11i System Administrator Fundamentals

Student Guide

Course Code	D14411GC10
Edition	1.0
Month Year	June 2002
Part Number	D34839

ORACLE

Author

Bill Sawyer, Clara Jaeckel, Mildred Wang

Technical Contributors and Reviewers

Bill Curtis, Teri North, Anne Carlson, Mike Weiss, Zoe Antliff-Mitchell

This book was published using: oracle*tutor*

Table of Contents

Introduction **1-1**
Introduction 1-2
Course Objectives 1-3
Overview 1-4
System Administration Component 1-5
Flexfield Component 1-6
Workflow Component 1-7
Alerts Component 1-8
Additional Material 1-9

Managing Application Security **2-1**
Managing Application Security 2-2
Objectives 2-3
Managing Application Security: Overview 2-4
Using Responsibilities 2-6
Defining a New Application User 2-8
Practice - Users 2-9
Defining an Application 2-15
Registering an Oracle ID 2-16
Data Groups 2-17
Defining a Data Group 2-18
Relating Data Groups to Forms and Programs 2-19
Practice - Data Groups 2-21
Managing Function Security 2-24
Using the Navigator 2-26
Excluding Functions and Menus 2-27
Responsibility Components 2-28
Responsibility Creation Process 2-29
Defining a New Responsibility 2-30
Practice - Responsibilities 2-32
Defining a New Menu Structure 2-40
Identifying Existing Menu Structures 2-42
Menu Guidelines 2-43
Modifying an Existing Menu Definition 2-44
Creating a Menu 2-45
The Menu Viewer 2-47
Using the Menu Viewer 2-48
Menu Tree Display Styles 2-49
Viewing Node Properties 2-50
Practice - Menus 2-51
Practice - Query-Only Forms 2-62
Summary 2-67

Managing Concurrent Programs and Reports **3-1**
Managing Concurrent Programs and Reports 3-2
Objectives 3-3
Concurrent Requests, Programs, and Managers 3-5
Business Needs for Concurrent Processing 3-6
Business Needs for Standard Request Submission 3-7
Definitions 3-8
Submit Request Flow 3-10
Submit a New Request 3-11
Using the Parameters Window 3-12

Defining a Submission Schedule 3-13
Defining Completion Options 3-15
Request ID 3-16
Reprinting a Report 3-17
Use the Requests Window to View and Change Requests 3-18
How to Use the Requests Window 3-19
Using the Requests Window 3-20
Viewer Options Window 3-22
Canceling a Request 3-23
Holding a Request 3-24
Changing Request Options 3-25
Four Phases of a Concurrent Request 3-26
Pending Phase 3-27
Running Phase 3-28
Completed Phase 3-29
Inactive Phase 3-30
System Administrator Monitoring Privileges 3-31
Controlling Log File Access 3-32
Review Log Files 3-33
Managing Log Files and Tables 3-34
Purge Request Results 3-35
Practice - Scheduling Requests 3-36
Grouping Concurrent Programs and Requests 3-43
Creating a Request Group 3-44
Practice - Request Groups 3-45
Using a Request Group with a Code 3-50
Submitting Requests: Form Compared with Responsibility 3-51
Implementing a Coded Request Group 3-52
Customization Example 3-53
Creating the New Form Function 3-54
Passable Request Parameters 3-55
Adding the Function to the Menu 3-56
Practice - Coded Request Groups 3-57
Defining a Request Set 3-64
Request Set Stages 3-65
Stage Status 3-66
Linking of Stages 3-67
Defining Request Sets Step 1: Enter Request Set Name 3-68
Defining Request Sets Step 2: Define a Stage 3-70
Defining Request Sets Step 3: Enter Requests for Stage 3-71
Defining Request Sets Step 4: Enter Request Parameter 3-72
Defining Request Sets Step 5: Link Stages 3-73
Submitting a Request Set 3-74
User Request Set Privileges 3-75
System Administrator Request Set Privileges 3-76
Practice - Request Sets Using Wizard 3-77
Summary 3-80

Administering Concurrent Managers 4-1
Administering Concurrent Managers 4-2
Objectives 4-3
Concurrent Requests, Programs, and Processes 4-4
Overview of Concurrent Processing 4-5
Defining a Concurrent Manager 4-6
Defining Work Shifts 4-8
Balancing Process Workload Over Time 4-9
Work Shift Hierarchy 4-10

Concurrent Managers Window: Program Libraries 4-11
Assigning Workshifts to a Manager 4-12
Specializing a Concurrent Manager 4-13
Specializing a Concurrent Manager: Combined Rules 4-14
Specialization Rules 4-15
Action Types 4-16
Combining Multiple Actions Within Rules 4-17
Concurrent Request Types 4-18
Using Request Types 4-19
Conflicts Domains 4-20
Processing Conflicts Domains 4-22
Control Functions of Concurrent Managers 4-23
Internal Monitor and ICM 4-25
Defining a Transaction Manager 4-26
Viewing Log and Output Files 4-27
Managing Parallel Concurrent Processing 4-28
Oracle Enterprise Manager 4-30
Oracle Applications Manager 4-31
Management Pack for Oracle Applications 4-32
Advanced Events 4-34
Oracle Performance Manager 4-35
Oracle Capacity Planner 4-36
Concurrent Processing Tuning Assistant 4-37
Practice - Administering Concurrent Managers 4-38
Summary 4-49

Managing Profile Options 5-1
Managing Profile Options 5-2
Objectives 5-3
Personal Profile Values 5-4
Profile Hierarchy Levels 5-5
System Profile Options 5-6
Using User Profile Values as Defaults 5-7
Auditing Related Profile Options 5-8
Currency-Related Options 5-9
Flexfield-Related Options 5-10
Online Reporting-Related Options 5-12
Personal Output Viewer Options 5-14
User-Related Profile Options 5-15
Security Signon Profile Options 5-17
Practice - Profile Options 5-19
Summary 5-23

Incorporating a Custom Program 6-1
Incorporating a Custom Program 6-2
Objectives 6-3
Using Custom Programs in Oracle Applications 6-4
Assumptions for Incorporating a New Program 6-5
Components of a Concurrent Program 6-6
Adding a Custom Program to Oracle Applications 6-7
Identifying the Executable 6-8
Creating the Concurrent Program 6-10
Concurrent Program Parameters Window 6-12
Associating a Program with a Request Group 6-14
Submitting the Concurrent Program 6-15
Practice - Incorporating Custom Programs 6-16
Summary 6-22

Auditing System Resources **7-1**
Auditing System Resources 7-2
Objectives 7-3
Auditing Oracle Applications 7-4
Auditing User Activity 7-5
Sign On Audit Profile Option 7-6
Interaction of Sign On Audit Values 7-7
Sign On Audit Reports 7-8
Sign On Audit Forms Report 7-9
Sign On Audit Concurrent Requests Report 7-10
Sign On Audit Responsibilities Report 7-11
Sign On Audit Unsuccessful Logins Report 7-12
Sign On Audit Users Report 7-13
Monitor Users Window 7-14
Auditing Database Changes 7-15
Steps for Setting Up AuditTrail 7-16
Identify Tables and Columns to Audit 7-17
Creating an Audit Group 7-18
Audit Shadow Tables 7-19
Shadow Table Views 7-20
Audit Tables Window 7-21
Identifying the Schema to be Audited 7-22
Enabling Audit Processing 7-23
Practice - Auditing Resources 7-24
Summary 7-27

Managing Printers **8-1**
Managing Printers 8-2
Objectives 8-3
Formatting Information 8-4
Oracle Print Definition Components 8-6
Relationships of the Printer Components 8-8
Defining Printers 8-9
Finding Existing Printer Types 8-10
Registering a New Printer 8-11
Defining a New Printer Type 8-12
Customization Materials for Oracle Applications Printing 8-13
Customizing Printer Components 8-14
Customizing Print Styles 8-15
Customizing Printer Drivers 8-16
Invoking a Printer Driver 8-18
Passing Print Arguments 8-20
Using the Program Driver Method 8-21
Initiating Printing from a System Shell Script 8-22
Printing from Standard Input Directly to the Printer 8-23
Driver Method Parameters for Spooling 8-24
Initialization and Reset Field Contents 8-26
Creating a Custom SQL*ReportWriter Driver 8-27
Sample SRW Driver File 8-29
Order of Reading Printer Settings 8-30
Testing Your Customization 8-31
Practice - Printers 8-33
Summary 8-40

Applications DBA Duties **9-1**
Applications DBA Duties 9-2
Objectives 9-3
Resource Consumer Groups Have Similar CPU Usage Requirements 9-4

Assigning Resource Consumer Groups 9-5
Hierarchy of Resource Consumer Groups 9-6
Cost-Based Optimization 9-7
CBO Statistics Gathering 9-8
Using the Network Test Window 9-10
Guided Demonstration - Consumer Groups 9-13
Summary 9-14

Incorporating Custom Help Files 10-1
Incorporating Custom Help Files 10-2
Objectives 10-3
Customizing Help 10-4
Oracle Applications Help System Utility 10-5
Help System Utility Setup: Define Directory Paths 10-6
Downloading Help Files 10-7
Identify Help Files for Customization 10-8
Identify the Language and Product of the Files 10-10
Downloading Help Files for Editing 10-11
Linking Help Files 10-13
Special Syntax versus Conventional HTML 10-14
Cross-Application Links 10-15
Related Topics Links 10-16
Related Topics Links Syntax 10-17
Uploading Your New or Changed Help Files 10-18
File Upload Steps 10-19
Creating Reports 10-21
Running Reports 10-22
Updating the Search Index 10-23
Customizing Help Navigation Trees 10-24
Help Builder Functions 10-25
The Help Builder 10-26
Opening a Tree for Editing 10-27
Adding New Help Files to a Tree 10-28
Adding New Nodes to a Tree 10-29
For Further Information 10-30
Summary 10-31

Administering Folders 11-1
Administering Folders 11-2
Objectives 11-3
Folders: Definition 11-4
Administering Folders: Topics 11-5
Assigning a Default Folder to a Responsibility 11-6
Assigning a Default Folder to a User 11-7
Assigning Ownership of a Folder 11-8
Deleting a Folder Definition 11-9
Guided Demonstration - Folders 11-10
Summary 11-17

Managing Document Sequences 12-1
Managing Document Sequences 12-2
Objectives 12-3
Using Document Sequences 12-4
Specifying Sequence Information 12-6
How to Define a Document Sequence 12-7
Specifying Document Category Information 12-9
Document Categories 12-10
Sequence Assignments 12-11

Using a Sequence in Your Application 12-12
Practice - Document Sequences 12-13
Summary 12-20

Overview of Flexfields 13-1
Overview of Flexfields 13-2
Objectives 13-3
Overview 13-4
Customizing Oracle Applications 13-5
Flexfield Structures and Flexfield Segments 13-6
Segment Prompts and Value Descriptions 13-7
Validating Input Using Value Sets 13-8
Key and Descriptive Flexfields 13-9
Key Flexfields 13-10
Intelligent Keys 13-11
Key Flexfield Example: Accounting Flexfield 13-13
Key Flexfield Example: Asset Category Flexfield 13-14
Oracle Applications Key Flexfields 13-15
Descriptive Flexfields 13-16
Using Reference Fields 13-17
Using Different Contexts: Asset Category Flexfield 13-18
Oracle Applications Descriptive Flexfields (Partial) 13-19
SRS and Flexfields 13-20
Benefits of Flexfields 13-21
Storing Flexfield Data Internally 13-22
Implementing a Flexfield Steps 13-23
Summary 13-24

Using Value Sets 14-1
Using Value Sets 14-2
Objectives 14-3
Overview 14-4
Validating Input Using Value Sets 14-5
Planning a Value Set 14-6
Types of Value Sets 14-7
Validation Type None 14-9
Independent Value Sets 14-10
Dependent Value Sets 14-11
Table-Validated Value Sets 14-12
Specialized Validation Types 14-13
Translatable Independent Value Sets 14-14
Translatable Dependent Value Sets 14-15
Validation Types: Summary 14-16
Planning Value Sets 14-17
Value Set Attributes 14-18
Planning Format Validation 14-19
Format Types and Options 14-20
Char and Number Format Editing 14-21
Time, Standard Date, and Standard DateTime Format Editing 14-22
Format Interaction 14-23
Formatting Requirements for Translatable Value Sets 14-24
Planning Validation Strategy 14-25
Planning Value Sets and Their Values 14-27
Planning Dependent Value Sets 14-28
Defining Value Sets 14-29
Predefined Value Sets 14-30
Using Existing Value Sets 14-31
Value Set Usage by Key Flexfields 14-32

Value Set Usage by Descriptive Flexfields 14-33
Defining a New Value Set 14-35
Defining a Dependent or Translatable Dependent Value Set 14-36
Selecting a Value Set for Value Definition 14-37
Defining Values for a Dependent or Translatable Dependent Value Set 14-38
Defining Values for a Value Set 14-39
Modifying Value Definitions 14-40
Value Hierarchies 14-41
Defining Value Hierarchies and Qualifiers 14-43
Summary 14-44

Defining Key Flexfields 15-1
Defining Key Flexfields 15-2
Objectives 15-3
Overview 15-4
Intelligent Keys 15-5
Key Flexfields' Dual Purpose 15-6
Oracle Applications Key Flexfields 15-7
Implementing Key Flexfields 15-8
Key Flexfield Structures 15-9
Code Combinations 15-10
Storing Code Combinations 15-11
Key Flexfield Application Tables 15-12
Key Flexfield Qualifiers 15-14
Types of Qualifiers 15-15
Flexfield Qualifiers 15-16
Oracle Applications Key Flexfields Using Qualifiers 15-17
Segment Qualifiers 15-18
Other Key Flexfield Options 15-19
Allowing Dynamic Insertion 15-20
Cross-Validating Values 15-21
Using Value Set Security 15-22
Using Shorthand Aliases 15-23
Planning Decisions 15-24
Designing the Key Flexfield Layout 15-26
Designing Segments 15-27
Specifying Default Values 15-28
Planning Key Flexfield Values 15-30
Definition Procedure 15-31
Defining Value Sets 15-32
Accessing the Key Flexfield Definition 15-33
Specifying Flexfield Behavior 15-34
Defining Segment Attributes 15-35
Defining Validation and Size Attributes 15-36
Defining Flexfield Qualifiers 15-37
Freezing and Compiling the Definition 15-38
Defining Value Set Values 15-39
Defining Segment Qualifiers 15-40
Defining Value Hierarchies 15-41
Account Generator 15-42
Practice - Define a KFF 15-43
Practice - Define a KFF with a Dependent Segment 15-56
Defining Key Flexfields Summary 15-67

Using Additional Flexfield Options 16-1
Using Additional Flexfield Options 16-2
Objectives 16-3
Overview 16-6

Cross-Validating Values 16-7
Cross-Validation Rule Syntax 16-8
Data Type Specifications and Cross-Validation Rules 16-9
Preparing for Cross-Validation 16-10
Defining Cross-Validation Rules 16-11
Cross-Validation Reports 16-12
Comparing Cross-Validation and Security Rules 16-13
Practice - Cross-Validations 16-14
Using Shorthand Aliases 16-18
Using Full and Partial Aliases 16-19
Typical Key Flexfields Using Aliases 16-20
Defining an Alias 16-21
Defining an Alias Template 16-22
Recompiling the Flexfield Definition 16-23
Flexfields: Shorthand Entry Profile Option 16-24
Flexfields: Show Full Value Profile Option 16-25
Modifying Existing Alias Definitions 16-26
Practice - Shorthand Aliases 16-27
Securing Value Set Access 16-30
Security and Value Sets 16-31
Constructing Security Rules 16-32
Specifying Acceptable Values 16-33
Using Multiple Clauses 16-34
Rules with Overlapping Values 16-35
Rules Without Overlapping Values 16-36
Implementing Flexfield Security 16-37
Enabling Security for a Value Set 16-38
Enabling Security for a Segment 16-39
Enabling Security for a Program Parameter 16-41
Defining Security Rules 16-42
Assigning Security Rules 16-43
Practice - Security Rules 16-45
Summary 16-49

Defining Descriptive Flexfields 17-1
Defining Descriptive Flexfields 17-2
Objectives 17-3
Overview 17-4
Planning a Descriptive Flexfield 17-5
Using Descriptive Flexfields 17-7
Typical Descriptive Flexfield Information 17-8
Some Oracle Applications Descriptive Flexfields 17-9
Identifying a Descriptive Flexfield 17-10
Determining the Descriptive Flexfield Name 17-11
Determining Available Resources 17-13
Planning Questions 17-14
Identifying Your Information 17-15
Grouping Information By Usage 17-16
Isolate Common Information 17-17
Determine Different Contexts 17-18
Descriptive Flexfield Components 17-19
Global Segments 17-20
Context-Sensitive Segments 17-21
Distinguishing Between Contexts 17-22
Reference and Context Fields 17-23
Using Reference Fields 17-24
Identifying Referenceable Columns 17-25

Using Context Fields ... 17-26
Planning Flexfields: Global Segments Only ... 17-27
Planning Flexfields: Context Segments Only ... 17-28
Planning Flexfields: Global and Context Fields ... 17-29
Specifying the Flexfield Attributes ... 17-30
Specifying the Context Field Information ... 17-31
Specifying the Segment Attributes ... 17-32
Storing Descriptive Flexfield Segments ... 17-33
Descriptive Flexfield Application Tables ... 17-34
Specifying the Segment Detail Attributes ... 17-35
Specifying Default Values ... 17-38
Defining a Descriptive Flexfield ... 17-40
Defining Value Sets ... 17-41
Locating the Flexfield Definition ... 17-42
Defining the Flexfield Header Attributes ... 17-43
Defining Segment Attributes ... 17-44
Specifying Validation and Field Sizes ... 17-45
Freezing and Compiling the Definition ... 17-46
Defining Context Field Information ... 17-47
Defining Context-Sensitive Segments ... 17-48
Defining Values for a Value Set ... 17-49
Practice - Define a DFF ... 17-50
Practice - Define a DFF with None Validation ... 17-60
Practice - Define a DFF with Context-Sensitive Segment ... 17-66
Practice - Define a DFF with a Dependent Segment ... 17-75
Summary ... 17-83

Using Advanced Validation Capabilities ... 18-1
Using Advanced Validation Capabilities ... 18-2
Objectives ... 18-3
Overview ... 18-4
Advanced Validation Options ... 18-5
Table-Validated Value Sets ... 18-6
Defining a Table-Validated Value Set ... 18-7
Restricting Values Retrieved from a Table ... 18-8
Limitations on Using SQL WHERE Clauses ... 18-9
Using Multiple Tables for Validation ... 18-10
Practice - Table Validation ... 18-11
Other Advanced Validation Options ... 18-14
Using $PROFILES$ ... 18-15
Using :block.field ... 18-16
Using $FLEX$ ... 18-17
Cascading Dependencies Using $FLEX$ ... 18-18
Practice - Cascading Dependencies with $FLEX$ ... 18-19
Using Special Value Sets ... 18-28
Using Pair Value Sets ... 18-29
Summary ... 18-30

Overview of Workflow ... 19-1
Introduction of Workflow ... 19-2
Objectives ... 19-3
Enabling E-Business ... 19-4
Inter-Enterprise Business Processes ... 19-5
Traditional Workflow ... 19-6
Workflow-Driven Business Processes ... 19-7
Sample Workflow Process ... 19-9
Example: Expense Report Processing ... 19-10
Example: New Hire Processing ... 19-11

Event-Based Workflow 19-12
Subscription-Based Processing 19-13
System Integration with Oracle Workflow 19-14
Event-Based Workflow Processes 19-15
Business Process-Based Integration 19-16
Supported System Integration Types 19-17
Designing Applications for Change 19-18
Designing Applications for Integration 19-21
Oracle Workflow Availability 19-23
Summary 19-24

Oracle Workflow Components 20-1
Oracle Workflow Components 20-2
Objectives 20-3
Oracle Workflow Architecture 20-4
Oracle Workflow Components 20-6
Business Event System Architecture 20-7
Business Event System Components 20-8
Oracle Advanced Queuing, an Enabling Technology 20-9
Oracle9i Communication Alternatives 20-10
Workflow Engine 20-11
Workflow Processes 20-12
Supported Process Constructs 20-14
Oracle Workflow Builder 20-16
Notification System 20-17
Notification Worklist Web Page 20-18
Notification Details Web Page 20-19
E-Mail Notifications 20-20
Directory Services 20-21
Oracle Workflow Monitor 20-22
Workflow Definitions Loader 20-24
Workflow XML Loader 20-25
Summary 20-26

Viewing and Responding to Notifications 21-1
Viewing and Responding to Notifications 21-2
Objectives 21-3
Accessing Oracle Workflow Web Pages 21-4
Viewing Notifications from a Web Browser 21-6
Simple Worklist 21-7
Advanced Worklist 21-8
Notification Details 21-10
Reassigning Notifications 21-12
Finding Notifications 21-13
Notifications Worklist 21-15
Notification Details 21-17
Notification Attachments 21-18
Reassigning Notifications 21-19
Viewing Notifications through E-mail 21-21
E-mail Response Methods 21-22
Plain Text E-mail Notifications Using Direct Response 21-23
Plain Text E-mail Notifications Using Templated Response 21-24
HTML-Formatted E-mail Notifications 21-25
Response Template 21-27
Viewing an E-mail Summary of Notifications 21-28
Automatic Notification Processing 21-29
Viewing Notification Rules 21-30
Defining Notification Rules 21-31

Viewing Notification Rules 21-34
Defining Notification Rules 21-36
Summary 21-39

Monitoring Workflow Processes 22-1
Monitoring Workflow Processes 22-2
Objectives 22-3
Workflow Monitor 22-4
Process Title 22-5
Process Diagram Window 22-6
Detail Tab Window 22-7
Administration Buttons 22-8
Application-Controlled Access to the Workflow Monitor 22-9
Direct Access to the Workflow Monitor 22-11
Finding Processes 22-12
Reviewing the Process List 22-13
Reviewing the Notifications List 22-14
Filtering Activities in the Activities List 22-16
Reviewing Parent/Child Instances 22-18
Testing Processes 22-19
Guided Demonstration - Workflow 22-21
Guided Demonstration - Workflow Builder 22-24
Summary 22-27

Purging Workflow Data 23-1
Purging Workflow Data 23-2
Objectives 23-3
Purging Workflow Data 23-4
Workflow Purge APIs 23-5
Purge Obsolete Workflow Runtime Data Concurrent Program 23-6
Guided Demonstration - Purging Workflow Data 23-7
Summary 23-8

Overview of Oracle Alert 24-1
Overview of Oracle Alert 24-2
Course Objectives 24-3
Oracle Alert Process Overview 24-4
What Is an Exception? 24-5
Examples of Event Alerts 24-6
Examples of Periodic Alerts 24-7
Action Types 24-8
Action Levels 24-9
Seven Steps to Implementing an Alert 24-10

Identifying Implementation Issues and Responsibilities for Oracle Alert 25-1
Identifying Implementation Issues and Responsibilities for Oracle Alert 25-2
Objectives 25-3
Implementation Steps 25-4
Implementation Terminology 25-7
Step 1: Reregister Oracle IDs 25-9
Step 2: Create Electronic Mail Accounts 25-12
Step 3: Register Oracle Alert Mailboxes 25-16
Step 4: Configure Concurrent Managers 25-19
Step 5: Define Database Links 25-21
Step 6: Register Custom Applications 25-25
Step 7: Define Custom Application Installations 25-27
Step 8: Register Custom Application Tables 25-29
Step 9: Define Response Options 25-31
Step 10: Create Mail Folders for Response Accounts 25-34

Step 11: Define Message Options 25-35
Step 12: Start the Periodic Alert Scheduler 25-38
Step 13: Start the Response Processor 25-41
Planning Responsibilities for Oracle Alert 25-44
Oracle Alert Manager Responsibility 25-45
Defining an Application Alert Manager Responsibility 25-46
Application Alert Manager Responsibility 25-47
Summary 25-48

Preface

Profile

Before You Begin This Course

Before you begin this course, you should have the following qualifications:

- Thorough knowledge of *11i Navigation*

Prerequisites

- There are no prerequisites for this course.

How This Course Is Organized

This is an instructor-led course featuring lecture and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills introduced.

Related Publications

Oracle Publications

Title	Part Number

Additional Publications

- System release bulletins
- Installation and user's guides
- *read.me* files
- *Oracle Magazine*

Typographic Conventions in Text

Convention	Element	Example
Bold italic	Glossary term (if there is a glossary)	The ***algorithm*** inserts the new key.
Caps and lowercase	Buttons, check boxes, triggers, windows	Click the Executable button. Select the Can't Delete Card check box. Assign a When-Validate-Item trigger to the ORD block. Open the Master Schedule window.
Courier new, case sensitive (default is lowercase)	Code output, directory names, filenames, passwords, pathnames, URLs, user input, usernames	Code output: `debug.set ('I", 300);` Directory: `bin` (DOS), `$FMHOME` (UNIX) Filename: Locate the `init.ora` file. Password: User `tiger` as your password. Pathname: Open `c:\my_docs\projects` URL: Go to `http://www.oracle.com` User input: Enter `300` Username: Log on as `scott`
Initial cap	Graphics labels (unless the term is a proper noun)	Customer address (*but* Oracle Payables)
Italic	Emphasized words and phrases, titles of books and courses, variables	Do *not* save changes to the database. For further information, see *Oracle7 Server SQL Language Reference Manual.* Enter `user_id@us.oracle.com`, where *user_id* is the name of the user.
Quotation marks	Interface elements with long names that have only initial caps; lesson and chapter titles in cross-references	Select "Include a reusable module component" and click Finish. This subject is covered in Unit II, Lesson 3, "Working with Objects."
Uppercase	SQL column names, commands, functions, schemas, table names	Use the SELECT command to view information stored in the LAST_NAME column of the EMP table.
Arrow	Menu paths	Select File > Save.
Brackets	Key names	Press [Enter].
Commas	Key sequences	Press and release keys one at a time: [Alternate], [F], [D]
Plus signs	Key combinations	Press and hold these keys simultaneously: [Ctrl]+[Alt]+[Del]

Typographic Conventions in Code

Convention	Element	Example
Caps and lowercase	Oracle Forms triggers	`When-Validate-Item`
Lowercase	Column names, table names	`SELECT last_name` `FROM s_emp;`
	Passwords	`DROP USER scott` `IDENTIFIED BY tiger;`
	PL/SQL objects	`OG_ACTIVATE_LAYER` `(OG_GET_LAYER ('prod_pie_layer'))`
Lowercase italic	Syntax variables	`CREATE ROLE` *`role`*
Uppercase	SQL commands and functions	`SELECT userid` `FROM emp;`

Typographic Conventions in Navigation Paths

This course uses simplified navigation paths, such as the following example, to direct you through Oracle Applications.

(N) Invoice > Entry > Invoice Batches Summary (M) Query > Find (B) Approve

This simplified path translates to the following:

1. (N) From the Navigator window, select **Invoice** then **Entry** then **Invoice Batches Summary**.
2. (M) From the menu, select **Query** then **Find**.
3. (B) Click the **Approve** button.

Notations:

(N) = Navigator

(M) = Menu

(T) = Tab

(B) = Button

(I) = Icon

(H) = Hyperlink

Typographical Conventions in Help System Paths

This course uses a “navigation path” convention to represent actions you perform to find pertinent information in the Oracle Applications Help System.

The following help navigation path, for example—

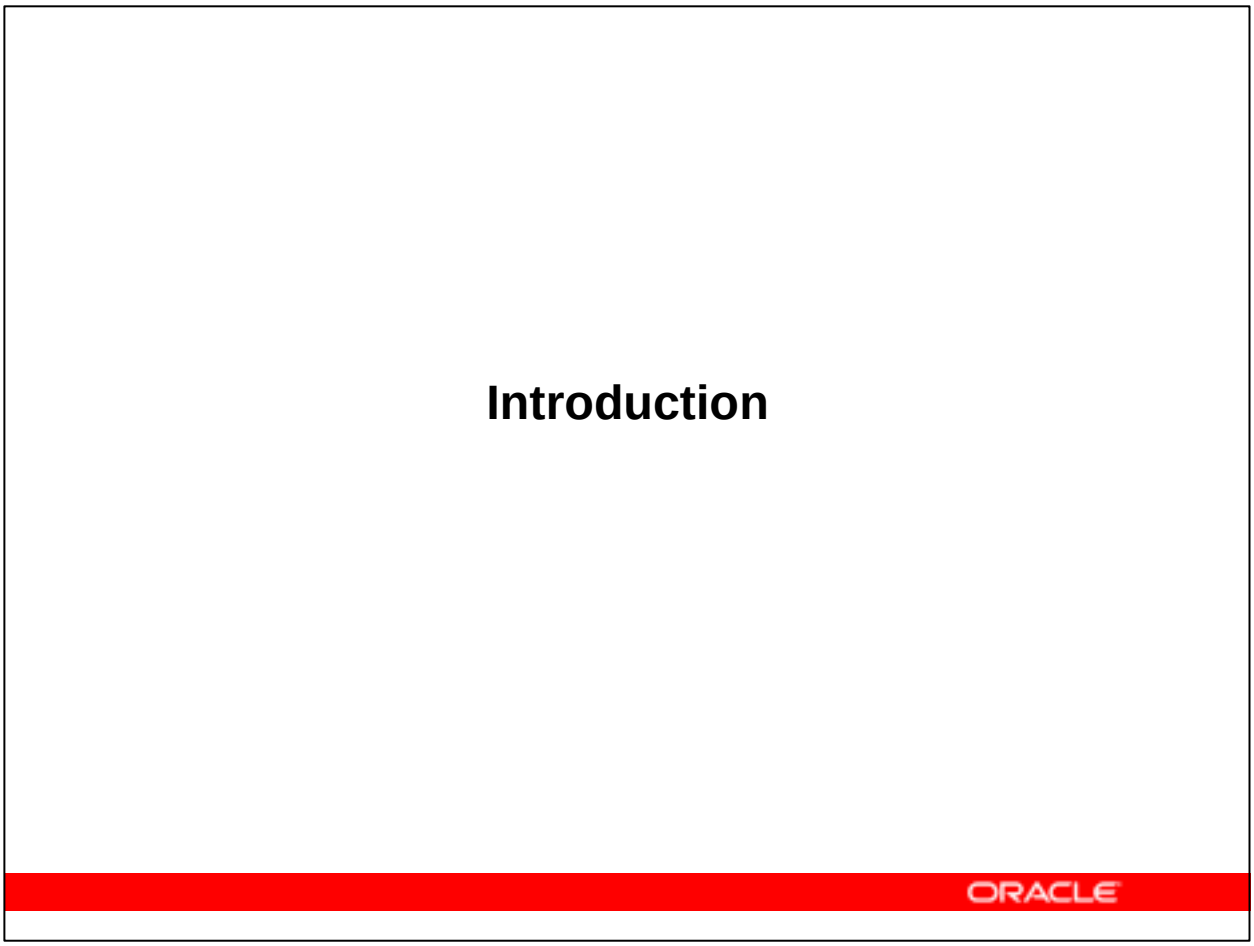
(Help) General Ledger > Journals > Enter Journals

—represents the following sequence of actions:

1. In the navigation frame of the help system window, expand the General Ledger entry.
2. Under the General Ledger entry, expand Journals.
3. Under Journals, select Enter Journals.
4. Review the Enter Journals topic that appears in the document frame of the help system window.

Introduction

Chapter 1



Course Objectives

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

- **Manage security, concurrent processing, and other 11i administrative functions**
- **Set up and manage flexfields**
- **Understand and administer workflow**
- **Understand alerts**

ORACLE

Overview

This course teaches 11i System Administrators the concepts and fundamentals of their role. Related topics, such as installation, patching, and architecture, are covered in other courses.

ORACLE

System Administration Component

In this component of the course, we will cover the specifics documented in the Oracle Applications System Administrator's Guide. This includes managing security, administrating concurrent processing, and other related topics.

ORACLE

Flexfield Component

In this component of the course, we will cover the specifics documented in the Oracle Applications Flexfields Guide. This includes understanding flexfields, setting up key flexfields, setting up descriptive flexfields, setting up value sets and their values, and other related topics.

Workflow Component

In this component of the course, we will cover a subset of the material documented in the Oracle Workflow Guide. We will cover the material that applies to the role of an 11i System Administrator.

ORACLE

Alerts Component

In this component of the course, we will cover a subset of the material documented in the Oracle Alert User's Guide. We will cover the material that applies to the role of an 11i System Administrator.

ORACLE

Additional Material

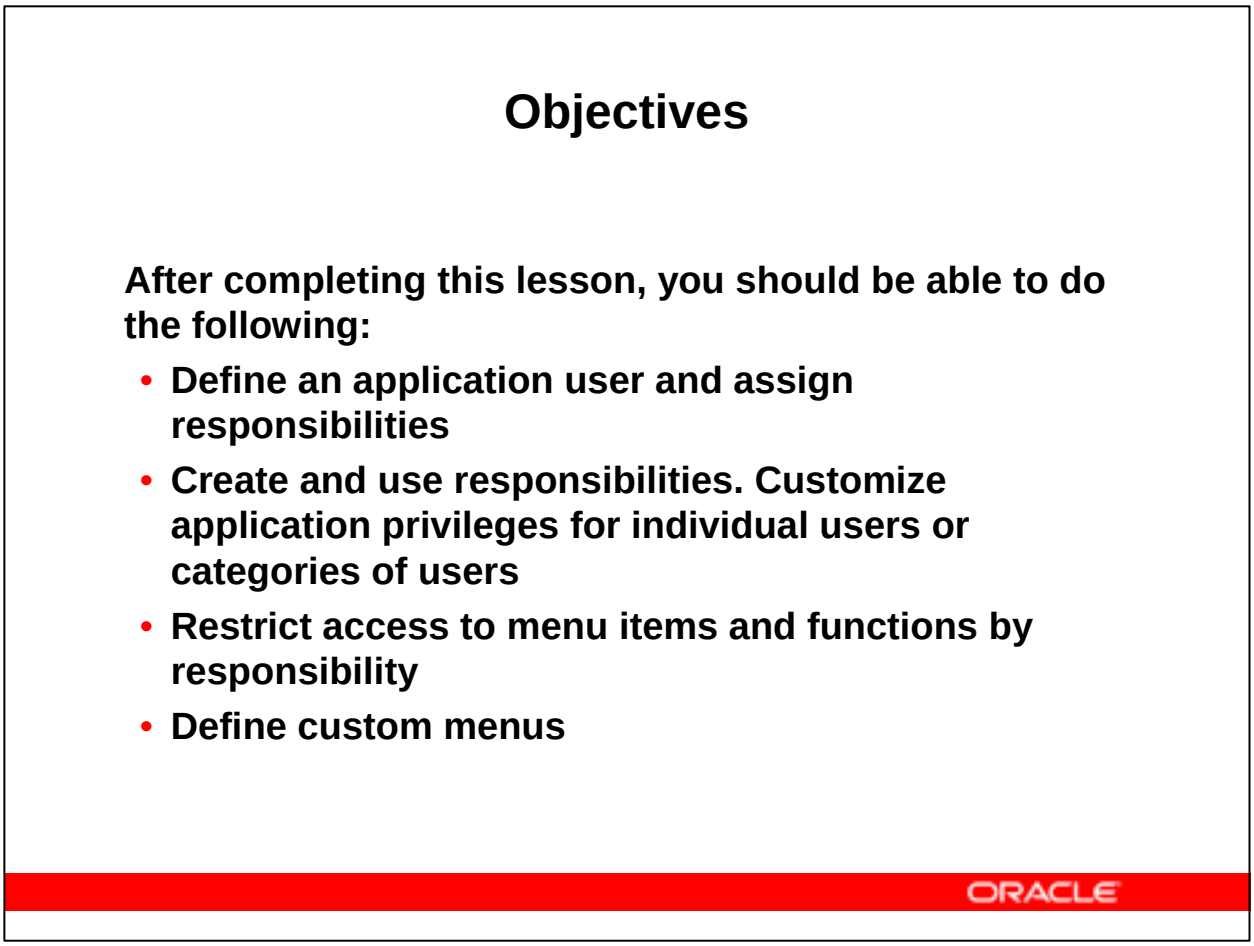
While course provides the concepts and fundamentals for an 11i System Administrator, you may find that you want or need added depth. Oracle University (OU) offers additional courses, both online and in the classroom, to address these needs. Visit the OU website for more information.

http://www.oracle.com/education/

ORACLE

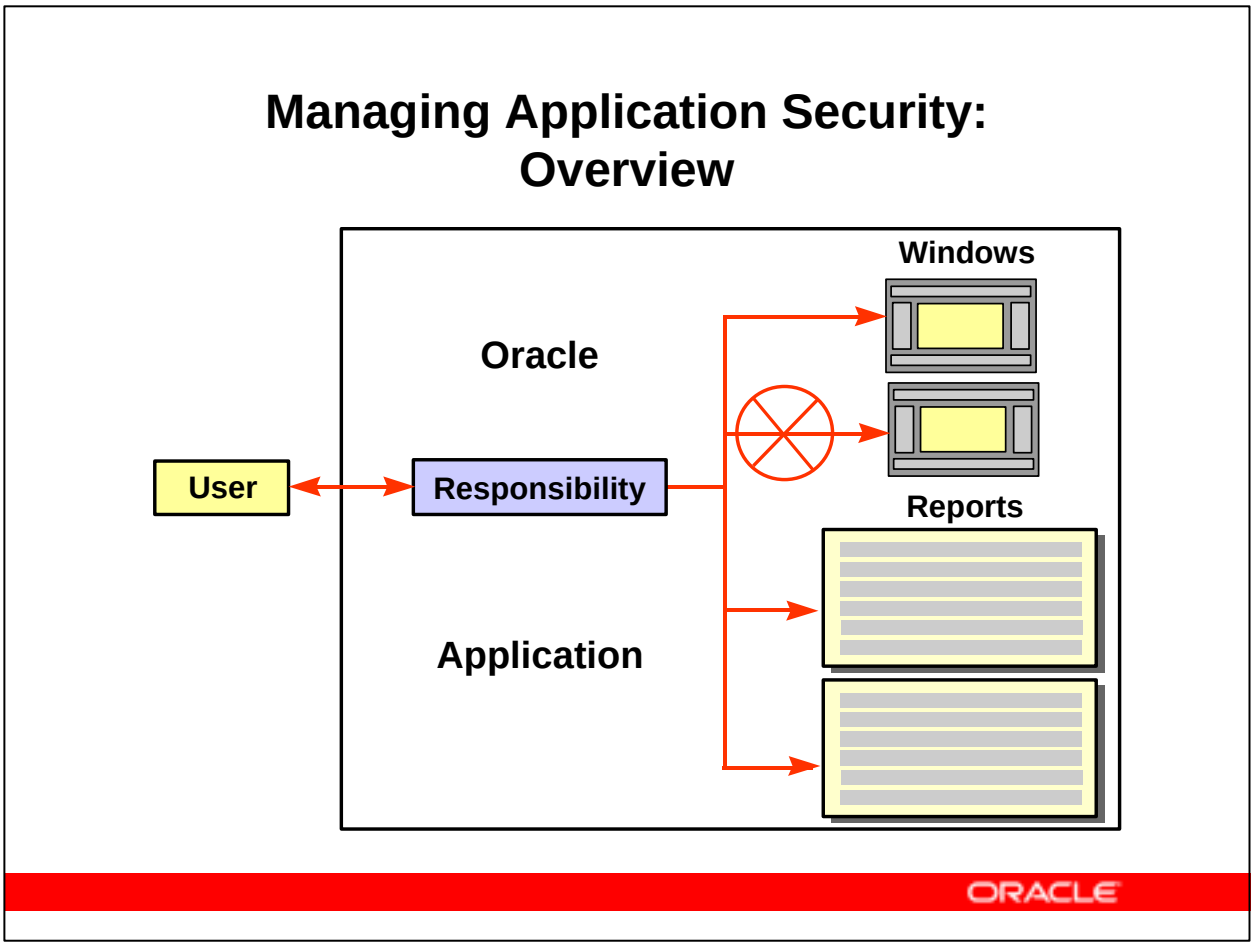
Managing Application Security

Chapter 2



Lesson Aim

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 users and relating them to one or more responsibilities. The responsibility specifies the actual access authorizations. This lesson shows you how to define both users and responsibilities.



Managing Application Security: Overview

In Oracle Applications, the system administrator manages security by creating users and assigning one or more responsibilities to them. Users then have access to all the functionality associated with that responsibility.

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:

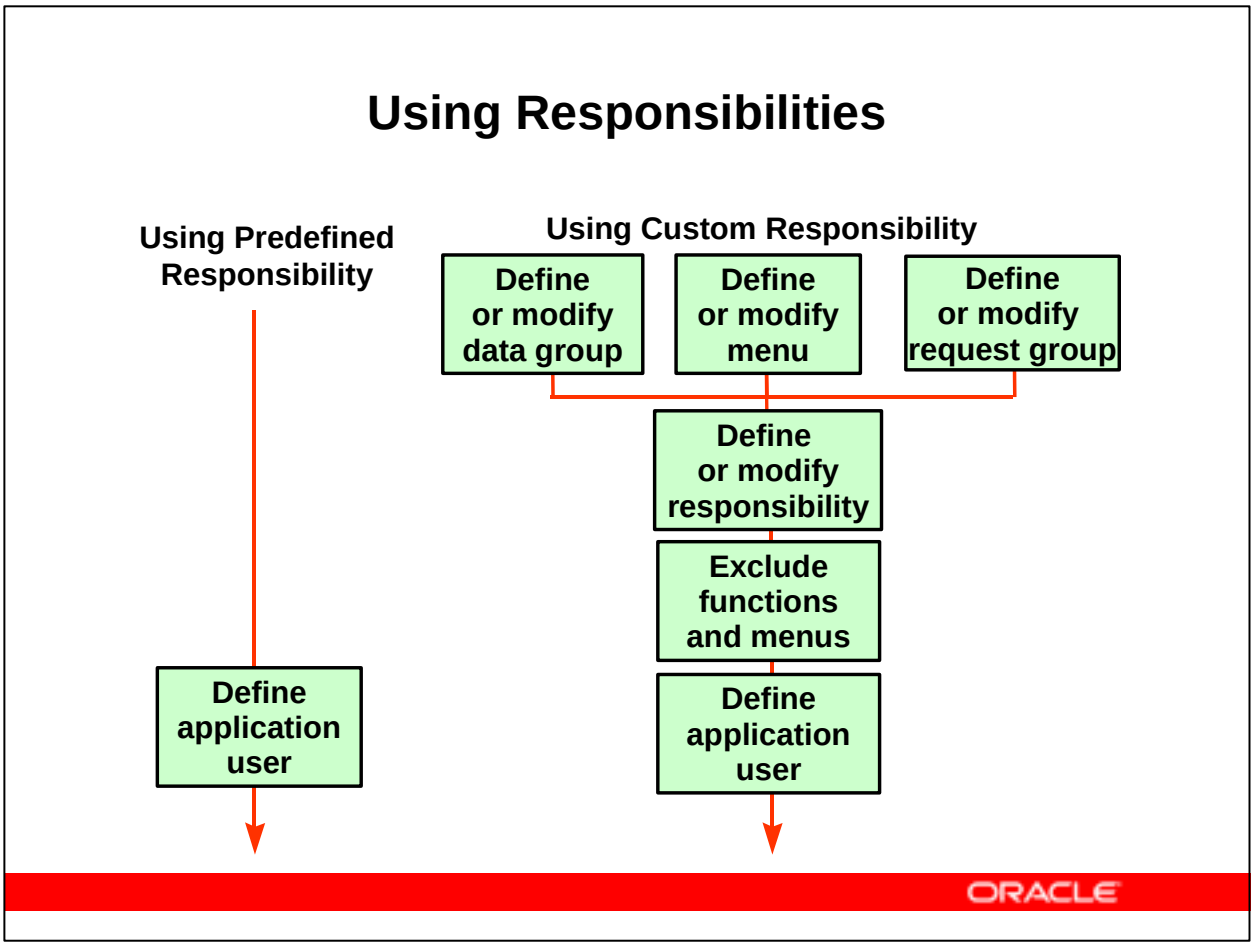
- A specific application or applications
- A set of books
- A restricted list of windows, functions, and reports

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

A system administrator can assign users any of the standard responsibilities provided with Oracle Applications or create new custom responsibilities as needed.

Self-Service Applications Security

Oracle Self-Service Web Applications use columns, rows, and values in database tables to define what information users can access. Table columns represent “attributes” that can be assigned to a responsibility. These attributes are defined in the Web Application Dictionary.



Creating a New Responsibility

Generally you relate new application users to existing predefined responsibilities. However, you can customize an existing responsibility or create new responsibilities to accommodate the needs of different users or different categories of users. When creating a new responsibility, it is generally easier to modify an existing responsibility in one of two ways:

- **Extend Privileges That a Responsibility Owns**

  When users require additional reporting and summary information, you can:

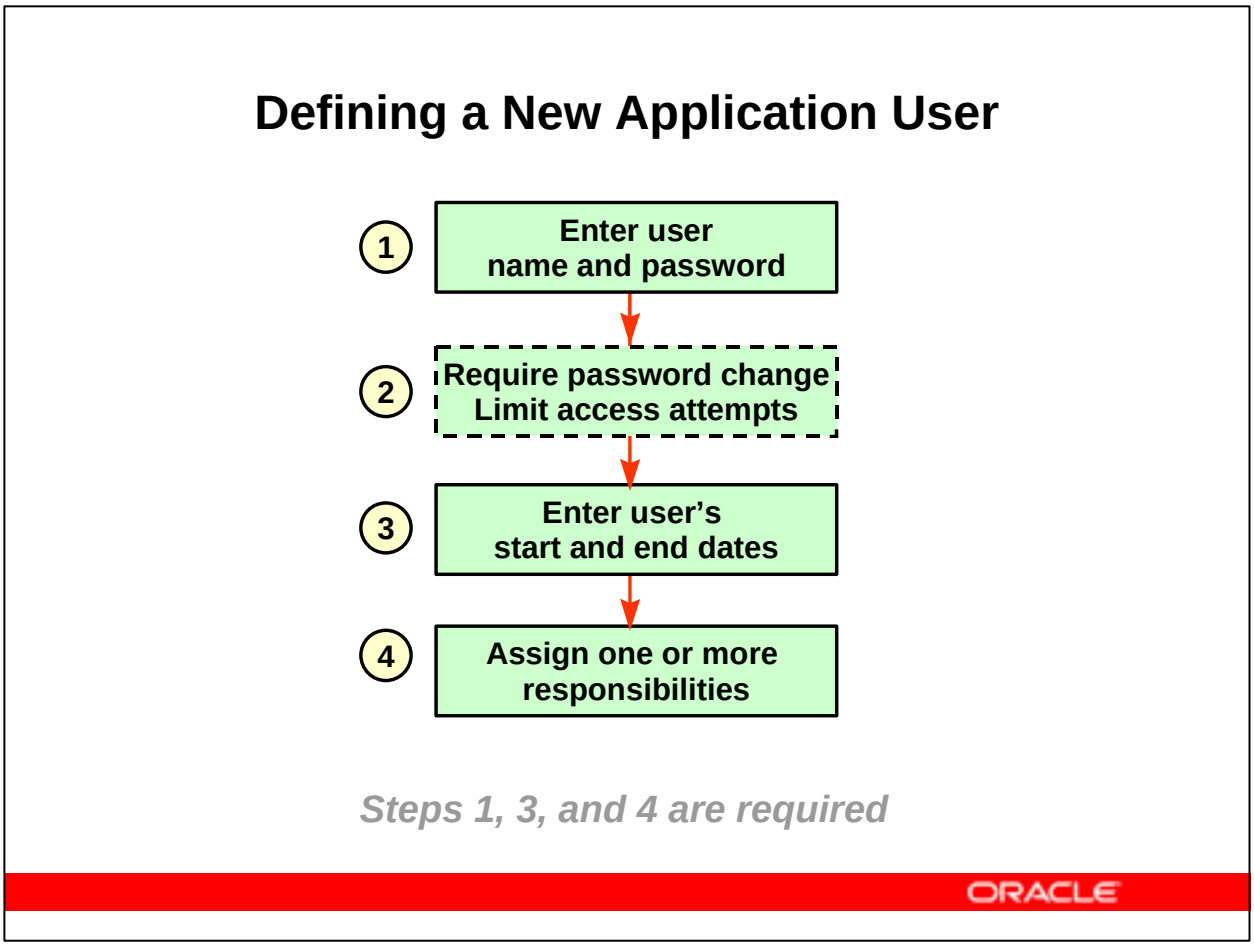
  - Use request groups to add additional program or report privileges to a responsibility
  - Use menus to add windows and tasks to a responsibility

- **Restrict Privileges That a Responsibility Owns**

  Sometimes it is easier to remove authorizations from an existing responsibility. To do this you can:

  - Use exclusions against a responsibility to limit menu and function access privileges to those required for job duties
  - Use request groups to limit program or report privileges for a responsibility that requires only data entry privileges

After you have defined a new responsibility, you can associate it with an application user.



Defining a New User

(N) Security > User > Define

Note: All Navigation paths, unless otherwise noted, are from the System Administrator Responsibility.

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.

For a complete explanation of the fields on the Users Form see:

(Help) Applied Technology > Oracle Applications System Administration > Overview of Oracle Applications Security > Users Window

Practice - Users

Overview

You are part of the final acceptance testing committee. Specifically, you have been assigned a series of System Administration tasks that you will be implementing over the course of this final week of testing.

Your task is to create an Application user that you will utilize over the course of your testing.

- Create the user
- Assign responsibilities to the user
- Test the user

If order to facilitate all the testing you will accomplish over this week, you will preface ALL your exercises with your initials (e.g., WHS). This will help you find the specific data you have created.

Tasks

Create the User

1. Use the information below to define your new user signon.
    - User Name: *Your Initials* User (e.g., WHS User)
    - Description: *Your Name* System Administrator
    - Password: WELCOME (re-enter to verify)
    - Password Expiration: 30 days
    - Effective Dates: From: today
    - Effective Dates: To: 3 months from today

Assign Responsibilities

2. Assign the following Responsibilities to your user.
    - System Administrator
    - System Administration
    - Application Developer

- General Ledger Super User

3. Accept the defaults for all other fields.

Test your User

4. Sign on again using your username with WELCOME as the password.

5. Change the password when prompted. You change the password by typing the old password, WELCOME, and then assigning a new password. You will re-type the new password to verify.

6. Verify that all the responsibilities you defined in Step 2 are available.

SOLUTION - USERS

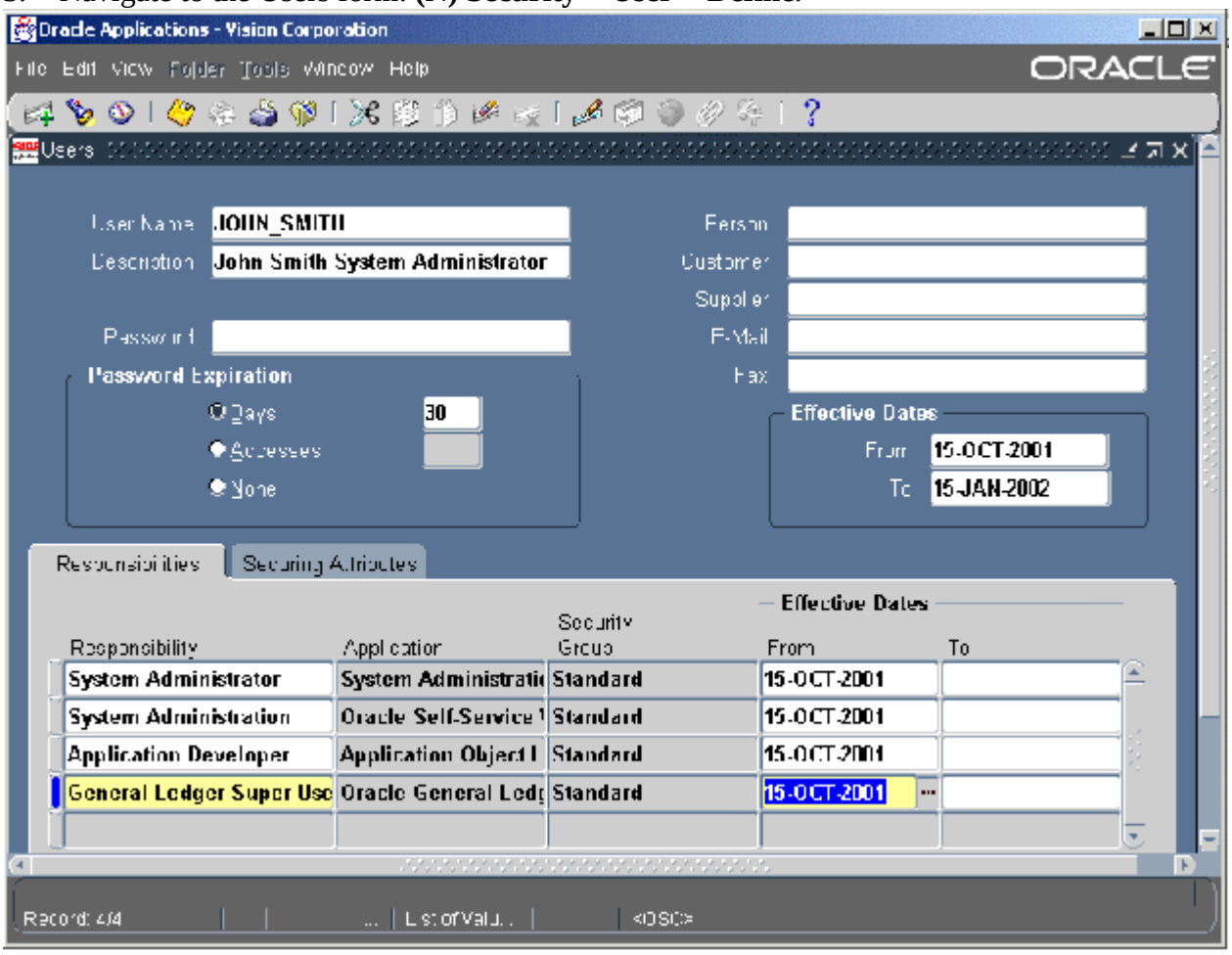
Create the User

Responsibility = System Administrator, Vision Operations

1. Log in to Oracle Applications with user name “SYSADMIN” and password “SYSADMIN”.

2. Select the **System Administrator** responsibility under the “Applications” heading of the Navigate region of the Personal Homepage.

3. Navigate to the Users form: **(N) Security > User > Define**.



4. Enter the fields as described in the instructions.

   - **User Name:** *Your Initials* User (e.g., WHS User)
   - **Description:** *Your Name* System Administrator
   - **Password:** WELCOME (re-enter to verify)

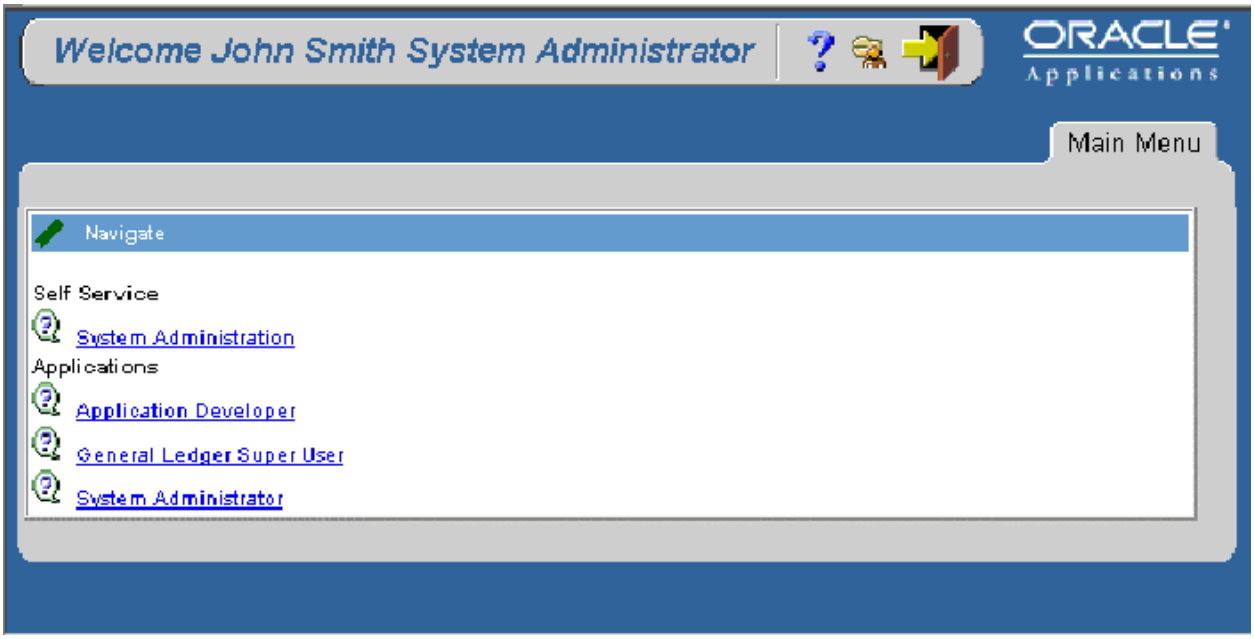
- **Password Expiration:** 30 days
- **Effective Dates:** From: Today
- **Effective Dates:** To: 3 months from today

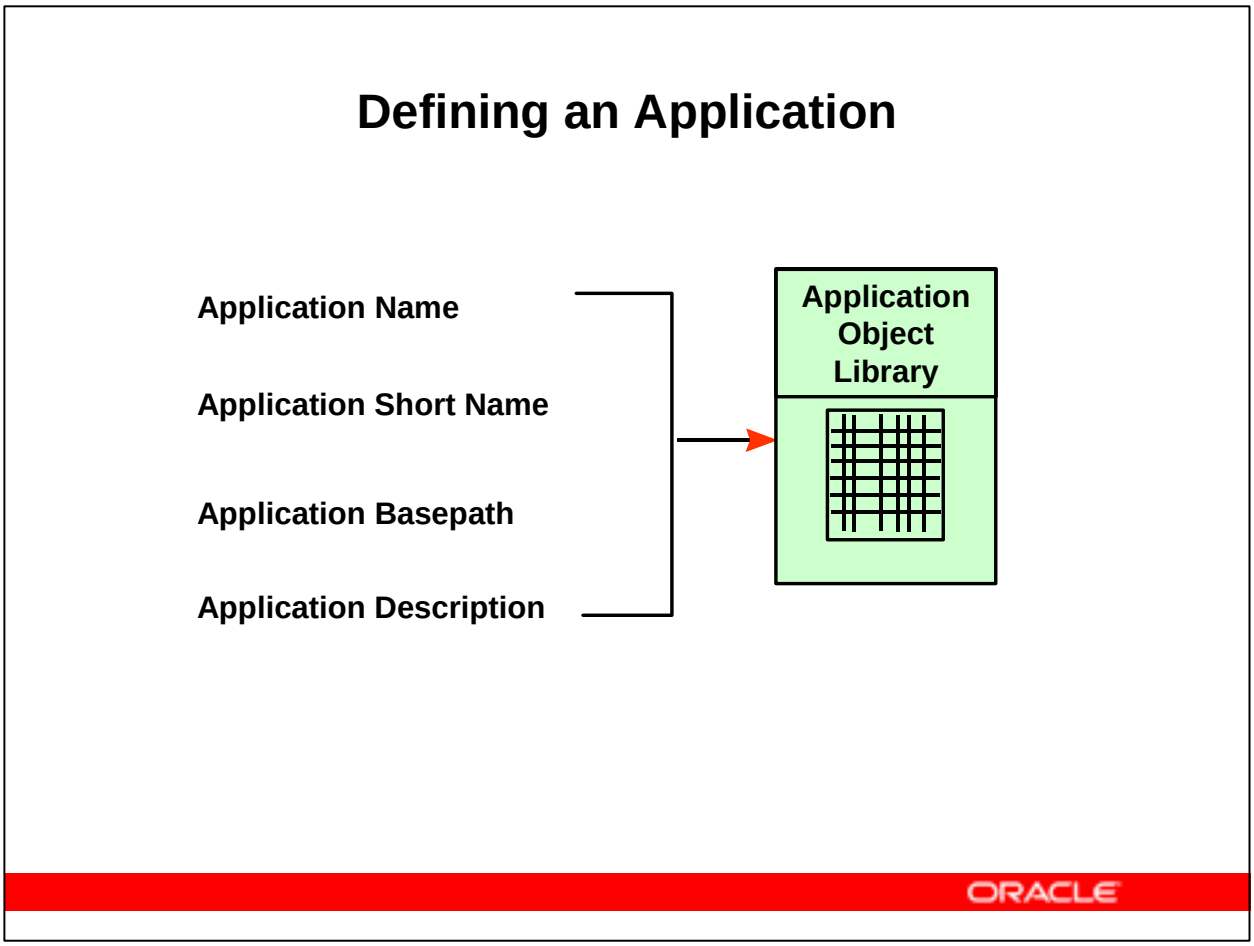
Assign Responsibilities

5. Assign the following Responsibilities to your user.
   - System Administrator
   - System Administration
   - Application Developer
   - General Ledger Super User
6. Click the **Save** icon to save your work. Your completed form should look similar to the example shown in the slide.

Test your User

7. Close the Users form. Choose **File > Exit Oracle Applications** to return to the Personal Homepage.
8. Click on the Exit icon (arrow pointing through the doorway on the Personal Homepage) to return to the signon window.
9. Use your new login and password (WELCOME) to log in again. You will be prompted to change your password. Be sure to note your user name and new password for use in later practices.
10. Verify that you have the correct Responsibilities displayed, and select the System Administrator Responsibility.





Defining an Application

(N) Application > Register

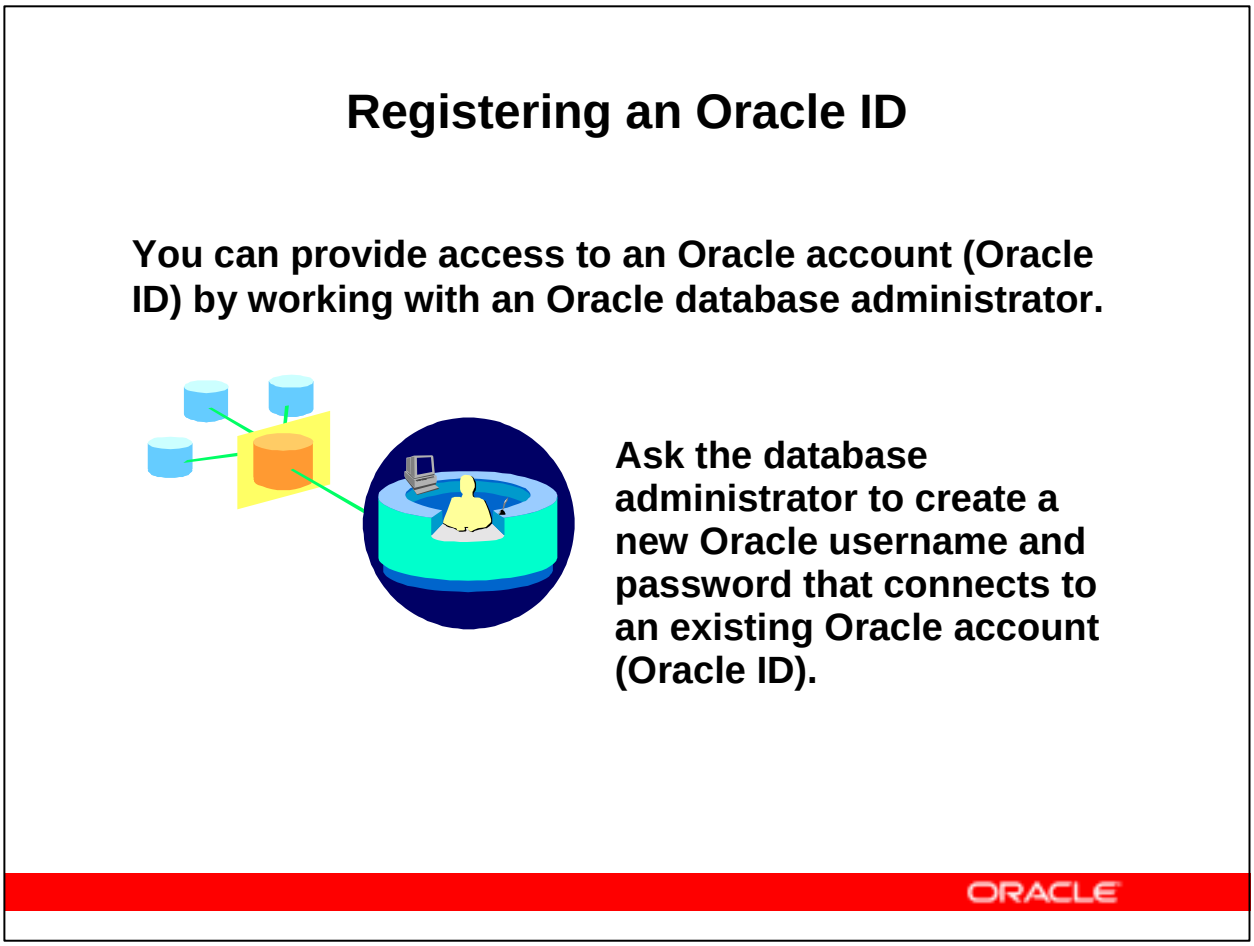
You can protect custom functions, forms, reports, and programs from being lost during upgrades by registering them.

In the Applications window you will supply the following information:

- Application – Enter a user-friendly name that will appear in lists seen by the user.
- Short Name – Oracle Applications uses this short name to identify forms, menus, concurrent programs, and other components of your application.
- Basepath – Enter the base path where your custom forms, reports, and program files are located. Make sure that your base path is unique to prevent other applications from writing to the same directory.

For a complete explanation of the fields in this form see:

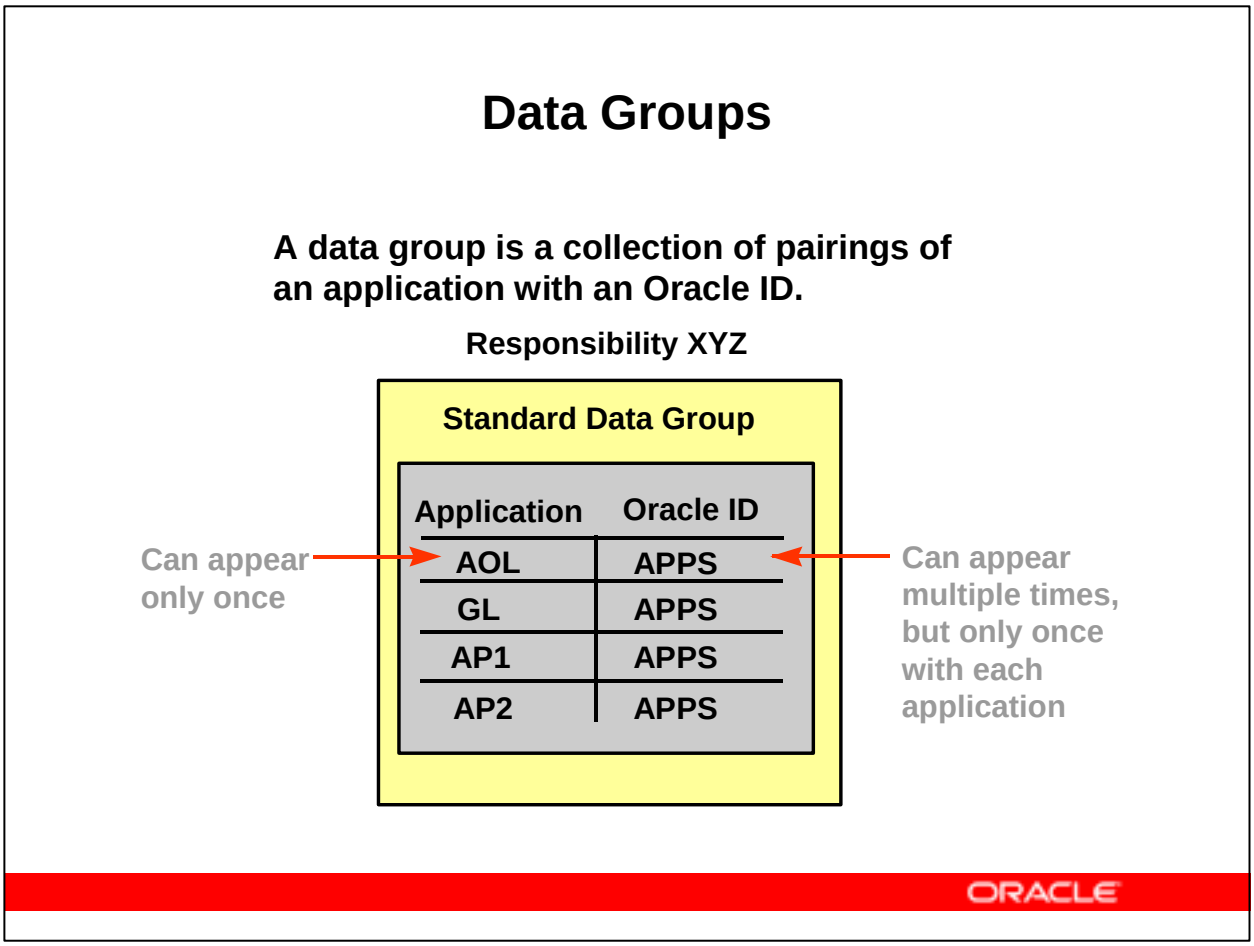
(Help) Applied Technology > Oracle Applications System Administration > Applications DBA > Applications Window



Registering an Oracle ID

(N) Security > ORACLE > Register

- Only database administrators can create Oracle accounts.
- The install group designates which data group the application is associated with. For applications that span all data groups, the install group is 0. Install groups numbered 1 or greater are associated with one specific data group. Your database administrator can supply the correct install group number.



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.

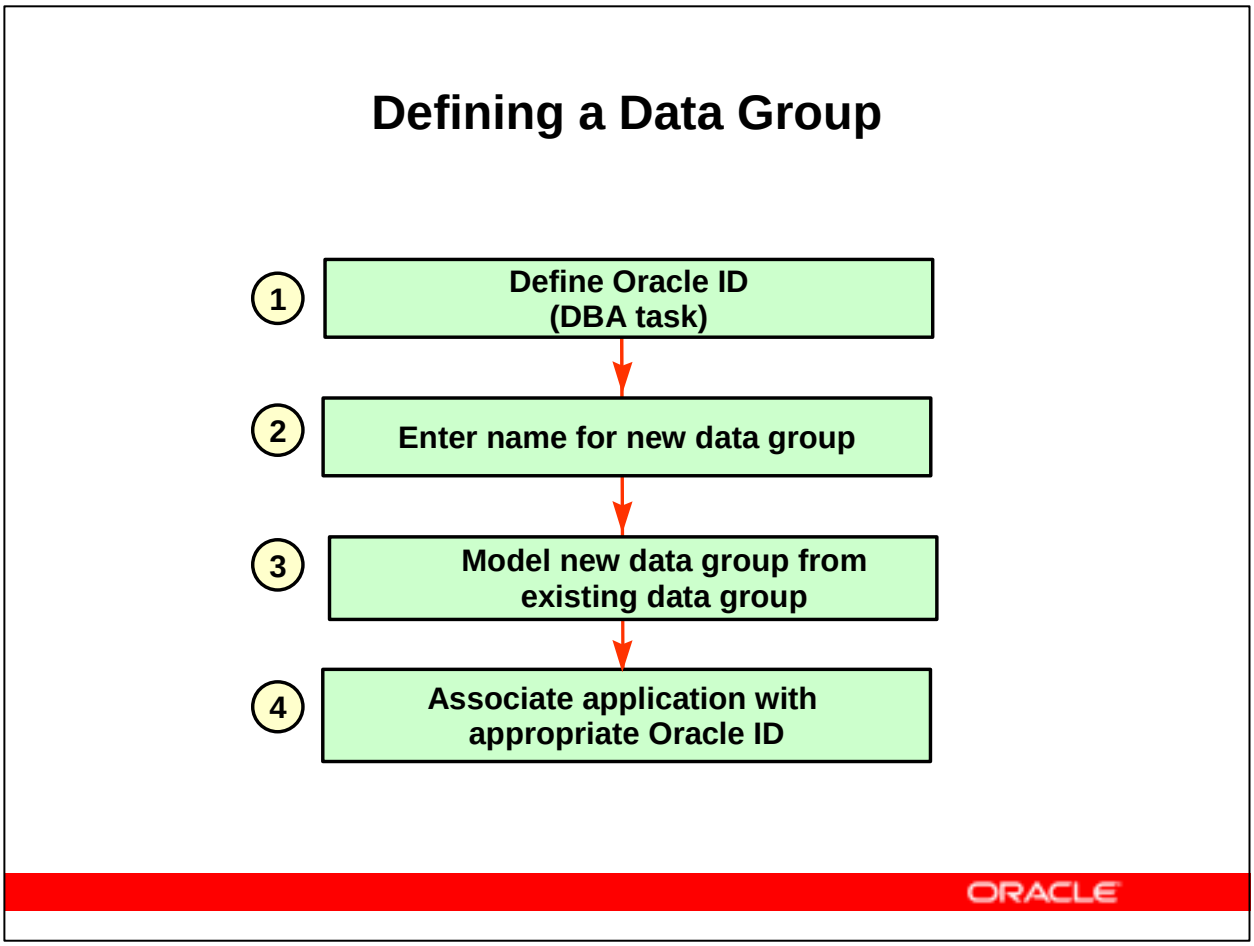
Note: The installation process automatically defines data groups for Oracle Applications, so you only need to define additional data groups based on your specific requirements.

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.



Defining a Data Group

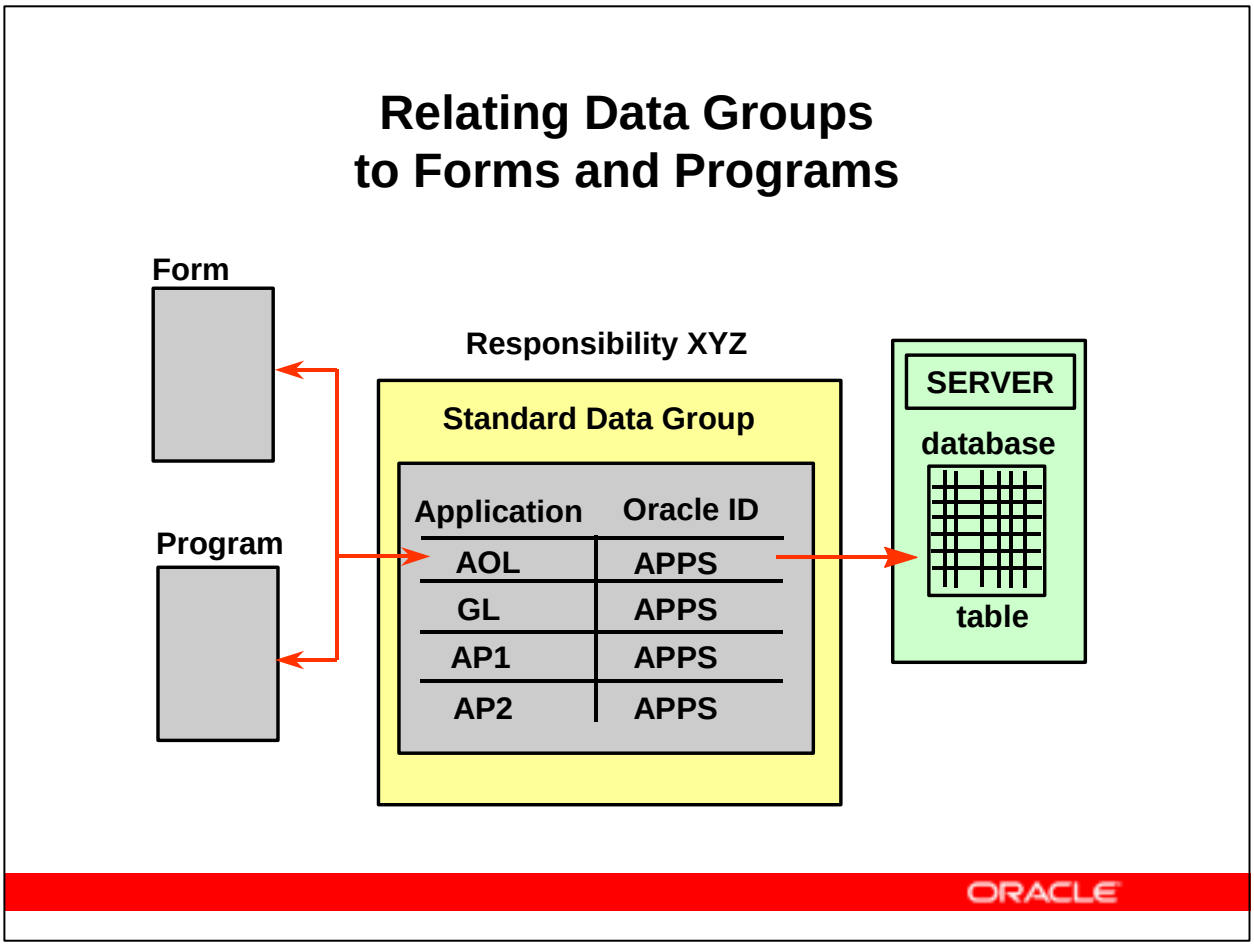
(N) Security > ORACLE > DataGroup

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.

Data Groups Serve Two Purposes

- Identify the Oracle ID to which forms connect when users select the responsibility. Choose one application-Oracle ID pair from the data group to associate with your responsibility's windows.
- Concurrent managers use a data group to match the application that owns a report or concurrent program with a unique Oracle ID.

Additionally, you can incorporate custom applications by granting application database access to custom applications that you develop. Alternatively, use custom applications to name customized versions of concurrent programs, responsibilities, and concurrent managers to protect your customizations during an upgrade.



Relating Data Groups to Forms and Programs

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

Applications, Forms, and Programs

- A window connects to the application database account designated by the responsibility associated with the application.
- A data group determines the pairing of an application with a unique application database account or Oracle ID.
- A program connects to the application database account associated with the application that owns the program.

Application-Oracle ID Pairs

- An Oracle ID is a username and password that allows access to application tables in an Oracle database.
- A data group lists the Oracle ID assigned to each Oracle application.
- A custom application registered with Oracle Applications can be included in a data group and paired with an Oracle ID.

Practice - Data Groups

Overview

Now that you have a functional user, your next assignment is to create a test area for customizations. In order to do that, you will have to complete several steps.

- Register your custom application
- Create a new data group
- Associate your new application to your new data group

Tasks

Register your Custom Application

1. Create your new custom application. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Application:** *Your Initials* Custom Application
   - **Short Name:** *Your Initials*CA (e.g., WHSCA)
   - **Base Path:** FND_TOP (**Note:** For class purposes we are using a predefined basepath. If you were defining a true custom application, this would be a unique basepath.)
   - **Description:** *Your Initials* Custom Application

Create your Data Group

2. Create your new data group. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Application:** *Your Initials* Data Group
   - **Click** the "**Copy Applications From …**" button to copy the Applications from the Oracle **Standard** data group to your custom data group.

Associate Application to Data Group

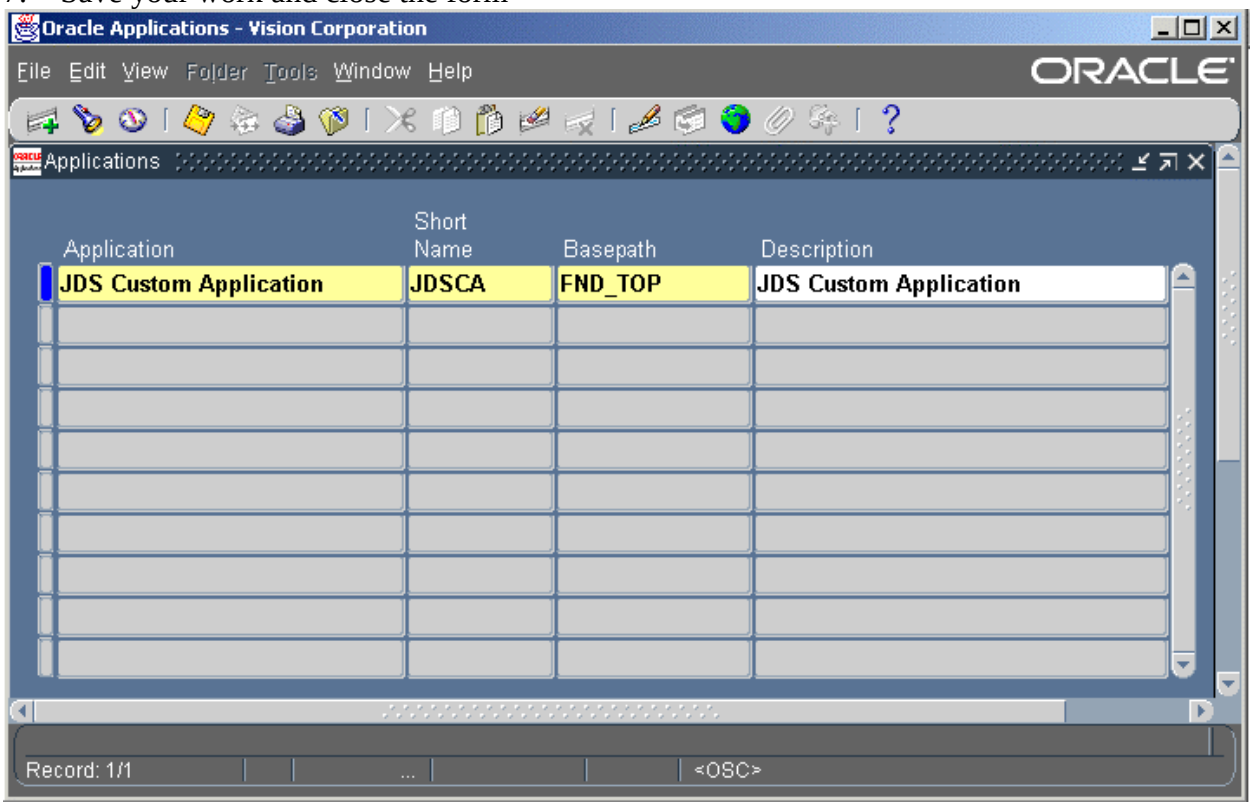
3. **Insert** a new record to add your custom application.
4. Save your record.

SOLUTION – DATA GROUPS

Register your Custom Application

Responsibility = System Administrator

1. If not currently logged in, log in to Oracle Applications with your user.
2. Navigate to (N) Application > Register
3. Enter your **Application**: *Your Initials* Custom Application.
4. Enter your **Short Name**: *Your Initials*CA (e.g., WHSCA)
5. Enter the **Basepath**: FND_TOP.
6. Enter your **Description**: *Your Initials* Custom Application.
7. Save your work and close the form



Create your Data Group

1. Navigate to (N) Security > ORACLE > DataGroup.
2. Enter your **Data Group**: *Your Initials* Data Group.

3. Enter your **Description**: *Your Initials* Data Group.

4. Place your cursor in the Application field and click the "**Copy Applications From…**" button.

5. Select "**Standard**" to copy the applications from the Oracle-provided data group to your custom data group.

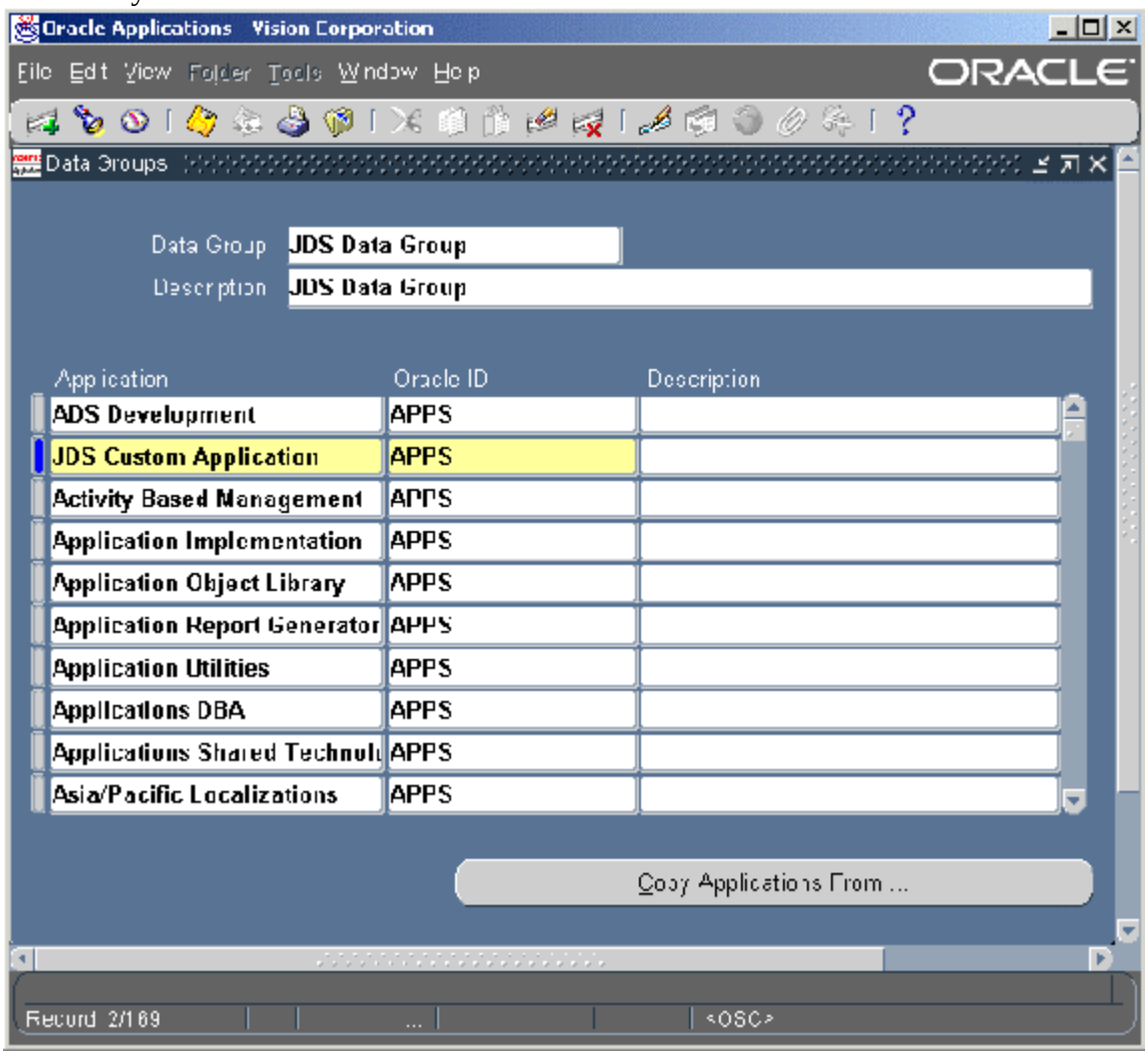
Associate Application to Data Group

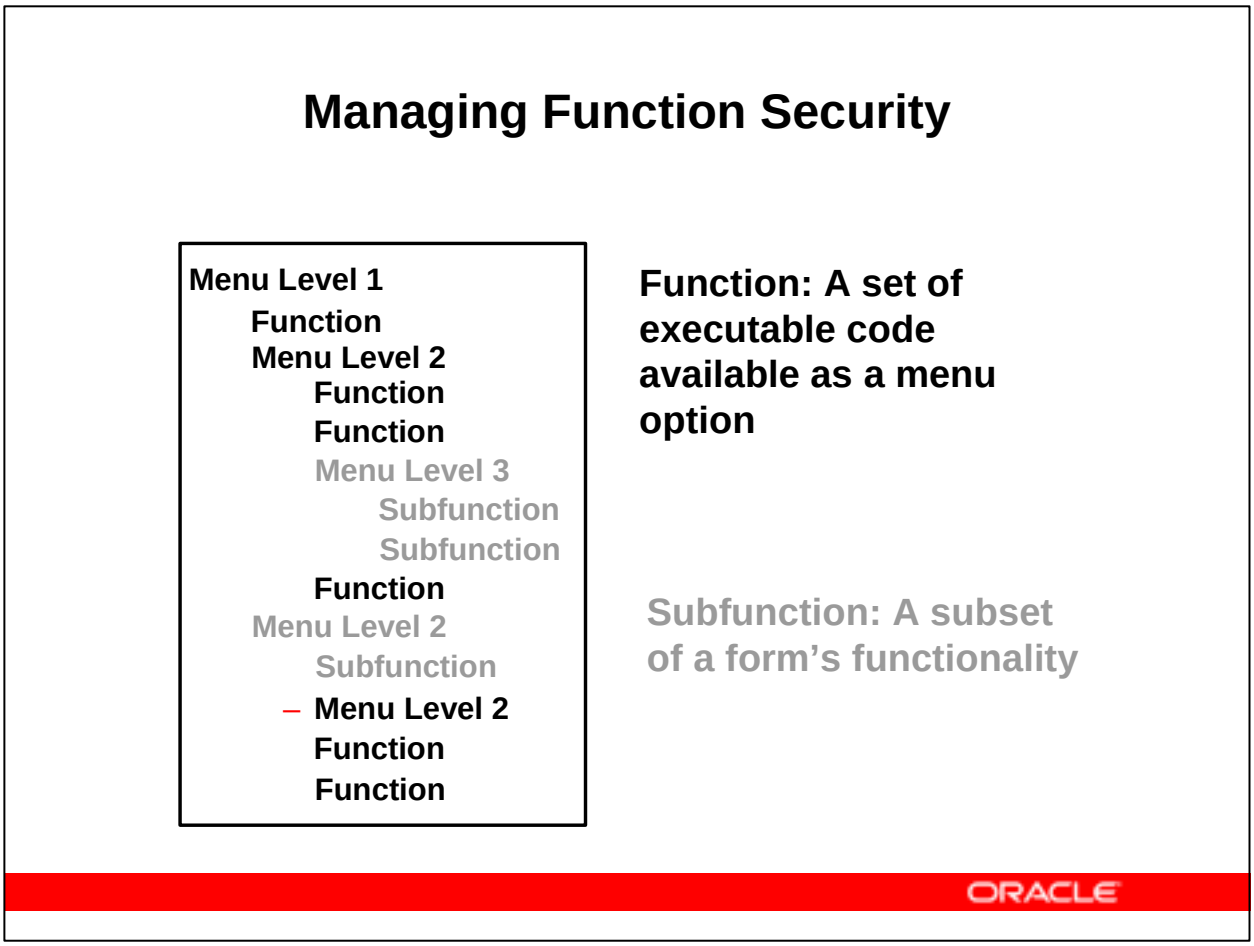
6. Ensure that your cursor is in the Application field and click the New toolbar icon or choose New from the File menu to add a row for your custom application.

7. Select your custom application name from the list of values for the Application field.

8. Select APPS from the list of values for the Oracle ID field.

9. Save your work.





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.

Web Regions

The fields in the Web regions are required only if your function will be accessed from Oracle Self-Service Web Applications.

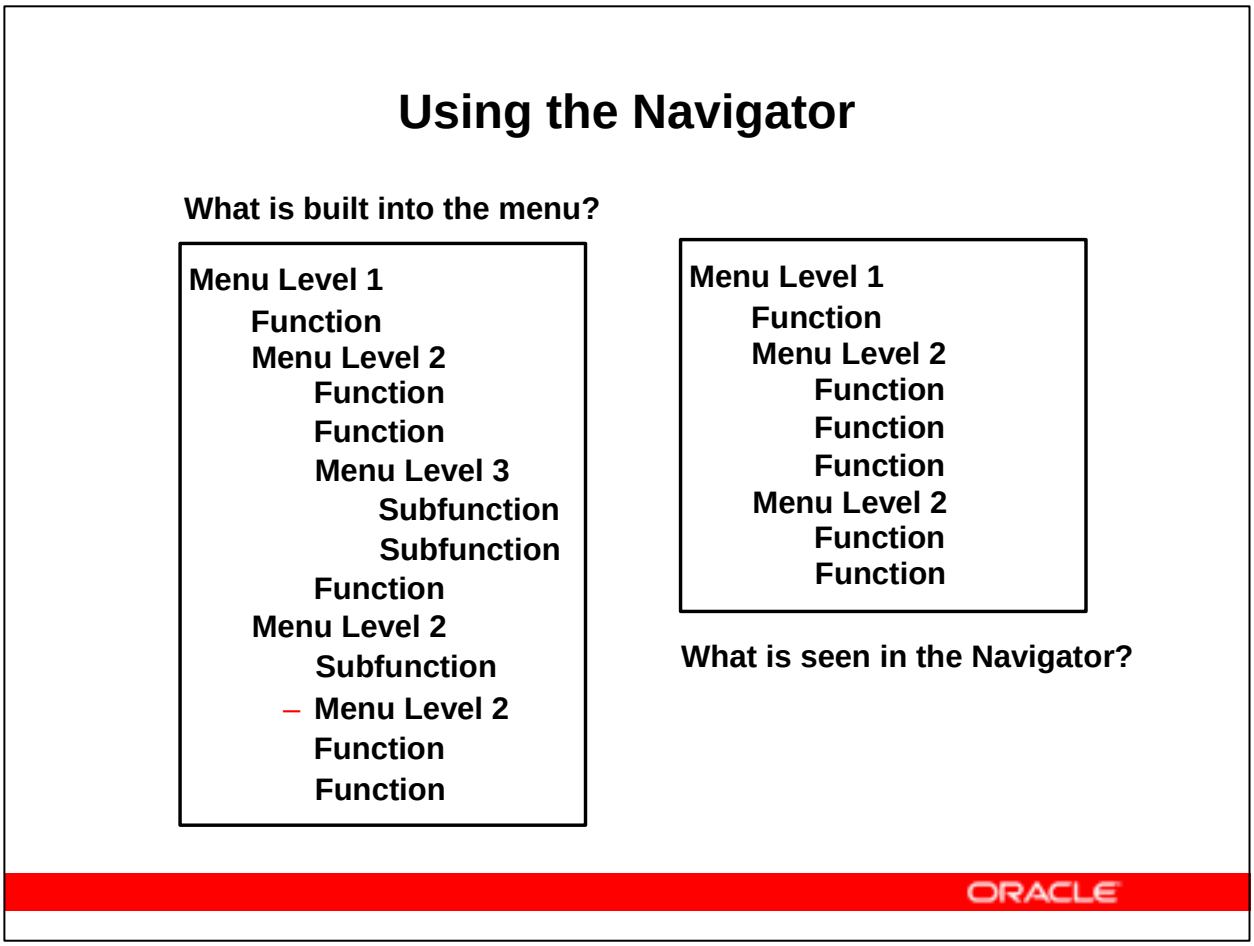
Adding Functions to or Removing Functions from a Responsibility

- Maintain menu structures while eliminating specific functionality.
- Exclude individual functions from a responsibility.

Adding or Removing Menus of Functions

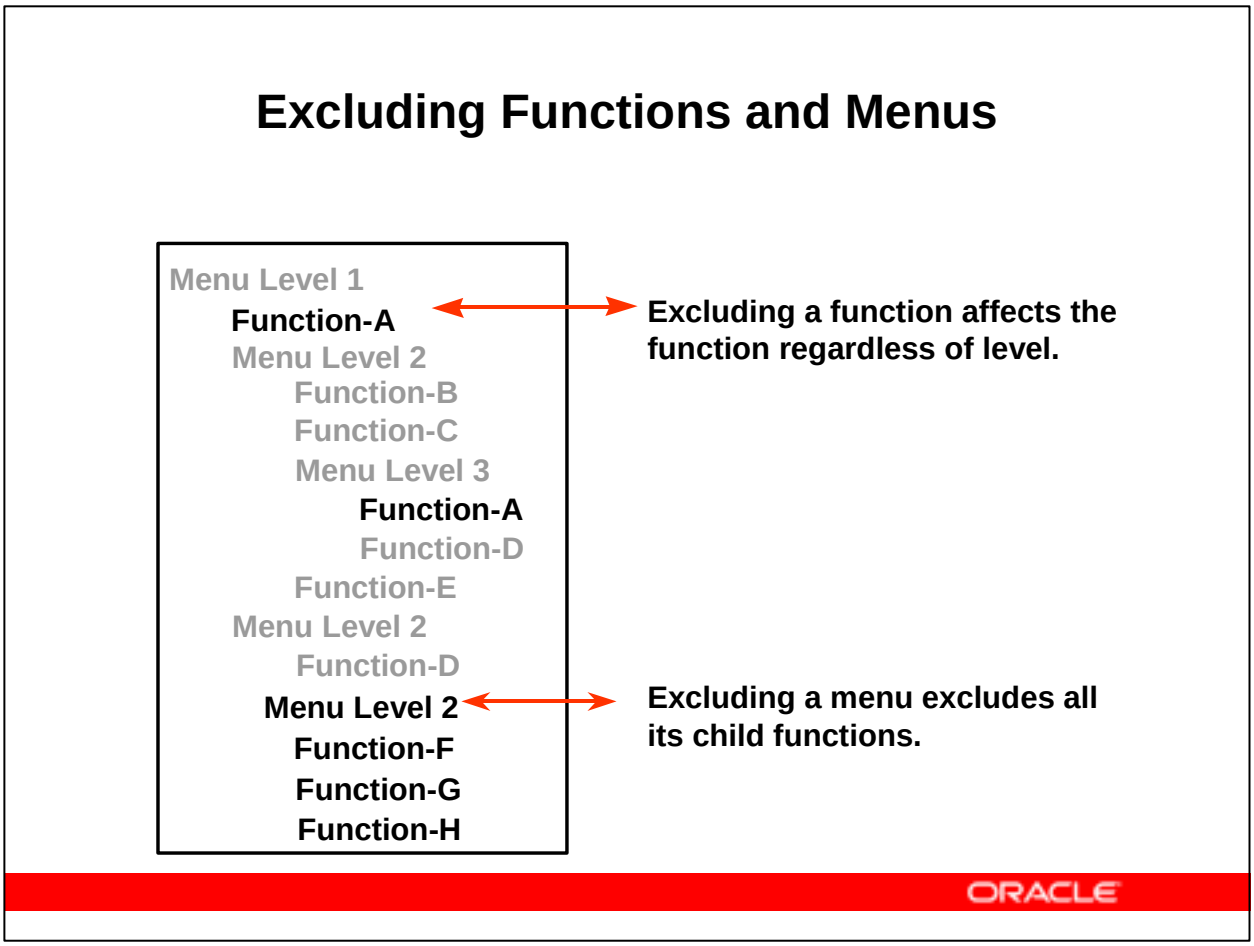
- Use menus to group functions together.

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



Navigator Menu Displays

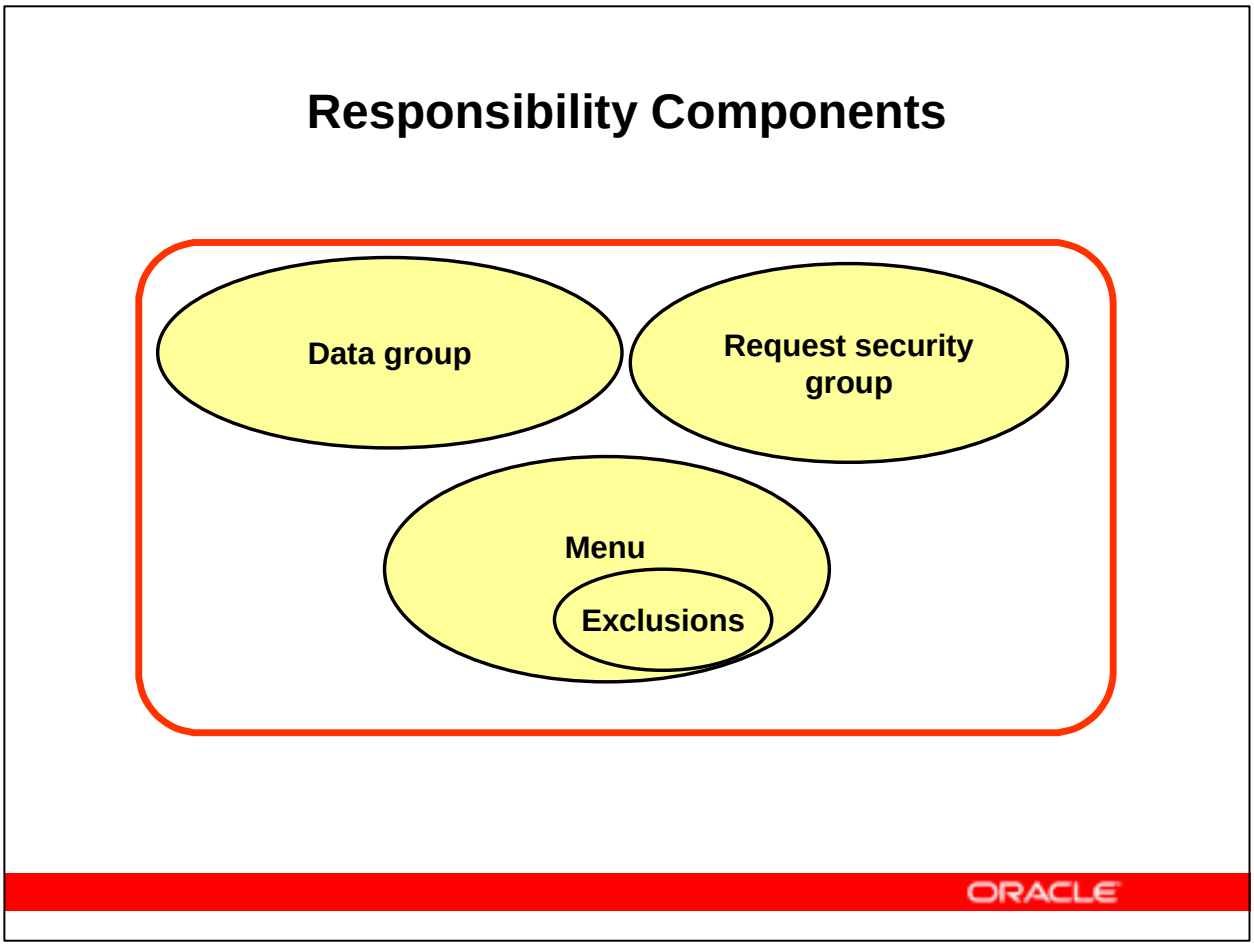
The Navigator displays only the menu items needed for navigation. Because you cannot choose subfunctions from a menu, they are not displayed. Submenus consisting only of subfunctions are also not displayed.



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.



Components of a Responsibility

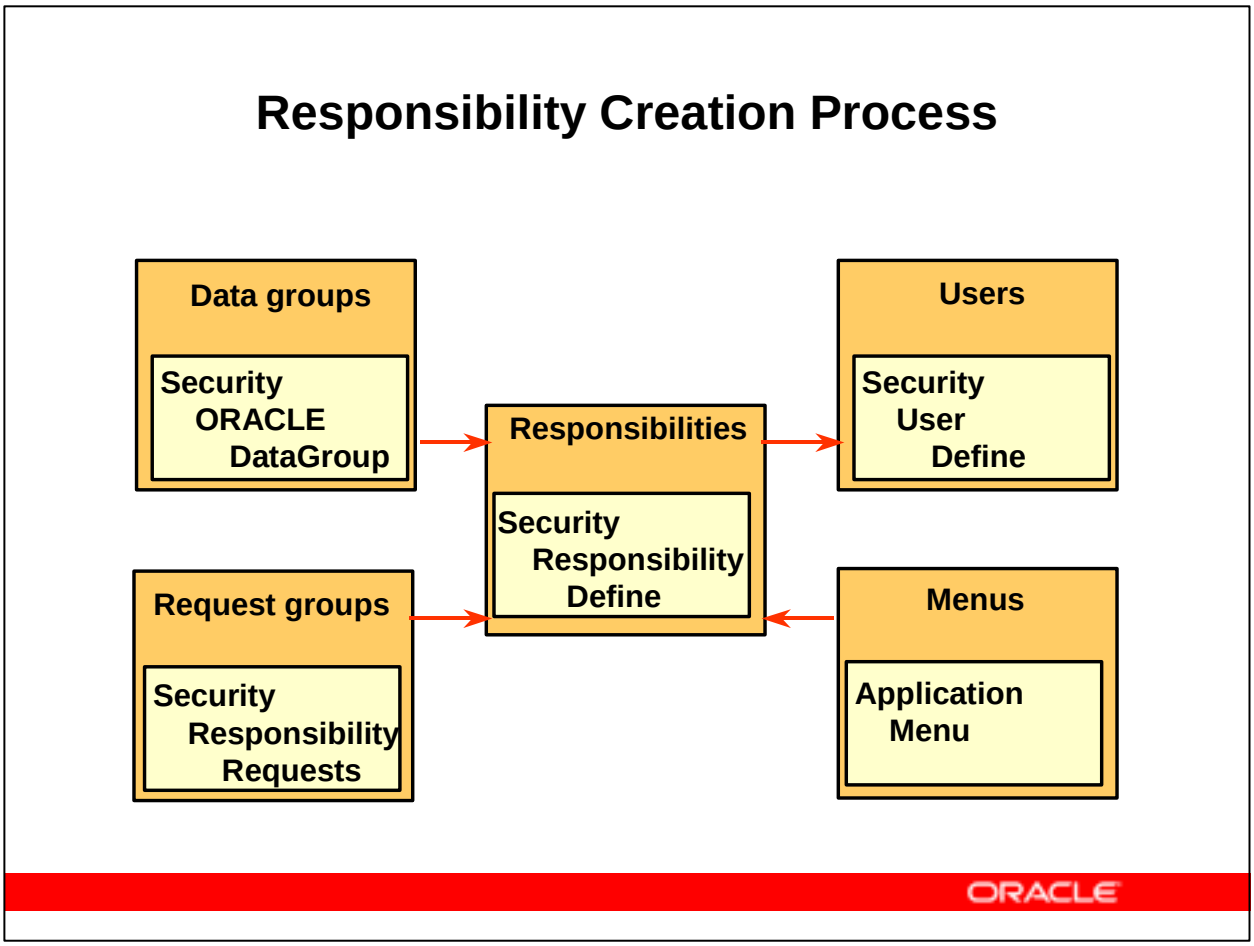
A responsibility has two required components and two optional components.

Required Components

- Data group: A data group specifies the Oracle Application database accounts to which a responsibility's forms and concurrent programs connect.
- Menu: A menu specifies the forms that a responsibility can display and the functions it can access.

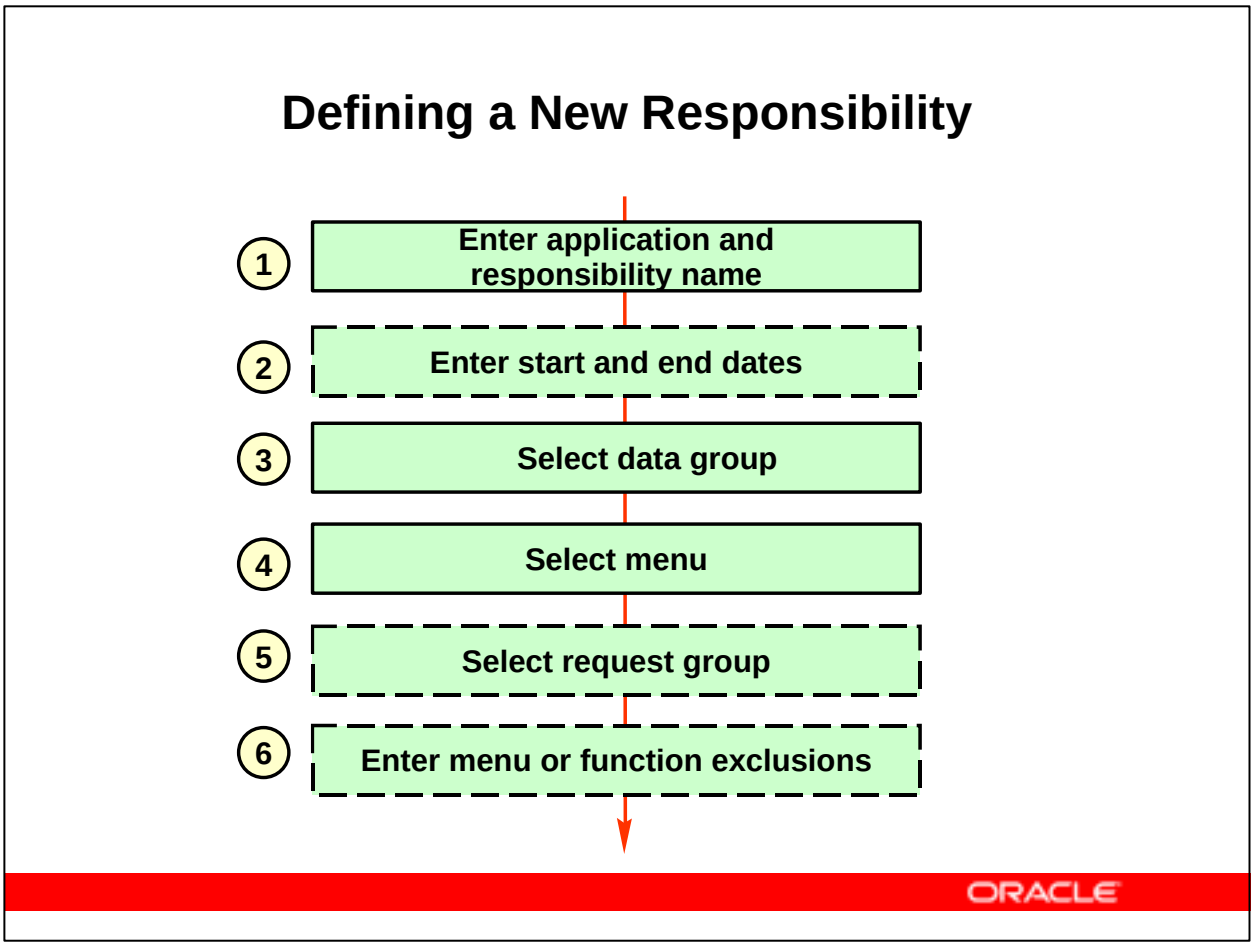
Optional Components

- Request security group: A request group lists the concurrent programs that a responsibility can run. When a request group is assigned to a responsibility, it is referred to as a request security group.
- Exclusions: Exclusions modify a responsibility's access to the forms and functions specified by a menu.



Responsibility Creation Process

There are five forms involved in the responsibility creation process. These are available under the System Administrator responsibility.



Defining a New Responsibility

- Assemble the components of application privileges to create a responsibility.
- Define the responsibility by assembling a menu, report security group, and data group and defining any function security (any menu or function exclusions).

Defining a Responsibility

(N) Security > Responsibility > Define

A responsibility determines if the user accesses Oracle Applications or Oracle Self-Service Web Applications, which application functions a user can use, which reports and concurrent programs the user can run, and which data those reports and concurrent programs can access.

