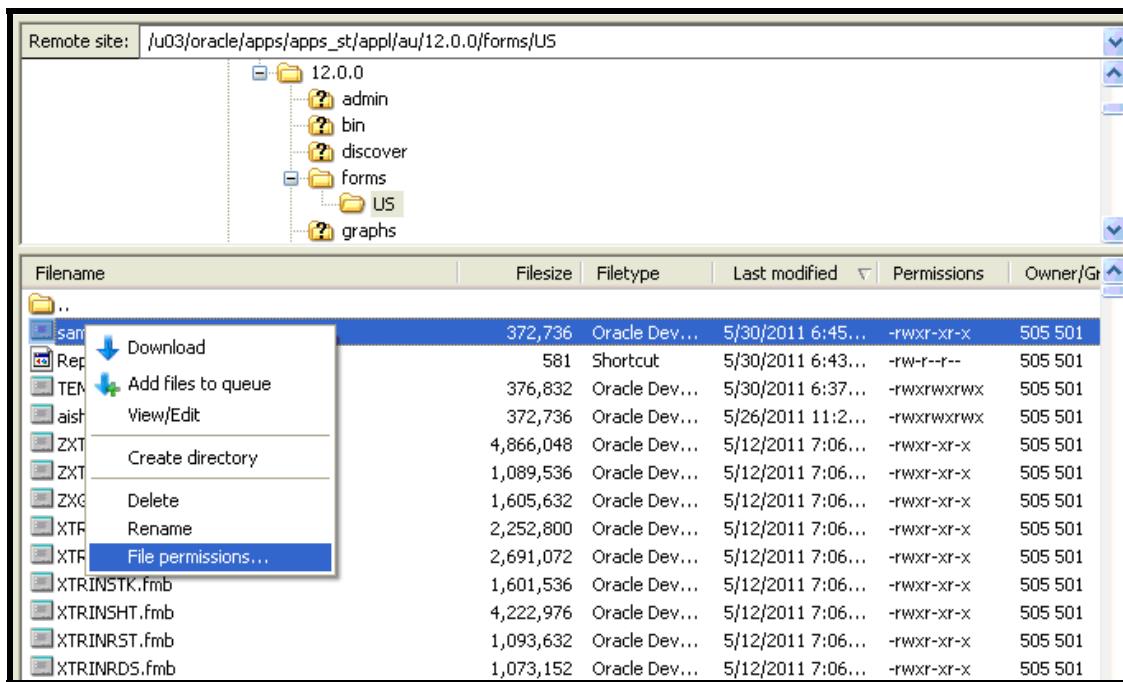


Figure 16: Form Upload step 3

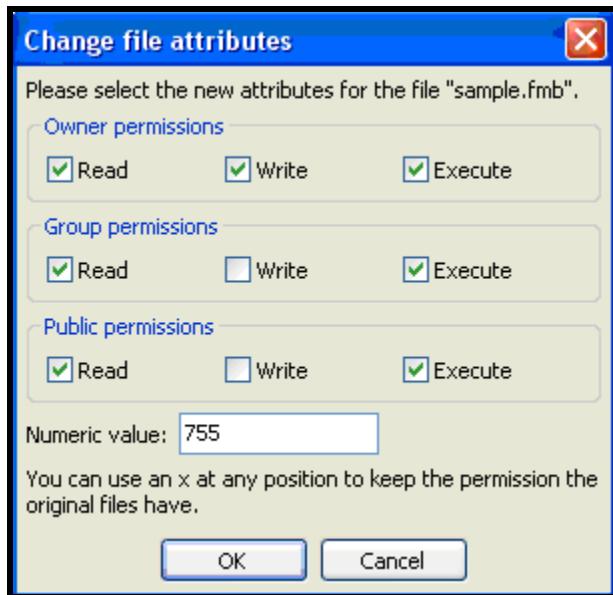
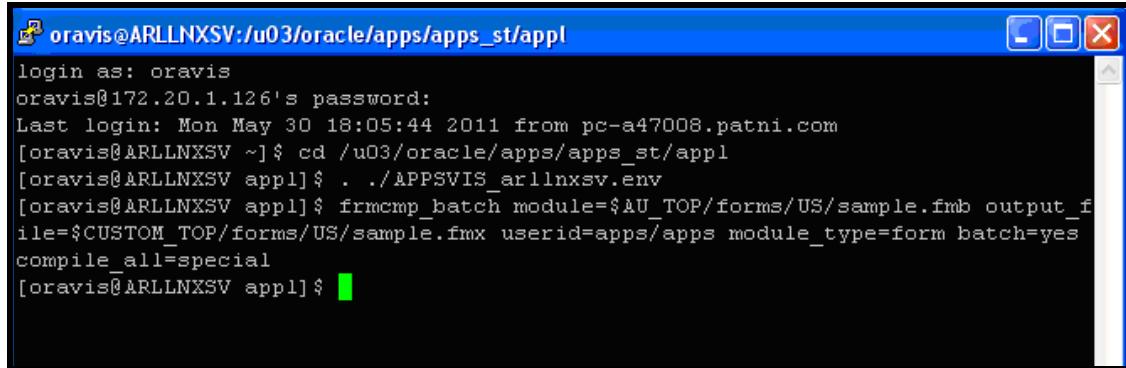