Practice - Responsibilities

Overview

The committee wants you to test creating a special, limited responsibility for an assistant System Administrator. The steps you will follow are as follows:

- Create a new responsibility
- Modify the responsibility by excluding functions from its menu
- Assign the new responsibility to a new user
- Assign the new responsibility to your existing user

Tasks

Create New Responsibility

1. Create a new responsibility. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Responsibility Name**: *Your Initials* Assistant System Administrator
   - **Application**: Application Object Library
   - **Responsibility Key**: *Your Initials*_Asst_SysAdmin (e.g., WHS_Asst_SysAdmin)
   - **Description**: Assistant System Administrator
   - **Available From**: Oracle Applications
   - **Data Group**: *Your Initials* Data Group
   - **Data Group Application**: *Your Initials* Custom Application
   - **Menu**: Navigator Menu - System Administrator GUI
   - **Request Group**: (leave blank)

Exclude Functions from Menu

2. Modify access to certain menus and functions for your new Responsibility by excluding the items below. Menu or Function Name to EXCLUDE:
   - Function, Monitor Application Users

- Function, Responsibilities
- Function, Web Enabled PL/SQL
- Function, Oracle Support MetaLink
- Menu, NAVSECVAL4.0
- Menu, ORACLE Menu – System Administrator GUI
- Menu, AuditTrail Menu – System Administrator GUI
- Menu, Concurrent Menu – System Administrator GUI
- Menu, Profile Menu – System Administrator GUI
- Menu, Application Menu – System Administrator GUI
- Menu, Install Menu – System Administrator GUI
- Menu, Workflow Administrator

Assign Responsibility to New User

3. In order to test this new responsibility, create a new user signon for your assistant system administrator and assign your newly created responsibility to this user. Use the information in the listed below.
   - **User Name:** *Your Initials*AssistSA (e.g., WHSAssistSA)
   - **Description:** *Your Initials* Assistant System Administrator
   - **Password:** WELCOME (re-enter to verify)
   - **Choose the Responsibility:** *Your Initials* Assistant System Administrator
4. Once you have defined it, sign off and back on using the new signon. Check that all the specialized responsibilities and exclusions that you intended apply to this new user.

Add Responsibility to Existing User

5. Query your user (*Your Initials* User) you created, and add this responsibility to that user.

Solution - Responsibilities

Create New Responsibility

Responsibility = System Administrator

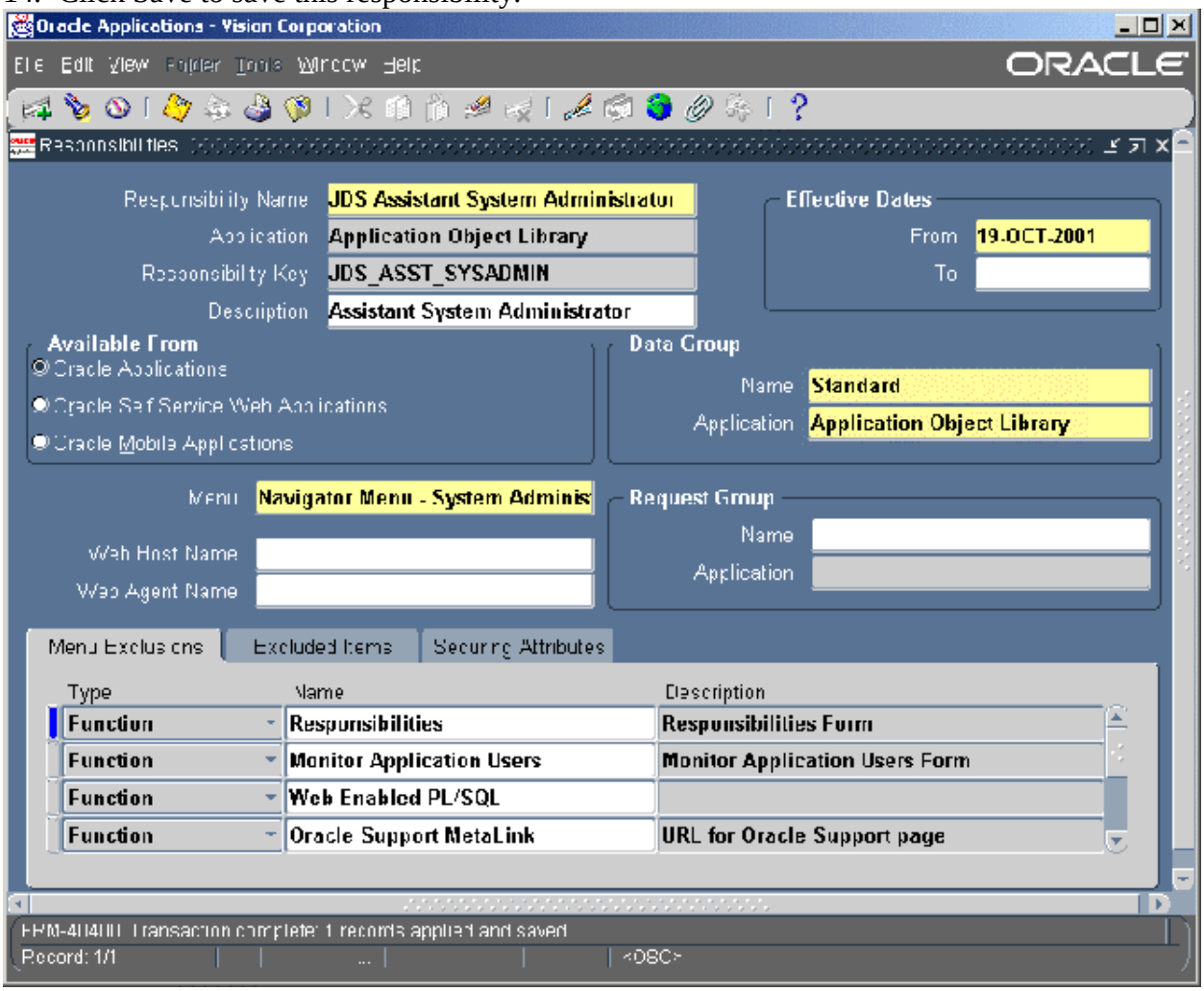
1. If not currently logged in, log in to Oracle Applications with your user.
2. Navigate to (N) Security > Responsibility > Define.
3. Enter the **Responsibility Name**: *Your Initials* Assistant System Administrator
4. Select the **Application**: Application Object Library
5. Enter the **Responsibility Key**: *Your Initials*_ASST_SYSADMIN
6. Enter the **Description**: Assistant System Administrator
7. Check **Available From**: Oracle Applications
8. Select the **Data Group**: *Your Initials* Data Group
9. Select the **Data Group Application**: *Your Initials* Custom Application
10. Select the **Menu**: Navigator Menu – System Administration GUI
11. Save your work.

Exclude Functions from Menu

12. Navigate to the Menu Exclusions tab of the window.
13. Click on Type of exclusion – Function or Menu – then use the list of values to select the appropriate function or menu to exclude in the name column. Continue until all items listed are excluded.
    - Function, Monitor Application Users
    - Function, Responsibilities
    - Function, Web Enabled PL/SQL
    - Function, Oracle Support MetaLink
    - Menu, NAVSECVAL4.0
    - Menu, ORACLE Menu – System Administrator GUI
    - Menu, AuditTrail Menu – System Administrator GUI

- Menu, Concurrent Menu – System Administrator GUI
- Menu, Profile Menu – System Administrator GUI
- Menu, Application Menu – System Administrator GUI
- Menu, Install Menu – System Administrator GUI
- Menu, Workflow Administrator

14. Click Save to save this responsibility.



15. Close the Responsibilities form.

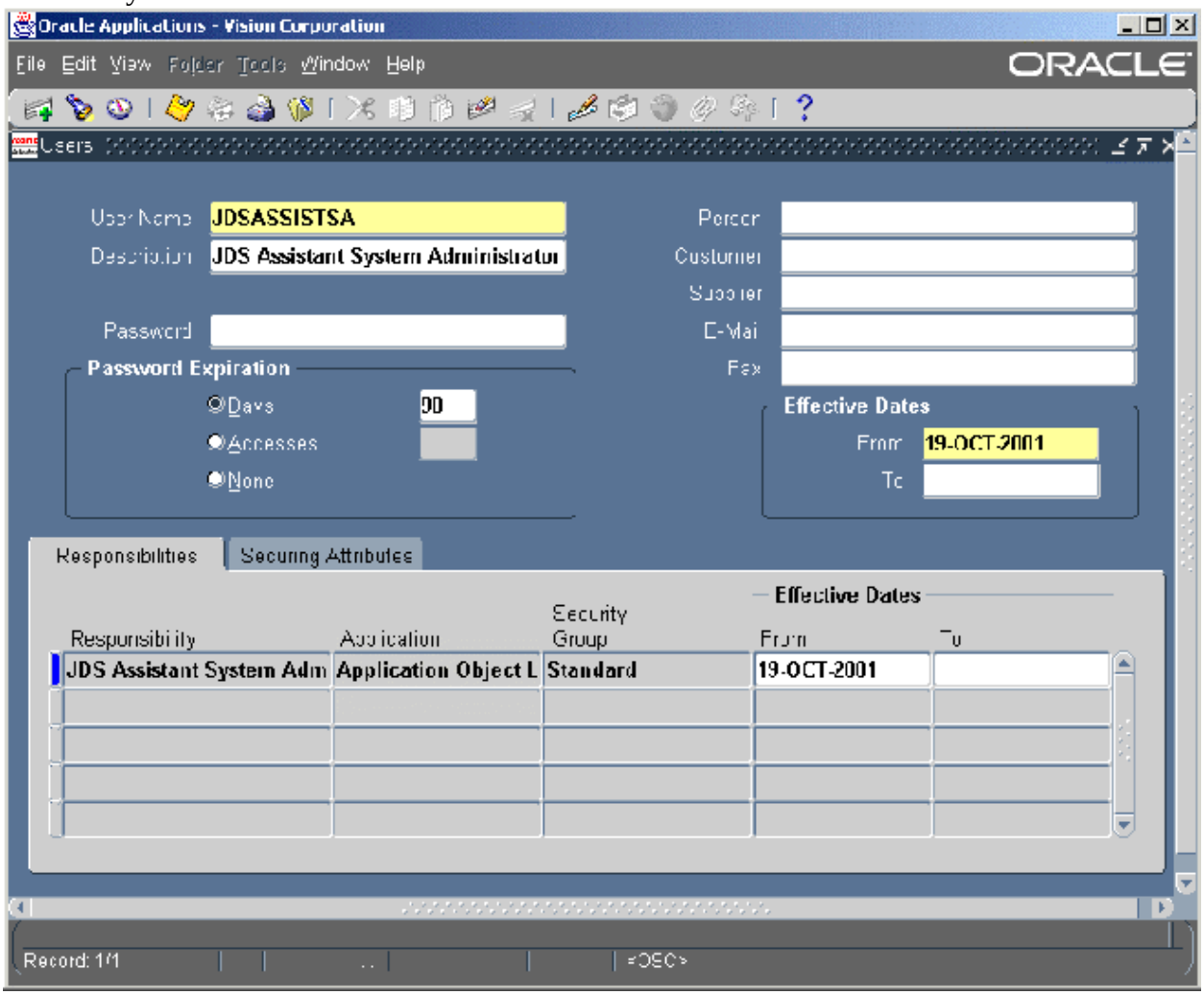
Assign Responsibility to New User

1. Navigate to (N) Security > User > Define
2. Enter the **User Name**: *Your Initials*AssistSA (e.g., WHSAssistSA)
3. Enter **Description**: *Your Initials* Assistant System Administrator

4. Enter **Password**: WELCOME (re-enter to verify)

5. Select **Responsibility**: *Your Initials* Assistant System Administrator

6. Save your work.



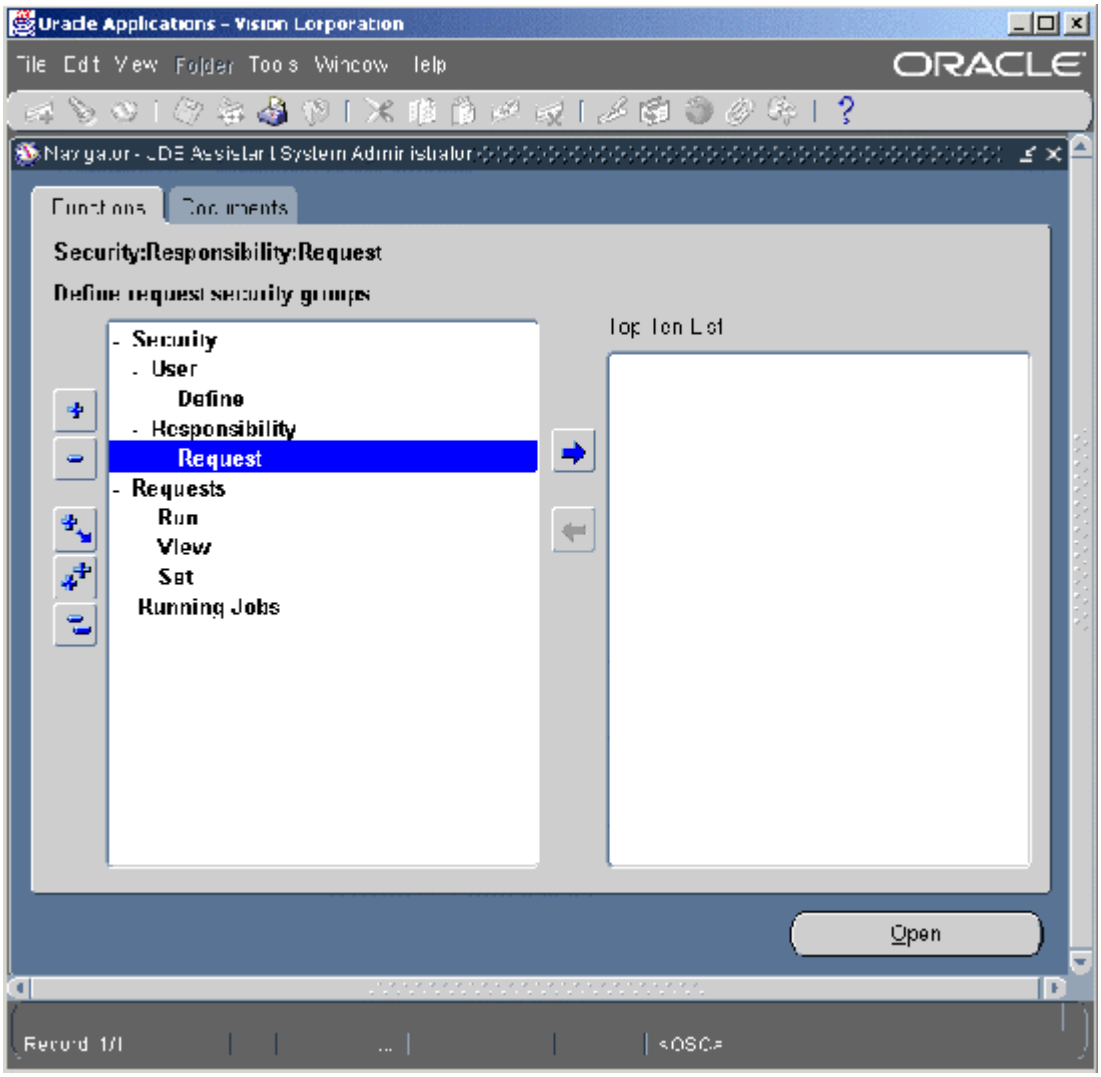
7. Close the Users form.

8. Exit Oracle Applications and exit the Personal Homepage.

9. Enter your new user name and password. You will be prompted to change your password.

10. From the Personal Homepage select *Your Initials* Assistant System Administrator responsibility.

11. Review the menu items for your new responsibility and verify that only the correct items appear.



Add Responsibility to Existing User

1. Navigate to the Users form (N) Security > User > Define.

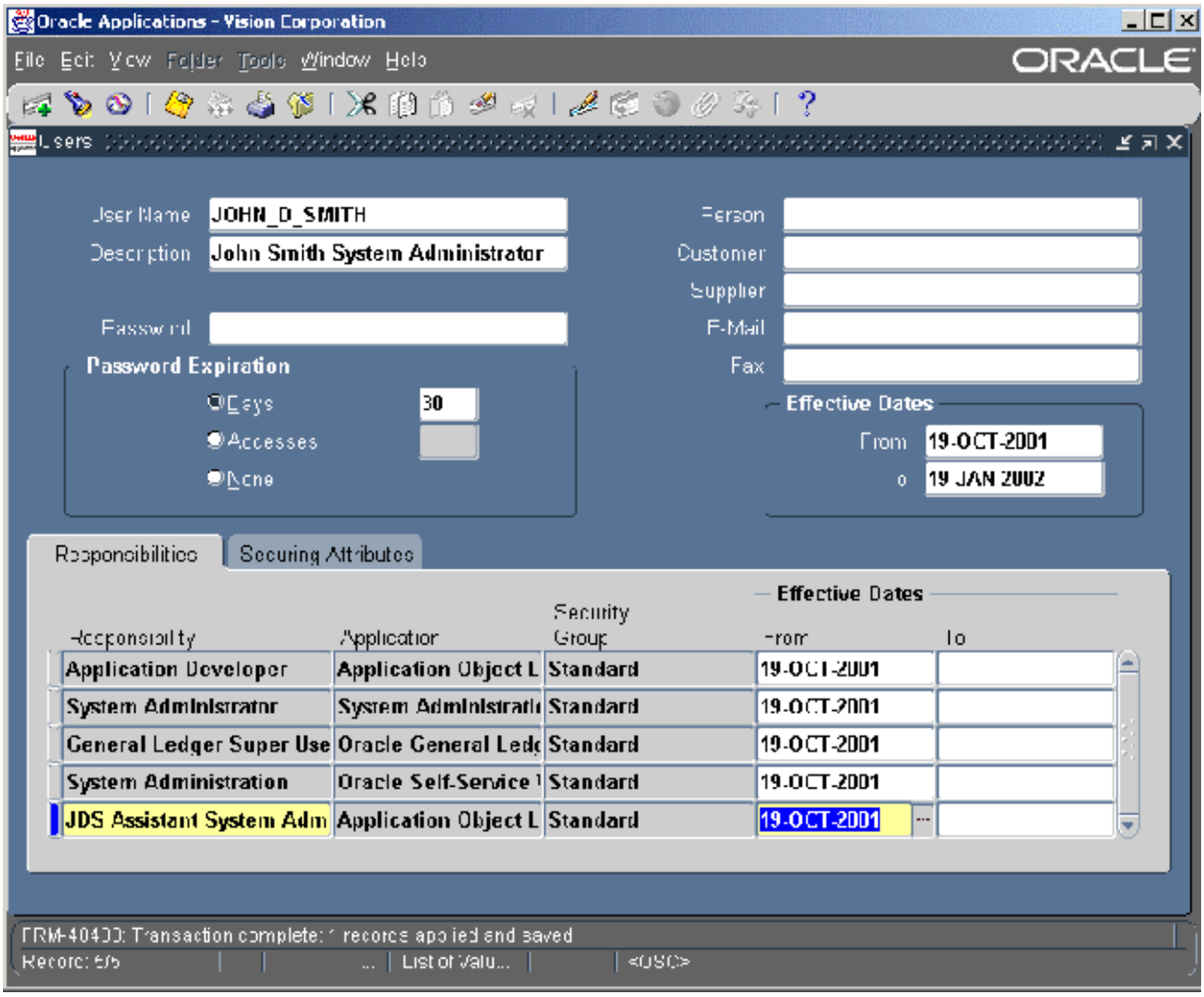
2. Put the form in query mode by pressing F11 or by selecting (M) View > Query by Example > Enter.

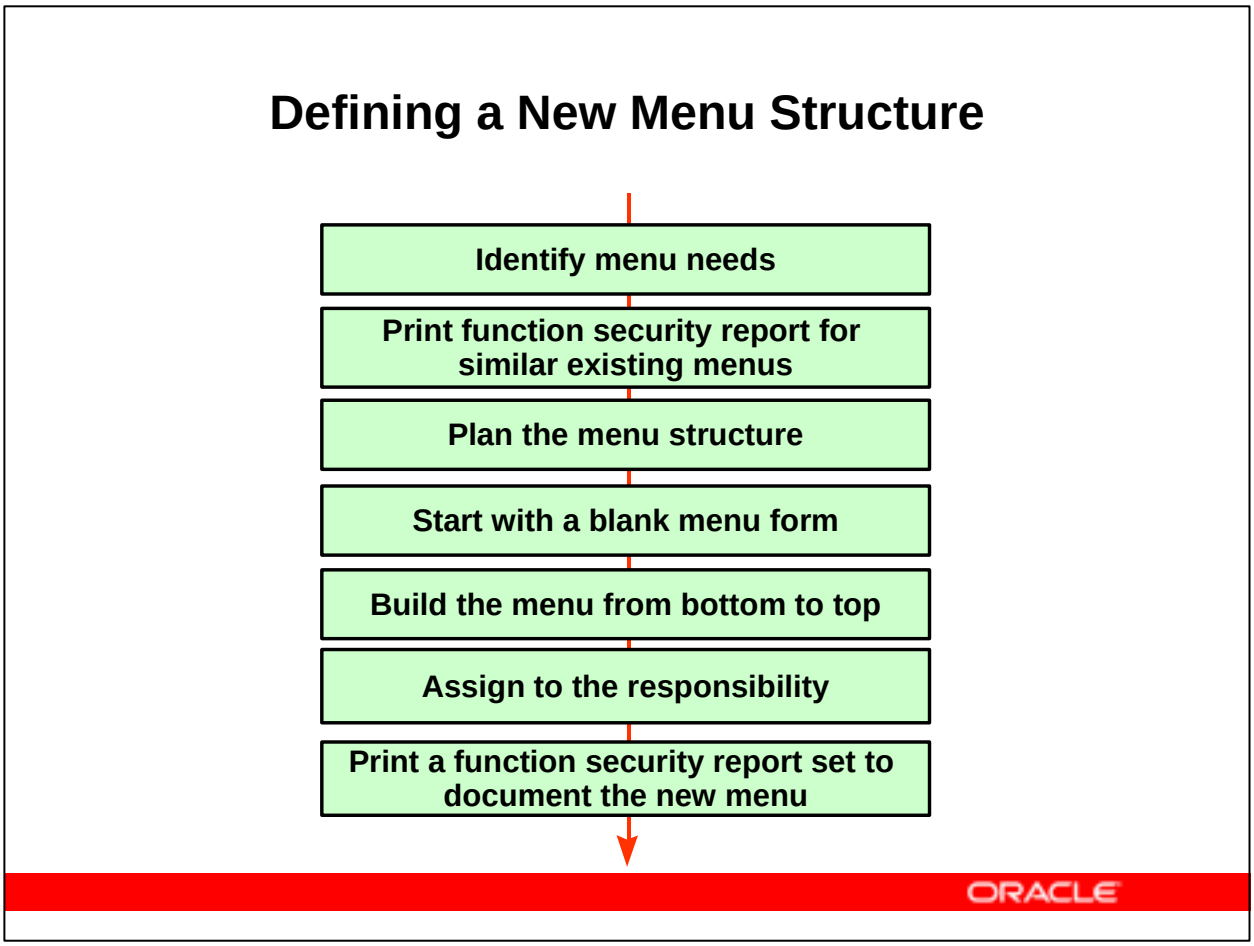
3. Enter the user name you created in the Name field (i.e., *Your Initials* User) and execute your query by pressing Ctrl-F11 or by selecting (M) View > Query by Example > Run.

4. Navigate to the Responsibilities tab.

5. Insert a new record by clicking the New icon on the toolbar, and select *Your Initials* Assistant System Administrator.

6. Save your work.





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.

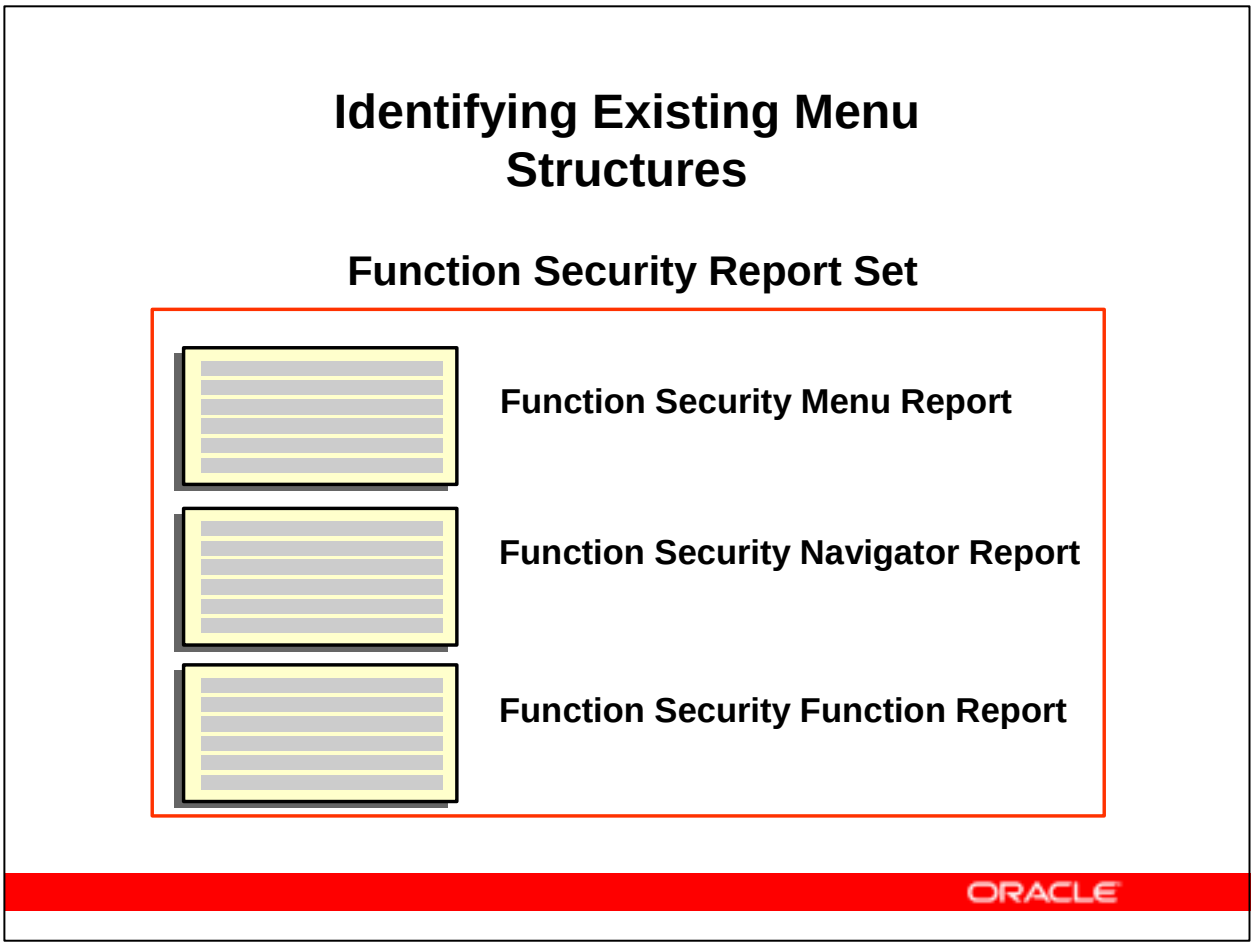
Determine the Application Functionality Required

- 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.

- Assign the menu structure to the new responsibility by using the responsibilities form.
- Document your menu structure by printing the Function Security Menu Report.



Designing a New Menu Structure

Before designing a new menu structure, identify any possible existing menu structures that can be modified or modeled from, and document custom menus by using reports in the Function Security Report Set.

Function Security Menu Report

- Lists the full menu name of the responsibility
- Indicates any excluded menu items, with the rules that exclude them

Function Security Navigator Report

- Lists the menu as it appears in the Navigator for the responsibility specified
- Does not include items excluded by function security rules

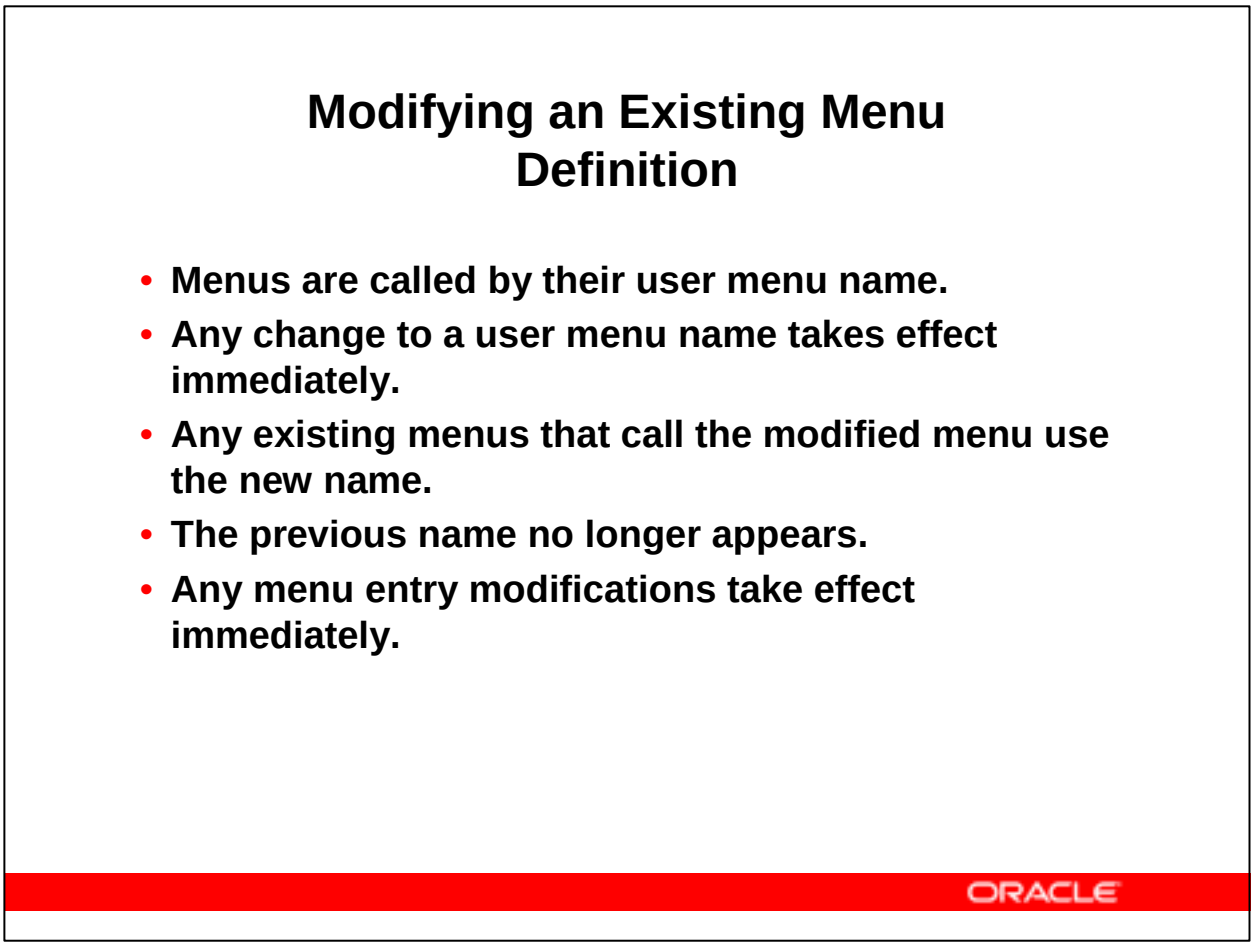
Function Security Function Report

- Lists the functions accessible by the responsibility specified
- Does not include items excluded by function security rules

Menu Guidelines

- Design prompts with unique first letters (typing the first letter will automatically select it).
- Sequence the prompts with the most frequently used functions first.
- Entries cannot be copied from one menu definition to another.
- Use acronyms only when an industry term or company word is so capitalized—for example, WIP or COGS.
- Use integers in numbering.

ORACLE



Modifying a Menu

You can modify predefined menus to suit specific job functions.

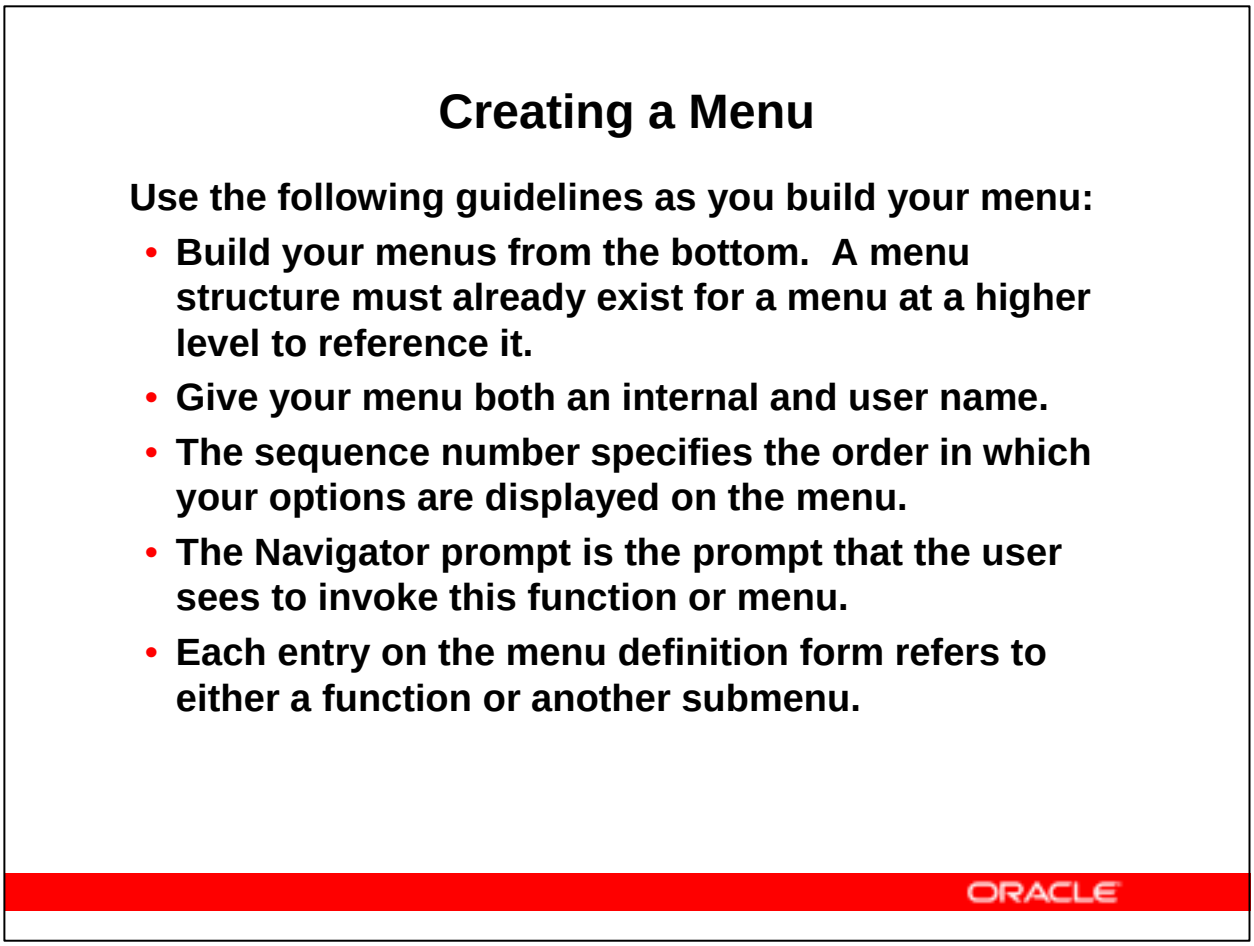
Overwrite a User Menu Name

When you change a menu's user menu name, the menu entries are not affected. The menu's definition still exists, but under a new name. Other menus that call the menu by its old user menu name now call the same menu by its new (revised) user menu name.

The old user menu name is no longer valid. When you are defining menus, or if you are selecting main menus when defining a responsibility, the previously named menu is no longer displayed in any list of values.

Modify a Menu Entry

When you are modifying a predefined menu, all other menus that call that menu display the menu's modifications. For example, if you modify GL_SUPERUSER4.0 by adding another prompt that calls a form, all menus that call GL_SUPERUSER4.0 display the additional prompt when GL_SUPERUSER4.0 is displayed.



Creating a Menu

Use the Menus window to create a menu:

(N) Application > Menu

After you save your changes in this form, the Compile Security concurrent program is automatically submitted to compile the menu data. Compiling your menu data allows for the system to determine more quickly during a user's session whether a function is available to a particular responsibility or menu.

New Menus form fields:

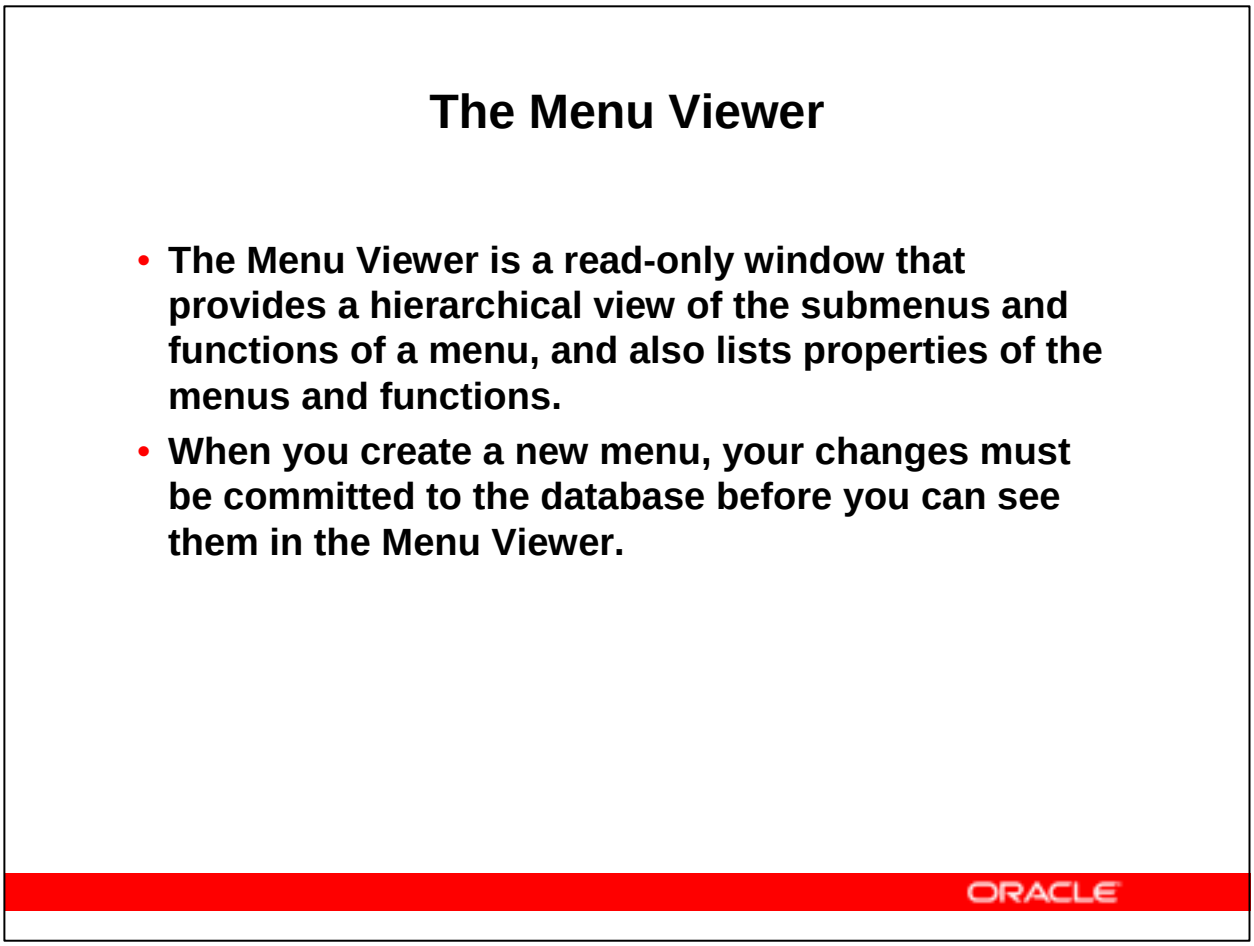
- **Menu Type** – used to specify the purpose of the menu. The possible values are:
  - Standard – for menus that would be used in the Navigator form
  - Tab – for menus used in Self-Service applications
  - Security – for menus that are used to aggregate functions for data security or specific function security purposes, but would not be used in the Navigator form.
- **Grant check box** – checking this box indicates that this function is automatically enabled for the user.

For a complete explanation of the fields on this form see:

(Help) Applied Technology > Oracle Applications System Administration > Function Security > Menus Window

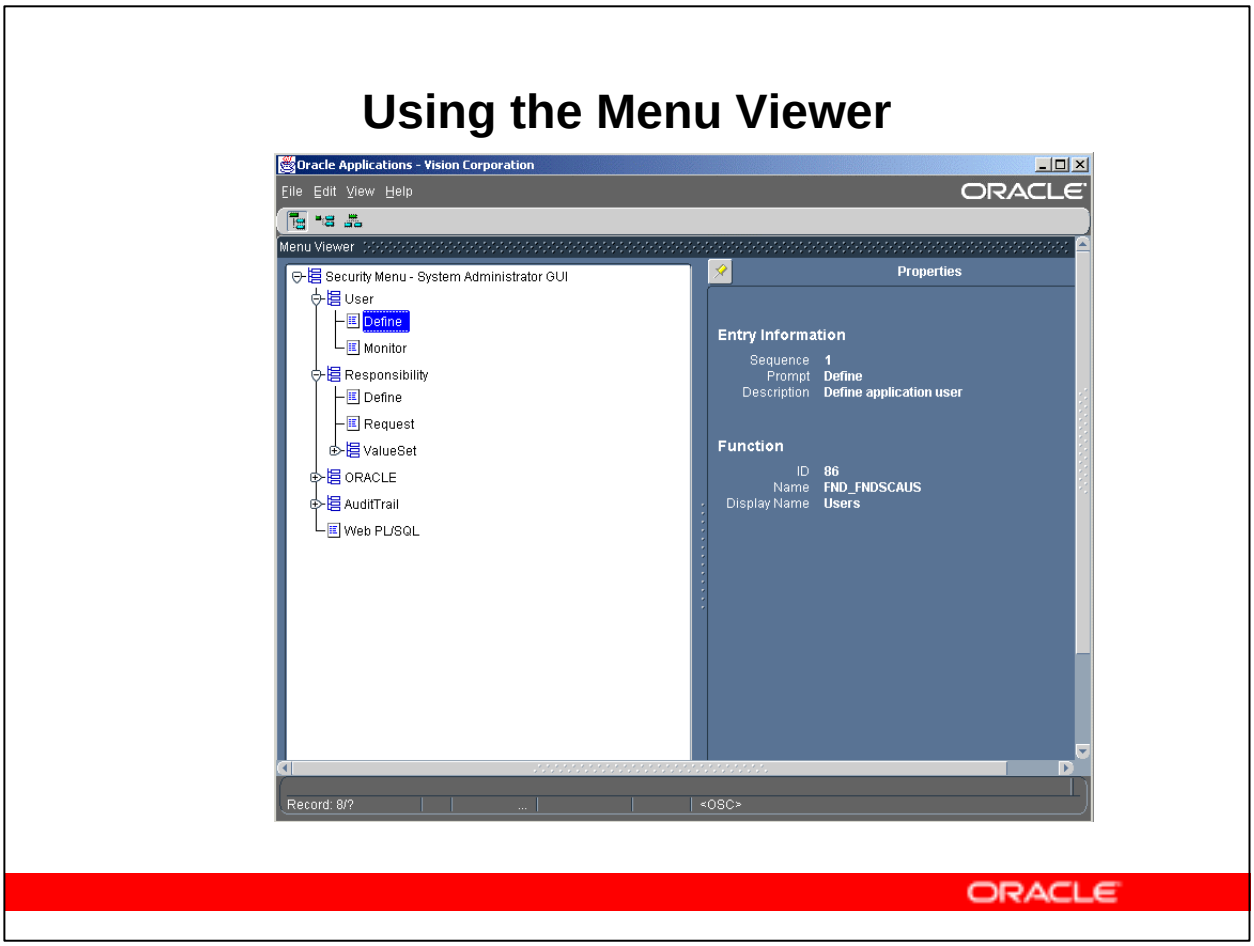
Note: If a menu entry has both a submenu and a function defined for it, then the behavior depends on whether or not the function is executable. It it is executable, then the submenu is treated as content to be rendered by the function. The submenu will not appear on the navigation tree, but will be available in function security tests (FND_FUNCTION.TEST calls). If the function is not executable, then it is treated as a "tag" for enforcing exclusion rules, and the submenu is displayed on the navigation tree.

A function is considered executable if it can be executed directly from the current running user interface.



The Menu Viewer

(N) Application > Menu > (B) View Tree...



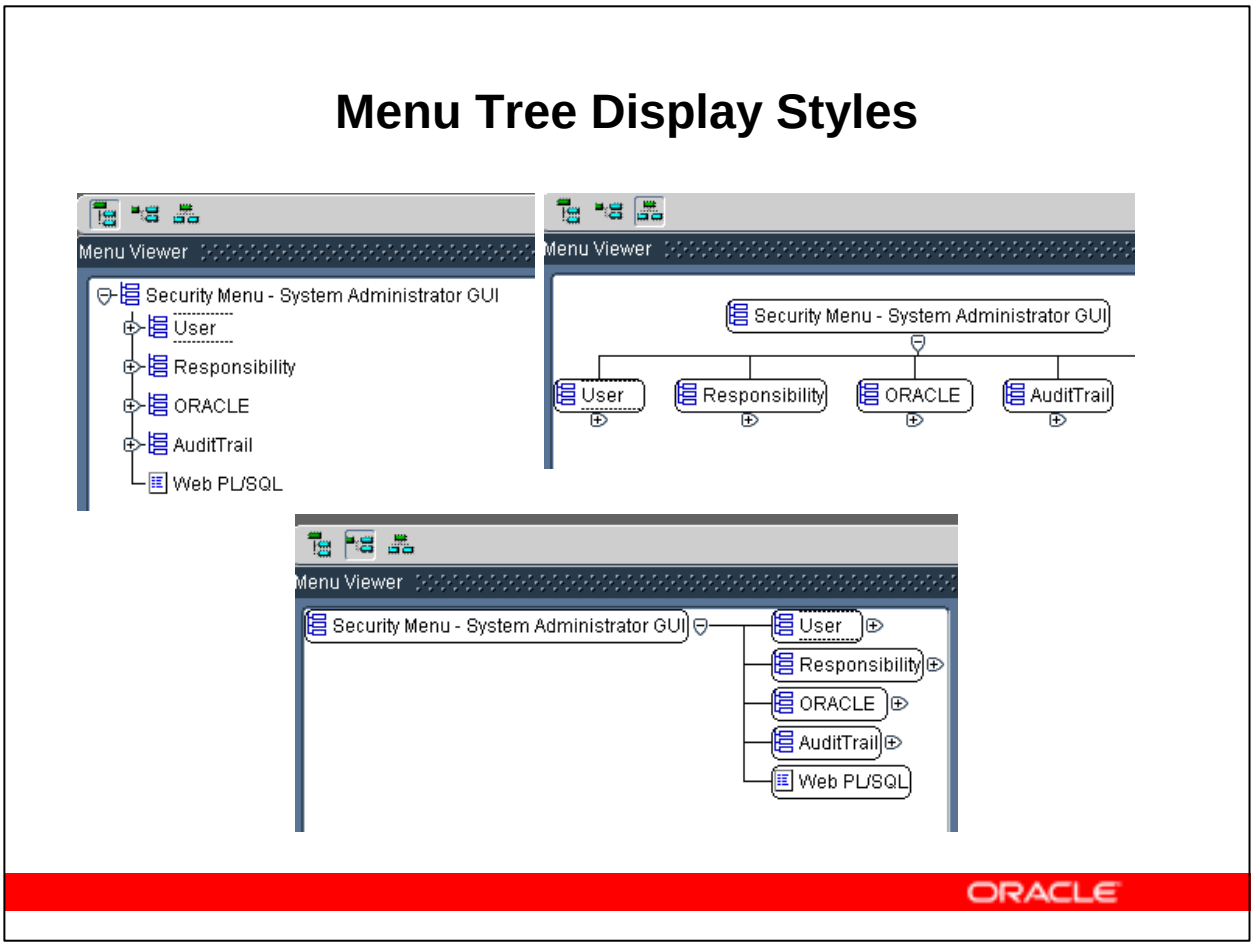
Using the Menu Viewer

(N) Application > Menu > (B) View Tree...

- To view the menu tree, click on the plus (+) sign next to the menu name. You will see a hierarchical tree with a number of nodes. Each node represents a function or submenu of your main menu.
- The menu tree displays the user menu name for the main menu, and displays the prompts from the Menus form for submenus and functions. If no prompt has been specified, then no label will appear for the node.

Printing the Menu Tree

To print a menu tree, choose Print from the File menu.



Menu Tree Display Styles

(N) Application > Menu (B) View Tree…

There are three styles for viewing your menu tree. You can select the style from the View menu or by clicking the appropriate toolbar icon.

- Vertical – Menu entries are displayed vertically, similar to how they appear in the Navigator window when you log on to Oracle Applications.
- Org-Chart – Menu entries are displayed horizontally as in an organizational chart.
- Interleaved – Menu entries are displayed horizontally and vertically.

Viewing Node Properties

Use one of the following methods to view a node's sequence number, prompt, and description:

- **Highlight the node in the menu tree to view the properties in the properties pane.**
- **Create a separate Properties window by clicking the "push pin" button at the top of the Properties pane.**
- **Select Properties from the View menu.**

ORACLE

Practice - Menus

Overview

The committee wants you to test the Menu processes. Specifically, they want you to create a System Auditor responsibility, and to create a custom menu to assign to that responsibility. It this testing process you will implement the following.

- Create a custom menu with several functions and submenus
- View your custom menu using the Menu Viewer
- Create a System Auditor responsibility, and associate your new menu to that responsibility
- Assign the System Auditor to a new user and test

Tasks

Create your Level 3 Menu

Menus in Oracle Applications must be created bottom-up. Your first task is to create your lowest level menu.

1. Create a new menu. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Menu Name**: *Your Initials*_FLEX_SECURITY (e.g., WHS_FLEX_SECURITY)
   - **User Menu Name**: *Your Initials*_FLEX_SECURITY (e.g., WHS_FLEX_SECURITY)
   - **Description**: Define and Assign Flexfield Security Rules
   - **Seq**: 10, **Prompt**: Define, **Function**: Flexfield Security Rules, **Description**: Define Flex Security Rule, **Menu Type**: Standard
   - **Seq**: 20, **Prompt**: Assign, **Function**: Assign Flexfield Security, **Description**: Assign Flex Security Rule, **Menu Type**: Standard

Create your Level 2 Menu

2. Create a new menu. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Menu Name**: *Your Initials*_RESPONSIBILITY (e.g., WHS_RESPONSIBILITY)
   - **User Menu Name**: *Your Initials*_RESPONSIBILITY (e.g., WHS_RESPONSIBILITY)

- **Description**: Define Responsibilities, Request Groups, and Security Rules

- **Seq**: 10, **Prompt**: Define Responsibilities, **Function**: Responsibilities, **Description**: Define Responsibilities, **Menu Type**: Standard

- **Seq**: 20, **Prompt**: Request Group, **Function**: Request Groups, **Description**: Define Request Groups, **Menu Type**: Standard

- **Seq**: 30, **Prompt**: Flexfield Security, **Submenu**: *Your Initials*_FLEX_SECURITY (i.e., Your Level 3 menu), **Description**: Define and Assign Flex Security Rules, **Menu Type**: Standard

Create your Level 1 Menu

3. Create a new menu. Remember to put your initials at the front of your data entries to keep your data unique.

    - **Menu Name**: *Your Initials*_TOP_SYSTEM_AUDITOR

    - **User Menu Name**: *Your Initials*_TOP_SYSTEM_AUDITOR

    - **Description**: Custom System Auditor Menu

    - **Seq**: 10, **Prompt**: Users, **Submenu**: User Menu - System Administrator GUI, **Description**: Define and Monitor Users, **Menu Type**: Standard

    - **Seq**: 20, **Prompt**: Responsibilities, **Submenu**: *Your Initials*_RESPONSIBILITY (i.e., Your Level 2 Menu), **Description**: Define Responsibility, Request Group, Security Rules, **Menu Type**: Standard

    - **Seq**: 30, **Prompt**: Personal, **Function**: Profile User Values, **Description**: View/Update Personal Profile Options, **Menu Type**: Standard

    - **Seq**: 40, **Prompt**: Requests, **Submenu**: Requests Menu - Other Responsibilities, **Description**: Run and View Requests, Define Request Sets, **Menu Type**: Standard

4. View your new menu in the Menu Viewer

Create your System Auditor Responsibility

5. Create a new responsibility. Remember to put your initials at the front of your data entries to keep your data unique.

    - **Responsibility Name**: *Your Initials* System Auditor

    - **Application**: Application Object Library

    - **Responsibility Key**: *Your Initials*_SYSTEM_AUDITOR (e.g., WHS_SYSTEM_AUDITOR)

- **Description**: *Your Initials* System Auditor
- **Effective Date**: From: Today
- **Effective Date**: To: (leave blank)
- **Available From**: Oracle Applications (default)
- **Data Group Name**: Standard
- **Data Group Application**: Application Object Library
- **Menu**: *Your Initials*_TOP_SYSTEM_AUDITOR (i.e.,Your Level 1 Menu)
- **Request Group**: (leave blank)

Create your System Auditor User

6. Create a new user. Remember to put your initials at the front of your data entries to keep your data unique.
   - **User Name**: *Your Initials* System Auditor
   - **Description**: System Auditor
   - **Password**: WELCOME
   - **Responsibility**: *Your Initials* System Auditor
7. Test the new user, new responsibility, and new menu to ensure that they are working properly.

Solution - Menus

Create your Level 3 Menu

Responsibility = System Administrator

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Application > Menu.

3. Enter the appropriate data for your Level 3 menu.

   - **Menu Name**: *Your Initials*_FLEX_SECURITY (e.g., WHS_FLEX_SECURITY)

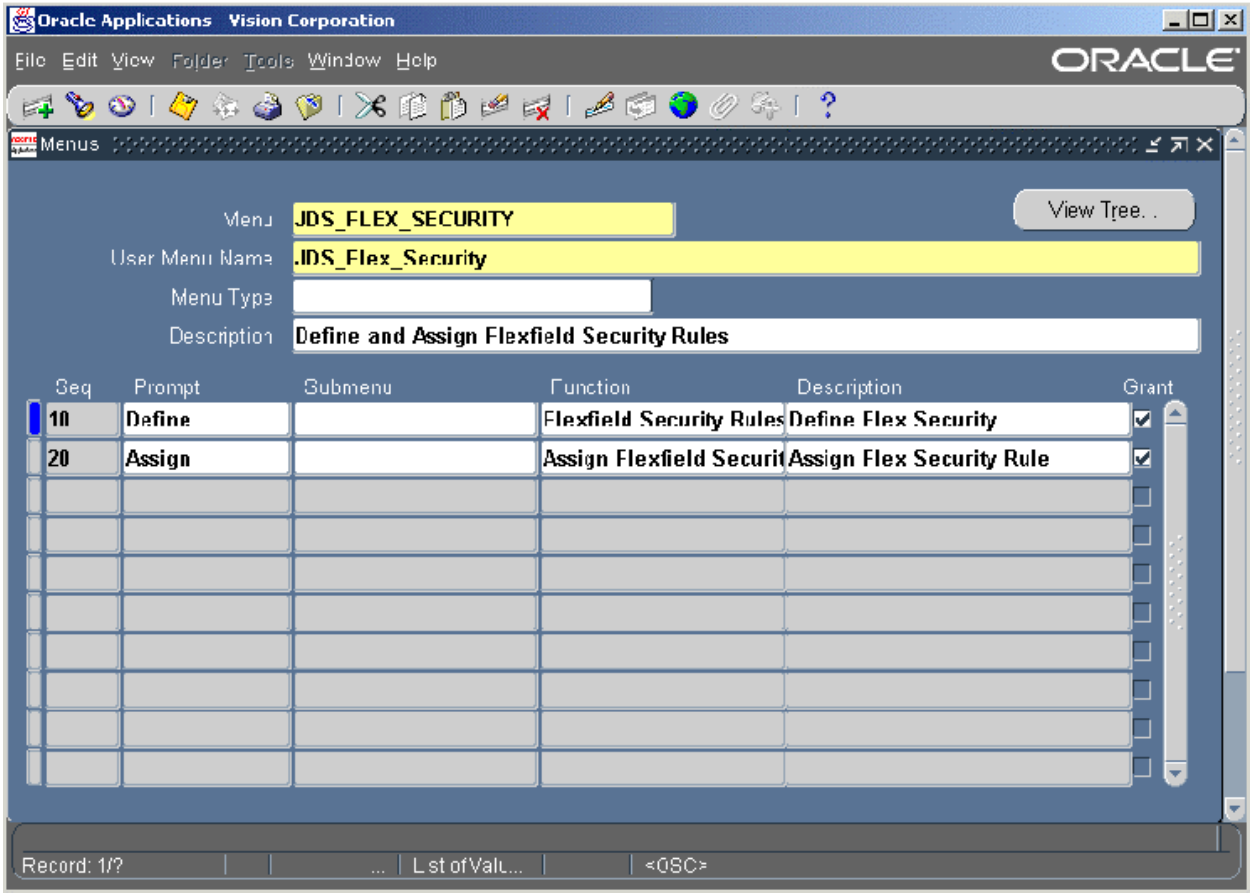
   - **User Menu Name**: *Your Initials*_FLEX_SECURITY (e.g., WHS_FLEX_SECURITY)

   - **Description**: Define and Assign Flexfield Security Rules

   - **Seq**: 10, **Prompt**: Define, **Function**: Flexfield Security Rules, **Description**: Define Flex Security Rule, **Menu Type**: Standard

   - **Seq**: 20, **Prompt**: Assign, **Function**: Assign Flexfield Security, **Description**: Assign Flex Security Rule, **Menu Type**: Standard

4. Save your work. Your work should look similar to the following.



Create your Level 2 Menu

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

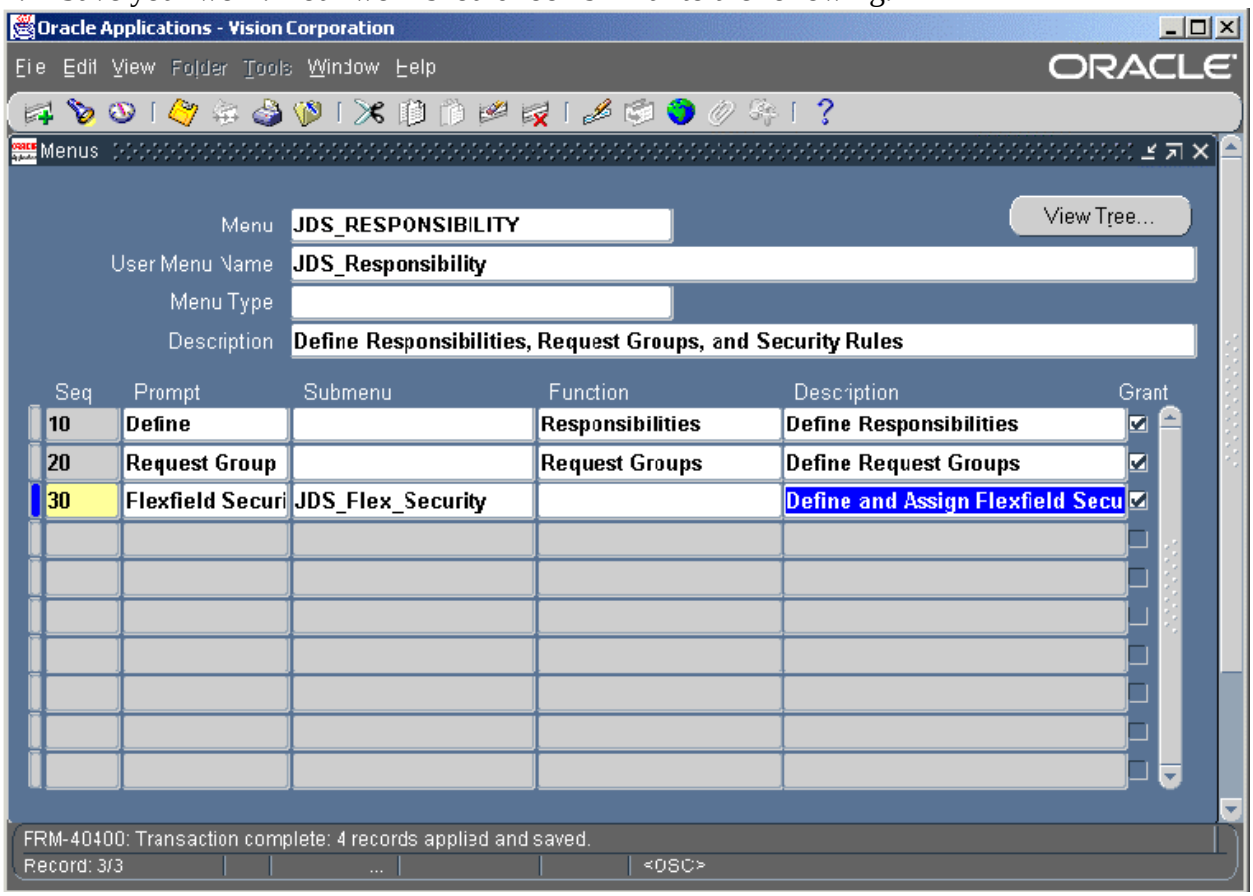
2. Navigate to (N) Application > Menu.

3. Enter the appropriate data for your Level 2 menu.

   - **Menu Name**: *Your Initials*_RESPONSIBILITY (e.g., WHS_RESPONSIBILITY)
   - **User Menu Name**: *Your Initials*_RESPONSIBILITY (e.g., WHS_RESPONSIBILITY)
   - **Description**: Define Responsibilities, Request Groups, and Security Rules
   - **Seq**: 10, **Prompt**: Define Responsibilities, **Function**: Responsibilities, **Description**: Define Responsibilities, **Menu Type**: Standard
   - **Seq**: 20, **Prompt**: Request Group, **Function**: Request Groups, **Description**: Define Request Groups, **Menu Type**: Standard

- **Seq**: 30, **Prompt**: Flexfield Security, **Submenu**: *Your Initials*_FLEX_SECURITY (i.e., Your Level 3 menu), **Description**: Define and Assign Flex Security Rules, **Menu Type**: Standard

4. Save your work. Your work should look similar to the following.



Create your Level 1 Menu

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Application > Menu.

3. Enter the appropriate data for your Level 1 menu.

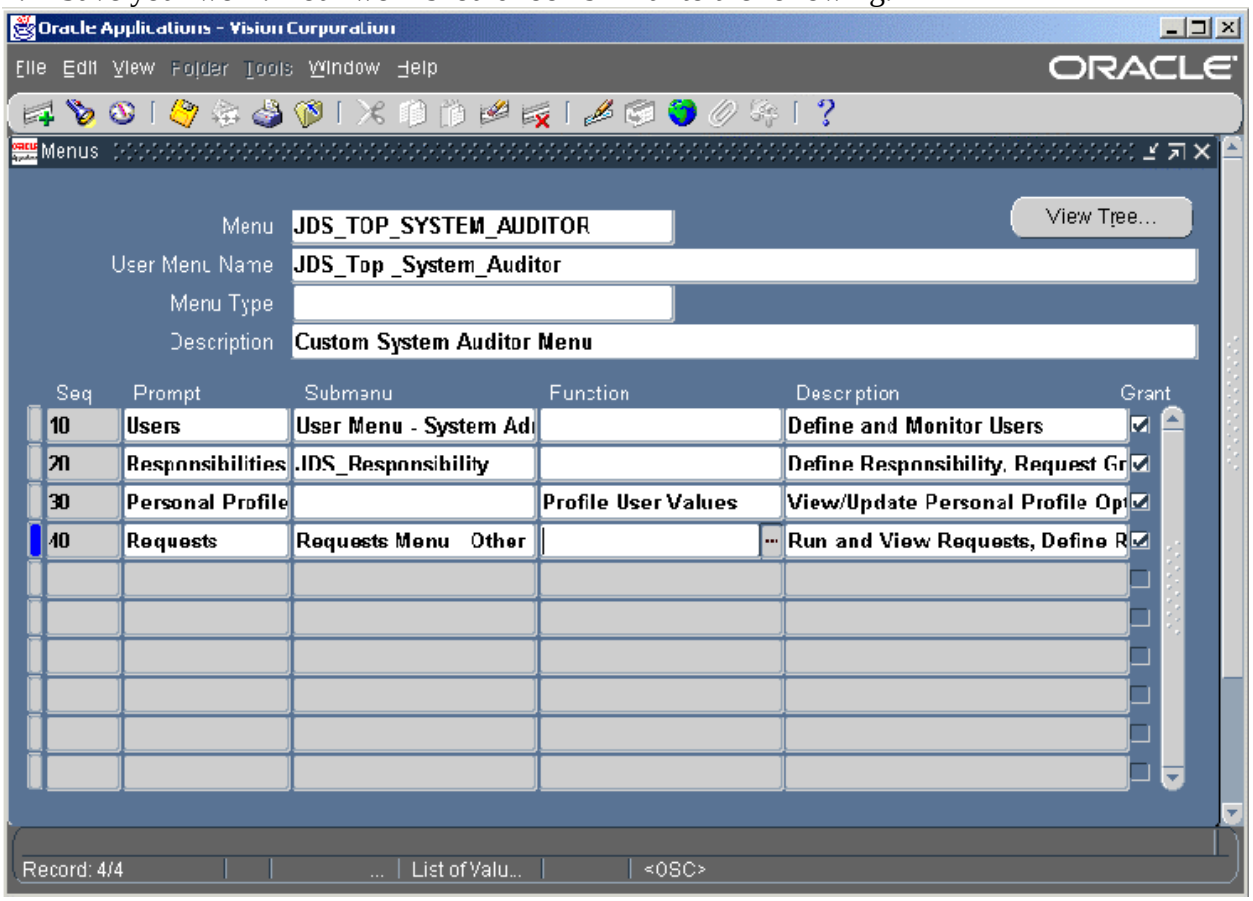
   - **Menu Name**: *Your Initials*_TOP_SYSTEM_AUDITOR
   - **User Menu Name**: *Your Initials*_TOP_SYSTEM_AUDITOR
   - **Description**: Custom System Auditor Menu
   - **Seq**: 10, **Prompt**: Users, **Submenu**: User Menu - System Administrator GUI, **Description**: Define and Monitor Users, **Menu Type**: Standard

- **Seq**: 20, **Prompt**: Responsibilities, **Submenu**: *Your Initials*_RESPONSIBILITY (i.e., Your Level 2 Menu), **Description**: Define Responsibility, Request Group, Security Rules, **Menu Type**: Standard

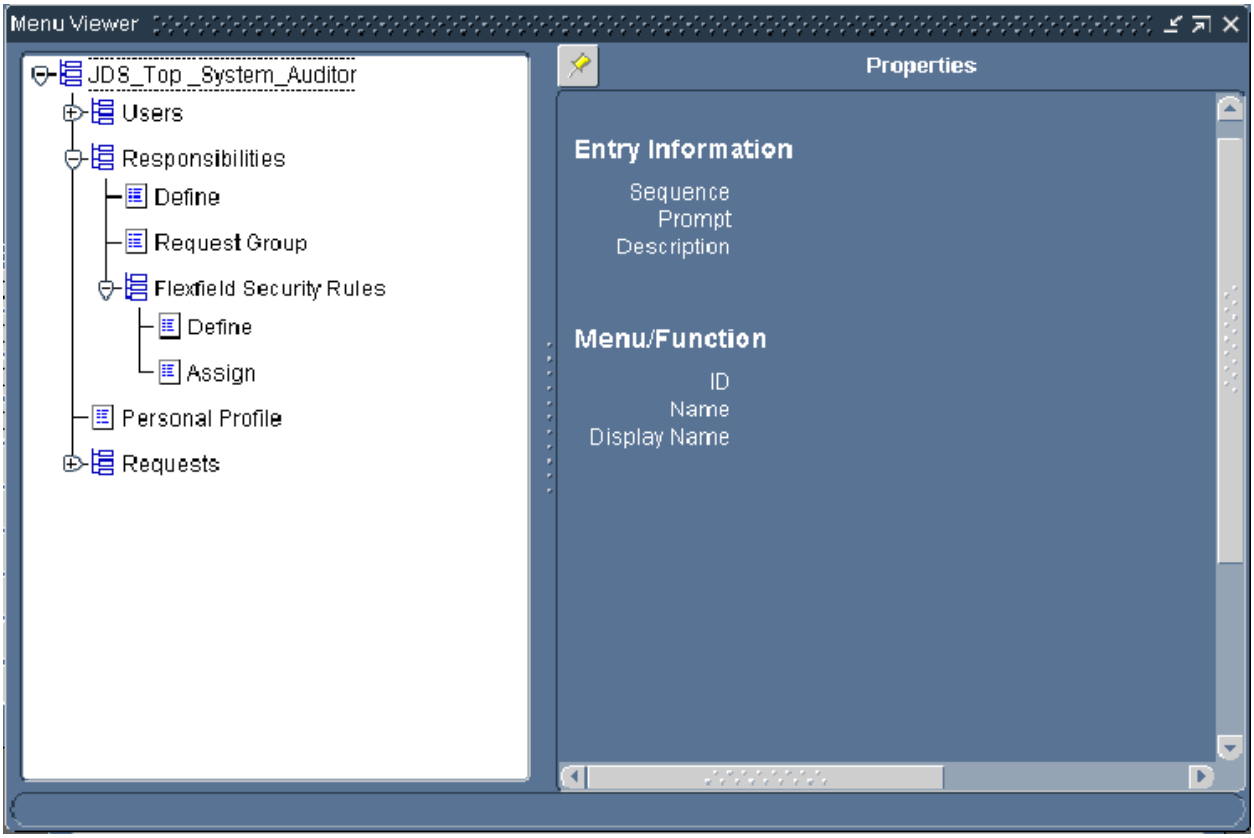
- **Seq**: 30, **Prompt**: Personal, **Function**: Profile User Values, **Description**: View/Update Personal Profile Options, **Menu Type**: Standard

- **Seq**: 40, **Prompt**: Requests, **Submenu**: Requests Menu - Other Responsibilities, **Description**: Run and View Requests, Define Request Sets, **Menu Type**: Standard

4. Save your work. Your work should look similar to the following.



5. Click the "**View Tree…**" button to see your new menu in the Menu Viewer. Your menu tree should appear similar to the example shown in the slide.



Create your System Auditor Responsibility

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

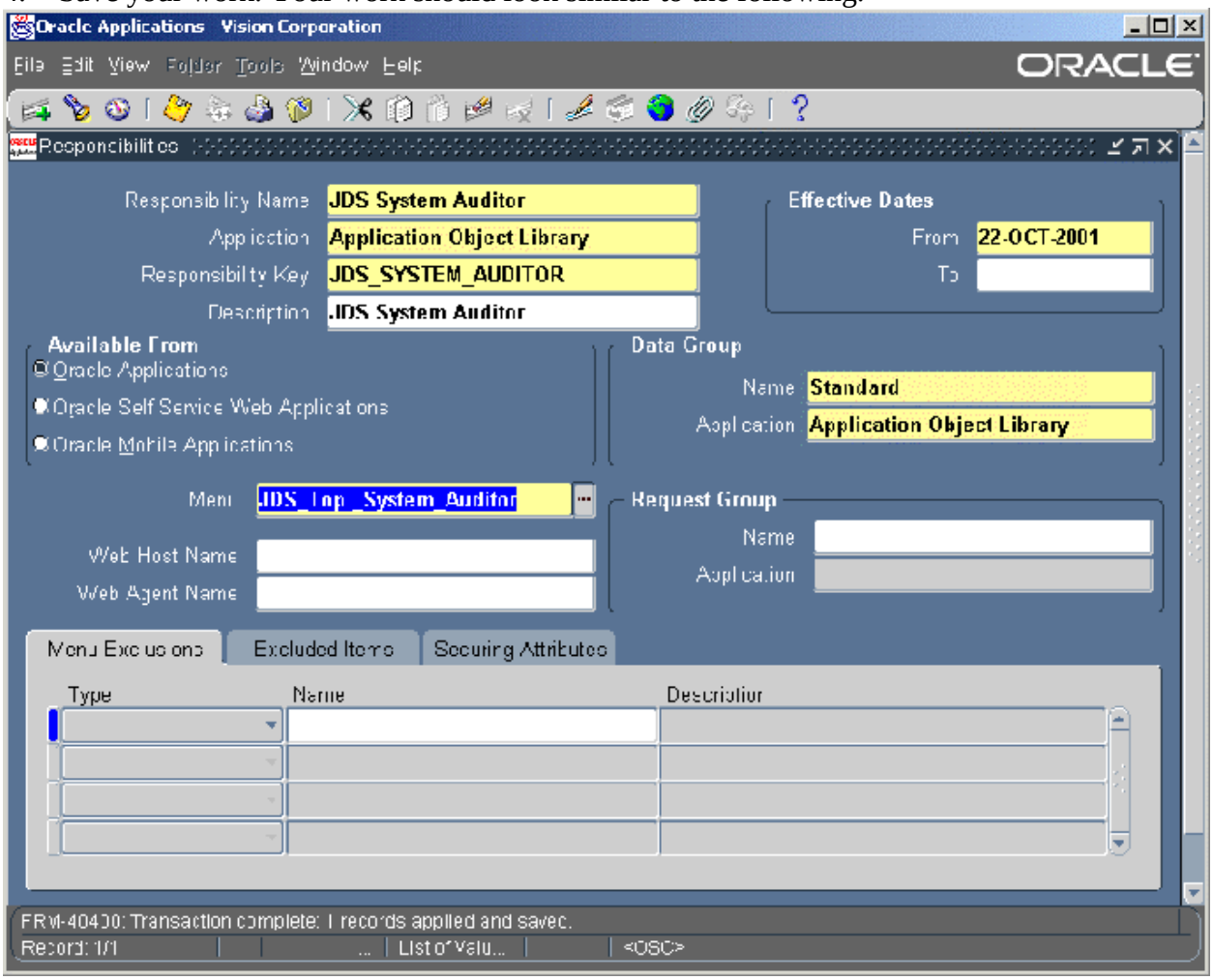
2. Navigate to (N) Security > Responsibility > Define

3. Enter the appropriate data for your responsibility.

   - **Responsibility Name**: *Your Initials* System Auditor
   - **Application**: Application Object Library
   - **Responsibility Key**: *Your Initials*_SYSTEM_AUDITOR (e.g., WHS_SYSTEM_AUDITOR)
   - **Description**: *Your Initials* System Auditor
   - **Effective Date**: From: Today
   - **Effective Date**: To: (leave blank)
   - **Available From**: Oracle Applications (default)
   - **Data Group Name**: Standard

- **Data Group Application**: Application Object Library
- **Menu**: *Your Initials*_TOP_SYSTEM_AUDITOR (i.e.,Your Level 1 Menu)
- **Request Group**: (leave blank)

4. Save your work. Your work should look similar to the following.

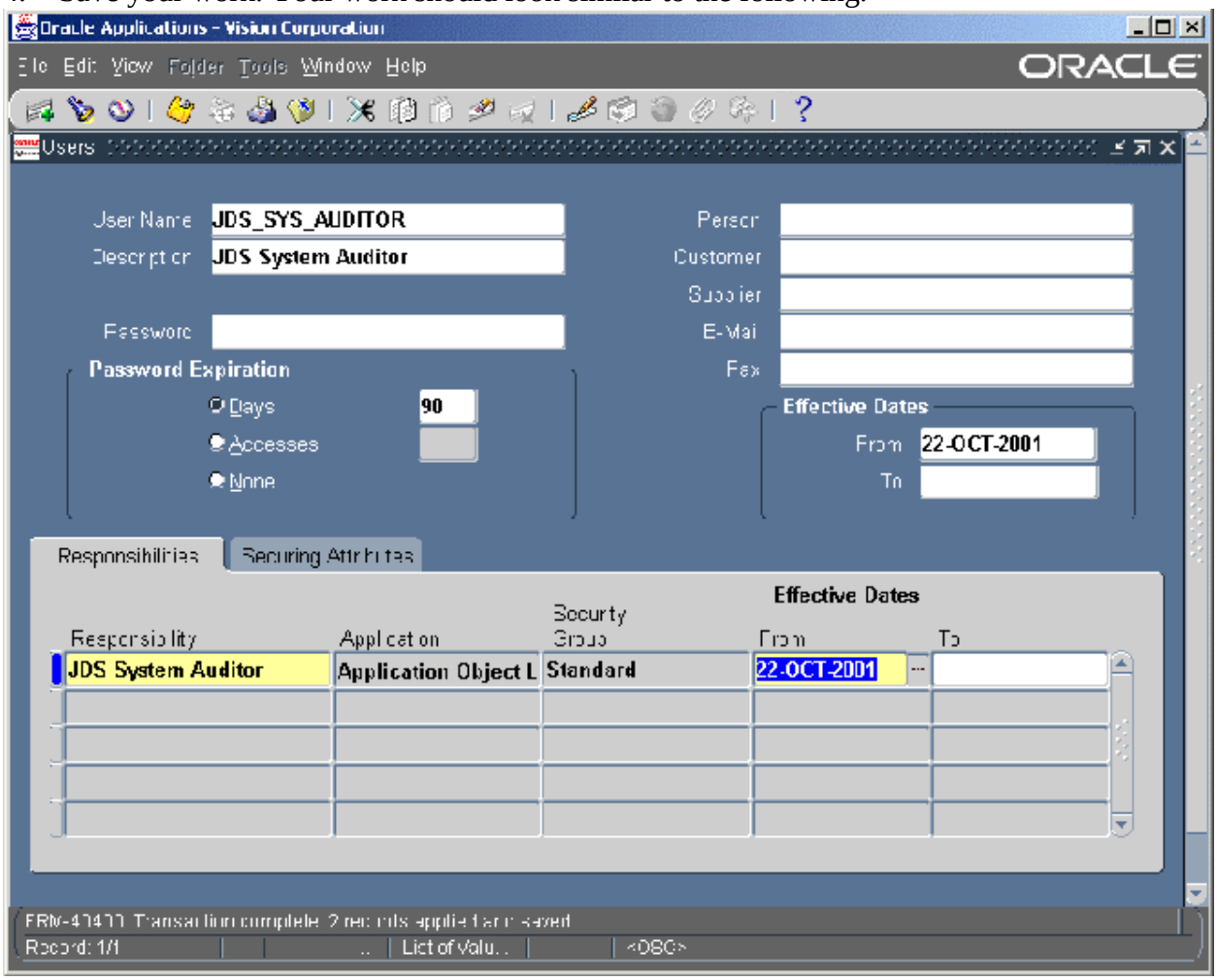


Create your System Auditor User

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.
2. Navigate to (N) Security > User > Define
3. Enter the appropriate data for your user.
   - **User Name**: *Your Initials* System Auditor
   - **Description**: System Auditor

- **Password**: WELCOME
- **Responsibility**: *Your Initials* System Auditor

4. Save your work. Your work should look similar to the following.



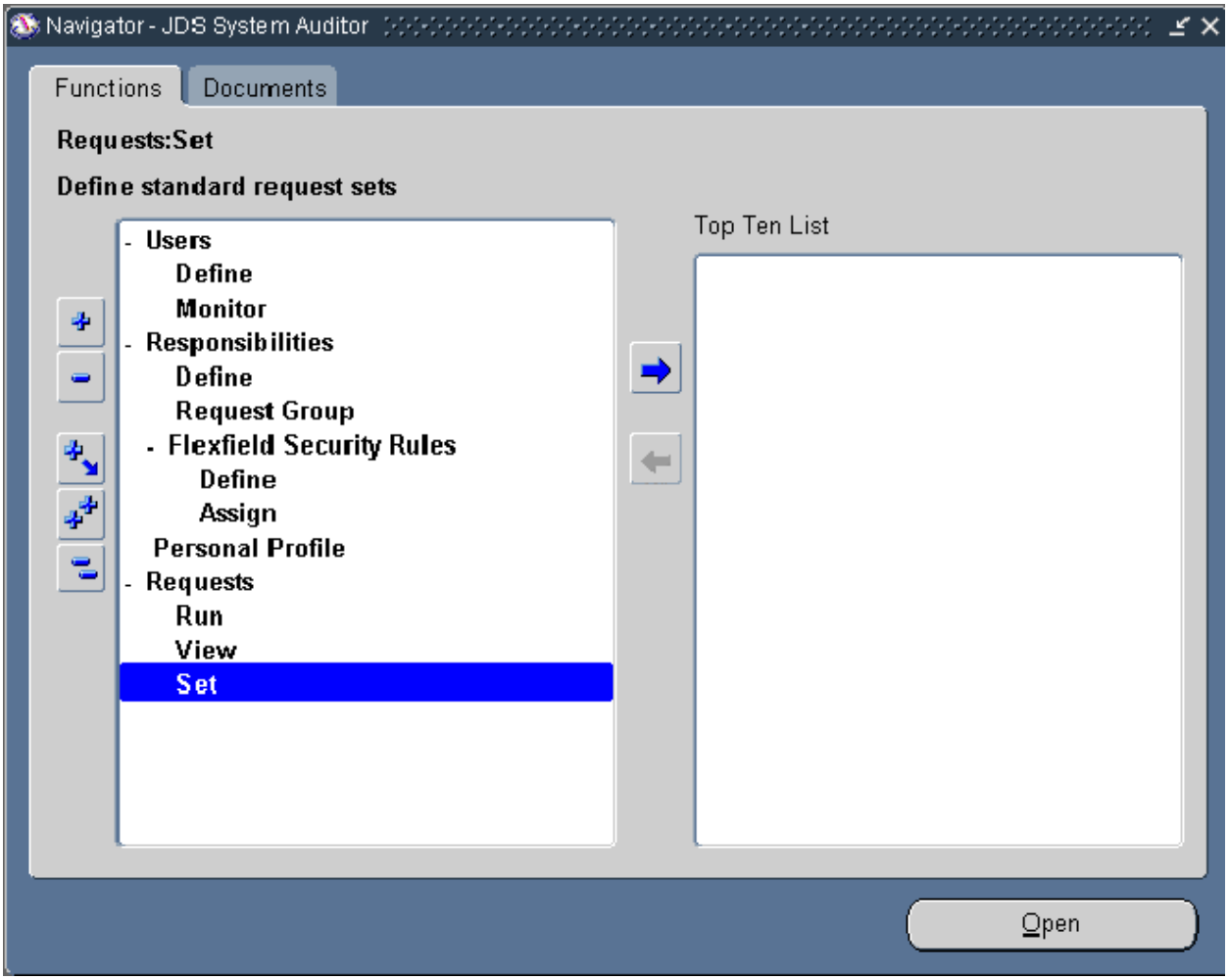
5. Test the new user, new responsibility, and new menu to ensure that they are working properly. Exit Oracle Applications and return to your Personal Homepage.

6. Click on the Exit icon to open the login screen.

7. Sign on again using your new login. You will be required to change your password. Note your user name and password for future access.

8. From the Personal Homepage click on *Your Initials* System Auditor responsibility.

9. Verify that your Navigator menu contains the correct items. Your screen should appear similar to the example shown in the slide.



Practice - Query-Only Forms

Overview

The committee has evaluated the System Auditor responsibility you created. The responsibility has access to its own Personal Profile Options. In this test process you will do the following:

- Create a query-only version of the System Profile Options function
- Add this query-only function to your existing System Auditor menu

Tasks

Create your Query-Only Function

1. Create a new Form Function. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Function**: *Your Initials*_FNDPOMPV (e.g., WHS_FNDPOMPV)
   - **User Function Name**: *Your Initials*_FNDPOMPV (e.g., WHS_FNDPOMPV)
   - **Type**: Form
   - **Description**: View System Profile Values
   - **Function**: *Your Initials*_FNDPOMPV (e.g., WHS_FNDPOMPV)
   - **Form**: Update System Profile Values
   - **Application**: Application Object Library
   - **Parameters**: QUERY_ONLY=YES

Assign your Query-Only Function to your Existing System Auditor Menu

1. Query the *Your Initials*_TOP_SYSTEM_AUDITOR menu in the Menu form. Enter the following for your new function.
   - **Sequence**: 35
   - **Prompt**: View System Profiles
   - **Function**: View System Profiles Values
2. Log in as your System Auditor and test your query-only form.

Solution - Query-Only Forms

Create your Query-Only Function

Responsibility = System Administrator

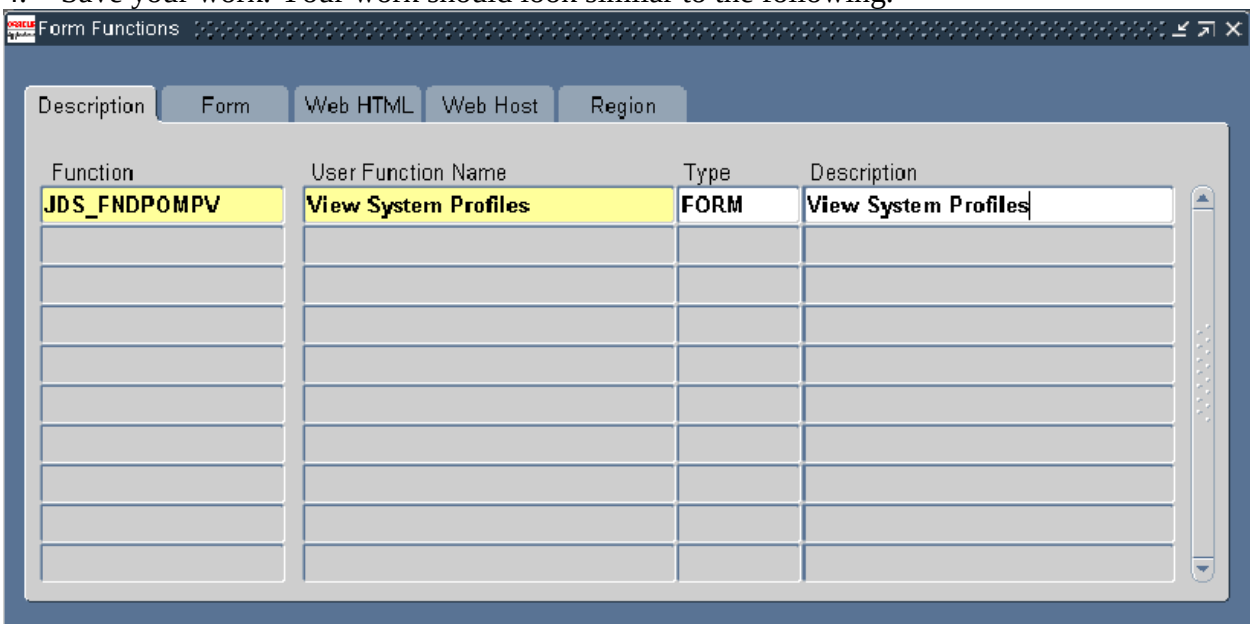
1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

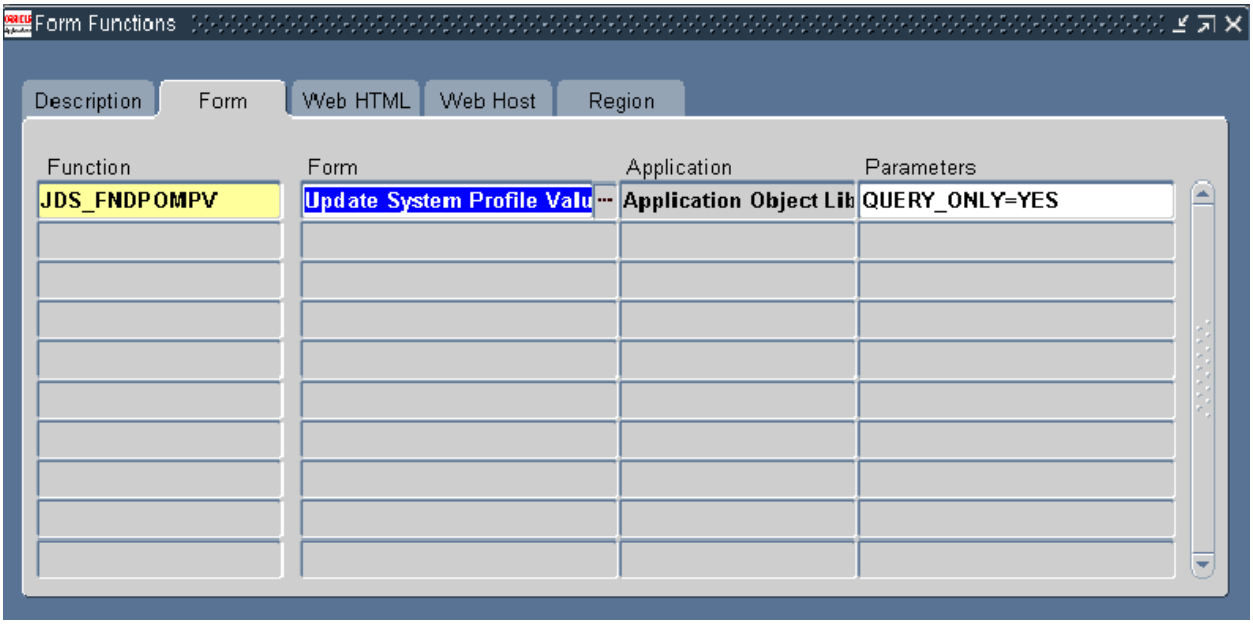
2. Navigate to (N) Application > Function.

3. Enter the appropriate data for your new Form Function.

   - **Function**: *Your Initials*_FNDPOMPV (e.g., WHS_FNDPOMPV)
   - **User Function Name**: *Your Initials*_FNDPOMPV (e.g., WHS_FNDPOMPV)
   - **Type**: Form
   - **Description**: View System Profile Values
   - **Function**: *Your Initials*_FNDPOMPV (e.g., WHS_FNDPOMPV)
   - **Form**: Update System Profile Values
   - **Application**: Application Object Library
   - **Parameters**: QUERY_ONLY=YES

4. Save your work. Your work should look similar to the following.





Assign your Query-Only Function to your Existing System Auditor Menu

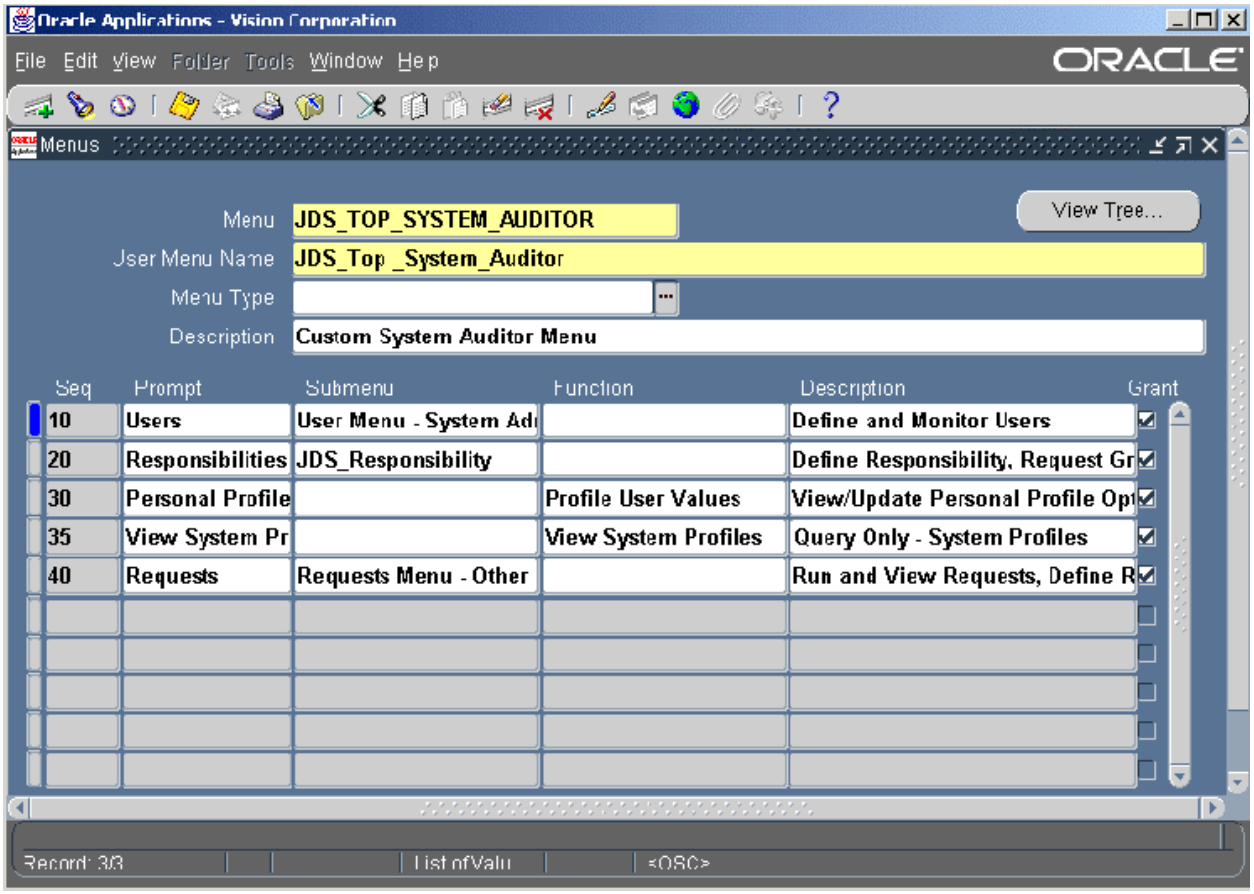
1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Application > Menu.

3. Query the *Your Initials*_TOP_SYSTEM_AUDITOR menu in the Menu form. Enter the following for your new function.

   - **Sequence**: 35
   - **Prompt**: View System Profiles
   - **Function**: View System Profiles Values

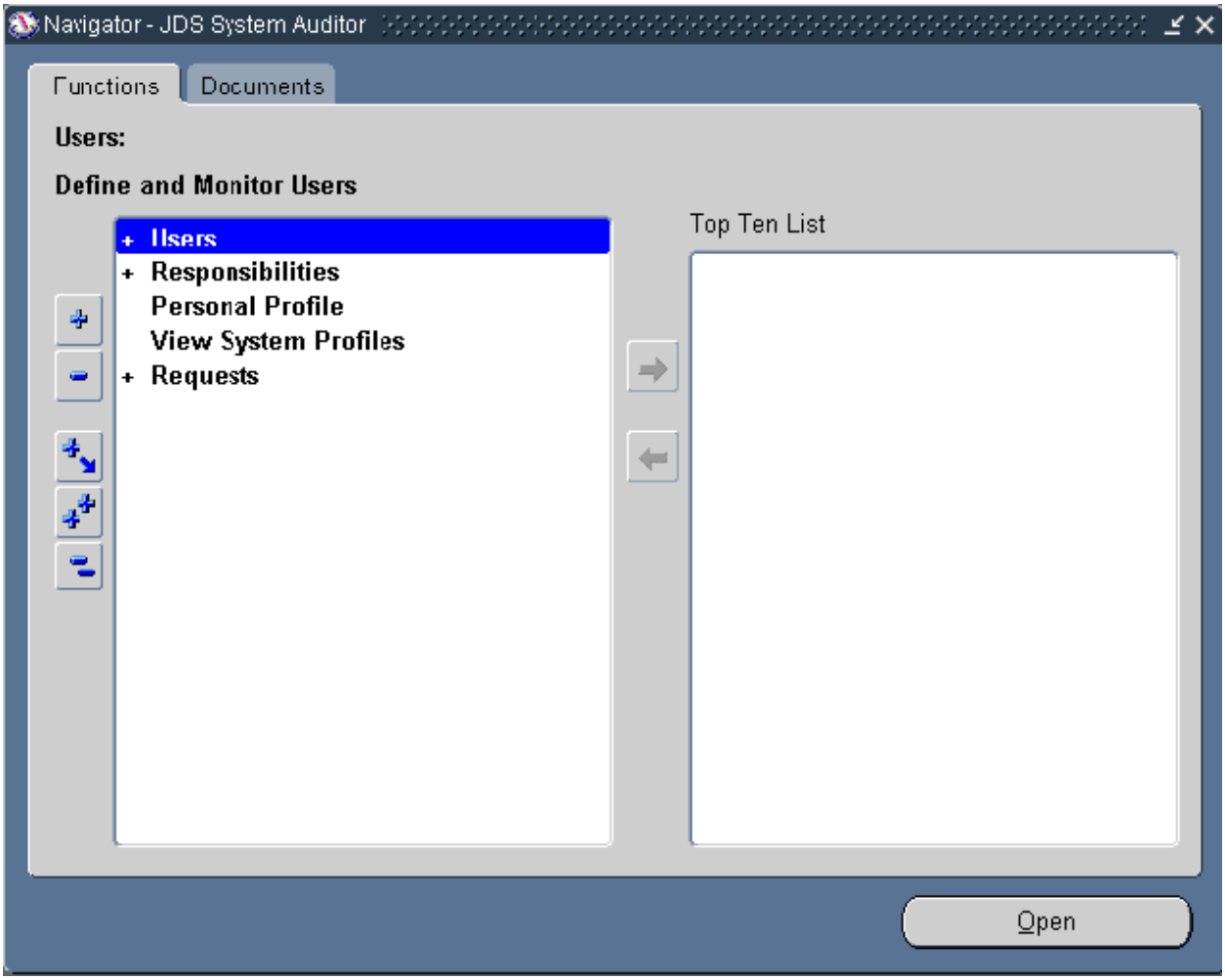
4. Save your work. Your work should look similar to the following.



5. Exit Oracle Applications and the Personal Homepage.

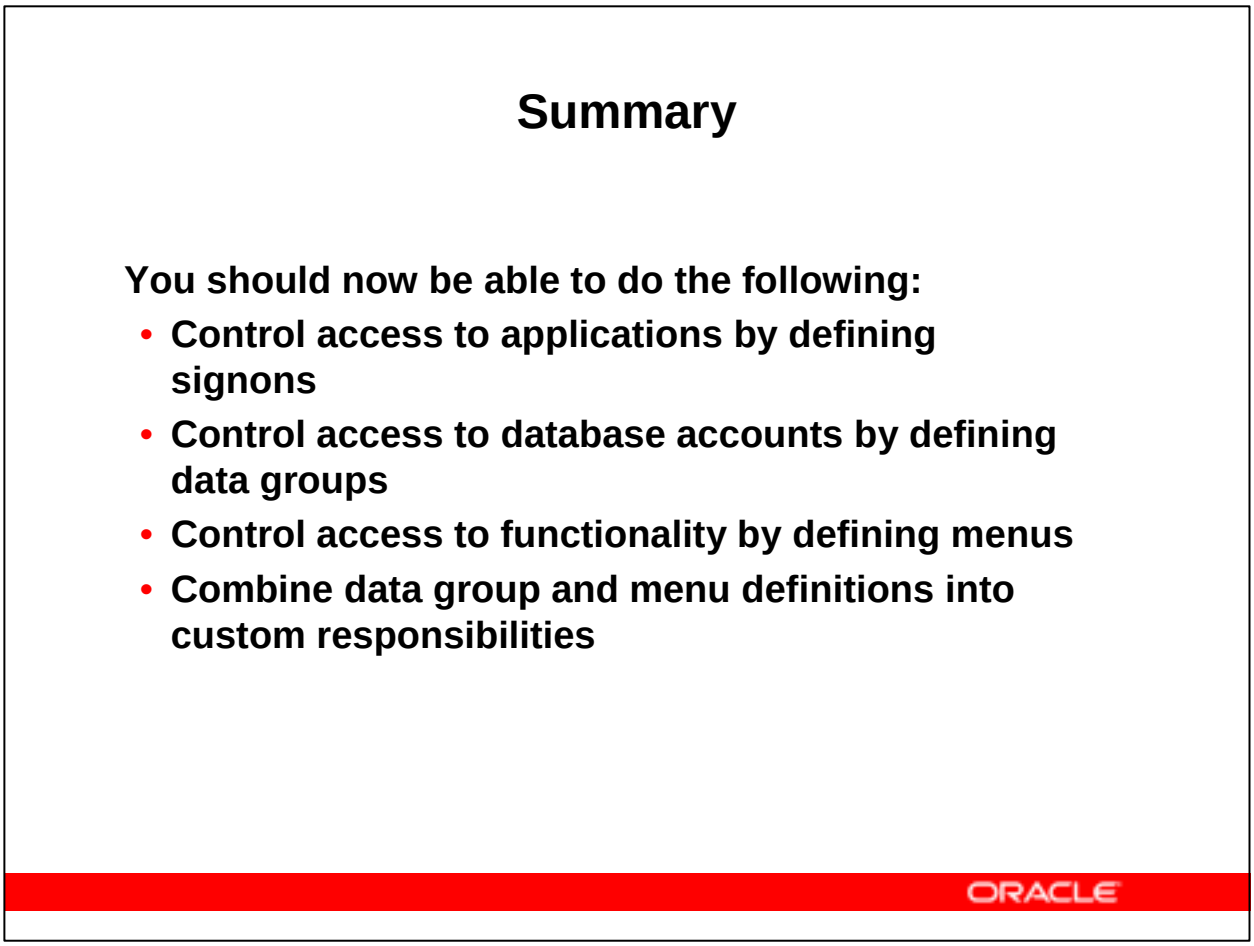
6. Log in as *Your Initials* System Auditor.

7. Your menu should be similar to the example shown in the slide.



8. Test your query-only form.

   For example, open View System Profiles and in the Find System Profile Values window enter “FND%” in the Profile field and click Find. This will display all the Site-level profile options that begin with “FND.” Note that you cannot update any fields.

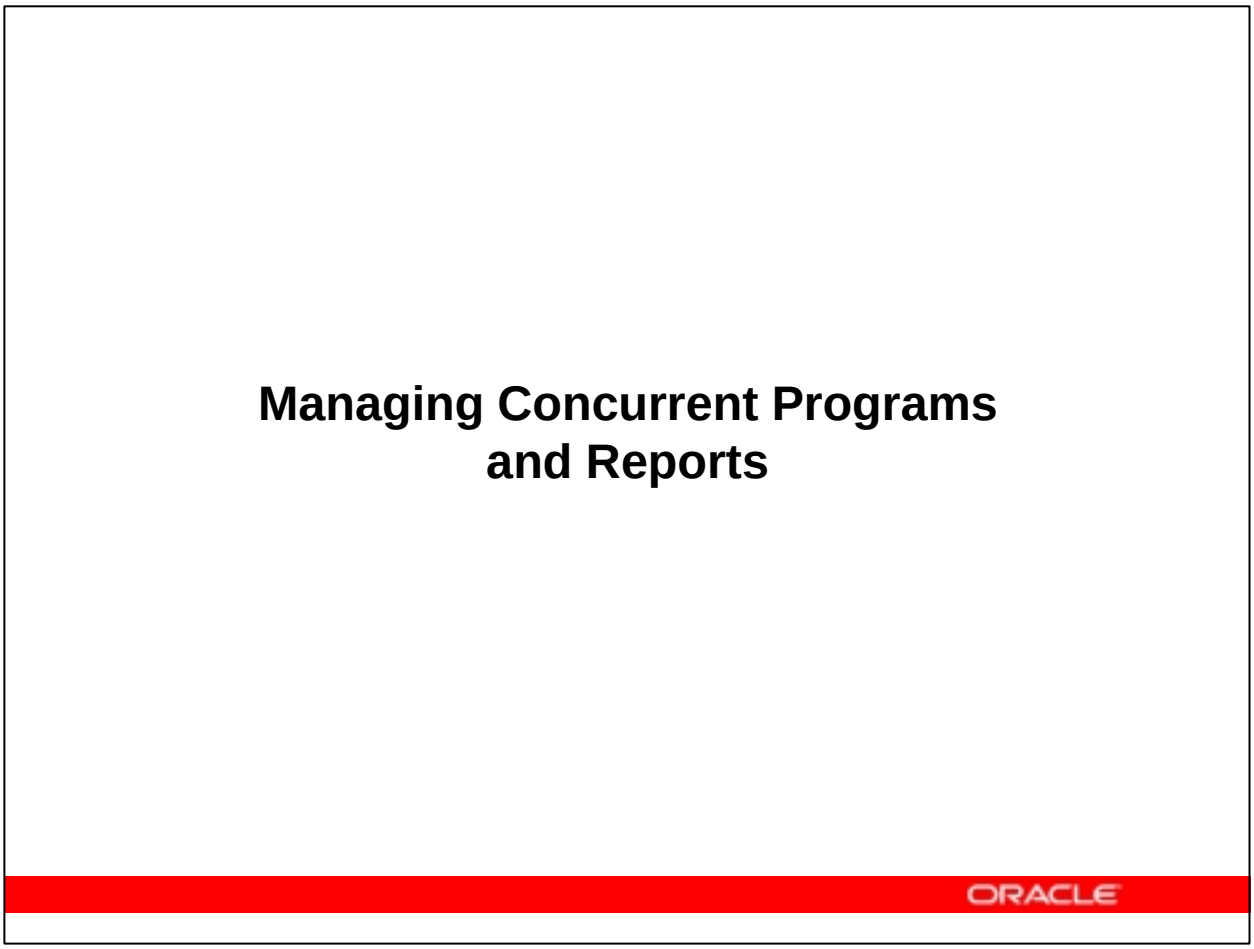


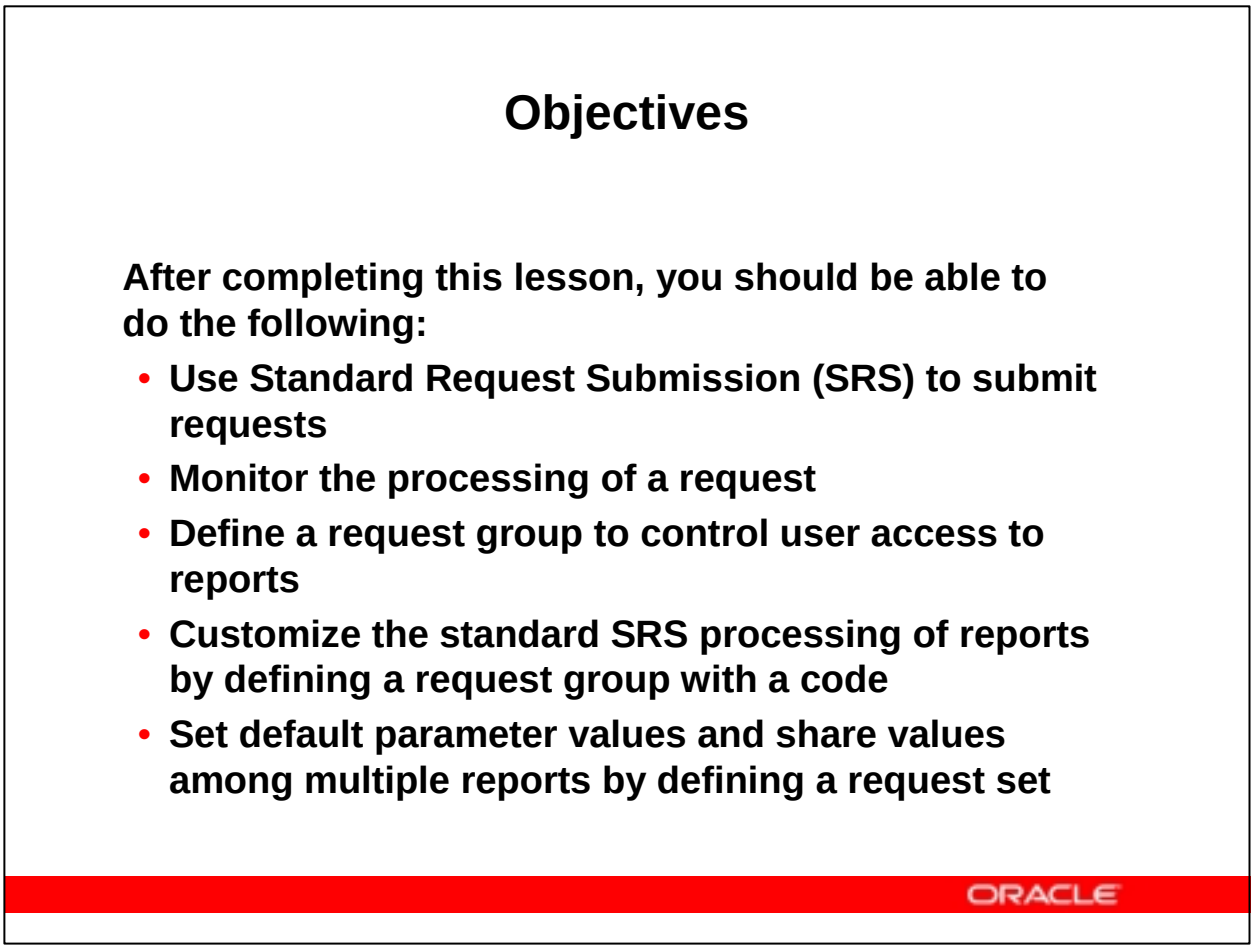
Lesson Summary

One of the most important functions of the system administrator is controlling applications security. The system administrator specifies which users have access to which functions. User signons provide the user access to an application. Responsibilities control that user's access to various functions of the application. Online access is controlled through the use of menus and menu exclusions. Report and concurrent program access is controlled through the use of request groups.

Managing Concurrent Programs and Reports

Chapter 3





Lesson Aim

An important area of security involves controlling access to reports and other concurrent programs. This level of security is provided by request group definitions. This lesson shows how the system administrator creates and uses these request groups.

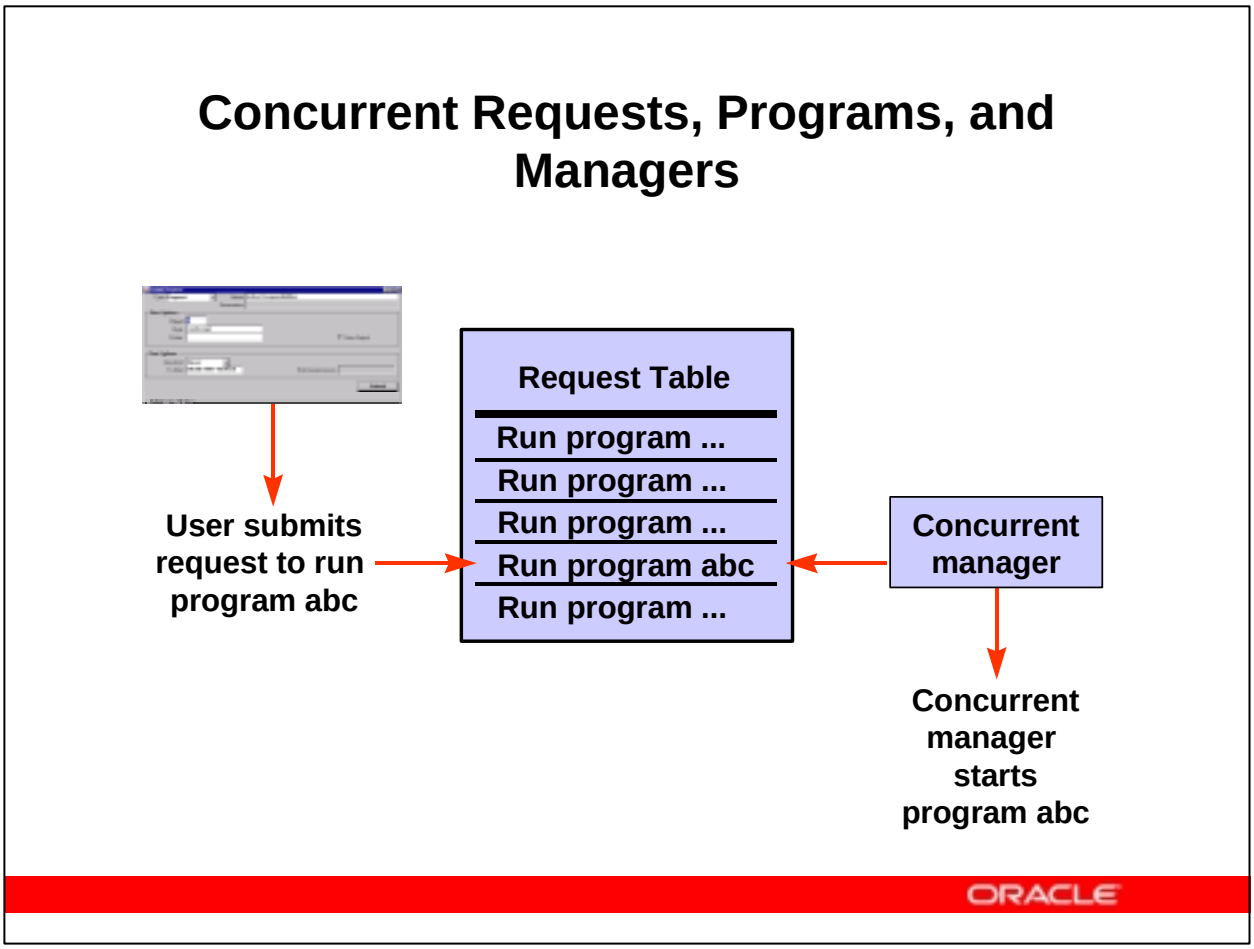
Similar to request groups are request sets – collections of programs along with their parameters and options. Request sets can be created by users, but they are still managed by the system administrator.

Objectives

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

- **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**

ORACLE



Concurrent Requests, Programs, and Managers

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.

Business Needs for Concurrent Processing

Concurrent processing helps you satisfy the following business needs:

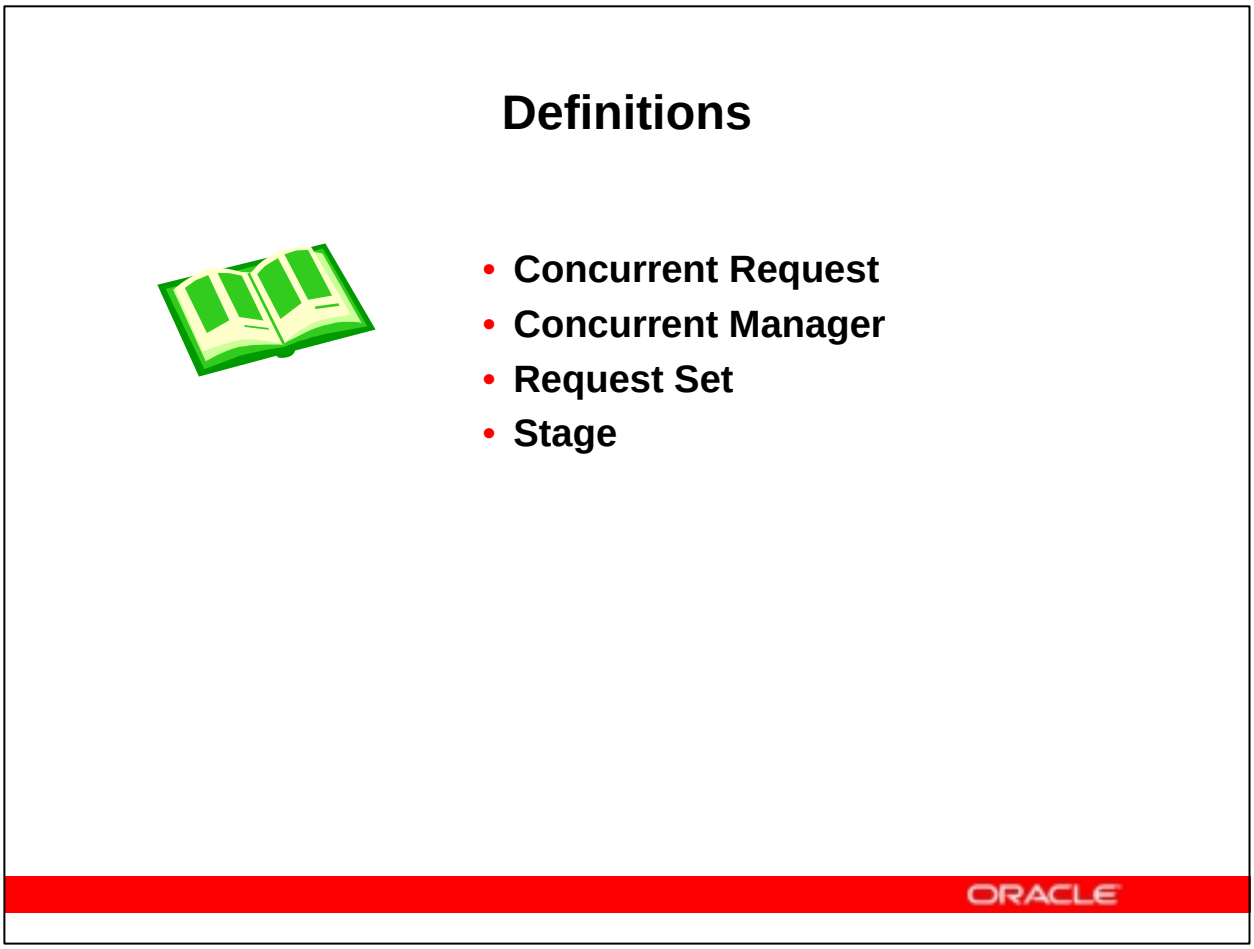
- **Continue working at your computer while running data-dependent reports and programs.**
- **Fully use the capacity of your hardware by executing many application tasks at once.**

Business Needs for Standard Request Submission

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**

ORACLE



Concurrent Processing Definitions

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 (SRS) or when you click an action button in a product-specific submission window.

Concurrent Manager

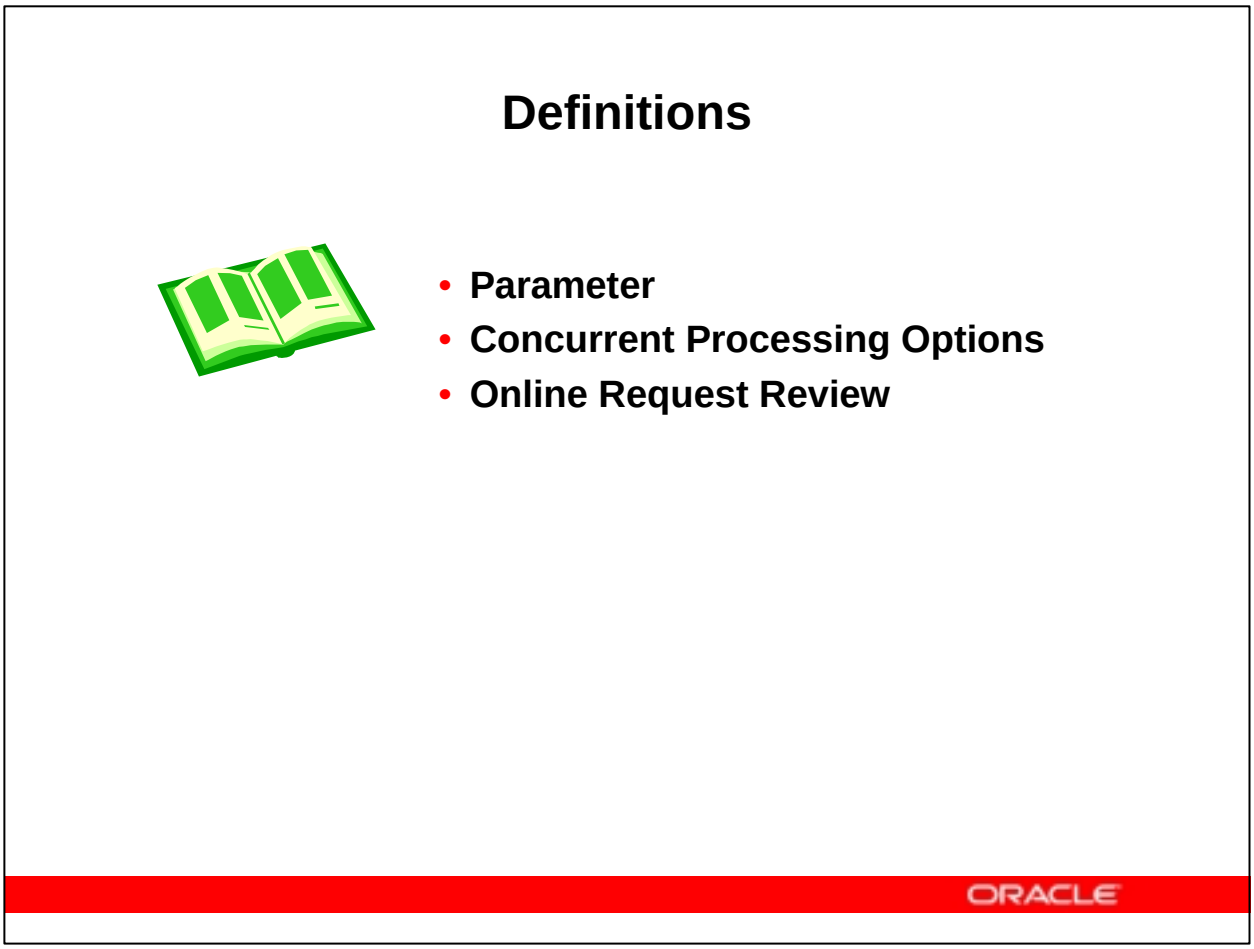
A *concurrent manager* is a component of concurrent processing that monitors and runs requests without tying up your computer.

Request Set

A *request set* is a collection of reports or programs that you group together. You submit the reports or programs in a request set all at once, using one transaction.

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.



Concurrent Processing Definitions (continued)

Parameter

In SRS, a *parameter* is a report variable whose value you can change each time you run a report. For example, you might run an account balance report and change the account number whenever you run the report. The account number is the parameter for the report.

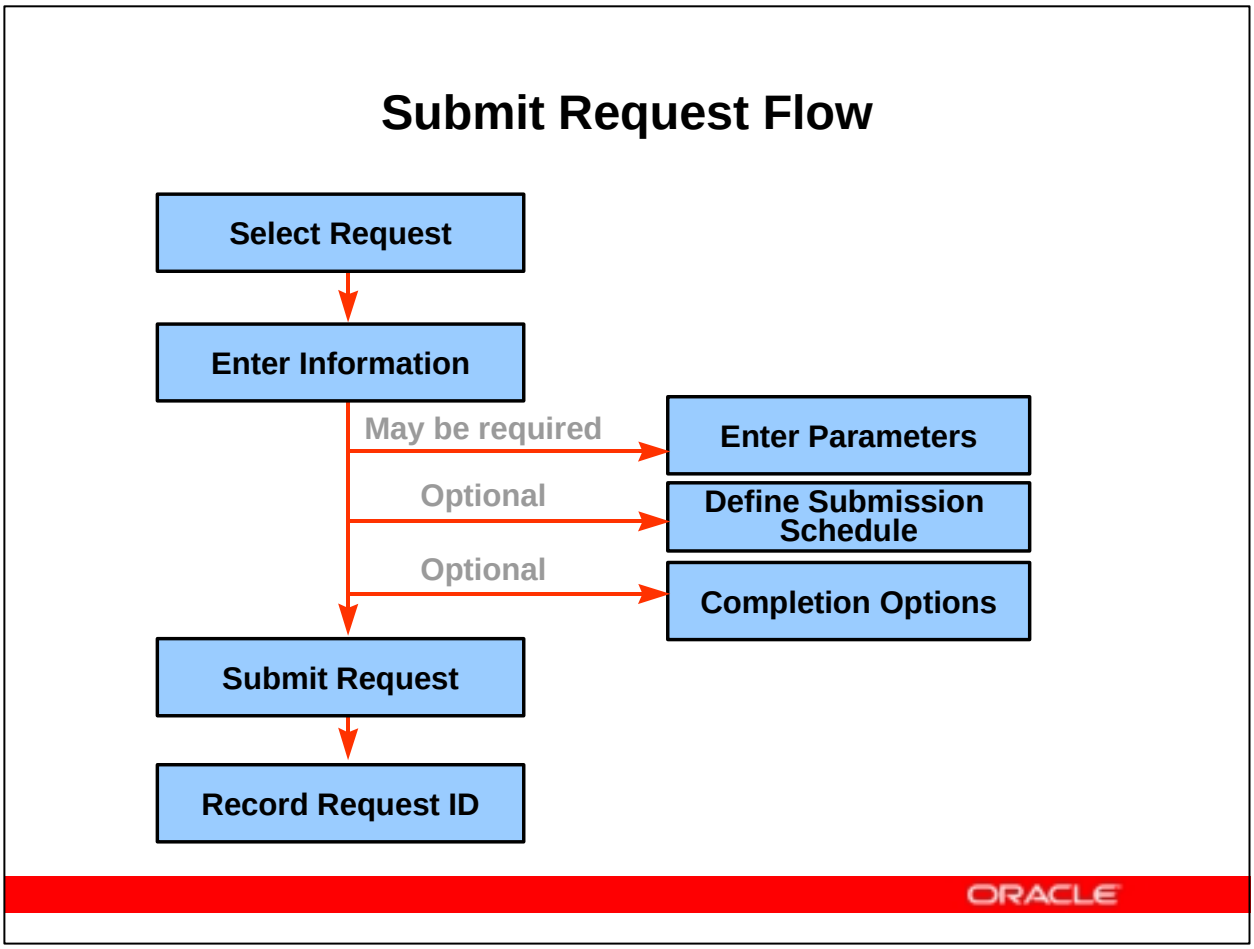
Concurrent Processing Options

You can control and change certain run options for each of your concurrent requests. Not only can you control and change the number of copies to print, which print style to use, and which printer to use, but you can also choose to hold a request from being run, choose a specific date or time to start a request, or choose to save the results of your concurrent request in a standard file format.

Online Request Review

You can review the output and log files from your concurrent requests online. You can see the results of a concurrent request without the delay of printing out the entire report or log file.

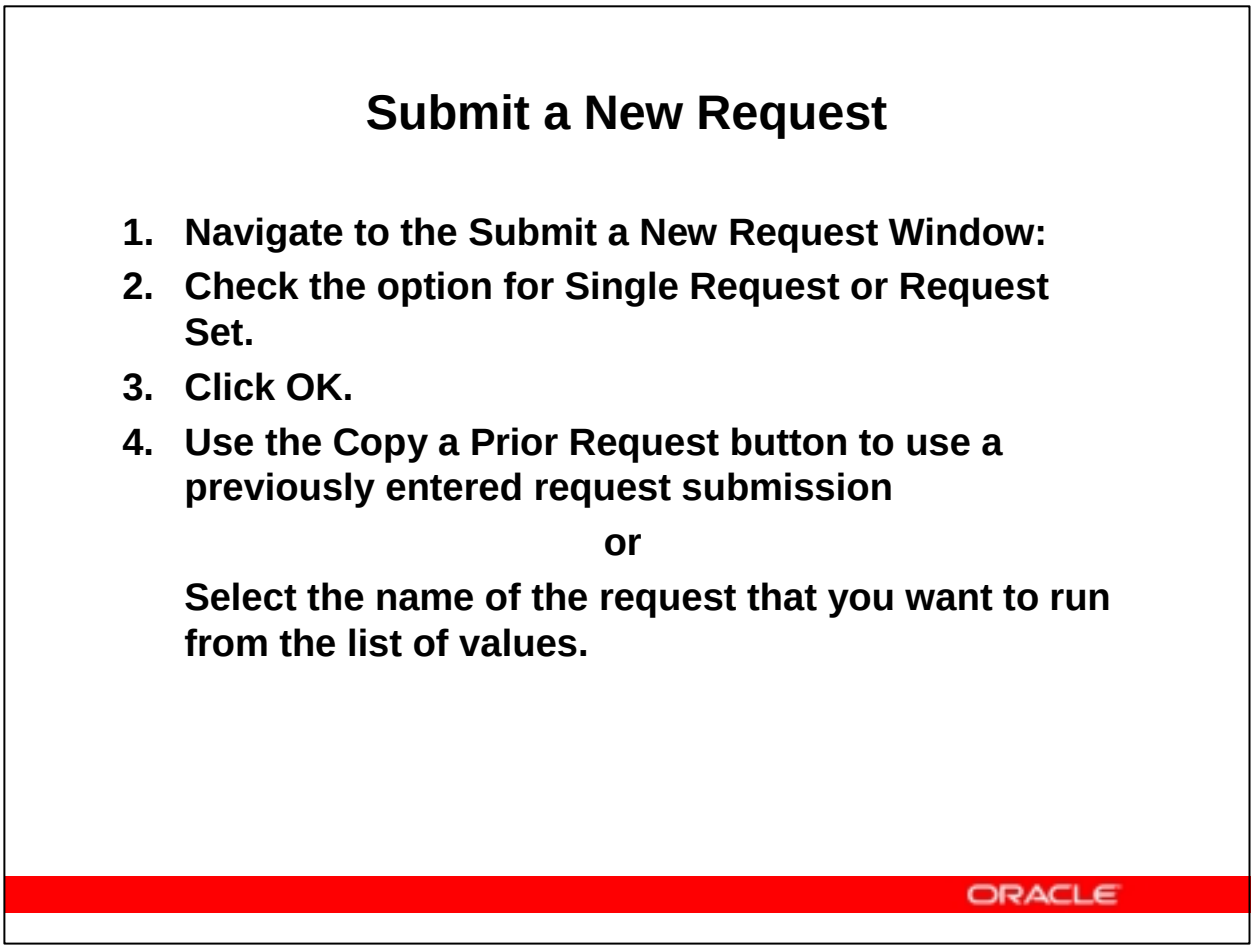
Submit Request Flow



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.



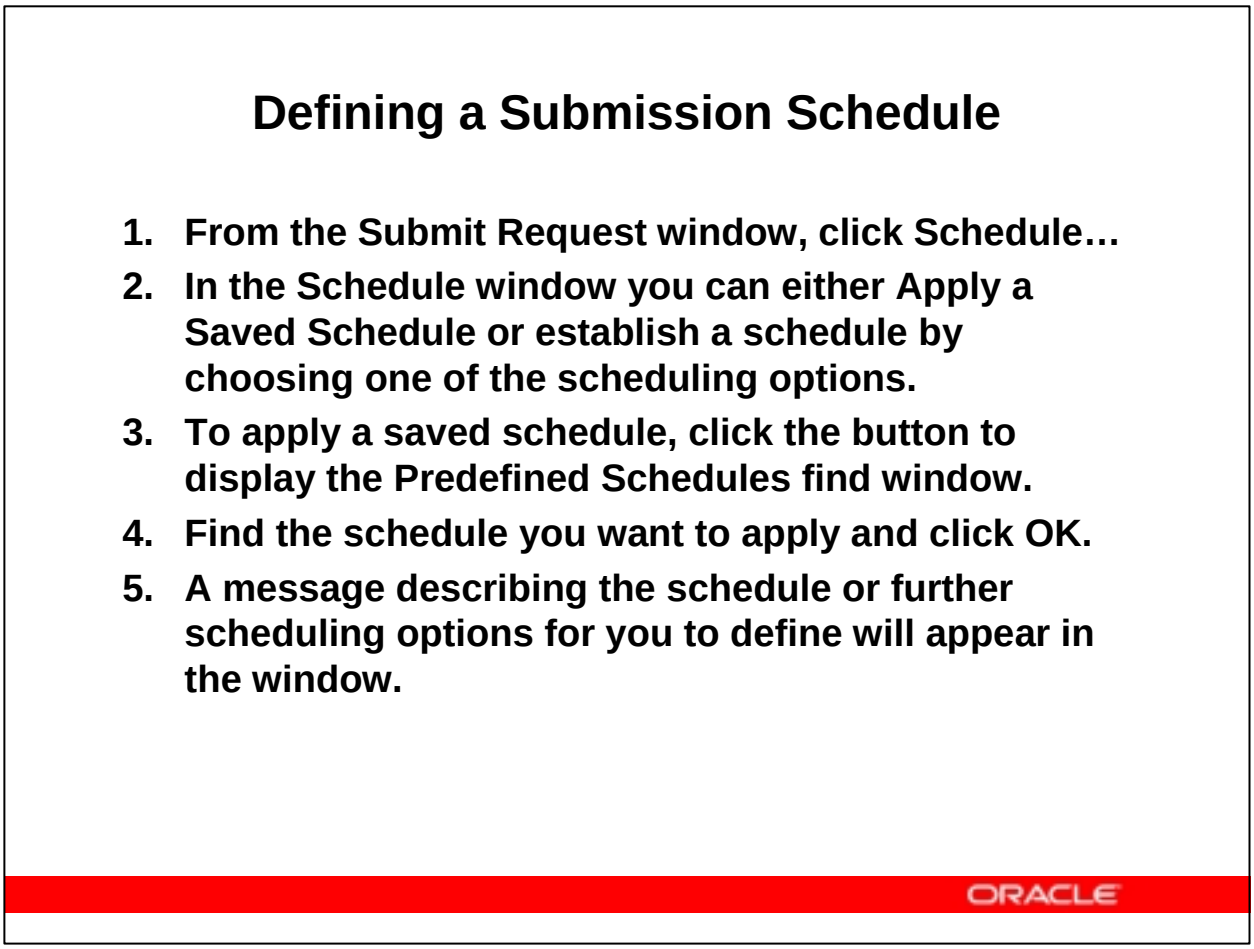
Submit a New Request

(N) Requests > Run

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

Using the Parameters Window

- **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.**



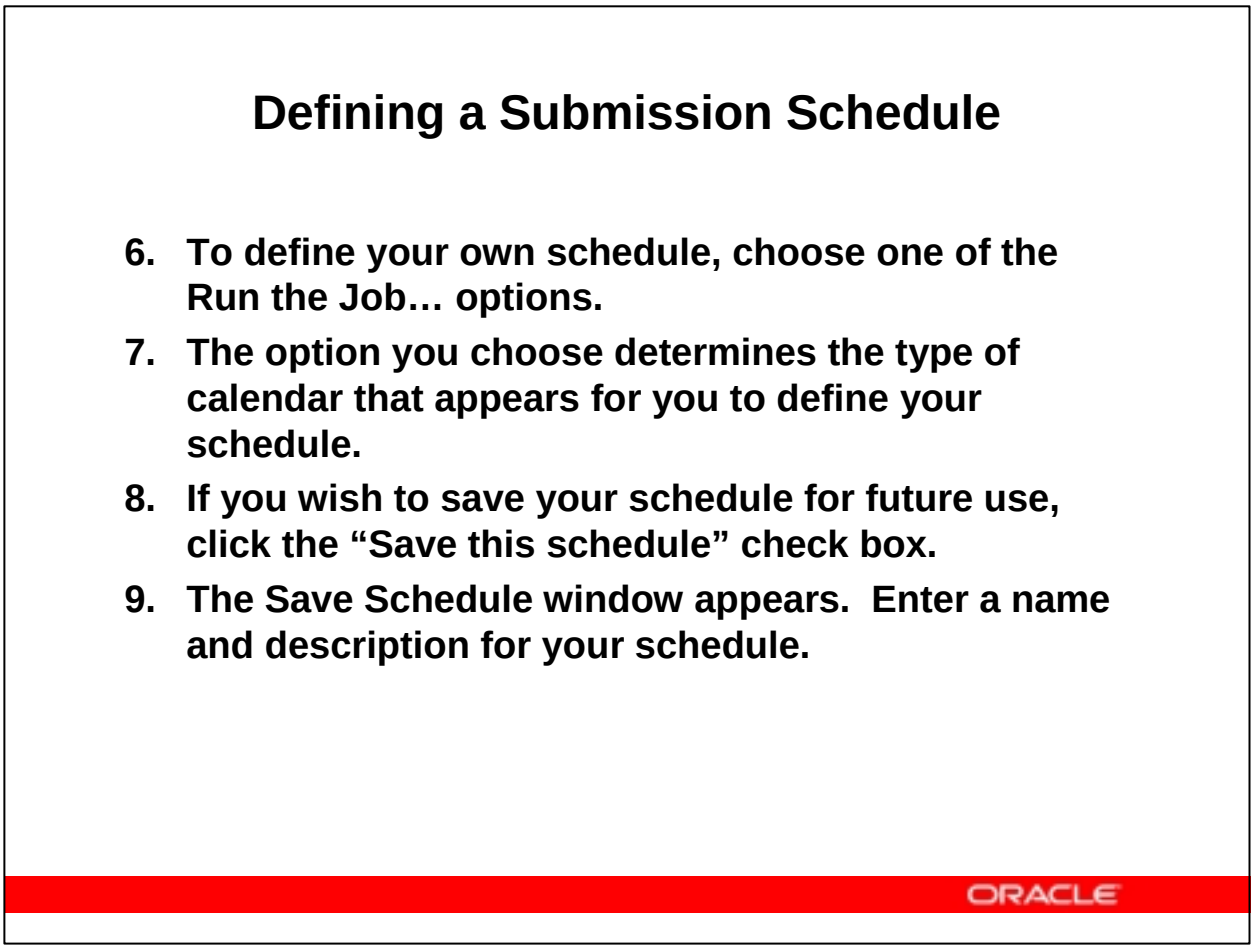
Submitting Requests

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

The Schedule Window

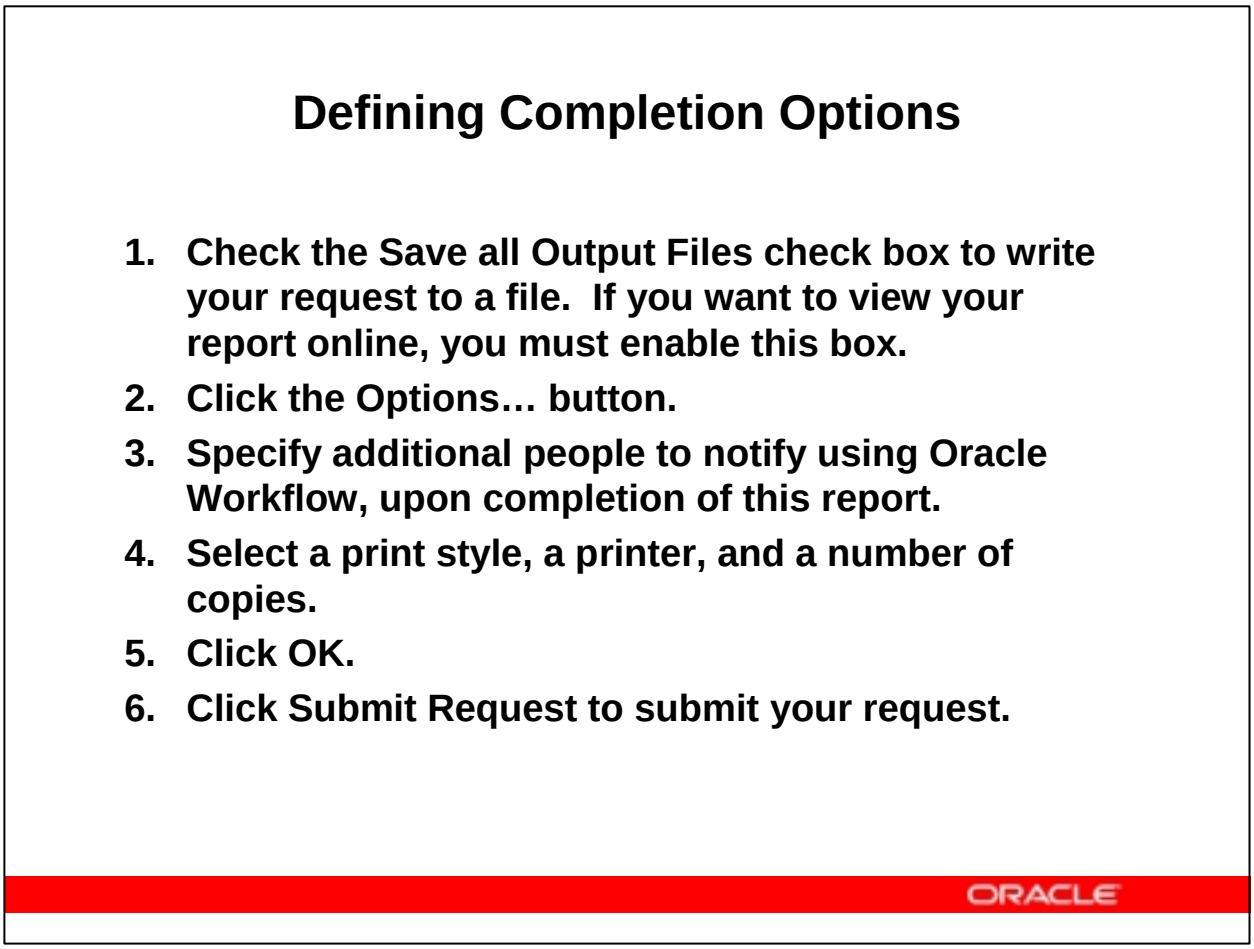
The scheduling window provides you with several scheduling options. You can choose to reuse a schedule that you previously defined and saved, or define a new schedule. You can define your schedule to run a request as soon as possible, at a specific time, or repeatedly at specific intervals, or on specific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



Define Submission Schedule (continued)

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



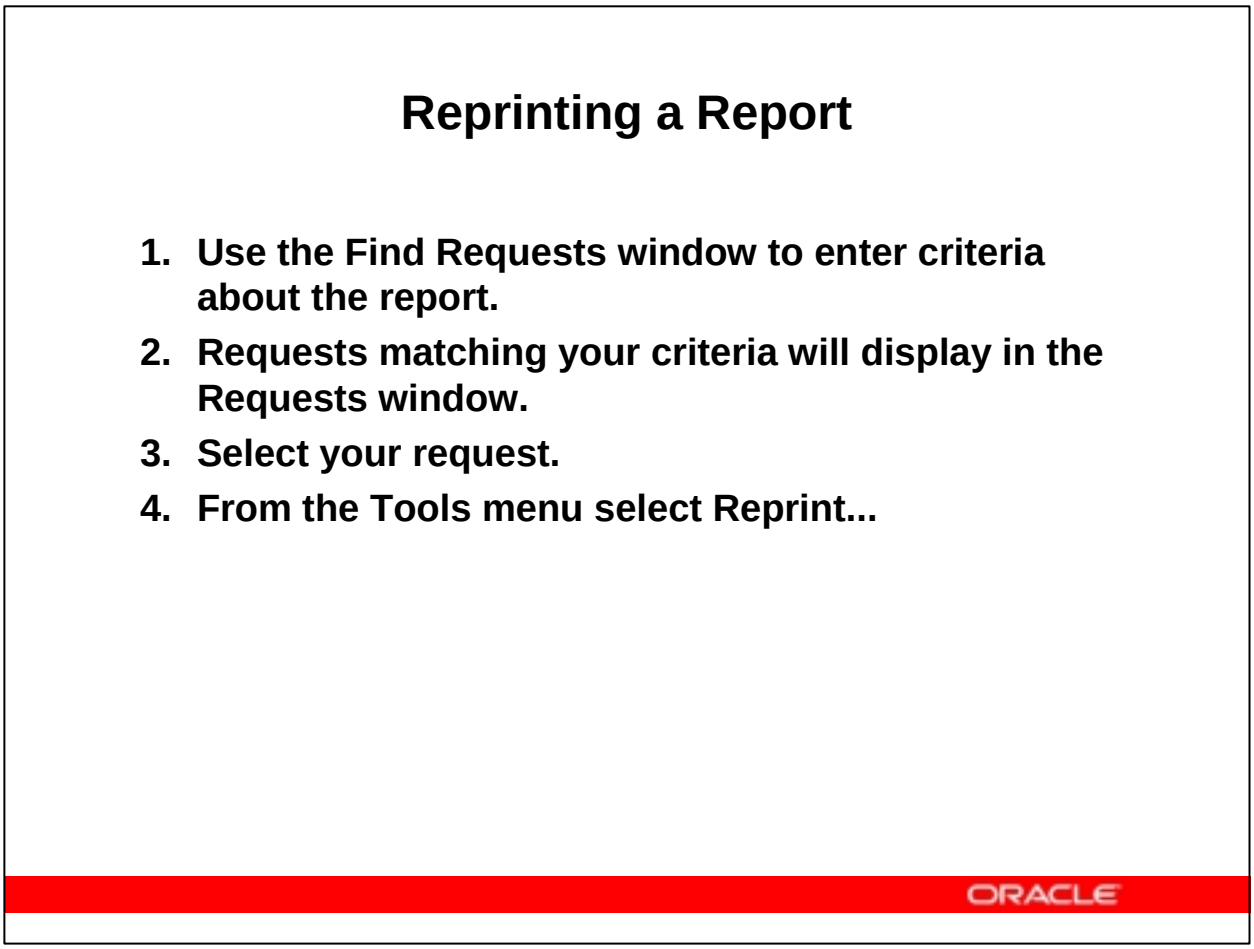
Submitting Requests

(N) Concurrent > Requests > Submit a New Request > (B) OK > (B) Options...

Request ID

- **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.**

ORACLE



The Find Requests Window appears when you navigate to (N) Requests > View.

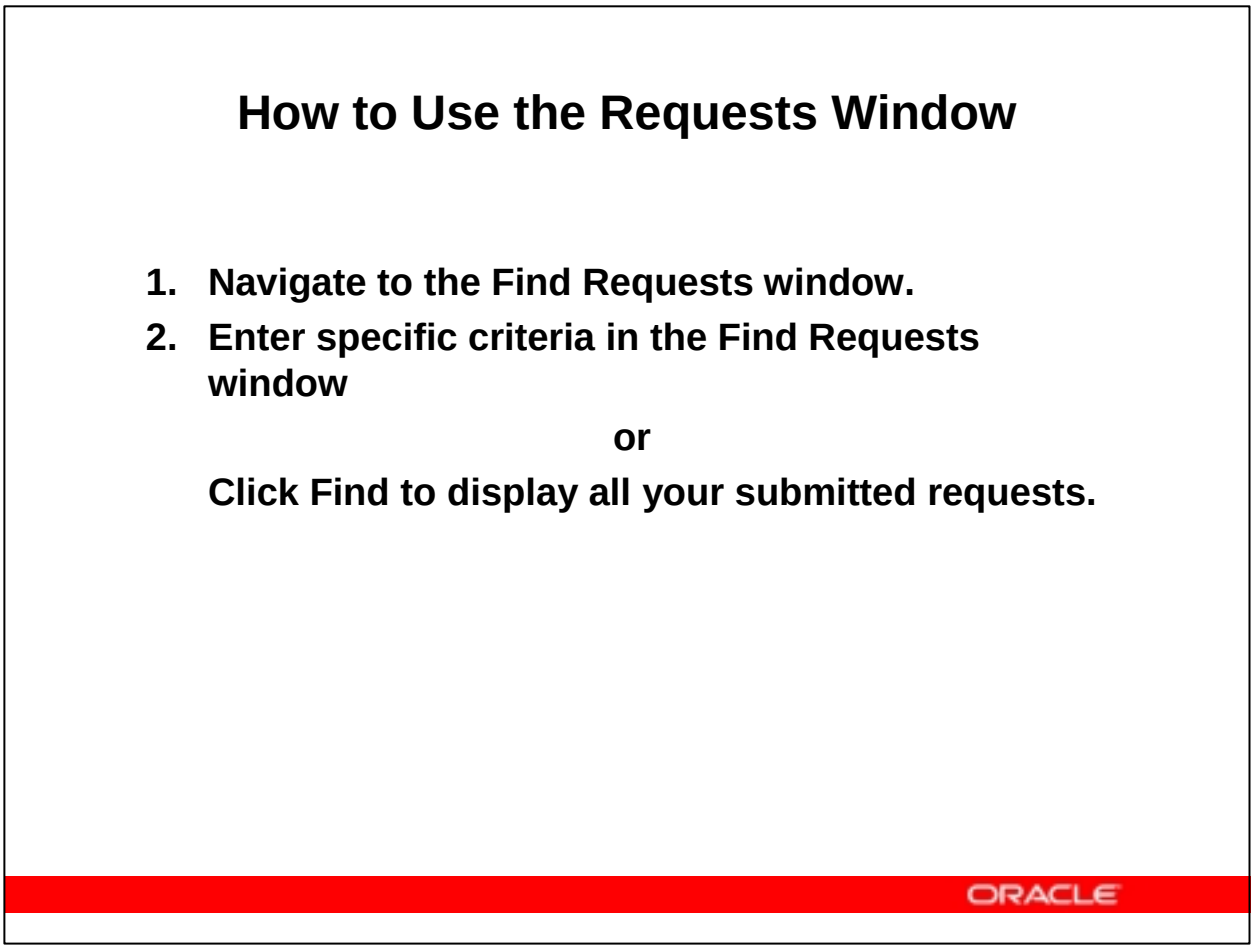
Note: The Profile Option Concurrent:Save Output must be set to Yes.

Use the Requests Window to View and Change Requests

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**

ORACLE



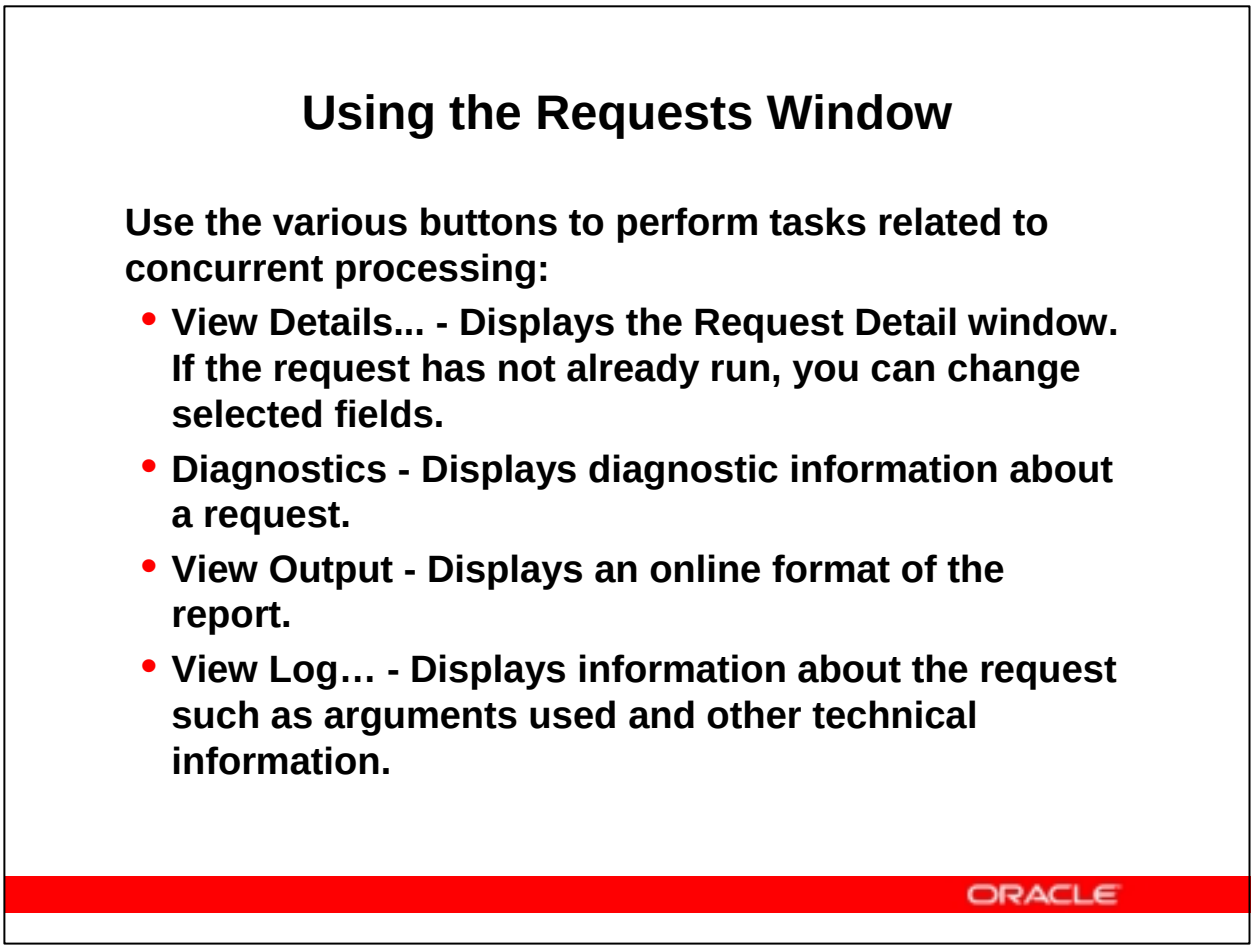
Display the Find Requests window by navigating to (N) Requests > View.

Using the Requests Window

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**

ORACLE



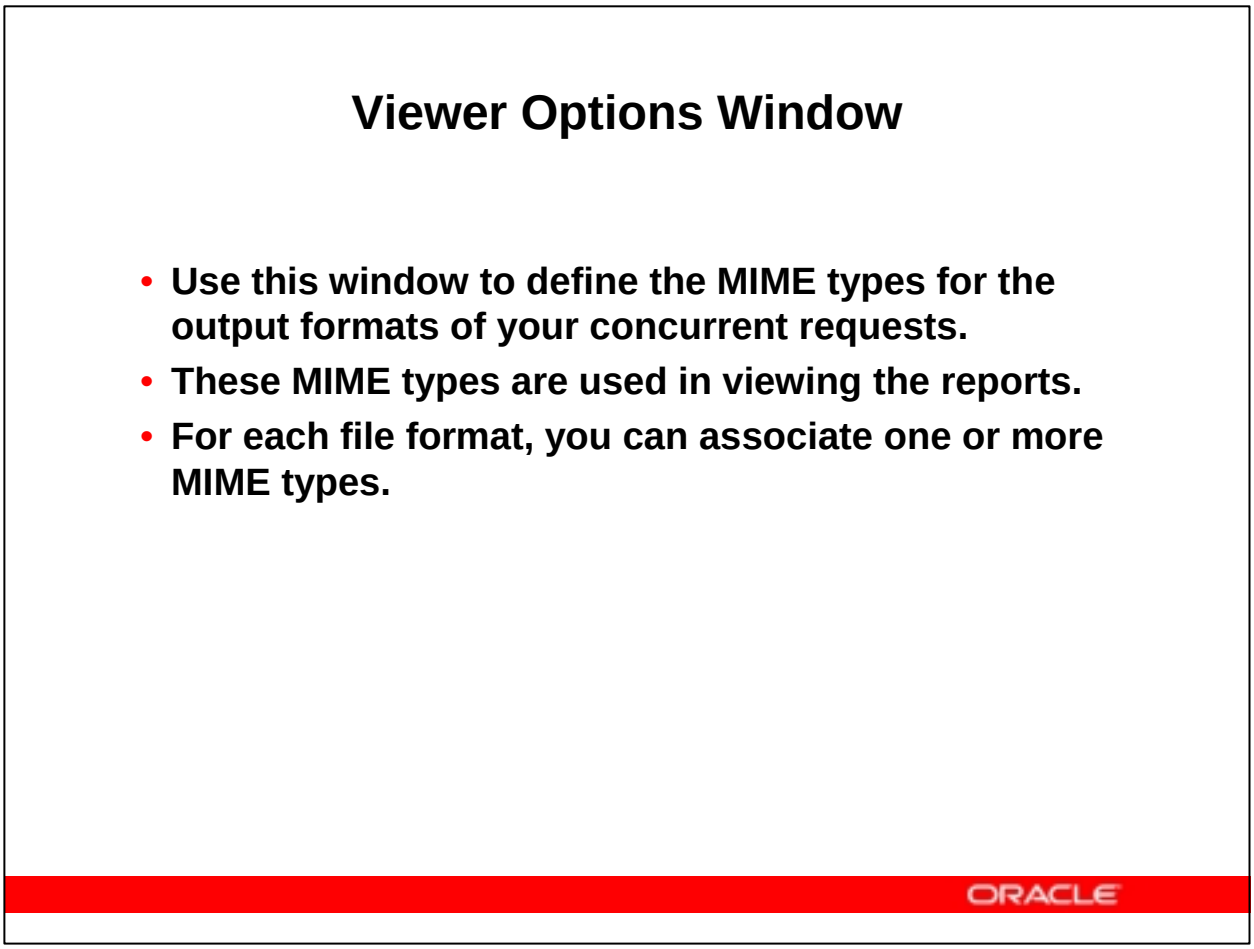
Using the Requests Window (continued)

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.



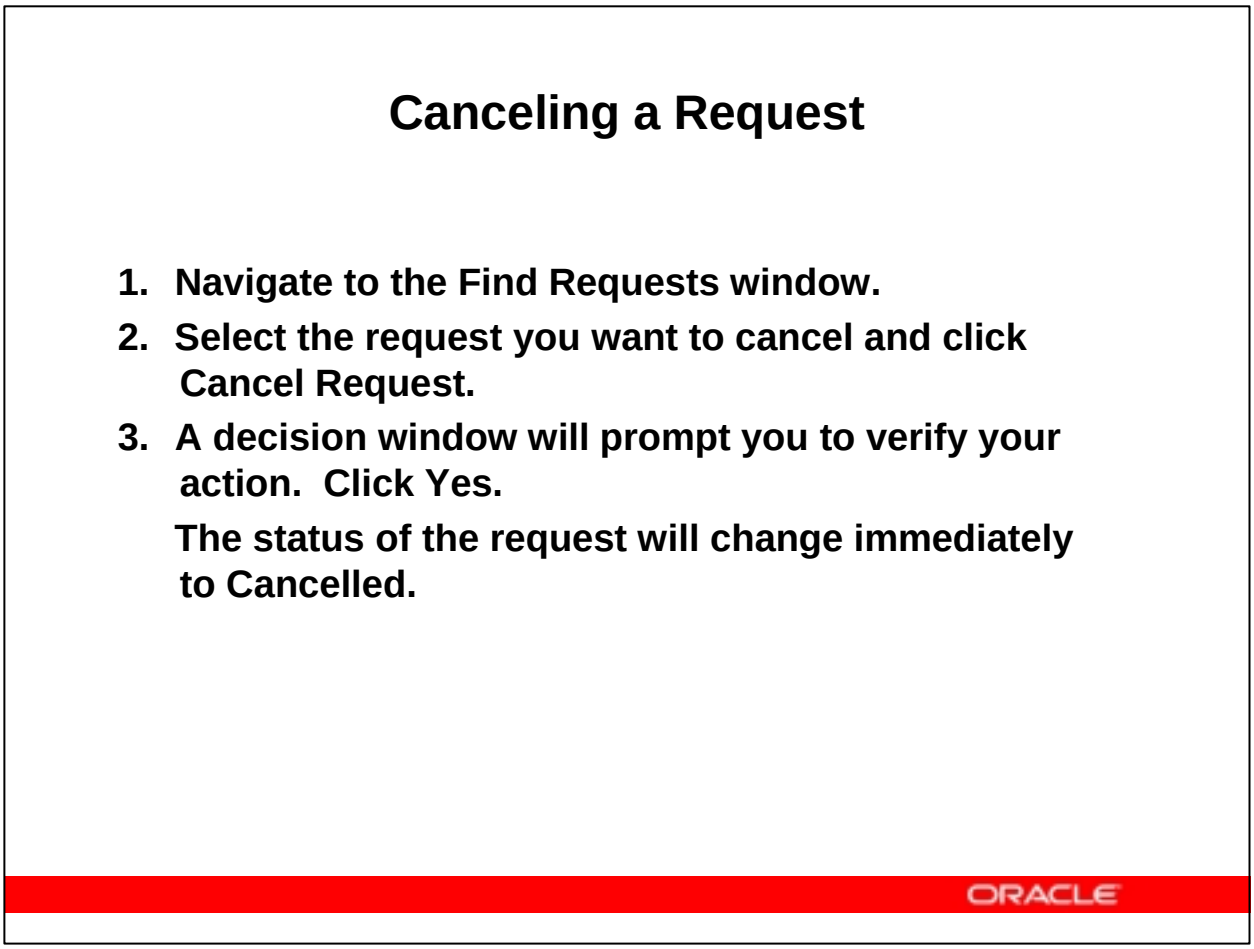
Use the Viewer Options Window to Define MIME Types

(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
- Viewer: Application for XML

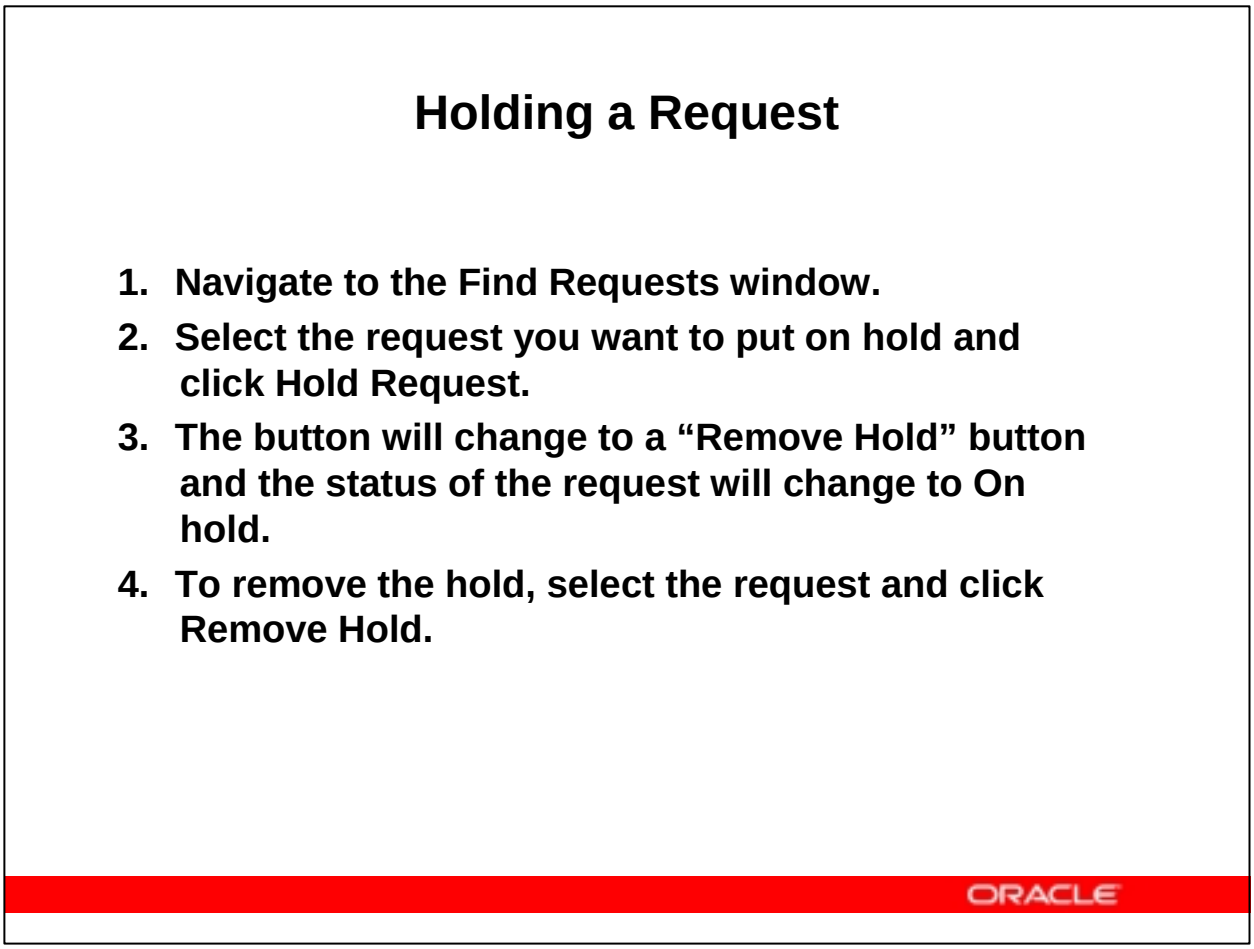
This MIME type is sent to a browser window when the user views a report of that file format.



Canceling a Request That Has Not Yet Completed

(N) Requests > View (B) Find

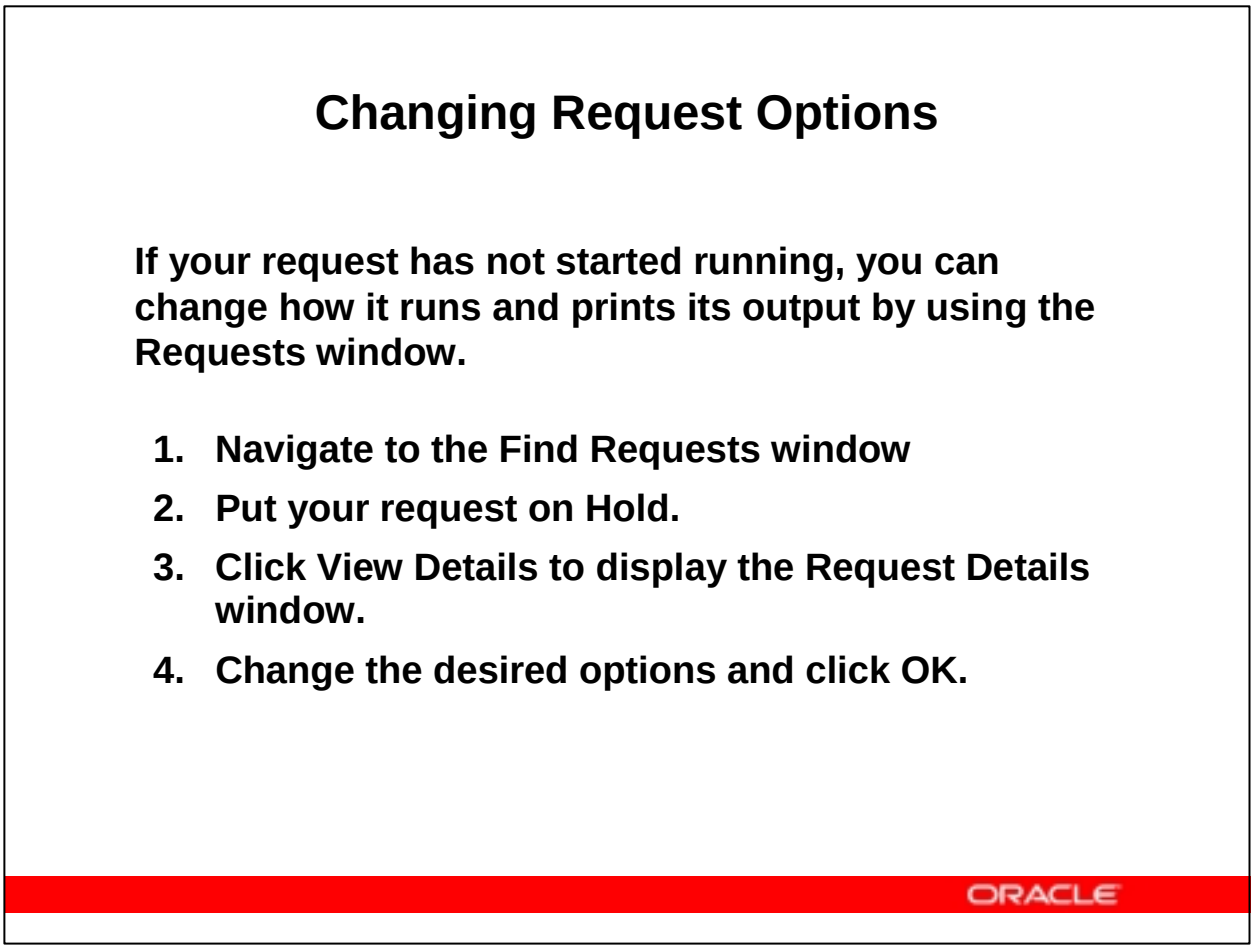
If you cancel a request set, then Oracle Applications automatically cancels all requests in the set.



Holding a Request That Has Not Started Running

(N) Requests > View (B) Find

Only requests that have not started running can be put on hold.

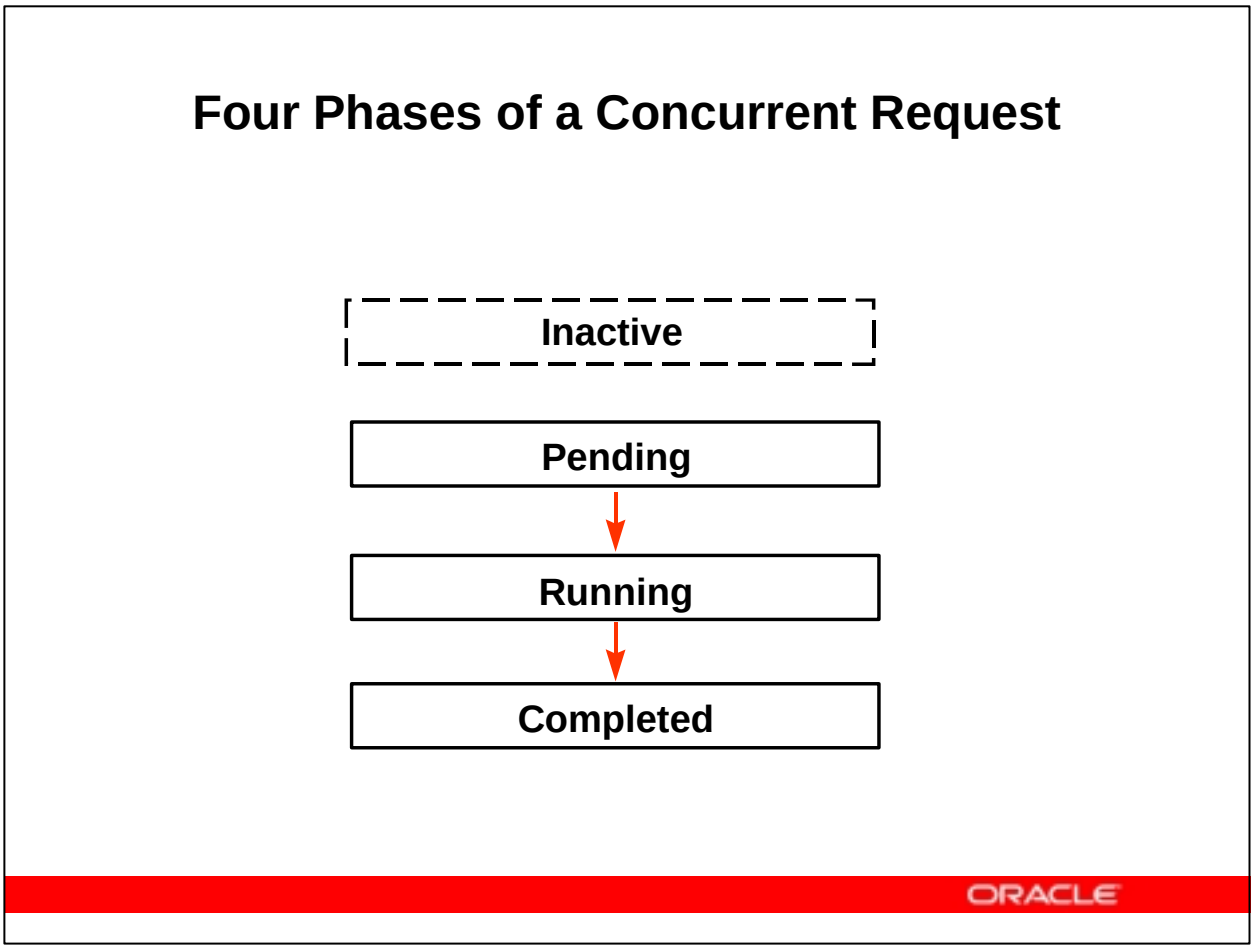


Changing Request Option

(N) Requests > View (B) Find

Note: You can change the following only if the request or request set is Pending or Scheduled:

- Start Date and Time
- Printer and number of copies
- Save output
- Resubmission options

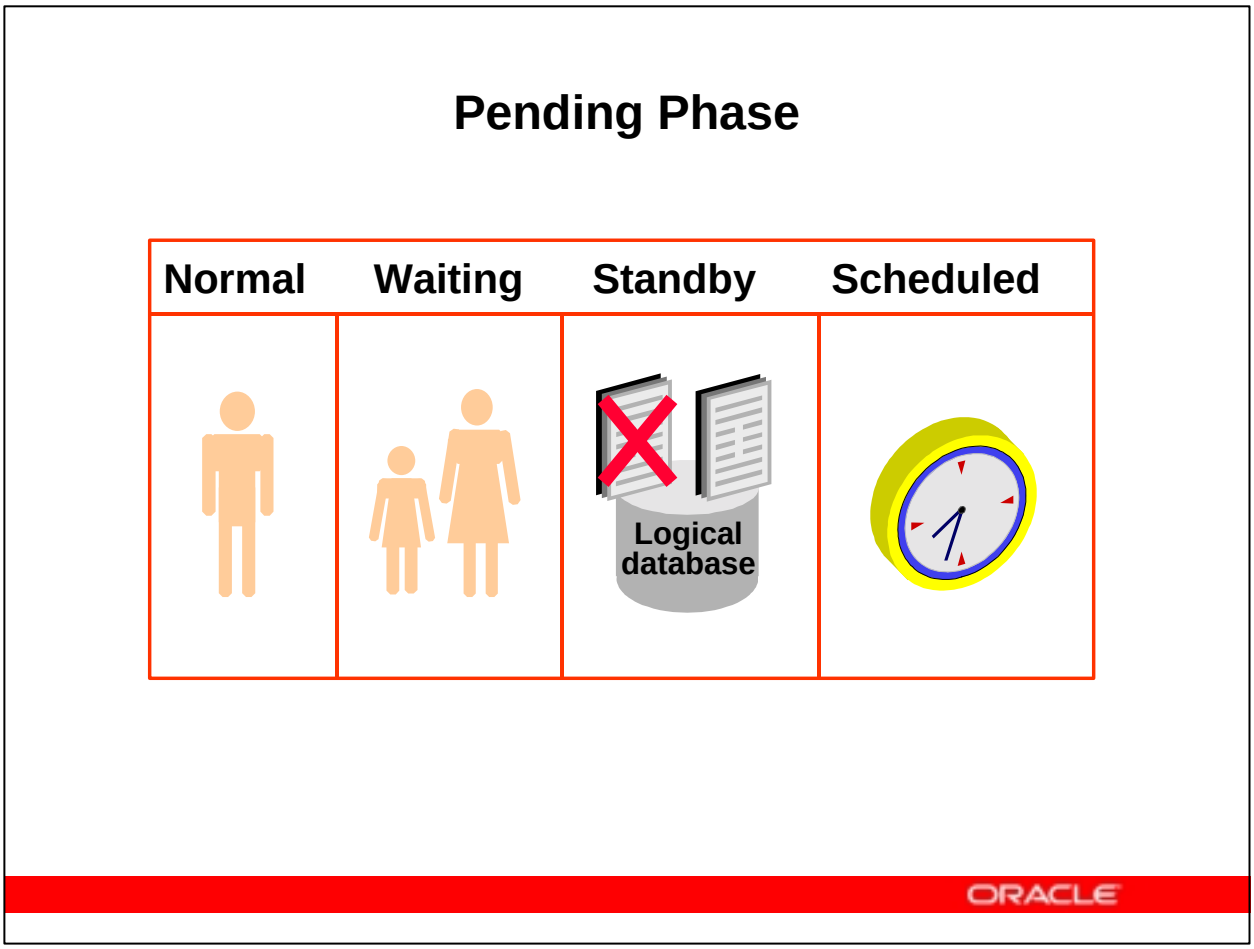


Four Phases of a Concurrent Request

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:

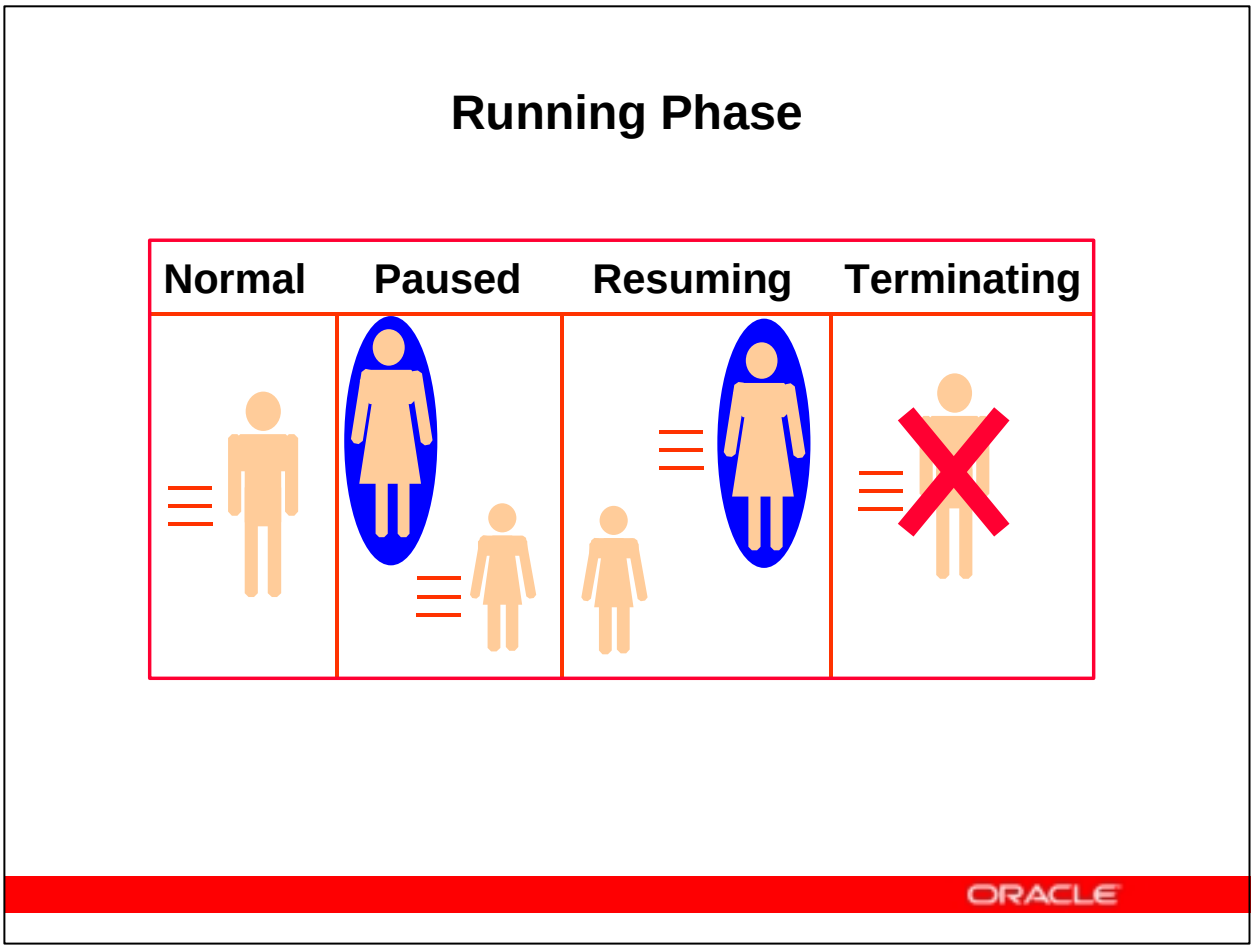
- **Pending:** The request is waiting to be run.
- **Running:** The request is running.
- **Completed:** The request has finished execution.
- **Inactive:** The request cannot yet be run.



Pending Phase

A program in the Pending phase can be in one of four statuses:

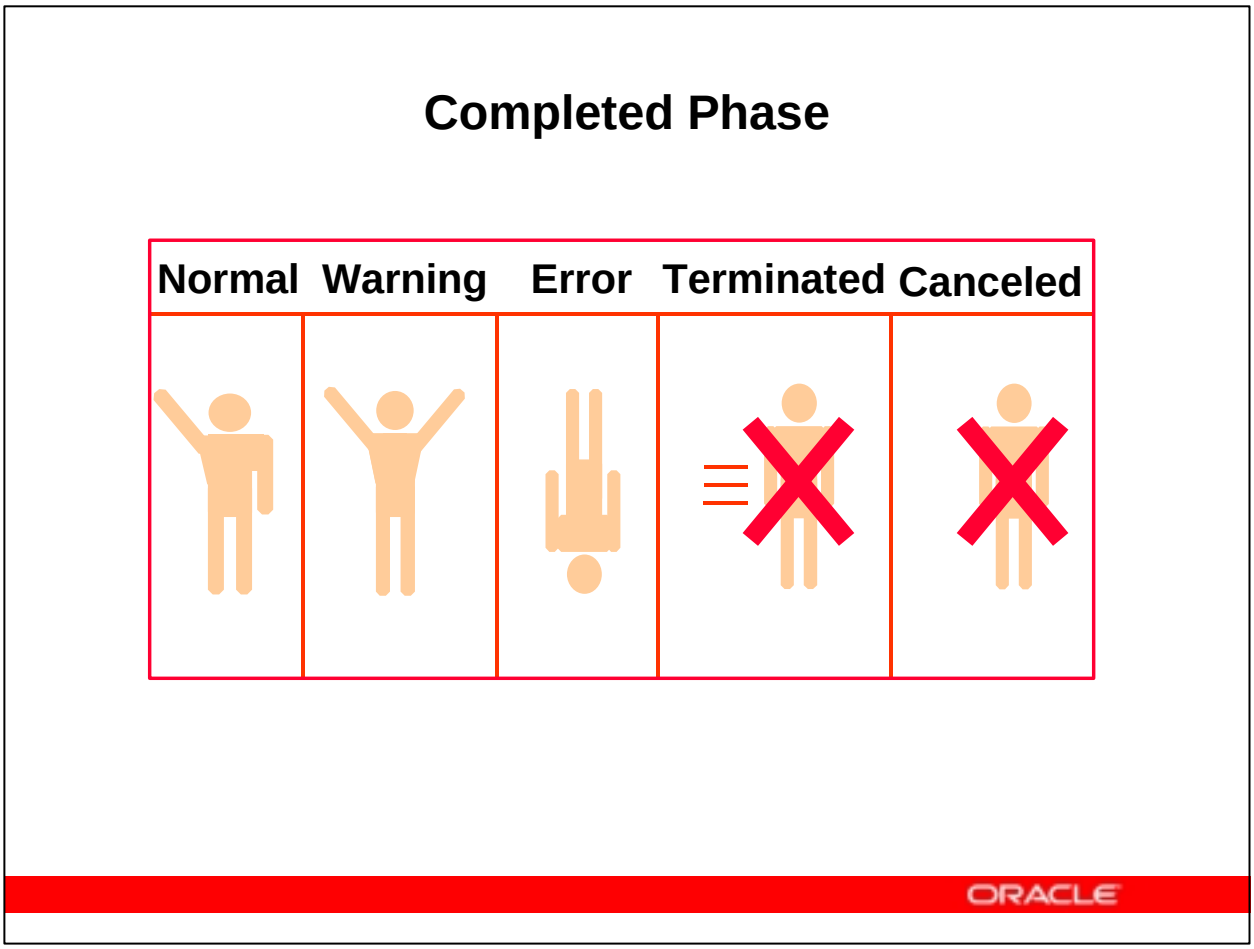
- **Normal:** The program is waiting for an available manager.
- **Waiting:** A child program is waiting for a parent to mark it ready to run.
- **Standby:** A program is waiting for another incompatible program in the same logical database to complete.
- **Scheduled:** A program's scheduled start time has not yet elapsed.



Running Phase

A program in the Running phase can be in one of four statuses:

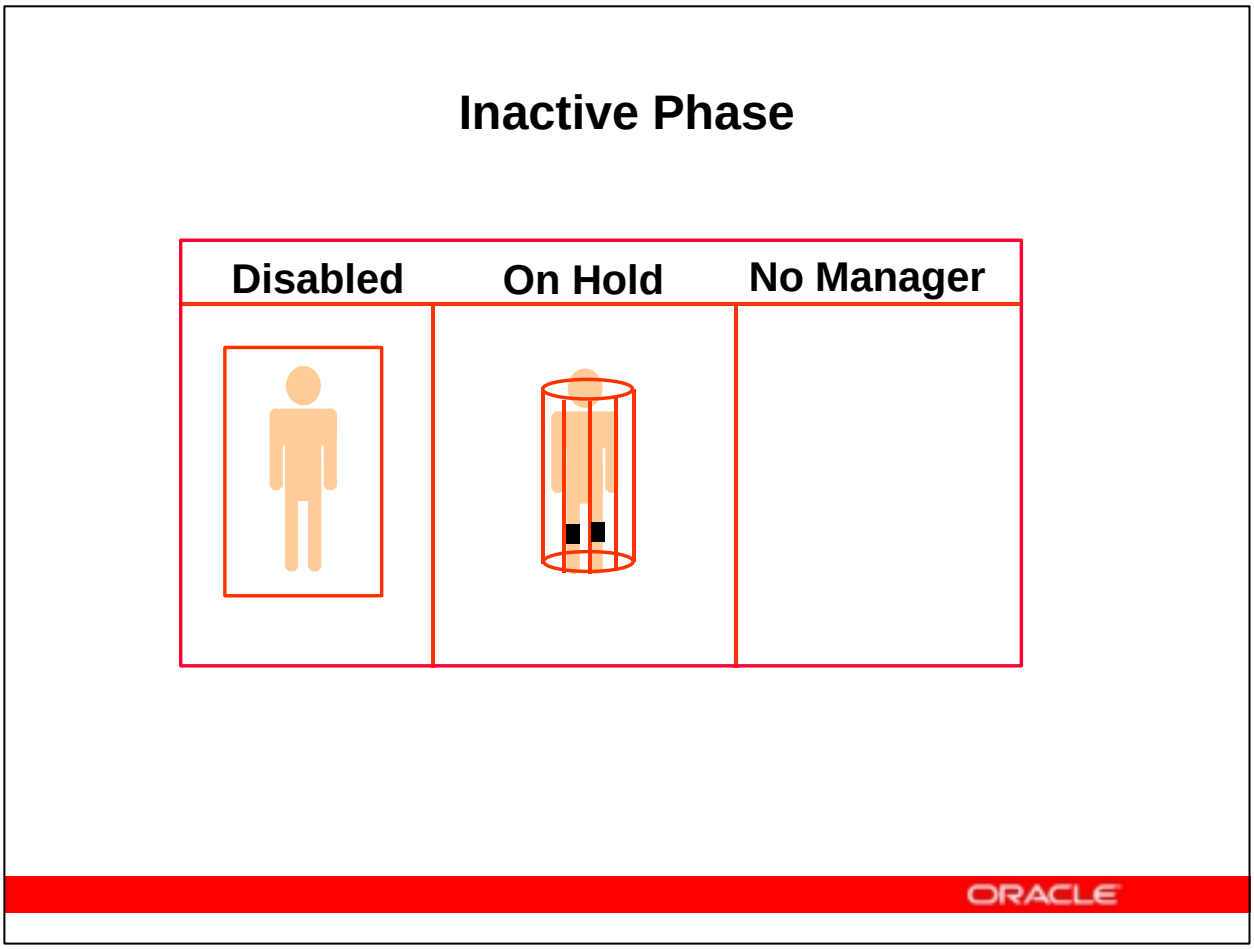
- **Normal:** Program is in progress.
- **Paused:** A parent program is waiting for one or more child programs to complete.
- **Resuming:** A parent program is continuing after the completion of one or more child programs.
- **Terminating:** The program is being terminated.



Completed Phase

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.

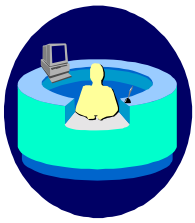


Inactive Phase

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.

System Administrator Monitoring Privileges

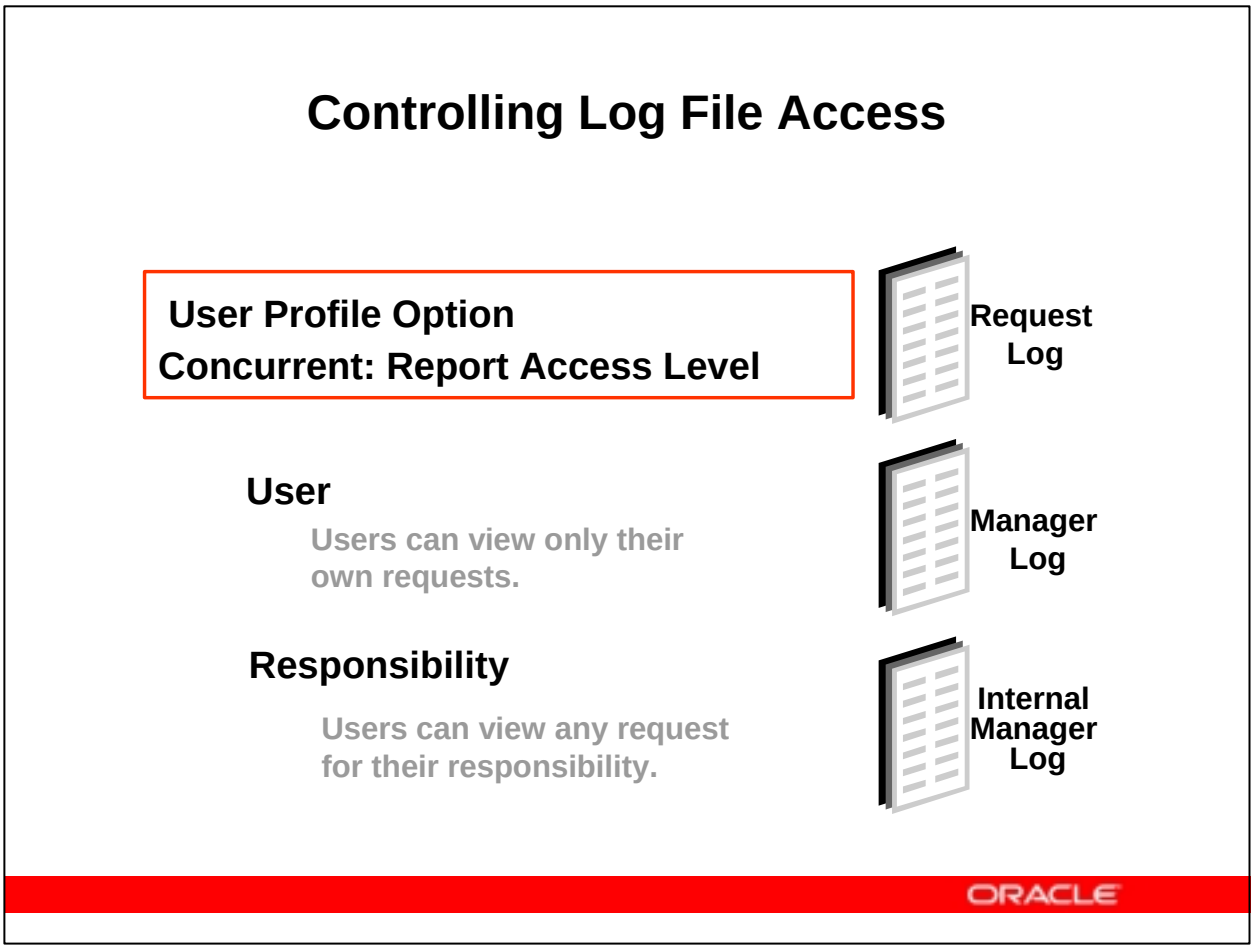


- Place any pending request on hold.
- Take any request off hold.
- Terminate any request.
- Change the priority of a request.
- View the log file and the manager log file.

ORACLE

Monitoring Privileges of the System Administrator

The system administrator's privileges extend beyond those of the end user. The system administrator can perform all the operations shown on the slide. However, the system administrator cannot view the actual output of another user's report.



Log File Access

The user profile option Concurrent: Report Access Level determines the report and log file access privileges for end users.

User: Access to reports and diagnostic log files is limited to the user who submitted the reports.

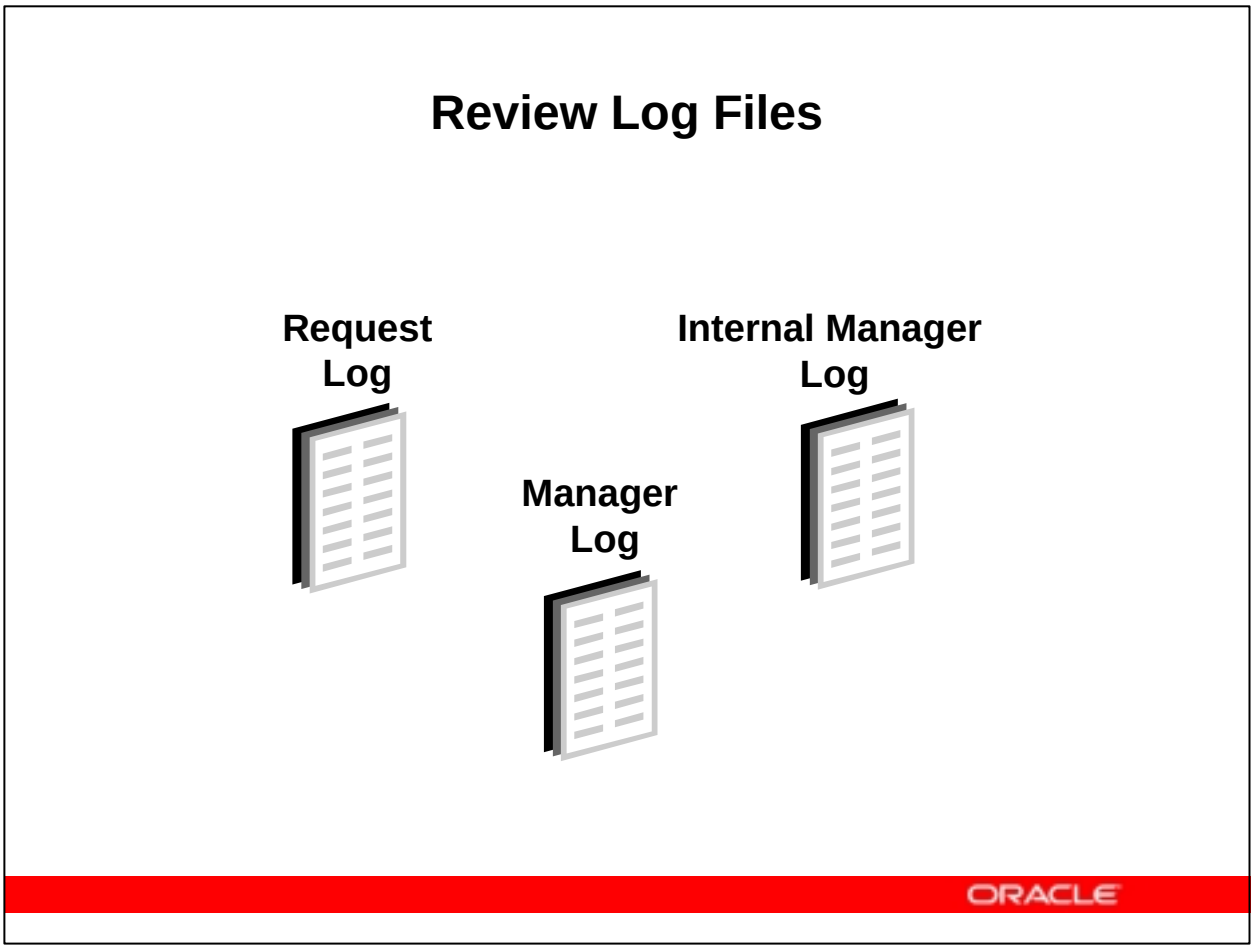
Responsibility: Access to reports and diagnostic log files is based on the responsibility that the user is currently using.

A user's report and log file access privileges include:

- Viewing the completed report output online
- Viewing the diagnostic log file online (also a system administrator privilege)
- Reprinting a completed report (if Concurrent: Save Output profile is Yes; also a system administrator privilege)

For pending requests, only the submitter of the request or the system administrator can view the details of the request.

Review Log Files

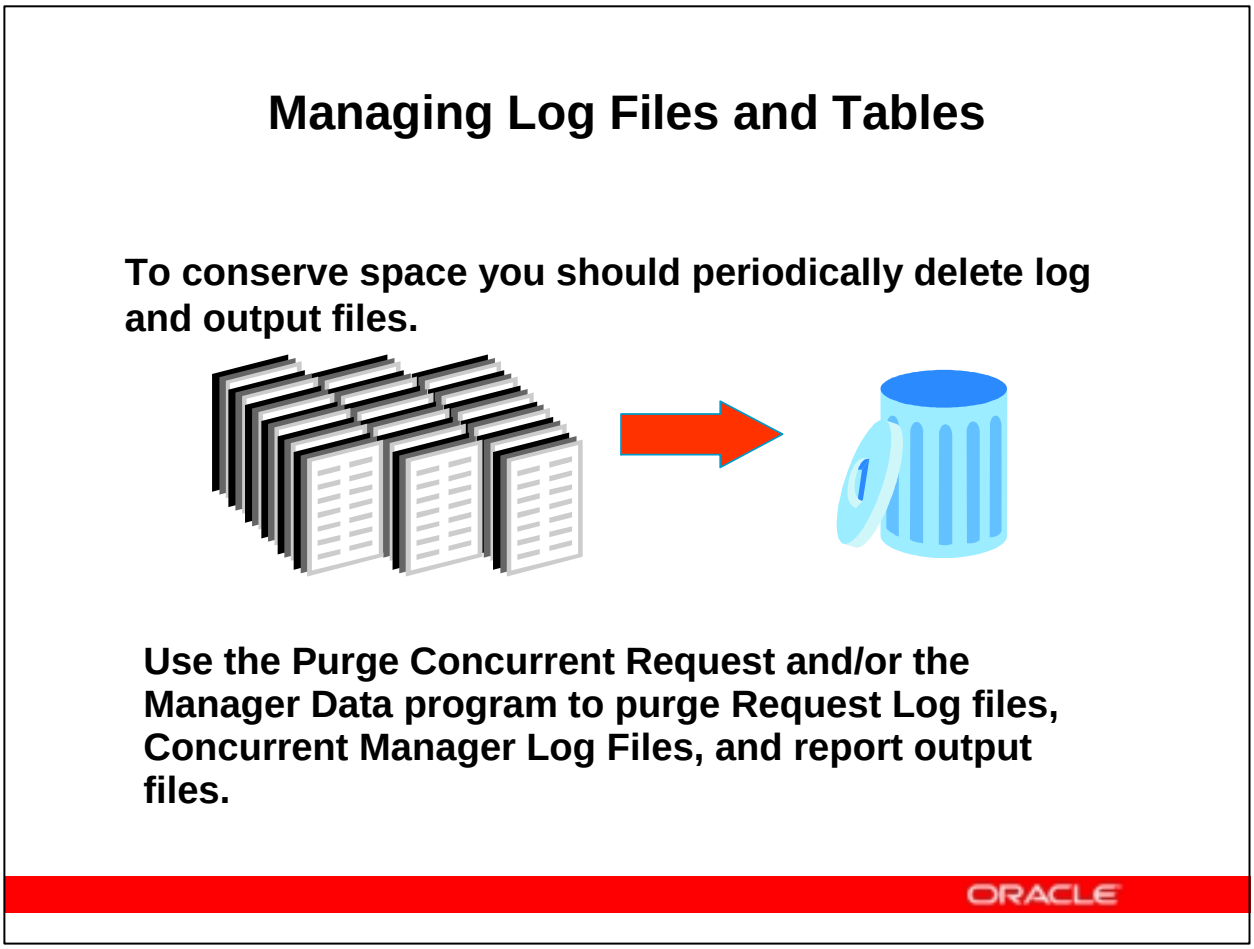


Log File Descriptions

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

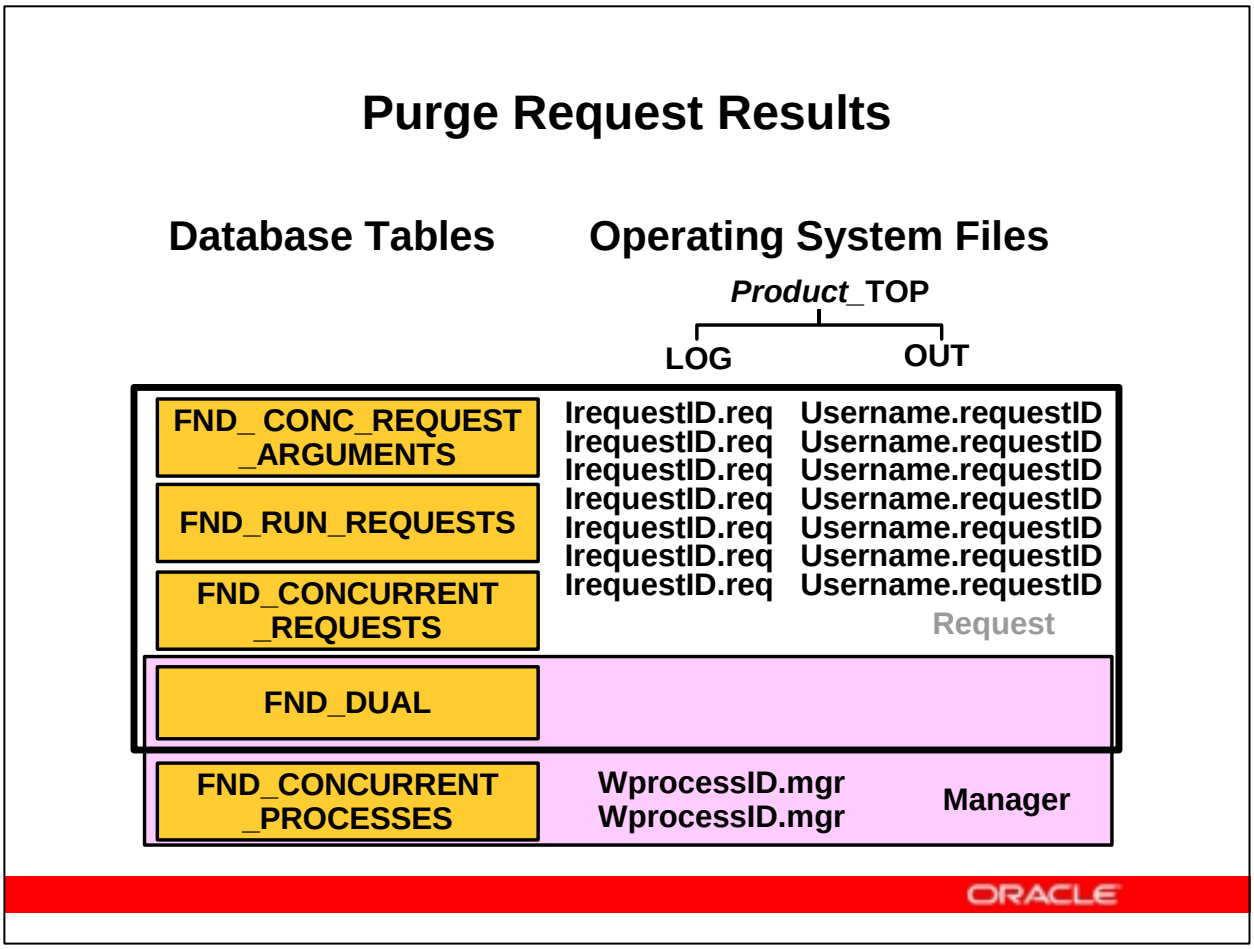
Purging Log Data Eventually the tables holding log information consume disk space with data that may no longer be of use to you. To conserve disk space, you should periodically delete Oracle Applications log files and output files. Your company's MIS department and application users should agree on an 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.

Purge Request Results



Results of Purge Requests

The slide shows the database tables and the operating system files affected by a purge request. On the Parameters form of the Purge Concurrent request and/or Manager Data program, you can specify that either request data (Entity=Request) or manager data (Entity=Manager) be purged. The slide shows the effects of the different requests.

Database Table Descriptions

- FND_CONC_REQUEST_ARGUMENTS: This table records arguments passed by the concurrent manager to each program it starts running.
- FND_RUN_REQUESTS: For submitted report sets, this table stores information about the reports in the report set and the parameter values for each report.
- FND_CONCURRENT_REQUESTS: This table contains a complete history of all concurrent requests.
- FND_DUAL: This table records when requests do not update database tables.
- FND_CONCURRENT_PROCESSES: This table records information about Oracle Applications and operating system processes.

Practice - Scheduling Requests

Overview

The implementation committee has asked you to test a number of scenarios for scheduling concurrent requests. The scenarios they would like to test are as follows.

- Submit a concurrent request to run once, immediately
- Submit a concurrent request to run at 1 minute intervals
- Submit a concurrent request to run in 1 week
- Submit a concurrent request to run every day at 2 pm for just 1 week

Tasks

Submit a Request to Run Once

1. Submit the “Active Responsibilities” report to run immediately this one time, and to not run again.

Submit a Request to at 1 Minute Intervals

2. Submit the “Active Users” report to run 1 minute after the completion of the previous “Active Users” request is completed.

Submit a Request to Run in 1 Week

3. Submit the “Reports and Sets by Responsibility” report to run 1 week from today.

Submit a Request to Run Every Day at 2pm for just 1 Week

4. Submit the “Work Shifts” report to every day at 2pm for the next 1 week.

View your Concurrent Requests

5. After you have submitted all of your concurrent requests, take a look at the status of your requests.

Solution - Scheduling Requests

Submit a Request to Run Once

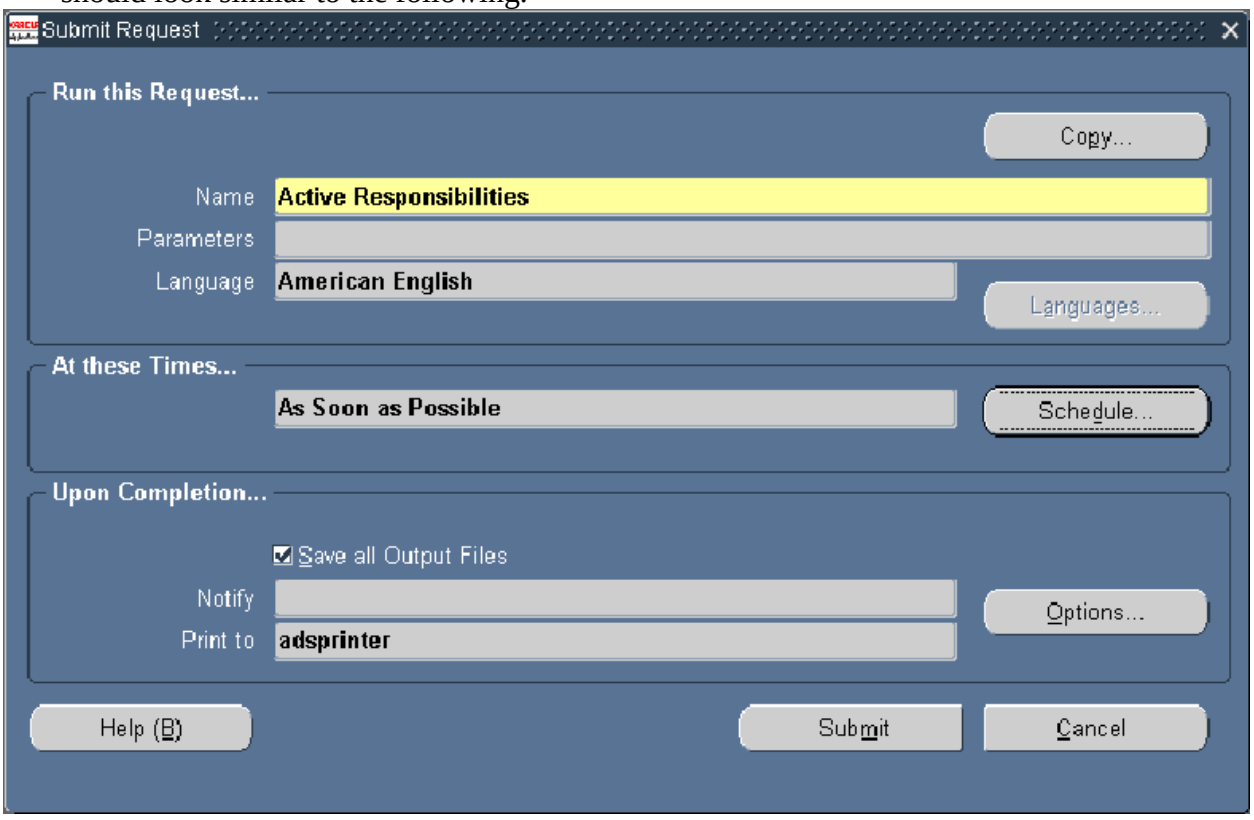
Responsibility = System Administrator

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Requests > Run.

3. Click OK to accept the default “Single Request.”

4. Select “Active Responsibilities” report from the list of values for the Name field. Your form should look similar to the following.



5. Click the “Submit” button.

6. Note the Request ID displayed in the Decision window.

7. Click the “Yes” button to submit another request.

Submit a Request to at 1 Minute Intervals

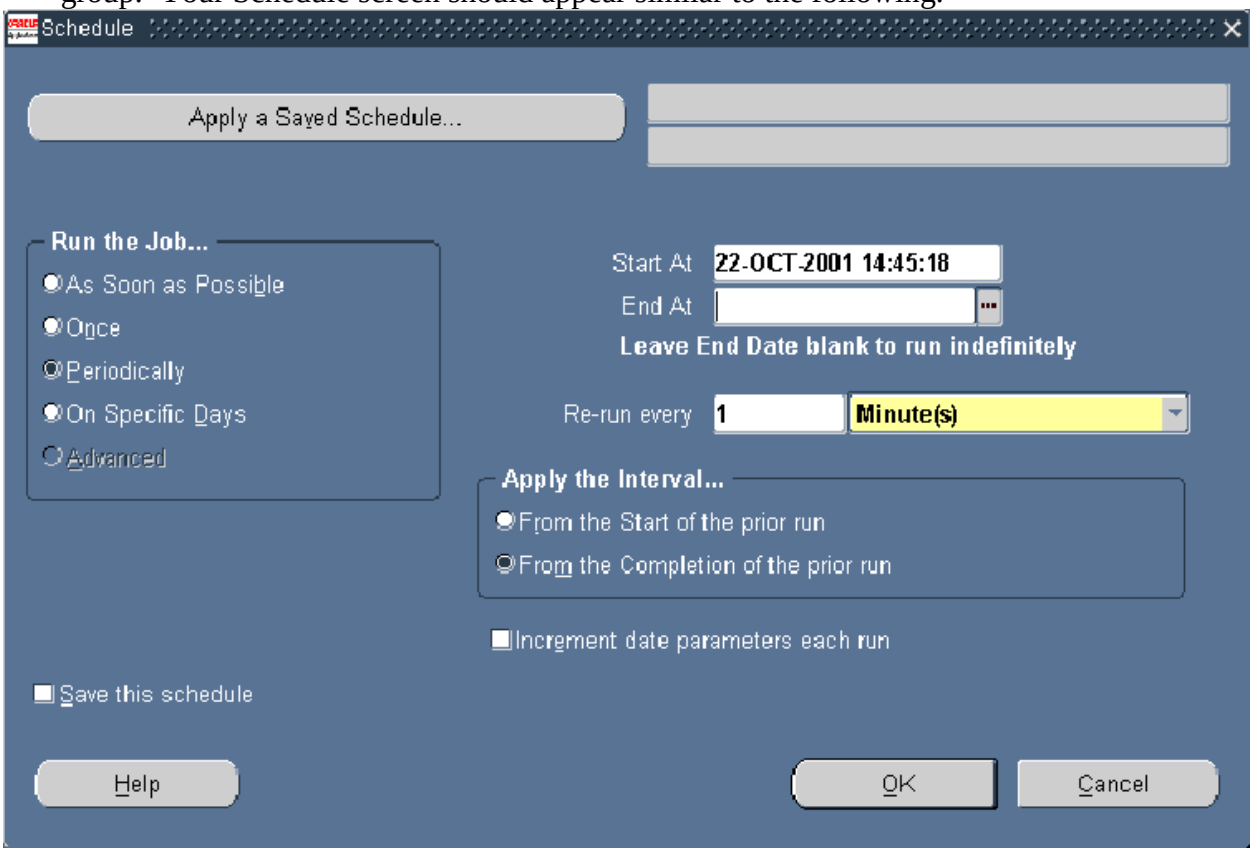
8. Select “Active Users” report from the list of values for the Name field.

9. Click the “Schedule…” button.

10. Select “Periodically” from the “Run this Job…” option group.

11. In the “Re-run every” fields, enter 1 and select Minute(s) from the list of values.

12. Select “From the Completion of the prior run” from the “Apply the Interval…” option group. Your Schedule screen should appear similar to the following.



13. Click the “OK” button.

14. Click the “Submit” button.

15. Note the Request ID displayed in the Decision window.

16. Click the “Yes” button to submit another request.

Submit a Request to Run in 1 Week

17. Select “Reports and Sets by Responsibility” report from the list of values for the Name field.

18. You will be prompted for parameters. Select “Application Object Library” from the “Application” list of values.

19. Click the “OK” button to close the Parameters window.

20. Click the “Schedule…” button.

21. Select “Once” from the “Run this Job…” option group.

22. In the “Run At” field change the date to one week from today by using the popup calendar or by typing in the date.

23. Click the “OK” button.

24. Click the “Submit” button.

25. Note the Request ID displayed in the Decision window.

26. Click the “Yes” button to submit another request.

Submit a Request to Run Every Day at 2pm for just 1 Week

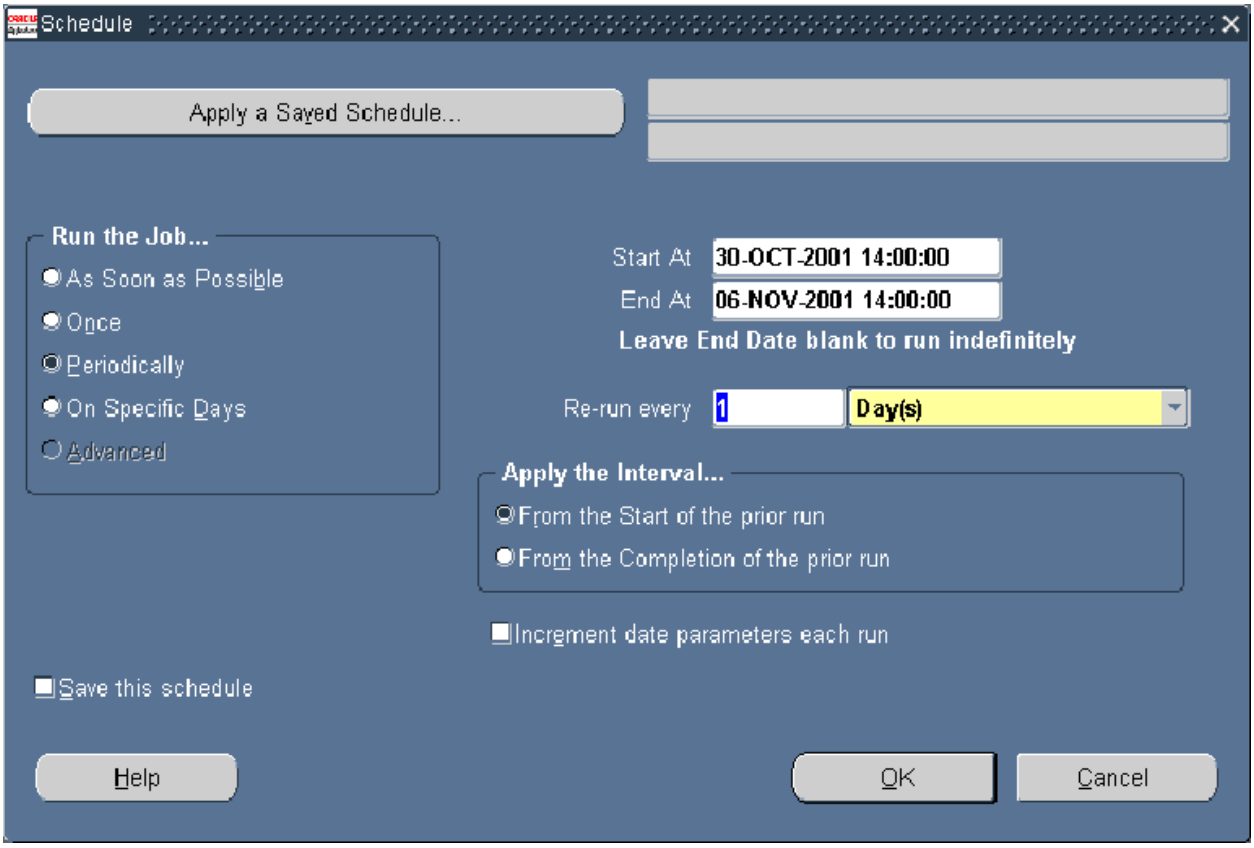
27. Select “Work Shifts” report from the list of values for the Name field.

28. Click the “Schedule…” button.

29. Select “Periodically” from the “Run this Job…” option group.

30. In the “Start At” field keep the current date, but change the time to 2:00 PM (14:00:00) by using the popup calendar or by typing it in.

31. In the “End At” field change the date to one week from today and the time to 2:00 PM (14:00:00) by using the popup calendar or by typing it in. Your form should look similar to the following.



32. Click the “OK” button.

33. Click the “Submit” button.

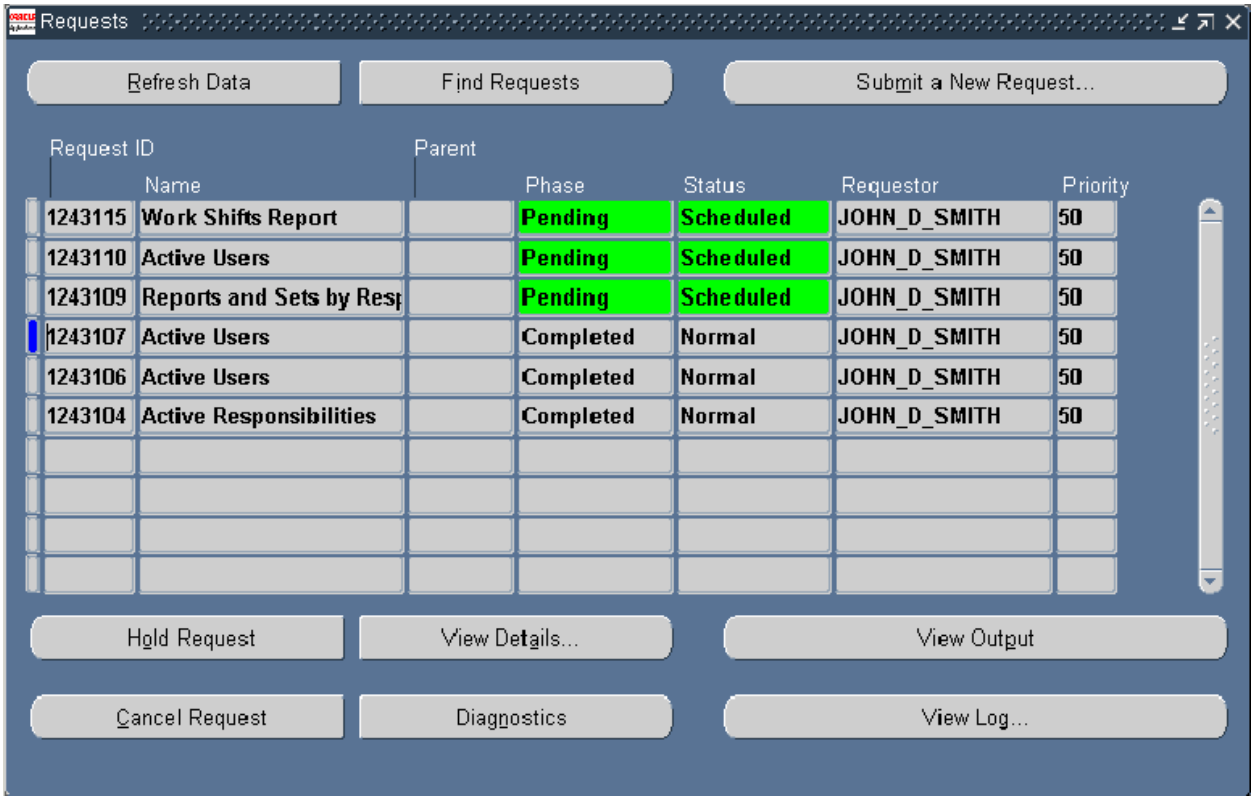
34. Note the Request ID displayed in the Decision window.

35. Click the “No” button to finish submitted requests.

View your Concurrent Requests

36. Navigate to (N) Requests > View.

37. Click the “Find” button to view all your requests. Your form should look similar to the following.



38. Use the appropriate buttons on this form to perform each of the tasks.

    - Find the "Active Responsibilities" request. The Phase should be Completed and the Status should be Normal.

    - Select a report with Phase Completed and Status Normal (for example, the Active Responsibilities report). Click the "Diagnostics" button. The Request Diagnostics window will appear. Review the entries and click the "OK" button to close the window.

    - Select a report with Phase Completed and Status Normal (for example, the Active Users report). Click the "View Log…" button. The log file will appear in a separate browser window. Review the entries for the log and close the browser window.

    - Select a report with Phase Completed and Status Normal (for example, the Active Users report). Click the "View Output" button. The report will appear in a separate browser window. Review the report and close the browser window.