Figure 17: Supplying 755 Permission

**Step 4:** Compile and generate executable file of the form using command below

**Step 5:** Now open PUTTY and give the command to link the form with ORAPPS.



```
oravis@ARLLNXSV:/u03/oracle/apps/apps_st/appl
login as: oravis
oravis@172.20.1.126's password:
Last login: Mon May 30 18:05:44 2011 from pc-a47008.patni.com
[oravis@ARLLNXSV ~]$ cd /u03/oracle/apps/apps_st/appl
[oravis@ARLLNXSV appl]$. ./APPSVIS_arllnxsv.env
[oravis@ARLLNXSV appl]$ frmcmp_batch module=$AU_TOP/forms/US/sample.fmb output_file=$CUSTOM_TOP/forms/US/sample.fmx userid=apps/apps module_type=form batch=yes compile_all=special
[oravis@ARLLNXSV appl]$
```

Figure 18: Generating executable of fmb

**Step 6:** Download the .fmx file from xxpatni custom and upload it in ONT\_TOP.

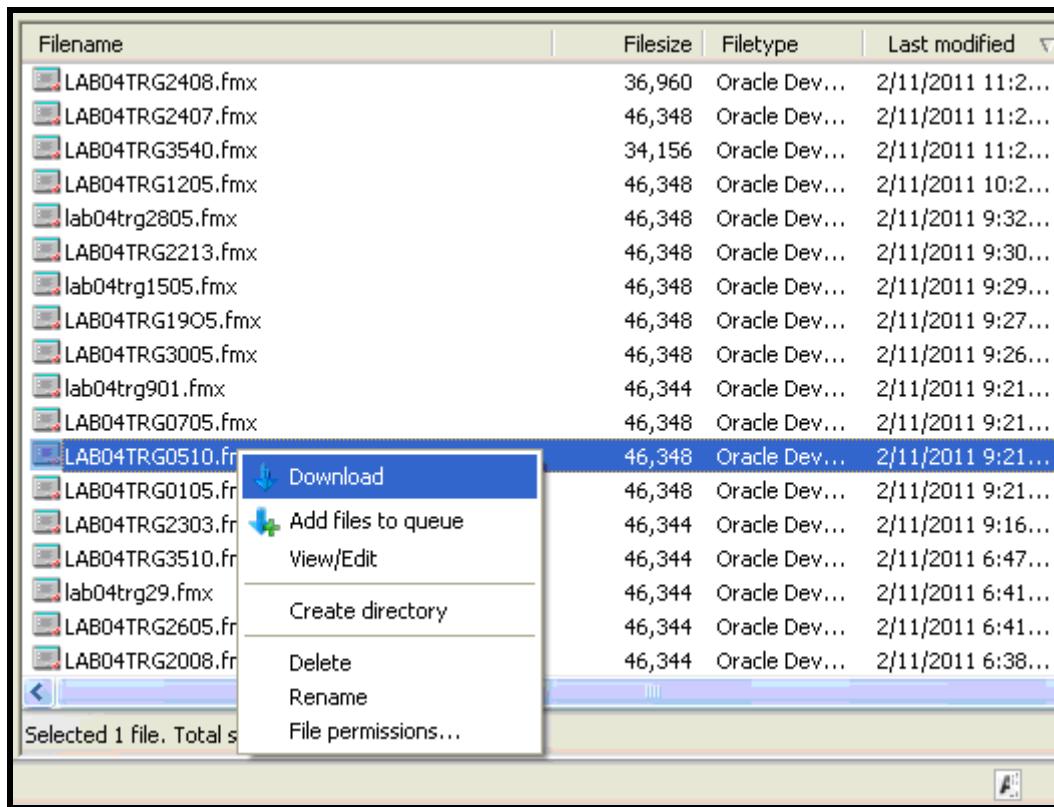


Figure 19: Downloading form

## Lab 6. Registering form in Oracle Application

|              |                                          |
|--------------|------------------------------------------|
| <b>Goals</b> | • To register form in Oracle Application |
| <b>Time</b>  |                                          |