Optional Additional Tasks

1. Cancel the Active Users request.

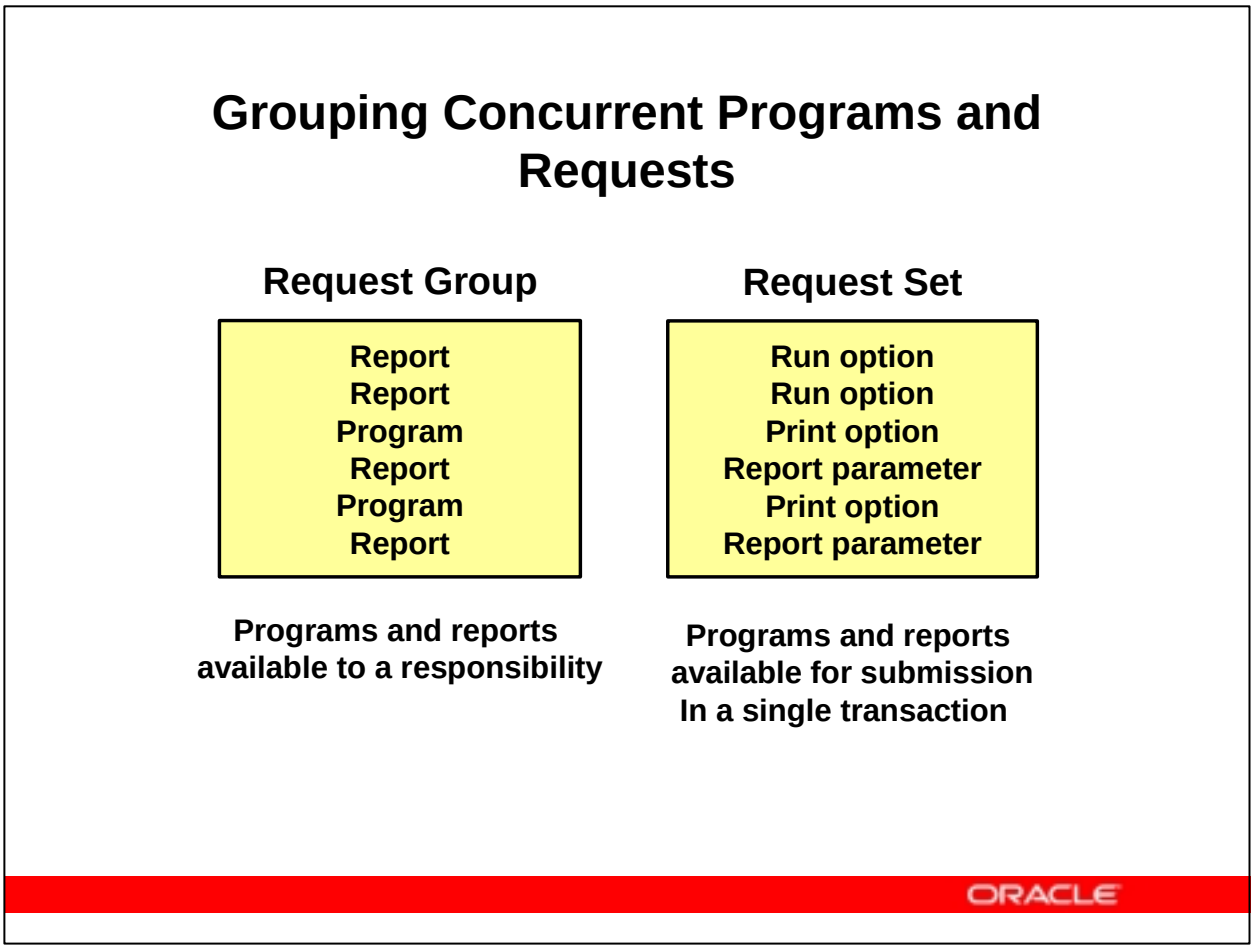
    - Click the "Refresh Data" button to verify that your display is current.

    - Select the "Active Users" request that is Running.

    - Click the "Cancel Request" button.

- A Decision window will warn you: “Cancelling a request cannot be undone. Continue?”
- Click the “Yes” button.
- The Phase will be updated to Completed and the Status will be set to Cancelled.

2. Put the “Reports and Sets by Responsibility” request on hold.
   - Select the “Reports and Sets by Responsibility” request.
   - Click the “Hold Request” button. The Phase of the request will be updated to Inactive and the Status will be set to On Hold.
   - Click the “Remove Hold” button to take the request off hold.
3. Reprint the Active Users report.
   - Select one of the “Active Users” requests that completed with a normal status.
   - From the Tools menu select the “Reprint…” item.
   - With 0 copies selected, click the “OK” button.



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.

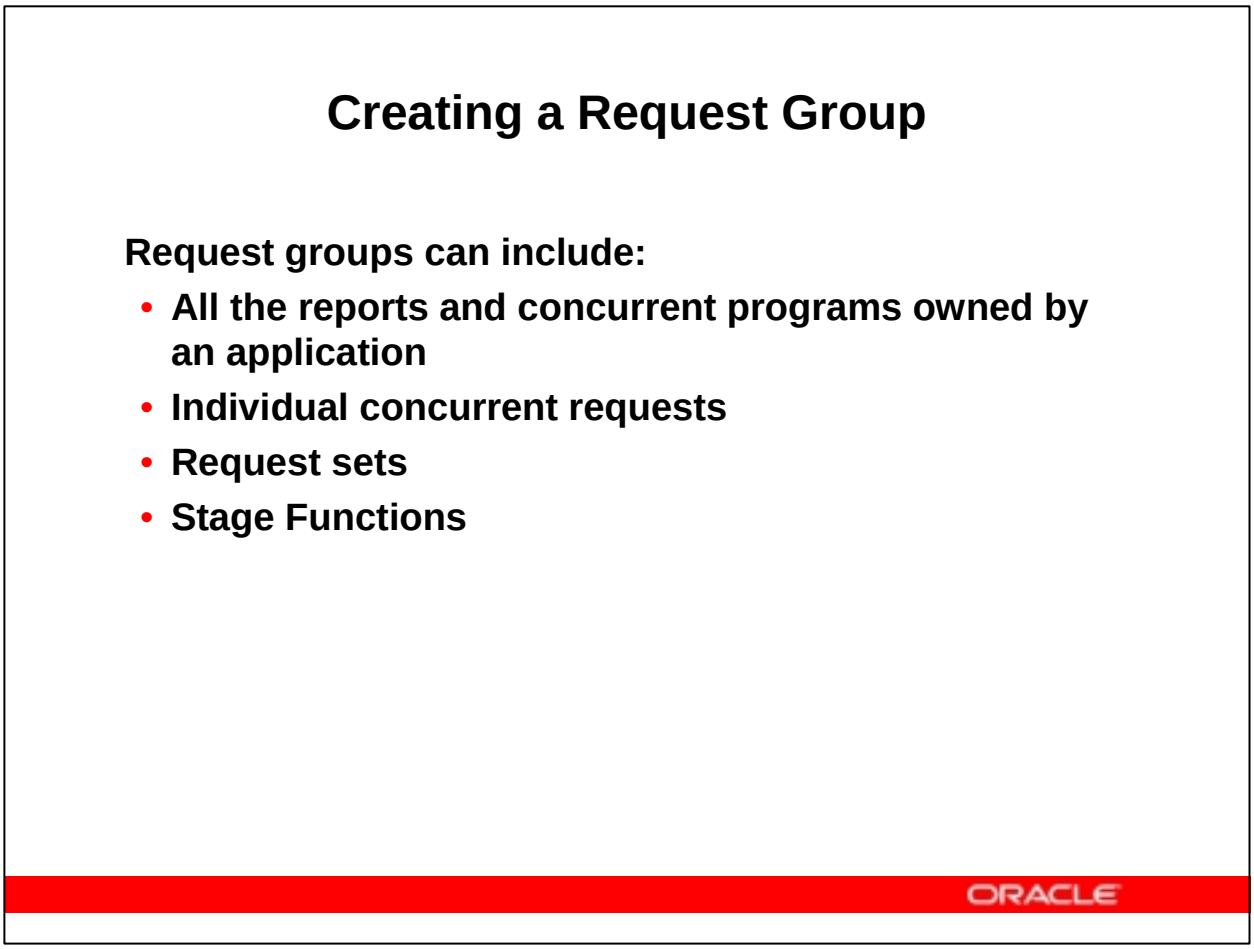
Request Groups

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

A request set is a collection of concurrent programs set up to run in a specified sequence from a single transaction. 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 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.

Practice - Request Groups

Overview

Up until this test process, the committee has not wanted you to deal with Request Groups. In this test process, you will create a new Request Group, and assign it appropriately. The steps that will be tested include the following.

- Create a new Request Group
- Limit the Request Groups access to concurrent reports, programs, and sets
- Assign the new Request Group to your *Your Initials* Assistant System Administrator (e.g., WHS Assistant System Administrator) responsibility
- Test your Request Group

Tasks

Create your Request Group.

1. Create a new Request. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Group Name**: *Your Initials* Assistant SysAdmin Group
   - **Application**: Application Object Library
   - **Code**: (leave blank)
   - **Description**: *Your Initials* Assistant SysAdmin Group
   - **Requests:**

Type	**Name**	**Application**
Application	Application Object Library	Application Object Library
Program	Employee Listing	Oracle Payables
Set	Period End	Oracle Payables

Assign your Request Group.

2. Assign the new Request Group to your Assistant System Administrator Responsibility.

Test your Request Group.

3. Verify your work by selecting the Assistant System Administrator Responsibility and viewing the LOV for both Single Request and Request Set.

Solution - Request Groups

Create your Request Group.

Responsibility = System Administrator

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

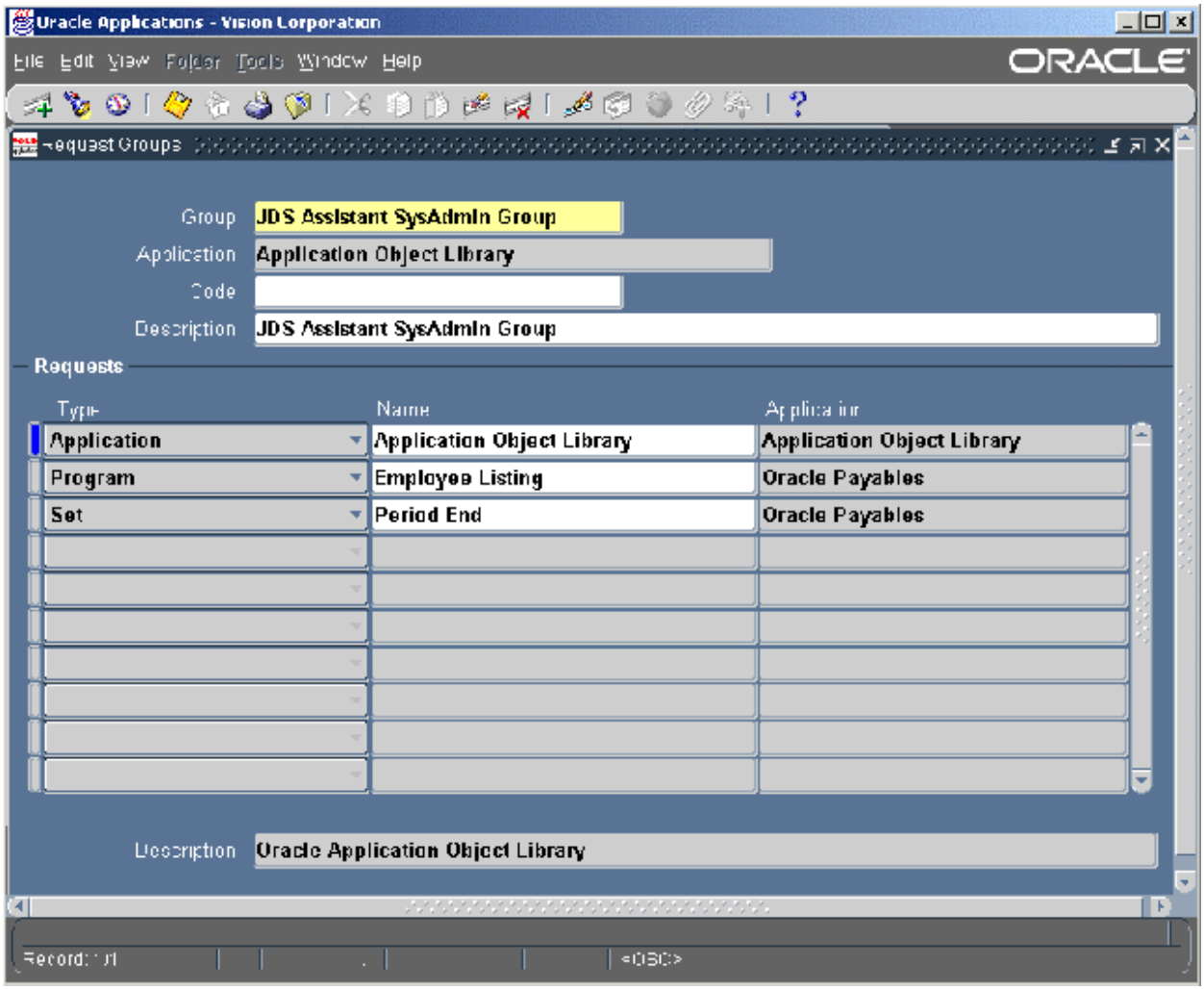
2. Navigate to (N) Security > Responsibility > Request.

3. Create a new Request Group.

    - **Group Name**: *Your Initials* Assistant SysAdmin Group
    - **Application**: Application Object Library
    - **Code**: (leave blank)
    - **Description**: *Your Initials* Assistant SysAdmin Group
    - **Requests:**

Type	**Name**	**Application**
Application	Application Object Library	Application Object Library
Program	Employee Listing	Oracle Payables
Set	Period End	Oracle Payables

4. Save your work. Your work should look similar to the following.



5. Close the form.

Assign your Request Group.

6. Navigate to (N) Responsibility > Define.

7. Query to find your *Your Initials* Assistant System Administrator responsibility. Choose your request group from the list of values for the "Request Group Name" field.

8. Click the "Save" icon to save your work.

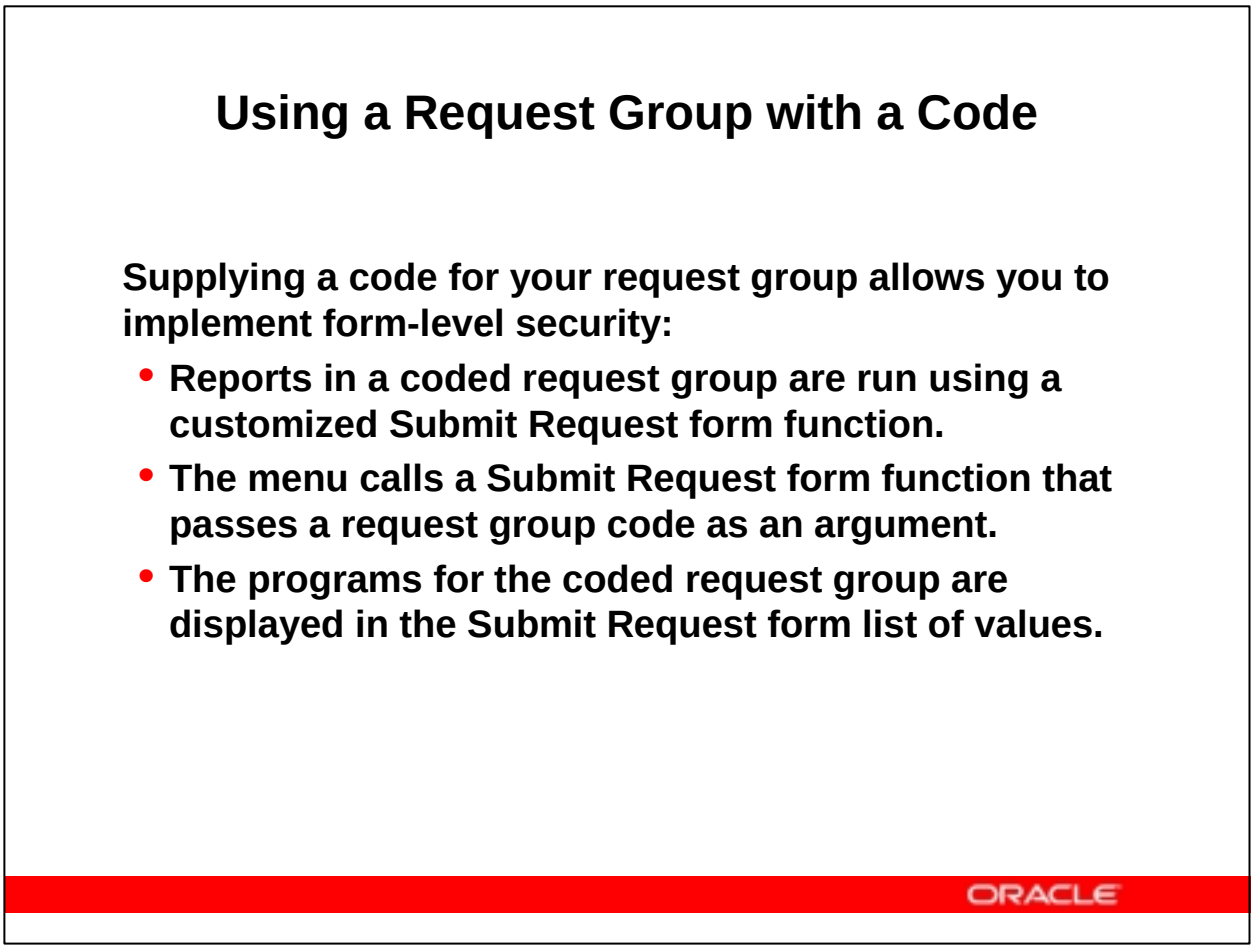
Test your Request Group.

9. Exit and sign on again as your *Your Initials* Assistant System Administrator.

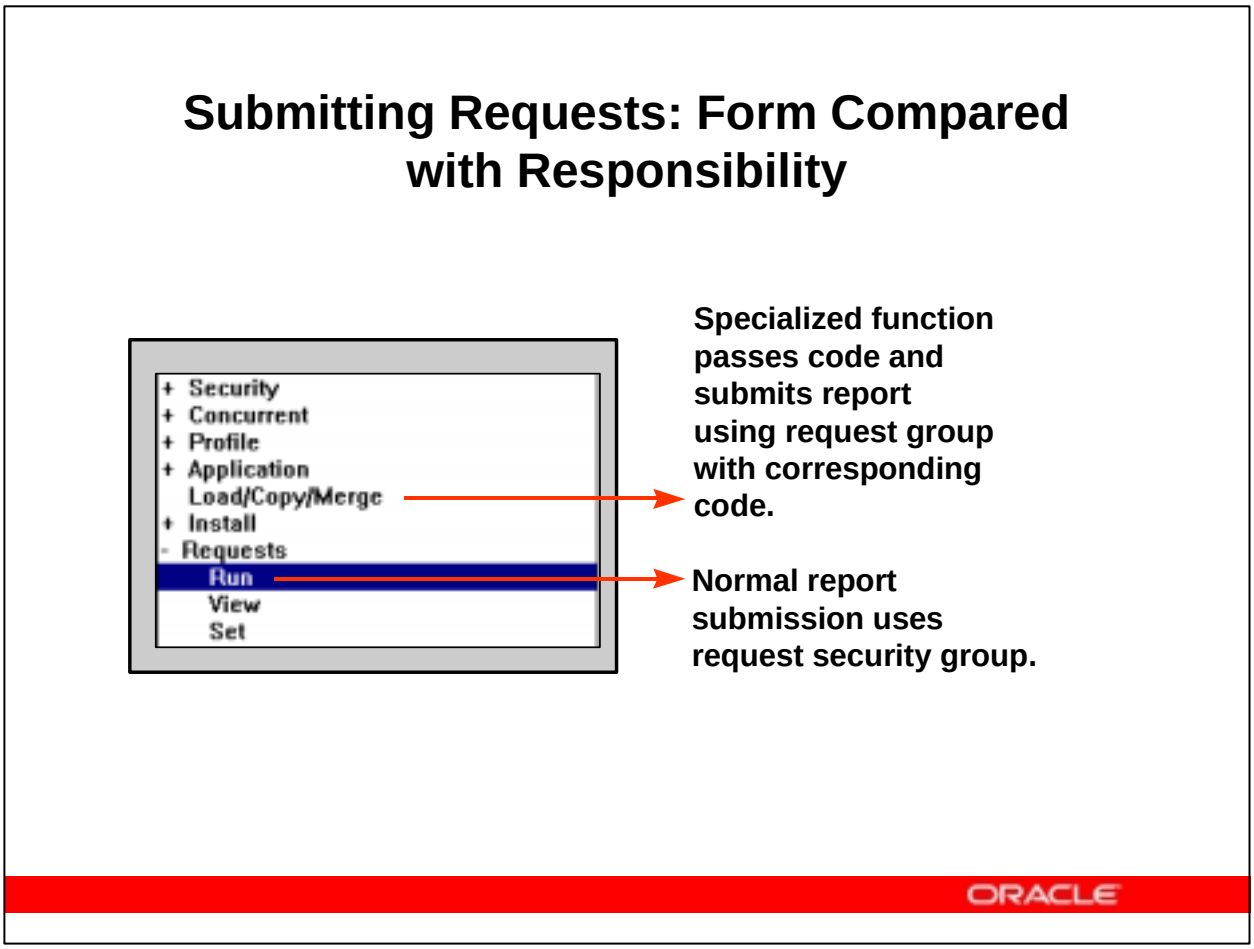
10. Navigate to (N) Requests > Run.

11. Click the "OK" button to accept the defaults.

12. Note the reports that now exist in the list of available reports for you to run.



The code must not contain any spaces. The combination of the application and the code identify the request group.



Submitting Requests: Form Versus Responsibility

The request processing is different depending on whether you are using a normal request group or a coded request group.

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.

Form-Based Access

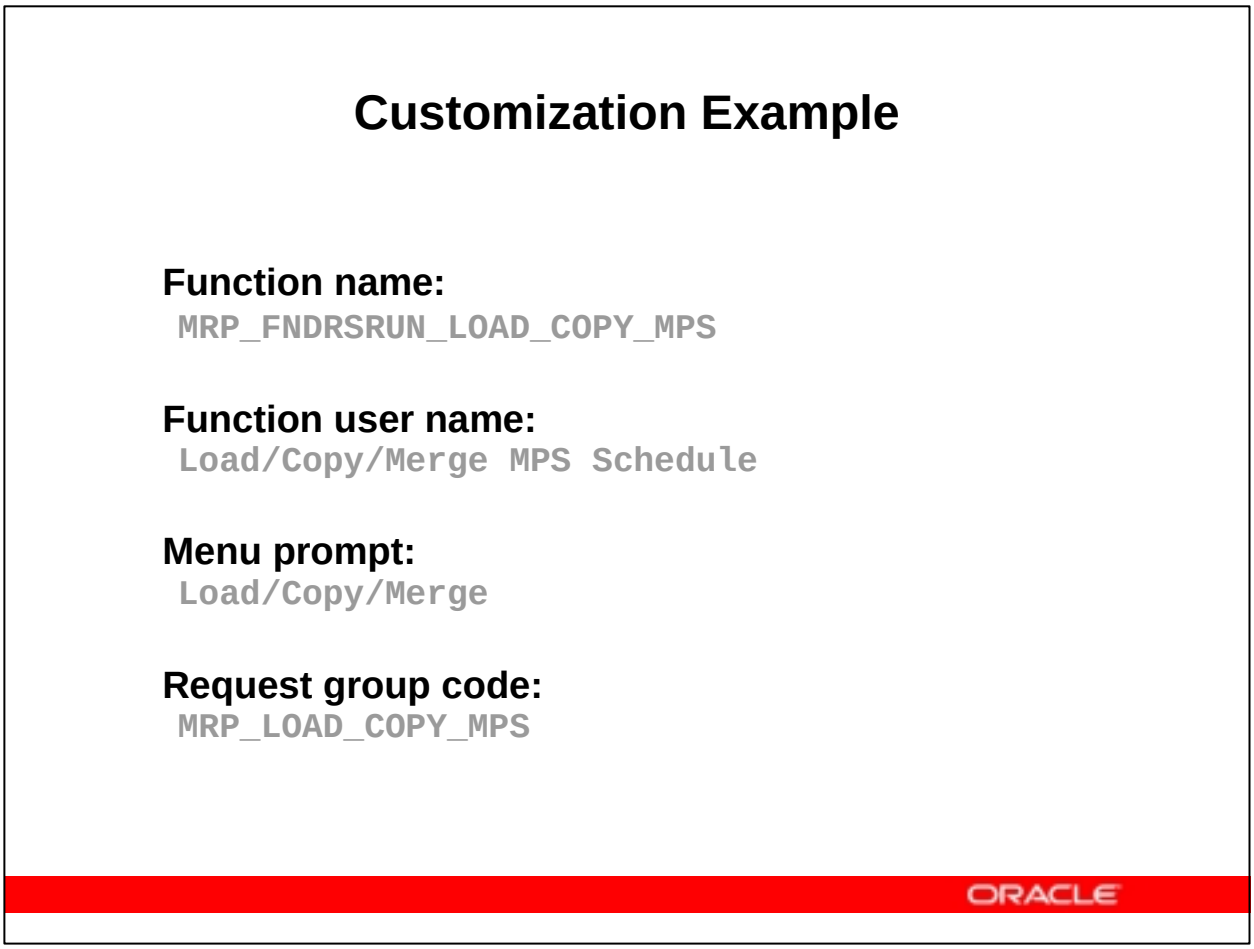
This type of access uses a coded request group. In this method, when a modified menu prompt is chosen, it calls a form function to pass an argument (the code) to the Submit Requests form.

The only programs displayed in the Submit Requests list of values are those defined to the coded request group.

Implementing a Coded Request Group

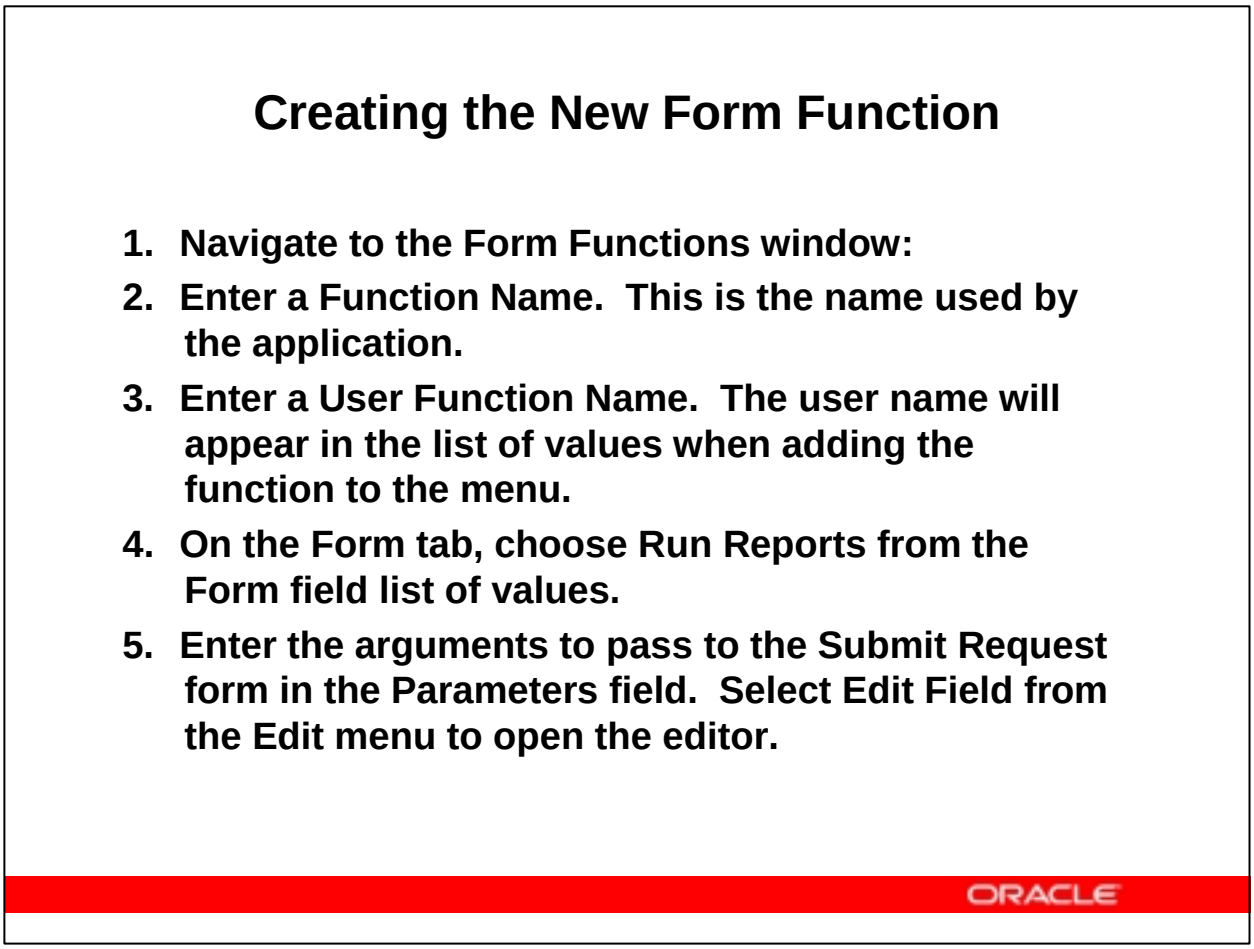
To implement a request group with a code:

1. **Create the request group supplying a code.**
2. **Create a new form function.**
3. **Add the new function to an existing menu.**



Customization Example

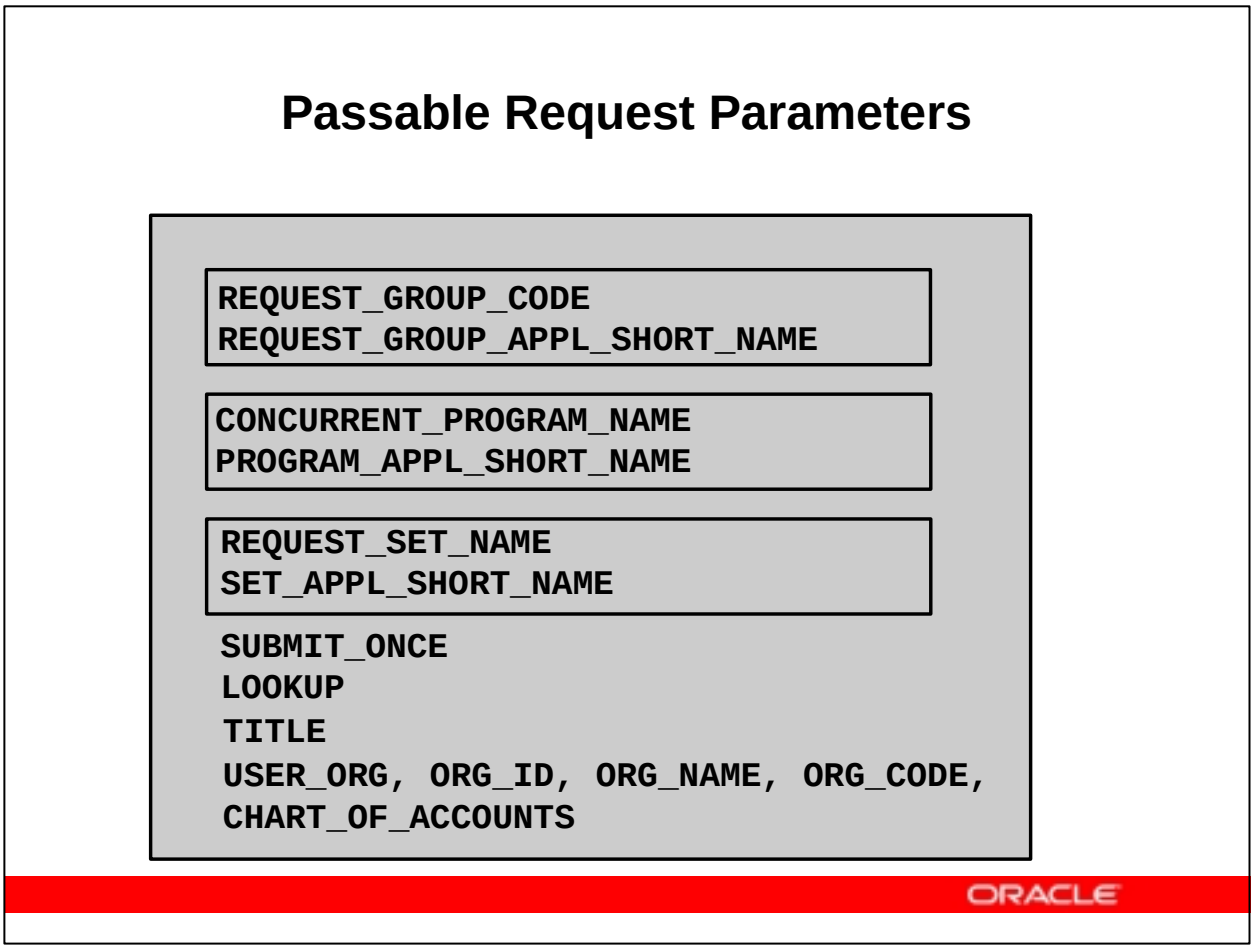
The slide shows the values that are used in the following example.



Creating the Form Function

(N) Application > Function

Type is a nonvalidated field used to enter a description of the function's use.



Passable Request Parameters

The slide shows the parameters you can pass in your new function. Several of the parameters are paired; these parameters must be used together. Remember that when the request group was created, it was associated with an application. You must pass both the code and the short name of the application for the system to identify the correct request group.

For a complete explanation of these parameters, see:

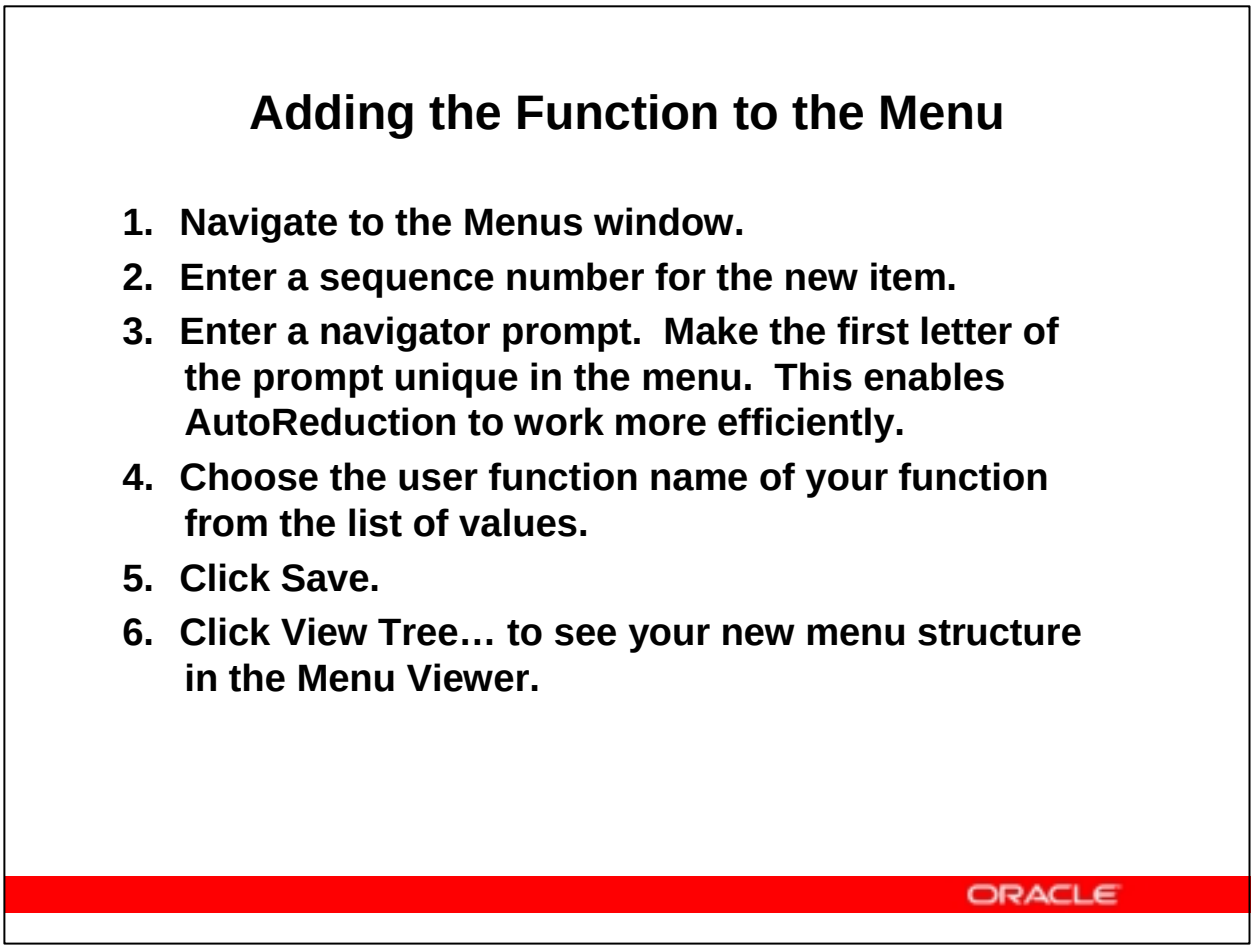
(Help) Applied Technology > Oracle Applications System Administration >

Overview of Concurrent Programs and Requests >

Organizing Programs into Request Groups >

Customizing the Submit Requests Window,

Customizing the Submit Requests Window using Codes.



(N) Application > Menu

Practice - Coded Request Groups

Overview

Now that you have completed the committee's tests on Request Groups, they want to go into the advanced testing of a Coded Request Group. Specifically, the committee wants to test the following.

- Create a Coded Request Group
- Create a new form function to call your Code Request Group
- Add the new form function to an existing menu
- Test your Coded Request Group

Tasks

Create your Coded Request Group

1. Create a new Coded Request Group. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Group Name**: *Your Initials* Journal Entries
   - **Application**: Application Object Library
   - **Code**: *Your Initials*JE (e.g., WHSJE)
   - **Description**: *Your Initials* Journal Entries
   - **Request Type**: Program
   - **Request Name**: Journal Entries Report
   - **Application**: Oracle Receivables

Create your New Form Function

1. Create a new Form Function. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Function**: *Your Initials*_FNDRSRUN_JOURNAL_ENTRIES
   - **User Function Name**: *Your Initials* Journal Entries
   - **Type**: Form

- **Description**: *Your Initials* Journal Entries
- **Form**: Run Reports
- **Parameters**: TITLE = "*Your Initials* Journal Entries"
- REQUEST_GROUP_CODE = "*Your Initials*JE"
- REQUEST_GROUP_APPL_SHORT_NAME = "FND"

Assign your Form Function to an Existing Menu

1. Query up your Level 1 custom menu, *Your Initials*_TOP_SYSTEM_AUDITOR
2. Add your function.
3. **Seq**: 50, **Prompt**: Journal Entries Report, **Function**: *Your Initials* Journal Entries, **Description**: *Your Initials* Journal Entries

Test your Coded Request Group

1. Test that your Code Request Group works as expected.

Solution - Coded Request Groups

Create your Coded Request Group

Responsibility = System Administrator

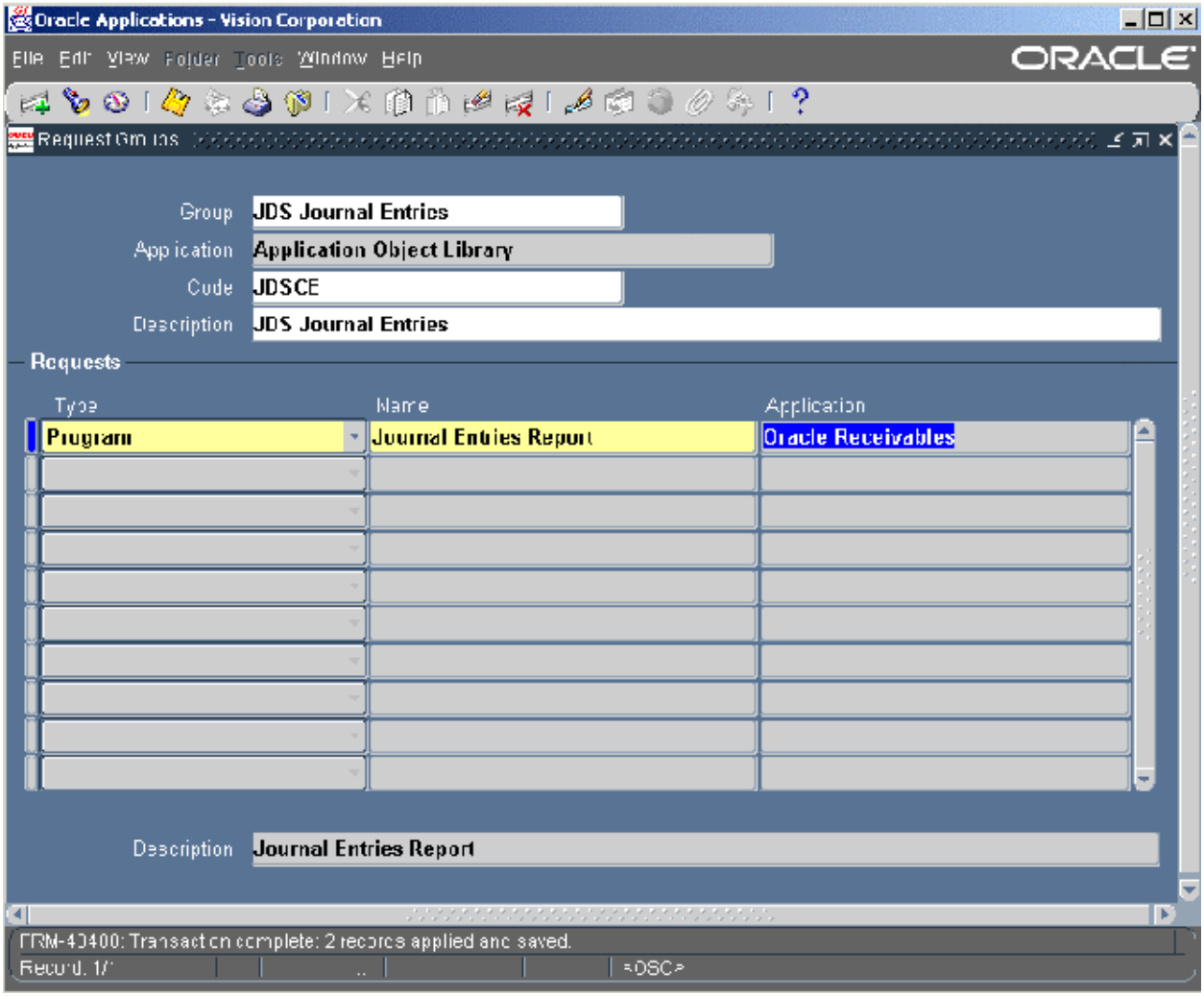
1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Security > Responsibility > Requests.

3. Create a new Coded Request Group.

   - **Group Name**: *Your Initials* Journal Entries
   - **Application**: Application Object Library
   - **Code**: *Your Initials*JE (e.g., WHSJE)
   - **Description**: *Your Initials* Journal Entries
   - **Request Type**: Program
   - **Request Name**: Journal Entries Report
   - **Application**: Oracle Receivables

4. Save your work. Your work should look similar to the following.



5. Close the form.

Create your New Form Function

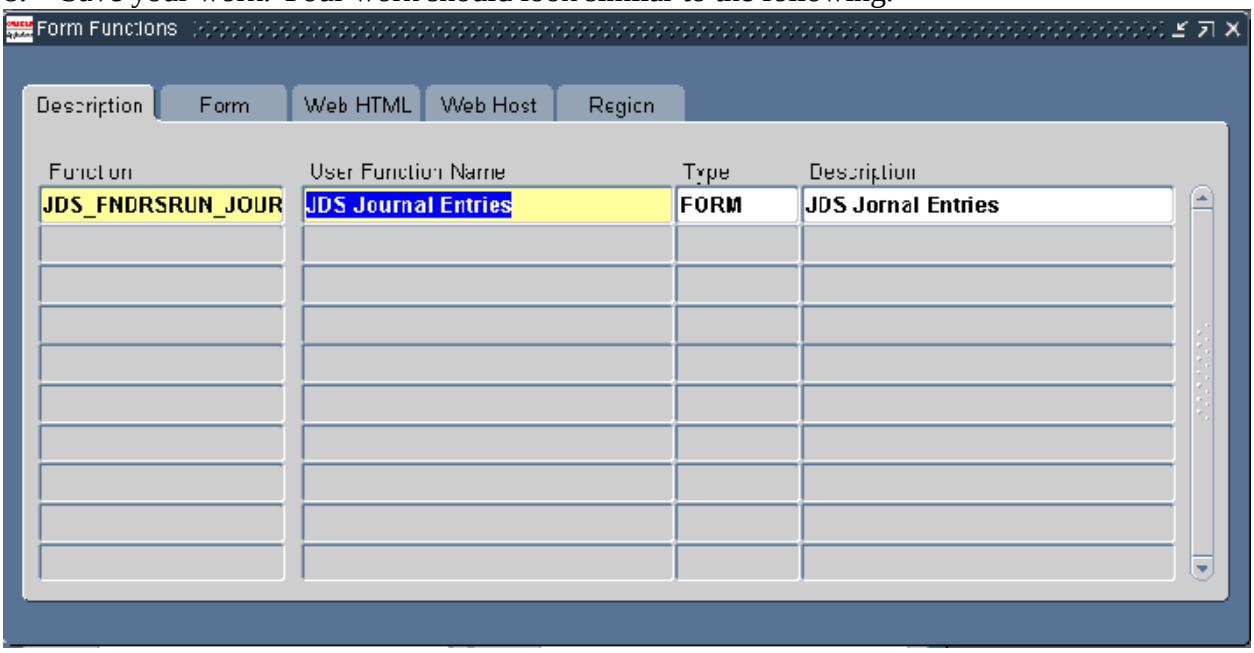
6. Navigate to (N) Application > Function.

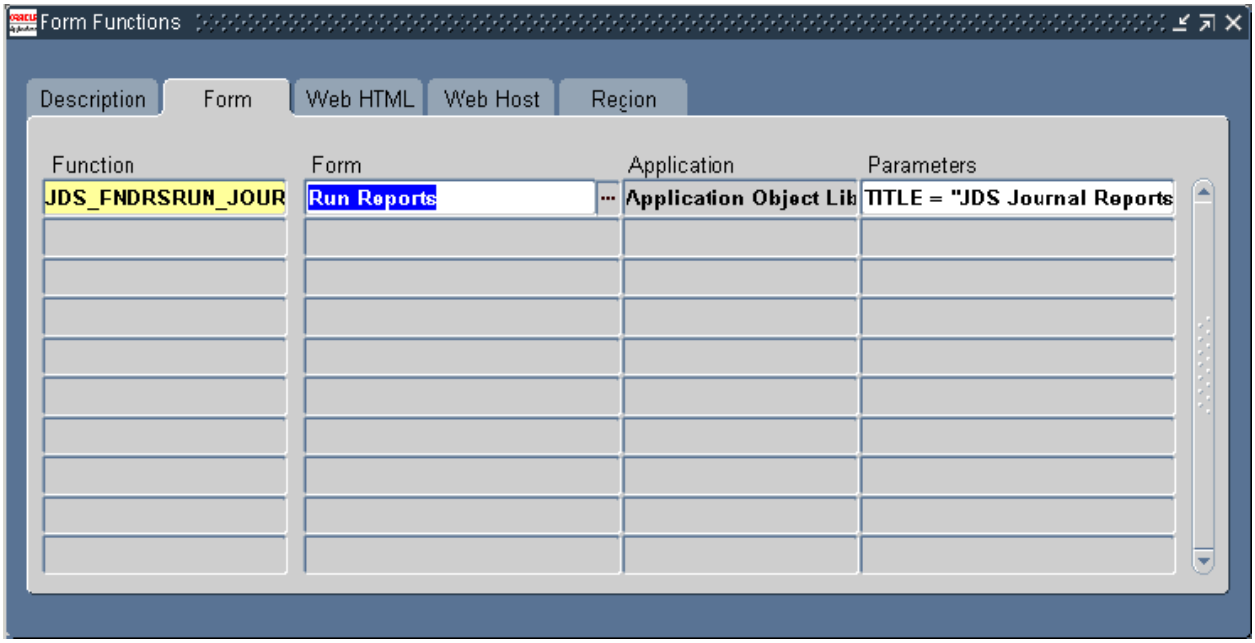
7. Create a new Form Function.

   - **Function**: *Your Initials*_FNDRSRUN_JOURNAL_ENTRIES
   - **User Function Name**: *Your Initials* Journal Entries
   - **Type**: Form
   - **Description**: *Your Initials* Journal Entries
   - **Form**: Run Reports
   - **Parameters**: TITLE = "*Your Initials* Journal Entries"

- REQUEST_GROUP_CODE = "*Your Initials*JE"
- REQUEST_GROUP_APPL_SHORT_NAME = "FND"

8. Save your work. Your work should look similar to the following.





9. Close the form.

Assign your Form Function to an Existing Menu

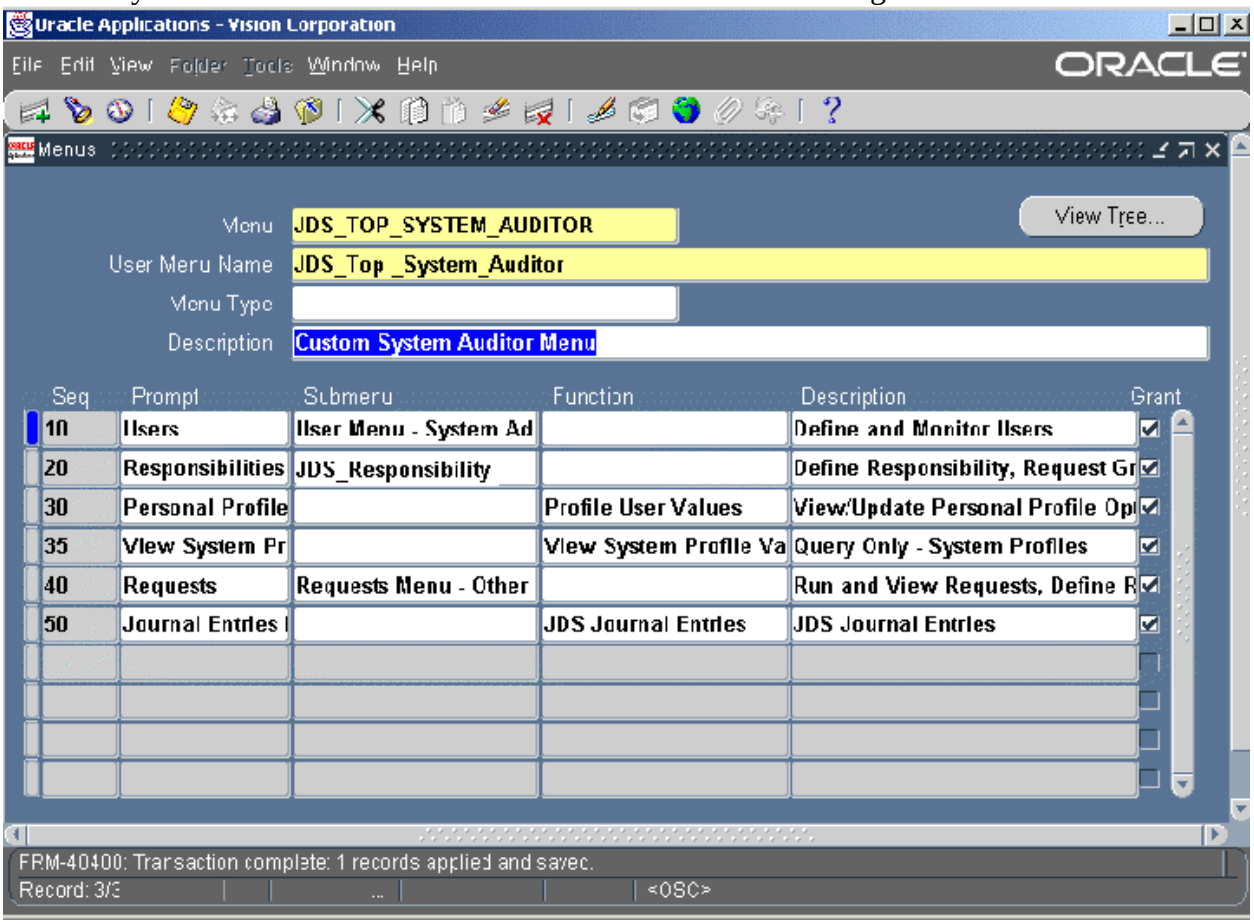
10. Navigate to (N) Application > Menu.

11. Query up your Level 1 custom menu, *Your Initials*_TOP_SYSTEM_AUDITOR

12. Add your function.

13. **Seq**: 50, **Prompt**: Journal Entries Report, **Function**: *Your Initials* Journal Entries, **Description**: *Your Initials* Journal Entries

14. Save your work. Your work should look similar to the following.



15. Close the form.

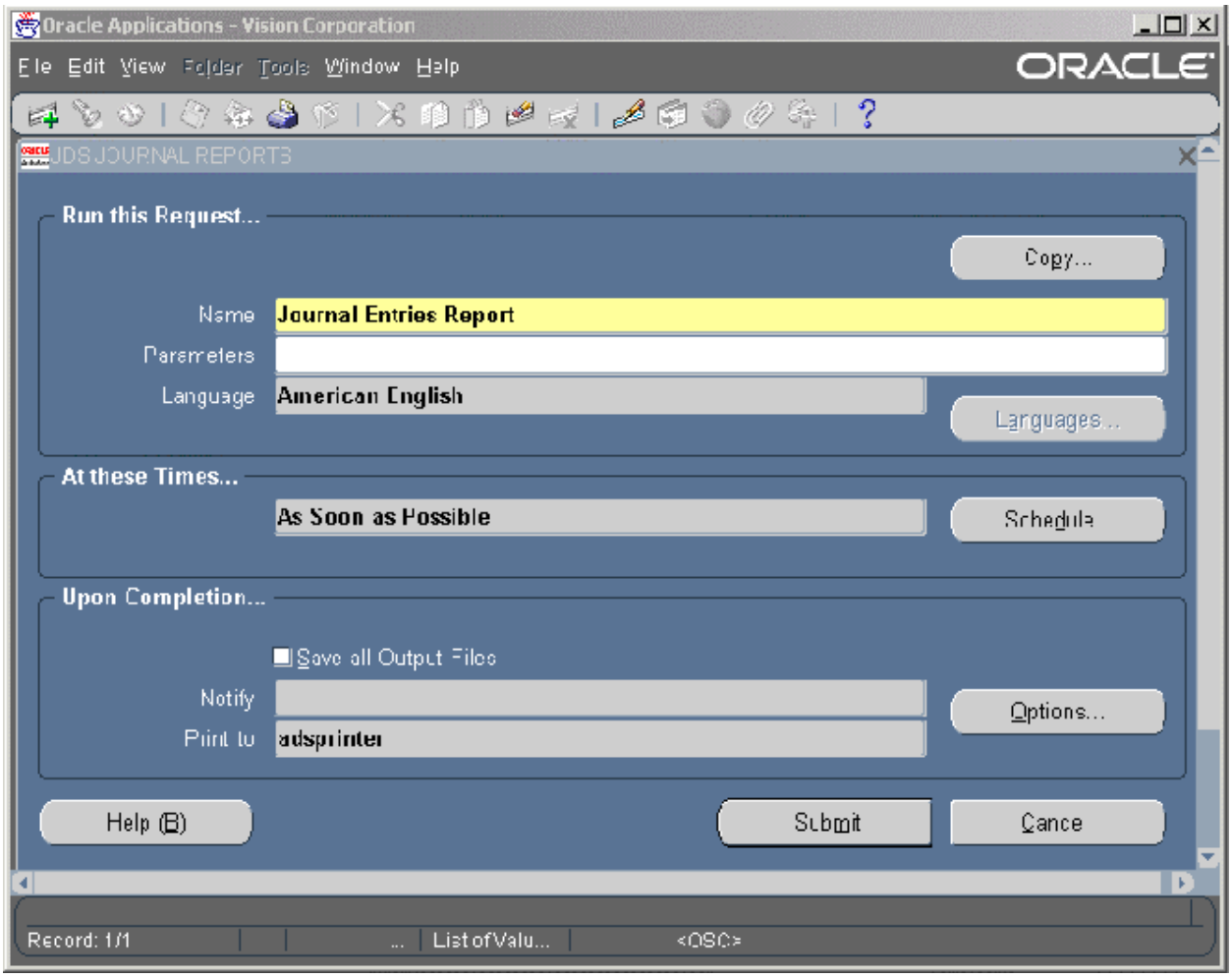
Test your Coded Request Group

16. Exit Oracle Applications and the Personal Homepage. Sign on again as the System Auditor (i.e., *Your Initials*_SYSTEM_AUDITOR) to see your new function on the menu. Your request group Journal Entries Report will appear on the Navigator for this responsibility.

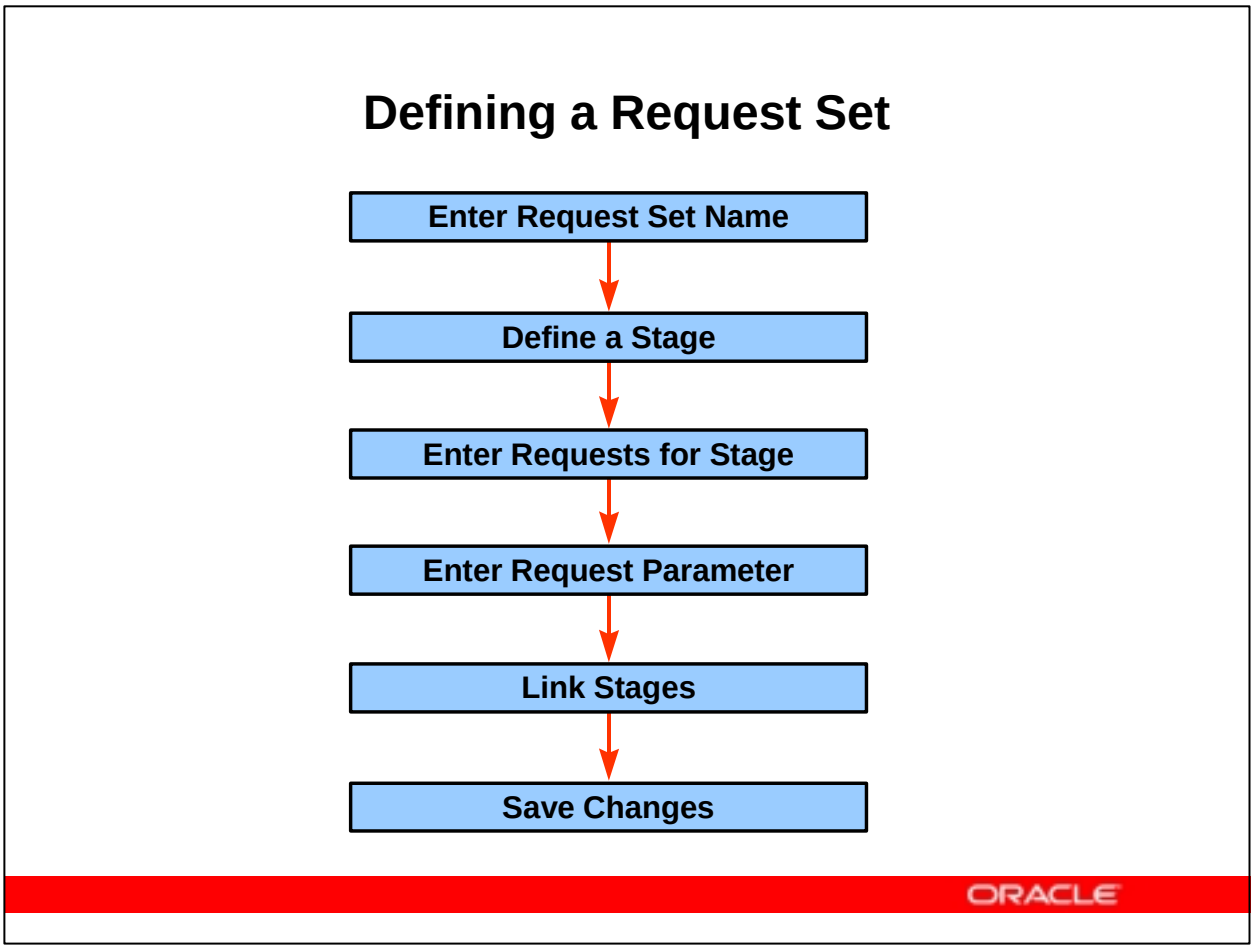
17. Navigate to (N) Journal Entries Report.

18. Click the “OK” button to accept “Single Request.”

19. When the submit request form appears, note the window title – it will be the value entered for your parameter TITLE. The title of the form defaults to the request title because it is the only request on the list.



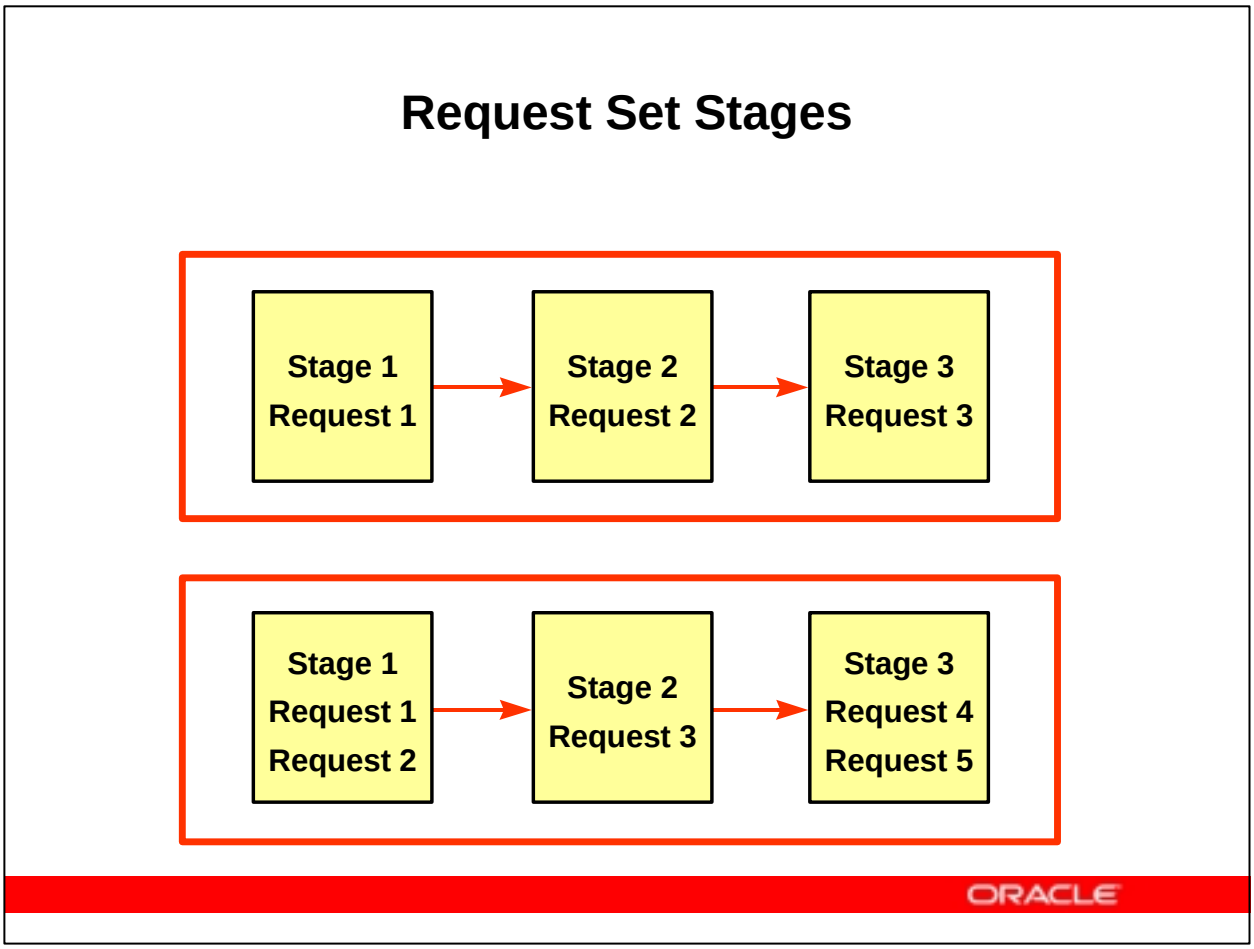
Defining a Request Set



Defining Request Sets

By defining request sets, you can submit the same set of requests regularly with a single transaction. Use the Request Set window to create and edit request sets. You can also use the Request Set Wizard to create simple request sets.

Request Set Stages



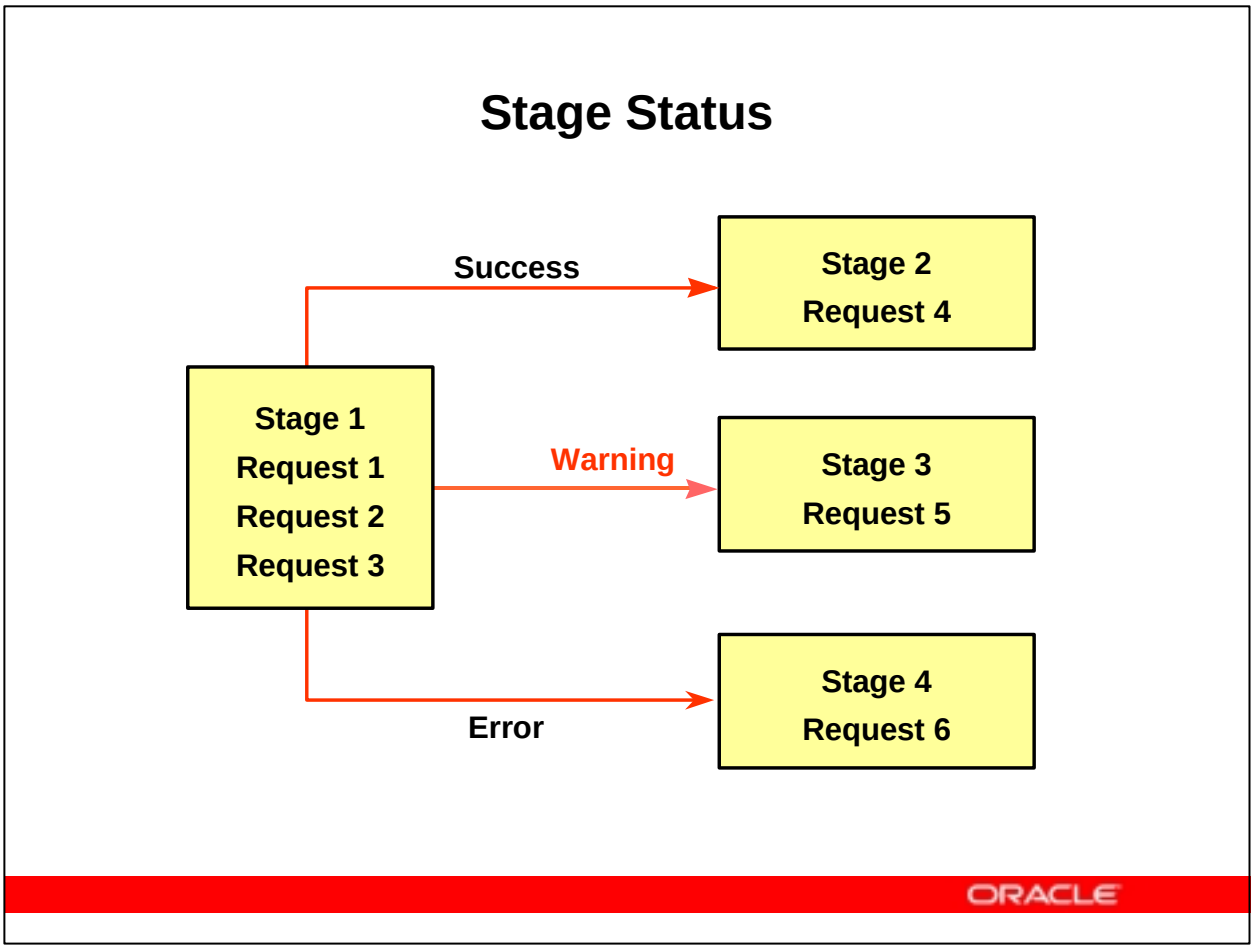
Organizing Requests with Stages

Request sets are divided into one or more *stages*, which are linked to determine the sequence in which the requests will 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.

Stage Status

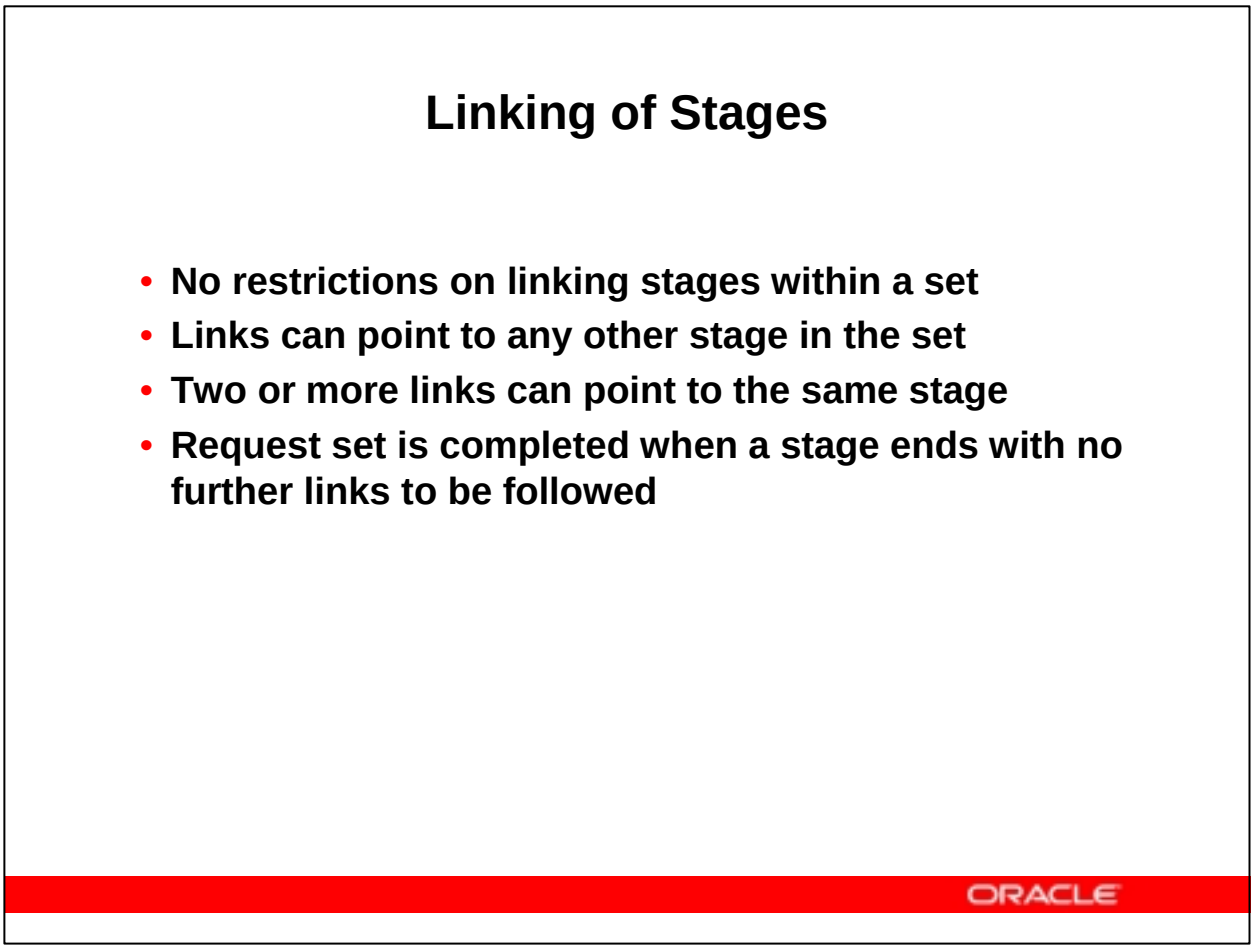


Using Stage Status

Like request sets and concurrent requests, stages can complete with different statuses. Each stage can complete with a status of Success, Warning, or Error. You can use these completion statuses to structure your request set by defining which stage will follow the current stage based on its completion status.

The request set in the slide always starts with Stage 1. If Stage 1 were to complete with the status of Warning, then the Warning link would be followed and Stage 3 would be submitted.

The status of a completed stage is set according to the highest error status returned by the individual requests comprising the stage. Therefore, in order for the stage to complete with a status of Success, all the requests that make up the stage must have completed with a status of Success. If one request completes with a status of Error, the status for the stage will be Error, regardless of the statuses of the other requests.



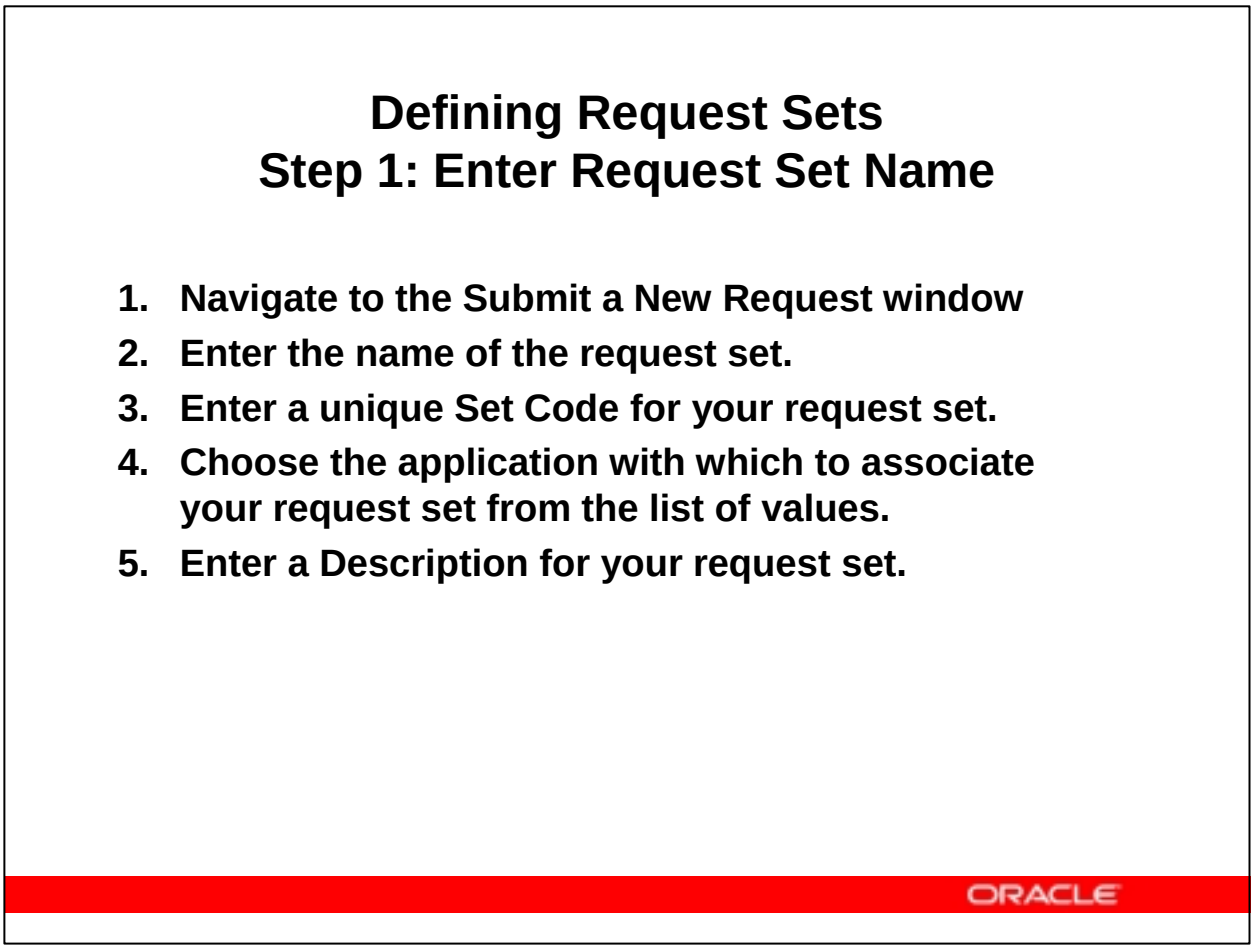
Linking of Stages

There are no restrictions on linking stages within a request set. Any stage can be linked to any other stage, including itself. Two or more links can point to the same stage. For example, one stage can link to another if the completion status of the first is Success or Warning, and link to a third stage if the status is Error.

You determine the end of a request by not specifying a follow-up stage for each completion status. You can end a request set after any stage in the set. When any stage completes with a status that does not link to another stage, the request set ends.

You can use the linking of stages to control your request set. By using the Request Set Wizard button in the Request Set window you can create the request set as follows:

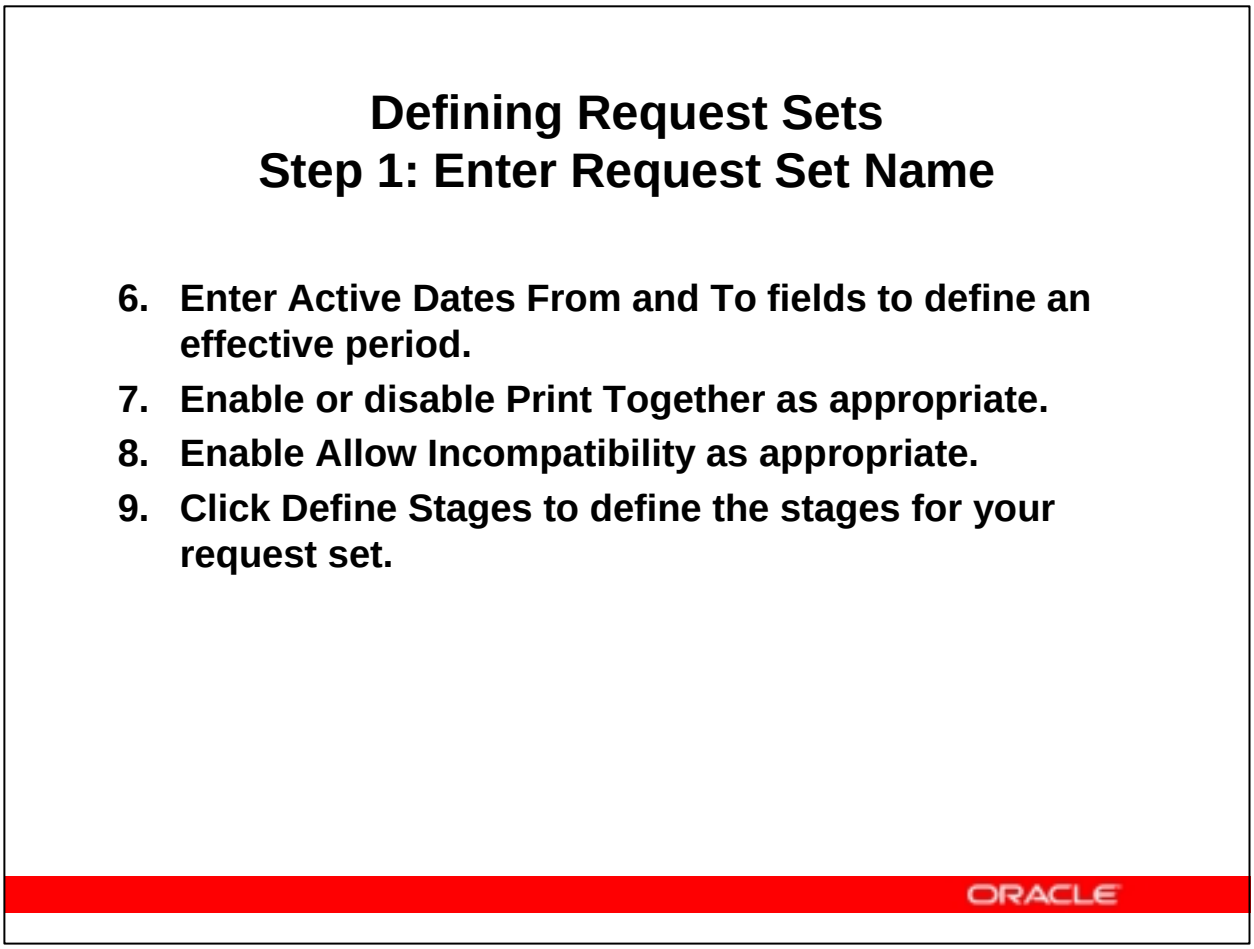
- Run in Parallel – creates one stage containing all of the requests you wish to run in parallel.
- Run Sequentially – creates a separate stage containing the request or requests for each step in the sequence and link in the appropriate order.
- Run Sequentially but abort on Error – sets up your sequence the same as it did for Run Sequentially, but when it links the stages, it does not enter a follow-up stage as a link in the Error completion status field.



Defining A Request Set

(N) Requests > Set

The Owner field defaults to your user name and can be changed only by your system administrator.

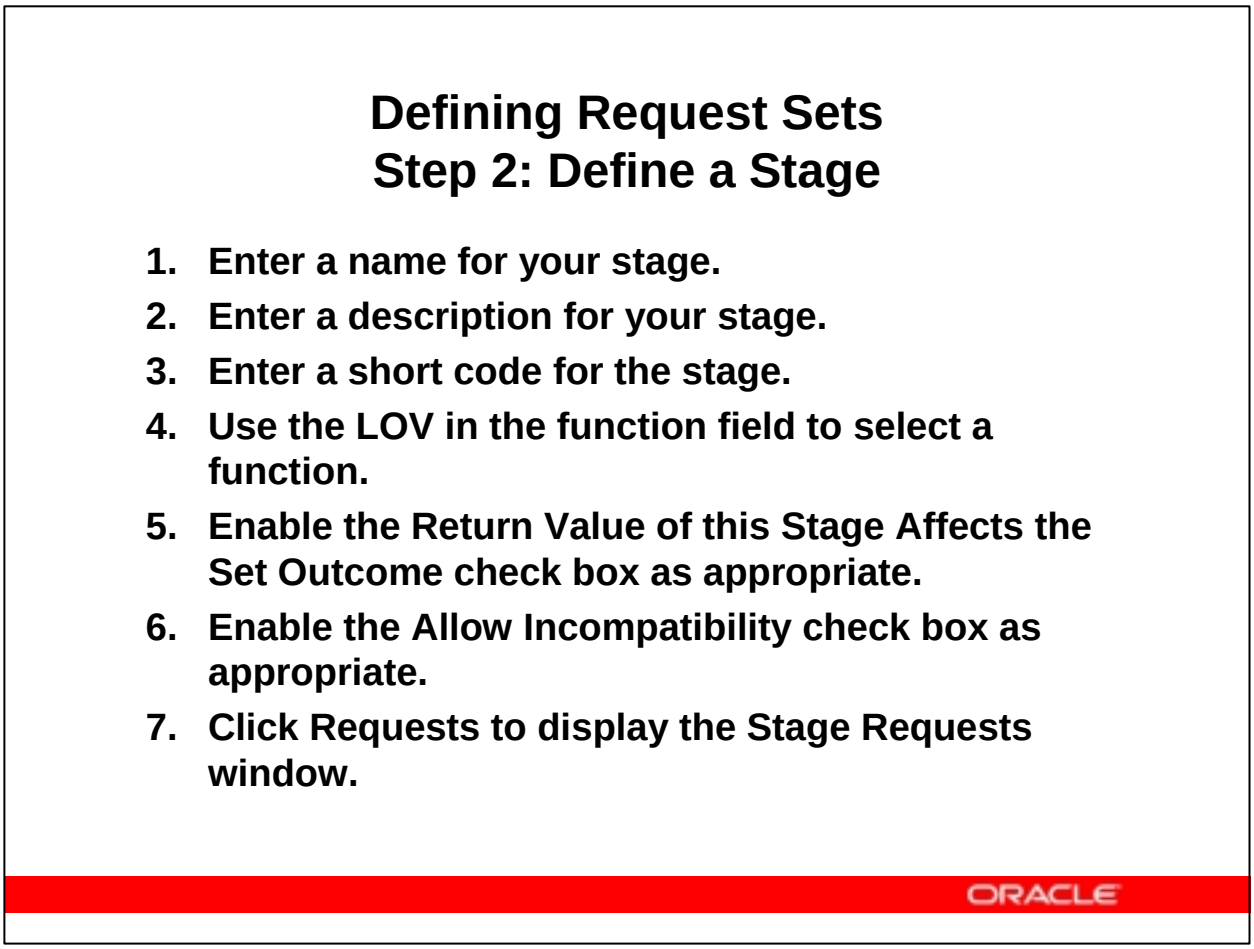


Defining A Request Set (continued)

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.



Stages

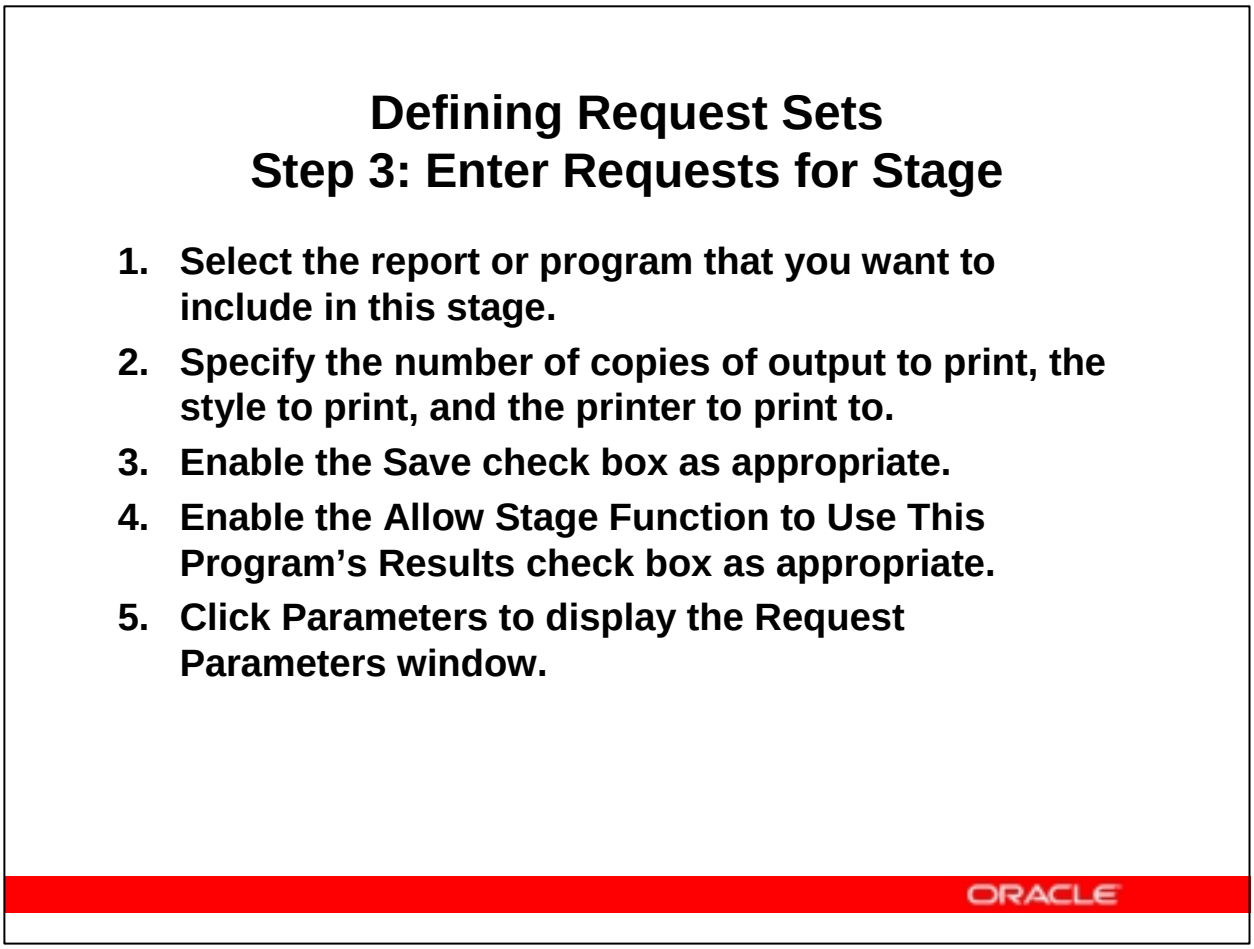
(N) Requests > Set > (B) Define Stages

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.

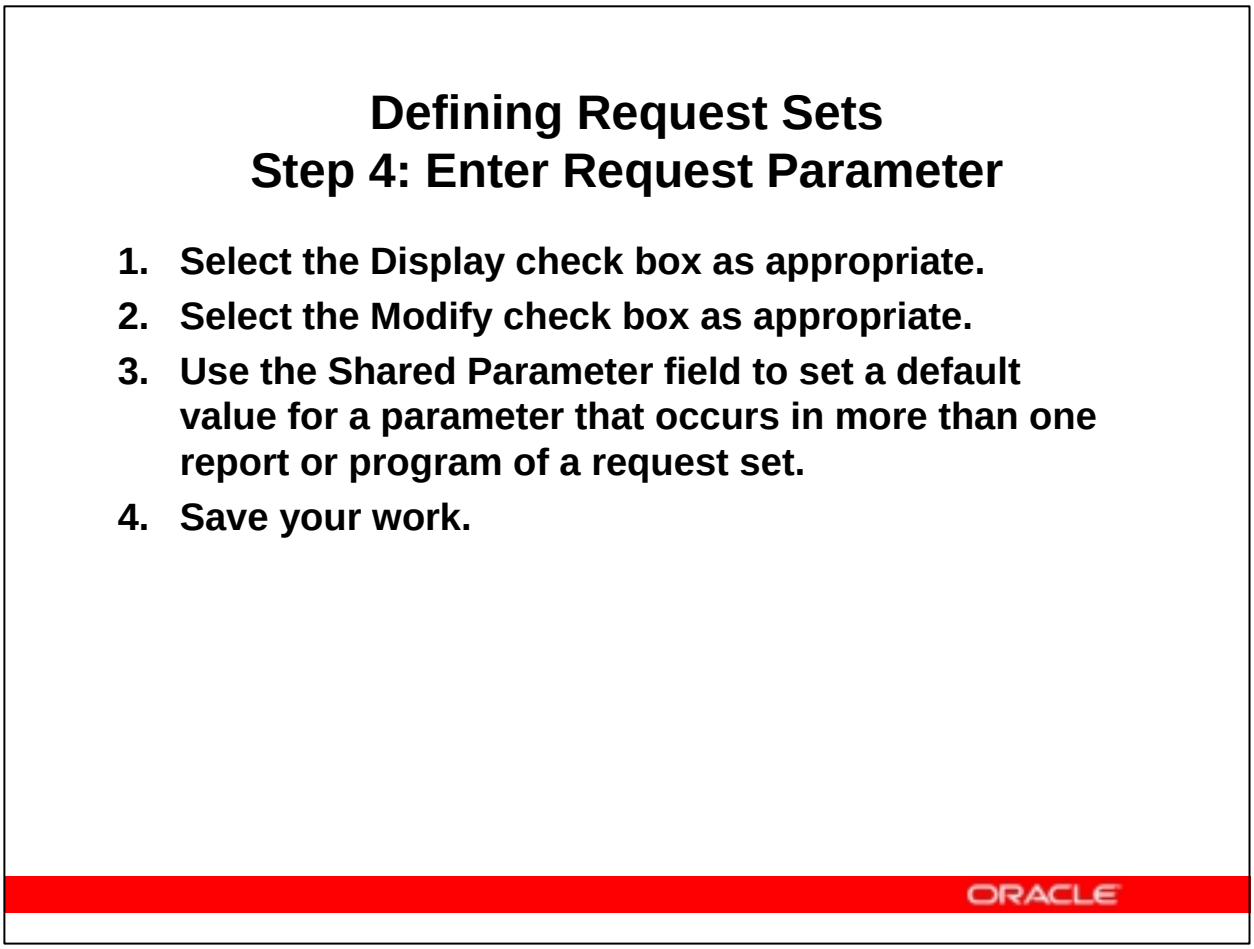
The default value for the Function field is Standard Evaluation that bases its completion status on the normal completion status of the requests it contains.

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 Stages (continued)

(N) Requests > Set > (B) Define Stages > (B) Requests



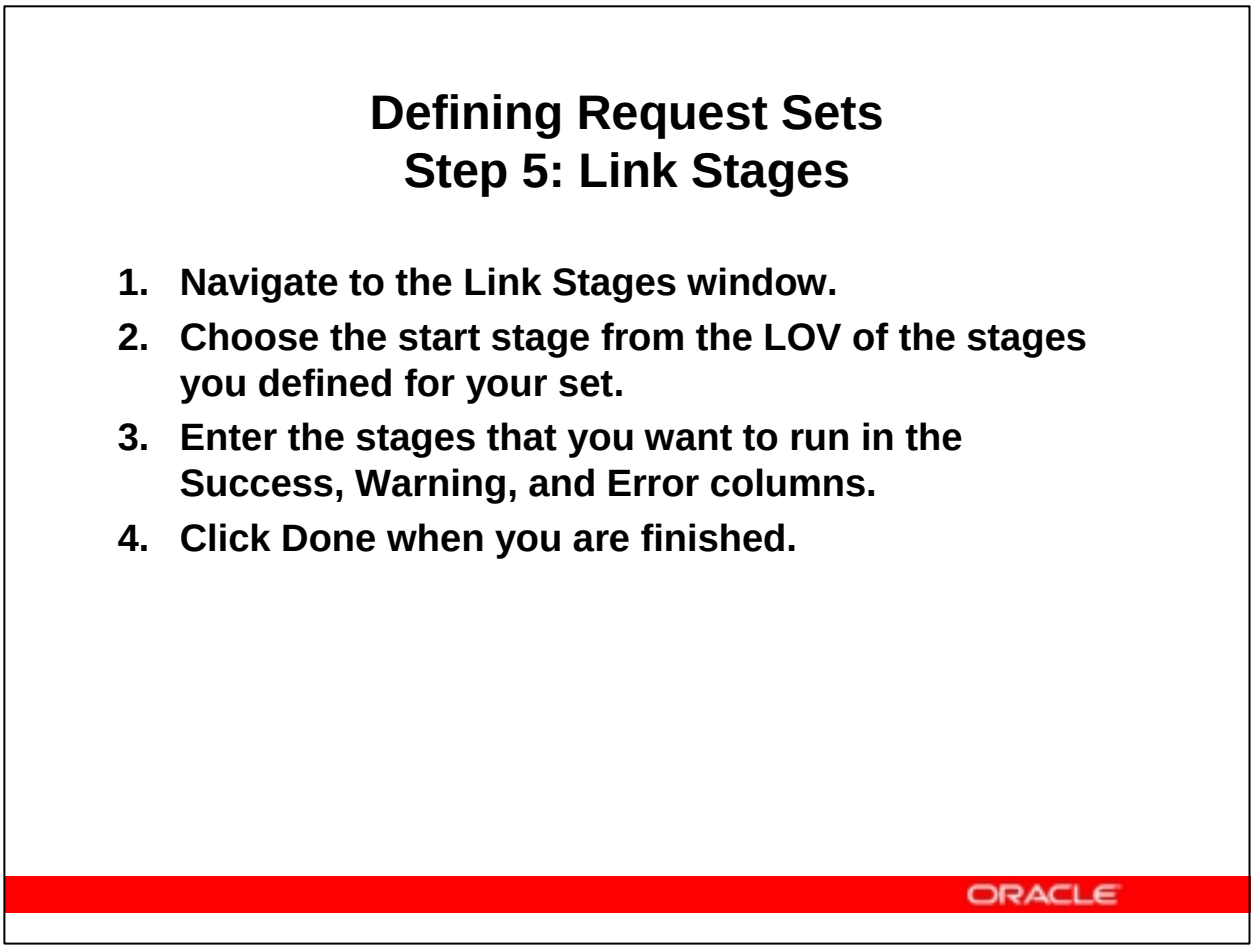
Request Parameters

((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.

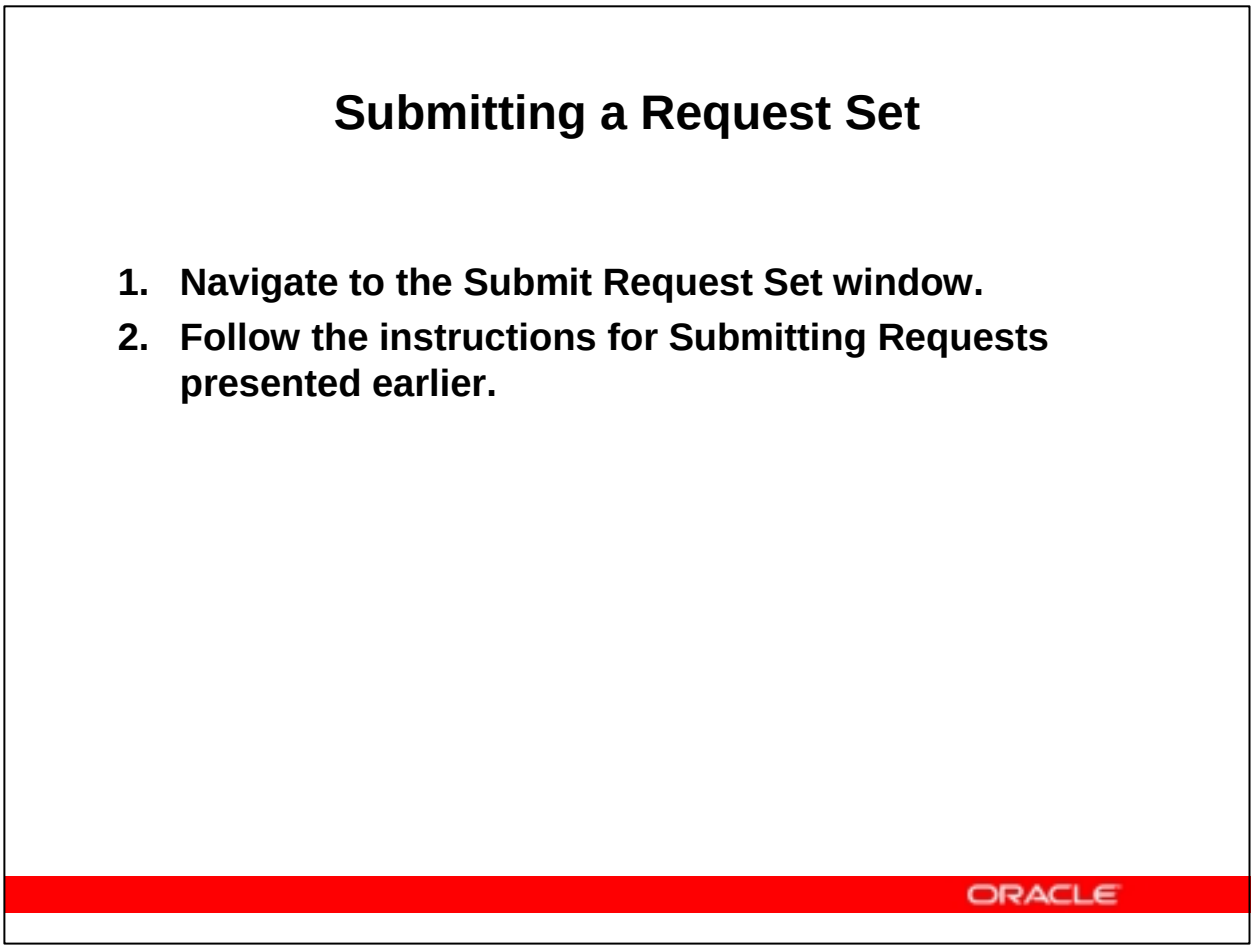
- The Sequence and Prompt fields are display only.
- Selecting the Display check box specifies that you can see a request parameter at submission time.
- Selecting the Modify check box to specifies that you can insert or change the value for a request parameter at submission time.
- Using the Shared 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.



Linking Stages in a Request Set

(N) Requests > Set > (B) Link Stages

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.

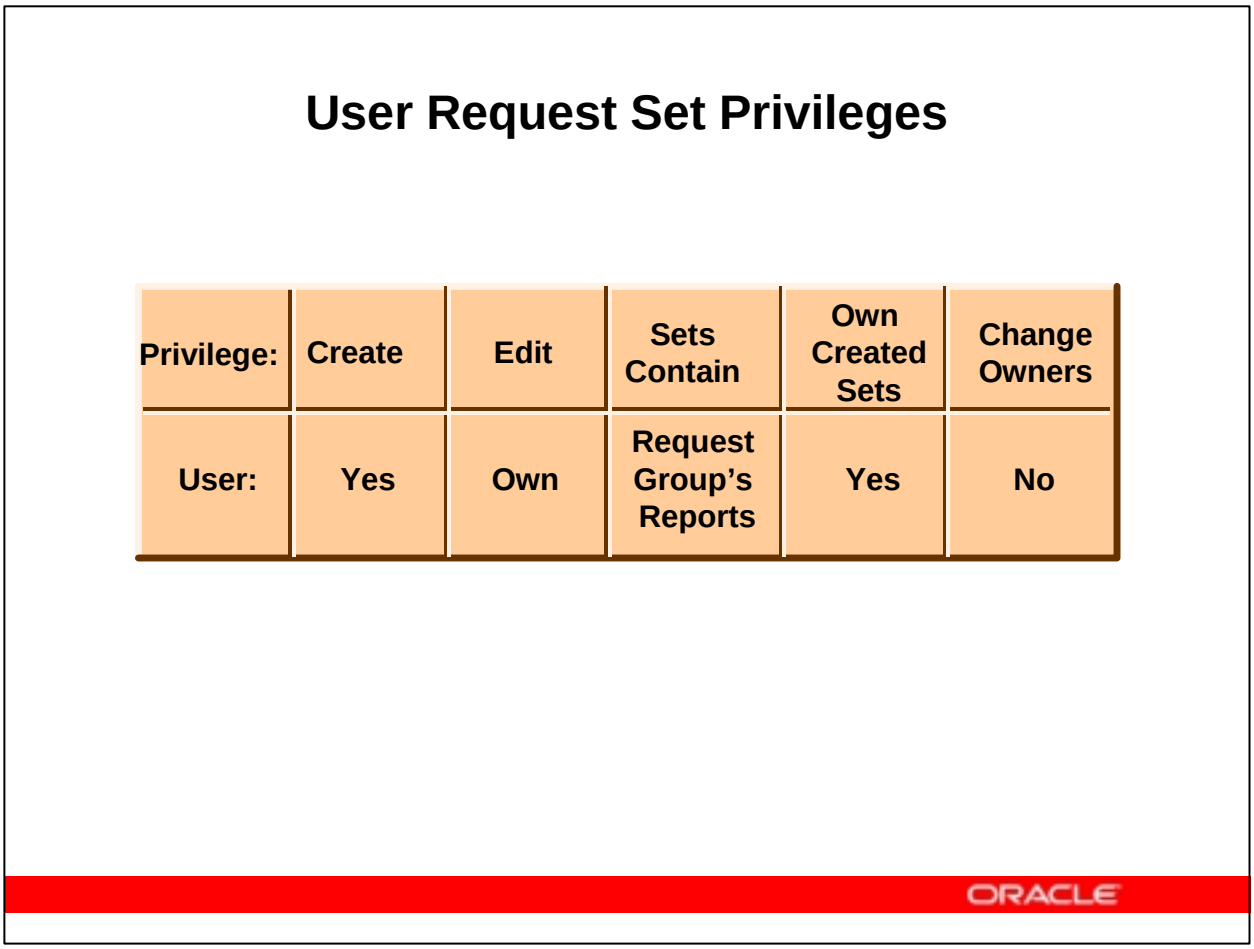


Who Can Use a Request Set

(N) Requests > Run > (Option) Request Set > (B) OK

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 about an individual request in the request set, however, the request set must be in your request group.



Request Set Privileges

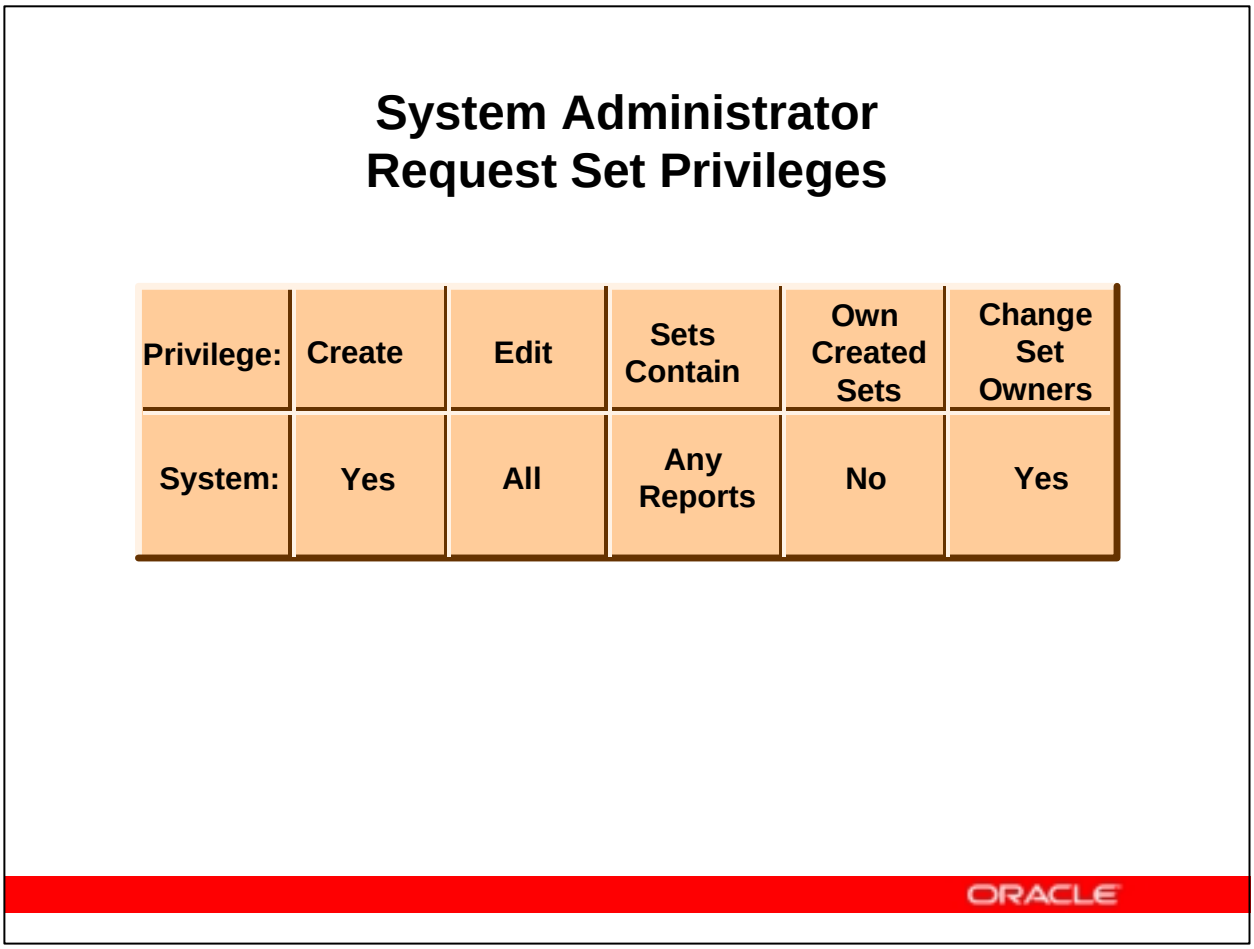
You can control request set ownership to manage request set editing and run privileges.

Application User Editing Privileges

- Application users define a request set by selecting reports or other concurrent programs from their current responsibility's request security group.
- Application users automatically own the sets that they create.
- Application users can always edit the sets that they own. They cannot edit a request set that they do not own.

Request Sets and Request Security Groups

- After changing responsibilities, application users can add reports from their current (new) responsibility's request security group to a request set that they own.
- After changing responsibilities, application users can delete reports from a request set that they own, regardless of whether the report is in the new responsibility's request security group.



System Administrator Request Set Privileges

System administrators can edit and assign ownership of any request set.

System Administrator Editing Privileges

- System administrators can define or edit a request set to include any standard submission report or concurrent program.
- System administrators do not automatically own the request sets that they define. Request sets defined by a system administrator initially do not have an owner.
- System administrators can assign or change ownership of any request set.
- If a request set is assigned to an owner but not to the owner's request security group as well, the reports contained in the request set are available only to the owner.
- System administrators can edit any request set, whether the set is owned or not.
- System administrators can assign any request set to any request security group, even if the request set contains individual reports that are not in the request security group. However, this privilege provides the members of a responsibility access to reports and programs outside their request security group.

Practice - Request Sets Using Wizard

Overview

To wrap up the testing of concurrent reports and programs, the implementation committee wants you to test creating a Request Set. They want you to use the Request Set Wizard to do this, and then test your Request Set when you are done.

Tasks

Create your Request Set Using Wizard

1. Create a new Request Group Set using the Wizard. Remember to put your initials at the front of your data entries to keep your data unique.

    - **Set Name**: *Your Initials* Assistant System Administrator
    - **Set Code**: (will be assigned by Wizard*)*
    - **Application**: Application Object Library
    - **Description**: *Your Initials* Assistant System Administrator
    - **Owner**: (will be assigned by the Wizard)

2. Accept all of the other default values.

3. Include the following programs in your request set, Users of a Responsibility, Reports and Sets by Responsibility, and Report Group Responsibilities.

Test your Request Set

4. Test your Request Set by submitting it to run through concurrent processing.

Solution - Request Sets Using Wizard

Create your Request Set Using Wizard

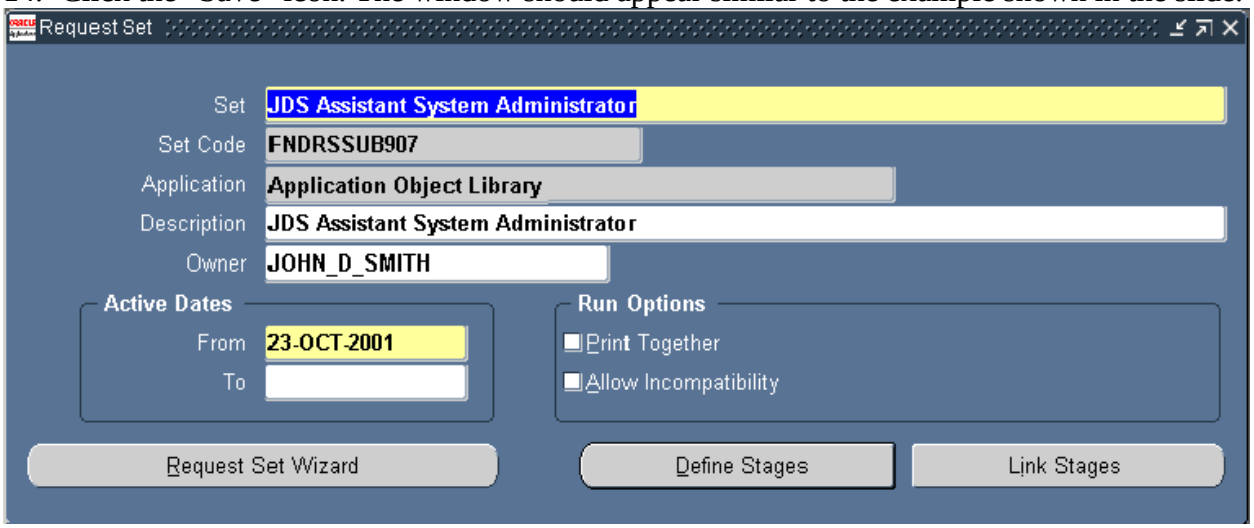
Responsibility = System Administrator

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.
2. Navigate to (N) Requests > Set.
3. Click the “Request Set Wizard” button (DO NOT enter any other information – the Wizard will prompt you).
4. Accept the following default values:
    - Run your set Sequentially.
    - Click the “Next” button (this will create three stages).
    - Abort your set when the status of the stage ends in Error.
    - Click the “Next” button (this will link the stages).
5. Enter the following:
    - **Set Name**: *Your Initials* Assistant System Administrator
    - **Application**: Application Object Library
    - **Description**: *Your Initials* Assistant System Administrator
6. Click the “Next” button.
7. Accept the default “As Each Request in the Set Completes”.
8. Click the “Next” button.
9. Select the following programs to be included in your set:
    - Users of a Responsibility
    - Reports and Sets by Responsibility
    - Report Group Responsibilities
10. Click the “Finish” button.
11. A note appears telling you that your request set has been created and saved.

12. Click the “OK” button.

13. Verify the Owner field has the user name of the user you are currently logged in as.

14. Click the “Save” icon. The window should appear similar to the example shown in the slide.



Test your Request Set

15. Sign on as your *Your Initials* Assistant System Administrator.

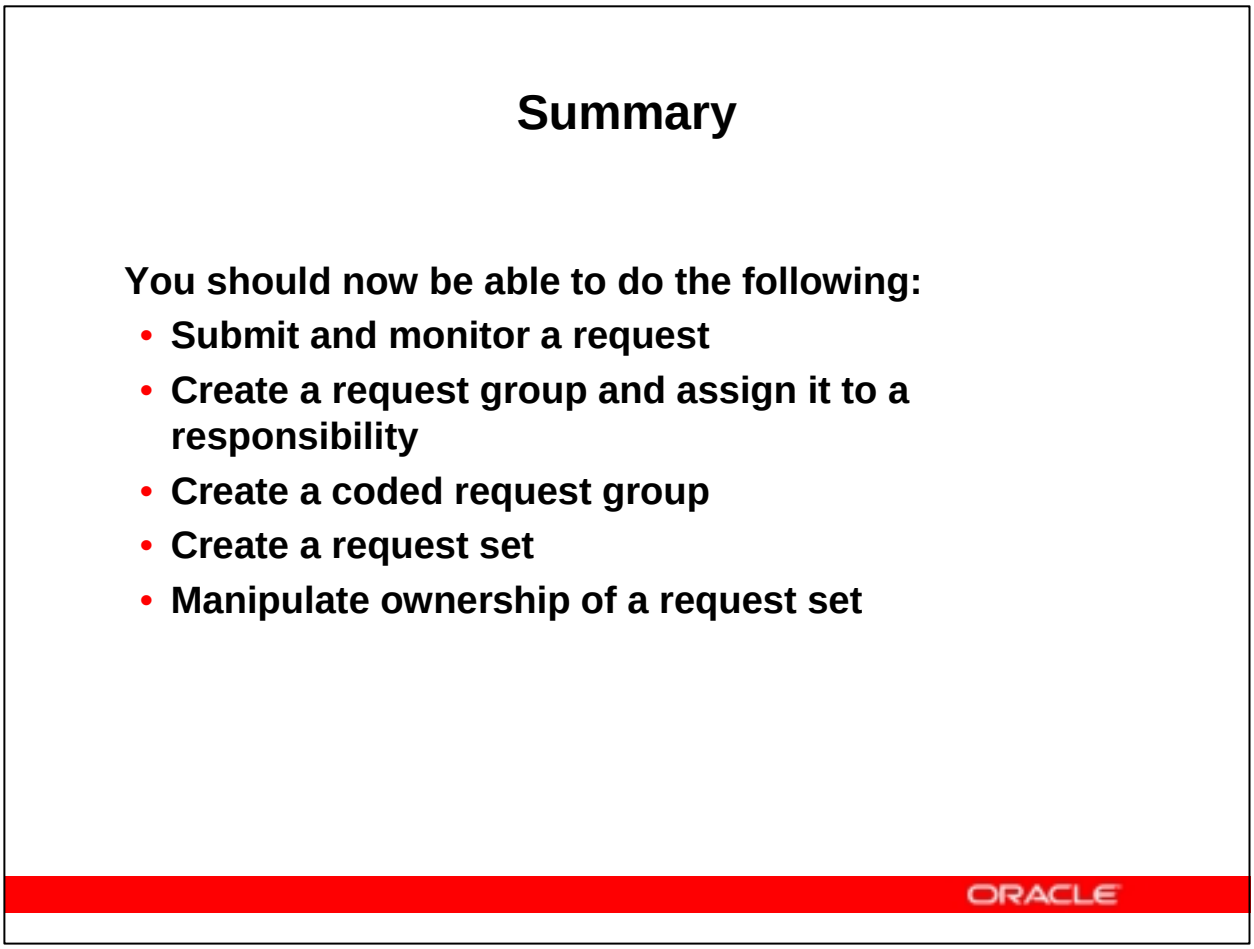
16. Navigate to (N) Requests > Run.

17. Select the “Request Set” option.

18. Click the list of values icon for the “Request Set” field. Your request set will automatically display because it is the only set available to your *Your Initials* Assistant System Administrator responsibility.

19. Supply the required parameters by clicking in the Parameters field for each request.

20. Submit your set.

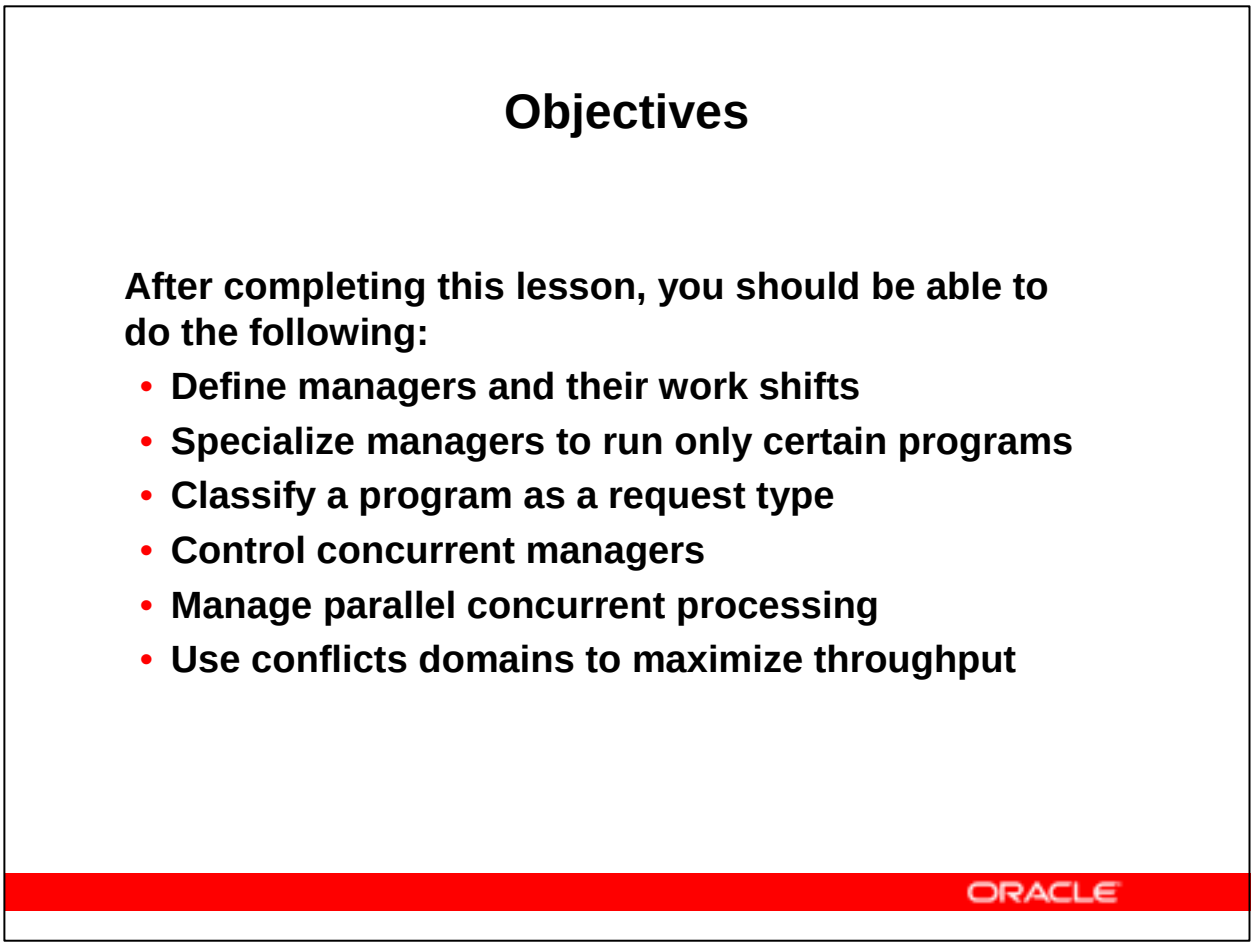


Lesson Summary

In this lesson you learned how to use request groups to control access to reports and concurrent programs. Additionally you learned how individual users can define request sets to streamline processing by setting default values for parameters and sharing parameter values among multiple reports.

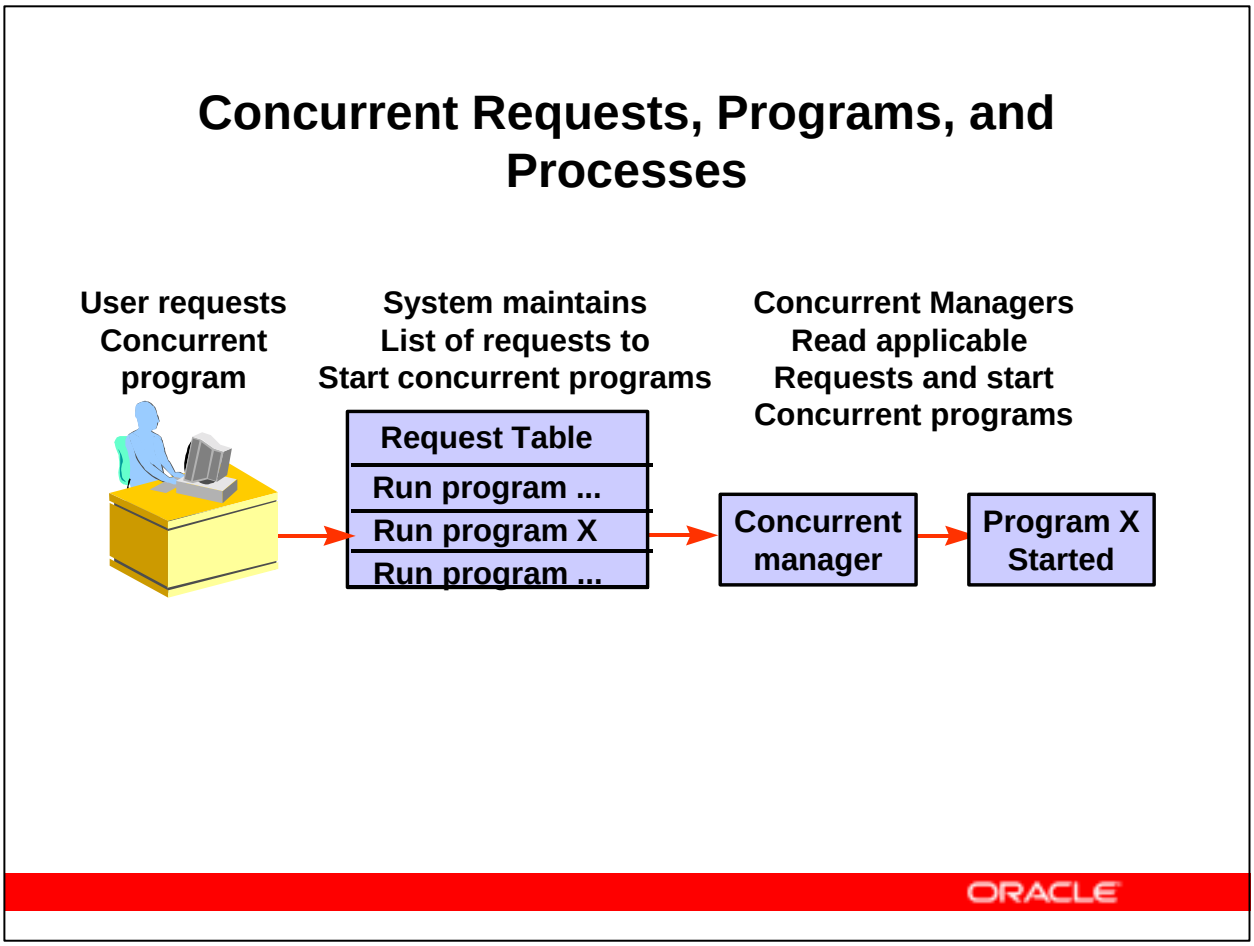
Administering Concurrent Managers

Chapter 4



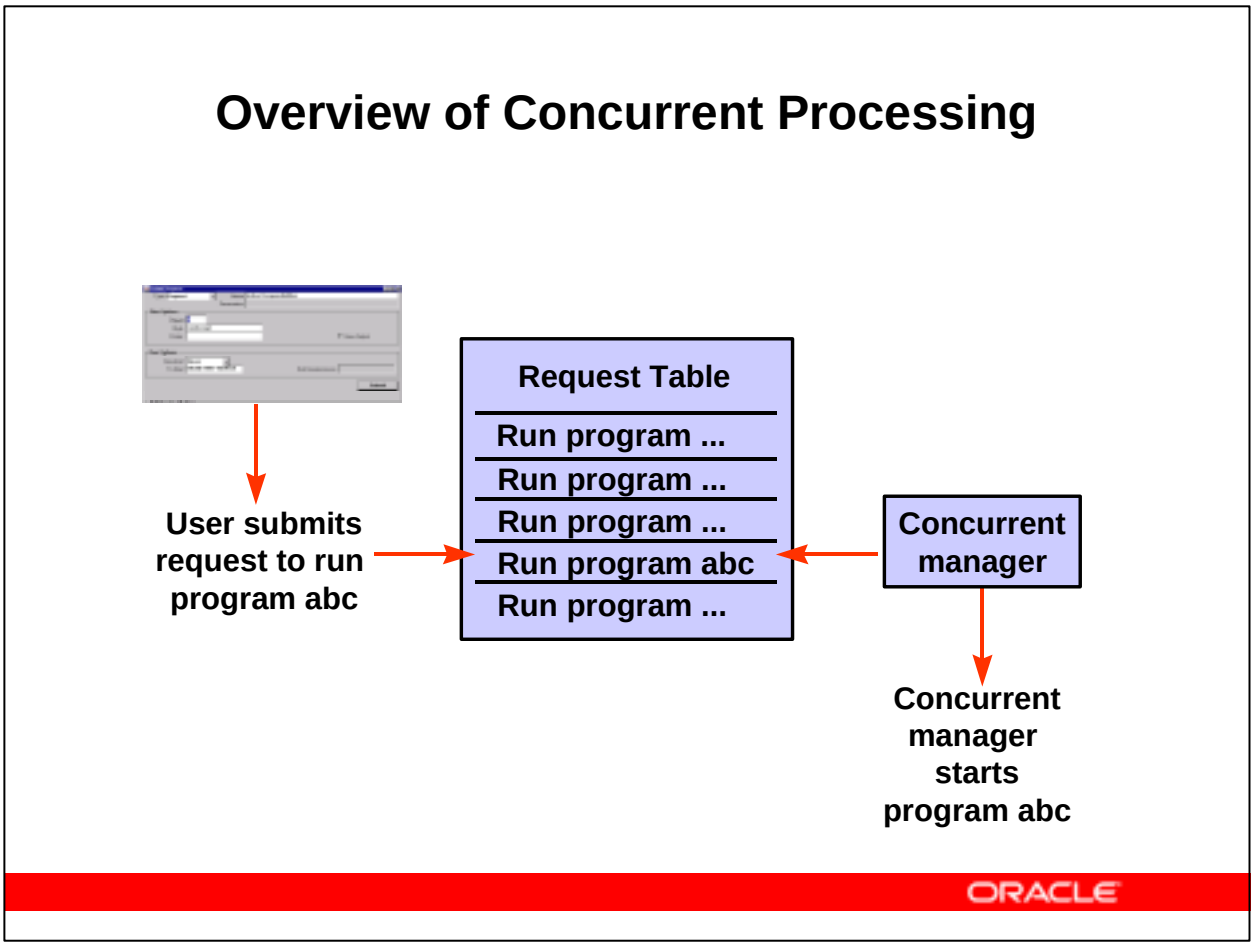
Lesson Aim

The Oracle system administrator is the end user's first point of contact for system questions. The system administrator should be able to monitor the processing of requests and deal with any problems. He or she should also be able to balance workloads to provide the optimum processing environment for users. This lesson teaches you how to monitor and control system processing and also how to plan and schedule your workload.



Concurrent Requests, Programs, and Processes

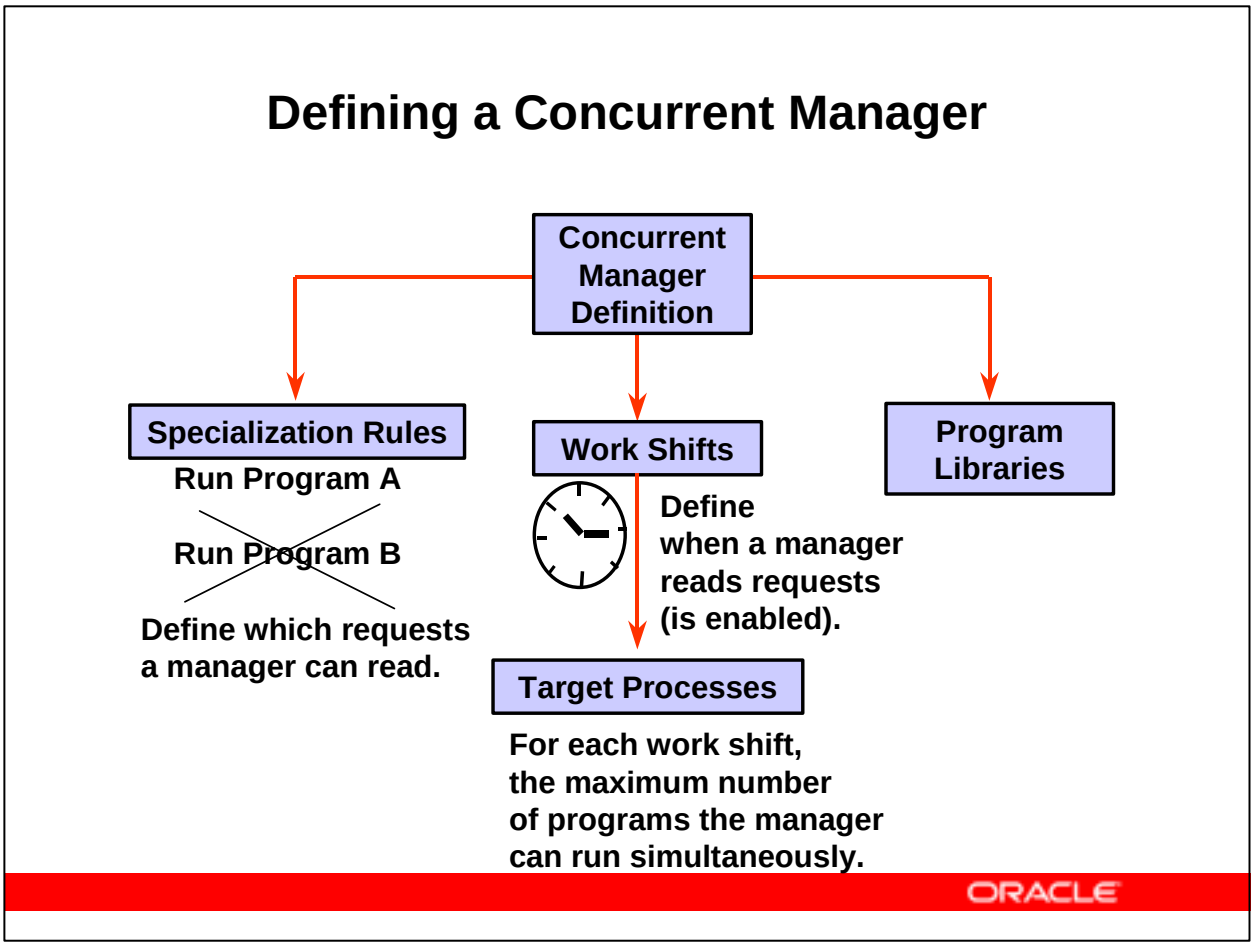
When a user runs a report, a request to run the report is generated. The command to run the report is a **concurrent request**. The program that generates the report is a **concurrent program**. Concurrent programs are started by a **concurrent manager**.



Overview of Concurrent Processing

A concurrent manager is itself a concurrent program that starts other concurrent programs running. When an application user submits a request to run a program, the request is entered into a database table that lists all of the requests. Concurrent managers read requests from the table and start programs running.

Part of a manager's definition is how many operating system processes it can devote to running requests. This number is referred to as the manager's number of **target processes**.

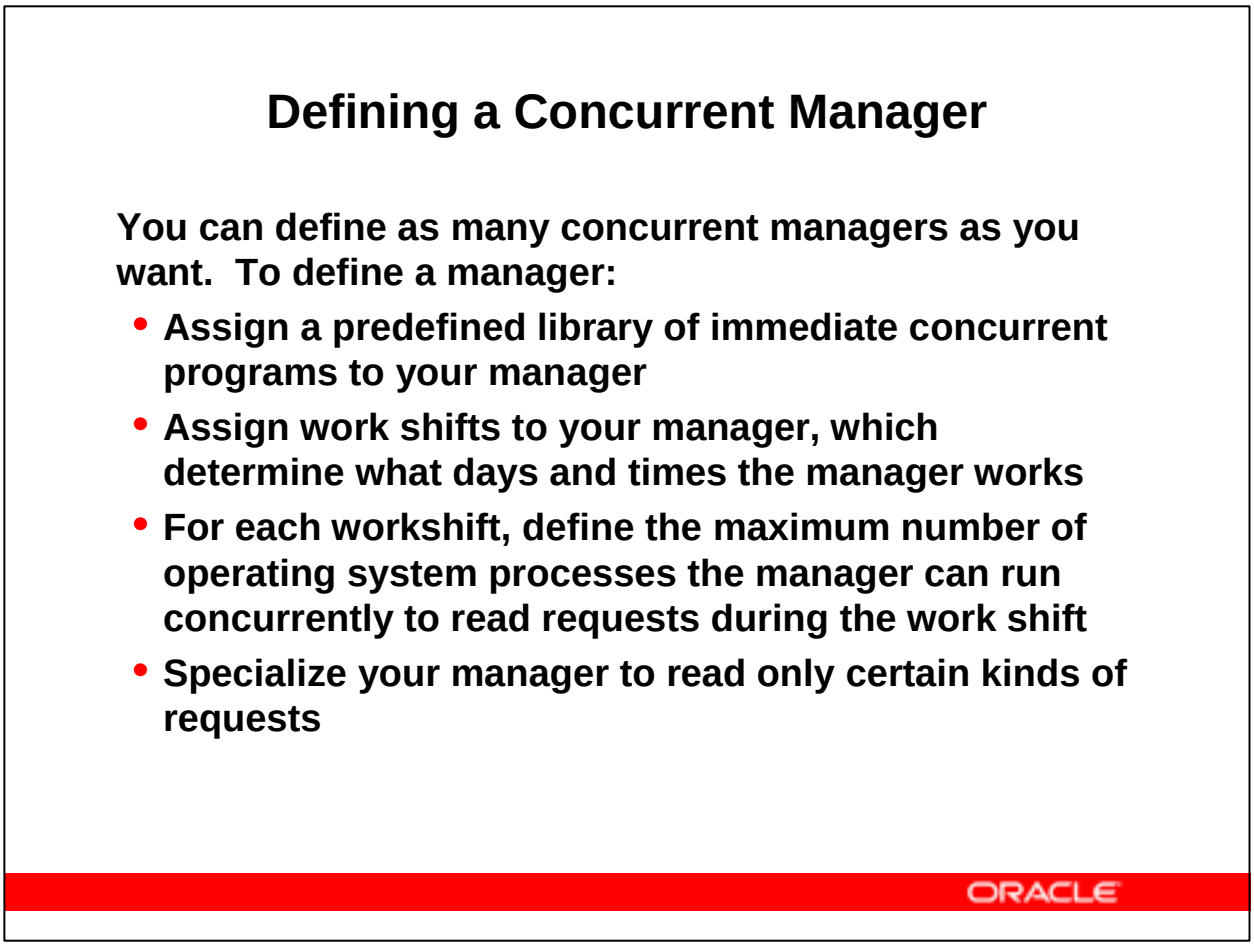


Defining a Concurrent Manager

Concurrent managers can be defined to process only certain types of requests and to operate during certain times of the day.

Scheduling Requests You can schedule when a manager operates by defining one or more work shifts and associating the manager with a work shift.

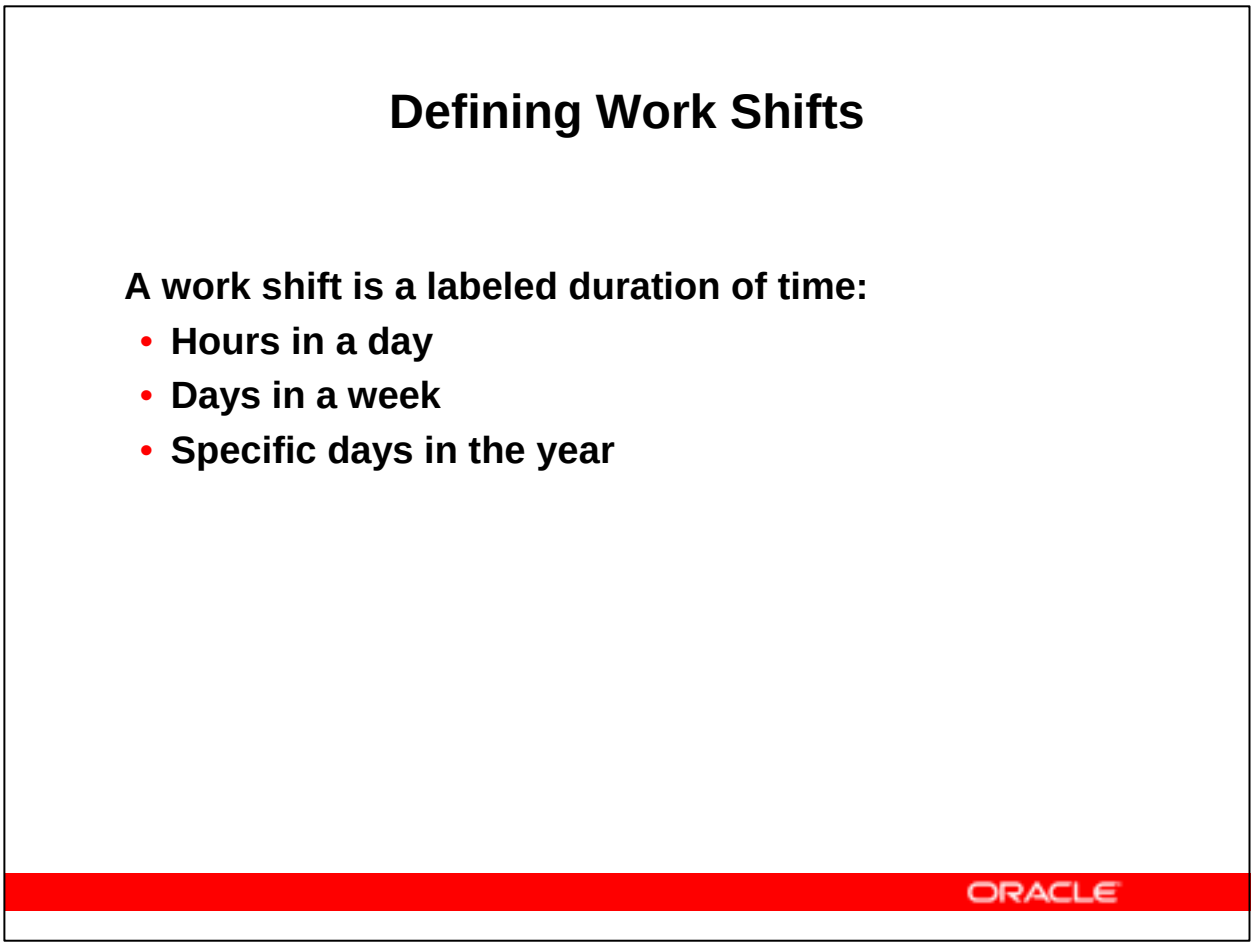
Segregating Requests You can define specialization rules to identify the kinds of requests that a manager will read. Specialize managers so that they only read requests to start certain kinds of programs.



Defining a Concurrent Manager

(N) Concurrent > Manager > Define

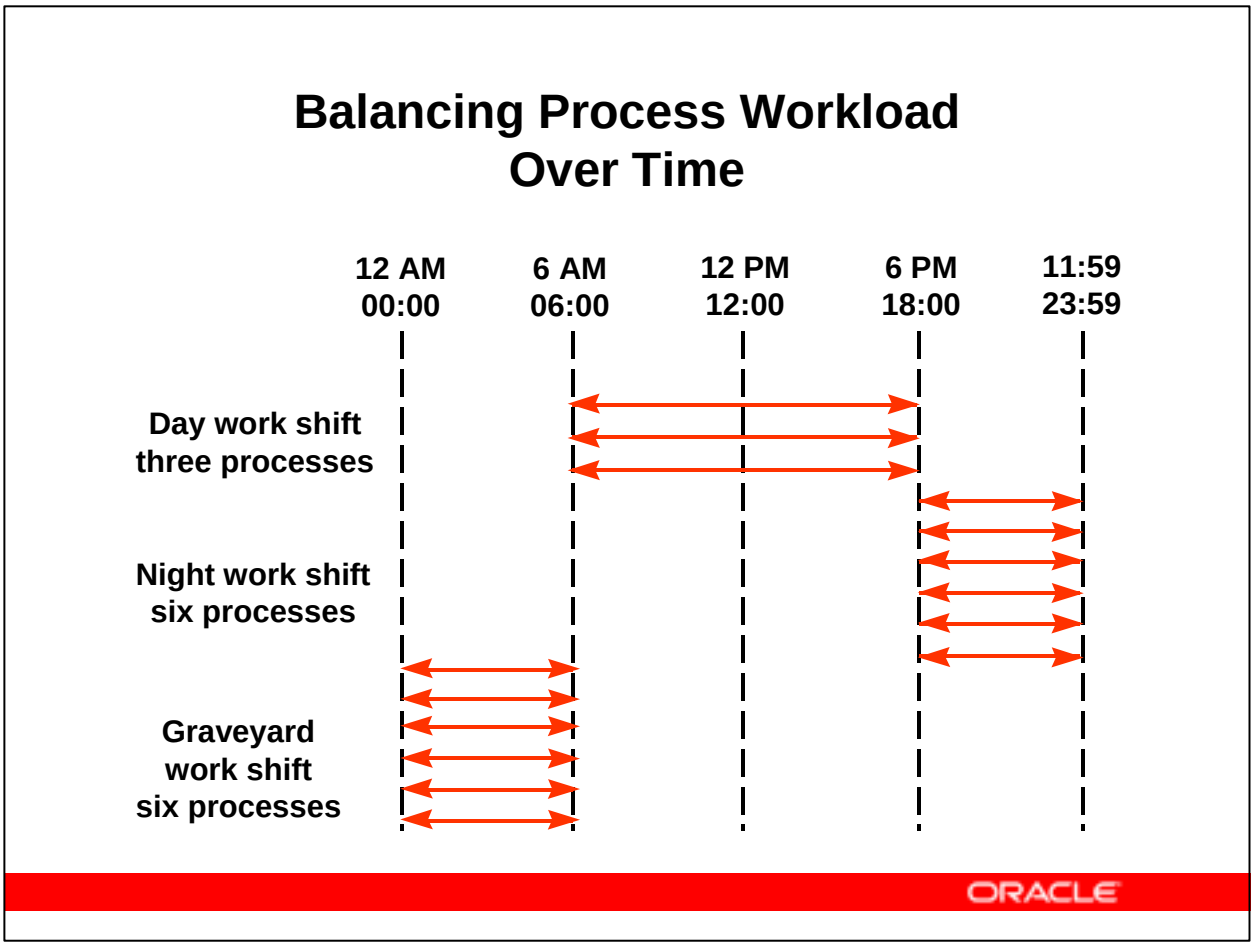
The Parallel Concurrent Processing Details region of this window is discussed in a later lesson.



Defining Work Shifts

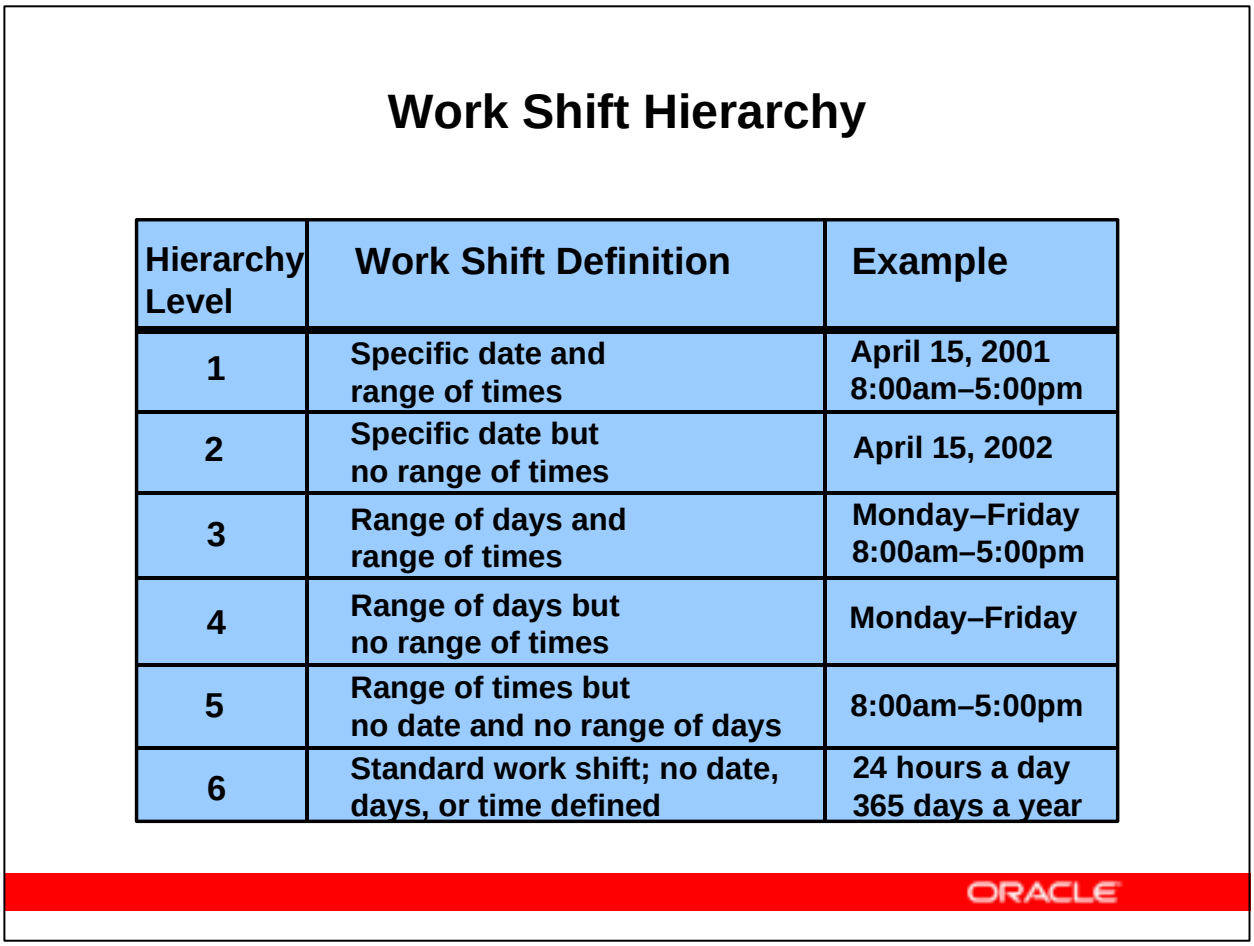
(N) Concurrent > Manager > Workshifts

A concurrent manager operates only during its specified work shift. Use military time notation to define a work shift—that is, use 24 hours rather than 12.



Balancing Process Workload

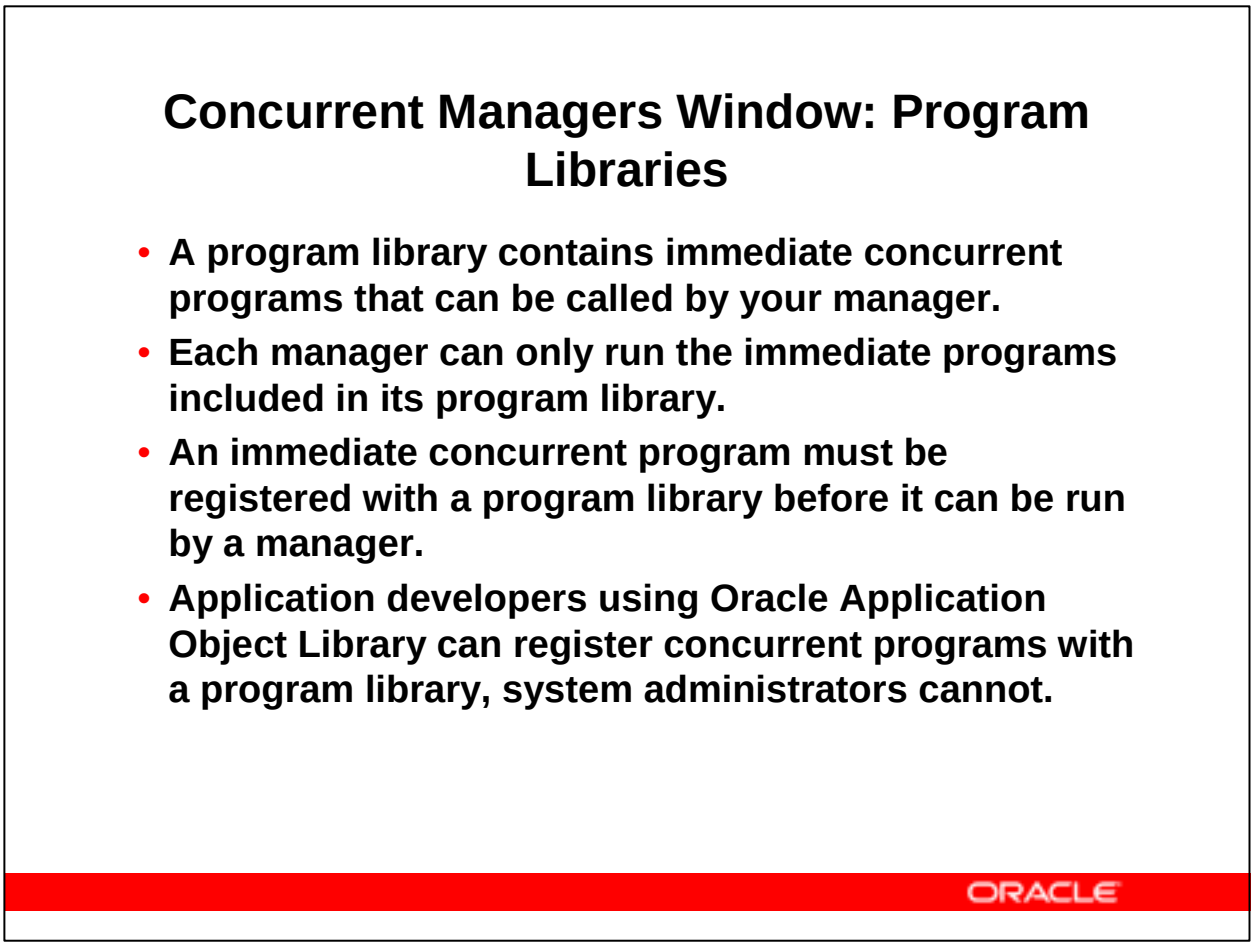
You can create multiple work shifts for a concurrent manager to regulate the number of operating system processes that the manager starts up at different times of the day and different days of the week.



Overlapping Workshifts

When work shifts overlap, the work shift with the more specific time period takes effect for the overlapping time period. For example, a work shift for January 1 overrides a work shift from 9:00 am to 5:00 PM every Monday through Friday.

When work shifts with the same hierarchy level overlap, the work shift with the largest number of target processes takes effect.



Concurrent Managers

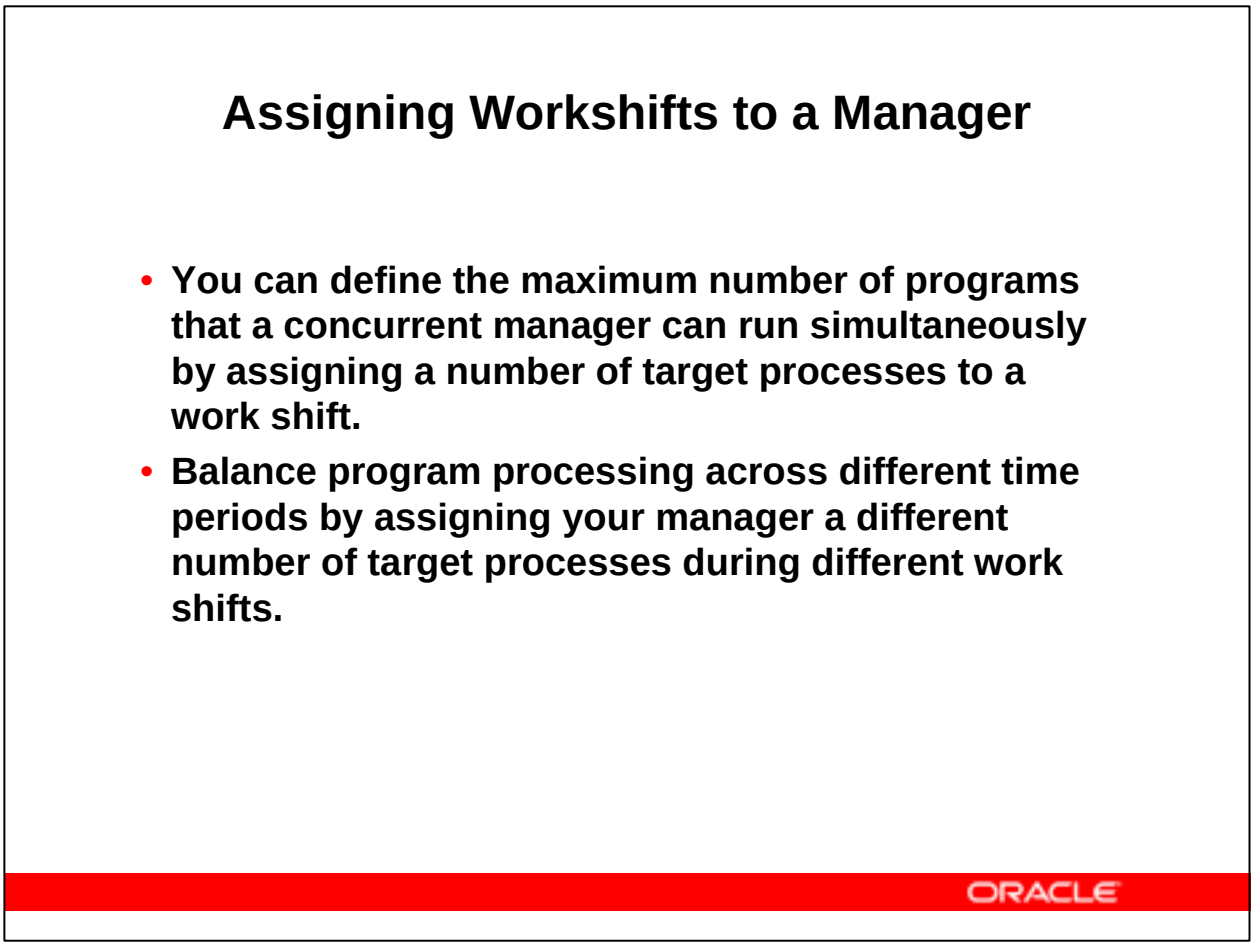
(N) Concurrent > Manager > Define

Immediate Programs and Spawned Programs

Immediate concurrent programs are C or PRO*C subroutines linked with concurrent managers. Most concurrent programs are spawned as independent processes at run time instead of running as subroutines. If you are defining a new concurrent program, you should run the program as a spawned program rather than as an immediate program.

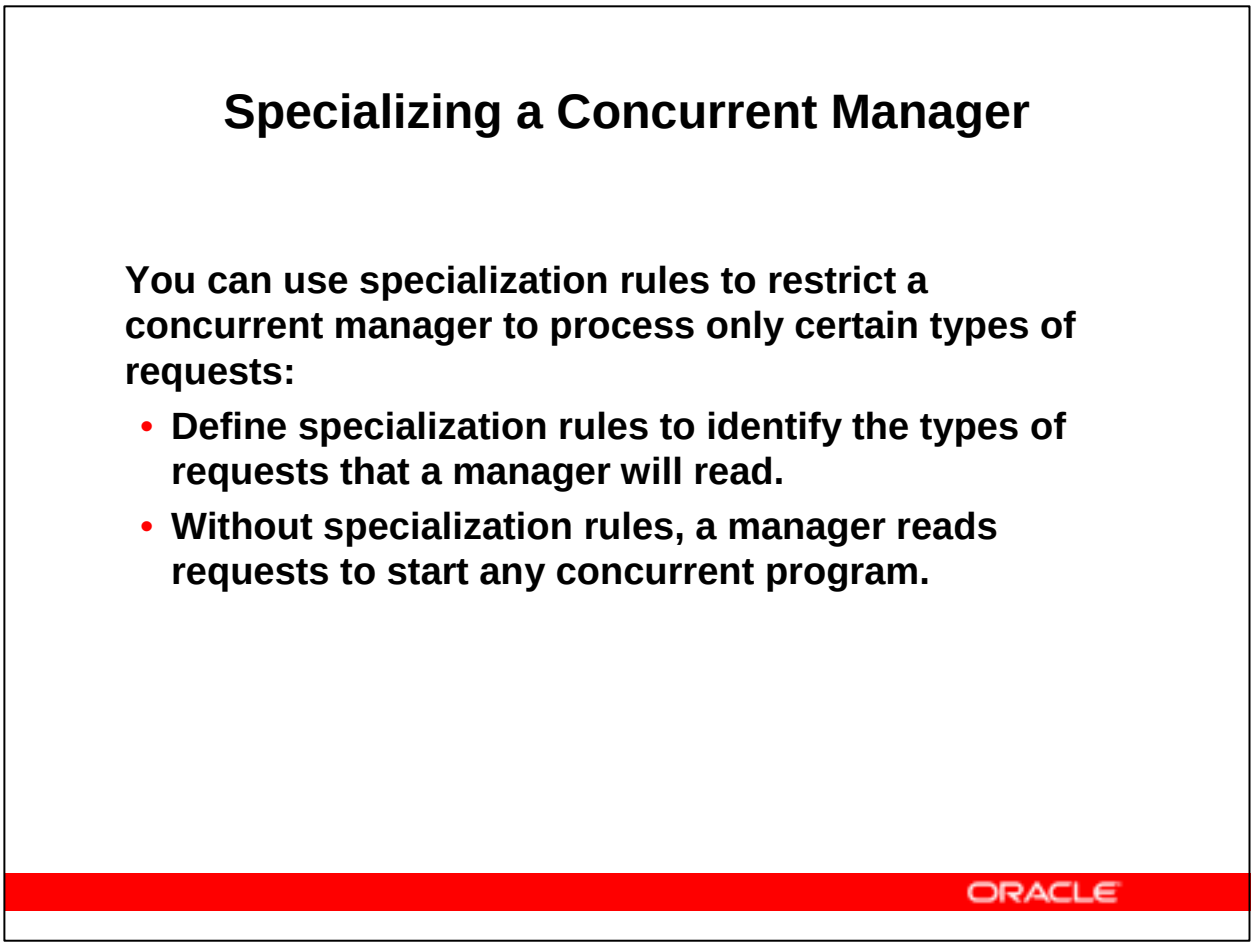
Examples of these are:

- FNDLIBR: Contains immediate concurrent programs used by Oracle Applications Object Library.
- INVLIBR: Contains immediate concurrent programs used by Oracle Inventory.
- MFCLIBR: Contains immediate concurrent programs used by Oracle Manufacturing.
- PALIBR: Contains immediate concurrent programs used by Oracle Projects.



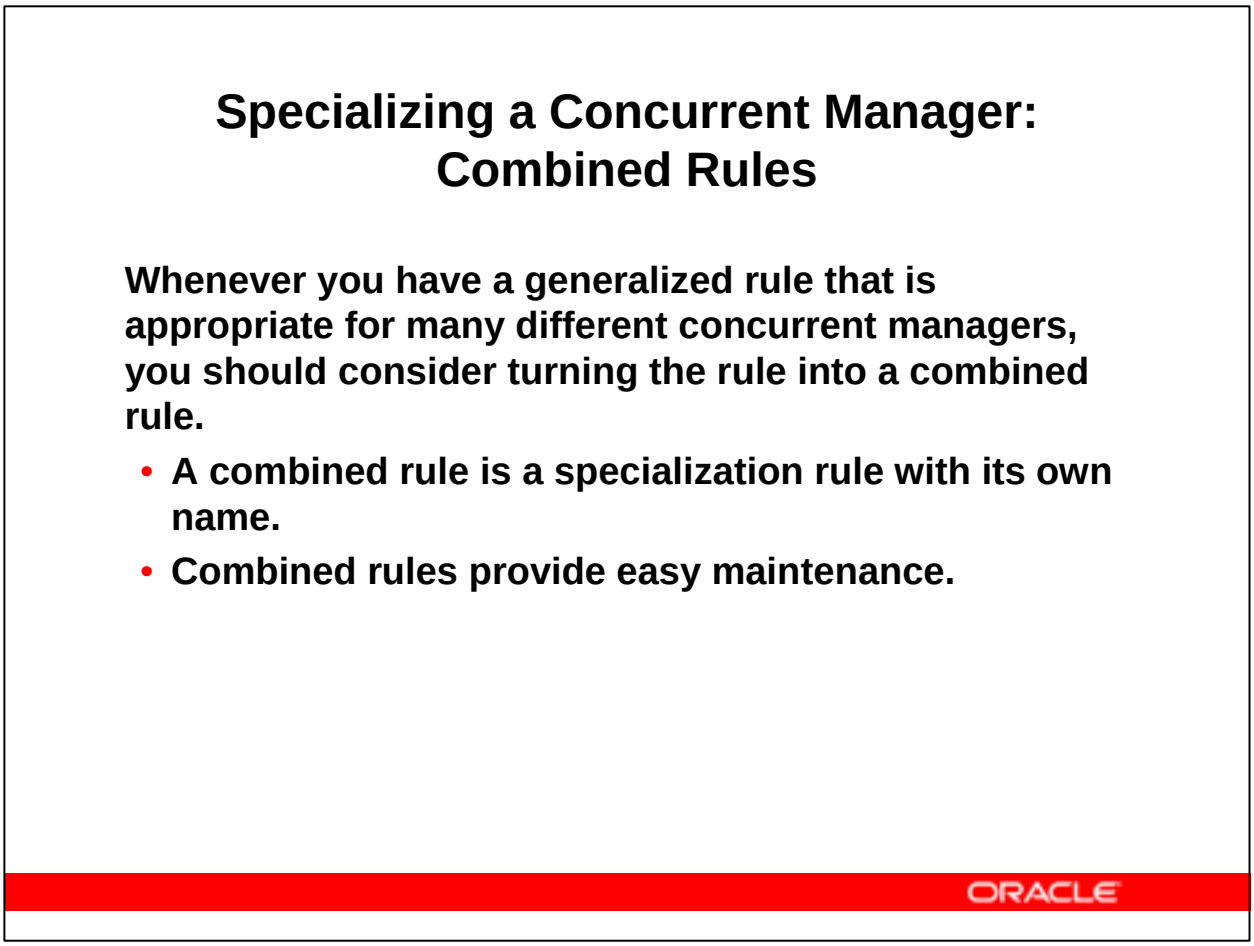
Assigning Work Shifts to a Manager

(N) Concurrent > Manager > Define > (B) Workshifts



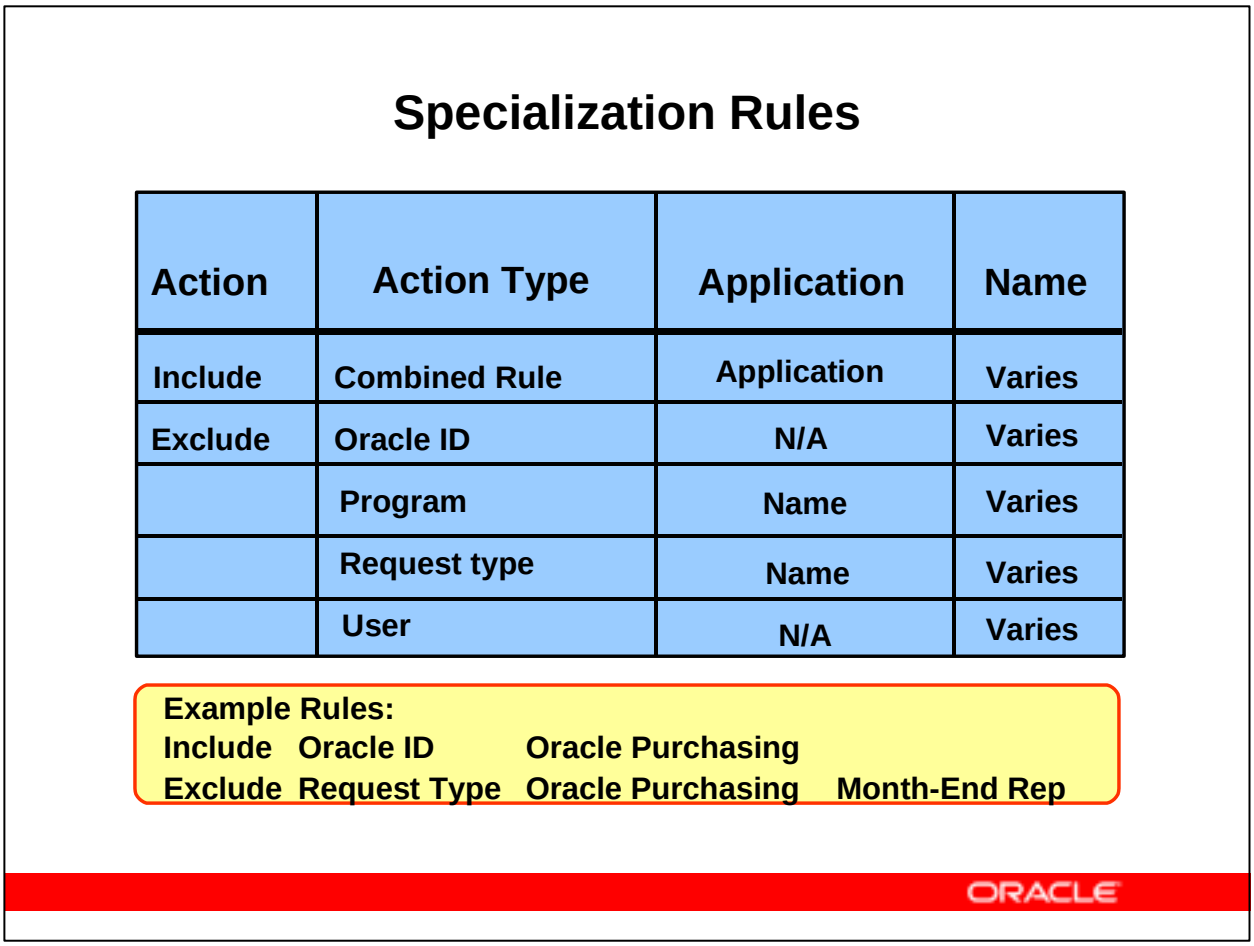
Specializing a Concurrent Manager

(N) Concurrent > Manager > Specialization Rules



Combined Rules

(N) Concurrent > Manager > Rule



Defining Specialization Rules

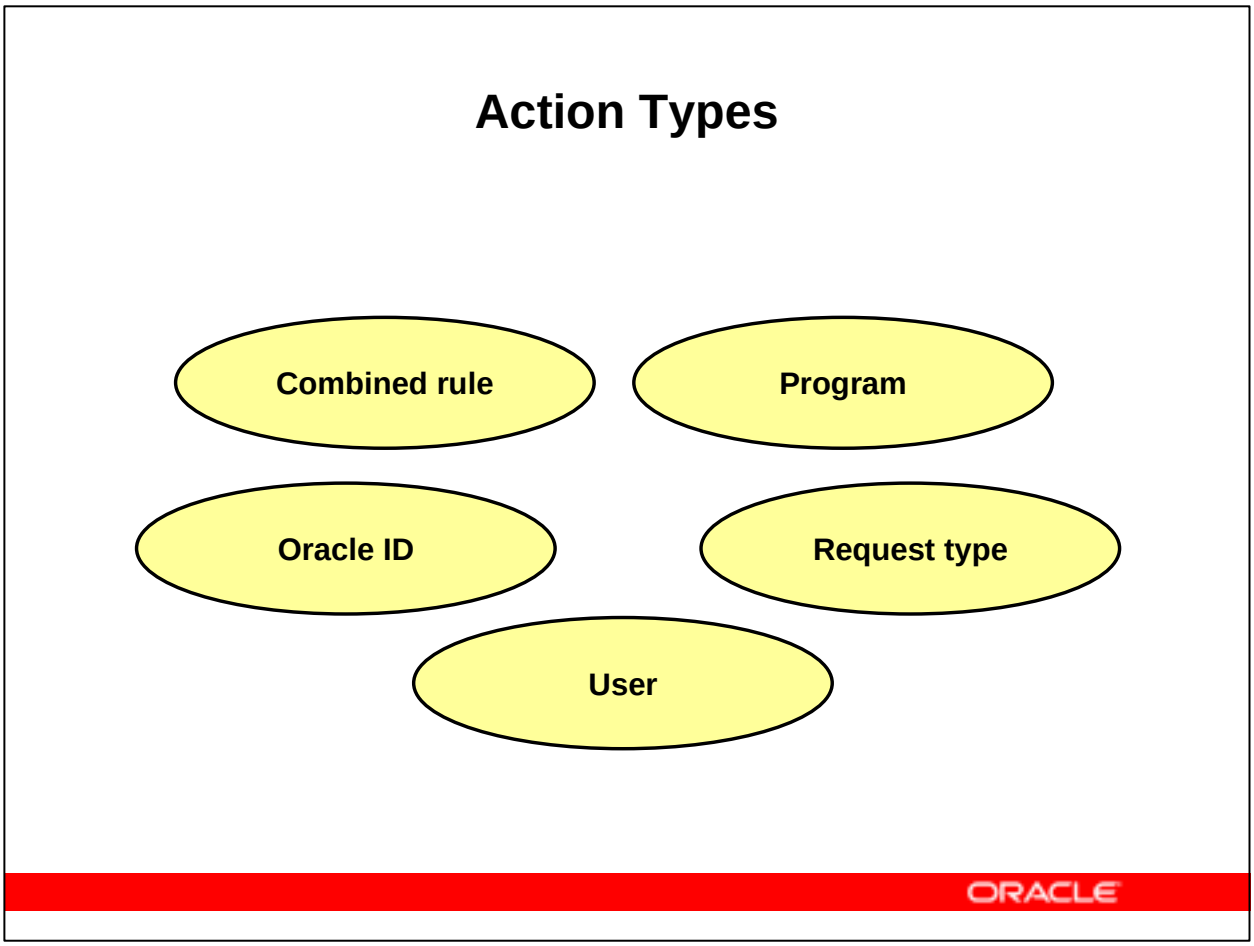
Use actions and qualifications to define specialization rules. An action defines a concurrent manager's behavior toward the request.

Include and Exclude Actions

- The manager runs requests that are included.
- The manager does not run requests that are excluded.
- Exclusion rules override inclusion rules.

Example Rules

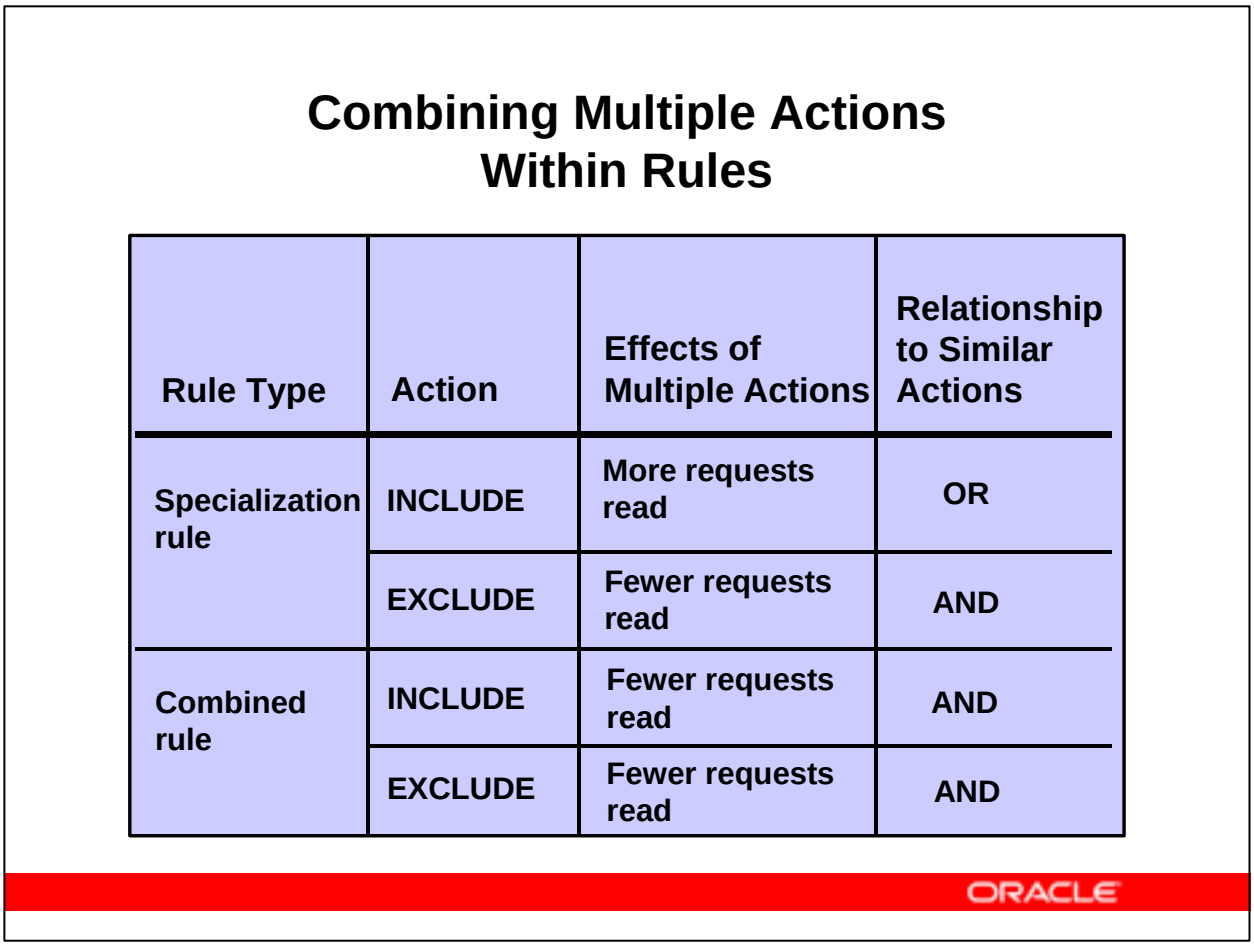
The example rules would result in a manager's reading requests for all programs assigned to Oracle Purchasing except programs defined with the request type Month-End Rep.



Action Types

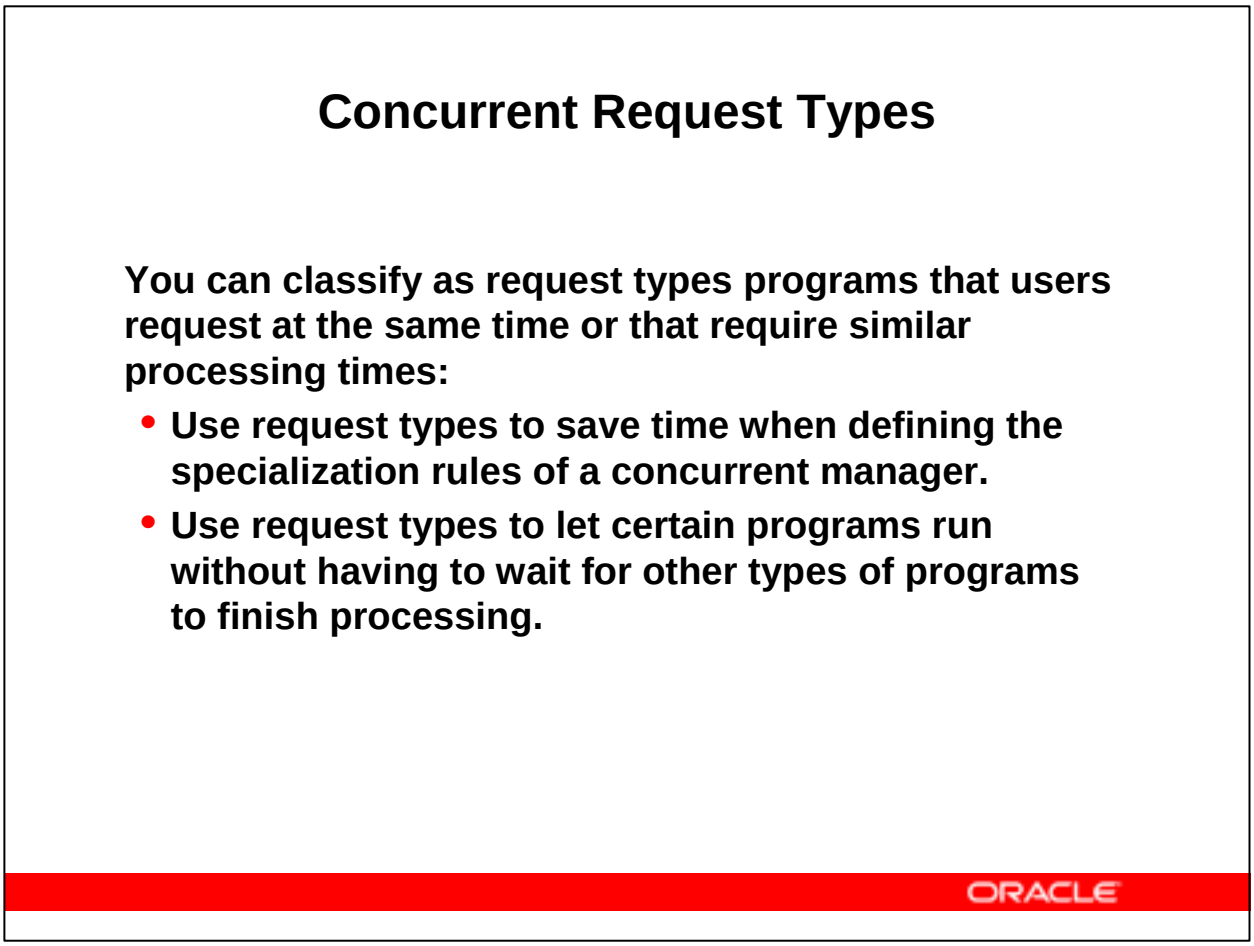
Use action types to specify the actions of a manager. An action type specifies the type of request to which an action refers.

- **Programs that connect to a specific Oracle Applications account (Oracle ID)**. For example, you could include for processing all programs that connect to the Oracle Payables account AP1.
- **Specific programs by name**. For example, you could include for processing the Oracle General Ledger program Trial Balance-Budget. When using an action type of program, you specify the application and the program name.
- **Certain request types**. For example, you could include for processing all programs that had been defined as a request type of Month-End Reports.
- **Programs requested by certain users**. For example, you could include for processing all programs requested by the user Sysadm.
- **Combined rule**. A combined rule combines more than one action to generate a single rule.



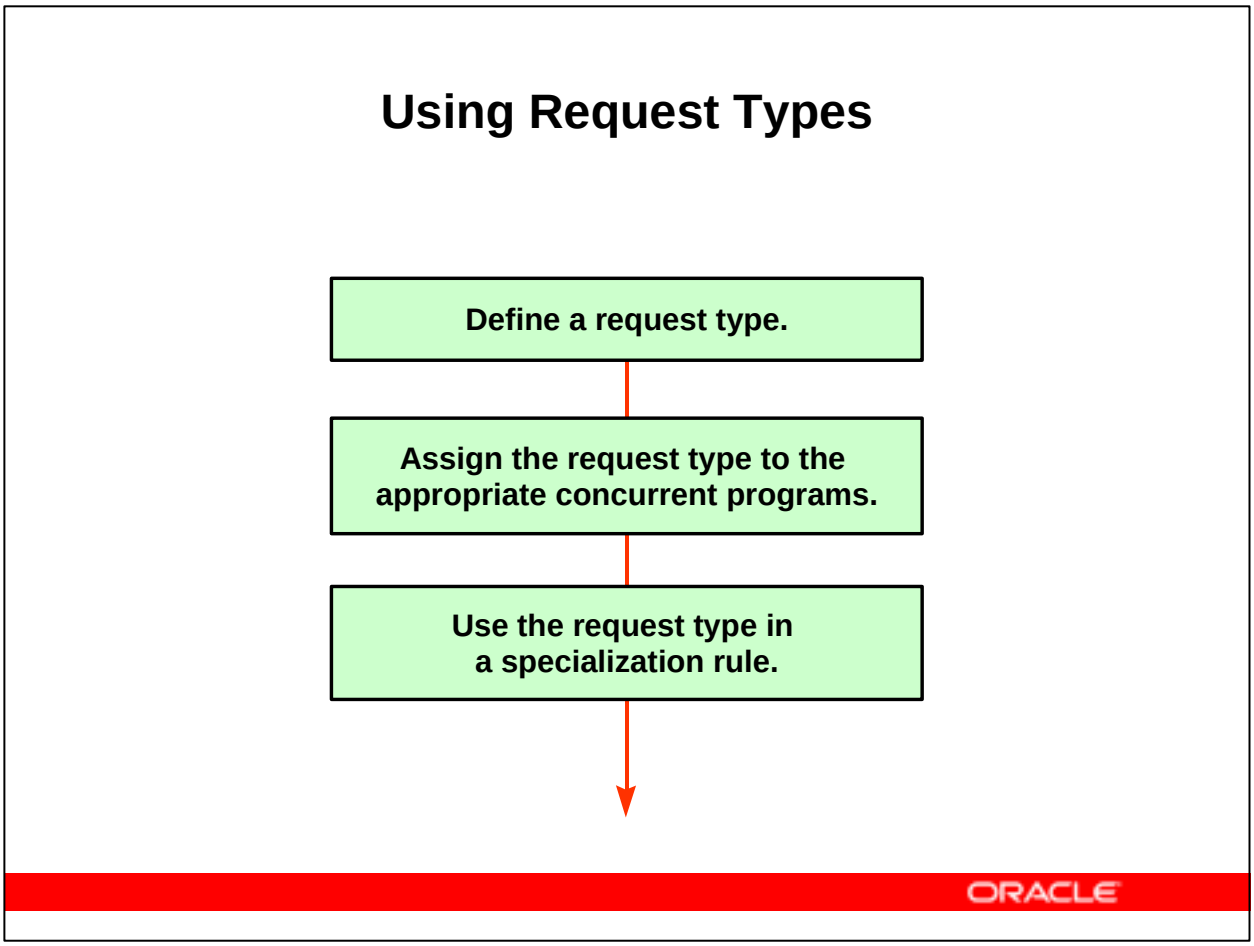
Combining Multiple Actions in Rules

- Multiple include actions in a combined rule behave differently from other multiple include actions in a specialization rule.
- A specialized include rule is associated with a specific concurrent manager. A combined include rule is an independent rule possibly associated with multiple managers.
- Multiple specialized include rules create OR clauses, allowing more requests to be read by the manager.
- Multiple combined include rules create AND clauses, further restricting a manager from reading the request.



Defining Request Types

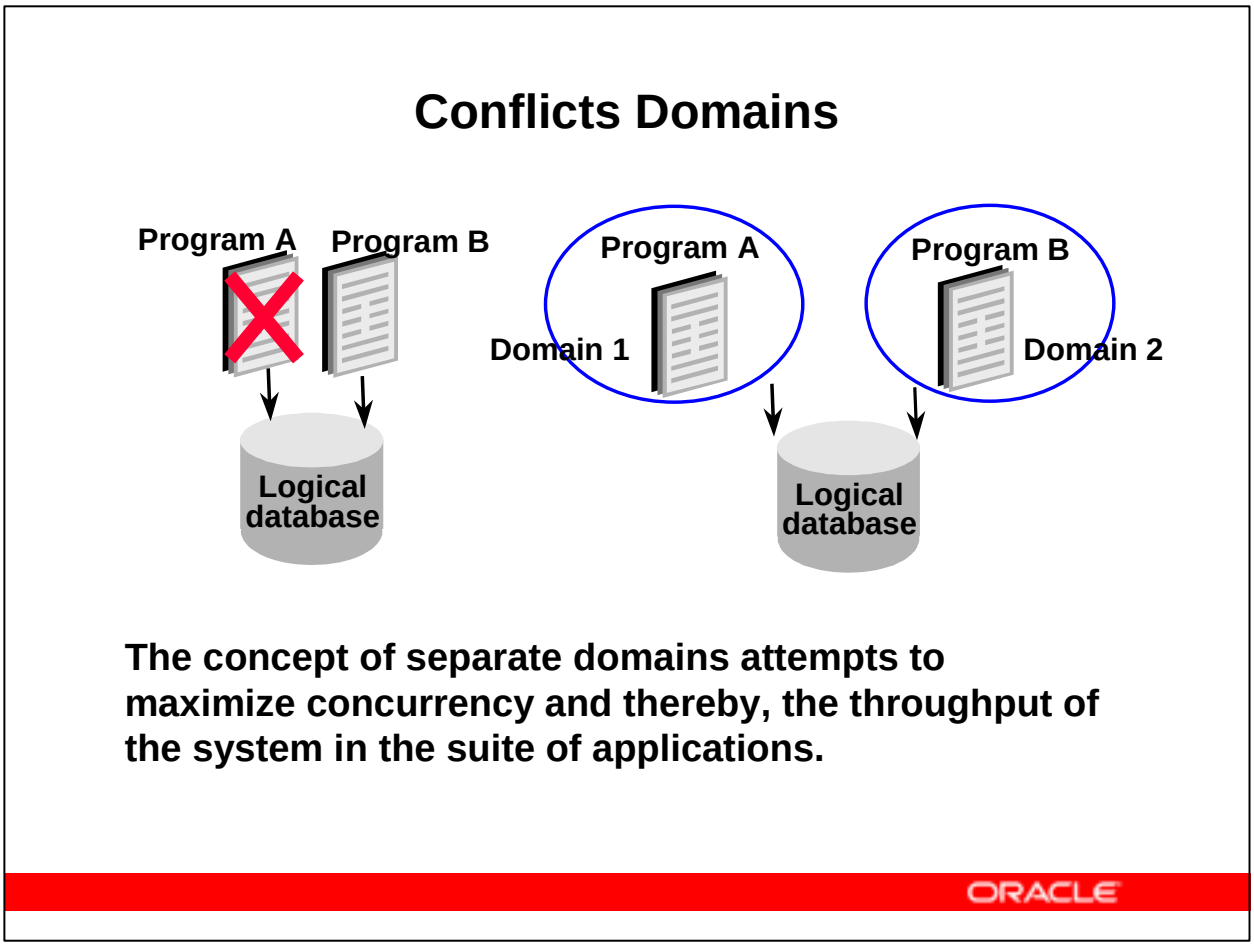
(N) Concurrent > Program > Types



How to Define and Use Request Types

There are three steps in defining and using request types:

1. Identify a category of programs according to some criterion. Define a request type for that category of programs.
2. Identify a concurrent program according to the new request type.
3. Use the new request type in a manager's specialization rule.



The Purpose of Conflicts Domains

Concurrent processing resolves and imposes conflicts and constraints within a domain and not across. That is, two incompatible programs A and B will not run concurrently when they are submitted within the same domain. But, if A is submitted within domain D1 and B is submitted within D2, they may run concurrently.

Defining Conflicts Domains

(N) Concurrent > Conflicts Domains

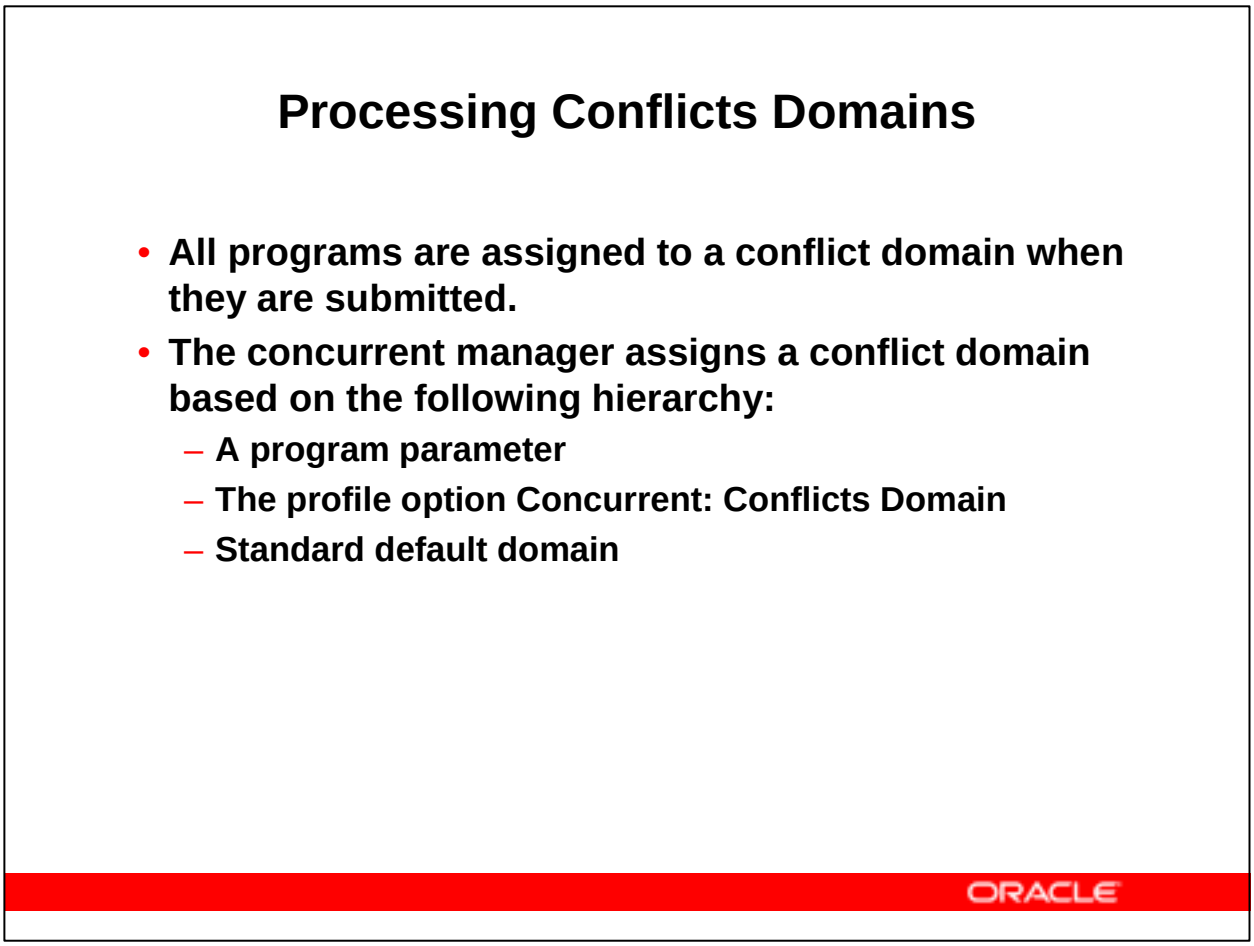
To prevent two programs from concurrently accessing or updating the same data, you have to know *where* in terms of data, they are incompatible. A Conflict Domain identifies the data that creates the incompatibility.

In Oracle Applications data is stored in database tables that belong to a particular application. Each table may also contain information used to determine what conditions need to be met to access the individual records. These conditions may consist of one or more of the following data groupings:

- SOB - based on the profile option GL_SET_OF_BOOKS
- Multiple installations (referred to as MSOB)

- Multiple Operating units (determined by profile option MO_OPERATING_UNIT) (referred to as MULTIORG)
- Multiple Orgs (determined by profile option INV_ORGANIZATION_ID, used by Manufacturing applications)
- HR may use business group as a conflict domain
- FA may use FA book

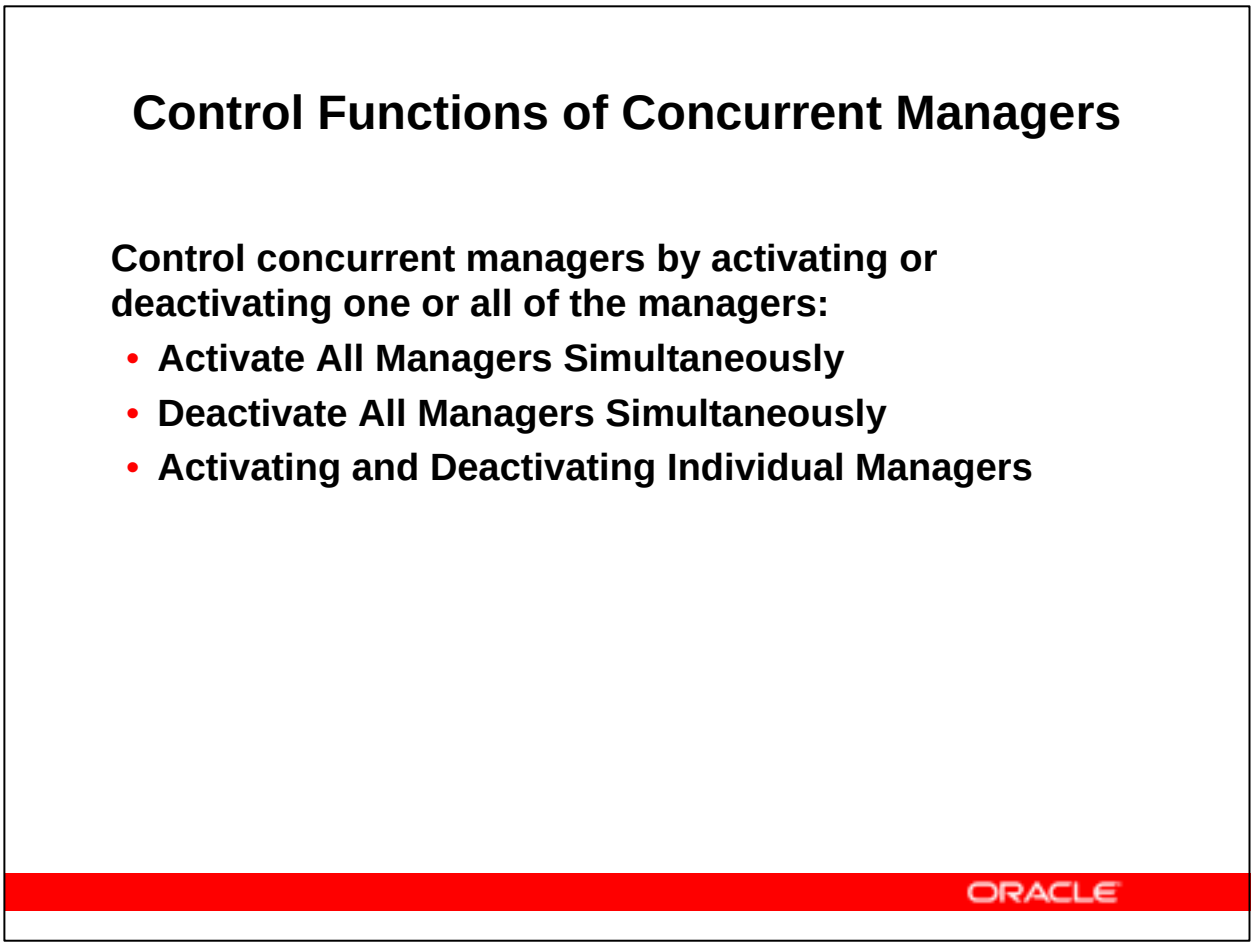
A conflict domain is an abstract representation of the groupings used to partition your data. There is no limit to the number of domains that can be defined, but excessive domains may hurt performance.



Processing Conflict Domains

All programs are assigned a conflict domain when they are submitted. If a domain is defined as part of a parameter the concurrent manager will use it to resolve incompatibilities. If the domain is not defined by a parameter, the concurrent manager uses the value defined for the profile option Concurrent: Conflicts Domain. Lastly, if the domain is not provided by a program parameter and the Concurrent: Conflicts Domain profile option has not been defined, the Standard domain is used. The Standard domain is the default for all requests.

Each request submitted uses parameters that identify the records that it will access. For programs that are defined with incompatibility rules the conflict domain parameter is used additionally. The conflict domain may be set automatically based on such variables as logon ID, set of books, or the organization the user is working in. The conflict domain parameter may in some cases be selected in the parameters field of the Submit Requests form.



Activating and Deactivating Concurrent Managers

(N) Concurrent > Manager > Administer

You can control concurrent managers by activating or deactivating one or all of the managers.

Activate All Managers Simultaneously When you activate the Internal Concurrent Manager, it starts up all the concurrent managers. You can activate the Internal Concurrent Manager from the operating system.

Deactivate All Managers Simultaneously When you deactivate the Internal Concurrent Manager, it shuts down all the concurrent managers. You can deactivate the Internal Concurrent Manager from either the Administer Concurrent Managers window or the operating system.

Activating and Deactivating Individual Managers You can activate individual concurrent managers by using the Administer Concurrent Managers window. The Internal Concurrent Manager must be active for an individual manager to be active. Deactivate individual concurrent managers by using the Administer Concurrent Managers form.

Internal Manager Control Functions:

Verify Concurrent Manager Status Manually executes the process monitoring (PMON) cycle.

Deactivate Concurrent Manager Deactivates the Internal Concurrent Manager and all other managers.

Terminate Requests and Deactivate Manager All running requests running concurrent programs) are terminated, and all managers are deactivated.

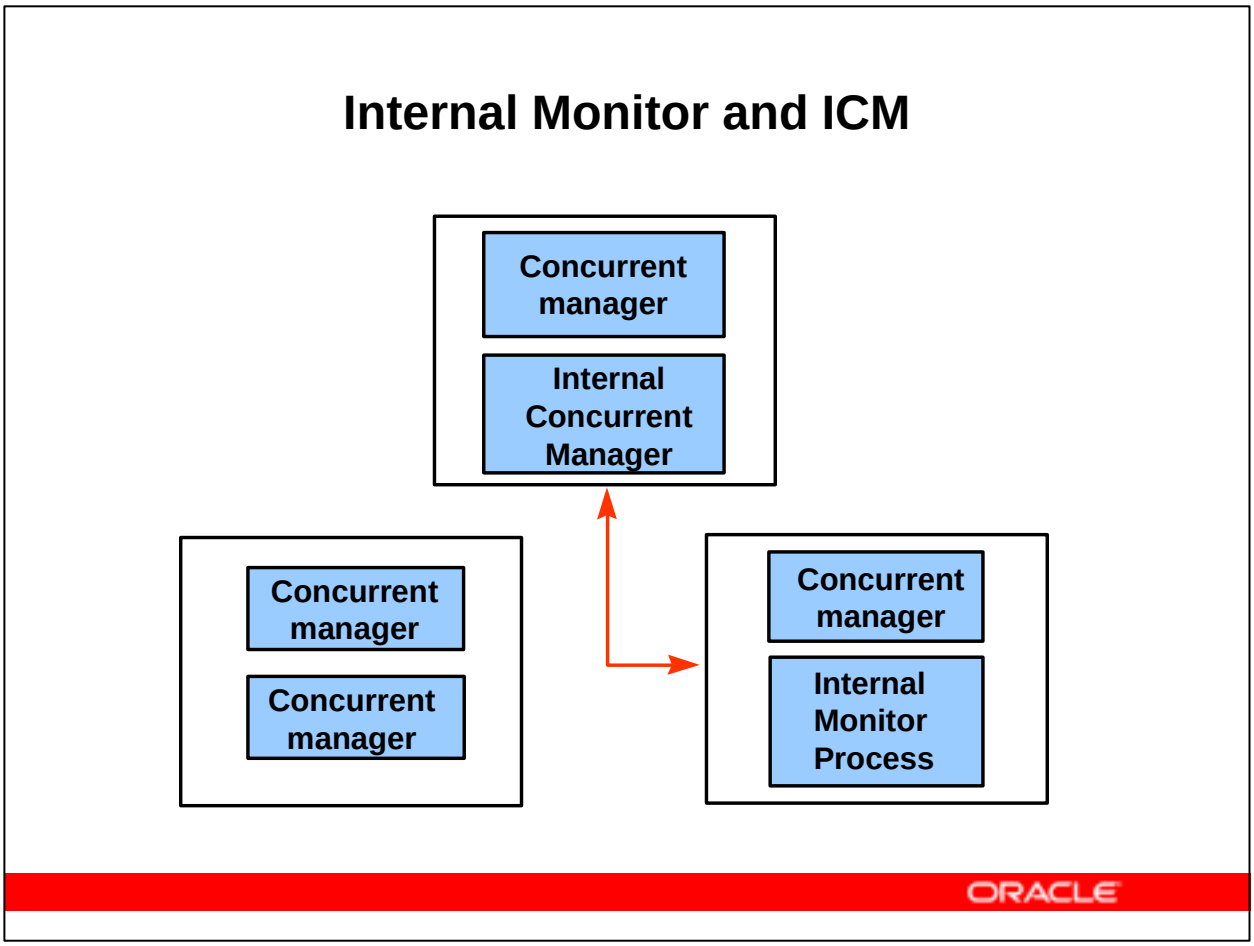
Any Other Manager Control Functions

Activate Concurrent Manager If the manager is defined to work in the current work shift, it starts immediately. Cancels Deactivate Concurrent Manager and Terminate Requests and Deactivate Manager Requests.

Restart Concurrent Manager Internal manager rereads the manager's definition and the rules for concurrent program incompatibilities.

Deactivate Concurrent Manager Deactivates the manager. All requests (concurrent programs) currently running are allowed to be completed before the manager shuts down. A manager will not restart until you select the manager and choose Activate Concurrent Manager.

Terminate Request and Deactivate Manager All running requests (running concurrent programs) handled by the manager are terminated. Once deactivated a manager will not restart until you select the manager and choose Activate Concurrent Manager.



Internal Monitor Processes

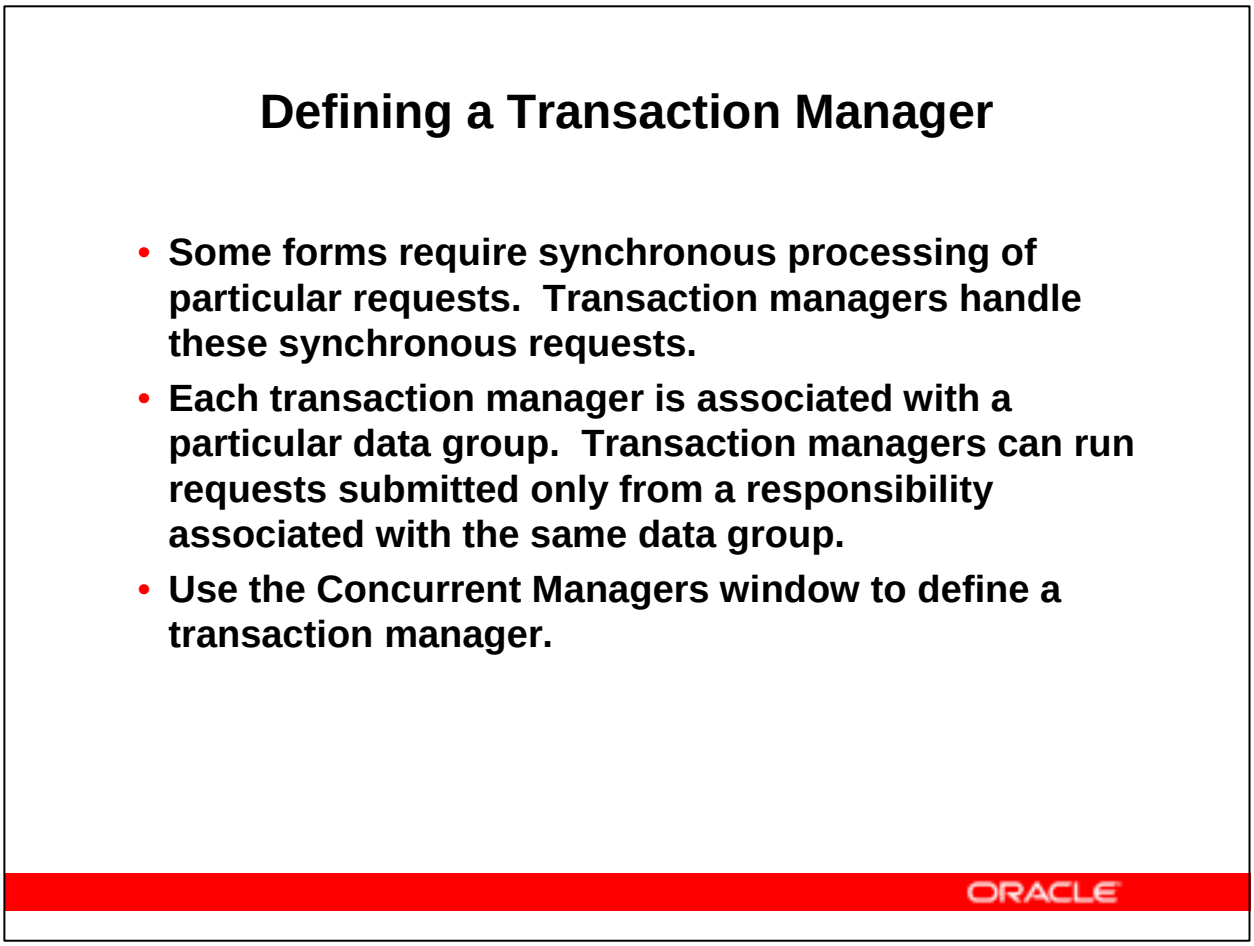
You can provide fault tolerance for the Internal Concurrent Manager by using Internal Monitor Processes.

The Internal Concurrent Manager can run on any node and can activate and deactivate concurrent managers on the same or other nodes. Because the Internal Concurrent Manager must be active at all times, it needs high fault tolerance. To provide this fault tolerance, parallel concurrent processing uses Internal Monitor Processes.

The sole job of an Internal Monitor Process is to monitor the Internal Concurrent Manager and to restart that manager if it fails. The first Internal Monitor Process to detect the failure of the Internal Concurrent Manager restarts the Internal Concurrent Manager on its own node.

Only one Internal Monitor Process can be active on a single node. You decide which nodes have an Internal Monitor Process when you configure your system. You can also assign each Internal Monitor Process a primary and secondary node to ensure its protection.

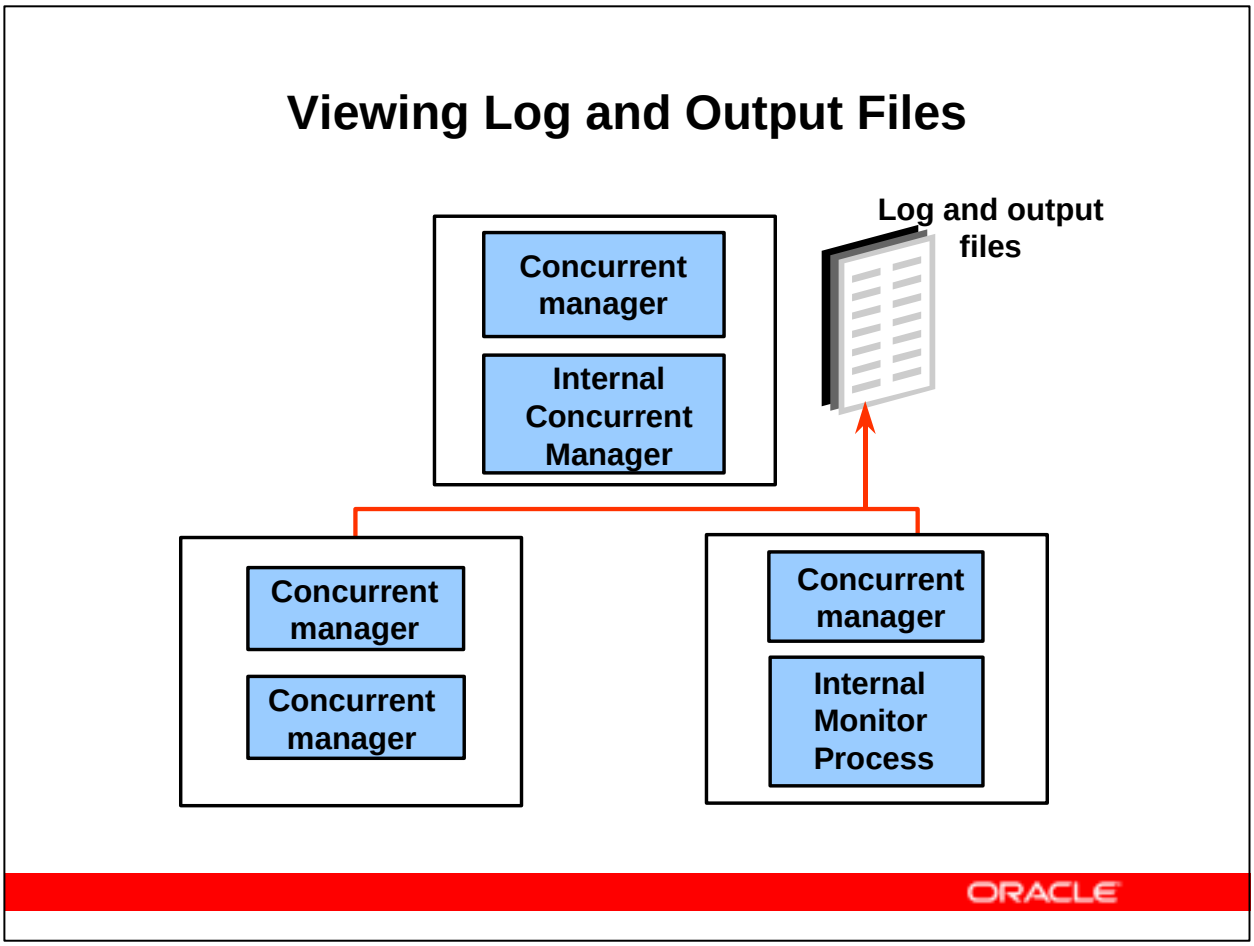
Internal Monitor Processes, like concurrent managers, can have assigned work shifts and are activated and deactivated by the Internal Concurrent Manager.



Creating Custom Transaction Managers

(N) Concurrent > Manager > Define

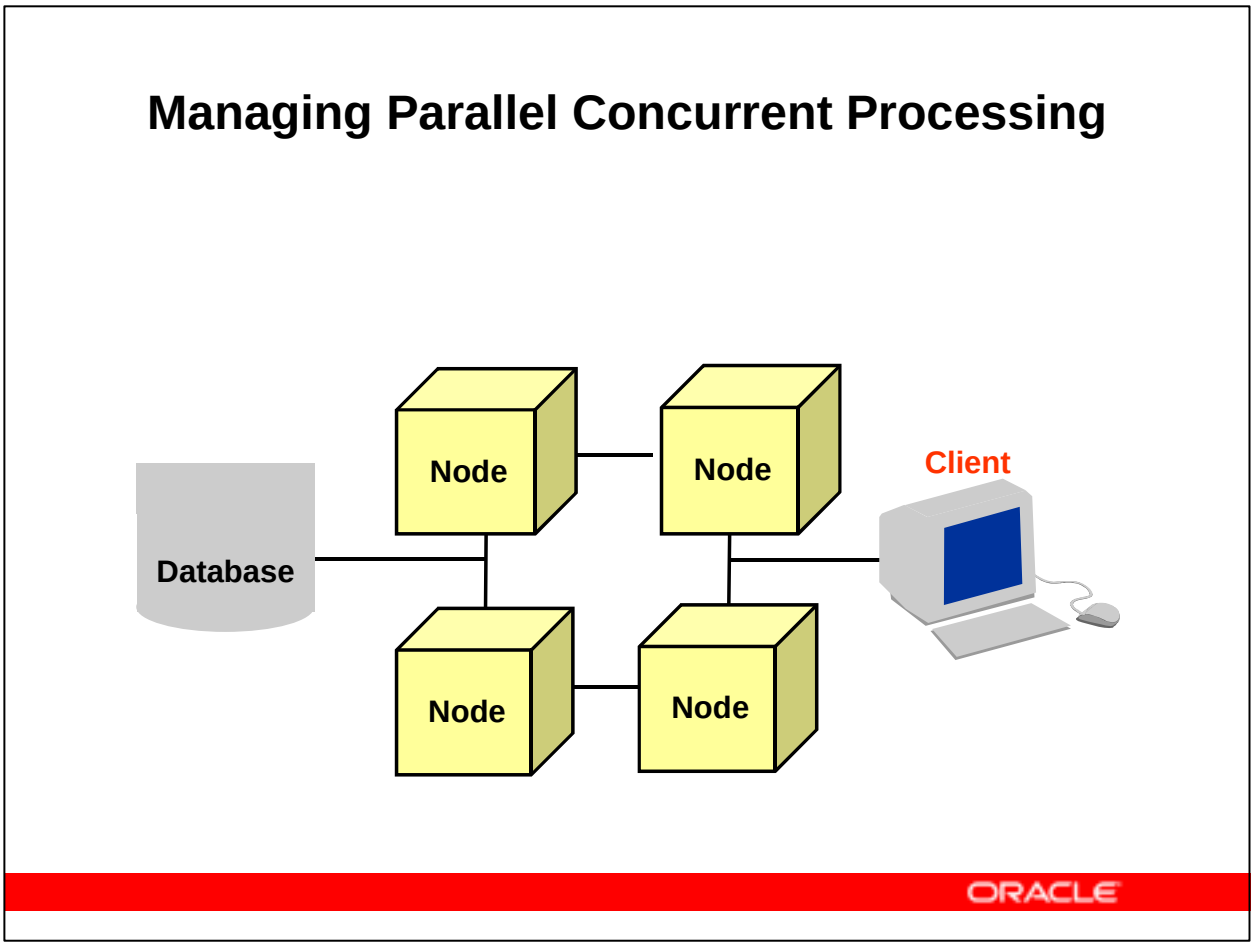
Because forms must be precoded to submit synchronous requests, you do not usually need to create custom transaction managers. If you create custom data groups, you should create custom transaction managers that use your custom data group. This applies only if the responsibilities using your custom data group access the forms submitting synchronous requests. Consult your product documentation. In general, you should create custom transaction managers whenever your custom data groups prevent your seeded transaction managers from operating.



Viewing Log and Output Files

You can review log and output files from any node, regardless of which node the concurrent process runs on. You do not need to log on to a node to view the log and output files from requests run on that node. The concurrent log and output files from requests that run on any node are accessible online from any other node. This capability relies on setup steps taken at install time. For more information, refer to the installation documentation for your platform.

- INVLIBR: Contains immediate concurrent programs used by Oracle Inventory.
- MFCLIBR: Contains immediate concurrent programs used by Oracle Manufacturing.
- PALIBR: Contains immediate concurrent programs used by Oracle Projects.



Managing Parallel Concurrent Processing

You can distribute concurrent processing across multiple nodes in a cluster, massively parallel, or homogeneous networked environment.

Benefits of Parallel Concurrent Processing

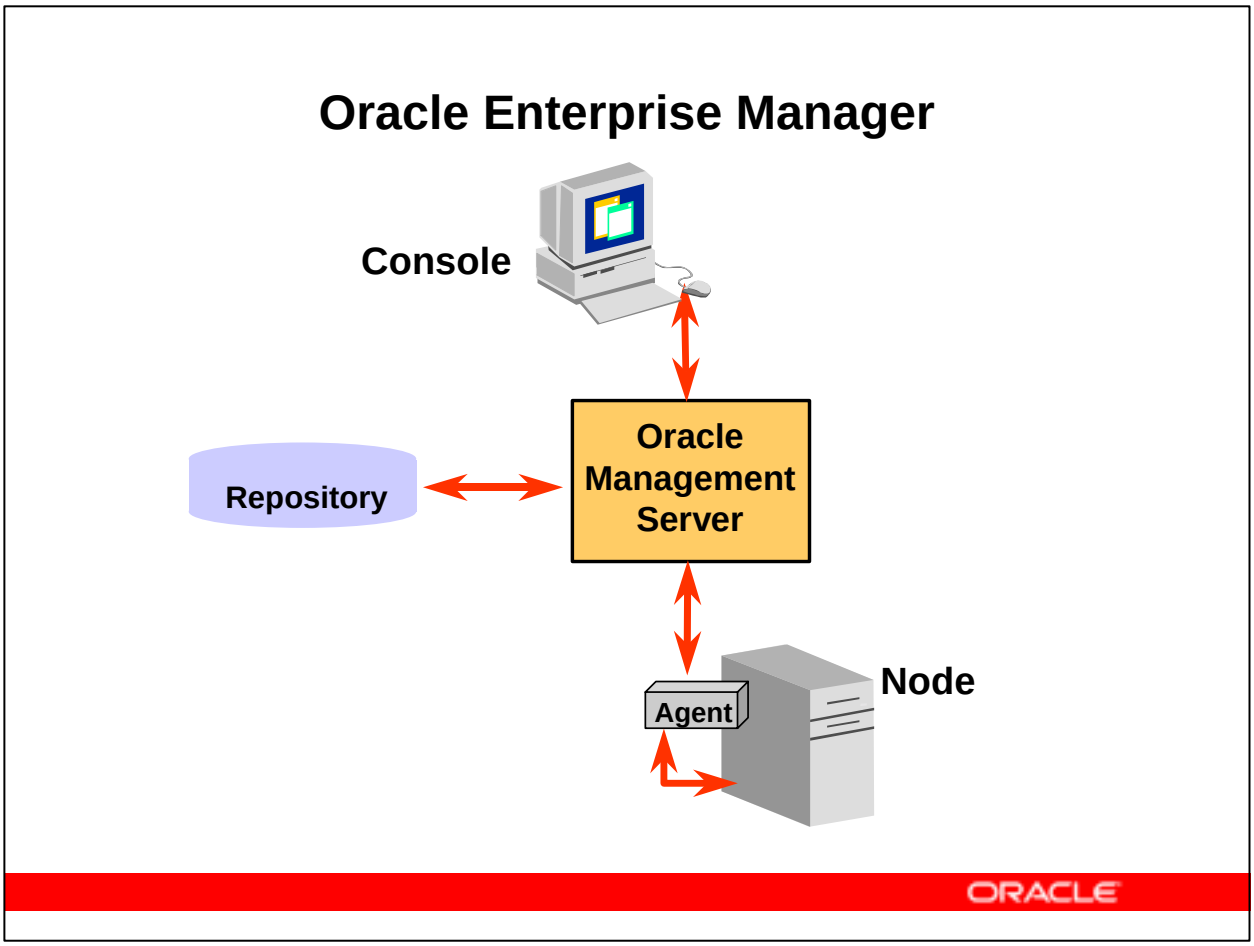
- **High Performance:** Processes can run on multiple nodes to improve throughput.
- **Fault Tolerance:** Processes can continue running on available nodes even when one or more nodes fail.
- **Adaptability:** Integrate with platform-specific batch queue and load-balancing systems to maximize concurrent processing performance on a particular platform.
- **Single Point of Control:** Managers can be administered from any node.

Nodes

- Each node consists of one or more processors (CPUs) and its associated memory.
- Each node has its own memory that is not shared with other nodes.
- Each node operates independently of other nodes except when sharing resources.