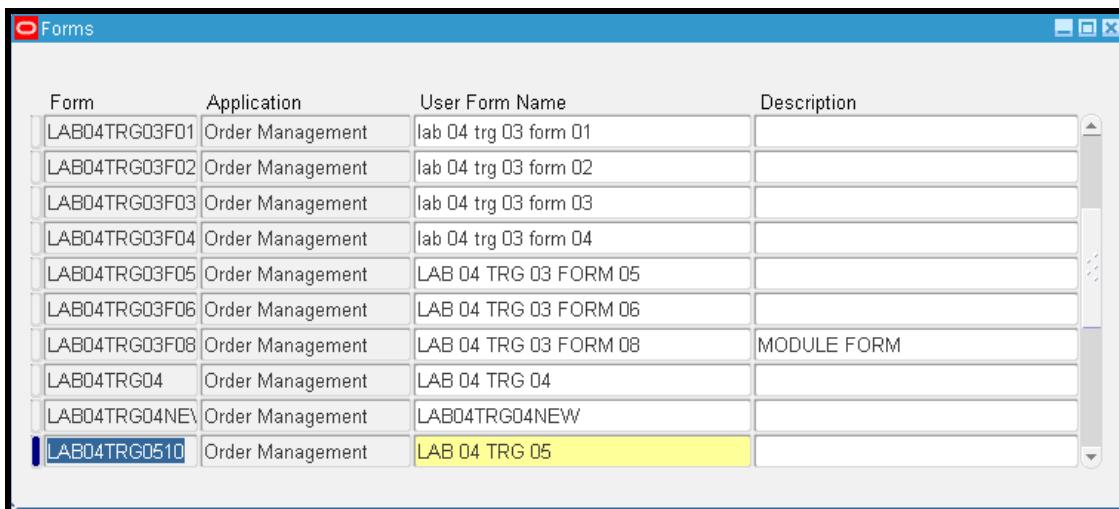
### 6.1: Registering form

#### Solution:

**Step 1:** Now Login to ORAPPS.

Go to application developer→application→forms

Fill the details in this screen and save.



The screenshot shows a window titled 'Forms' with a grid of data. The columns are labeled 'Form', 'Application', 'User Form Name', and 'Description'. The 'Description' column for the last row ('LAB04TRG0510') is highlighted in yellow.

| Form          | Application      | User Form Name        | Description |
|---------------|------------------|-----------------------|-------------|
| LAB04TRG03F01 | Order Management | lab 04 trg 03 form 01 |             |
| LAB04TRG03F02 | Order Management | lab 04 trg 03 form 02 |             |
| LAB04TRG03F03 | Order Management | lab 04 trg 03 form 03 |             |
| LAB04TRG03F04 | Order Management | lab 04 trg 03 form 04 |             |
| LAB04TRG03F05 | Order Management | LAB 04 TRG 03 FORM 05 |             |
| LAB04TRG03F06 | Order Management | LAB 04 TRG 03 FORM 06 |             |
| LAB04TRG03F08 | Order Management | LAB 04 TRG 03 FORM 08 | MODULE FORM |
| LAB04TRG04    | Order Management | LAB 04 TRG 04         |             |
| LAB04TRG04NEW | Order Management | LAB04TRG04NEW         |             |
| LAB04TRG0510  | Order Management | LAB 04 TRG 05         |             |

Figure 20: Registering a form

**Step 2:** Application→ Function →Description

Fill the details and save.

| Form Functions   |            |      |          |                  |
|------------------|------------|------|----------|------------------|
| Description      | Properties | Form | Web HTML | Web Host         |
| <b>Function</b>  |            |      |          |                  |
| XX_LAB04TRG03F02 | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG03F03 | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG03F04 | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG03F05 | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG03F06 | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG03F08 | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG04    | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG0510  | Form       | None | ▼        | Responsibility ▼ |
| XX_LAB04TRG0604  | Form       | None | ▼        | Responsibility ▼ |