Nodes and Concurrent Managers

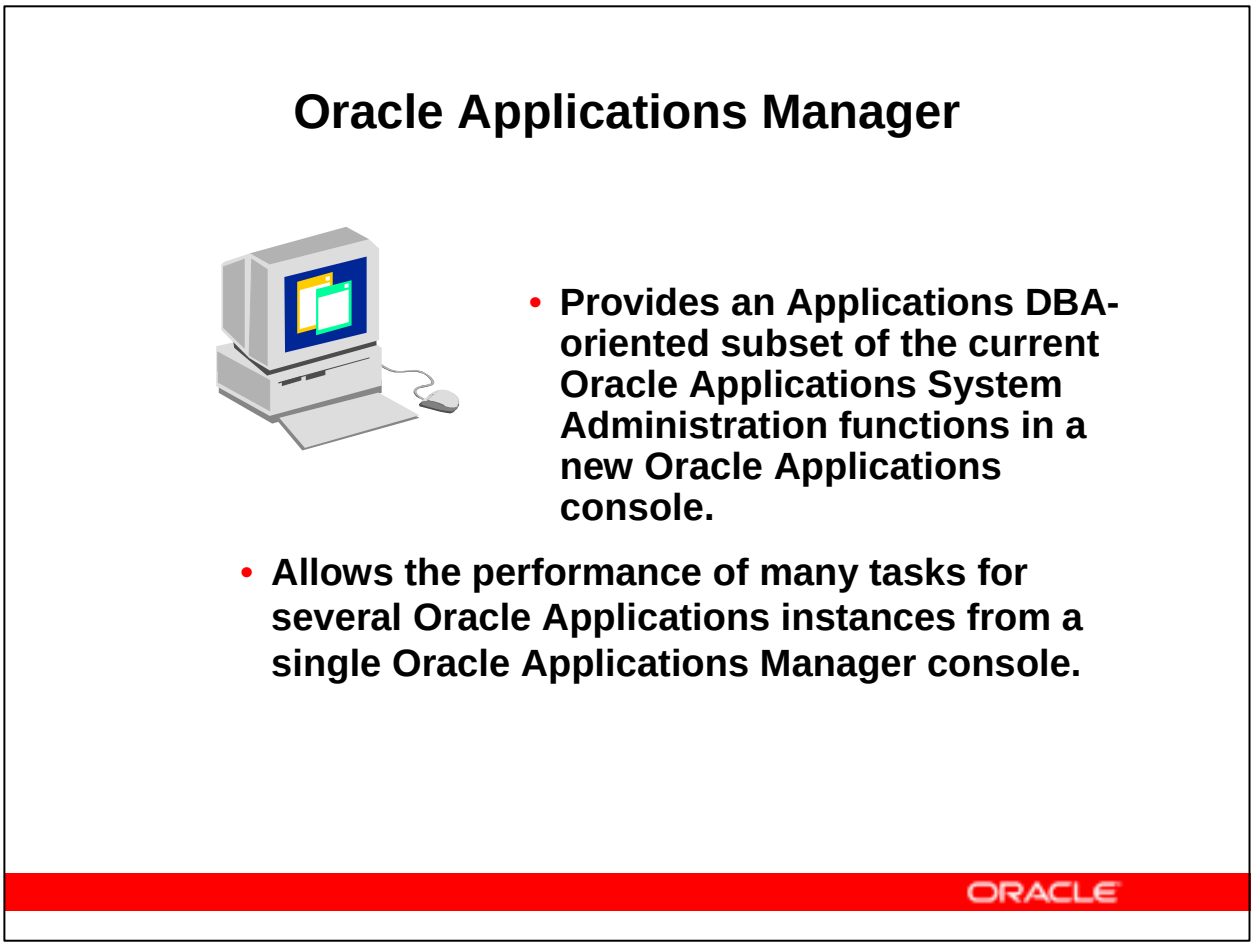
With parallel concurrent processing, one or more managers can run on one or more nodes. You decide where concurrent managers run when configuring the system. You can define any set of specialization rules and apply them across nodes in any way desired.



Oracle Enterprise Manager

Oracle Applications has integrated its concurrent manager administrative interface with Oracle Enterprise Manager, thus enabling administrators to better manage their systems. Oracle Enterprise Manager provides a single point of administration for all available Oracle Applications instances on a system.

Oracle Enterprise Manager combines a central console, agents, common services, and tools to provide an integrated, comprehensive system management platform for managing Oracle products.



The Oracle Applications Manager

The Oracle Applications Manager is integrated with the Oracle Enterprise Manager. The Oracle Applications Manager provides an Applications DBA-oriented subset of the current Oracle Applications System Administration functions. These functions include administration of concurrent managers, processes, and requests.

Note that this new functionality is in addition to the multi-window Oracle Applications forms, and administrators can choose which tools they wish to use. Requests submitted within the standard Oracle Applications windows can be viewed from the Oracle Applications Manager console. Likewise, concurrent managers defined in the console can be accessed from within the Oracle Applications windows.

For more information see:

Oracle Enterprise Manager Concepts Guide

Oracle Enterprise Manager Configuration Guide

Oracle Enterprise Manager Administrator's Guide

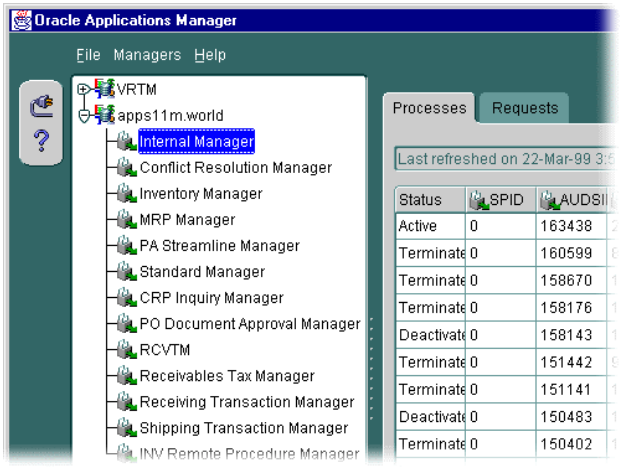
Management Pack for Oracle Applications

Extends Oracle Enterprise Manager to enable the monitoring, diagnosing, and capacity planning of the multitiered Oracle Applications environment. Features include:

- **Discovery and graphical representation of services to be monitored.**
- **Automated data collection and management services.**
- **Central monitoring and administration of remote systems using intelligent agents.**

Management Pack for Oracle Applications

The Oracle Applications Manager applet allows administrators to perform all Concurrent Manager administration tasks from the Enterprise Manager console.



Applications Manager features the same tree and detail view as other Enterprise Manager tools.

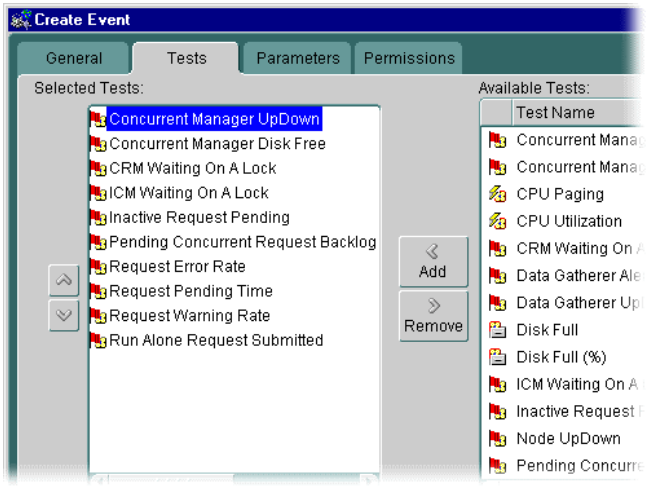
ORACLE

Management Pack Tools

Oracle Applications Manager Console

The Oracle Enterprise Manager console is extended to discover concurrent managers and to notify you should any of the servers go down. You can also define jobs for any of the Oracle Applications subsystems, allowing central administration of a distributed Oracle Applications system.

Advanced Events



The Management Pack for Oracle Applications adds nine Advanced Events to monitor Concurrent Managers for specific conditions that may require action.

ORACLE

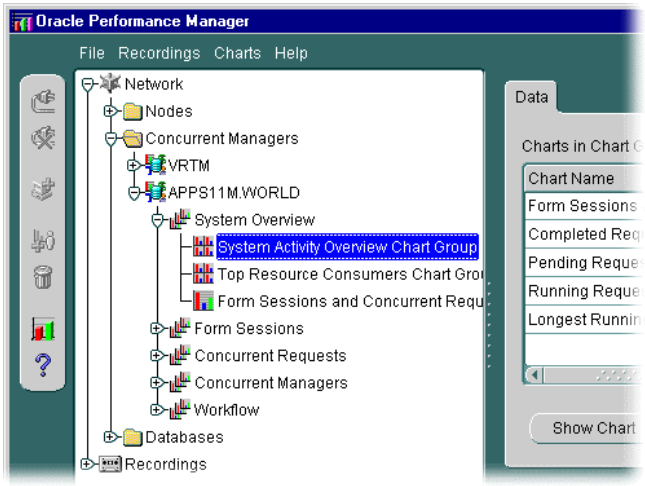
Management Pack Tools

Advanced Events

Oracle Applications Advanced Events is a library of events specific to Oracle Applications provided for lights-out event monitoring and problem detection. Additional lights-out problem resolution is provided with fix-it jobs that are configured to run automatically when an event triggers. These fix-it jobs are either custom-built or chosen from a predefined set provided with the Management Pack.

Oracle Performance Manager

Oracle Performance Manager allows administrators to monitor performance statistics for Oracle Applications Concurrent Managers and host operating systems. Administrators can display this data in different chart formats.



ORACLE

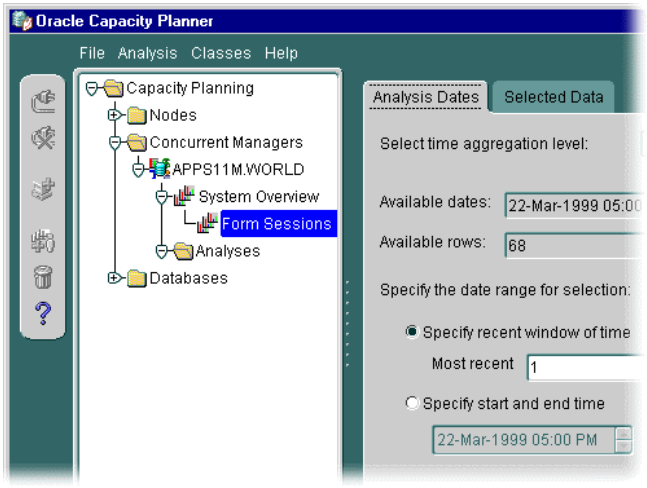
Management Pack Tools

Oracle Performance Manager

A new Oracle Applications data gatherer cartridge feeds data to Oracle Performance Manager, providing you with an extensive array of real-time monitoring charts on all concurrent managers and Forms sessions that are used by your instance.

Oracle Capacity Planner

Oracle Capacity Planner allows administrators to collect and analyze historical performance statistics from Oracle Applications Concurrent Managers and host operating systems to estimate future capacity requirements.



ORACLE

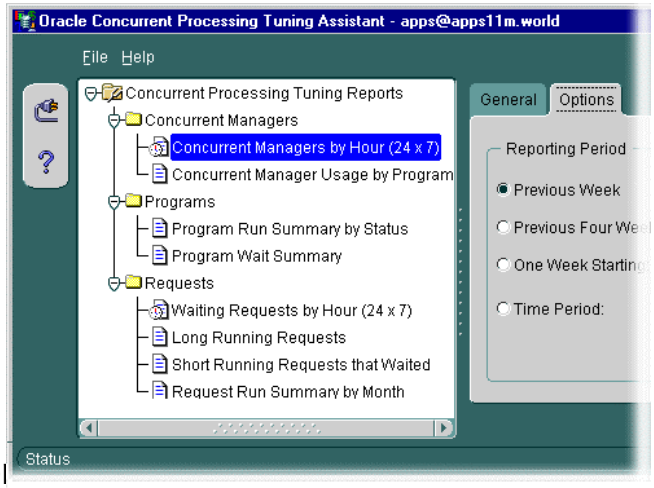
Management Pack Tools

Oracle Capacity Planner

Concurrent manager performance data is gathered over time and fed to Oracle Capacity Planner for analysis of resource consumption and detection of performance anomalies.

Concurrent Processing Tuning Assistant

The Concurrent Processing Tuning Assistant reports historical information about Oracle Concurrent Managers, Concurrent Programs, and Co



You can use these reports to achieve better throughput and performance.

ORACLE

Management Pack Tools

Concurrent Processing Tuning Assistant

Concurrent Processing Tuning Assistant allows you to examine historical processing information about Oracle concurrent processing requests and concurrent managers.

Practice - Administering Concurrent Managers

Overview

The implementation committee wants you to test several aspects of managing concurrent processing. Specifically, they want you to test the following.

- Define a work shift
- Define a concurrent manager
- Assign your work shift to your concurrent manager
- Create specialization rules for your concurrent manager
- Activate your concurrent manager

Tasks

Define your Work Shifts

1. Create a new Work Shift. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Name**: *Your Initials* DAYSHIFT
   - **From**: 08:00
   - **To**: 17:00
   - **From**: Monday
   - **To**: Friday
   - **Description**: Dayshift 8:00 – 5:00

Define your Concurrent Manager

2. Create a new Concurrent Manager. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Manager**: *Your Initials* Specialized Manager
   - **Enabled**: (Checked)
   - **Short Name**: *Your Initials*SPECIAL

- **Application**: Application Object Library
- **Description**: Concurrent Manager to run reports for *Your Initials*
- **Type**: Concurrent Manager
- **Cache Size**: 1
- **Program Library**: FNDLIBR

Assign your Work Shift to your Concurrent Manager

3. Use the following information to assign your work shift to your concurrent manager.
   - **Workshift**: *Your Initials* DAYSHIFT
   - **Processes**: 3
   - **Sleep Seconds**: 60

Create Specialization Rules for your Concurrent Manager

4. Use the information below to define your specialization rules for your Concurrent Manager.
   - **Include/Exclude**: Include
   - **Type**: User
   - **Name**: *Your Initials* User
5. Exclude Your User from the Standard Concurrent Manager. A concurrent program will run in whatever concurrent manager is available unless it has been explicitly excluded from the manager. Exclude your user from the Standard Manager by adding the following specialization rules to the Standard Manager.
   - **Include/Exclude**: Exclude
   - **Type**: User
   - **Name**: *Your Initials* User

Activate your Concurrent Manager

6. In the Administer Concurrent Managers window scroll through the list of managers until you find the manager you defined.
7. Select your manager.
8. Click the “Activate” button.

9. The status will update to Activating.

10. Go to the View Requests window to verify that your request to activate your manager completes successfully.

11. Go to the Submit Request window and submit the “Active Users” or “Active Responsibilities” report to run once every minute for the next five minutes.

12. Go to the Administer Concurrent Managers window. Your concurrent manager will show one request pending.

13. Click on the “Requests” button to view your request in the Concurrent Requests screen.

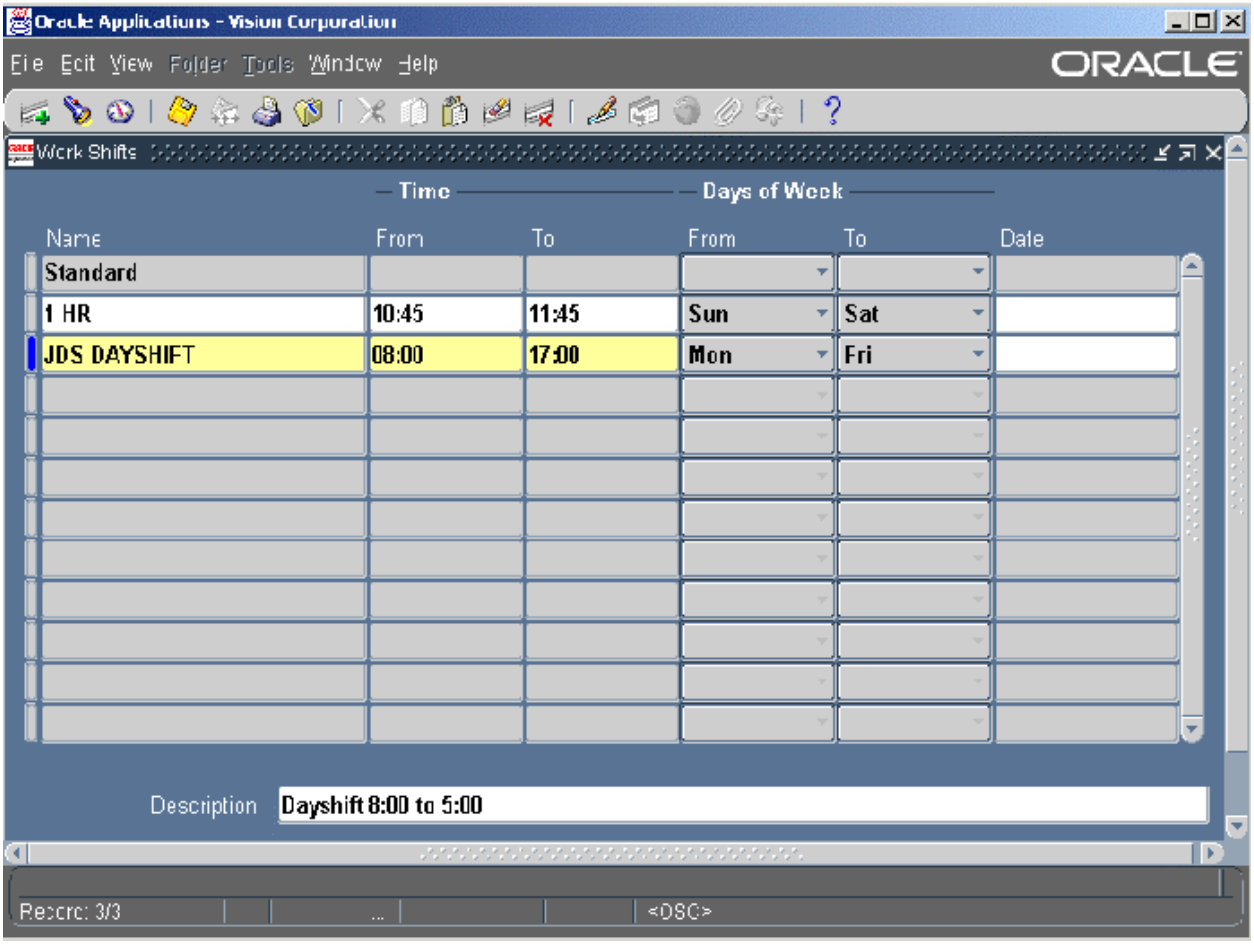
14. Return to the Administer Concurrent Managers window and click on the “Processes” button to view the Processes window.

Solution - Administering Concurrent Managers

Define your Work Shifts

Responsibility = System Administrator

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.
2. Navigate to (N) Concurrent > Manager > WorkShifts.
3. From the menu, select File > New, or click the New icon on the toolbar.
4. Create a new Work Shift.
    - **Name**: *Your Initials* DAYSHIFT
    - **From**: 08:00
    - **To**: 17:00
    - **From**: Monday
    - **To**: Friday
    - **Description**: Dayshift 8:00 – 5:00
5. Save your work. The window should appear similar to the example shown in the slide.



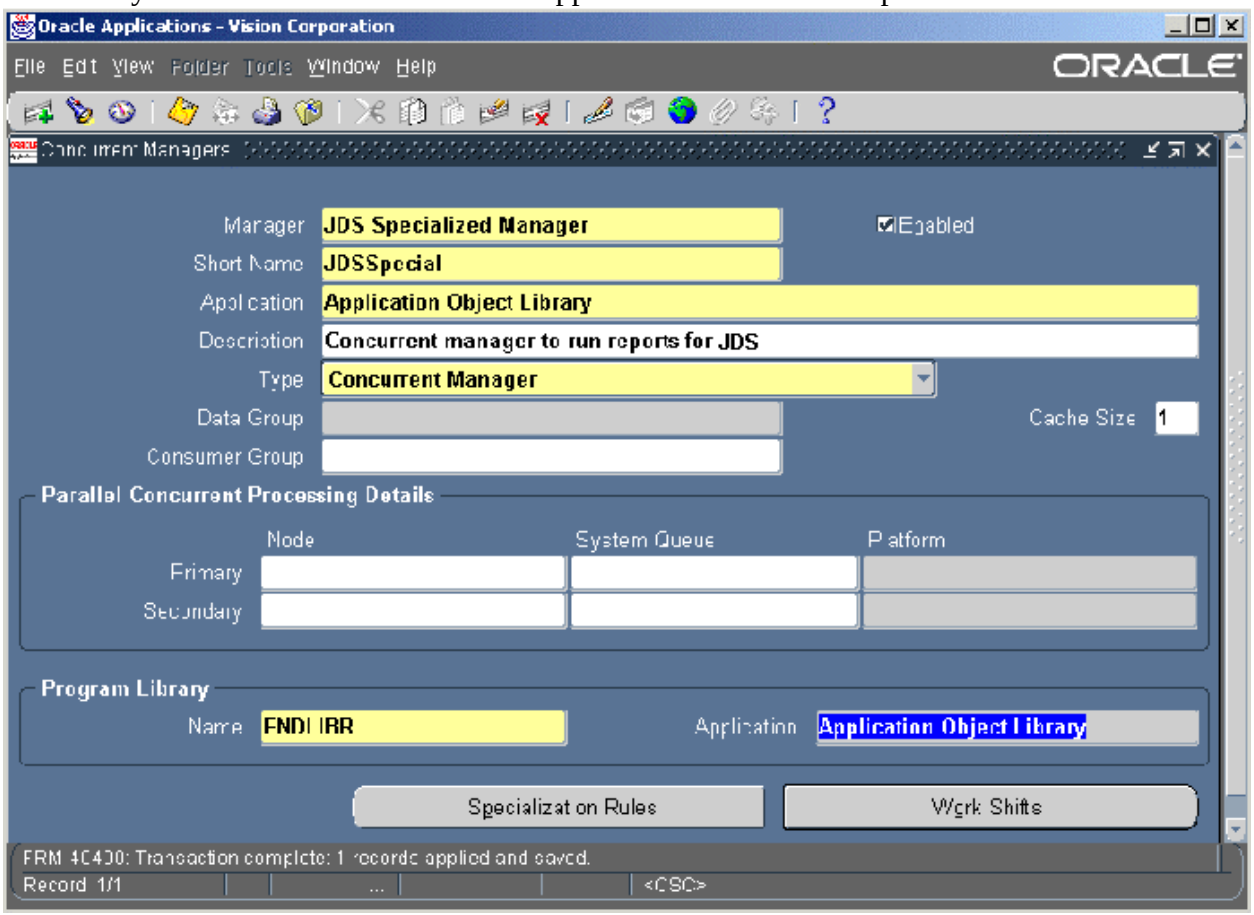
6. Close the form.

Define your Concurrent Manager

7. Navigate to (N) Concurrent > Manager > Define

   - **Manager**: *Your Initials* Specialized Manager
   - **Enabled**: (Checked)
   - **Short Name**: *Your Initials*SPECIAL
   - **Application**: Application Object Library
   - **Description**: Concurrent Manager to run reports for *Your Initials*
   - **Type**: Concurrent Manager
   - **Cache Size**: 1
   - **Program Library**: FNDLIBR

8. Save your work. The window should appear similar to the example shown in the slide.

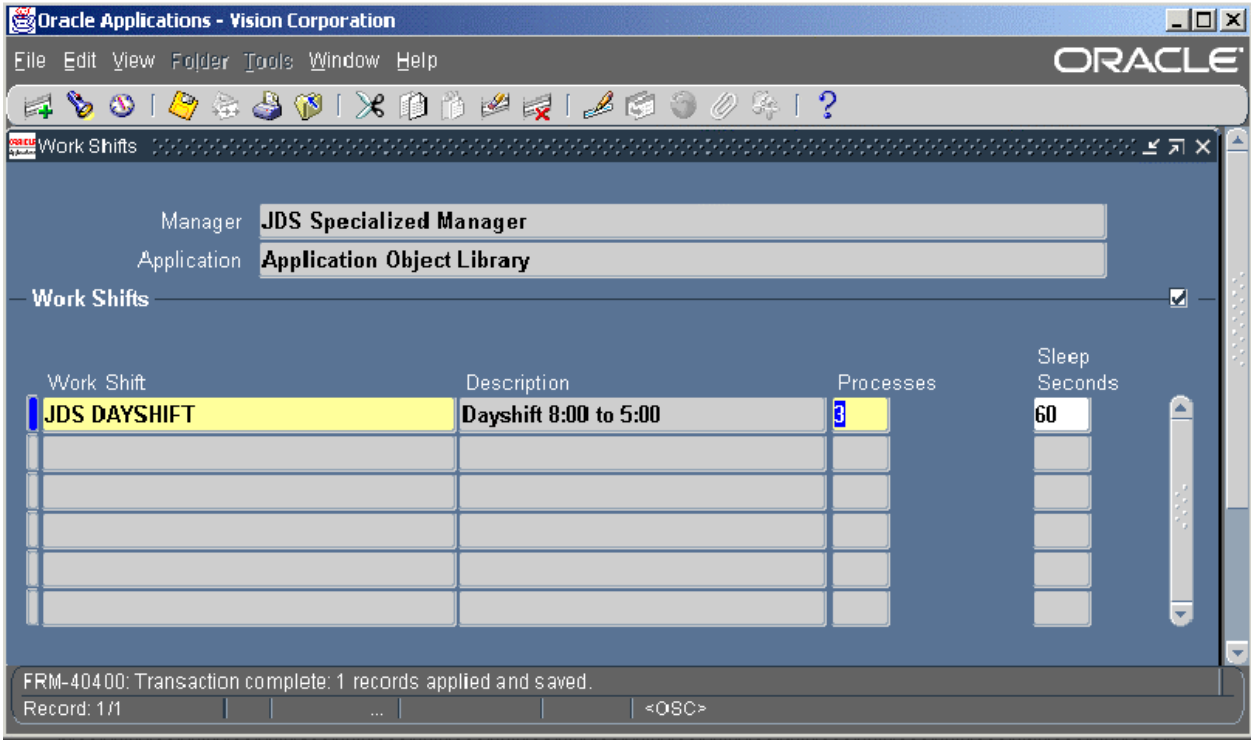


Assign your Work Shift to your Concurrent Manager

9. Click the “Work Shifts” button.

   - **Workshift**: *Your Initials* DAYSHIFT
   - **Processes**: 3
   - **Sleep Seconds**: 60

10. Save your work. The window should appear similar to the example shown in the slide.



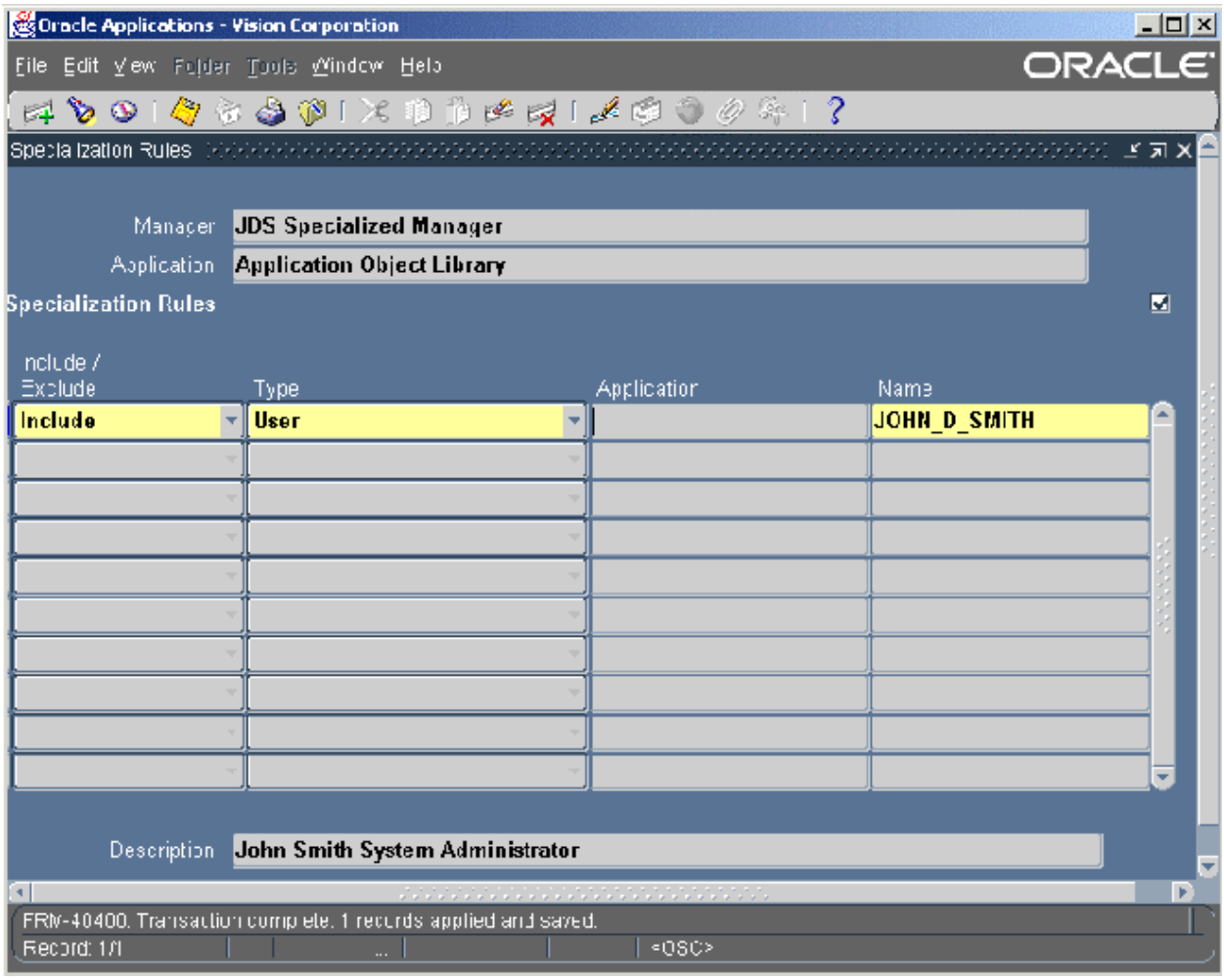
11. Close the Work Shifts window.

Create Specialization Rules for your Concurrent Manager

12. In the Concurrent Managers window, click the “Specialization Rules” button.

    - **Include/Exclude**: Include
    - **Type**: User
    - **Name**: *Your Initials* User

13. Save your work. Your completed form should appear similar to the example shown in the slide.



14. Close the Specialization Rules window to return to the Concurrent Managers window.

15. From the Concurrent Managers window, query up the Standard Manager.

16. Click the “Specialization Rules” button.

    - **Include/Exclude**: Exclude
    - **Type**: User
    - **Name**: *Your Initials* User

17. Save your work. Your completed form should appear similar to the example shown in the slide.

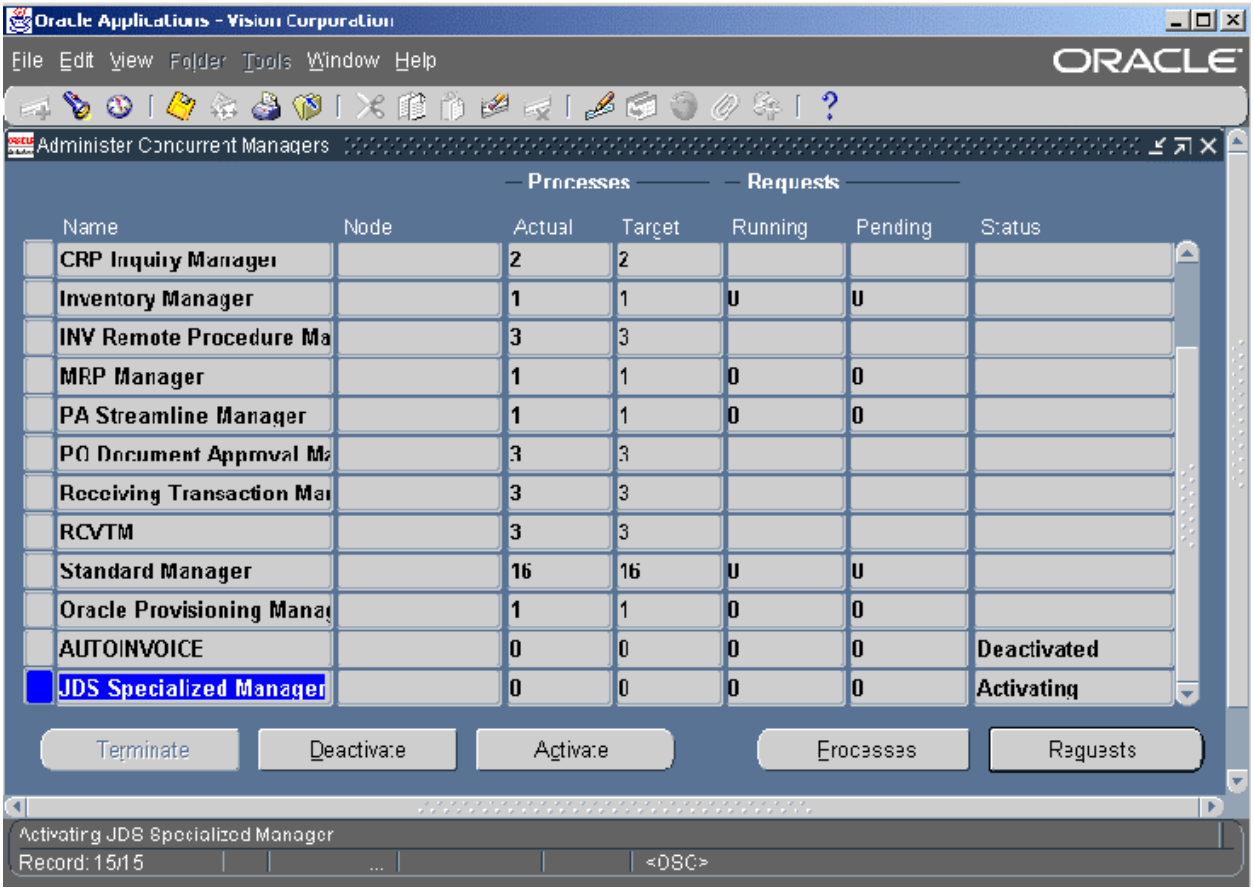
18. Close the form.

Activate your Concurrent Manager

19. Navigate to the Administer Concurrent Managers window: (N) Concurrent > Manager > Administer.

20. In the Administer Concurrent Managers window scroll through the list of managers until you find the manager you defined. Select your manager and click the Activate button.

21. The status will update to Activating. The window will appear similar to the example shown on the slide.



22. Go to the View Requests window (N) Requests > View to verify that your request to activate your manager completes successfully.

23. Navigate to (N) Requests > Run.

24. Choose “Single Request”.

25. In the Submit Request window choose the “Active Users” or “Active Responsibilities” report.

26. Click the “Schedule” button.

27. Choose Periodically.

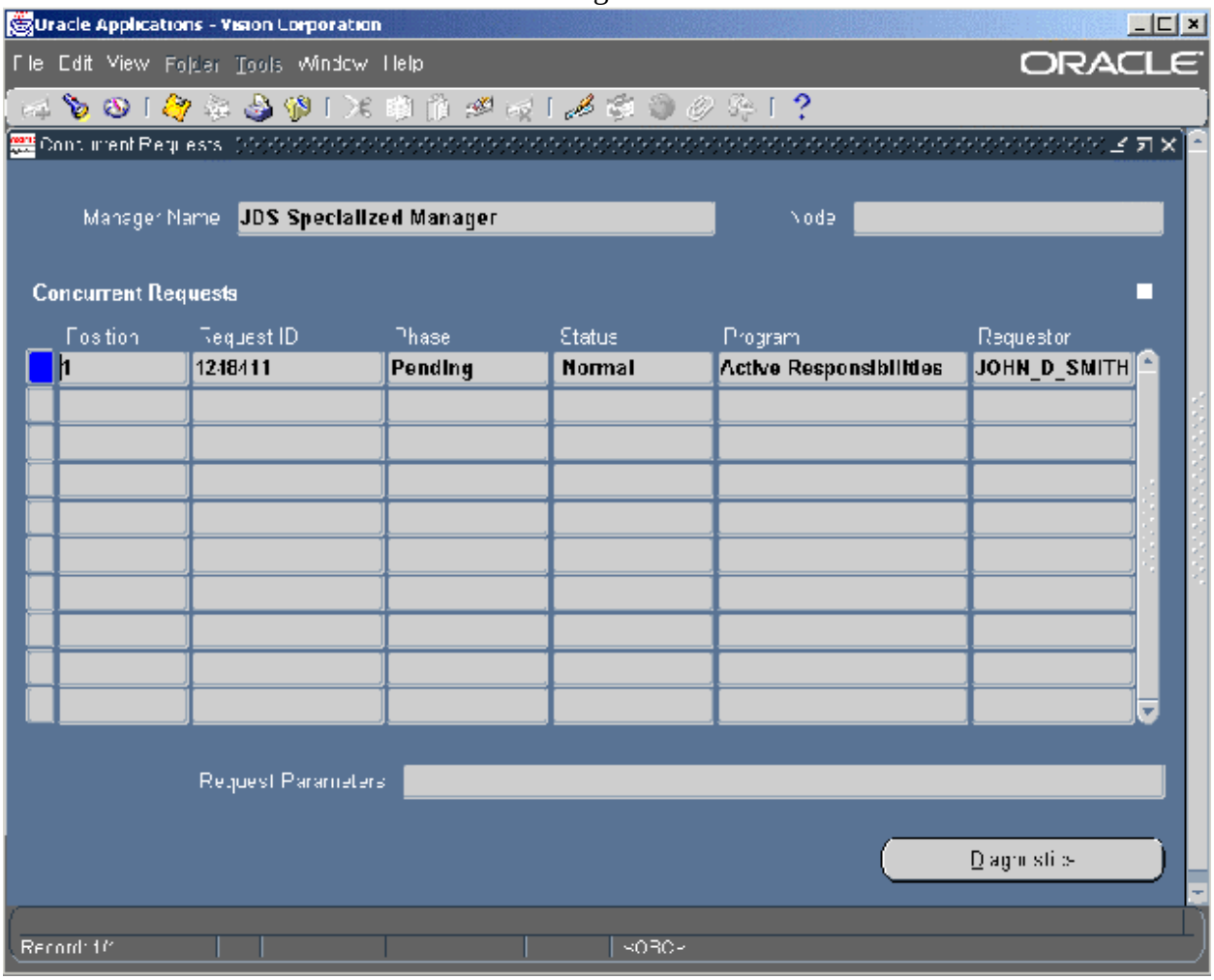
28. Leave the “Start At” time as the current time and enter five minutes from now for the “End At” time.

29. In the “Re-run every” field select 1 and Minute(s) to run the report once a minute for the next five minutes.

30. Click the “OK” button and then click the “Submit” button.

31. Go to the Administer Concurrent Managers window (N) Concurrent > Manager > Administer. Your concurrent manager will show one request pending as shown in the example above. **Note:** You may have to close and reopen the Administer Concurrent Managers window until you “catch” your process running.

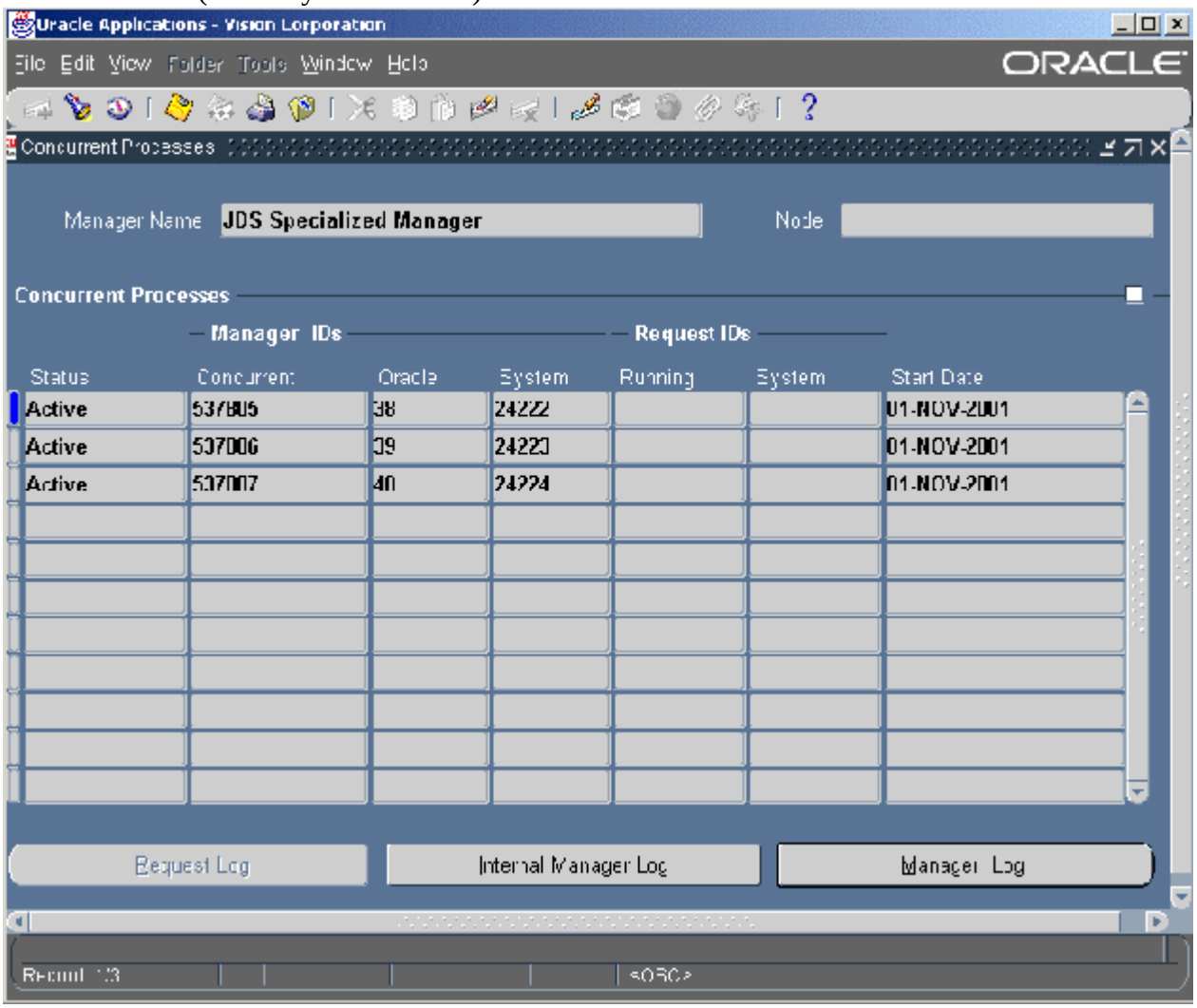
32. Click on the “Requests” button to view your request in the Concurrent Requests screen. Your form will look similar to the following.

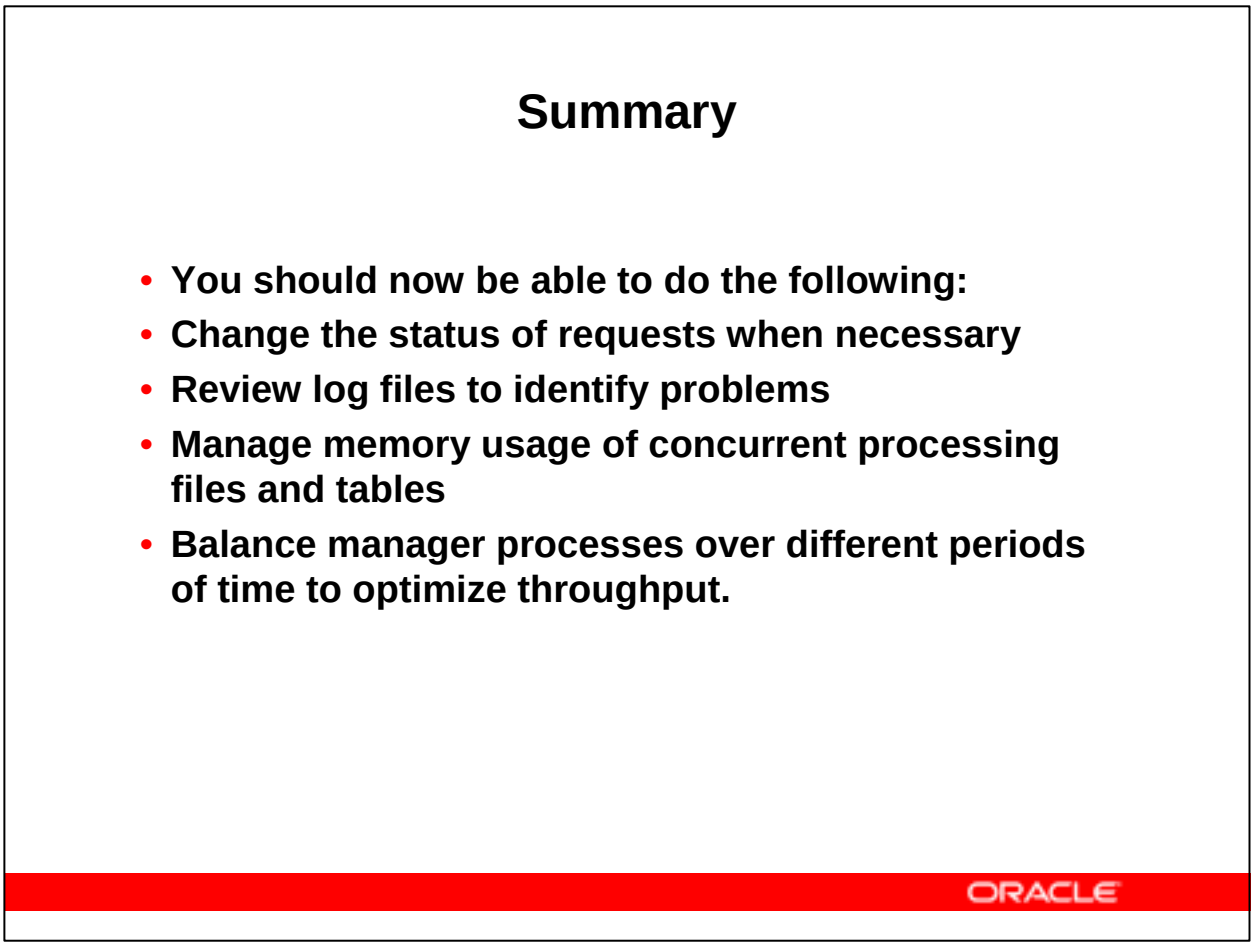


33. Close this window to return to the Administer Concurrent Managers window.

34. Click on the “Processes” button to view the Processes window.

35. In the Concurrent Processes screen note that you see all of the requests that have run in your specialized manager, the Oracle Process ID that was used to run the report, and the UNIX Process ID (in the System column) that was used.





Lesson Summary

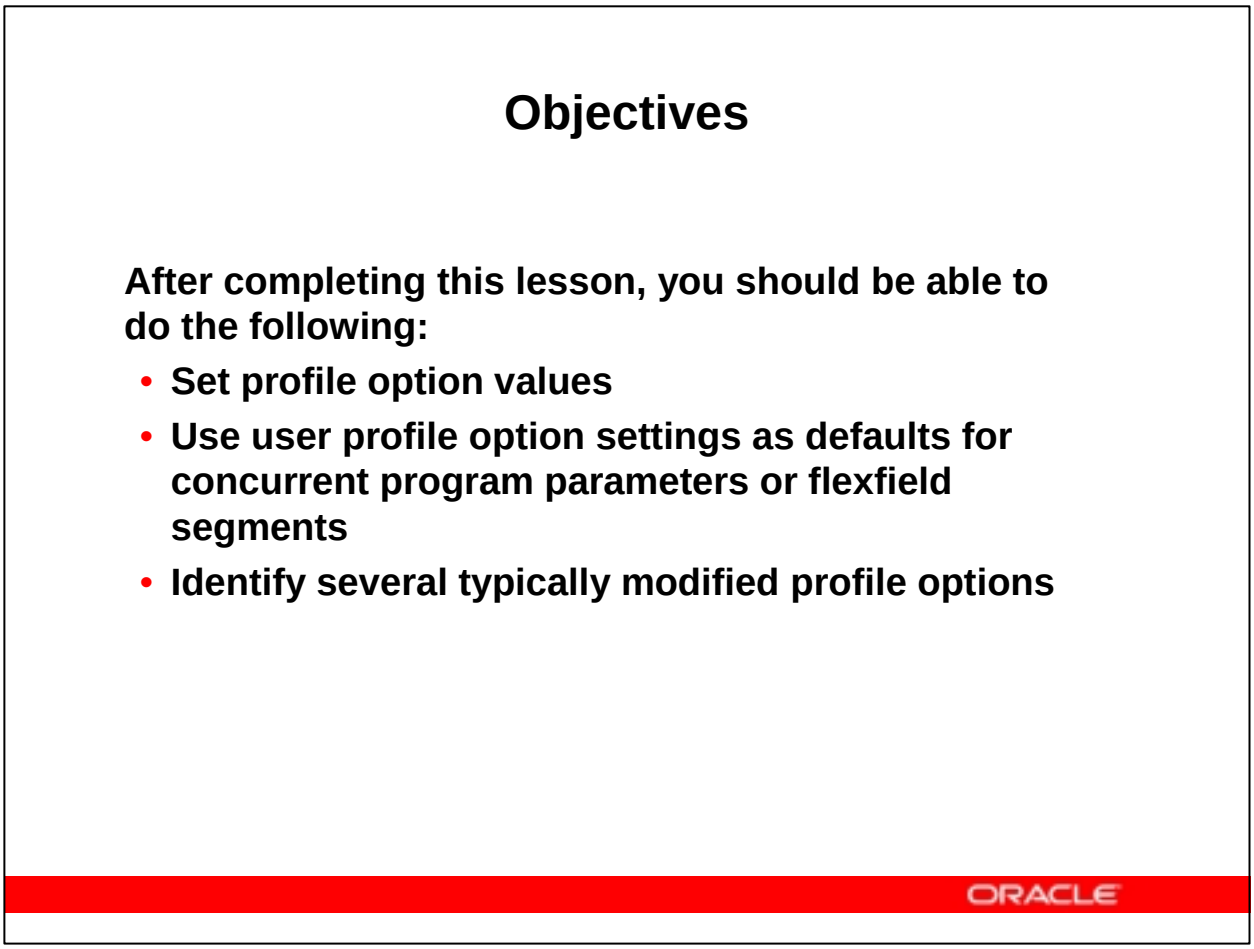
In addition to controlling access to Oracle Applications, monitoring the day-to-day processing of Oracle Applications is a very important job of the system administrator.

The system administrator is the first point of contact when users have, or suspect they have, a problem with either the results or the processing of a request. The system administrator has access to several application log files containing diagnostic information that can be of use in correcting problems.

The system administrator is also responsible for balancing workload to maintain optimum performance for users. He or she does this by creating concurrent managers to process user requests. Managers can be specialized to operate only during certain times and to process only certain types of requests. System administrators also monitor and control the activity of managers in a parallel environment.

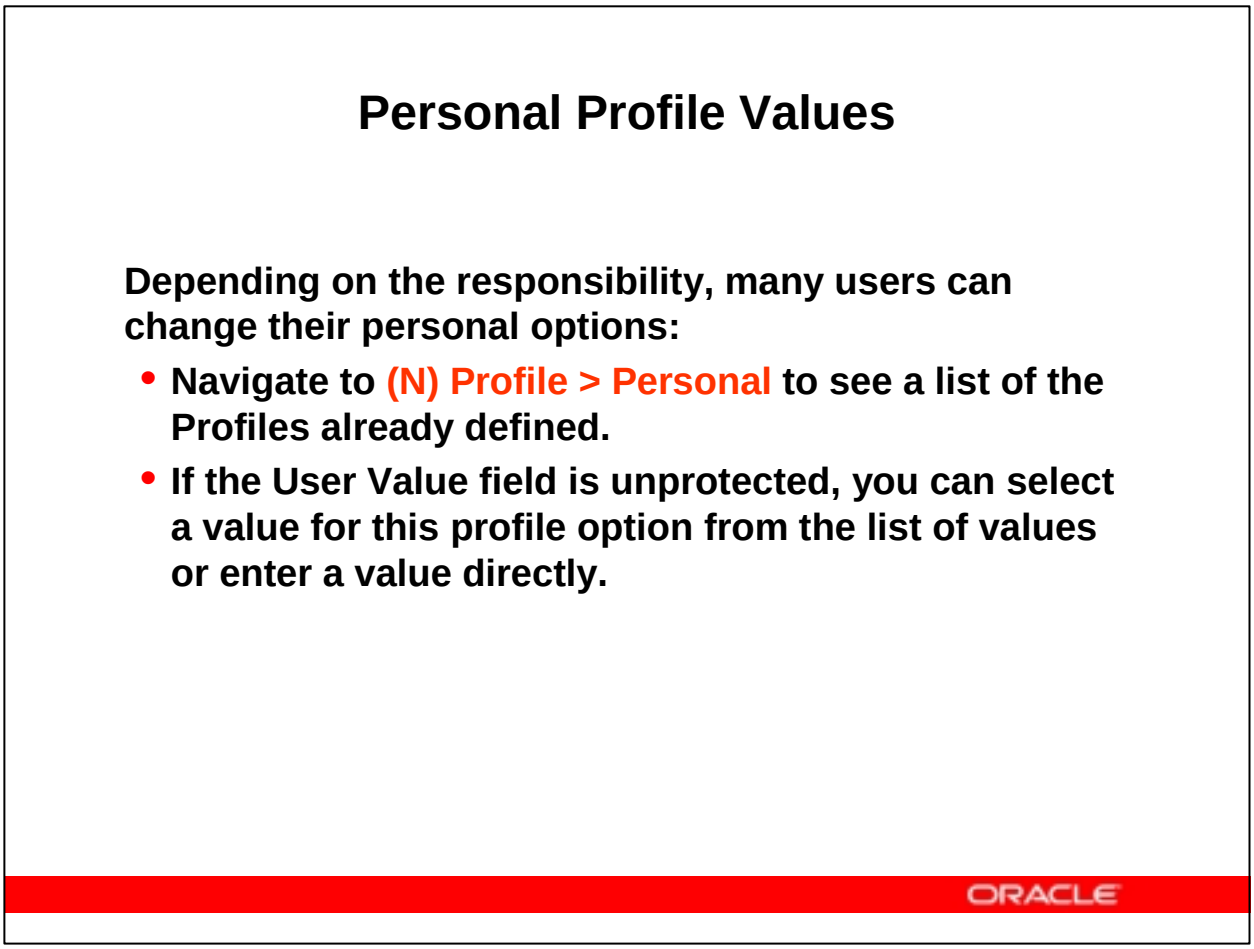
Managing Profile Options

Chapter 5



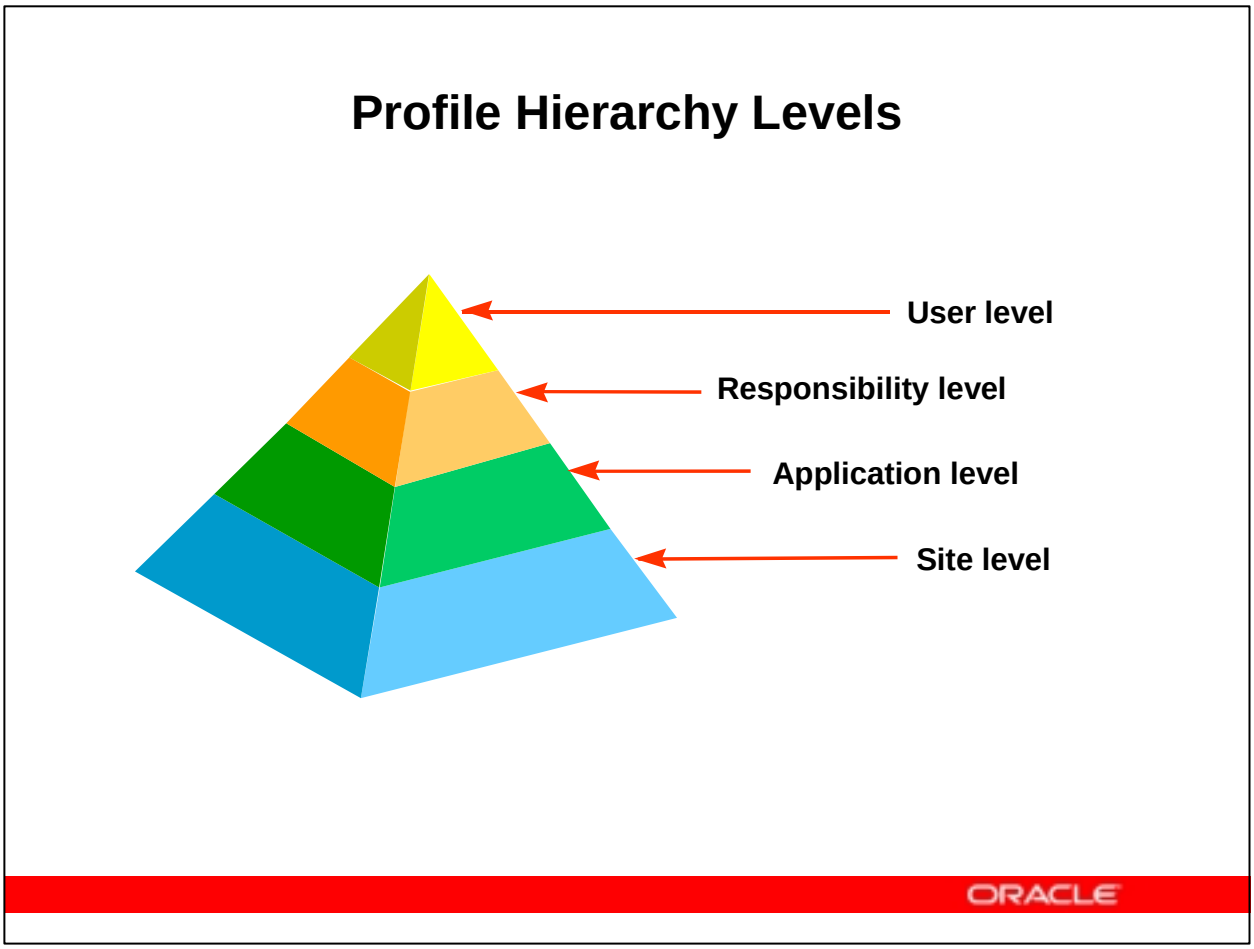
Lesson Aim

As a system administrator, you control various options in Oracle Applications that determine how your applications look and feel. These options can also control how each application operates. This lesson shows you how to specify these values.



For further information on using the Personal Profile Values window see:
(Help) Oracle Applications User's Guide > Profile Options >
Setting Your Personal User Profile

Profile Hierarchy Levels



Setting Profile Hierarchy Levels

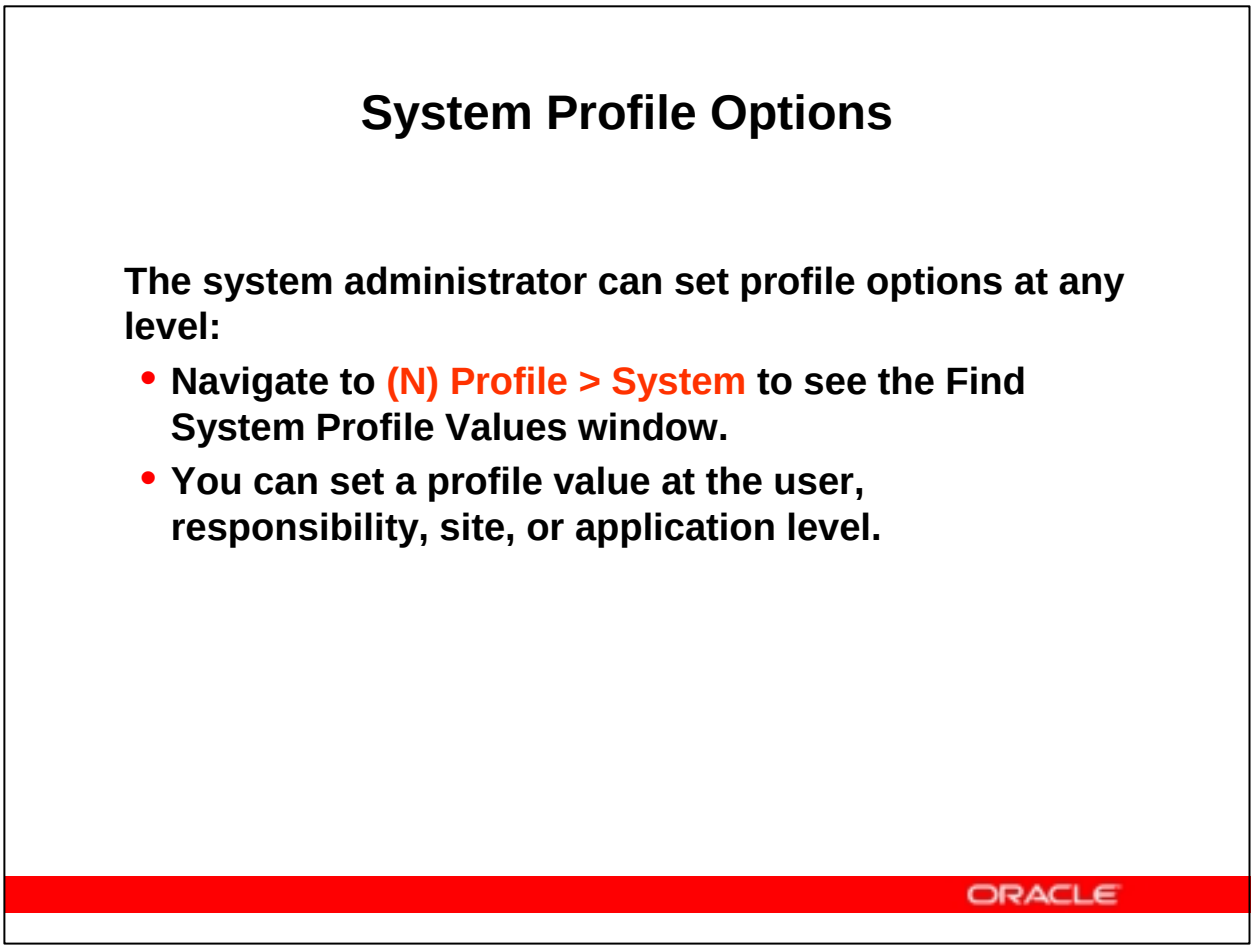
Setting a user profile affects application users across one of four different levels.

Site Level Site-level settings apply to all users at an installation site. To display the name of your installation site, select About Oracle Applications from the Help menu.

Application Level Application-level settings apply to all users of the specified application. For example, a profile could be set that applies to all Oracle General Ledger users. Profile options that can be set at the application-level override options set at the site level.

Responsibility Level Responsibility-level settings apply to all users currently signed on under the responsibility. For example, a profile could be set that applies to all users of the Oracle General Ledger GL budget supervisor responsibility. Profile options that can be set at the responsibility level override options set at the site and application levels.

User Level User-level settings apply to individual users, identified by their application usernames. For example, a user profile could be set that applies only to user JDoe. Profile options set at the user level override all other options.



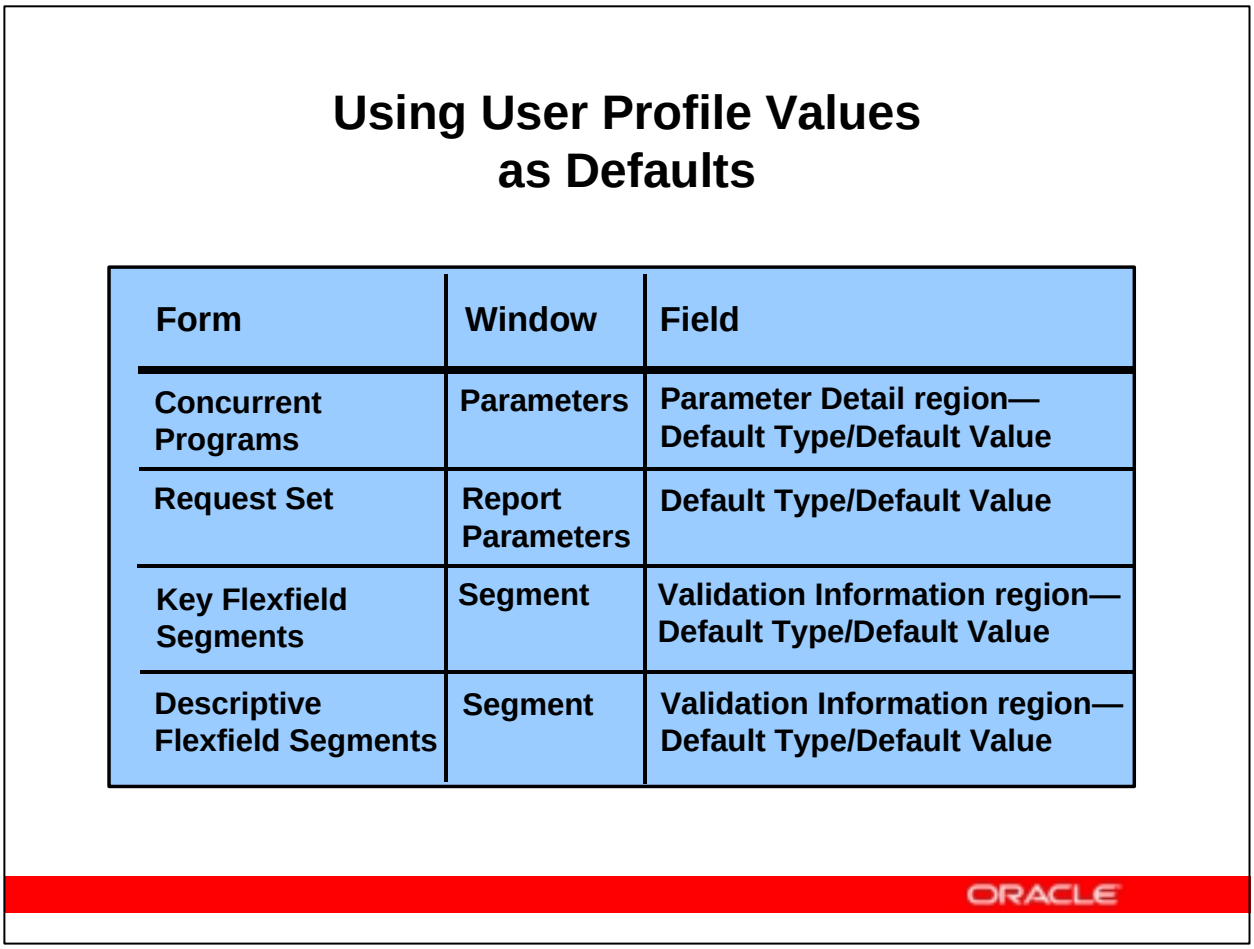
Setting System Profile Options

If you choose to set a value at the Application, Responsibility, or User level, you must also specify the particular Application, Responsibility, or User. Any values defined at a lower level than the level chosen will also be displayed.

For a complete description of the fields in the System Profile Values window see:

(Help) Applied Technology > Oracle Applications System Administration >

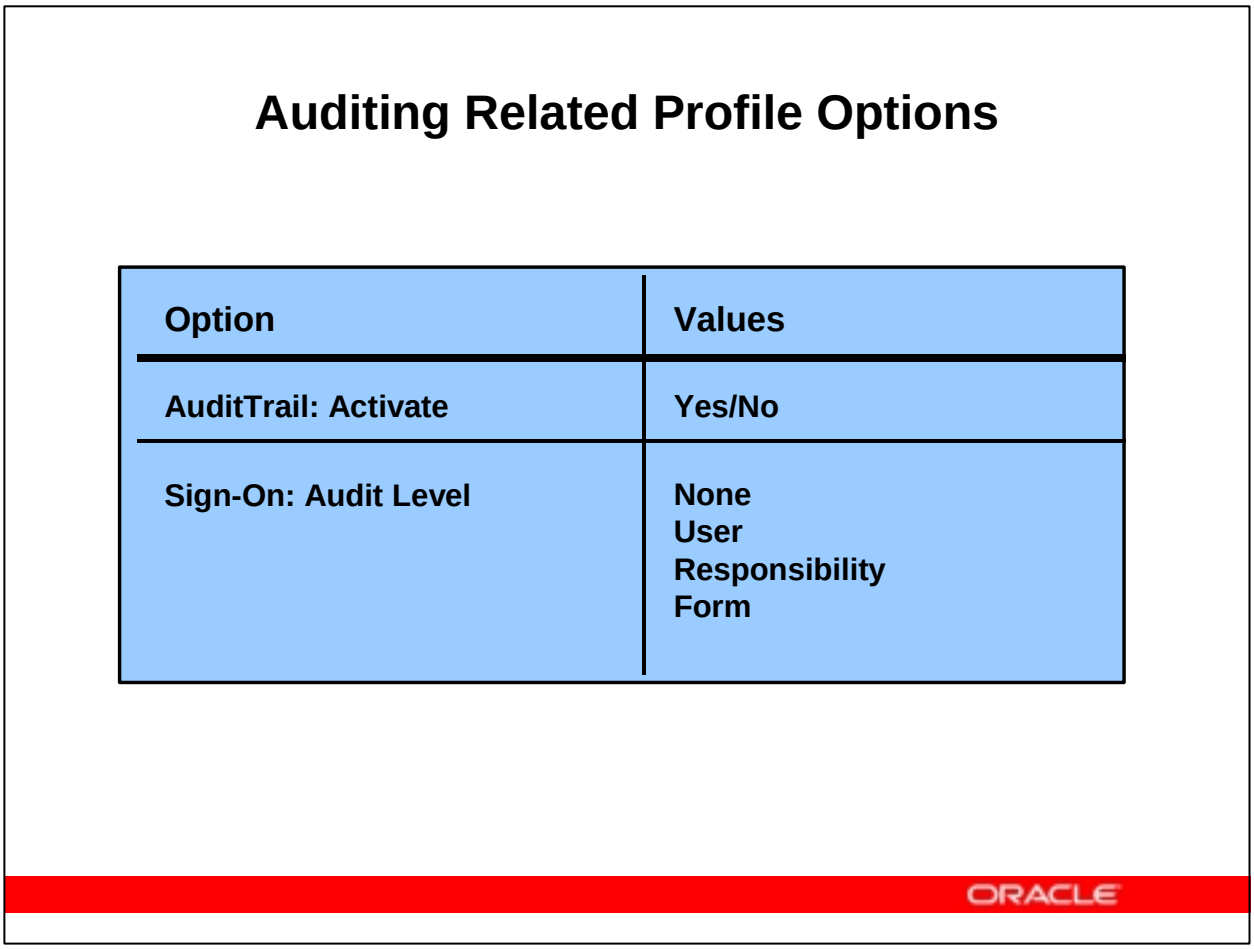
Setting Profile Options > System Profile Values Window



Default User Profile Values

After profile values have been defined, they can be referenced elsewhere in Oracle Applications. The figure shows the locations where profile values can be used to supply default processing values.

- Enter the setting of a profile option as a default value by selecting Profile as the default type, and then enter the internal name (not end-user name) of the profile option as the value in the Default Value field.
- Look up the internal name of the profile option in the User Profiles appendix of the *Oracle Applications System Administrator's Guide*.



How to Use Profile Options Related to Auditing

These two profile options are used to enable auditing within Oracle Applications.

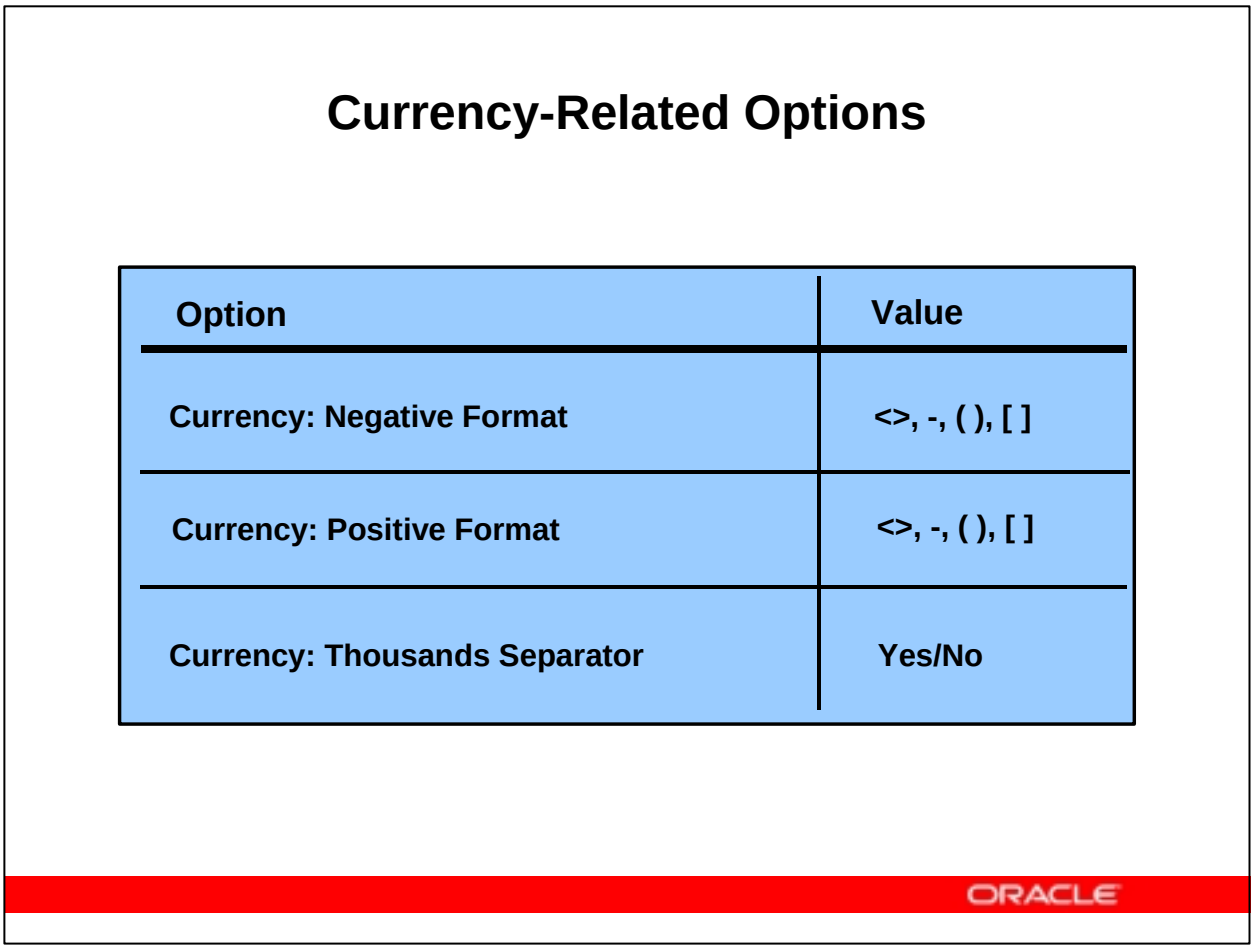
AuditTrail: Activate

This option enables auditing of changes to database tables. AuditTrail tracks which rows in a database table or tables were updated, at what time, and which user was logged in using the form or forms.

Sign-On:Audit Level

This option allows you to select a level at which to audit users signing on to Oracle Applications. Choose from four audit levels which increase in functionality:

- None – default value. No users will be audited.
- User – tracks who signs on to your system, the times the users log on and off, and the terminals used.
- Responsibility – performs the User-level audit as well ass tracking the responsibilities the users choose and how much time spent as each responsibility.
- Form – performs the Responsibility-level audit and also tracks the forms the users choose and how long each form is used.



How to Use Currency-Related Options

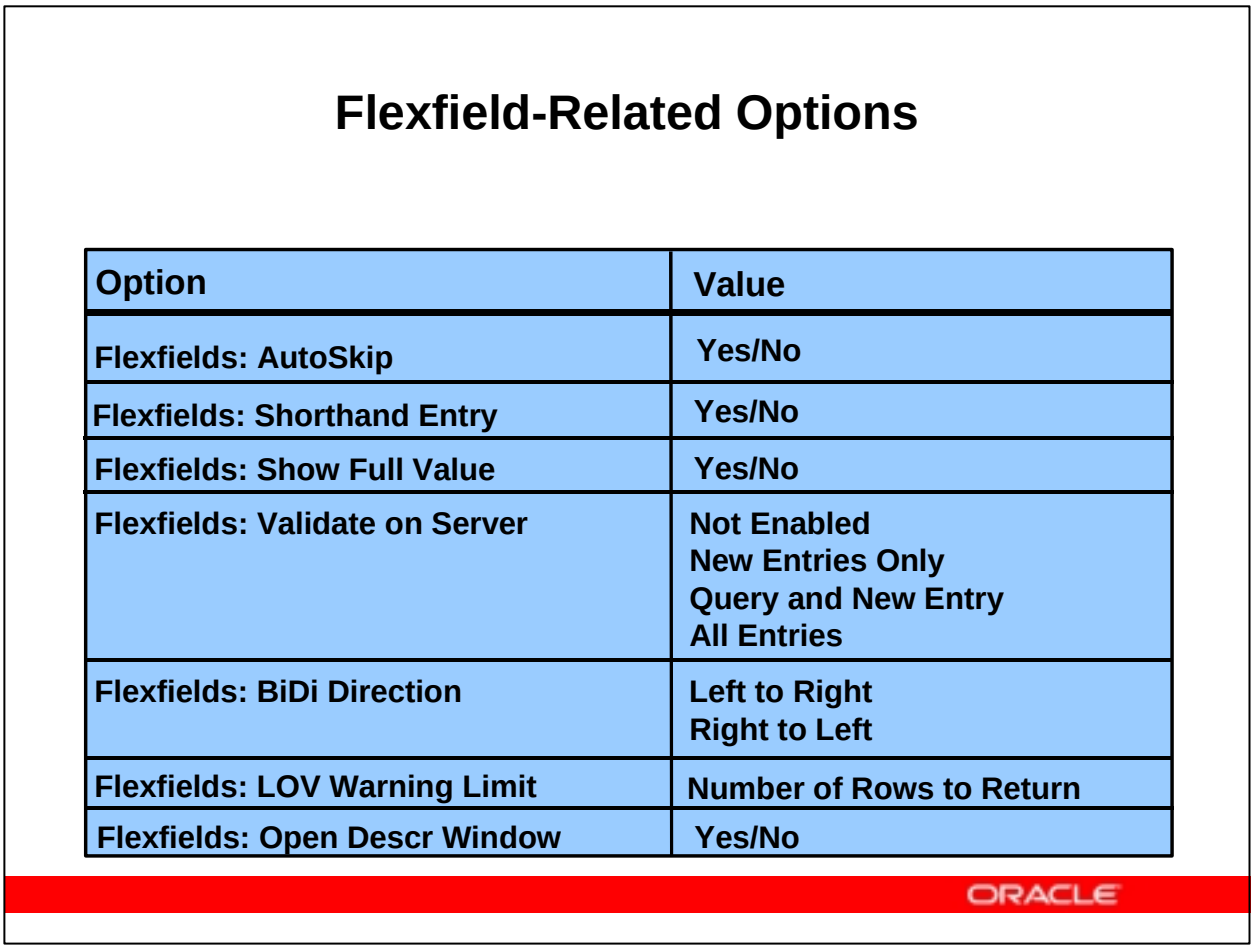
The options shown in the figure control the editing of monetary fields.

Currency: Negative Format

This option enables the user to select the indicators for a negative amount.

Currency: Thousands Separator

This option enables the user to request that a comma be inserted appropriately within a number of a thousand or more.



How to Use Flexfield-Related Options

The options shown on the figure control the behavior of flexfield processing.

AutoSkip

As soon as you enter a valid value into a flexfield segment, AutoSkip automatically positions your cursor in the next segment.

Shorthand Entry

This option controls the behavior of shorthand entry of flexfields if shorthand entry has been defined.

Show Full Value

If both shorthand entry and a complete alias have been defined for a flexfield, and this option is set to no, the complete flexfield window does not open when the alias is entered. If this option is set to Yes, the window always opens.

Validate on Server

Set this option to Yes to enable server-side validation of key flexfields. This improves performance when using key flexfields over a wide-area network.

BiDi Direction

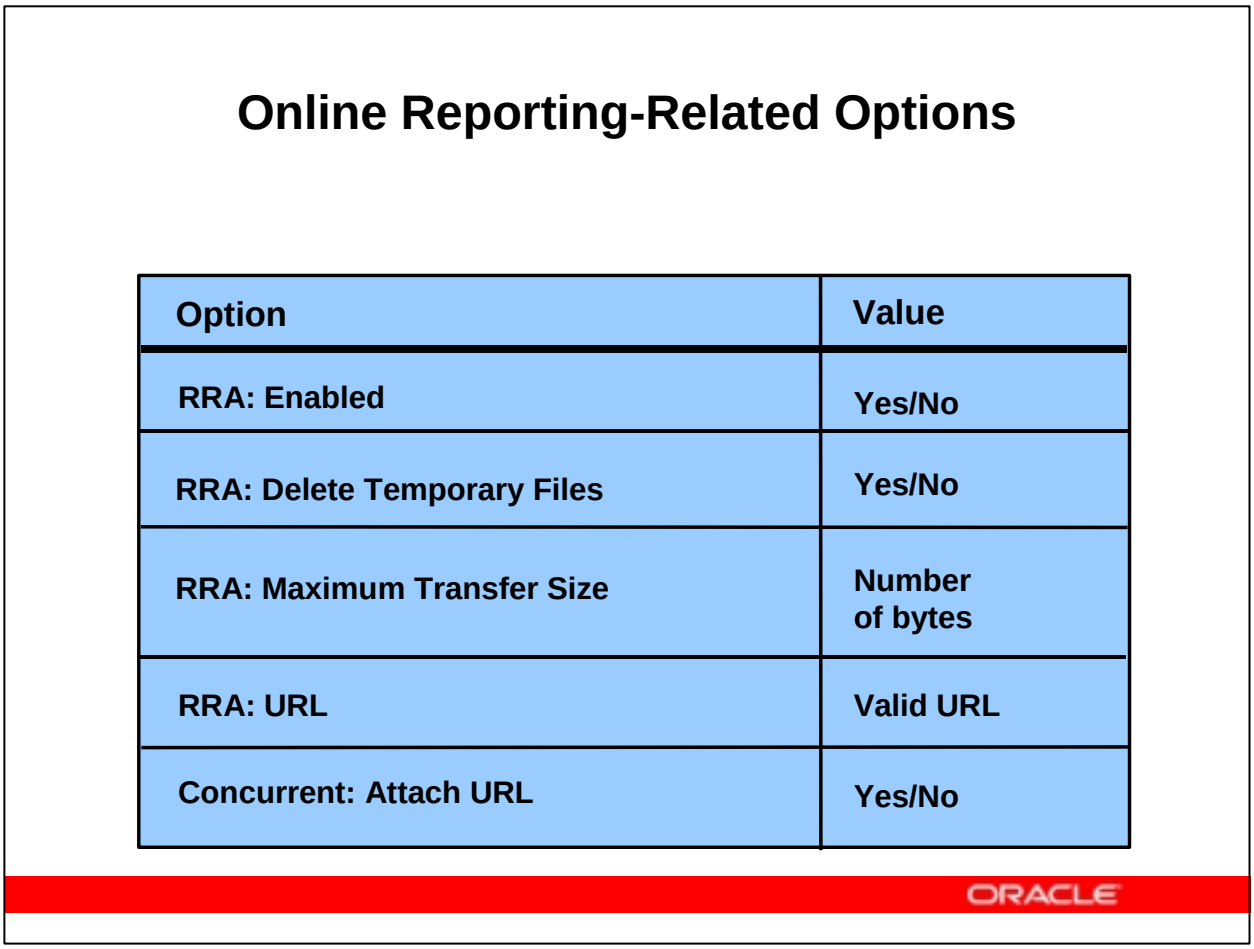
This option controls the appearance of the flexfields window in Applications running in Semitic languages.

LOV Warning Limit

Sometimes an LOV can take a very long time to run if there is a very significant amount of data in it. Set this option to the number of rows to be returned before you are asked to continue retrieving the entire list.

Open Descr Window

This option allows you to control whether a descriptive flexfield window automatically opens when you navigate to a customized descriptive flexfield.



How to Control Output Processing at the System Level

The options shown in the figure control output processing at the system level.

RRA: Enabled

Set this user profile to Yes to use the Report Review Agent to access files on concurrent processing nodes.

RRA: Delete Temporary Files

When users use a custom editor to view a concurrent output or log file, the Report Review Agent makes a temporary copy of the file on the client. Setting this option to Yes automatically deletes these files when the user exits Oracle Applications.

RRA: Maximum Transfer Size

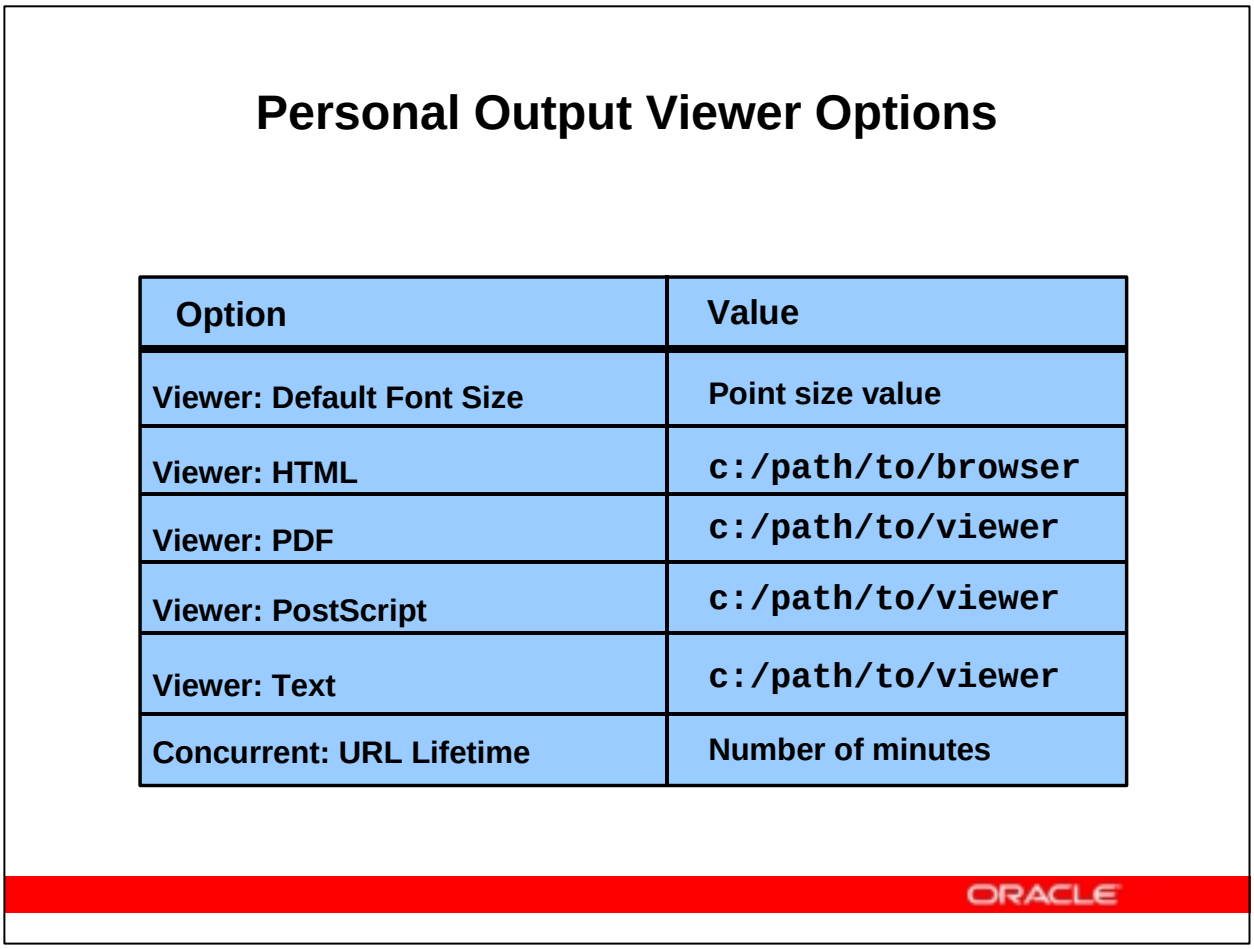
Specify, in bytes, the maximum allowable size of files transferred by the Report Review Agent. If you enter no value, there is no size limit.

RRA: URL

Specify a URL that points to the CGI script on your WebServer to use the Report Review Agent to access files on concurrent processing nodes.

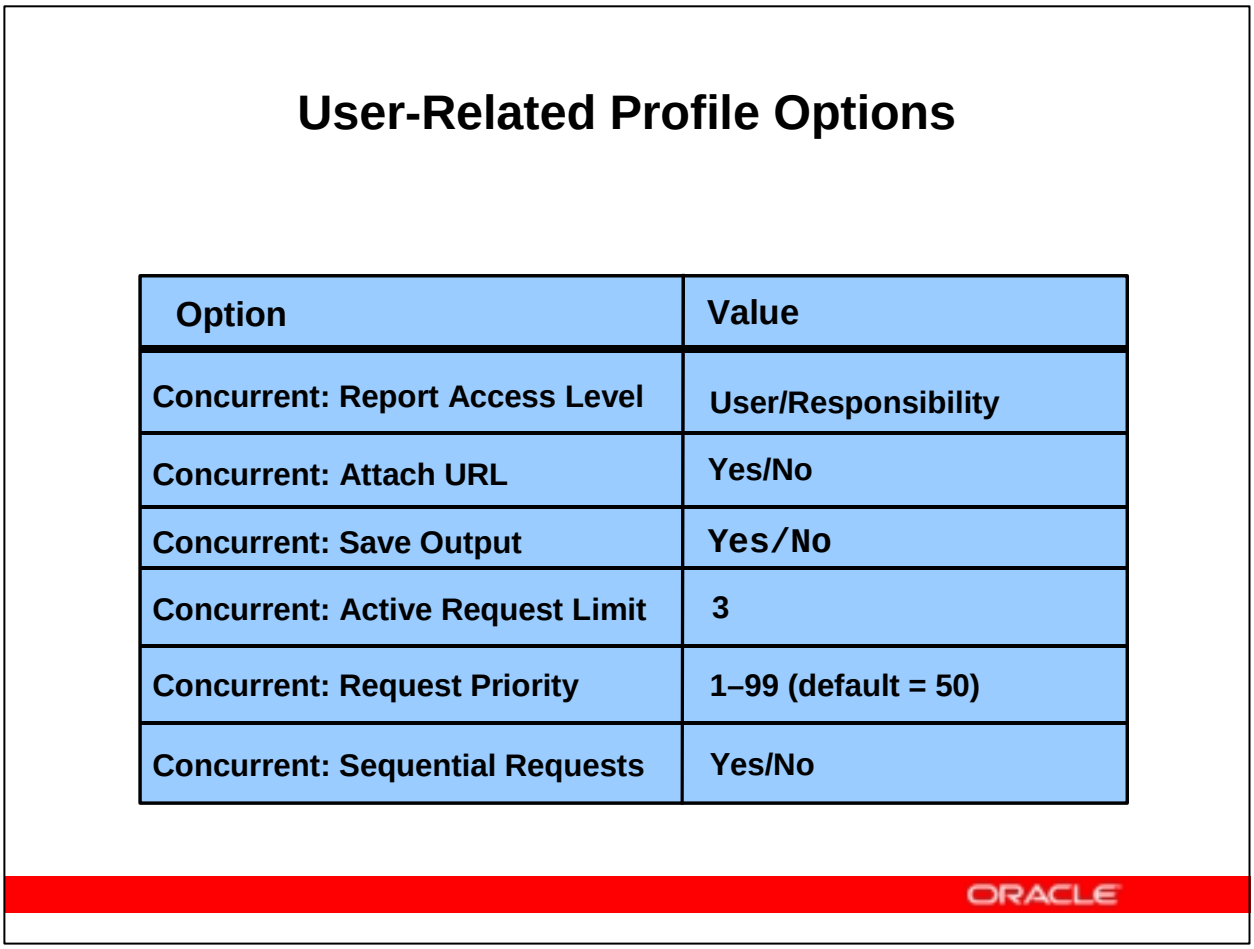
Concurrent: Attach URL

When a user submits a request and specifies people to be notified in the Completion Options region of the Requests window, everyone is sent a notification when the request is finished. When this option is set to Yes, the recipients also receive a URL that they can access to see the request results.



How the User Can Control Request Output

A user can set several options to control request output. You can specify different tools to view different output types, as well as different font sizes. If a user asks to send request completion notifications to other people, a URL pointing to request results can be sent.



How to Use User-Related Profile Options

Concurrent: Report Access Level

This option determines the level of output and log files that are available to a user for online viewing.

Concurrent: Attach URL

Setting this option to "Yes" causes a URL to be attached to request completion notifications. When a user submits a request, and specifies people to be notified in the defining completion options region, a URL is appended to the notification that enables them to view the request on-line.

Concurrent: Save Output

You can save your request outputs in a file if one is generated. This allows you to reprint a request.

Concurrent: Active Request Limit

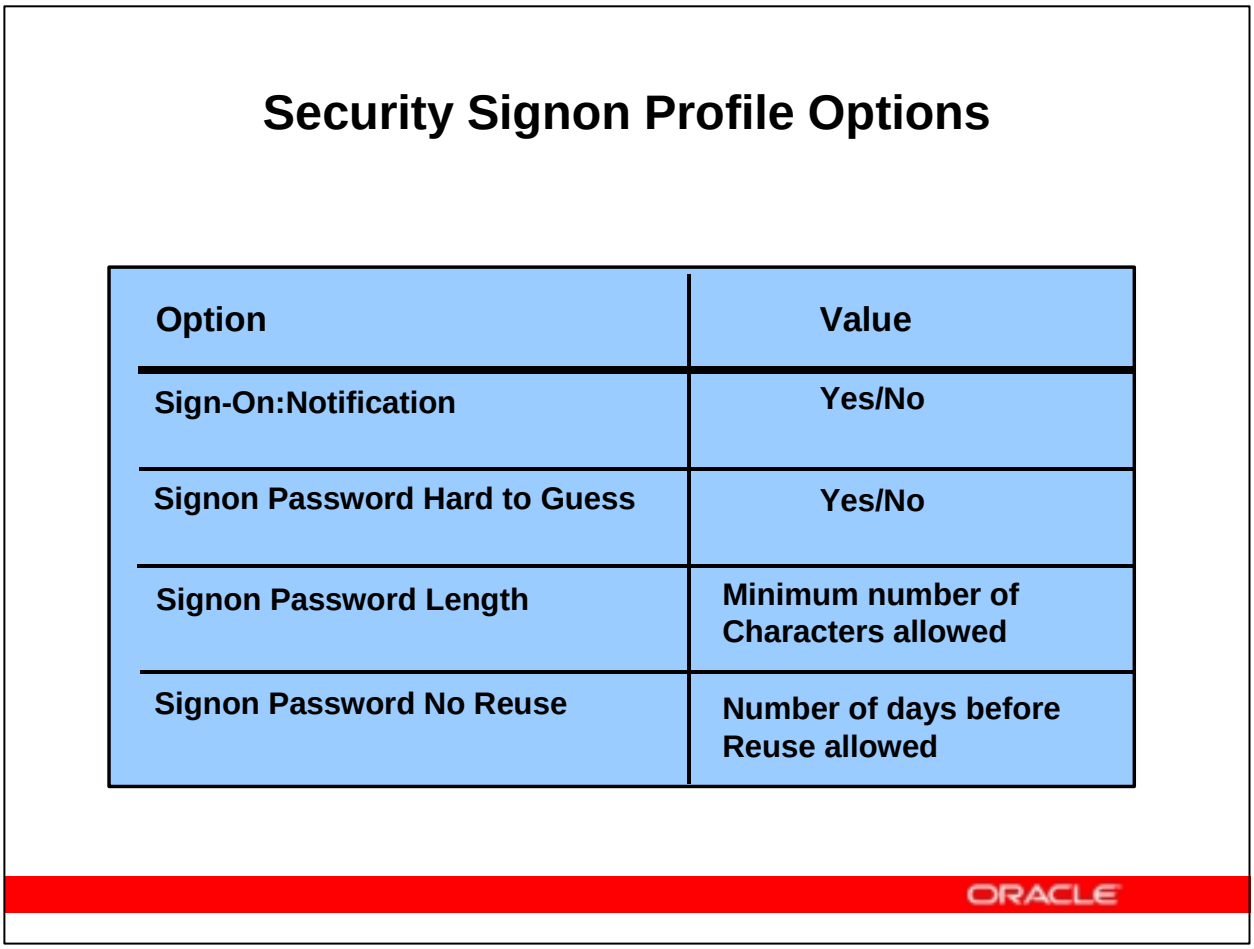
You can limit the number of requests that can be run simultaneously by each user, or for every user at a site.

Concurrent: Request Priority

This displays the default priority number for your concurrent requests. Priorities range from 1 (highest) to 99 (lowest). The default is 50.

Concurrent: Sequential Requests

Setting this option to Yes forces requests to run sequentially in the order in which they were submitted.



How to Use Security Signon Profile Options

Sign-On:Notification

Setting this option to "Yes" displays a message at login that indicates:

- Concurrent request failures since your last session
- How many times someone tried to log on to Oracle Applications with your username and an incorrect password
- When the default printer identified in your user profile is unregistered or not specified

Signon Password Hard to Guess

This profile option sets rules for choosing passwords to ensure that they will be "hard to guess." A password is considered hard-to-guess if it follows these rules:

- It contains at least one letter and at least one number
- It does not contain the username
- It does not contain repeating characters

Signon Password Length

Signon Password Length sets the minimum length of an Applications signon password. If no value is entered, the minimum length defaults to 5.

Signon Password No Reuse

This profile option specifies the number of days that a user must wait before being allowed to reuse a password.

Practice - Profile Options

Overview

The implementation committee wants you to testing setting several personal profile options, and to examine several system profile options.

Tasks

Set your Personal Profile Options

1. Query the “Concurrent: Request Priority” profile value. Note its value, and try to update it.
2. Query the “Java Color Scheme”. Choose a value from the LOV and Save your new value.
3. Log off to see the color scheme changes.
4. Query the “Viewer: Default Font Size”. If there is no value, set it to 10 and Save it.
5. Query the “Printer”. Use the LOV to change or add a value and Save it.
6. Query the “Concurrent: Hold Requests”. Change the user value to “Yes” and Save it.
7. Submit the “Active Users” report.
8. Click the “Options…” button, and verify that the printer you selected for the Printer profile option has defaulted in the Printer field.
9. Submit your request.
10. View your concurrent request.
11. Take your request off hold.
12. Cancel your request.

Examine your System Profile Options

1. Use the Find System Profile Values window to find “Currency:Negative Format” at the Site level.
2. What is “Currency:Negative Format” for the Site? _____
3. Use the Find System Profile Values window to find the “Flexfields:Shorthand Entry” profile option at the Site and User level.
4. What is the default value at the Site level? _____

5. At which levels can the System Administrator change the values? _____

6. Use the Find System Profile Values window to find the “GL Set of Books Name” profile option for the Application Oracle General Ledger and the Responsibility General Ledger Super User.

7. Can this profile be set at the Application level? _____

8. What is the current value and where is it set? _____ _____

Solution - Profile Options

Set your Personal Profile Options

Responsibility = System Administrator

1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Profile > Personal.

3. Query up the profile option “Concurrent:Request Priority”.

   - Note its value here: _____

   - Try to update the value.

   - *(This option cannot be updated at the user level.)*

3. Query up the value for “Java Color Scheme”. Choose a color scheme from User Value list of values. Click Save.

4. Sign off and back on for your color scheme to take effect.

5. Query up the profile option “Viewer: Default Font Size”. If there is no value specified, set it to 10 and click Save.

6. Query up the profile option “Printer”. Use the List of Values to change or add a value. Click Save.

7. Query up the option “Concurrent:Hold Requests”. Change the User Value to “Yes”. Click Save.

8. Navigate to the Submit Request screen (N) Requests > Run.

9. Select the “Active Users” program.

10. Click the “Options…” button and verify that the printer you selected for the “Printer” profile option has defaulted in the Printer field.

11. Submit your request.

12. Navigate to the View Requests window (N) Requests > View.

13. Your request should show a status of “On Hold”.

14. Take your request off hold by clicking the “Remove Hold” button.

15. Cancel your request by clicking the “Cancel Request” button.

Examine your System Profile Options

1. Navigate to (N) Profile > System.

2. Use the Find System Profile Values window to find “Currency:Negative Format” at the Site level.

3. What is the current setting of “Currency:Negative Format” for the Site? _____

4. Use the Find System Profile Values window to find the “Flexfields:Shorthand Entry” profile option at the Site and User level.

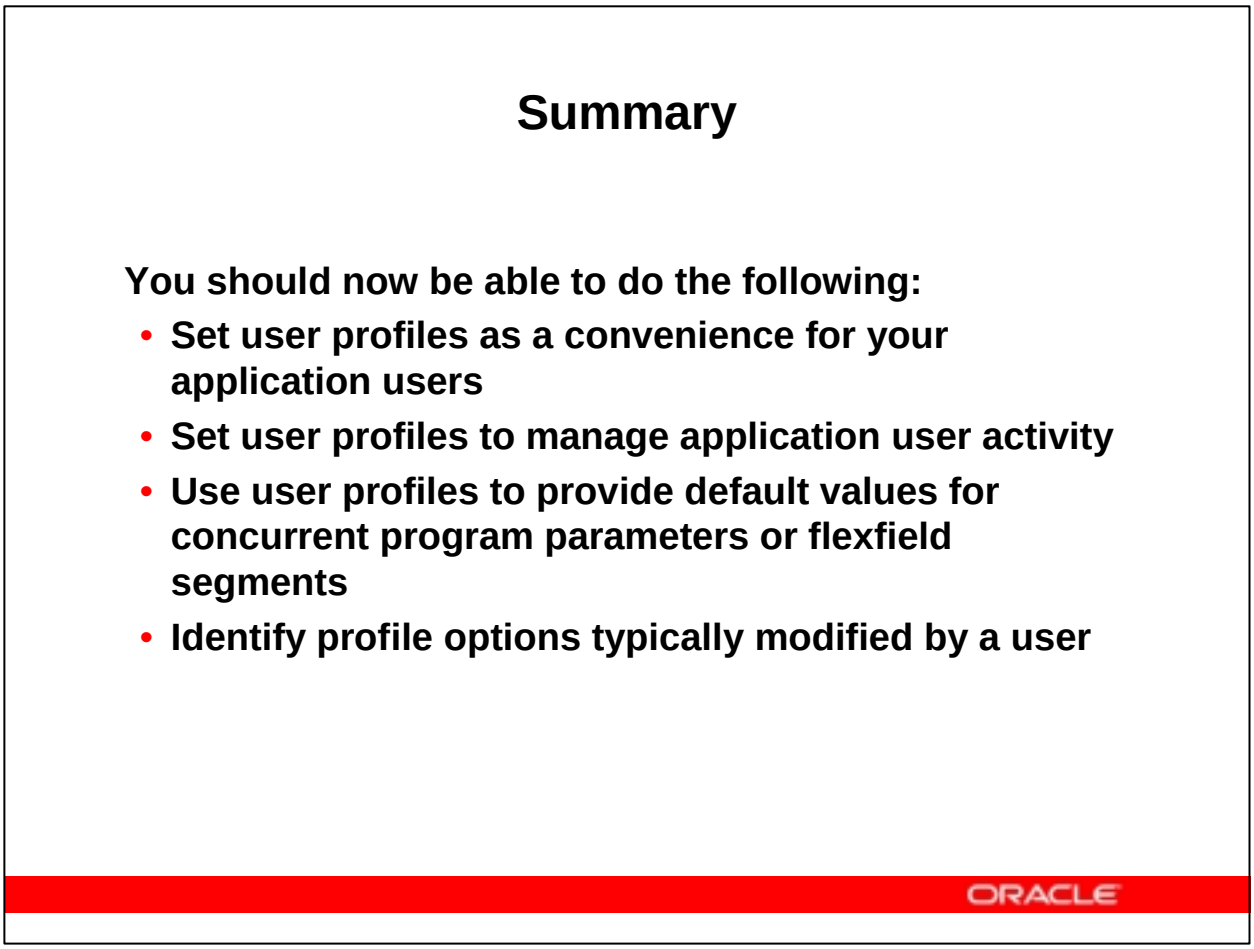
5. What is the default value at the Site level. _____

6. At which levels can the System Administrator change the values? _____

7. Use the Find System Profile Values window to find the “GL Set of Books Name” profile option for the Application Oracle General Ledger and the Responsibility General Ledger Super User.

8. Can this profile be set at the Application level? _____

9. What is the current value and where is it set? _____ _____

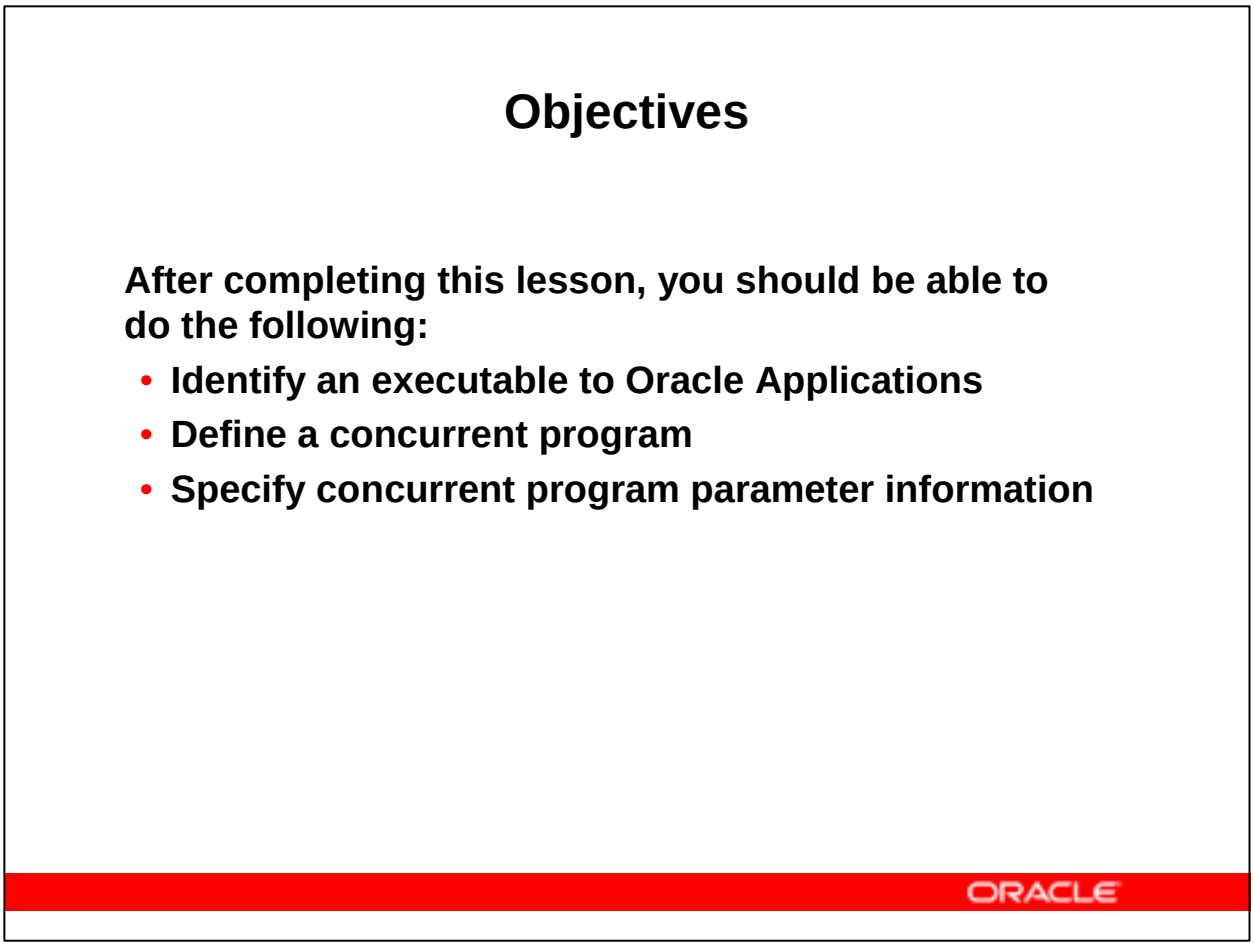


Lesson Summary

An important function of the system administrator is to control the default behavior and appearance of Oracle Applications. This is done by specifying profile values. In this lesson you saw how to display and update profile values. You also learned how to reference previously defined profile values elsewhere in Oracle Applications.

Incorporating a Custom Program

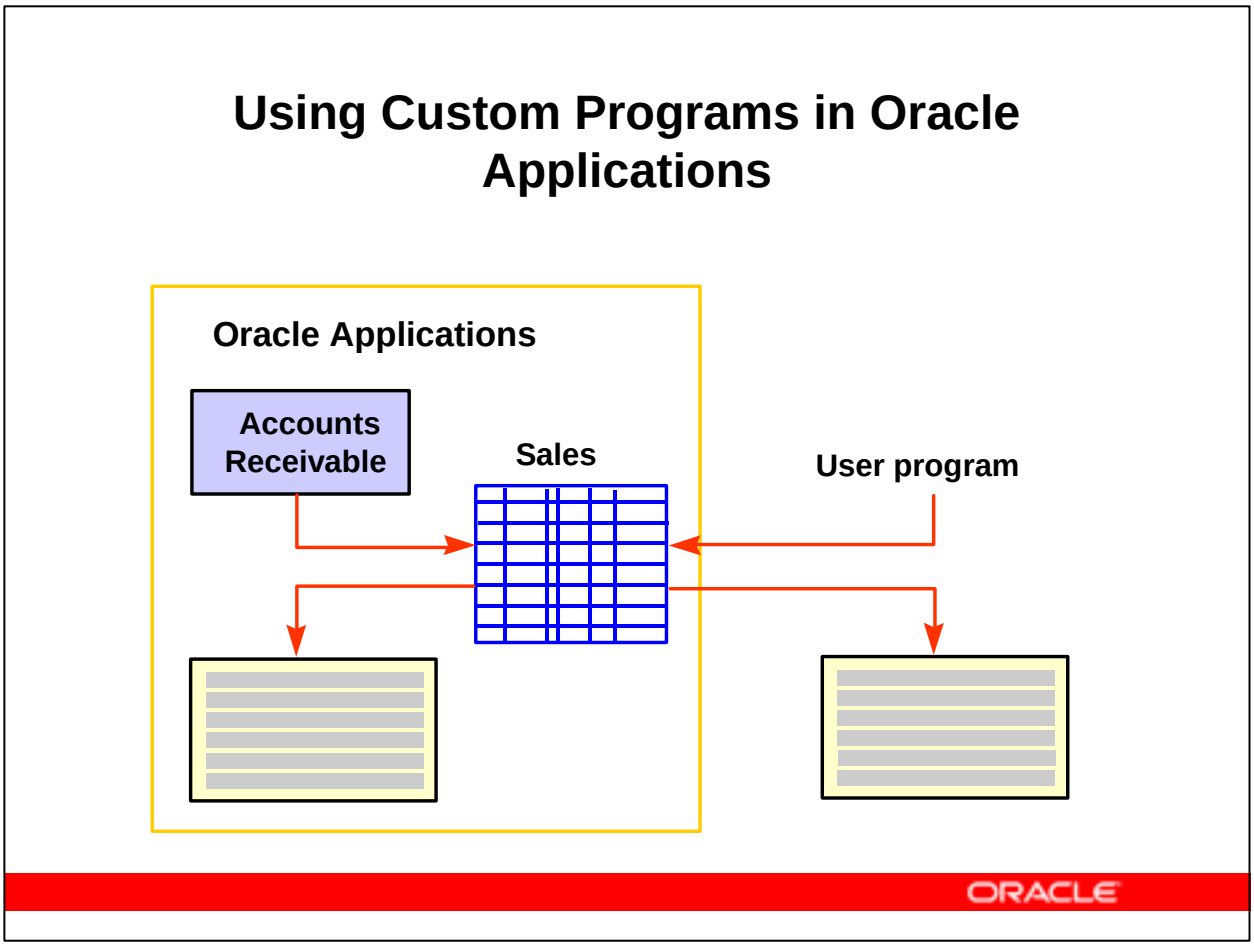
Chapter 6



Lesson Aim

Oracle Applications uses concurrent programs to process and report on data stored in Oracle Applications tables. These programs are secured, processed, and managed within the Oracle Applications environment, using the operations that you have seen in other lessons. However, you can take your custom applications and programs and integrate them into the Oracle Applications environment as well. In that environment they can be secured and managed the same way as regular Oracle Applications programs.

This lesson covers how to integrate user-developed programs into Oracle Applications.



Integrating Custom Programs with Oracle Applications

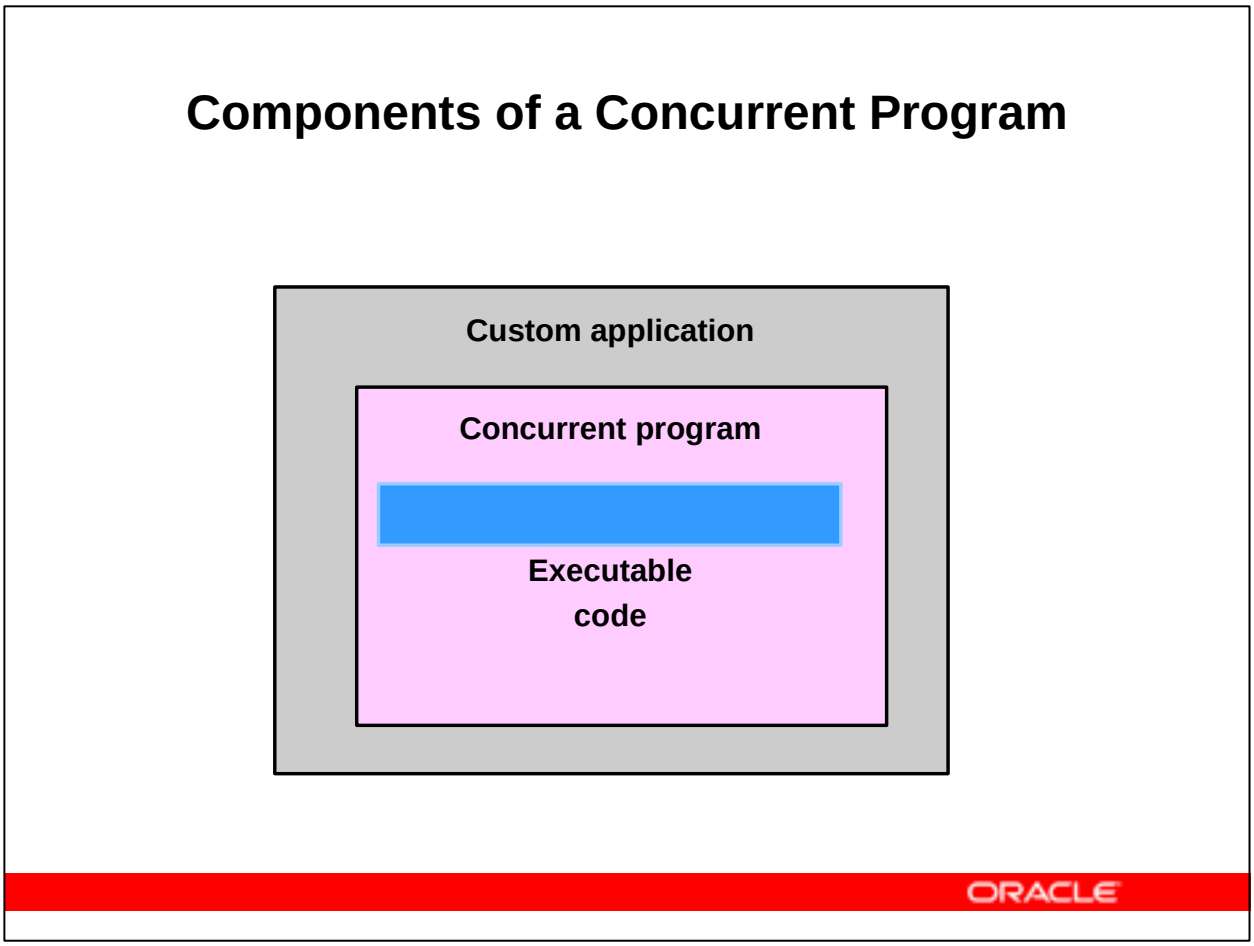
Although Oracle Applications provides you with a complete range of functionality, you may need to develop and implement your own custom programs to perform specialized processing. You can integrate such custom applications into the Oracle Applications environment so that they are processed like all your other Oracle Applications programs and reports.

For example, a sales table used by Oracle Receivables can also be accessed by a user program for special reporting needs. Such a user reporting program can be added to the Oracle Applications environment.

Assumptions for Incorporating a New Program

- The program has been completed
- The program parameters have been documented
- Any value sets needed already exist
- Tokens for Oracle Reports are defined

ORACLE



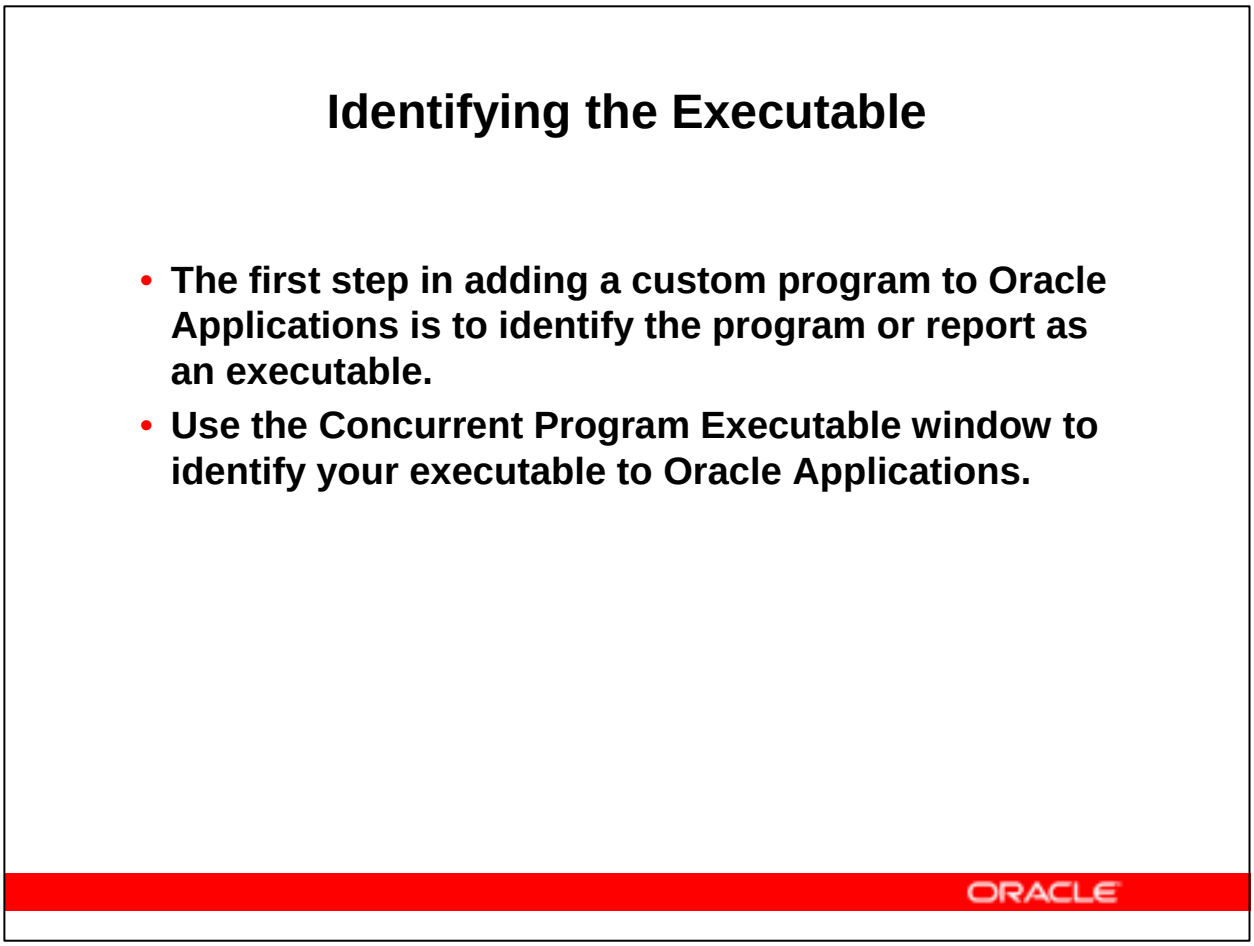
Components of a Concurrent Program

You need to define several components of a concurrent program when adding your custom program to the Oracle Applications environment. A concurrent program consists of an executable module—for example, an Oracle Reports program or a PL/ SQL procedure—and the input parameters required by the program or procedure. A concurrent program must be associated with either an existing Oracle application or a custom application. In most cases, you will associate any custom concurrent programs with a registered custom application. This protects your concurrent programs during an upgrade.

Adding a Custom Program to Oracle Applications

1. Develop the program or report.
2. Identify the program as an executable and register it with an application.
3. Create a concurrent program containing the executable and its parameters.
4. Add the concurrent program to a request group or set for processing.

ORACLE



Identifying the Executable

(N) Concurrent > Program > Executable

The Concurrent Program Executable window prompts for the following information:

- Executable – enter a name for your concurrent program executable. In the Concurrent Programs window you assign this name to a concurrent program to associate your concurrent program with your executable logic.
- Short Name – enter a short name for your concurrent program executable.
- Application – the concurrent managers use the application to determine in which directory structure to look for your execution file.
- Execution Method – specifies the type of program, such as a PL/SQL procedure or an Oracle Reports program. The execution method cannot be changed once the concurrent program executable has been assigned to one or more concurrent programs in the Concurrent Programs window.
- Execution File Name – enter the operating system name of your execution file.
- Subroutine Name – enter the name of your C or Pro*C program subroutine. Only immediate programs or spawned programs using the Unified C API use this field.

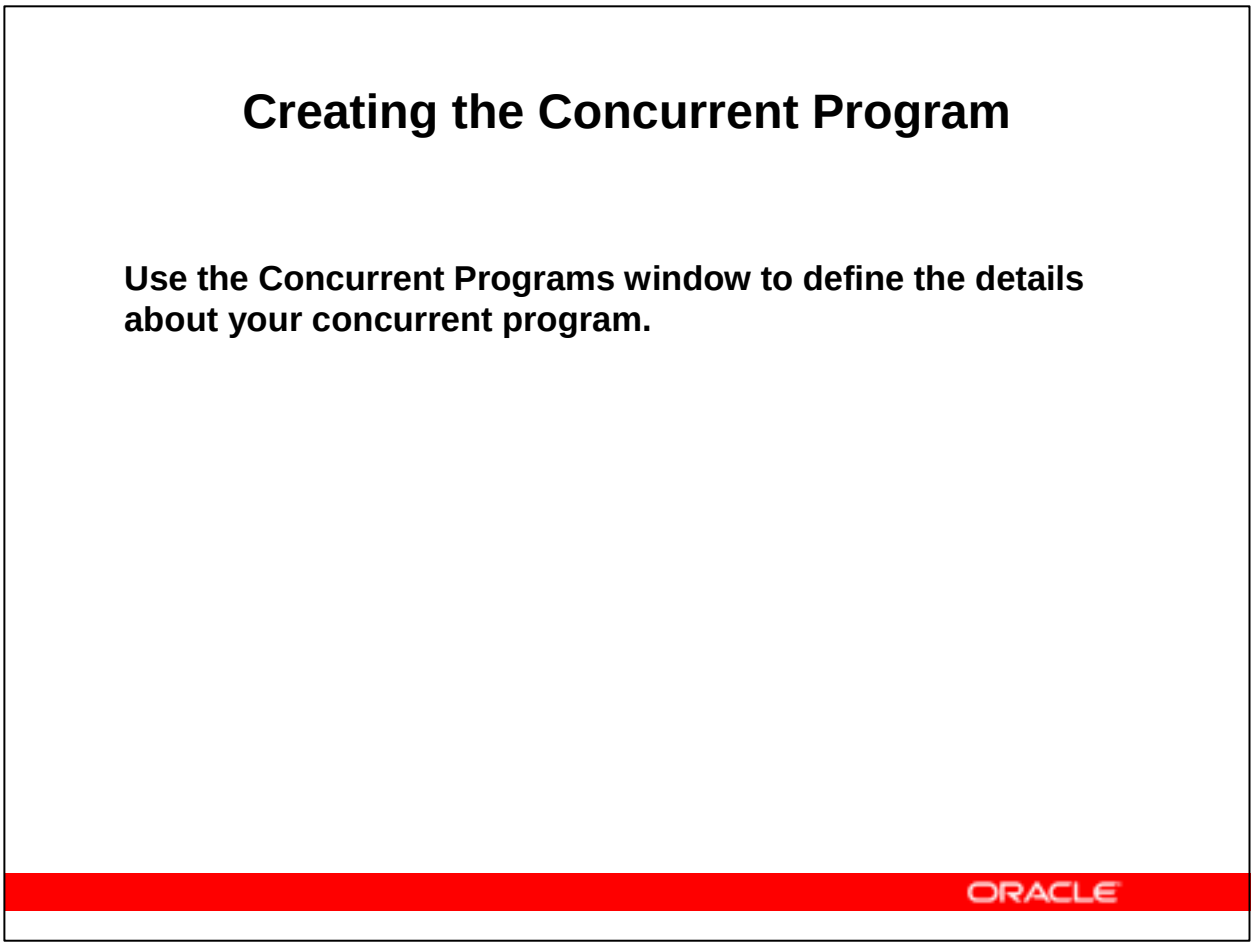
Note: Defining new immediate concurrent programs is not recommended. Use either a PL/SQL stored procedure or a spawned C program instead.

- Stage Function Parameters button – opens a window that allows you to enter parameters for the Request set Stage Function. This button is only enabled when you select Request Set Stage Function as you Execution Method.

For more information see:

(Help) Applied Technology > Oracle Applications System Administration >

Concurrent Program Executable Window



Concurrent Programs Window Fields

(N) Concurrent > Program > Define

Program – enter a descriptive name for your concurrent program. This is the name you see when you view your requests in the Requests window. If this concurrent program runs through Standard Request Submission, you see this name in the Submit Requests window.

Short Name – enter a short name that Oracle Applications can use to associate your program with a concurrent program executable.

Application – the program's application determines what ORACLE username your program runs in and where to place the log and output files.

Enabled – indicate whether users will be able to submit requests to run this program and the concurrent managers will be able to run your program. Disabled programs do not display in users' lists, and do not appear in any concurrent manager queues.

Executable: Name – select the concurrent program executable that can run your program. The executable is defined using the Concurrent Program Executable window.

Executable: Method – the execution method used by your concurrent program displays in this field.

Options – contains options for SQL*Report programs.

Priority – assign a priority for this program. If you do not assign a priority, the user's profile option Concurrent:Priority sets the request's priority at submission time.

Request: Type – associate your program with a request type if desired. Certain concurrent managers are specialized to run only certain request types.

Incrementor – for use by Oracle Applications internal developers only.

MLS Function – the MLS function used by the program (if applicable). The Multilingual Concurrent Request feature allows a user to submit a request once to be run multiple tomes, each time in a different language. If this program utilizes this feature the MLS function determines which installed languages are needed for the request.

Use in SRS – check this box to indicate that this program can be run using Standard Request Submission. If this box is checked, you must register the parameters of this program.

Allow Disabled Values – for a program authorized for SRS submission, check this box to allow a user to enter disabled or outdated parameter values.

Run Alone – check this box if your program is incompatible with other programs in its logical database and should therefore be run alone.

Enable Trace – turns on SQL tracing when program runs.

NLS Compliant – check this box if the program allows a request to reflect a language and territory that are different from the language and territory that the user is operating in. This option should be set only by the developer of the program. The program must be NLS Compliant to utilize this feature. See: *Oracle Applications Developer's Guide.*

Output Fields – use these fields to specify the handling or output from executions of this program.

Concurrent Programs Window Buttons

Copy to… - choose this button to create another concurrent program using the same executable, request, and report information.

Session Control – choose this button to specify options for the database session of the concurrent program when it is executed.

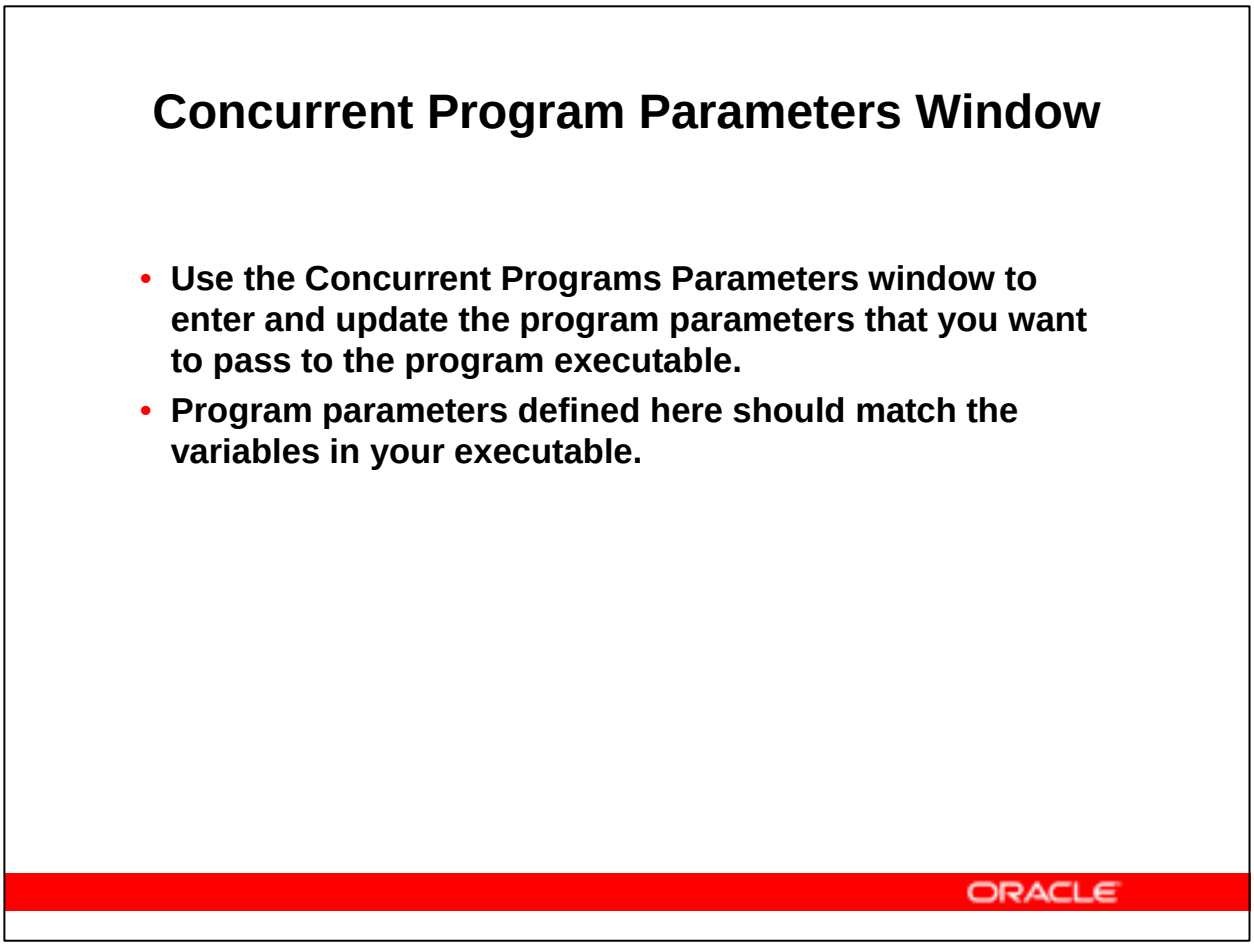
Incompatibilities – choose this button to open the Incompatible Programs window.

Parameters – chose this button to open the Program Parameters window.

For more information see

(Help) Applied Technology > Oracle Applications System Administration >

Concurrent Programs Window



Concurrent Programs Parameters Window Fields

(N) Concurrent > Program > Define > (B) Parameters

Sequence – specify the sequence number for the parameter that you are defining.

Parameter – specify the parameter name.

Enabled – disabled parameters are not displayed at request submission time and are not passed to your program.

Validation Information: Value Set – specify an independent, table, or nonvalidated value set for use in checking values passed for this parameter.

Default Type – if you intend to specify a default for this parameter, declare the default type.

Default Value – for certain types of default, you can specify a default value to use. This default value appears automatically when you enter the parameter fields in the Validation region.

Required – choose this box if a value for this parameter is required by your program.

Enable Security – if the value set for this parameter does not allow security rules, this field is display only. Otherwise, you can choose to apply any security rules defined for this value set to affect your parameter list also.

Range – choose either Low or High if you want to validate your parameter value against the value of another parameter in this structure.

Display – indicate whether to display this parameter in the Parameters window when a user submits a request to run this program form the Submit Requests window.

Display Size – enter the field length in characters for this parameter. The user sees and and fills in the field in the Parameters window launched from the Submit Request window.

Description Size – enter the display length in characters for the parameter value description.

Prompt – enter a prompt to appear in the Parameter window of the Submit Request window.

Concatenated Description Size – enter the length for the parameter value description field. This field displays all the parameter values as a concatenated string.

Token – this refers to the name of a keyword or parameter for an Oracle Reports program. Entries in this field are case-sensitive. Any values entered in this field must exactly match the value expected by the Oracle Reports program.

For a complete description of the fields in this window see

(Help) Applied Technology > Oracle Applications System Administration >

Concurrent Programs Window > Concurrent Program Parameters Window

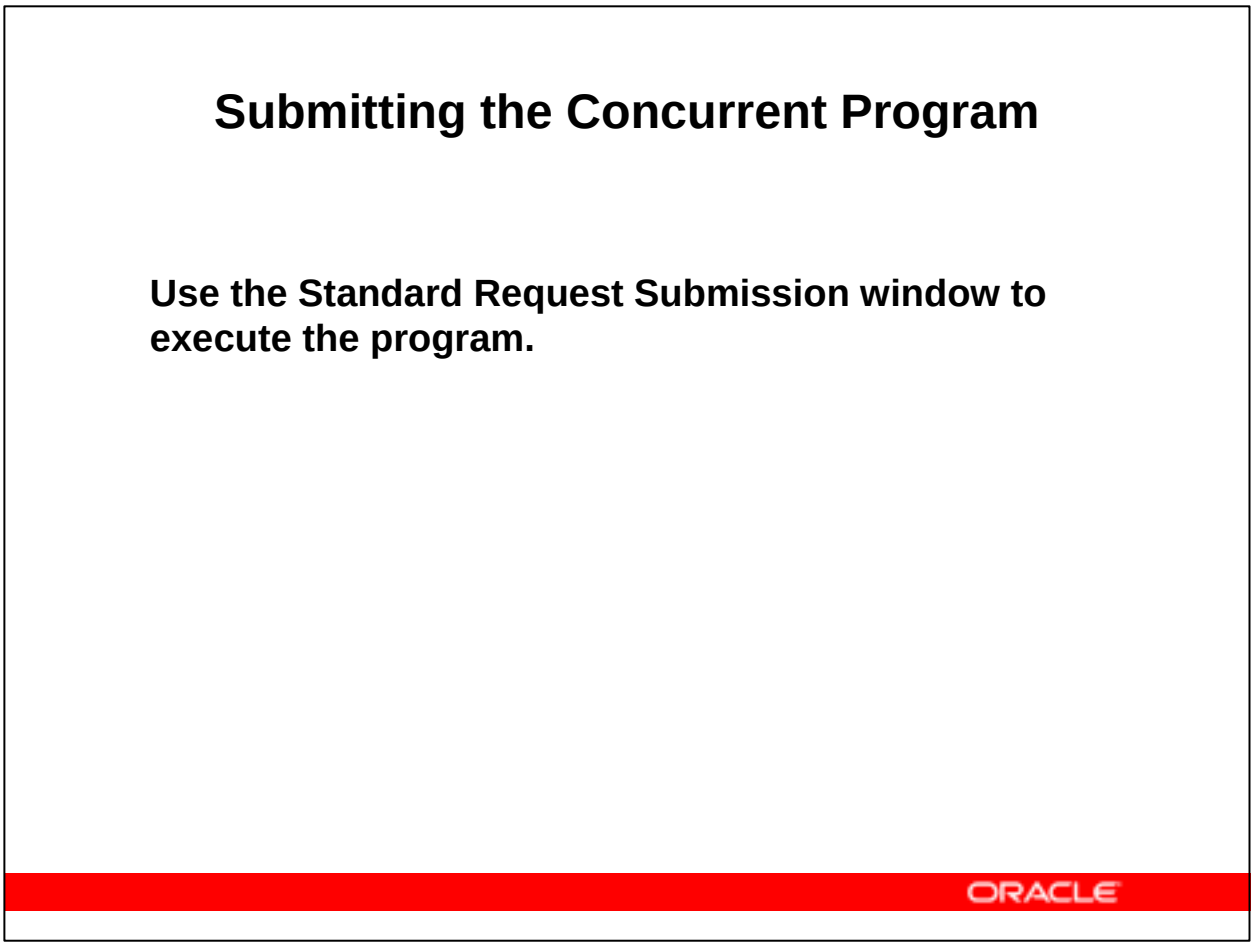
Associate a Program with a Request Group

(N) Security > Responsibility > Request

Request Group Window Fields

For the definition of the fields in this window see

(Help) Applied Technology > Oracle Applications System Administration > Request Group Window



(N) Requests > Run

Practice - Incorporating Custom Programs

Overview

The committee wants you to test the process for adding a custom program. In order to speed the testing, the committee has agreed to use an existing program, the Completed Concurrent Requests report, to test this functionality. In order to test this process, you will do the following.

- Create a concurrent program executable
- Create a concurrent program definition
- Add the concurrent program to a request group
- Add the request group to a responsibility
- Test your custom program

Tasks

Create your Concurrent Program Executable

1. Create a new Concurrent Program Executable. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Executable**: *Your Initials*CPCRQ
   - **Short Name**: *Your Initials*CPCRQ
   - **Application**: Application Object Library
   - **Description**: *Your Initials* Custom Completed Concurrent Requests
   - **Execution Method**: Oracle Reports
   - **Execution File Name**: FNDCPCRQ

Create your Concurrent Program Definition

2. After you have created your executable, you need to create a concurrent program to contain it. Query up the Completed Concurrent Requests program in the Concurrent Programs window and use the Copy to button to create your custom program.
   - **Program**: *Your Initials* Completed Concurrent Requests
   - **Short Name**: *Your Initials*CPCRQ

- **Application**: Application Object Library
- Choose to Include Program Parameters when copying
- Choose to Include Program Incompatibilities when copying

Add your Concurrent Program to a Request Group

3. Add your concurrent program to your existing *Your Initials* Assistant SysAdmin request group. Once you have added it to this group, it will have already been assigned to the *Your Initials* Assistant System Administrator responsibility and to the *Your Initials*AssistSA user.

Test your Concurrent Program

4. Sign-on as the *Your Initials*AssistSA user, and submit a concurrent request for your new report.

Solution - Incorporating Custom Programs

Create your Concurrent Program Executable

Responsibility = System Administrator

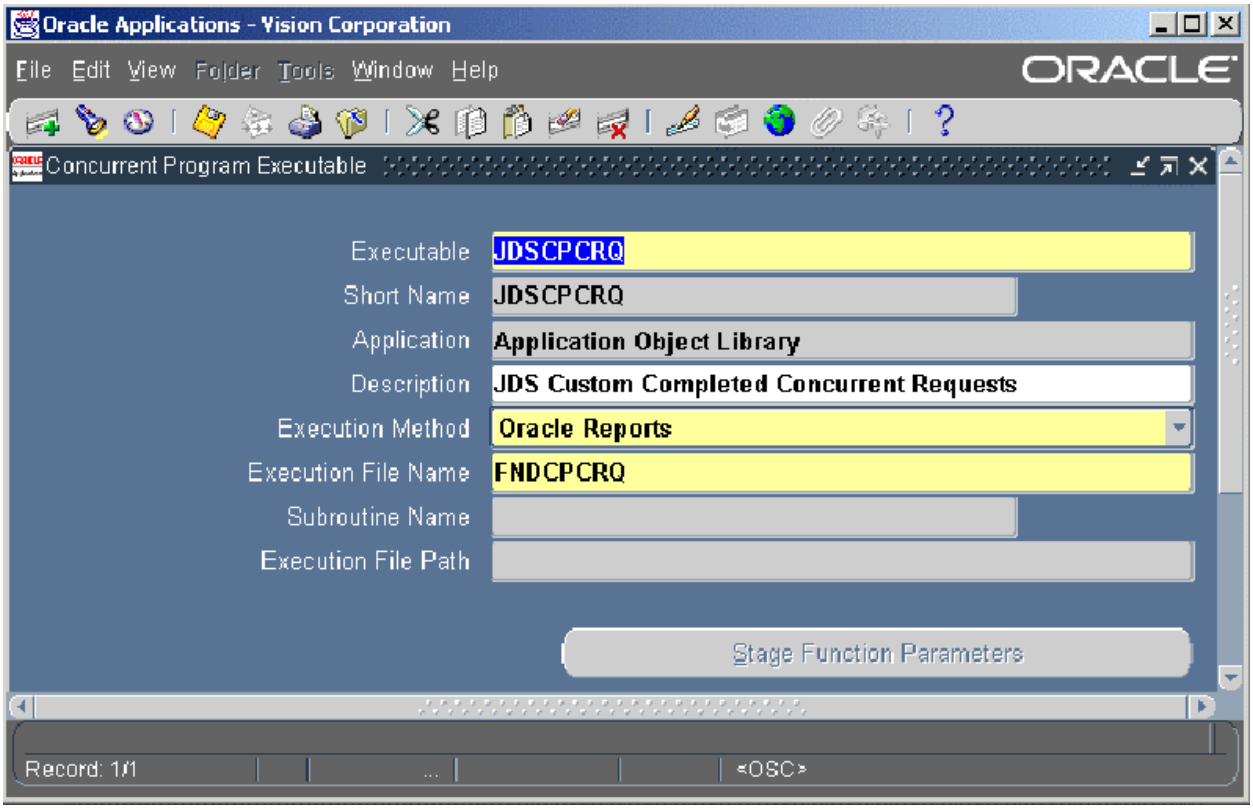
1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Concurrent > Program > Executable.

3. Create your concurrent program executable. Your fields should look as follows.

   - **Executable**: *Your Initials*CPCRQ
   - **Short Name**: *Your Initials*CPCRQ
   - **Application**: Application Object Library
   - **Description**: *Your Initials* Custom Completed Concurrent Requests
   - **Execution Method**: Oracle Reports
   - **Execution File Name**: FNDCPCRQ

4. Save your work. Your completed form should appear similar to the example shown in the slide.

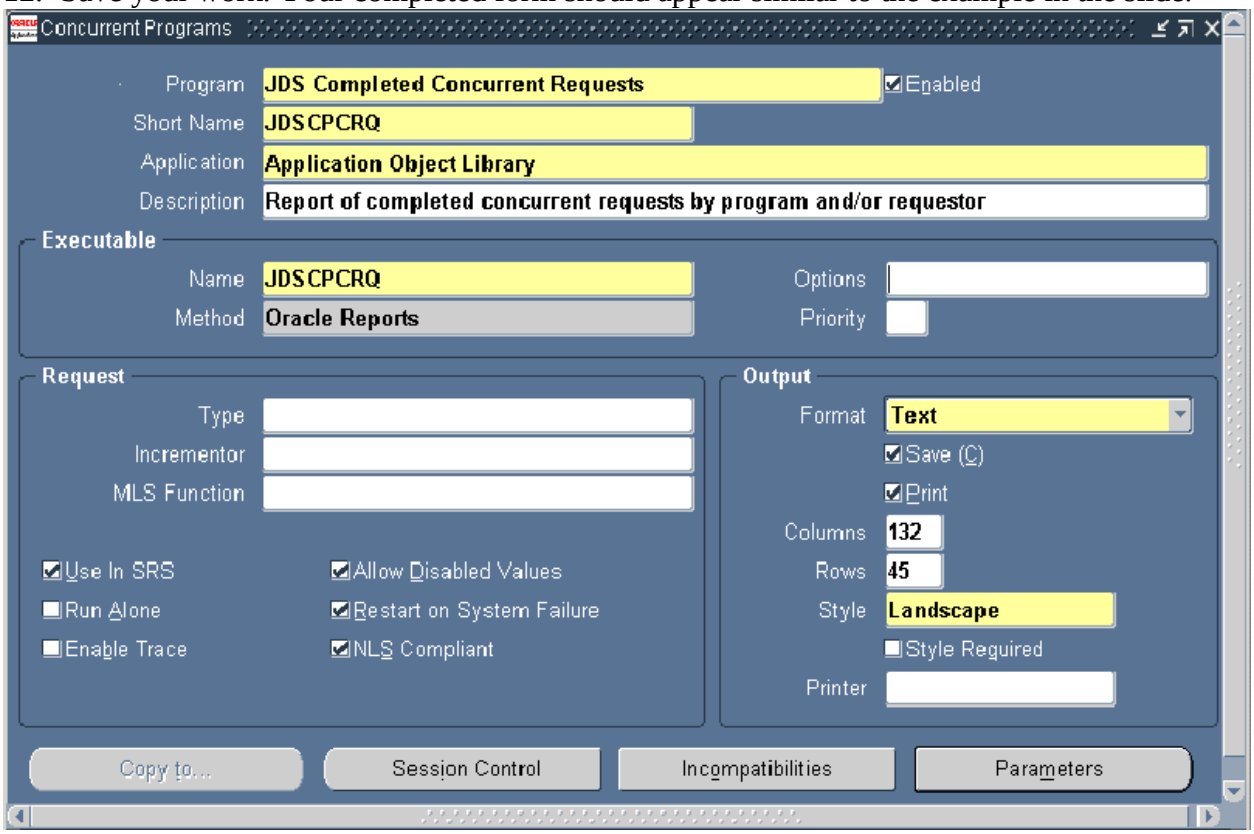


5. Close the form.

Create your Concurrent Program Definition

6. Navigate to (N) Concurrent > Program > Define.
7. Query up the Completed Concurrent Requests program.
8. Click the “Copy to…” button.
9. In the Copy to window enter the following:
    - **Program**: *Your Initials* Completed Concurrent Requests
    - **Short Name**: *Your Initials*CPCRQ (e.g., WHSCPCRQ)
    - **Application**: Application Object Library
    - Choose to **Include** Program Parameters when copying
    - Choose to **Include** Program Incompatibilities when copying
10. Click the “OK” button to close the “Copy to …” window.
11. Navigate to the Executable region of the form and replace the executable listed with the executable you created (*Your Initials*CPCRQ).

12. Save your work. Your completed form should appear similar to the example in the slide.



13. Close the form.

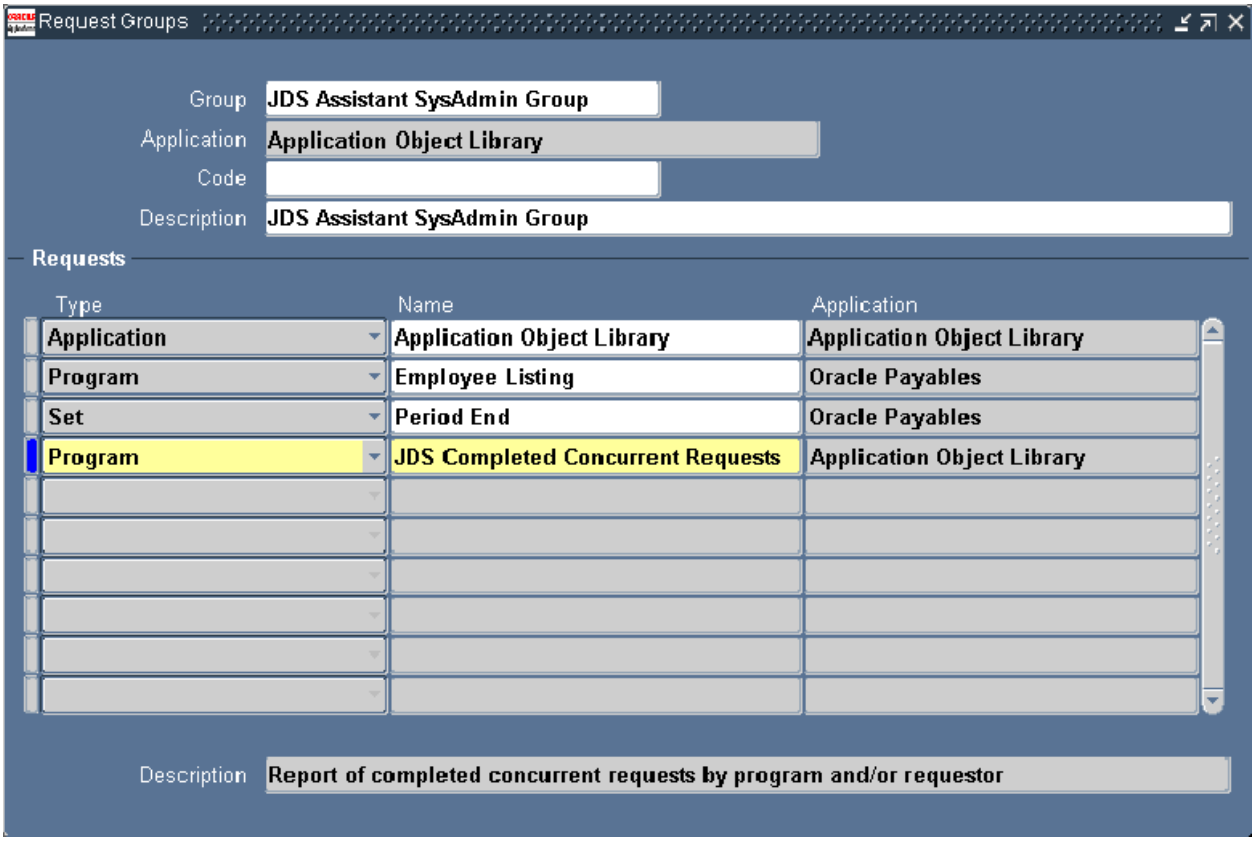
Add your Concurrent Program to a Request Group

14. Navigate to (N) Security > Responsibility > Request.

15. Query up your request group *Your Initials* Assistant SysAdmin Group.

16. Insert a new record and the Program *Your Initials* Completed Concurrent Requests.

17. Save your work. Your completed form should appear similar to the example in the slide.



18. Close the form.

Test your Concurrent Program

19. Switch Responsibilities to or log in as the *Your Initials*AssistSA.

20. Navigate to (N) Requests > Run.

21. Click the “OK” button to submit a single request.

22. Submit your new program and watch it run successfully.

Summary

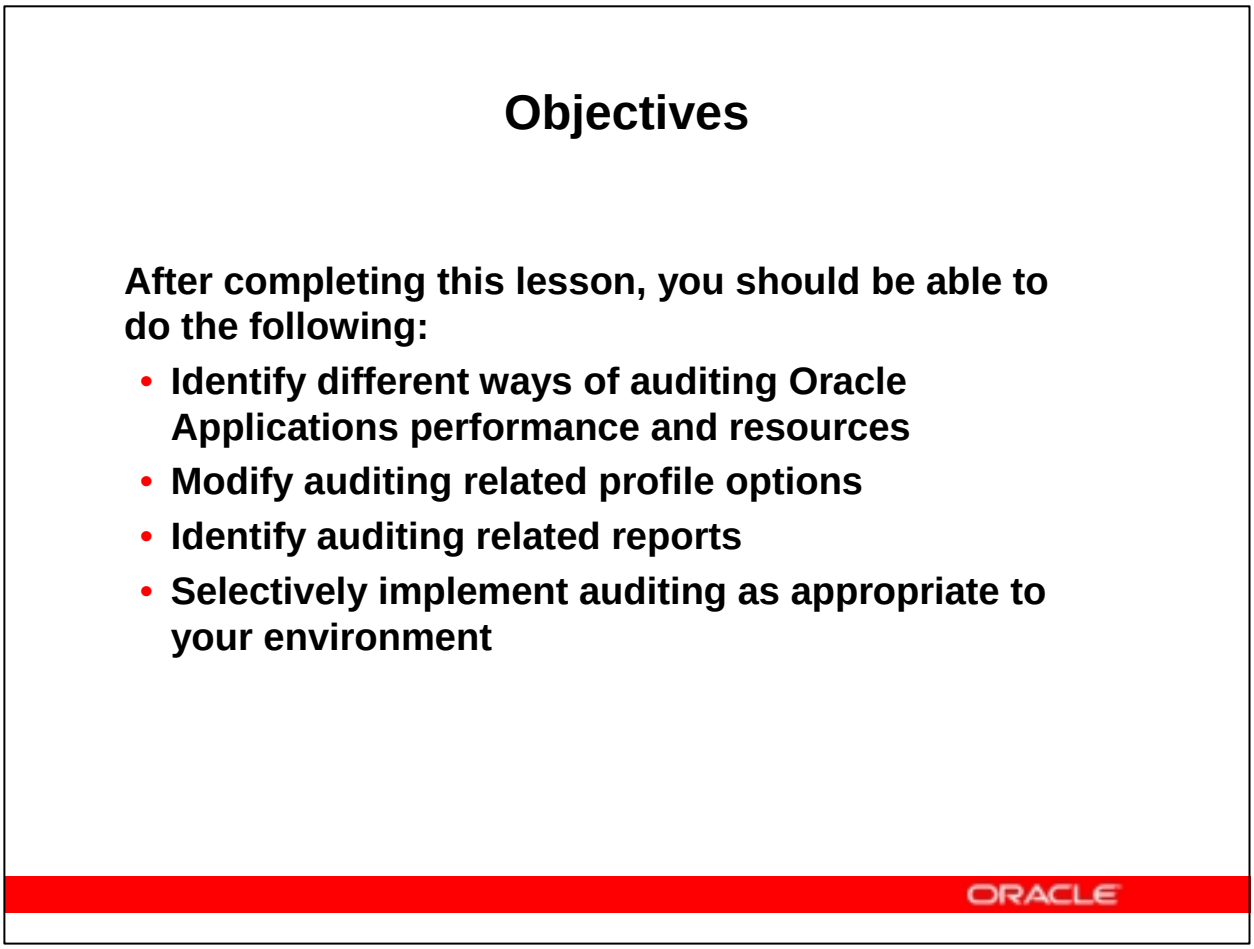
You should now be able to:

- **Identify an executable to Oracle Applications**
- **Define a concurrent program**
- **Specify concurrent program parameter information**

ORACLE

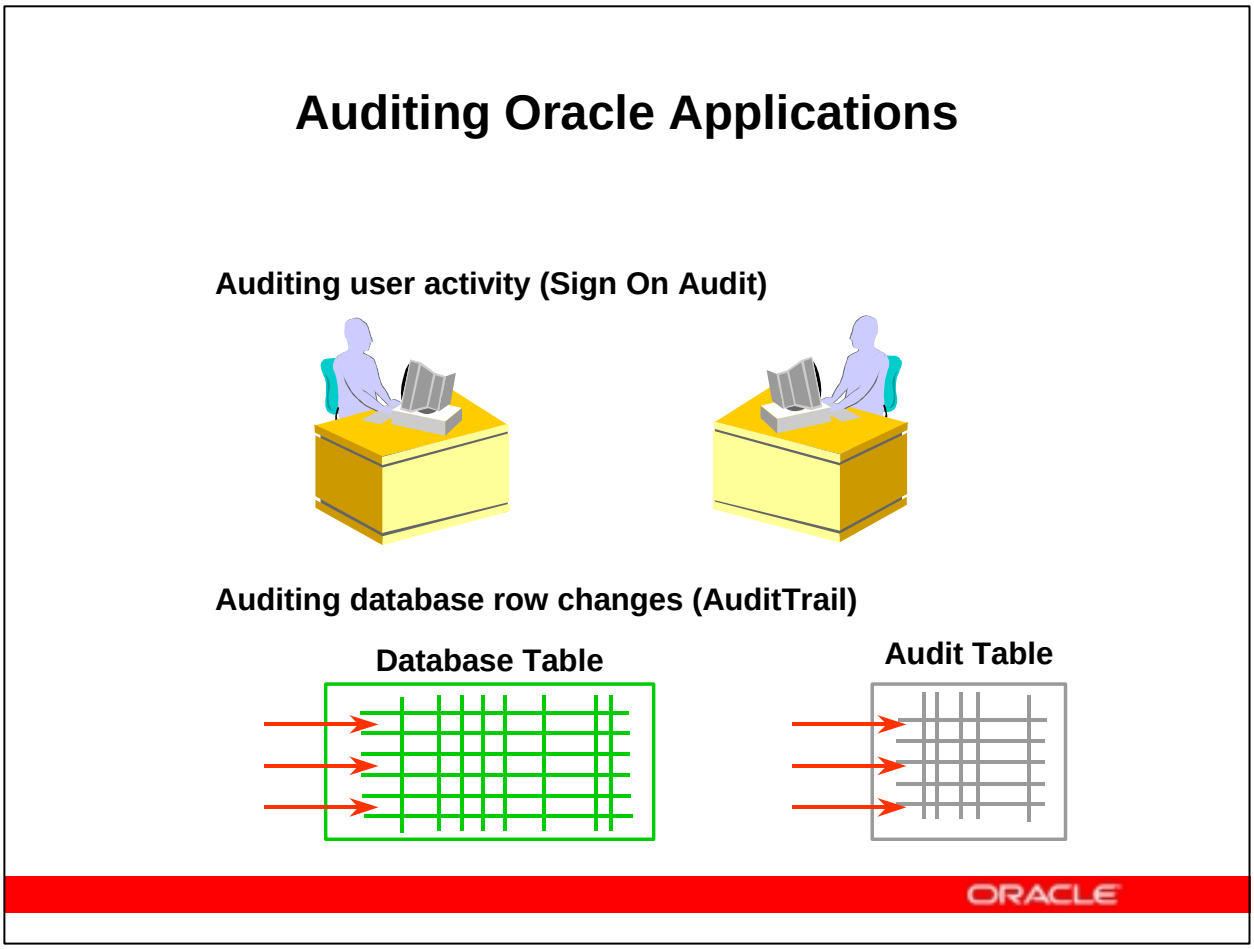
Auditing System Resources

Chapter 7



Lesson Aim

As system administrator you may audit application users and the changes that they make to application data. You will learn how both of these options are set and why you use them.



Changes to Audit in Oracle Applications

Within Oracle Applications, you can audit user activity and database row changes.

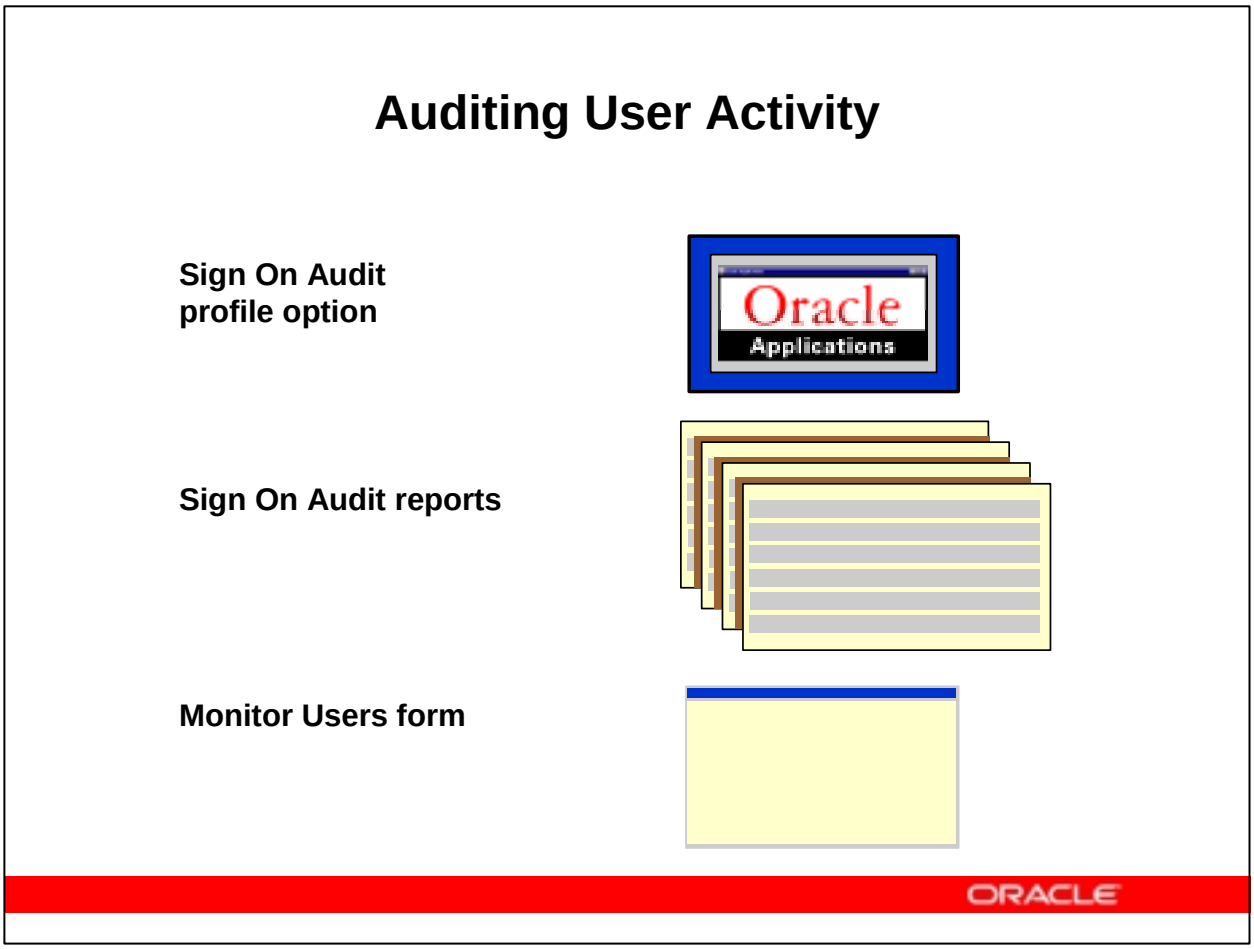
Sign-On Audit Feature

The Sign-On Audit feature enables you to monitor user activity. You can audit at the user, responsibility, or form level. When you enable Sign-On Audit, you specify the desired level of auditing. Sign-On Audit provides two methods for viewing audit data:

- The Monitor Users form provides online access to audit data.
- Various Sign-On Audit reports are available.

AuditTrail Feature

With the AuditTrail feature you can keep a history of changes to your important data: what changed, who changed it, and when it changed. With AuditTrail, you can easily determine how any data row or element obtained its current value. The audit data is stored in database tables that can be queried by using any standard Oracle SQL tool.



Sign-On Audit Feature

You can audit and monitor user activity by enabling the Oracle Applications Sign-On Audit feature. This enables you to track the activity of users signed on to Oracle Applications. Implement the Sign-On Audit feature by updating the Sign-On Audit Level system profile.

With Sign-On Audit you can choose whom to audit and what type of user information to track. You can selectively determine what audit information you need to match the needs of your organization.

Sign-On Audit Reports

Sign-On Audit reports give you historical, detailed information on what your users do in your application. You control the data selected for the reports as well as how the data is presented.

Monitor Users Window

The Monitor Users window gives you online, real-time information about who is using Oracle Applications and for what purpose.

You can see which users are signed on, which responsibilities, forms (windows), and terminals they are using, as well as other information.

Sign-On Audit Levels

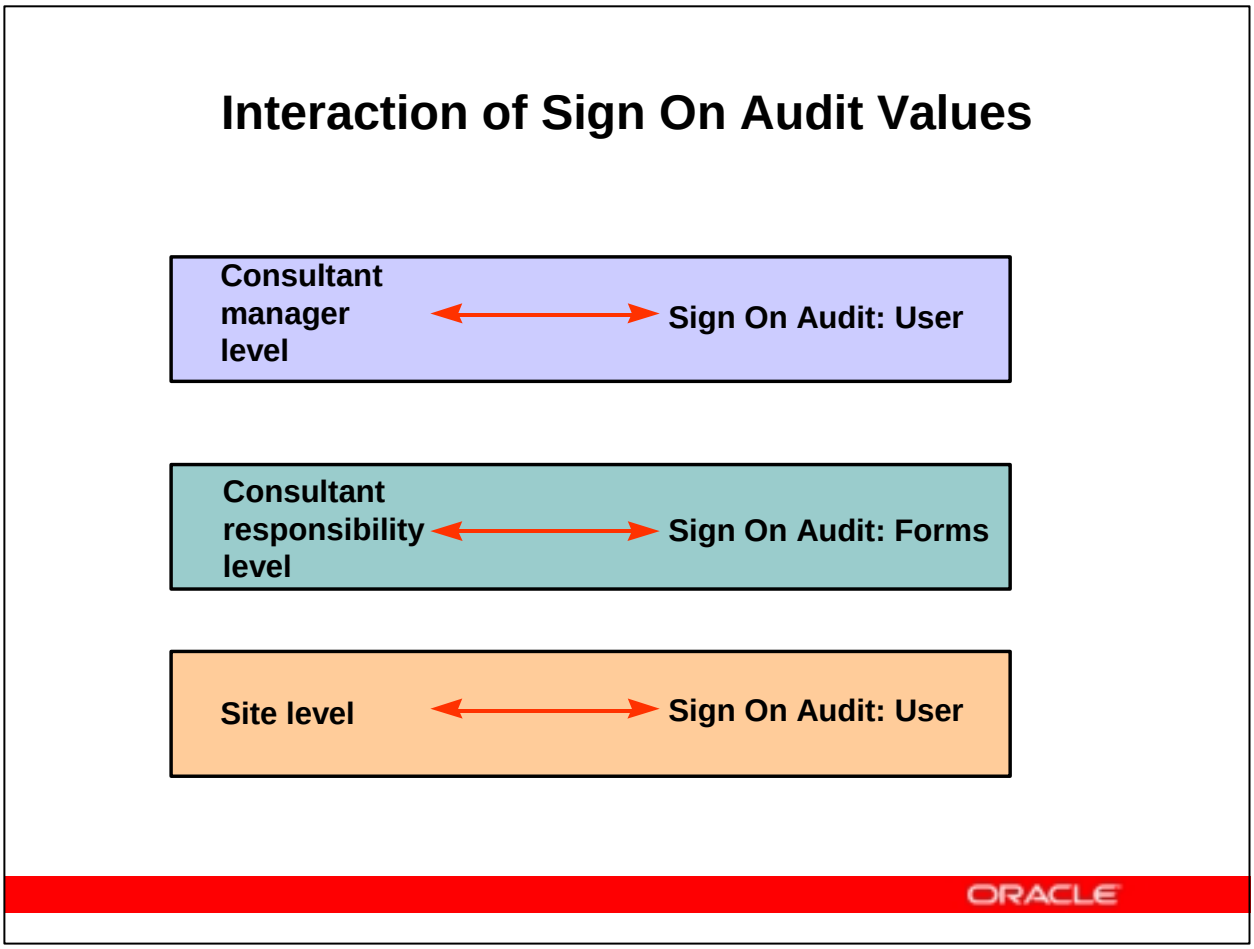
The Sign-On Audit profile option enables you to select a level at which to audit users who sign on to Oracle Applications. The four possible audit levels increase in functionality and detail of auditing.

None This value is the default and disables the Sign-On Audit feature.

User Auditing at the user level tracks who signs on to your system, the times that users log on and off, and the terminals they use.

Responsibility Auditing at the responsibility level tracks all the information tracked at the user level, including the responsibility that the user is using and how much time the user used the responsibility.

Form Auditing at the form level tracks all the information tracked at the user and responsibility levels as well as which forms the user chooses and how long he or she uses those forms.

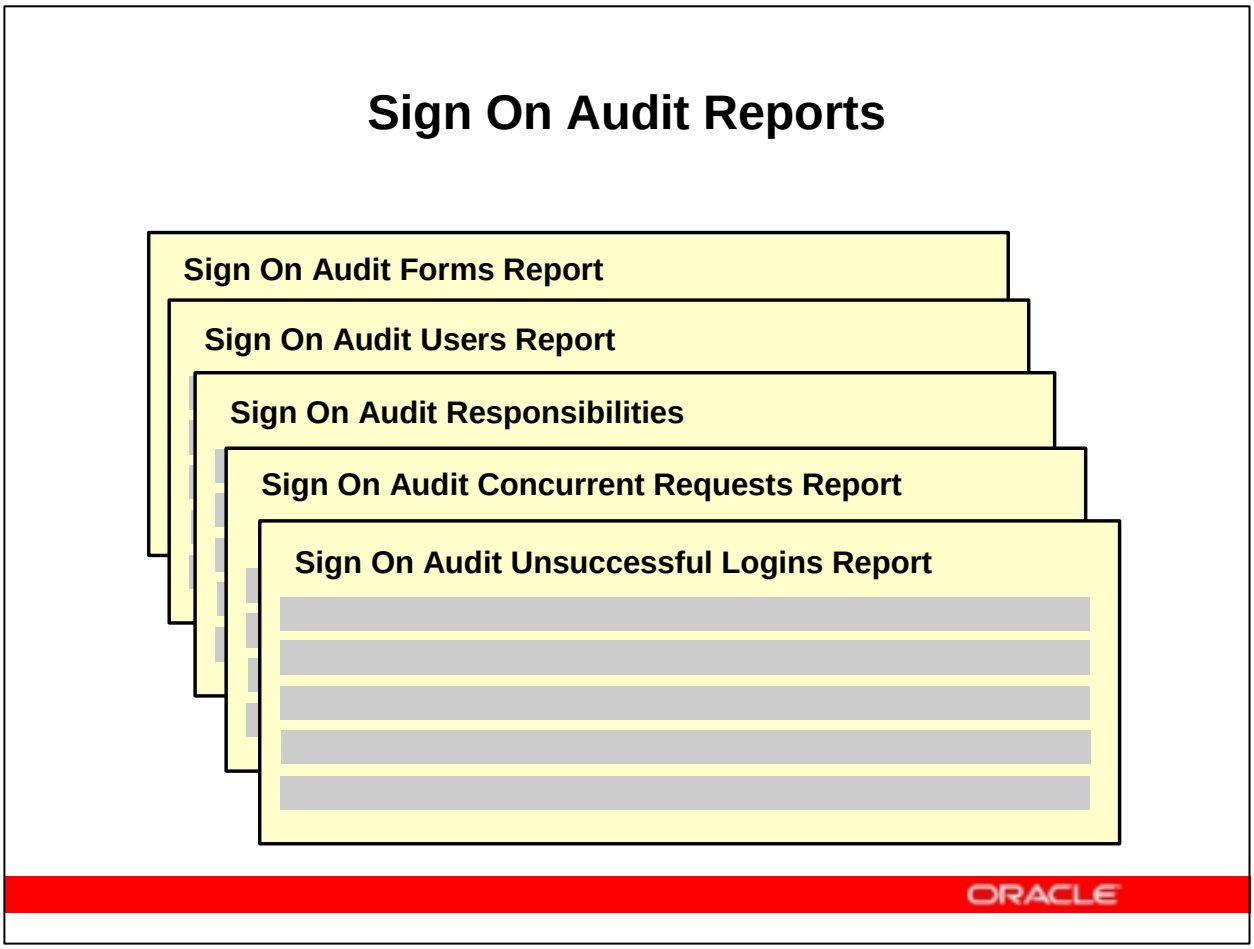


Specifying the Sign-On Audit Level

You can specify a Sign-On Audit value at multiple profile levels. A value in effect at a higher level overrides a value at a lower level. For example, assume a scenario where your business regularly has independent consultants on-site whose activity you need to document for billing purposes. In such a situation, you could use the following auditing setup:

- At the site level specify Sign-on Audit: User to implement the minimum auditing level for all users of Oracle Applications.
- Set up a custom responsibility to be used by your consultants. Set Sign-On Audit for this responsibility to the responsibility or even the forms level.
- At the user profile level for the manager overseeing the consultants, set Sign-on Audit back to the user level.

Sign On Audit Reports



Reports

The Sign-On Audit feature can generate several reports detailing information gathered by Sign-On Audit. These reports are shown in the figure.

- Sign-On Audit Forms
- Sign-On Audit Users
- Sign-On Audit Responsibilities
- Sign-On Audit Concurrent Requests Report
- Sign-On Audit Unsuccessful Logins Report

Using the Sign-On Audit Forms Report

Use the Sign-On Audit Forms Report to view who is navigating to what form and when. You can use this report to check for bottlenecks in the system. The report contains the following columns:

- Username: The Oracle Applications username of the user who accessed the form
- Login Name: The operating system login name of the user who accessed the form
- Terminal Name: The operating system ID of the terminal from which the user accessed the form
- Responsibility Name: The name of the responsibility from which the user accessed (The responsibility is displayed only if you audited the user at the responsibility or form Sign-On Audit level.)
- Start Active Time/End Active Time: The dates and times when the user accessed and exited the form (This information is displayed only if you audited at the form level.)
- Form Name: The name of the form that the user accessed (The form name is displayed only if you audited the user at the form level.)

Using the Sign-On Concurrent Requests Report

You can use this report to view information about concurrent requests.

- Login Name: The operating system login name of the user who submitted the concurrent request
- Request ID: The concurrent request ID of the submitted concurrent request
- Concurrent Program Name: The name of the concurrent program that the user submitted
- User Name: The Oracle Applications username of the user who submitted the concurrent request
- Responsibility Name: The name of the responsibility from which the user submitted the concurrent request (The responsibility is displayed only if you audited the user at the responsibility or form level.)
- Form Name: The name of the form from which the user submitted the concurrent request. (This information is displayed only if you audited the user at the form level.)
- Requested Start Time: The date and time when the request started running

Using the Sign-On Audit Responsibilities Report

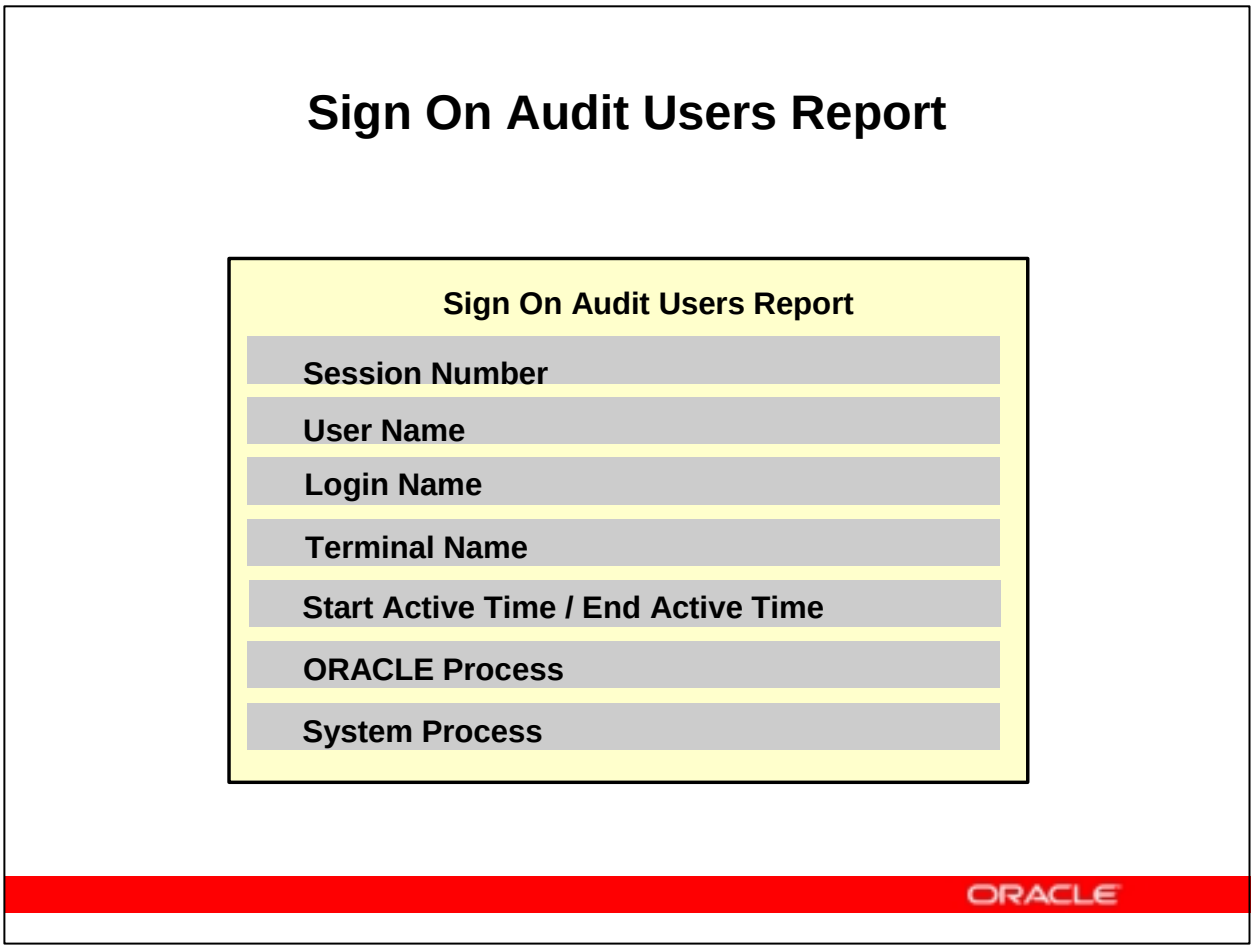
You can use this report to view who is selecting what responsibility and when. The report contains the following columns:

- Username: The Oracle Applications username of the user who selected the form
- Login Name: The operating system login name of the user who selected the responsibility
- Terminal Name: The operating system ID of the terminal from which the user selected the responsibility
- Responsibility Name: The name of the responsibility that the user used (The responsibility is displayed only if you audited the user at the responsibility or form level.)
- Start Active Time/End Active Time: The dates and times when the user selected or exited the responsibility. The start active time and end active time are displayed only if you audited the user at the responsibility or form level.

Using the Sign-On Audit Unsuccessful Logins Report

You can use this report to view who attempted unsuccessfully to sign on to Oracle Applications as another user. An unsuccessful login occurs when a correct username is entered with an incorrect password. The report contains the following columns:

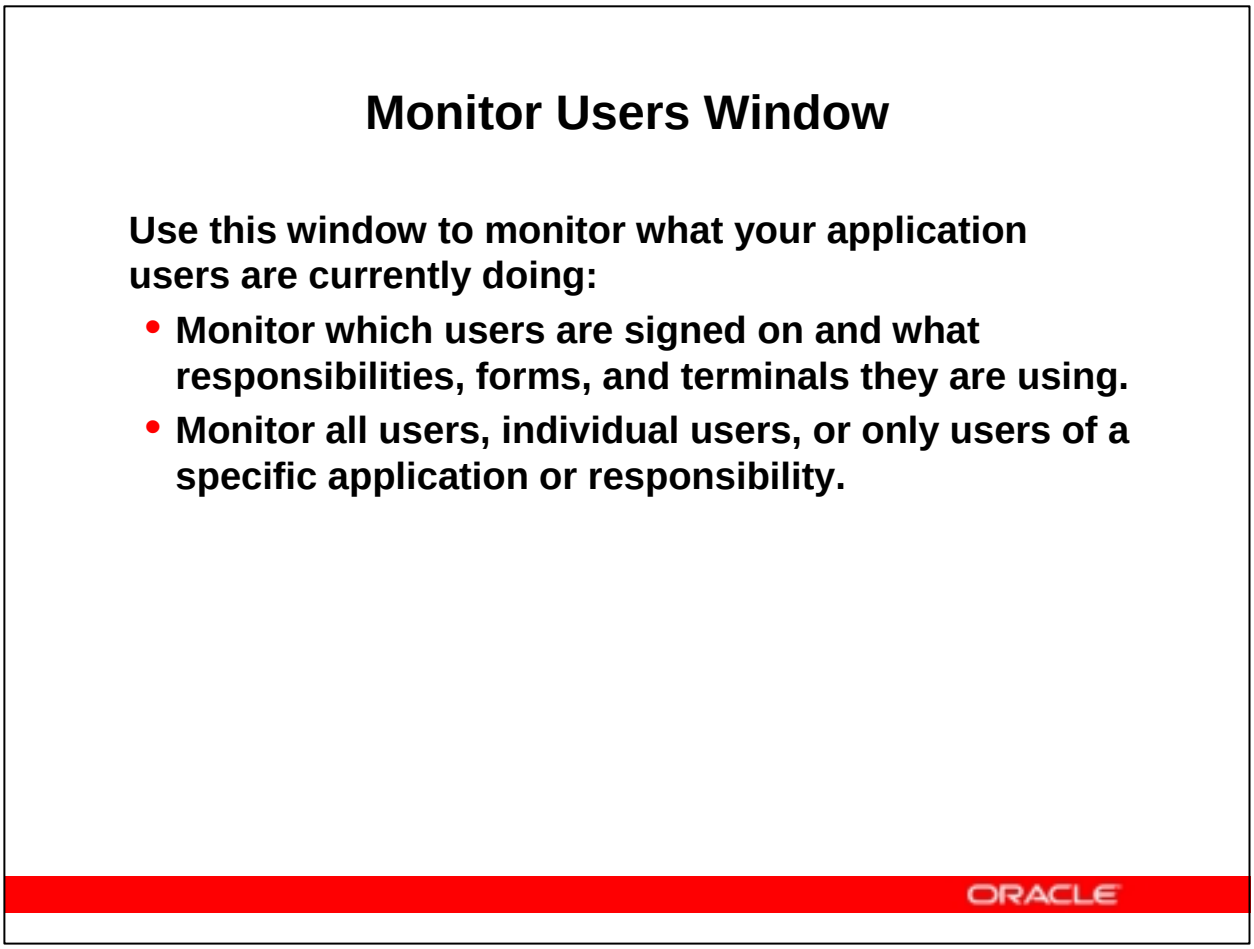
- Username: The Oracle Application username of the user who unsuccessfully signed on
- Login Name: The operating system login name of the user who unsuccessfully tried to sign on
- Terminal: The operating system ID of the terminal from which the user unsuccessfully tried to sign on
- Attempt Time: The date and time when the user unsuccessfully tried to sign on



Using the Sign On Audit Users Report

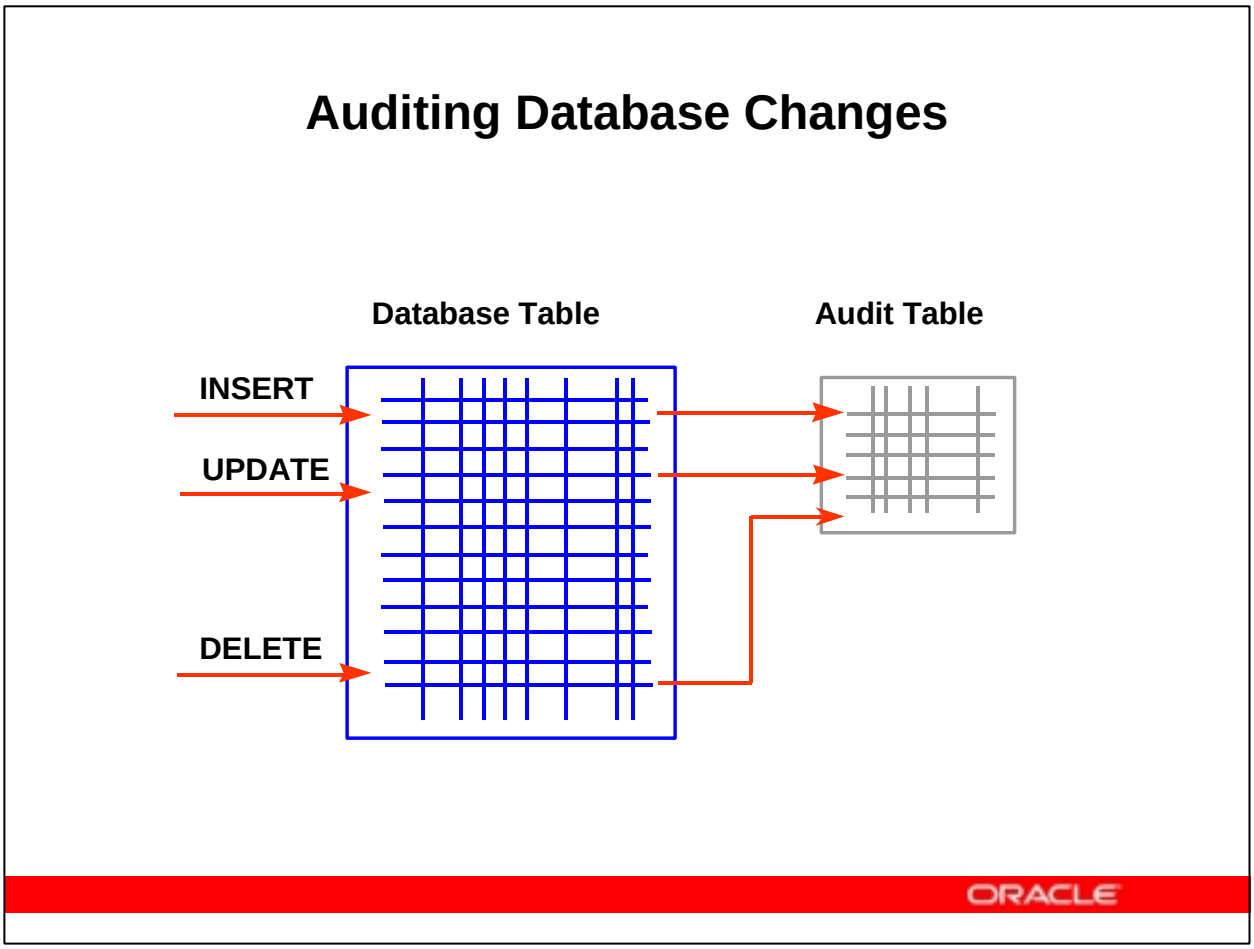
You can use this report to view who signs on and for how long. The report contains the following columns:

- Session Number: The Oracle Applications session number uniquely identifying each application user sign-on
- User Name: The Oracle Applications username of the user who signed on
- Login Name: The operating system login name of the user who signed on
- Terminal Name: The operating system ID of the terminal from which the user signed on
- Start Active Time/End Active Time: The dates and times when the user accessed/ exited Oracle Applications (The start active time and end active time display only if you audited the user at the user Sign-on Audit level.)
- ORACLE Process: The Oracle Process ID used during the user's sign-on
- System Process: The operating system process ID used during the user's sign-on



Monitor Users Window

(N) Security > User > Monitor



Auditing Important Data

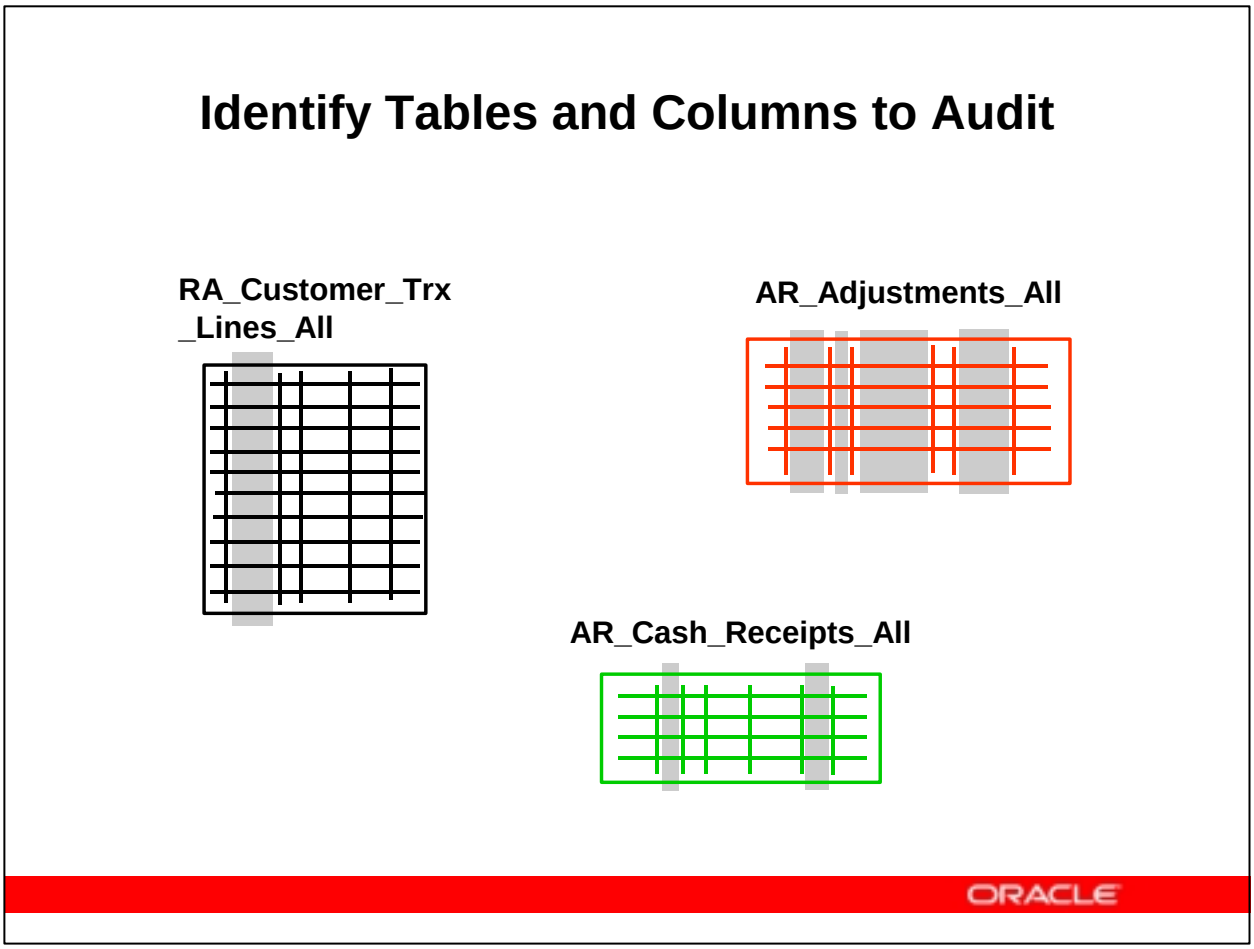
AuditTrail enables you to keep a history of changes to your important data. You can track what changed, who changed it, and when it was changed. Using AuditTrail, you can easily determine how any data row or column obtained its current value.

When you enter or update data through your forms, you change the underlying database tables. AuditTrail tracks which rows in the database were updated. Audit information for each of these changes is stored in audit tables that can then be queried and reported upon.

How to Set Up the AuditTrail

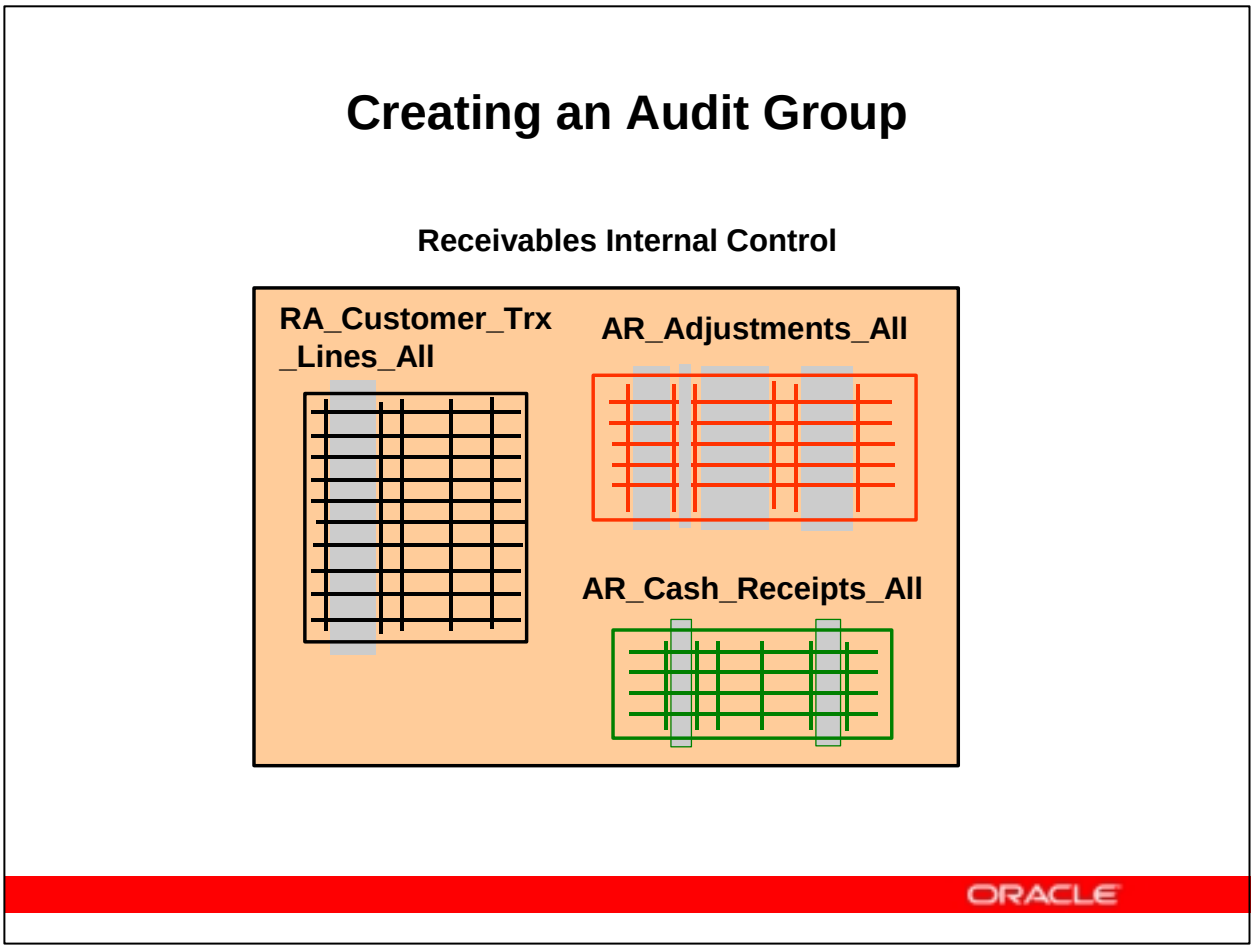
The figure shows the steps involved in implementing AuditTrail. These steps are covered in the subsequent slides.

- Because there is overhead involved in recording audit information, it is important to determine which columns of which tables should be audited; not all changes to all tables need to be tracked.
- After you have identified what to audit, you can begin setting up AuditTrail definitions. You should select tables for auditing that are functionally related. Create an Audit Group that contains these related tables.
- For each of the tables in the audit group, you must define which columns are to be audited. The primary key columns for these tables are automatically included.
- Enable auditing of a particular Oracle ID (schema) by defining audit installations. This enables you to audit across multiple application installations.
- After your definitions are complete, run the AuditTrail Update Tables Report program to create your audit tables and enable auditing for your audit group.
- There are no predefined audit reports. You can use any SQL tool to retrieve information from your audit tables.



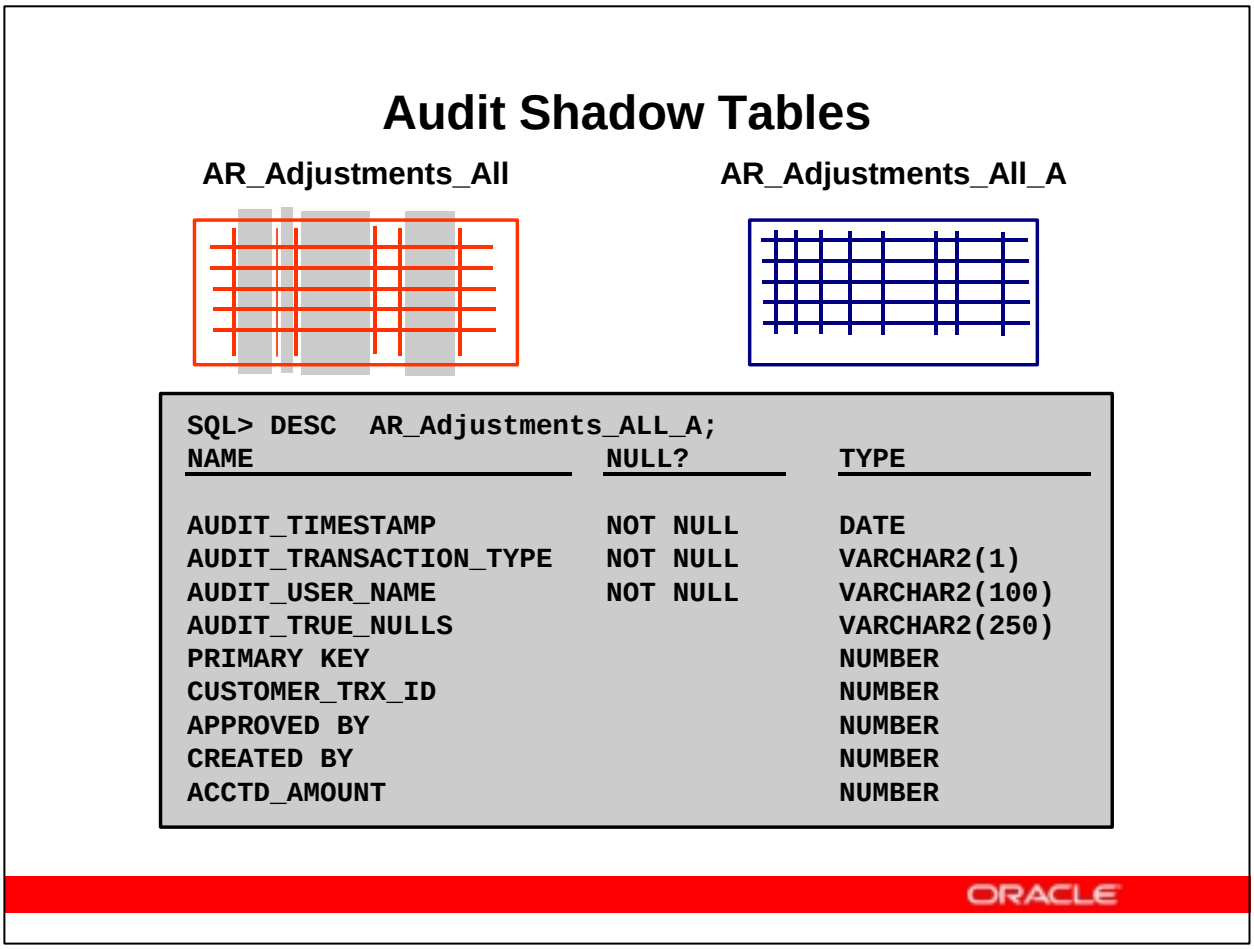
Determining Columns and Tables to Audit

You should select columns from tables that are functionally related. For example, if the accounting department wants to audit changes to any customer transactions involving cash receipts, columns from the Customer Transactions Lines table, the Cash Receipts table, and the Adjustments table would be good candidates for auditing.



Audit Groups

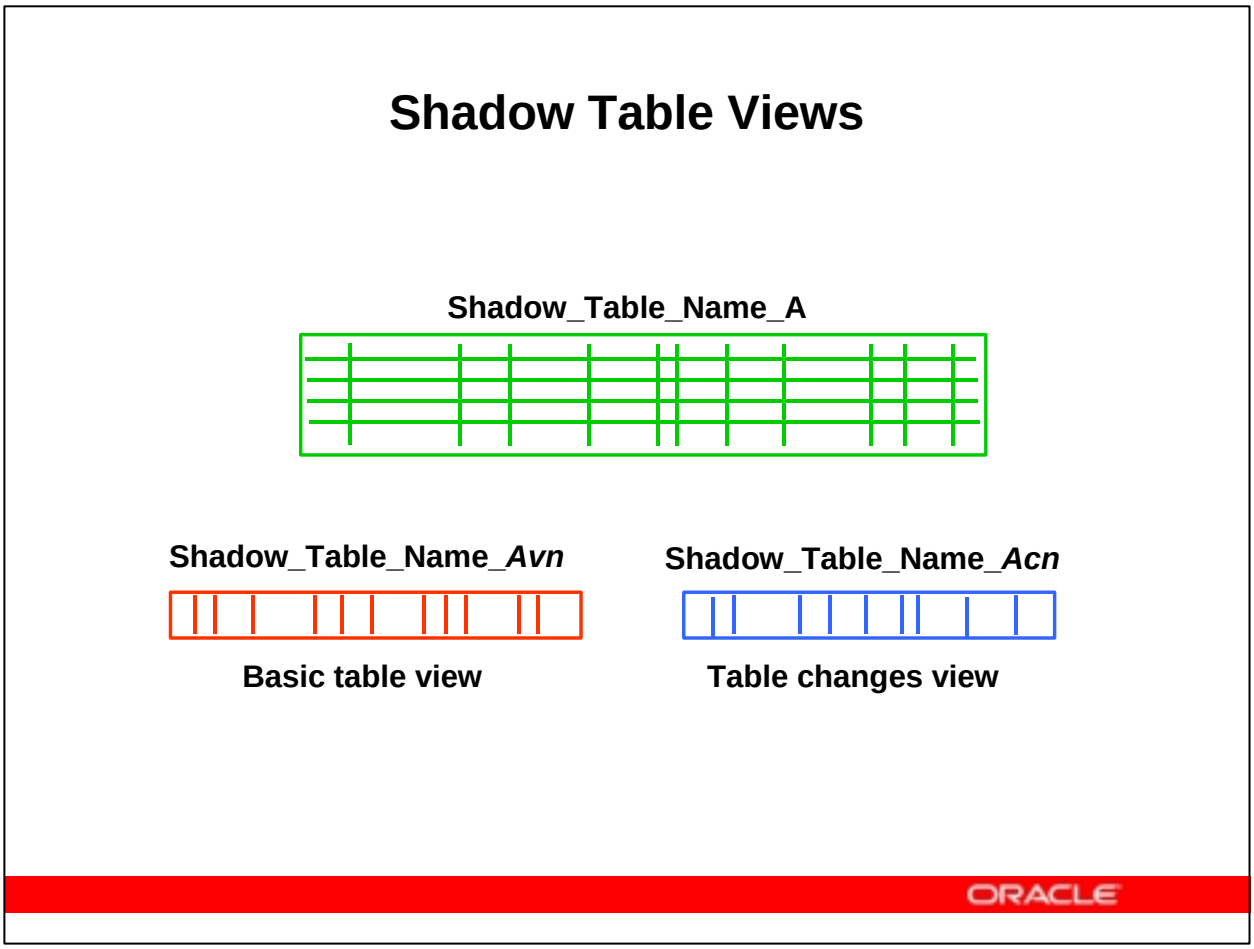
Once you have your tables and columns identified, you should group those tables into an audit group. In the figure, the three tables mentioned have been placed together in a Receivables Internal Control audit group.



What Audit Shadow Tables Identify

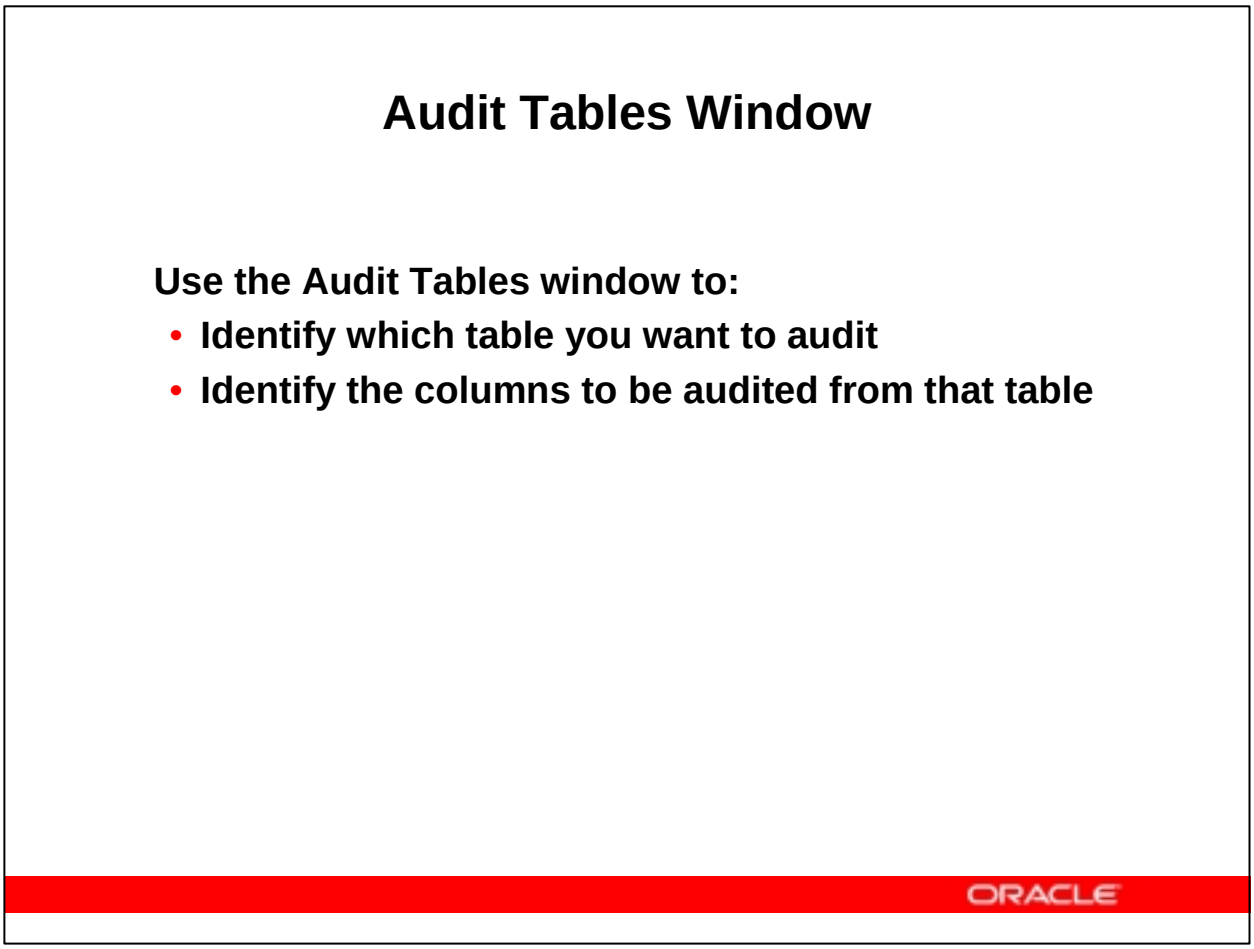
For each table being audited, an audit shadow table is created. This table contains audit processing data such as the transaction type (for instance, INSERT) and time-stamp data, the primary key columns of the audited table, and whatever columns are being audited. In the example above, the CUSTOMER_TRX_ID, APPROVED_BY, CREATED_BY, and ACCT._AMOUNT of the AR_ADJUSTMENTS_ALL table are being audited so the shadow table includes these columns as well as the other columns required by the audit process.

Shadow Table Views



Using Shadow Table Views

Two views are created for a shadow table. These views are called Shadow_Table_Name_*Avn* and Shadow_Table_Name_*Acn* where *n* is a number. If the number of columns being audited is very large, multiple views are created and numbered sequentially. The presentation of the audit table data is different depending on whether you are querying the AV view or the AC view. The AC view enables you to reconstruct the value for a row at a given time while the AV view provides simple access to when a value was changed.

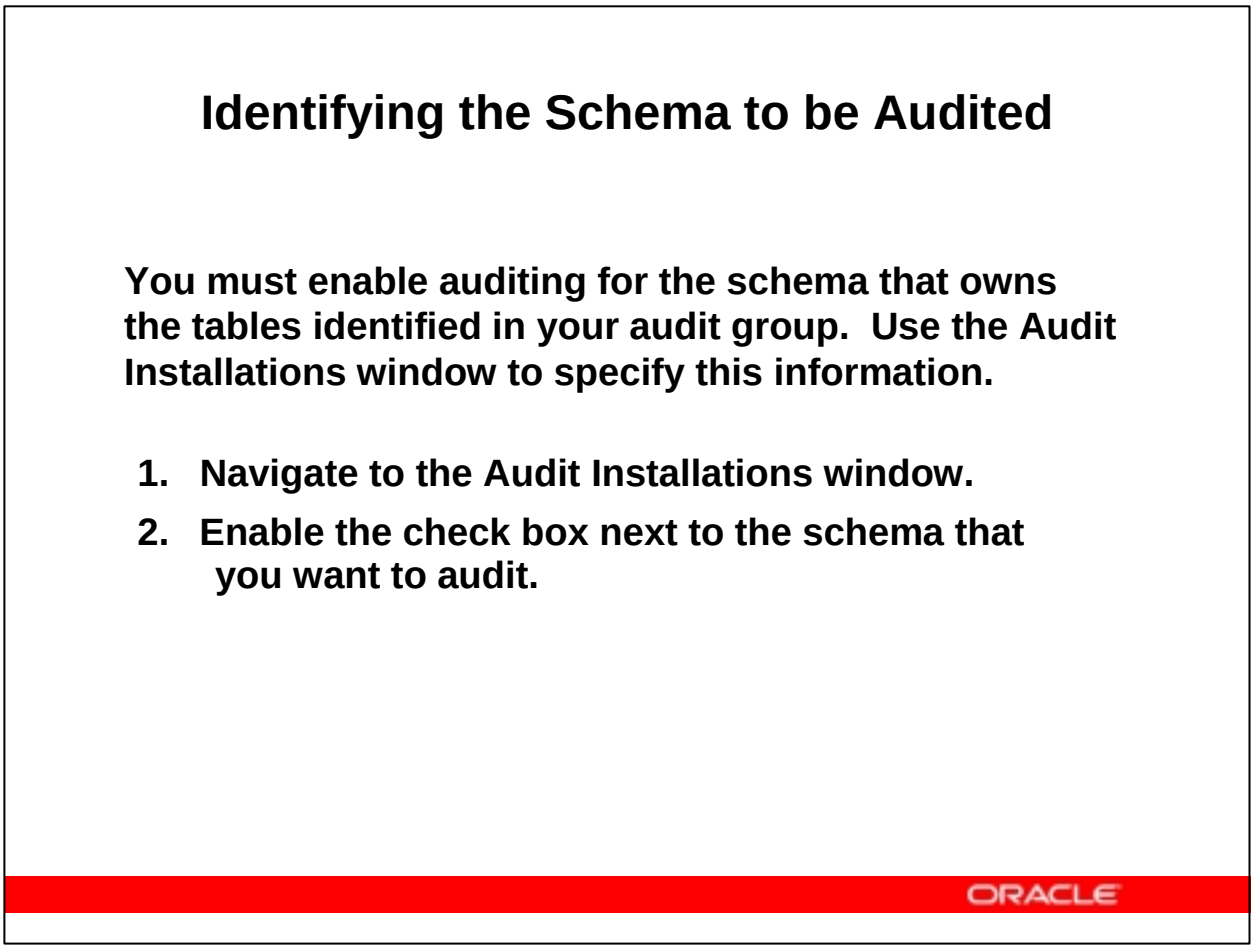


Audit Tables Window

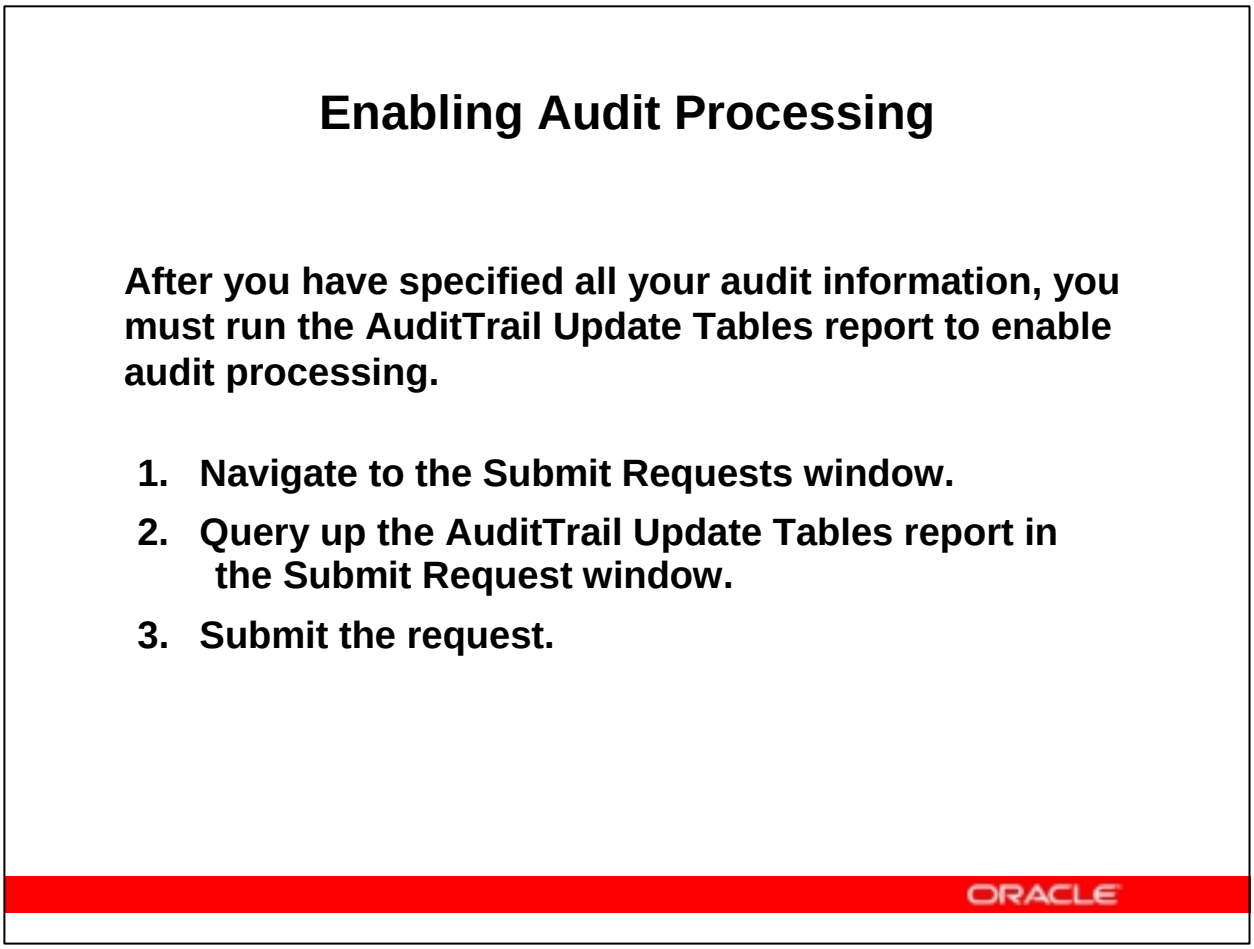
(N) Security > AuditTrail > Tables

For further descriptions of the Audit Tables window see:

(Help) Applied Technology > Oracle Applications System Administration > User and Data Auditing > Audit Tables Window



(N) Security > Audit Trail > Install



(N) Requests > Run

Practice - Auditing Resources

Overview

In this test, the committee wants to look at the results of several Signon Audit reports. They would also like you to test the Monitor Users form. They have asked you to complete the following tasks.

- Run the Signon Audit reports to determine what concurrent programs were run
- Run the Signon Audit reports to determine what responsibilities and users have been accessed
- Run the Signon Audit reports to see what forms have been accessed
- Use the Monitor Users form

Tasks

Run the Signon Audit Reports

1. Run the complete set of Signon Audit Reports using the time frame of the last week.
2. Answer the following questions.
   - What concurrent programs were run in the last week? _____
   - What responsibilities and users were accessed in the last week? _____
   - What forms were accessed in the last week? _____

Monitor Users

2. Access the Monitor Users form, and practice monitoring users.
3. If you don't see any users in your Monitor Users form, change the profile option "Sign-On Audit Level" to "Form" at the user level. Then, sign on again to your user and recheck the Monitor Users form.

Solution - Auditing Resources

Run the Signon Audit Reports

Responsibility = System Administrator

1. Navigate to (N) Requests > Run.
2. Click the “OK” button to accept the default to submit a single request.
3. Click the List of Values icon to get a list of all reports available for you to run.
4. Select the “Signon Audit Concurrent Requests” report.
5. Enter the following in the Parameters window:
    - **Sort By**: User Name
    - **Request Start Time**: one week ago (use the format 01JAN2002)
6. Click the “Submit” button.
7. Click the “Yes” button to submit another request.
8. Select the “Signon Audit Responsibilities” report.
9. Enter the following in the Parameters window:
    - **Sort By**: User Name
    - **Request Start Time**: one week ago (use the format 01JAN2002)
10. Click the “Submit” button.
11. Click the “Yes” button to submit another request.
12. Select the “Signon Audit Forms” report.
13. Enter the following in the Parameters window:
    - **Sort By**: User Name
    - **Request Start Time**: one week ago (use the format 01JAN2002)
14. Click the “Submit” button.
15. Click the “No” button.
16. Navigate to (N) Requests > View.

17. View each report, and answer the following questions:

    - What concurrent programs were run in the last week? _____
    - What responsibilities and users were accessed in the last week? _____
    - What forms were accessed in the last week? _____

Monitor Users

18. Navigate to (N) Security > User > Monitor.
19. Query up your User Name and note the activity.
20. Navigate to (N) Profile > System.
21. Find the "Sign-On Audit Level" profile option at the user level for your specific signon. Change it to "FORM" or "NONE" (**Note**: You want to change it to something other than what it is currently set to. This will enable you to note the differences.)
22. Save your work.
23. Sign off and on again.
24. Navigate to (N) Security > User > Monitor.
25. Review the changes of the data in the form.