Figure 21: Providing details of properties

| Form Functions   |                       |                  |          |          |
|------------------|-----------------------|------------------|----------|----------|
| Description      | Properties            | Form             | Web HTML | Web Host |
| <b>Function</b>  |                       |                  |          |          |
| XX_LAB04TRG03F02 | lab 04 trg 03 form 02 | Order Management |          |          |
| XX_LAB04TRG03F03 | lab 04 trg 03 form 03 | Order Management |          |          |
| XX_LAB04TRG03F04 | lab 04 trg 03 form 04 | Order Management |          |          |
| XX_LAB04TRG03F05 | LAB 04 TRG 03 FORM 05 | Order Management |          |          |
| XX_LAB04TRG03F06 | LAB 04 TRG 03 FORM 06 | Order Management |          |          |
| XX_LAB04TRG03F08 | LAB 04 TRG 03 FORM 08 | Order Management |          |          |
| XX_LAB04TRG04    | LAB 04 TRG 04         | Order Management |          |          |
| XX_LAB04TRG0510  | LAB 04 TRG 05         | Order Management |          |          |
| XX_LAB04TRG0604  | LAB 04 TRG 06 04      | Order Management |          |          |

Figure 22: Providing details of function

**Step 3:** Application→ Menu

Menus

| Menu           | TESTING         | View Tree... |                     |               |                                     |
|----------------|-----------------|--------------|---------------------|---------------|-------------------------------------|
| User Menu Name | Testing         |              |                     |               |                                     |
| Menu Type      |                 |              |                     |               |                                     |
| Description    |                 |              |                     |               |                                     |
| Seq            | Prompt          | Submenu      | Function            | Description   | Grant                               |
| 40             | Lab 04 trg 04   |              | LAB 04 TRG 04       |               | <input checked="" type="checkbox"/> |
| 41             | LAB 04 TRG 04 f |              | LAB04TRG04NEW       | form test     | <input checked="" type="checkbox"/> |
| 42             | lab04trg04mod   |              | lab04 trg04 mod     | form          | <input checked="" type="checkbox"/> |
| 43             | LAB04TRG04_M    |              | lab04trg04module    |               | <input checked="" type="checkbox"/> |
| 50             | LAB04TRG0510    |              | LAB 04 TRG 05       |               | <input checked="" type="checkbox"/> |
| 55             | LAB 04 TRG 29 F |              | LAB04 TRG 29 F1     | CREATED BY 29 | <input checked="" type="checkbox"/> |
| 59             | PC              |              | XX 25PRACF          |               | <input checked="" type="checkbox"/> |
| 60             | LAB 04 TRG 06 0 |              | LAB 04 TRG 06 04    |               | <input checked="" type="checkbox"/> |
| 61             | LAB 04 TRG 06 0 |              | LAB 04 TRG 06 04 AB |               | <input checked="" type="checkbox"/> |
| 70             | LAB 04 TRG 07   |              | LAB 04 TRG 07 05    |               | <input checked="" type="checkbox"/> |

Figure 23: Attaching form to menu

## Appendices

---

### Appendix A: Table of Figures

|                                                   |    |
|---------------------------------------------------|----|
| Figure 1: Start Report Builder.....               | 5  |
| Figure 2: Design the report .....                 | 6  |
| Figure 3: Upload the report on custom top .....   | 8  |
| Figure 4: Executable Registration.....            | 9  |
| Figure 5: Program Registration.....               | 10 |
| Figure 6: Parameter Registration .....            | 10 |
| Figure 7: Attaching program to request group..... | 11 |
| Figure 8: Running a program .....                 | 12 |
| Figure 9: Output of the report.....               | 13 |
| Figure 10: Executable Registration .....          | 15 |
| Figure 11: All Programs menu .....                | 16 |
| Figure 12: Creating from using template.fmb ..... | 17 |
| Figure 13: Sample Form .....                      | 18 |
| Figure 14: Form Upload step 1.....                | 19 |
| Figure 15: Form Upload step 2 .....               | 19 |
| Figure 16: Form Upload step 3 .....               | 20 |
| Figure 17: Supplying 755 Permission .....         | 20 |
| Figure 18: Generating executable of fmb .....     | 21 |
| Figure 19: Downloading form.....                  | 21 |
| Figure 20: Registering a form.....                | 22 |
| Figure 21: Providing details of properties.....   | 23 |
| Figure 22: Providing details of function.....     | 23 |
| Figure 23: Attaching form to menu .....           | 24 |

### Appendix B: Table of Examples

|                              |    |
|------------------------------|----|
| Example 1: Sample Code ..... | 14 |
|------------------------------|----|