Summary

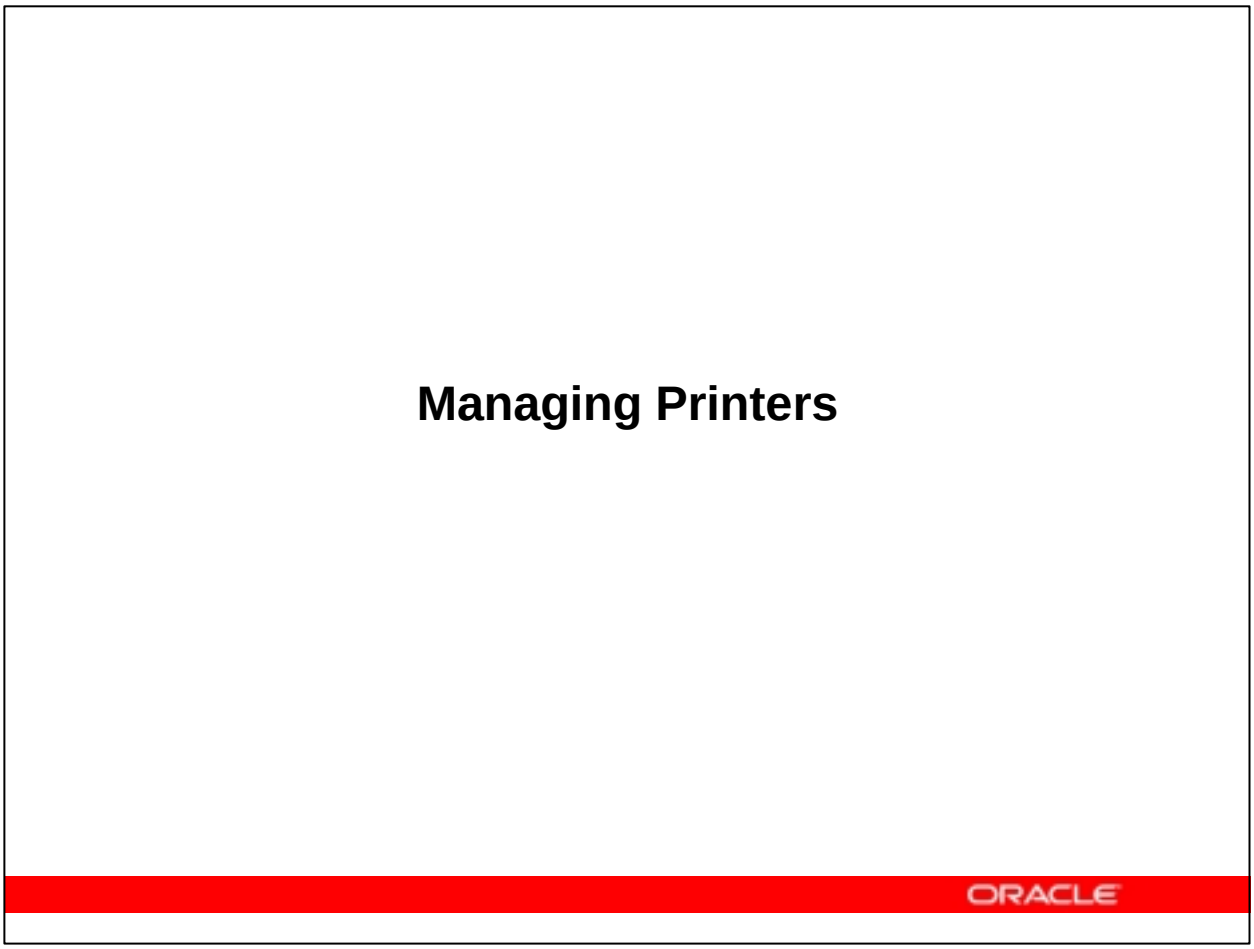
You should now be able to do the following:

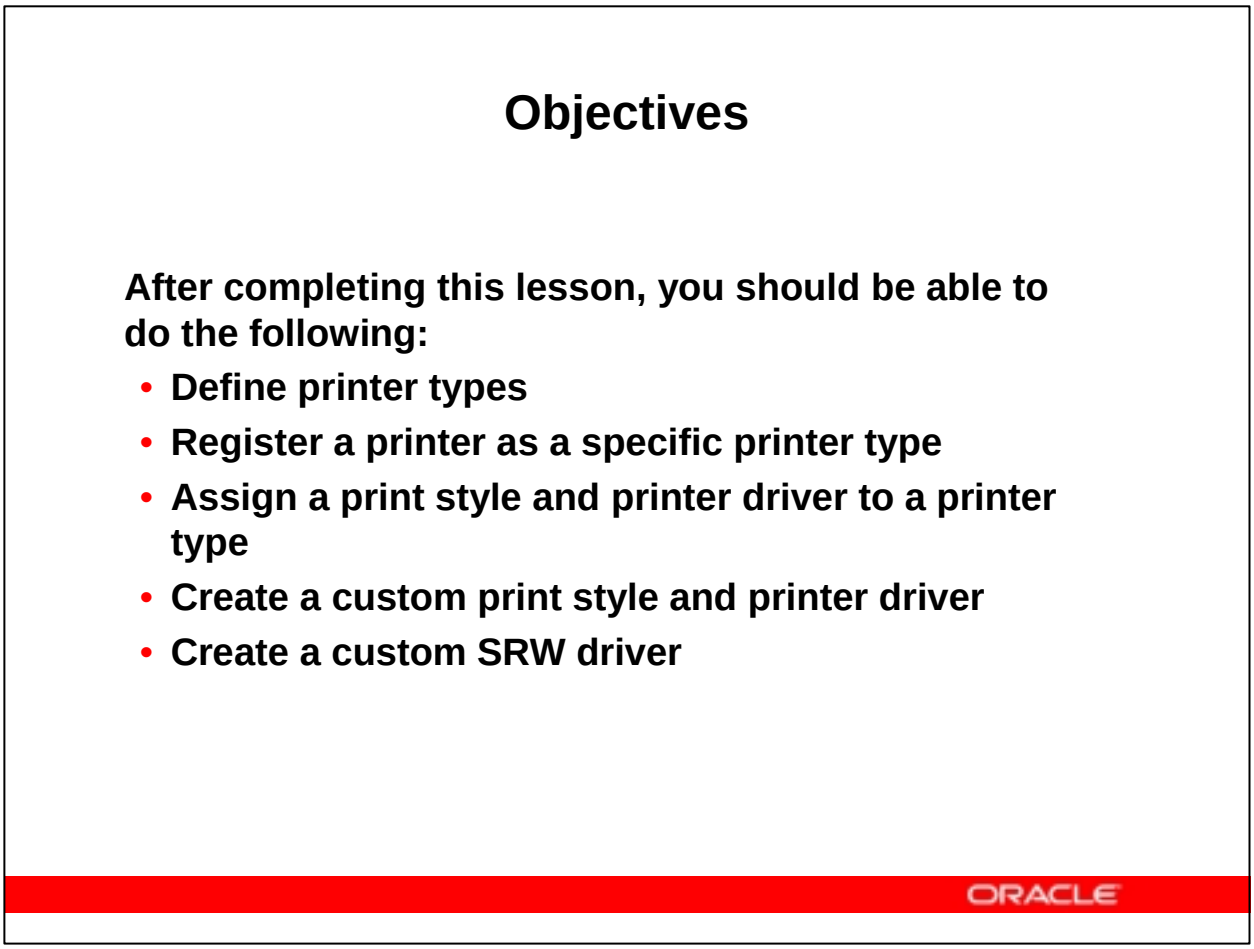
- **Identify different ways of auditing Oracle Applications performance and resources**
- **Modify auditing-related profile options**
- **Identify auditing-related reports**
- **Selectively implement auditing as appropriate to your environment**

ORACLE

Managing Printers

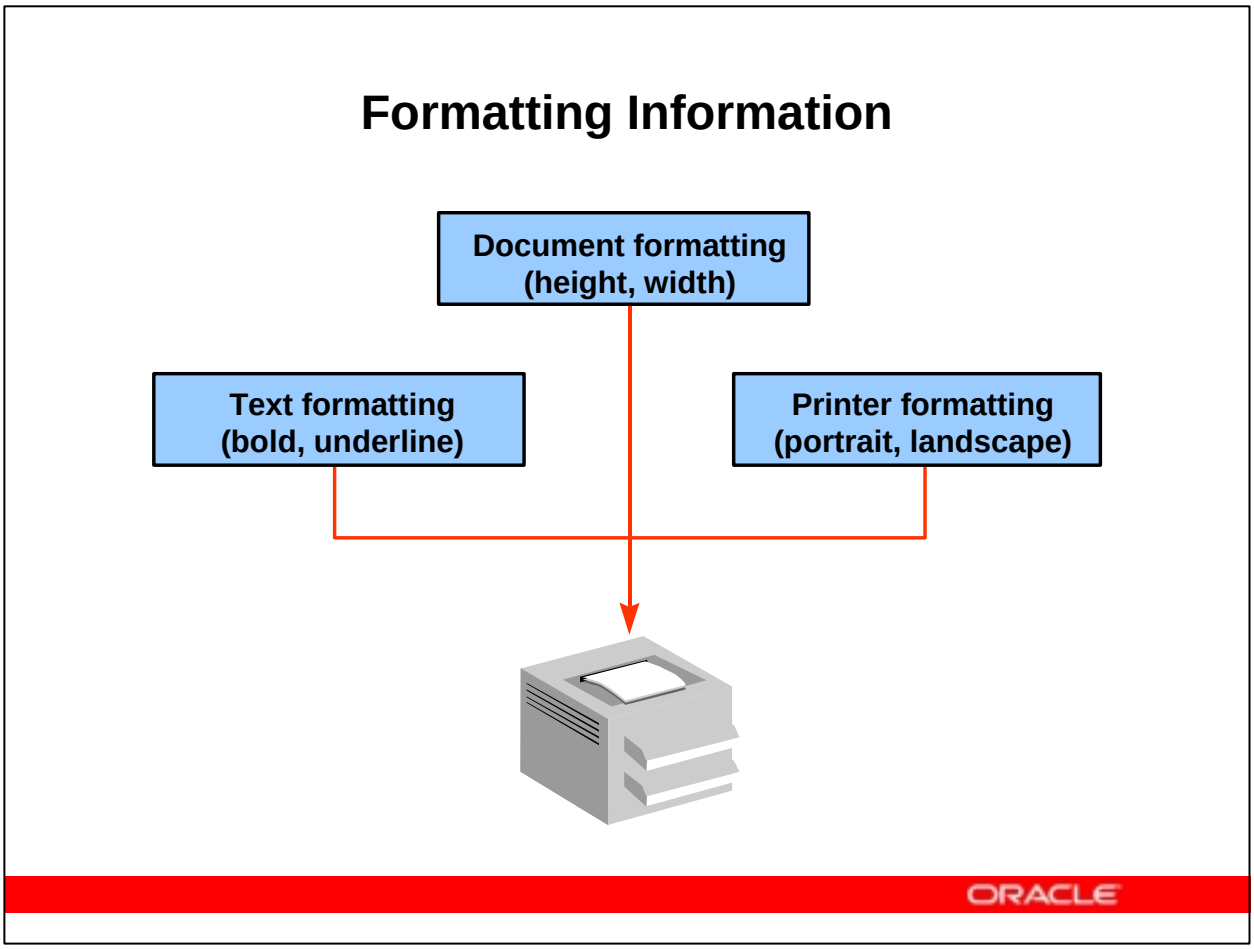
Chapter 8





Lesson Aim

Installing a new printer is a typical responsibility of the system administrator. A less common need is to customize the software that supports printers. This lesson shows you how to accomplish both tasks.



Formatting Information

For a printer to produce a report, it requires three basic types of formatting instructions: text, document, and printer.

Text Formatting

- Oracle Reports generates reports for the various Oracle Applications.
- When Oracle Reports formats text as bold or underlined, and sets page breaks, formatting codes are used.
- A SQL*ReportWriter (SRW) driver defines text formatting codes used by Oracle Reports.

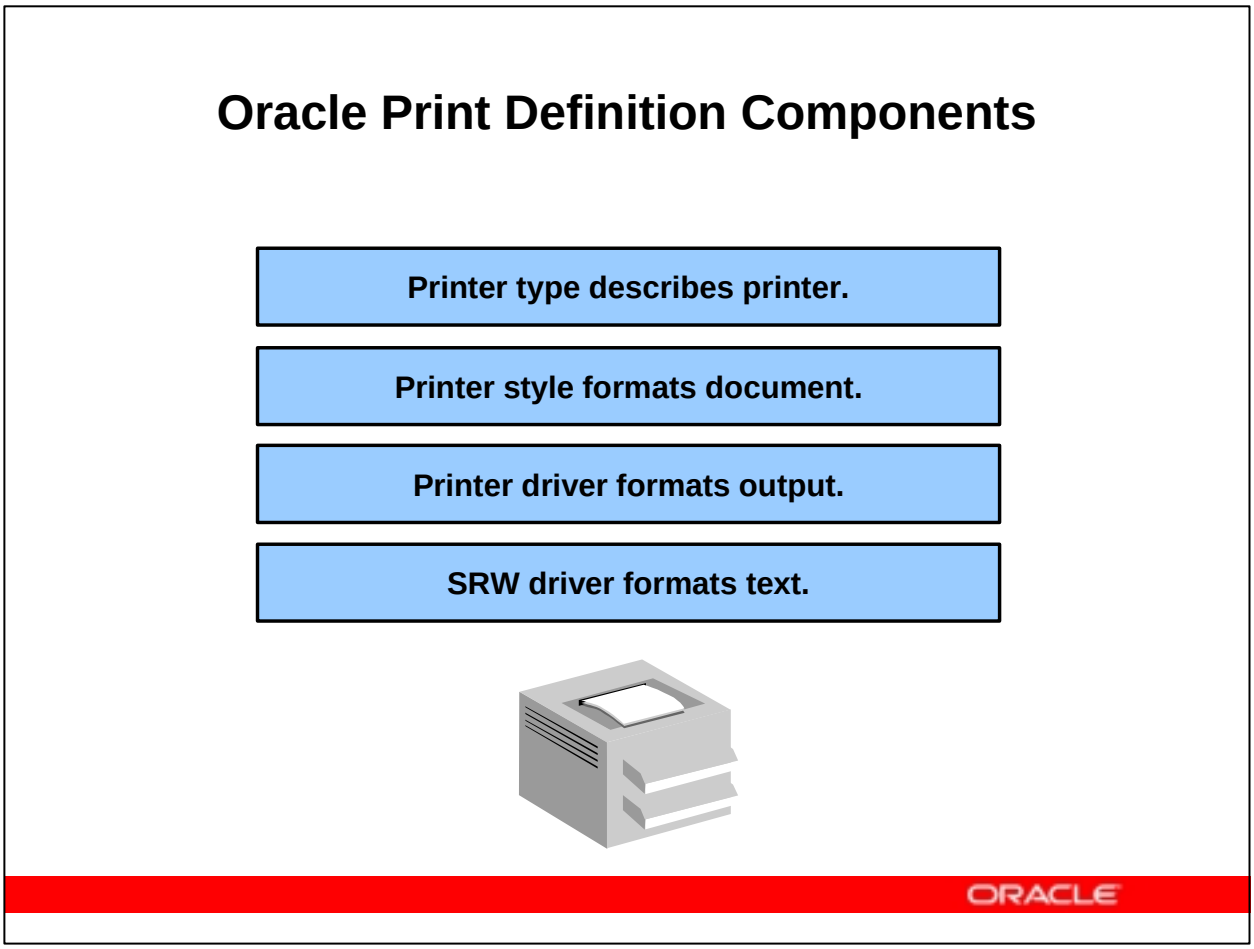
Document Formatting

- The dimensions for a report output file are predefined as the number of columns and rows (width and height).
- A print style defines the dimensions for a report output file.

Printer Formatting

- A printer prints in a certain orientation, portrait or landscape.
- A printer can also start printing with a specific font and type size. These instructions are contained in an initialization string.

- A printer driver defines initialization and reset strings.



Required Formatting Information for Components

The formatting information required by your printer is specified by the components shown on this figure.

Printer Type

Describes what kind of a printer you have. This is the manufacturer and model—for example, a DEC LN03 printer or an HP LaserJet III printer.

Printer Style

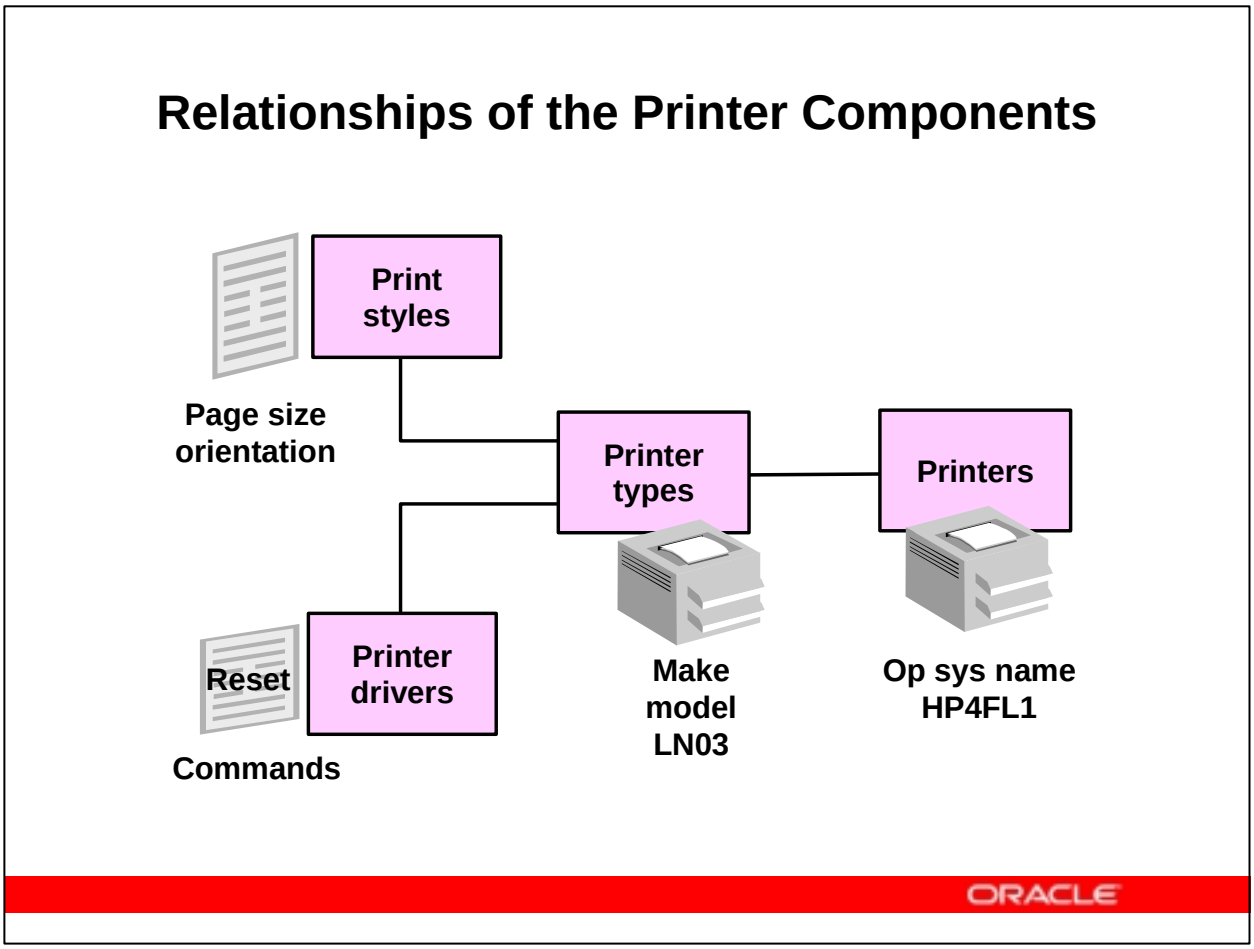
- Describes how a report should be printed
- Determines the number of lines (rows) per page
- Determines the width of each line (number of columns)
- Controls whether to print a header page

Printer Driver

- Contains the set of commands that tell a printer how to print in the print style chosen
- Initializes printing orientation
- Resets printer driver's instructions for next print job

Printer Drivers Versus SRW

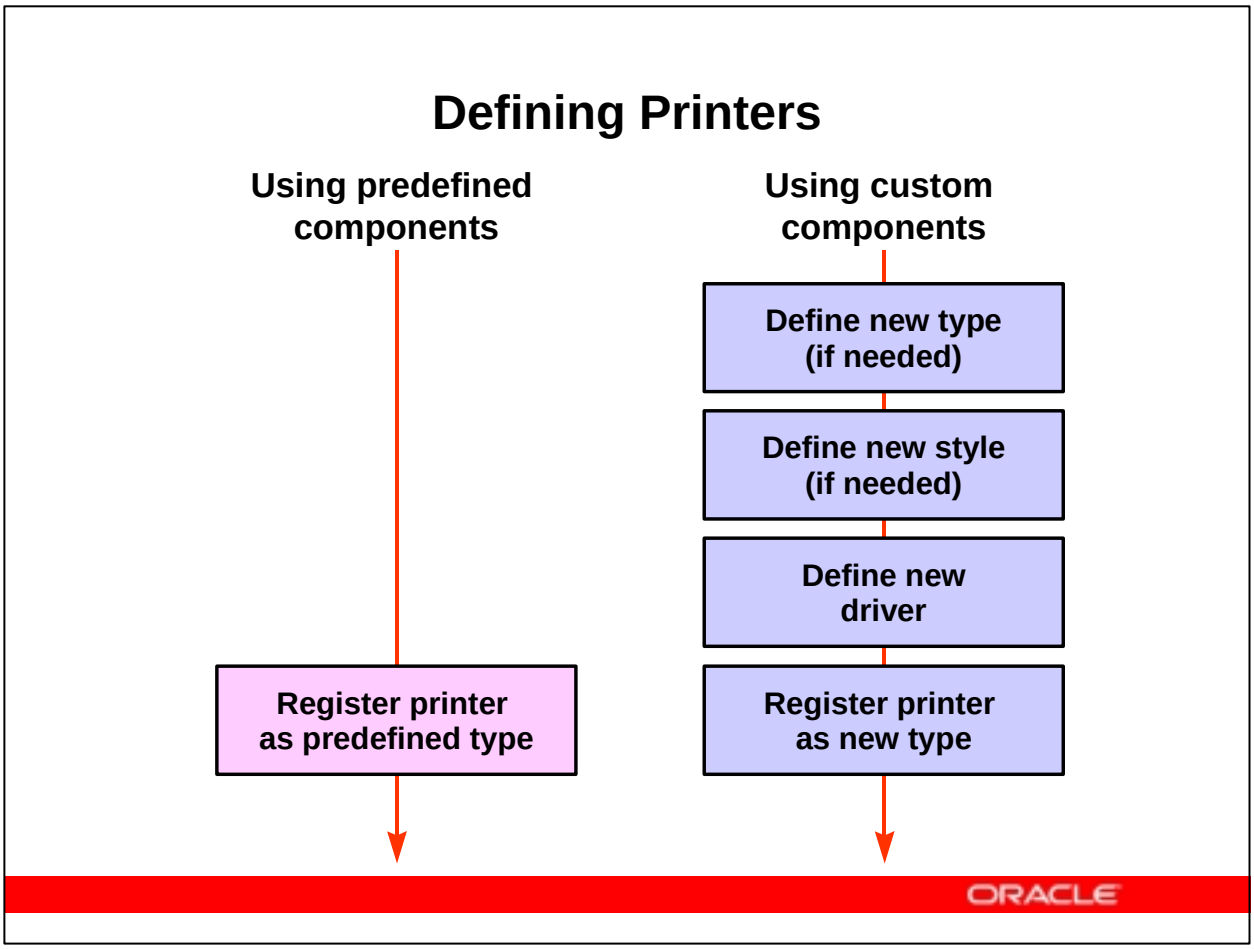
- A printer driver formats the destination printer.
- An SRW driver formats text and sets page breaks in an Oracle Reports file.



How the Print Components Are Related

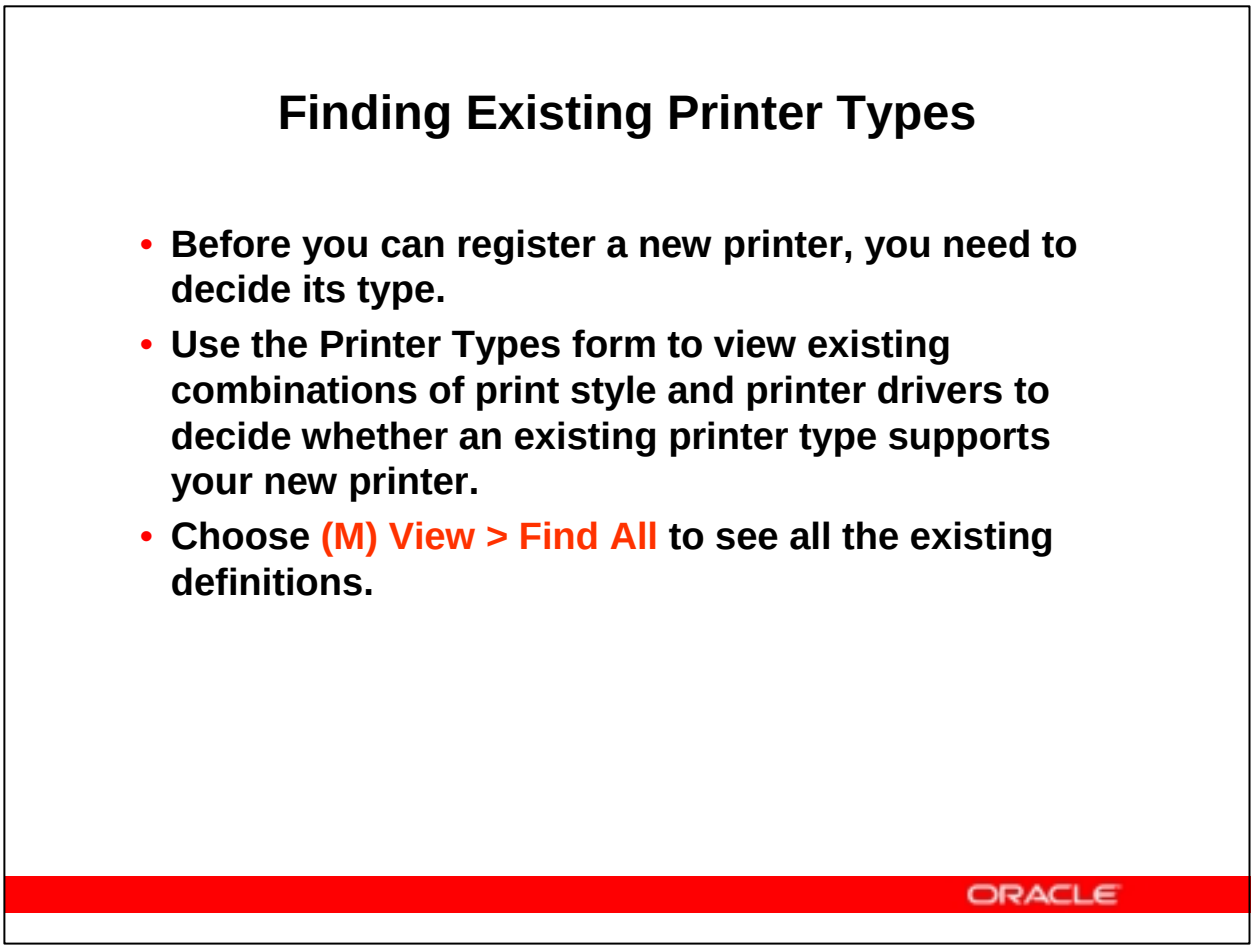
Oracle Applications uses the components shown on the previous figure to store the information necessary to print a report. This figure shows how the components are related to provide a flexible method of defining your print environment. The printer type is at the heart of your print definitions. The figure shows that:

- A printer is registered as a printer type.
- A printer style is associated with a printer type.
- A printer driver is assigned to a printer type.



How to Define Printers

The figure shows the sequence of operations to define printers. Generally, you simply register a new printer by using existing definitions as shown in the path on the left. However, in some cases you must modify existing definitions or create new definitions to accommodate a new printer as shown on the right.

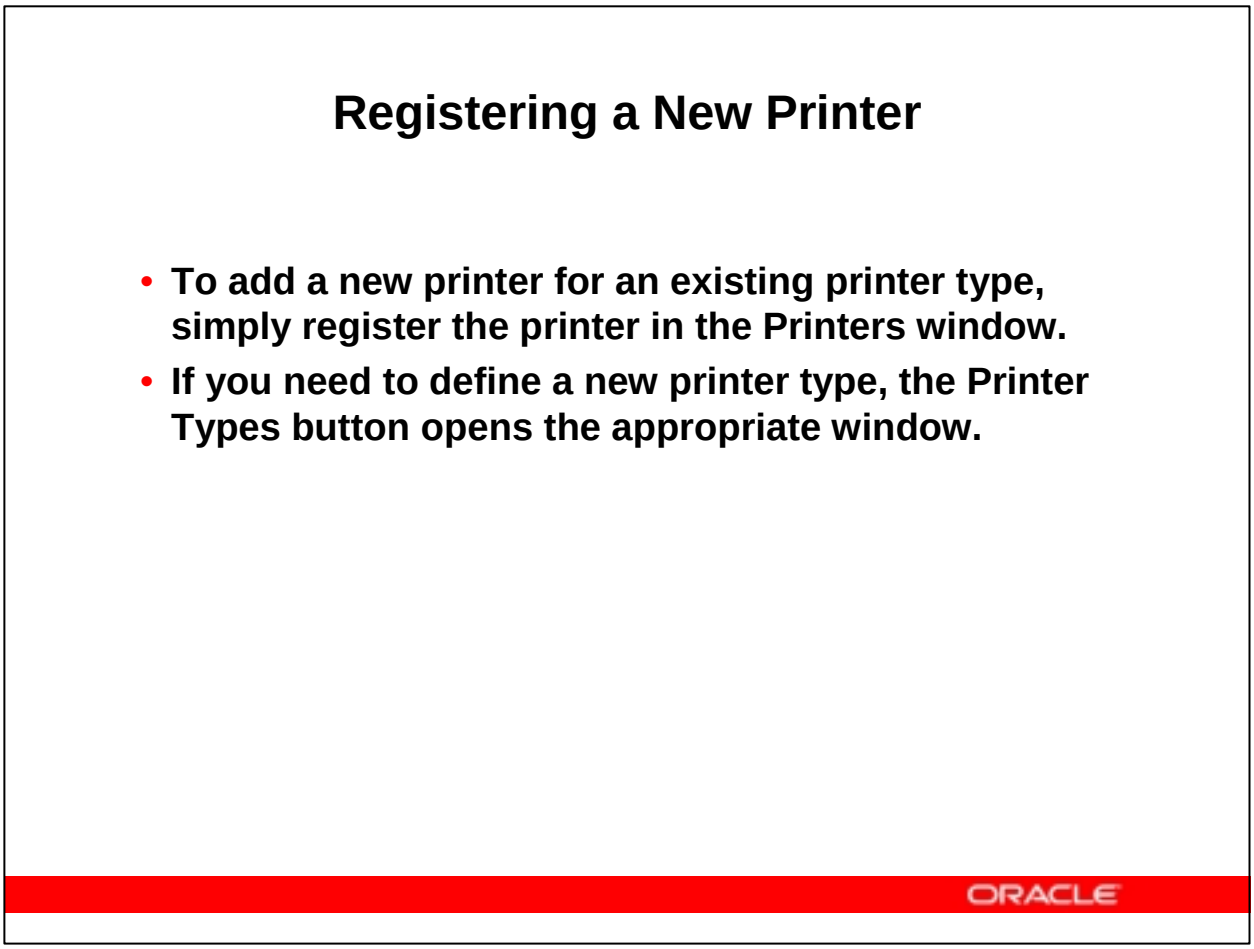


Printer Types Window

(N) Install > Printer > Types

For a complete explanation of the fields in this form see:

(Help) Applied Technology > Oracle Applications System Administration > Printers > Printer Types Window

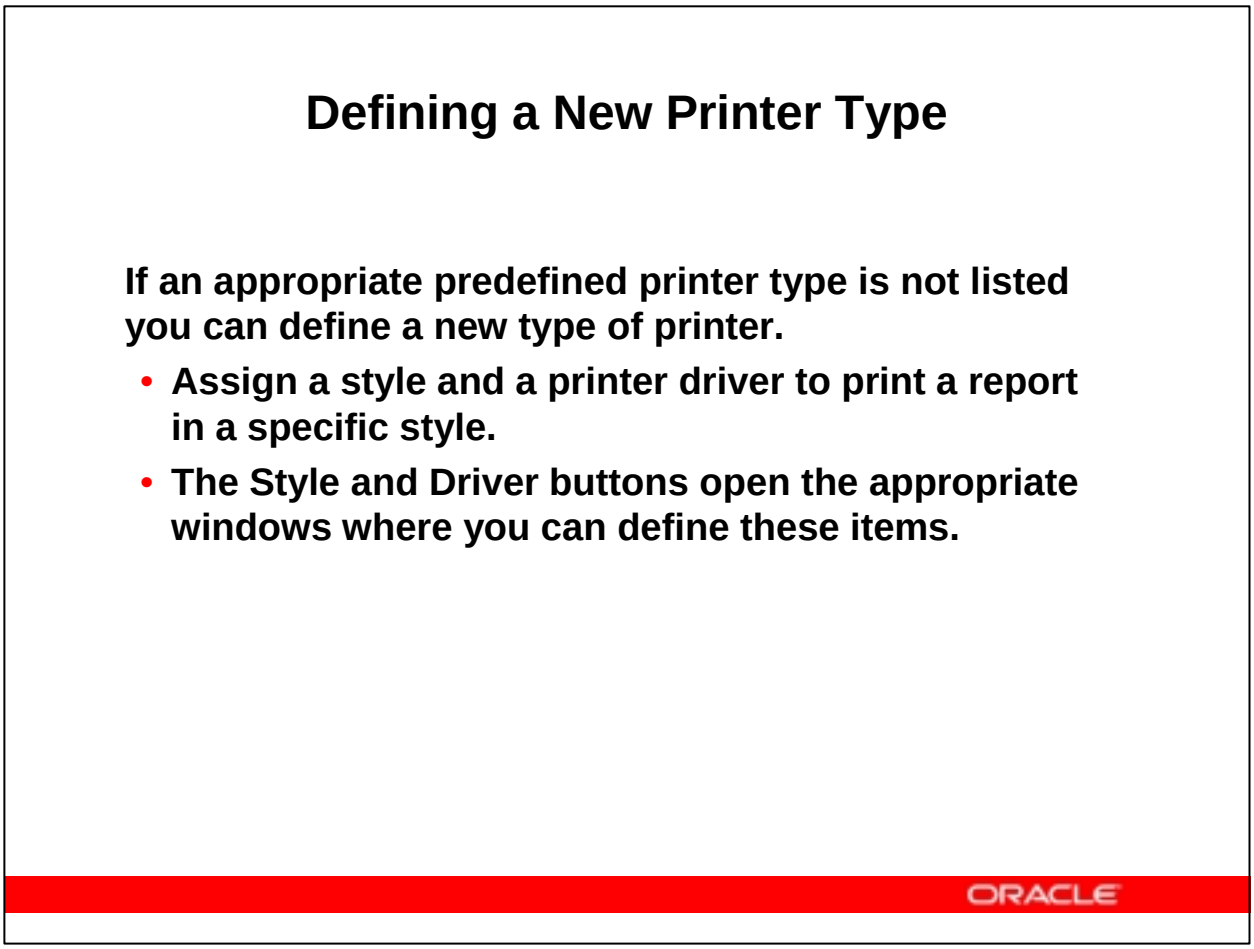


Registering a New Printer

(N) Install > Printer > Register

For a complete explanation of the fields in this form see:

(Help) Applied Technology > Oracle Applications System Administration > Printers > Printers Window

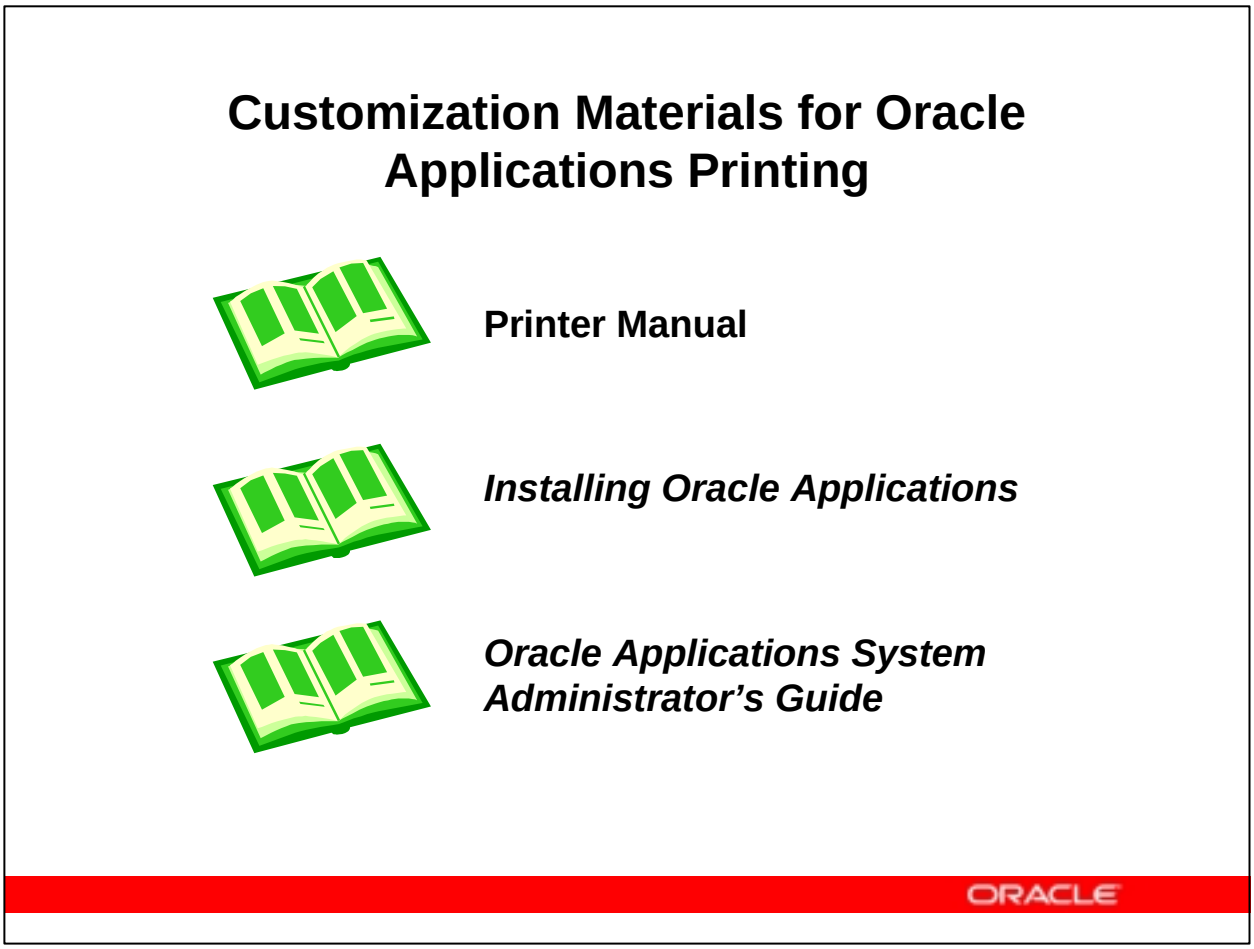


Defining a New Printer Type

(N) Install > Printer > Types

For a complete explanation of the fields in this form see:

(Help) Applied Technology > Oracle Applications System Administration > Printers > Printer Types Window

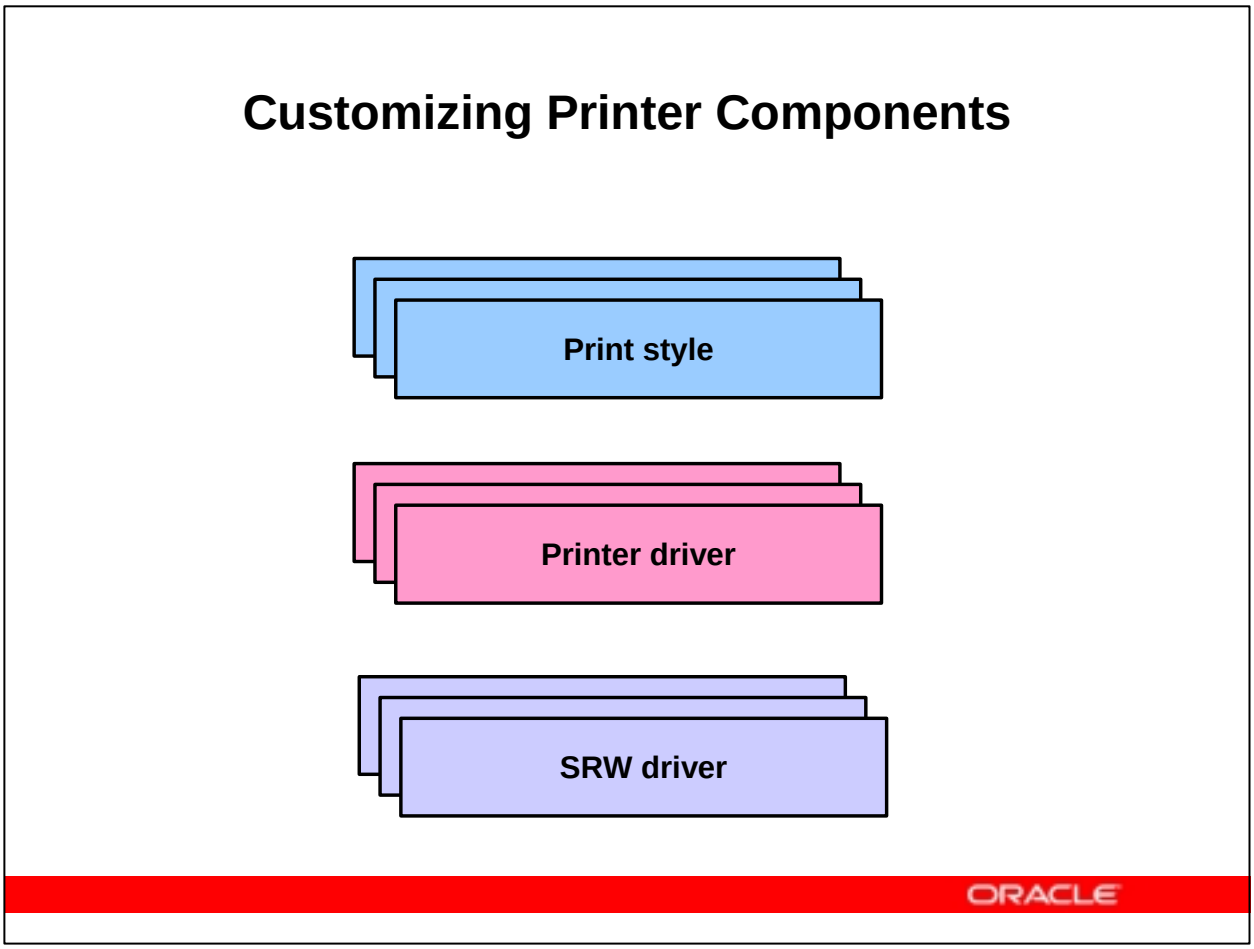


Customization Considerations

You can customize Oracle Applications printer support components to use custom print styles and custom print programs.

Materials Needed:

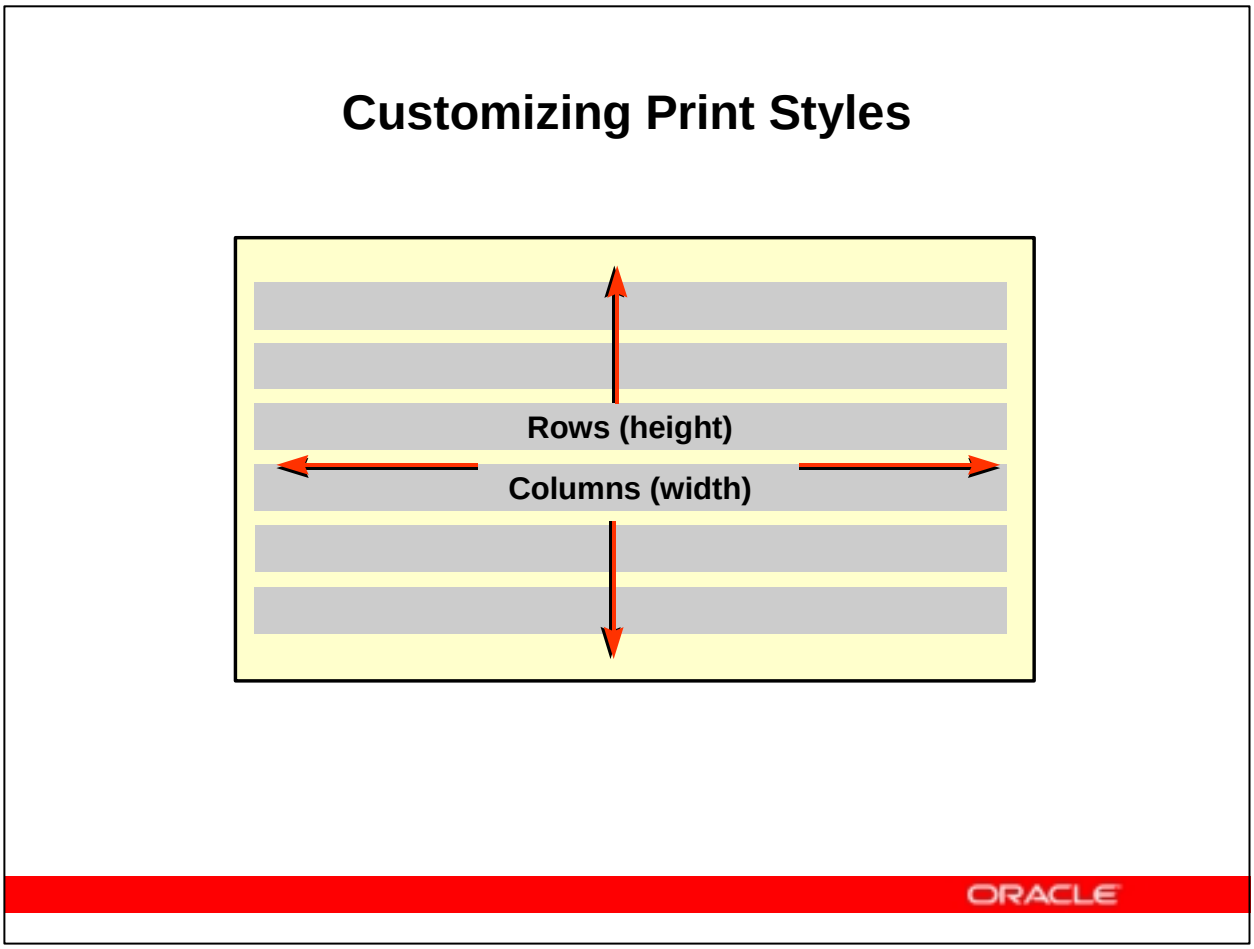
- Manual for your specific printer to look up control codes required by a printer type
- *Installing Oracle Applications* for your platform to look up control codes required by an operating system platform
- *Oracle Applications System Administrator's Guide*



Ways to Customize Printer Components

You can customize your Oracle Applications printing capabilities by:

- Customizing a print style using the Print Styles window
- Customizing a printer driver using the Printer Drivers window
- Creating or modifying an SRW (SQL*ReportWriter) driver using a text editor



Customizing Print Report Rows and Columns

(N) Install > Printer > Style

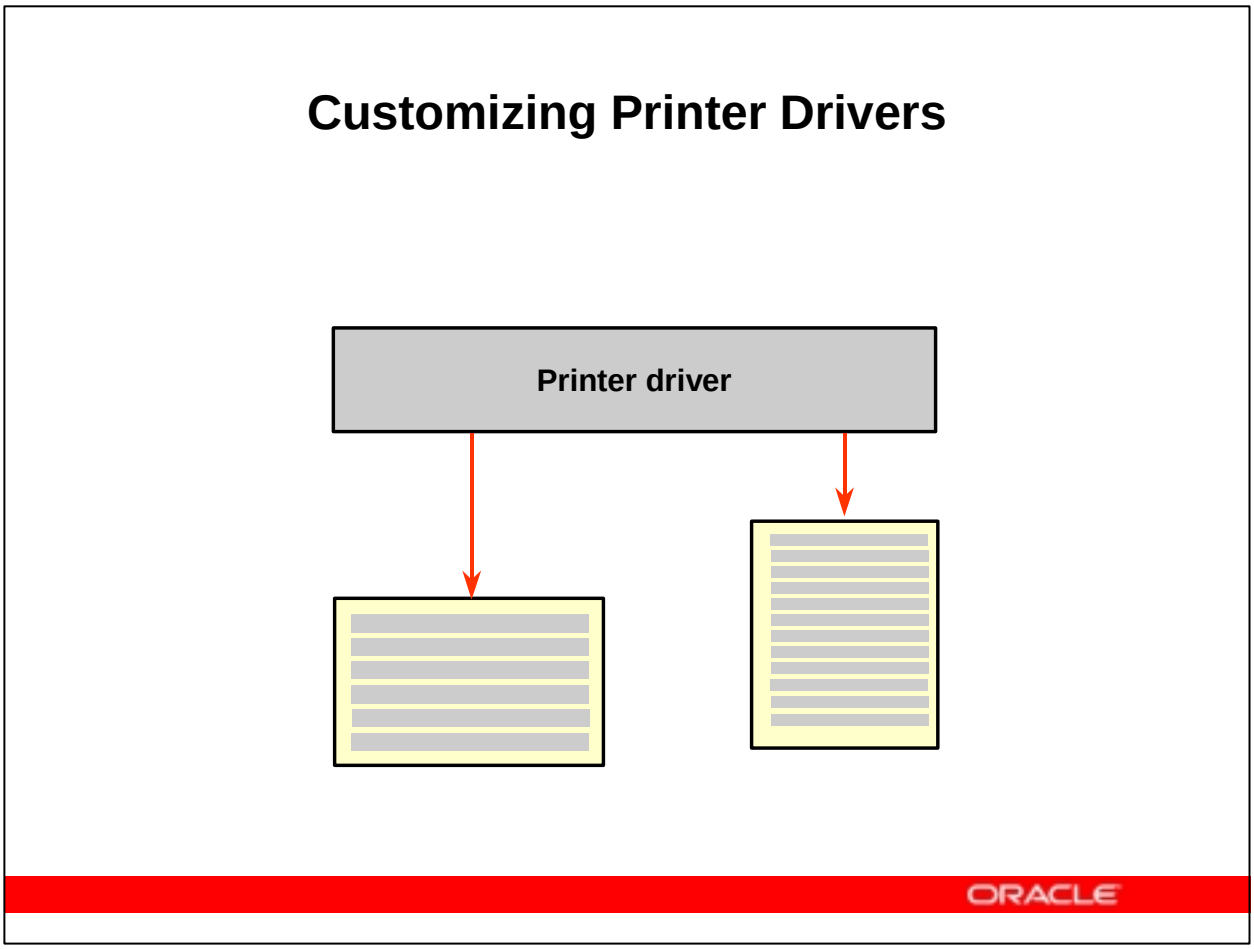
You can define a custom print style to change the number of columns (width) and rows (height) for a report.

Style Requirements

- A print style must satisfy the concurrent program's definition of minimum and maximum number of columns (width) and rows (height).
- A print style must be assigned to the target printer type.

For a complete explanation of the fields in the Print Styles window see:

(Help) Applied Technology > Oracle Applications System Administration > Printers > Printer Styles Window



Customizing Printer Drivers

(N) Install > Printer > Driver

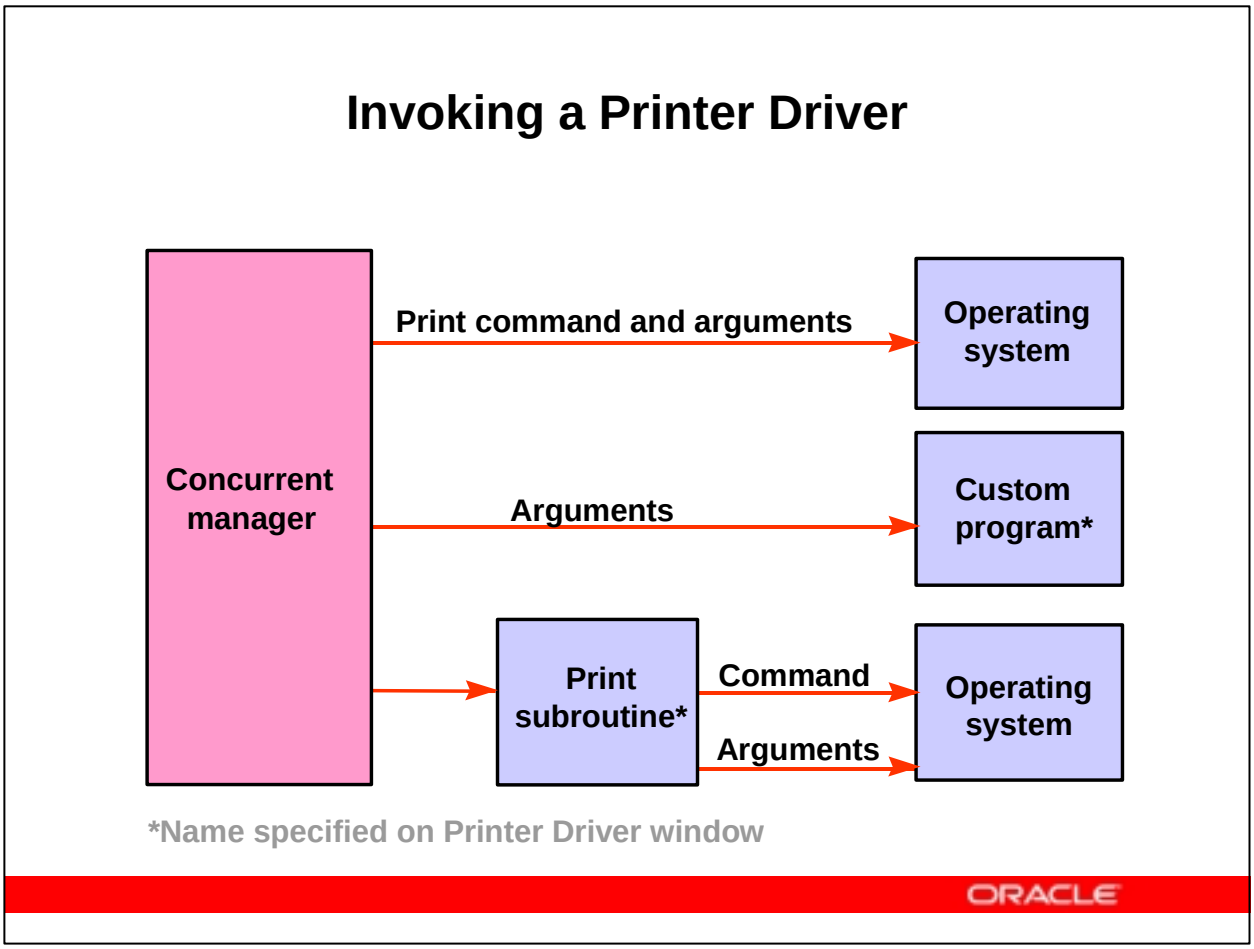
Printer drivers are responsible for communicating to the printer the information necessary to print the report in the style required by the printer style selected. Create custom printer drivers to support print styles for new or existing printer types. Here are some other possible situations in which you could need to define a new printer driver.

- Customize a printer driver when a printer type requires different control characters.
- Customize a printer driver when the control characters have a different meaning because of your operating system and platform.
- Customize a printer driver when language translation changes the meaning of the control characters.
- Customize a printer driver when the printer needs special control characters to select different character sets.
- Customize a printer driver to change the printer's default font for a report (Initialization string only).

Refer to the printer manufacturer's documentation for the control codes and escape sequences the printer understands. Refer to the printer's manual—for example, a PCL or PostScript manual—for additional information if necessary.

For a complete explanation of the fields in the Printer Drivers window see:

(Help) Applied Technology > Oracle Applications System Administration > Printers > Printer Drivers Window



Invoking a Printer Driver

There are three methods to invoke a printer driver.

Command

- The concurrent manager issues an operating system print command and its arguments.
- An operating system print command along with all its arguments is entered in the Arguments field of the Printer Drivers form.

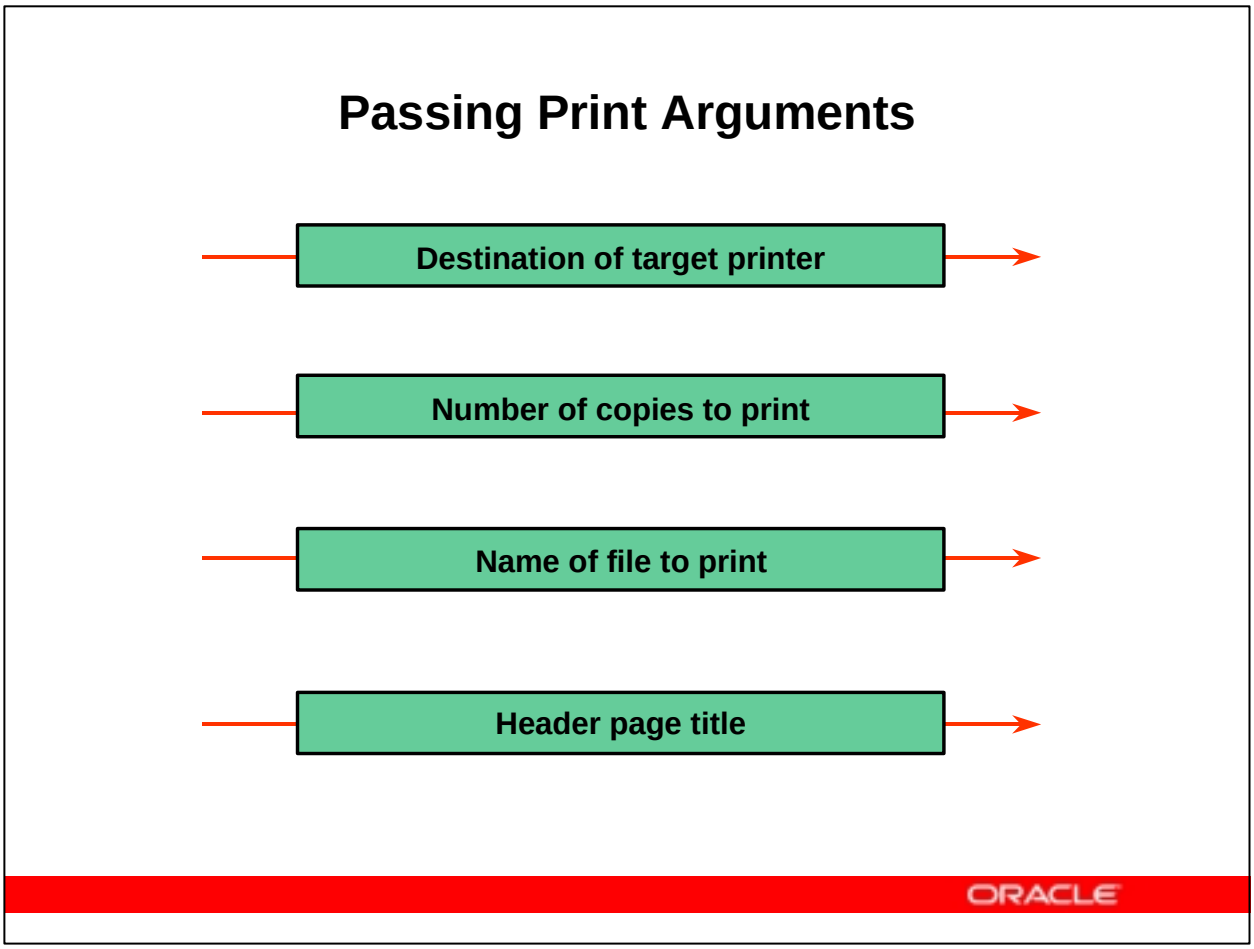
Program

- The concurrent manager calls a custom print program and passes arguments to the program.
- The name of a custom print program is entered in the Name field, and any arguments to be passed to the program are entered in the Arguments field of the Printer Drivers form.

Subroutine

- The concurrent manager calls a predefined Oracle Applications subroutine that passes a print command and arguments to the printer via the operating system.
- A subroutine name appears in the Name field of the Printer Drivers form.

- The Arguments field is disregarded. The concurrent manager reads the Initialization and Reset escape sequences.
- On UNIX systems this method, unlike the command method, does not start an operating system shell along with the print command.



Types of Print Arguments Passed

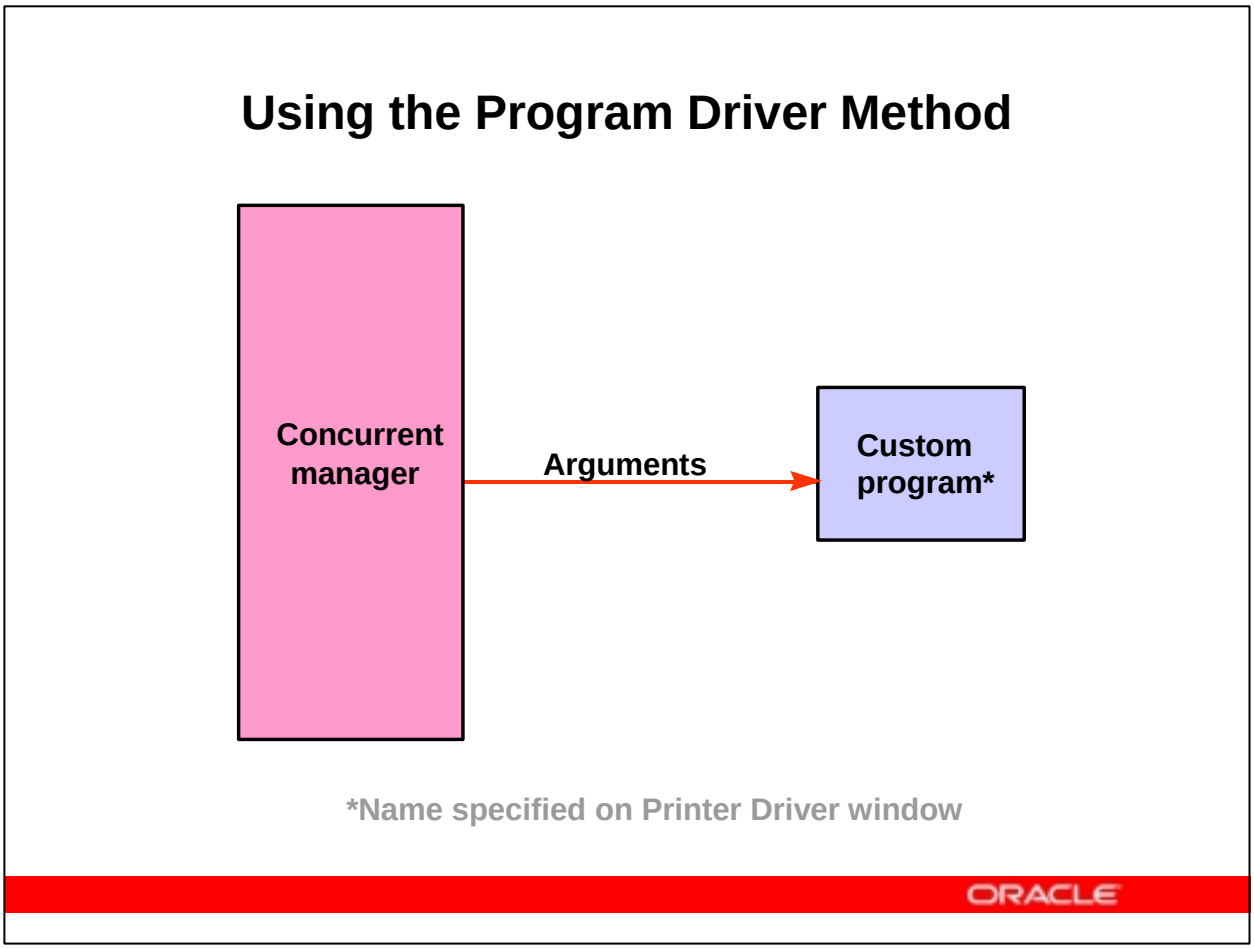
You can rely on a concurrent manager to supply four values as arguments to the operating system print command it issues or a custom print program that it calls.

Print Arguments

- Destination or target printer
- Number of copies to print
- Name of the file to print
- Title that appears on the header page
- The header page is printed when the Suppress Header Page field is set to No in the Print Styles window.

Print Commands

- Print commands vary; however, the tokens for which values are retrieved are always the same.
- Print commands are operating system-dependent. Refer to *Installing Oracle Applications*.



Program Driver Method Considerations

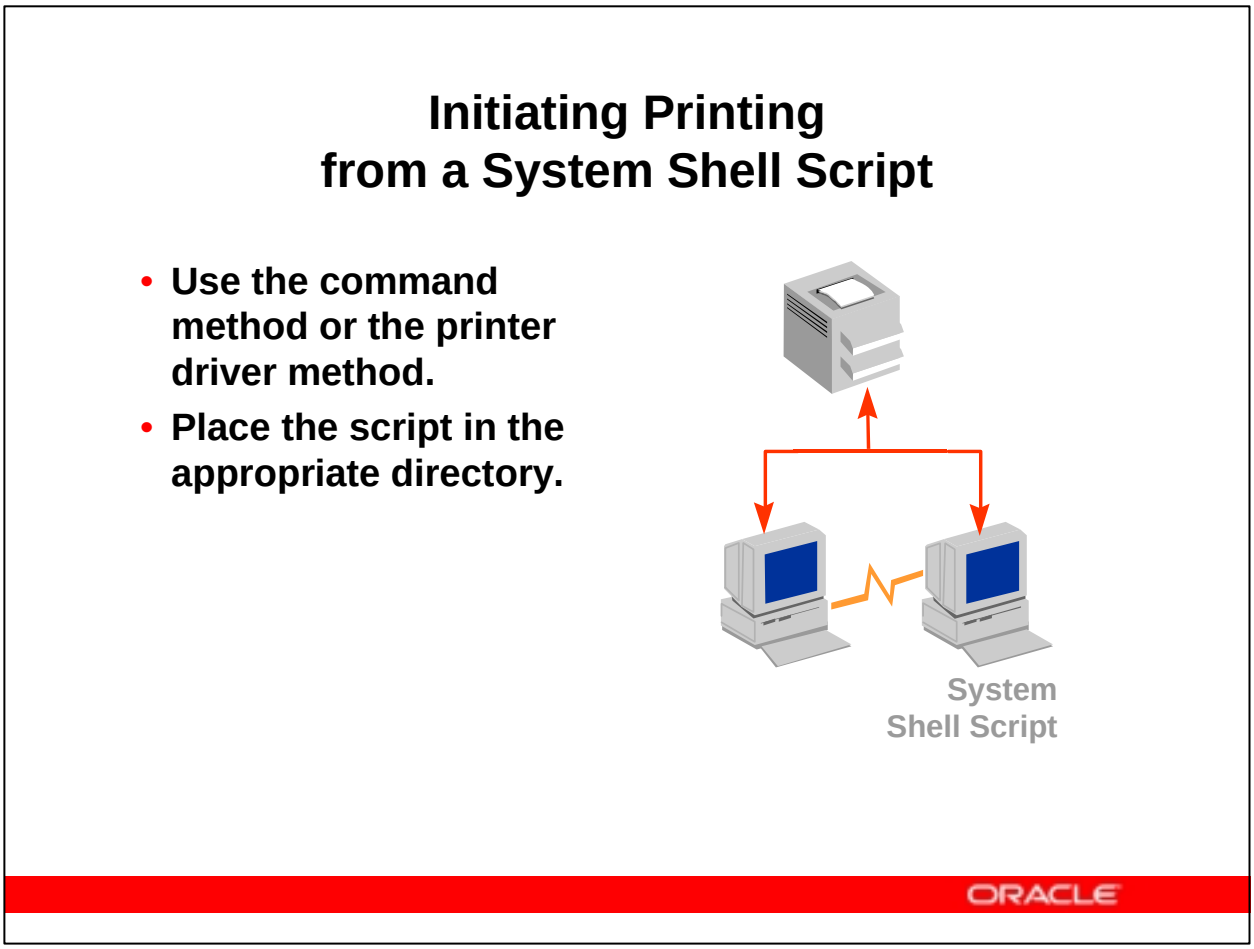
(N) Install > Printer > Driver

Location of Custom Print Programs

To call a custom print program by using the Printer Drivers window, enter the program name, including the full path to the program, in the Program Name field. The path to the program name is not necessary if the program location can be identified by the operating system's PATH environment variable—that is, it is in the $PATH variable name.

For platforms where the equivalent of a $PATH variable does not exist, use the full path name. A path can be up to 255 characters.

Custom print programs are not registered as concurrent programs with Oracle Application Object Library but are called after the concurrent process has completed.



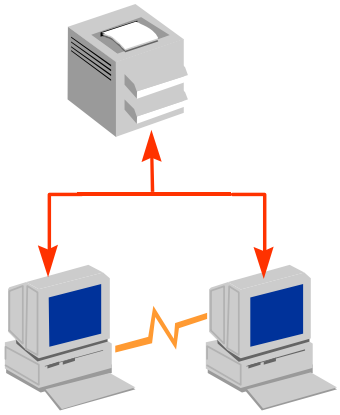
How to Initiate Printing from a System Shell Script

To initiate printing from an operating system shell script:

- For operating system shell scripts, the printer driver method can be either Command or Program, as long as you populate the argument field correctly.
- The script for a command shell procedure—for example, a UNIX shell or a VMS dcl—should reside in $FND_TOP/$APPLBIN.

Printing from Standard Input Directly to the Printer

- **Invoke the printer driver by using the command method.**
- **Select the Standard Input check box.**



Command Method

ORACLE

How to Print from Standard Input

Standard input refers typically to data or commands entered from the keyboard. It can be from a redirected source such as a file or the output of a program.

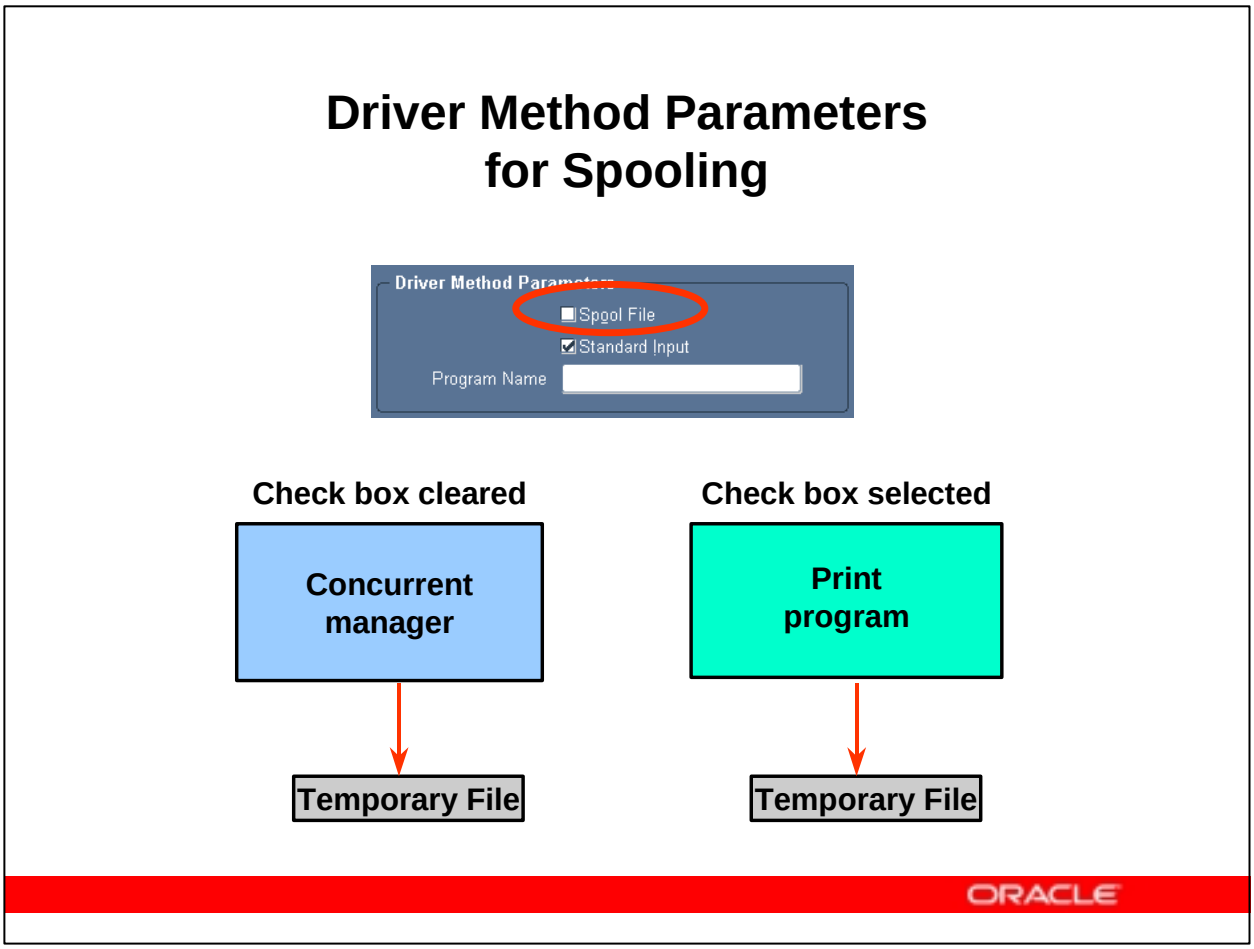
Examples of Using Standard Input

- When running a pipe in UNIX such as “cat filename | lpr” rather than just “lpr filename,” the output file is sent to the stdin (standard input).
- The UNIX print command lpr accepts standard input when a filename is not specified.

Standard Input Settings

- When you select the Standard Input check box, the printer driver can send standard input to the printer.
- Clear the Standard Input check box if the driver method is Program or Subroutine or the operating system print command or print program does not accept standard input.
- When the Standard Input check box is cleared, the print command issued by the concurrent manager runs asynchronously. The concurrent manager issues the command and does not wait for an operating system response.

Driver Method Parameters for Spooling



Setting Spooling Specifications

A spool file contains a copy of a file to be printed and the initialization and reset strings of the printer driver. Spooling is the process of sending a spool file to the spooling area of the printer. Spool files are deleted after printing.

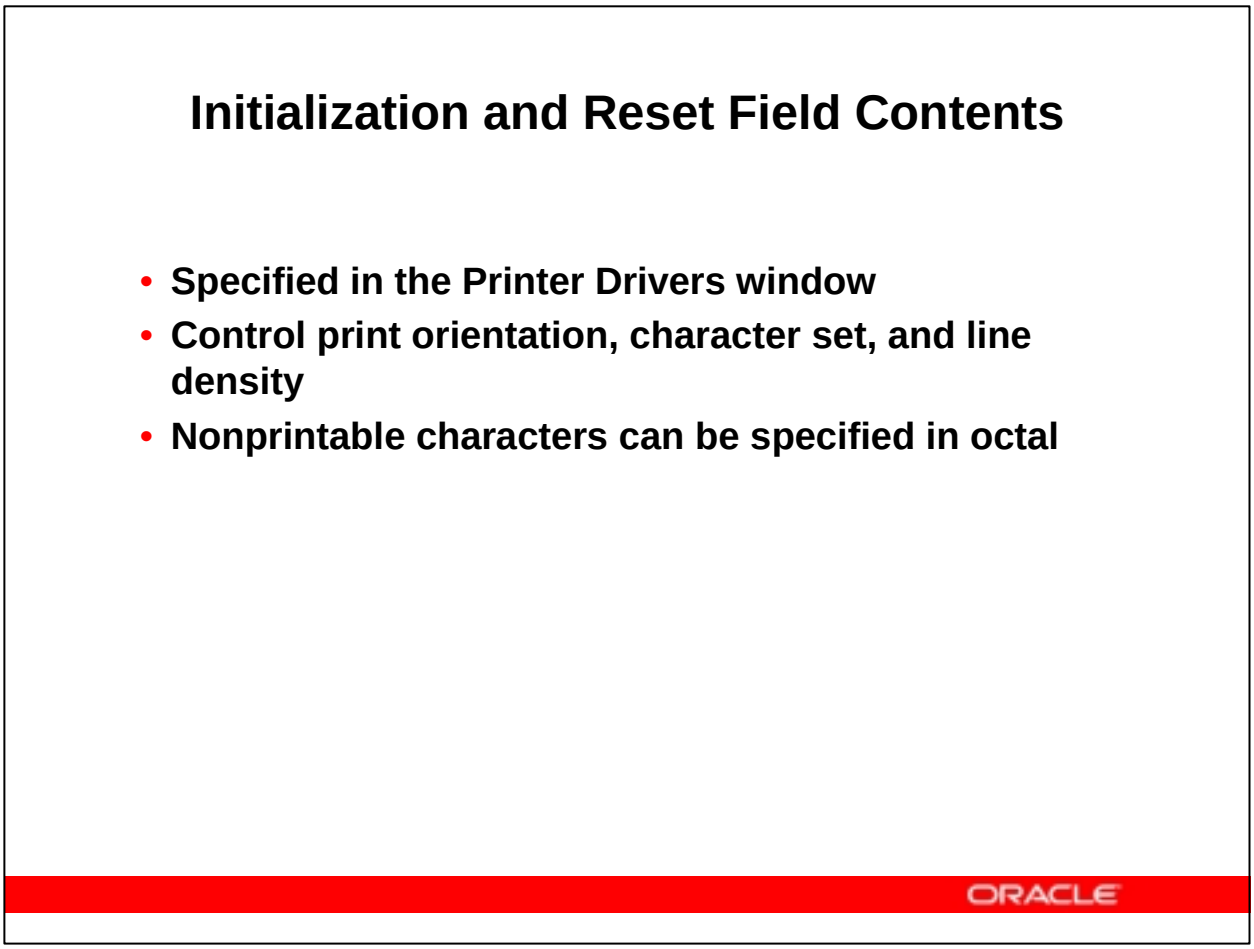
Spool File Check Box Selected

- Spool filing specifications are set on the Printer Drivers window in the Define Method Parameters region. (N) Install > Printer > Driver.
- The Spool File check box is selected only if the print program creates its own temp file. Selecting the Spool File check box prevents the concurrent manager from creating its own temp file.
- When you select the Spool File check box, it is recommended that Initialization and Reset fields are null and the Standard Input check box is cleared.

Spool File Check Box Cleared

- When the Spool File check box is clear, the concurrent manager makes a copy of the file to be printed and sends the copy to the spooling area of the printer.

- Clear the Spool File check box when an operating system print command or a print program does not make a copy of each file to be printed.



When to Use Initialization and Reset Strings

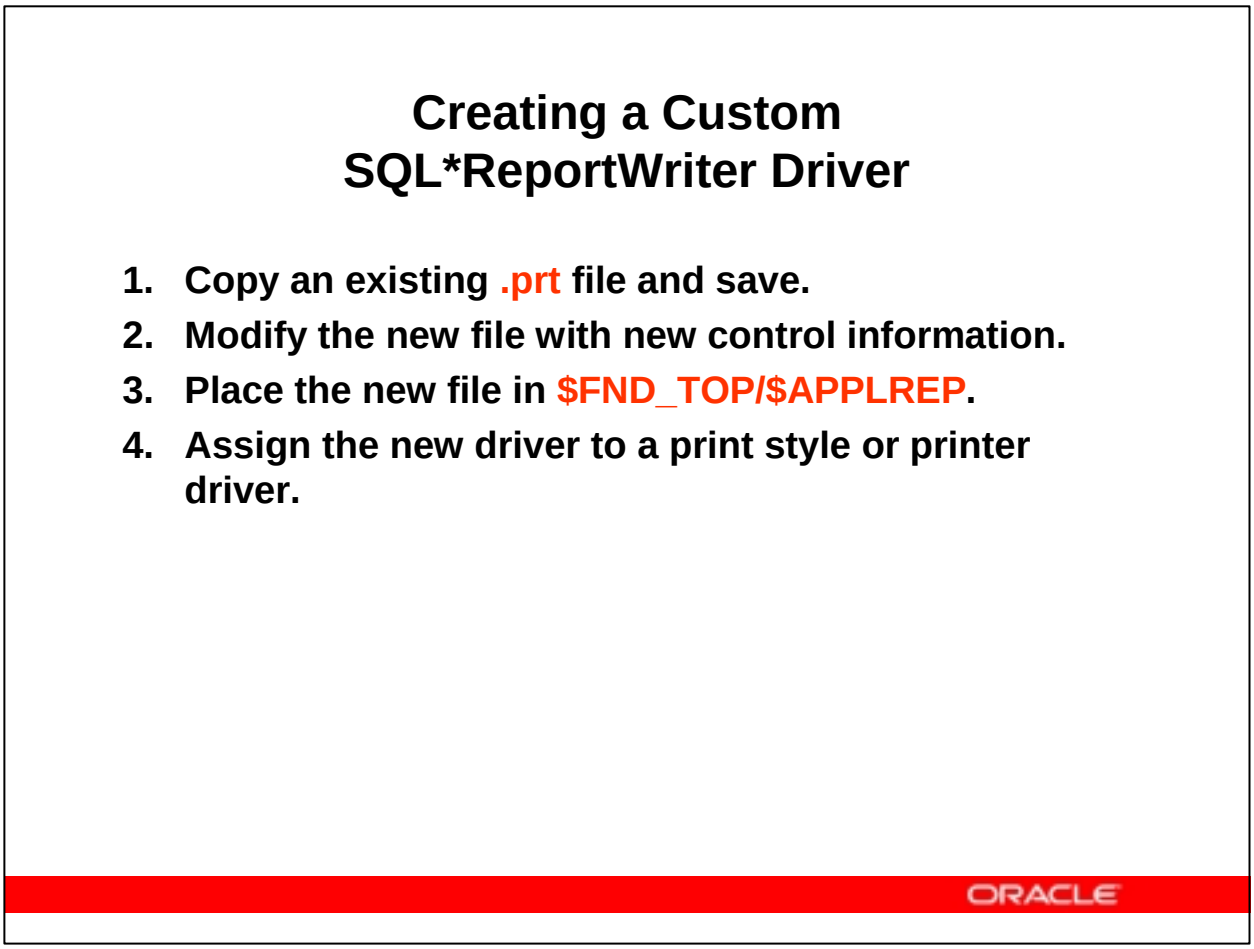
Edit Initialization and Reset strings when a printer type requires different control characters, escape sequences, or instructions.

Initialization and Reset Strings

- Use the Initialization and Reset fields to set and reset the orientation, character set, and line density for your printer.
- Initialization and Reset strings consist of control characters and escape sequences.
- A control character can be represented by a caret (^) followed by another character.
- An escape sequence can be represented by a slash (/).

Example: Escape Sequence

For nonprintable characters, you can represent their value in octal mode. For example, represent 0x26 as /046.



Creating a Custom SQL*ReportWriter Driver

Customize Oracle Reports SRW drivers if a printer type does not properly interpret the control characters that set page breaks, or bold or underlined type in applications reports.

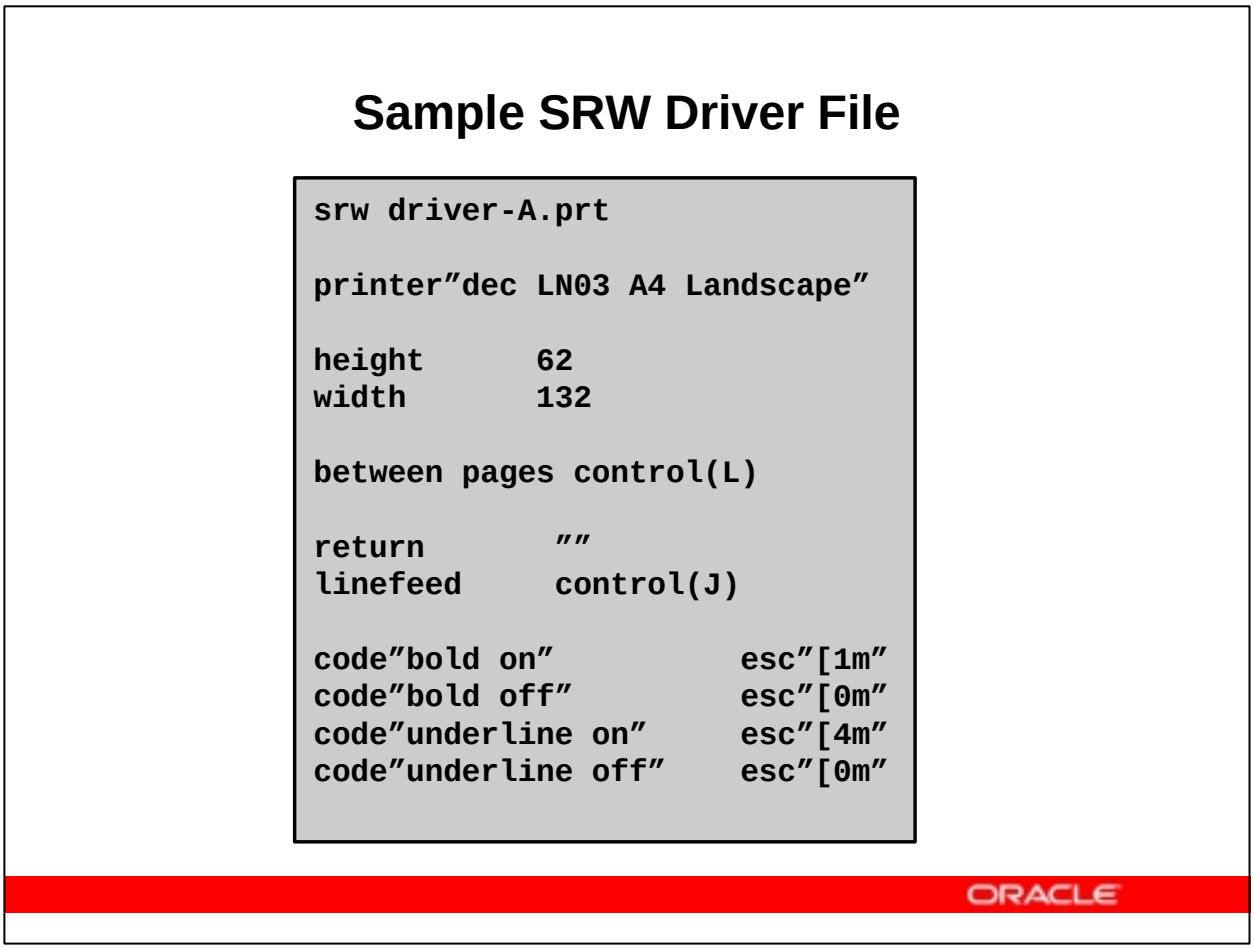
SRW Drivers

- SRW drivers are read by Oracle Reports when a report is generated and insert control characters that tell the destination printer where to set page breaks and which characters to format as bold or underlined.
- SRW drivers are designed for the DEC LN03 printer and all printers that understand the same control characters as the LN03.
- An SRW driver is used during the generation of a report. A printer driver is used when the completed output file is sent to the printer.
- SRW driver files reside in $FND_TOP/$APPLREP and have the extension .prt.

Creating a New SRW Driver

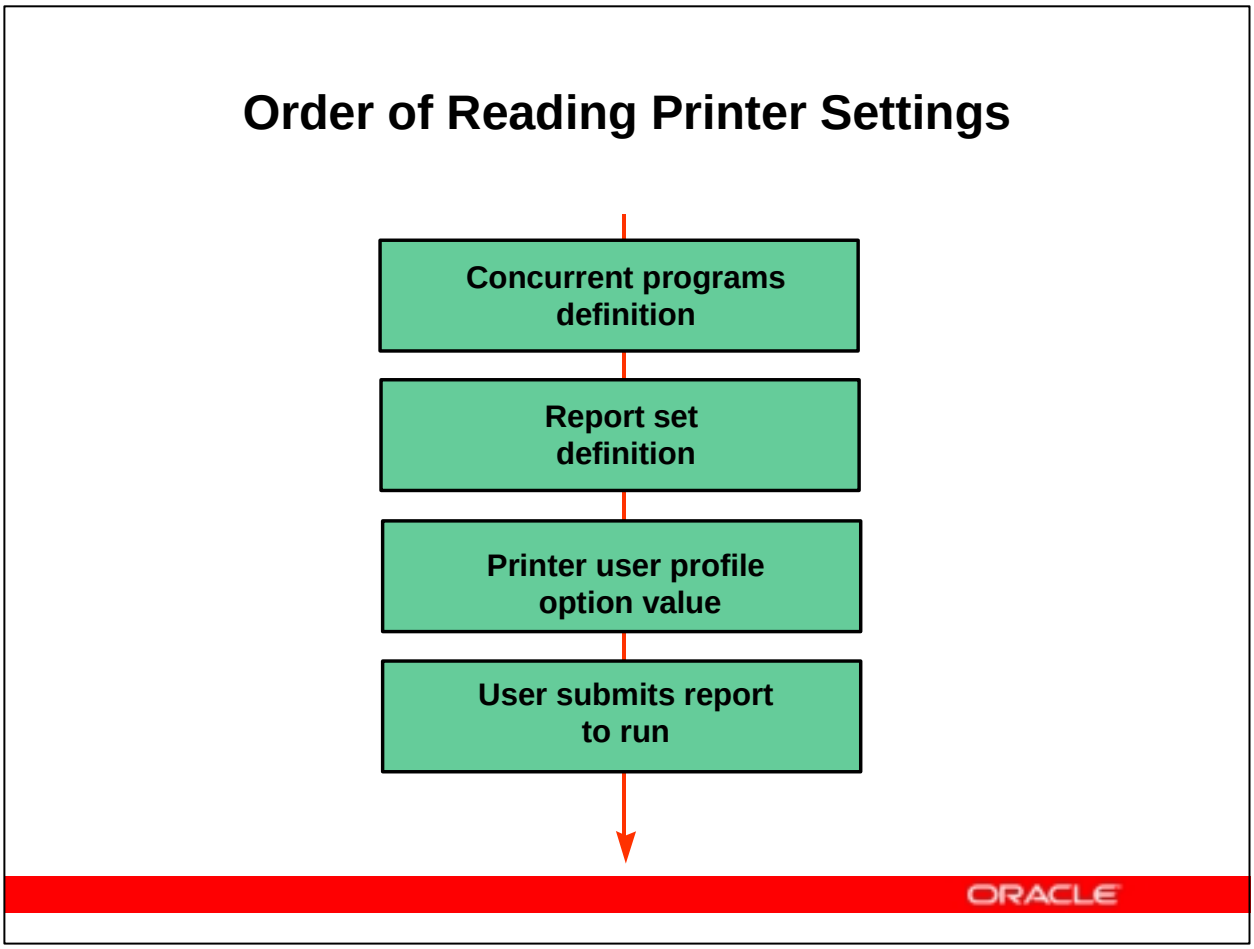
1. Copy an existing SRW Driver (.prt file) and rename the copy before starting any text editing.
2. Modify the new file with new control characters.

3. Place the modified copy of the SRW driver file in $FND_TOP/$APPLREP. Oracle Reports will use the new driver if it is associated with a print style and/or printer driver definition.



Sample SRW Driver File

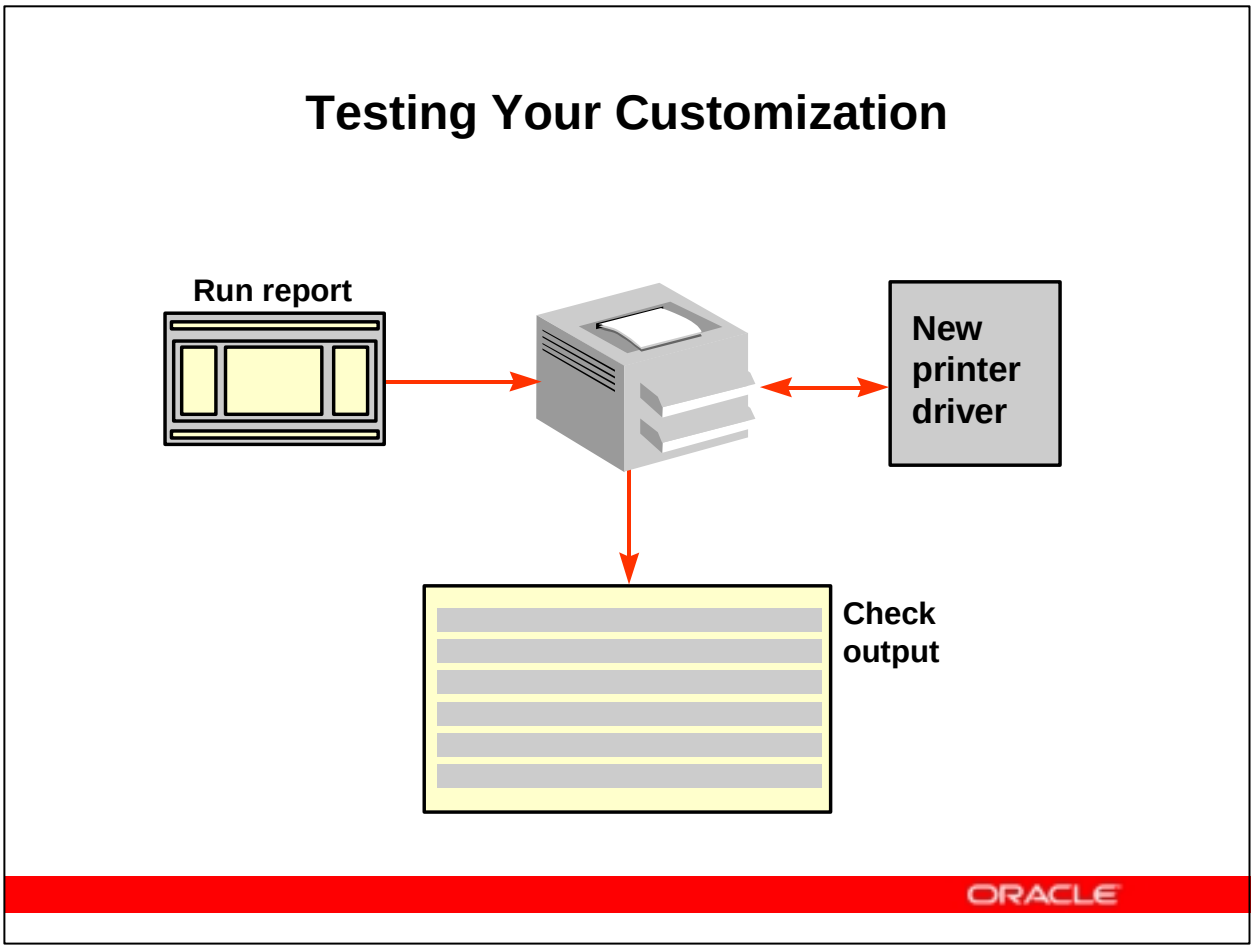
The slide shows the definition for the A.prt driver file and shows the information contained in a typical SRW file.



Hierarchy of Printer Assignments

You can restrict concurrent programs and reports to send their output to a specific printer, or you can provide a default printer.

- At the concurrent program level the system administrator can define a concurrent program to always direct its output to only one printer. This setting cannot be overridden by users.
- At the request set level, the system administrator can assign a default printer for each report in the request set or for the entire request set. If the request set is unowned, no one can override this setting.
- Also at the request set level, an application user can assign a default printer to a report within the set or to the entire set. This value can be overridden by the administrator.
- At the personal profile level, an application user can assign a default printer for all of his or her reports. This value overrides any other values.
- At the site level, the system administrator can assign a default printer for all reports. This can be overridden by users.



Testing Your Customization

You can verify printer driver definitions by printing different styles from each printer type you are using.

Initialization String

- Print a short report to verify that the page's printing orientation is correct. Edit page orientation instructions in the Initialization string.
- If you want to change the printer's default font for the report, include that information in the Initialization string.

Reset String

- Print two short reports with different printing orientations to verify that the printer is resetting itself properly.
- For example, print one report that is landscape and another that is portrait. Edit printer reset instructions in the Reset string.

Arguments

- Print a short report to verify that the arguments to the operating system's print command or custom print program are being interpreted correctly.

- Check that the correct file is being printed at the destination printer and check the number of copies that are printed. Edit arguments in the Arguments field.

Practice - Printers

Overview

The implementation committee wants you to test printing by completing the following.

- Define a printer style
- Define a printer driver
- Define a printer type
- Register the printer

Tasks

Define your Printer Style

1. Create a new Printer Style. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Style Name**: *Your Initial* Portrait
   - **Sequence Number**: *your terminal number* + 300
   - **User Style**: *Your Initials* Portrait
   - **SRW Driver**: P-*Your Initials*
   - **Description**: *Your Initial* Portrait
   - **Layout Columns**: 80
   - **Rows**: 66
   - **Suppress Header**: enabled
   - **Orientation**: Portrait

Define your Printer Driver

2. Use the information below to define a driver to be associated with your new print style. This driver specifies the initialization and reset strings as well as other system-related information
   - **Driver Name**: *Your Initials* Printer Driver
   - **User Driver**: *Your Initials* - Portrait

- **Description**: *Your Initials* Printer Driver
- **SRW Driver**: P–*Your Initials*
- **Platform**: Windows - NT
- **Driver Method**: Command
- **Arguments**: lp -c -d$PROFILES$.PRINTER -n$PROFILES$.CONC_COPIES
- **Initialization**: /e[!p
- **Reset**: /e[!p

Define your Printer Type

3. Your printer type definition specifies the allowable printer styles and drivers for your printer. Use the information below and define all the printer styles and drivers listed

- **Type**: *Your Initials* - HPLJ5
- **Description**: *Your Initials* - HPLJ5

Style:	**Driver Name:**
Your Initials Portrait	*Your Initials* Printer Driver
A4	A4PRINTCMDLQ1070
DYNAMIC	PORTRAITCMDLQ1070
LANDSCAPE	LANDSCAPECMDLQ1070
PORTRAIT	PORTRAITCMDLQ1070

Register your Printers

4. The last step in setting up your printers is to register your new printer with Oracle Applications and associate a printer type with it.

- Printer: *Your Initials*Custom
- Type: *Your Initials* - HPLJ5
- Description: Printer for *Your Initials*'s office

Solution - Printers

Create your Printer Style

Responsibility = System Administrator

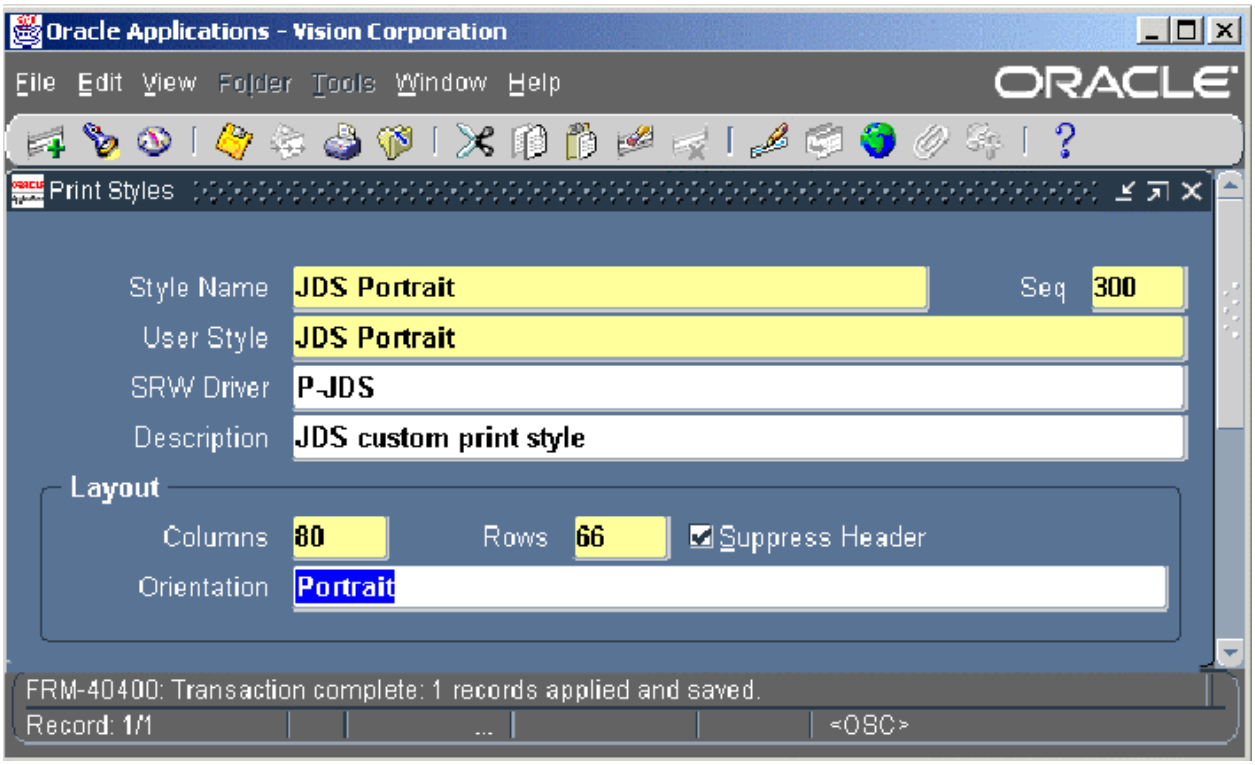
1. If necessary, log in to Oracle Applications and select the System Administrator responsibility.

2. Navigate to (N) Install > Printer > Style.

3. Create a new Printer Style.

   - **Style Name**: *Your Initial* Portrait
   - **Sequence Number**: *your terminal number* + 300
   - **User Style**: *Your Initials* Portrait
   - **SRW Driver**: P-*Your Initials*
   - **Description**: *Your Initial* Portrait
   - **Layout Columns**: 80
   - **Rows**: 66
   - **Suppress Header**: enabled
   - **Orientation**: Portrait

4. Save your work. Your form should look similar to the following.



5. Close the form

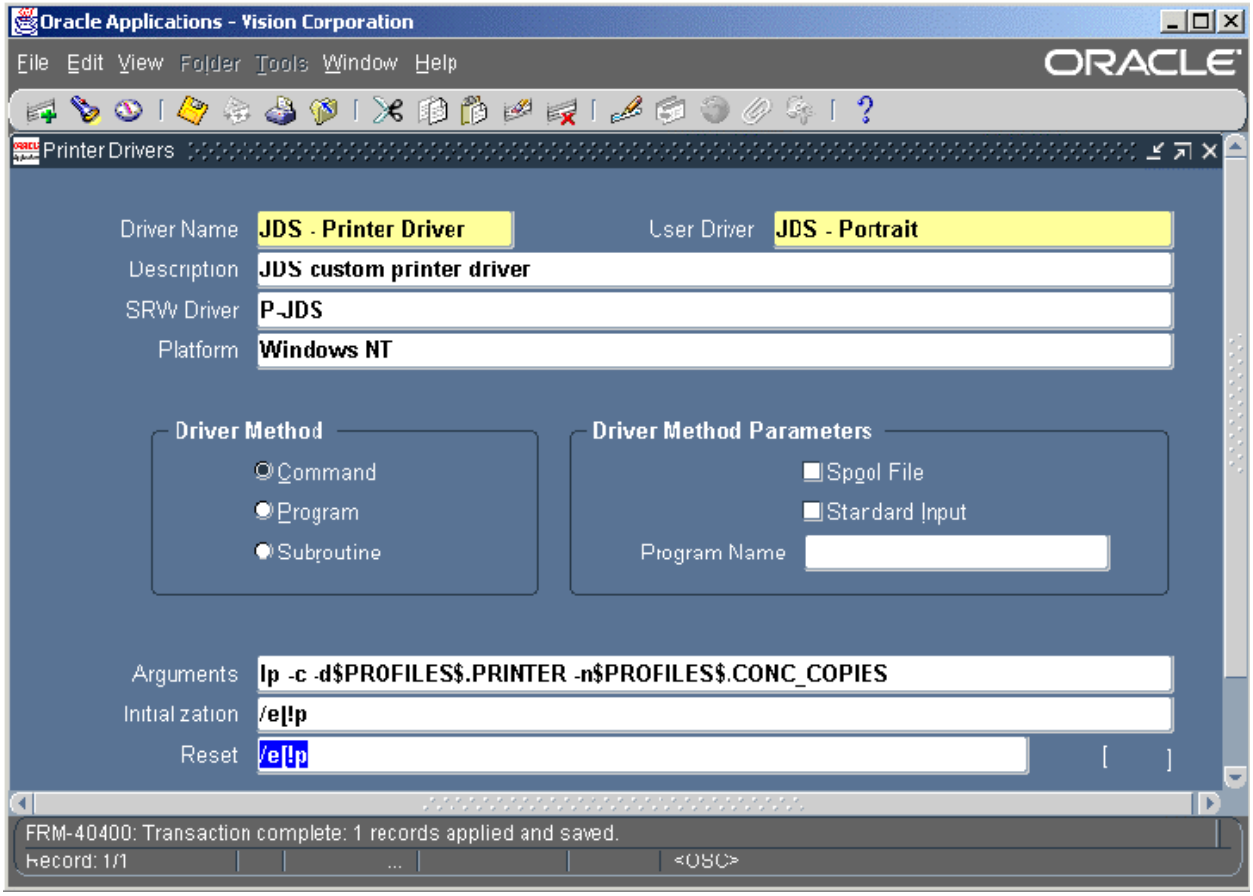
Define your Printer Driver

6. Navigate to (N) Install > Printer > Driver.

7. Define your printer driver.

   - **Driver Name**: *Your Initials* Printer Driver
   - **User Driver**: *Your Initials* - Portrait
   - **Description**: *Your Initials* Printer Driver
   - **SRW Driver**: P–*Your Initials*
   - **Platform**: Windows - NT
   - **Driver Method**: Command
   - **Arguments**: lp -c -d$PROFILES$.PRINTER -n$PROFILES$.CONC_COPIES
   - **Initialization**: /e[!p
   - **Reset**: /e[!p

8. Save your work. Your form should look similar to the following.



9. Close the form.

Define your Printer Type

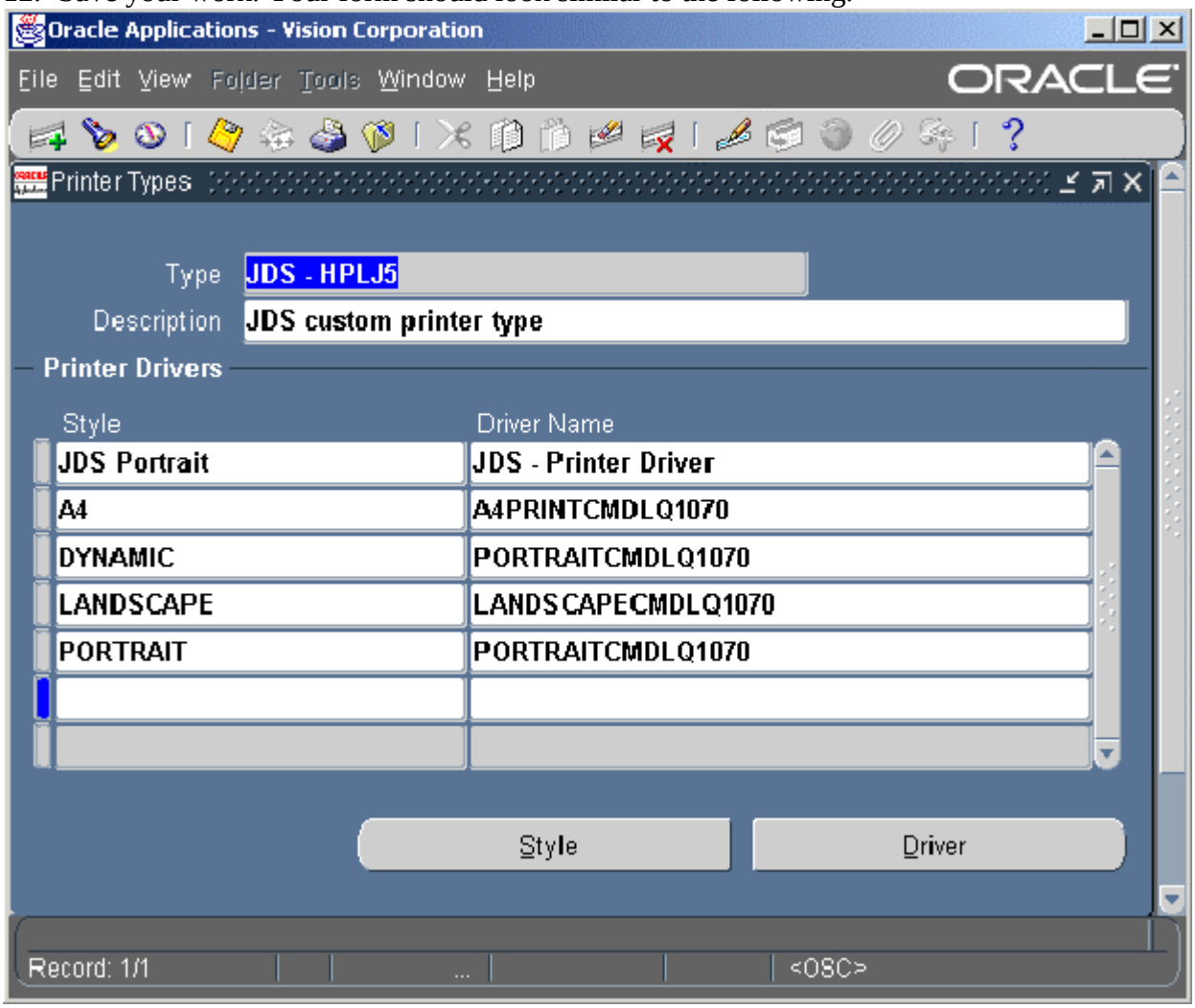
10. Navigate to (N) Install > Printer > Type.

11. Use the information below and define all the printer styles and drivers listed

- **Type**: *Your Initials* - HPLJ5
- **Description**: *Your Initials* - HPLJ5

Style:	**Driver Name:**
Your Initials Portrait	*Your Initials* Printer Driver
A4	A4PRINTCMDLQ1070
DYNAMIC	PORTRAITCMDLQ1070
LANDSCAPE	LANDSCAPECMDLQ1070
PORTRAIT	PORTRAITCMDLQ1070

12. Save your work. Your form should look similar to the following.



13. Close the form.

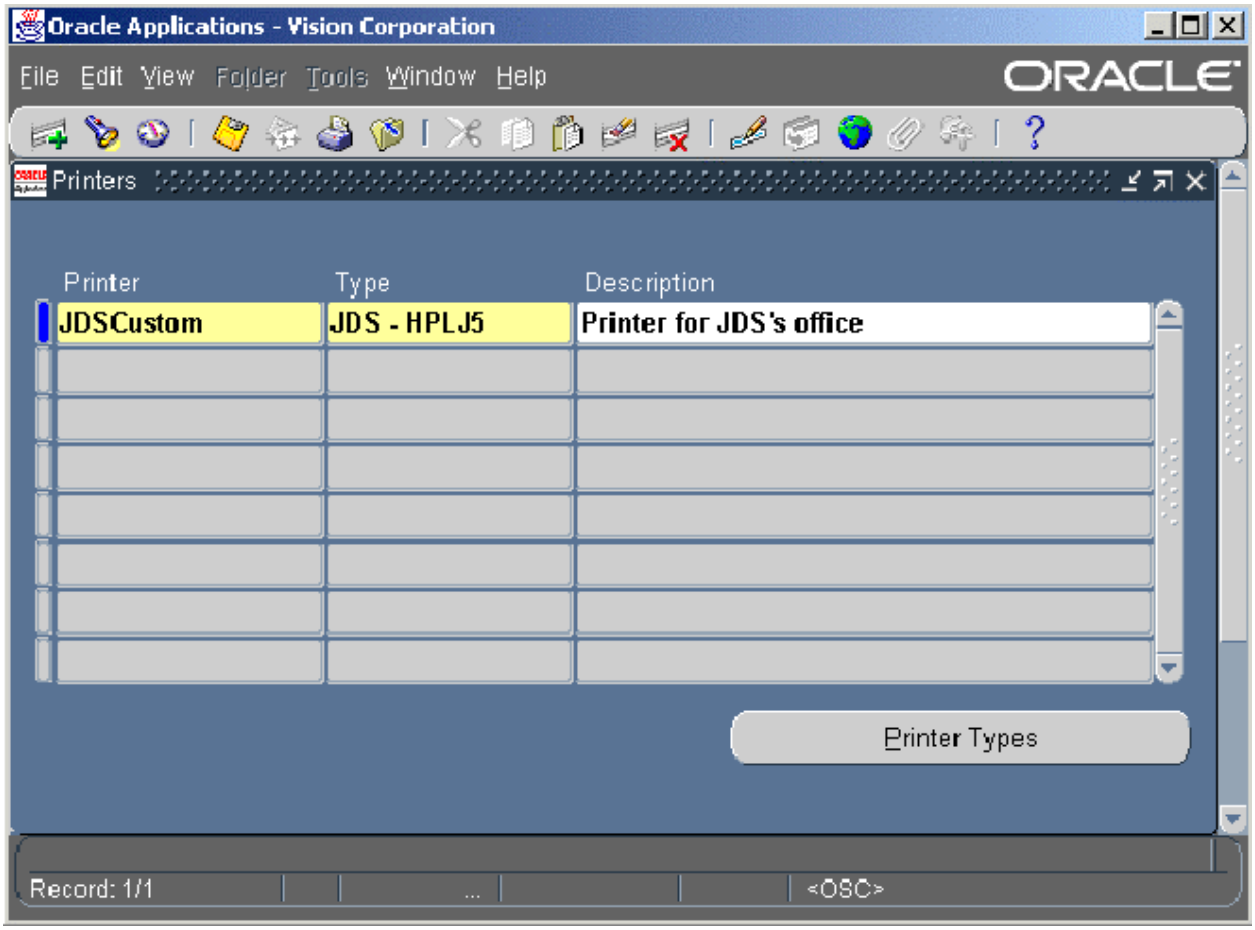
Register your Printers

14. Navigate to (N) Install > Printer > Register

15. The last step in setting up your printers is to register your new printer with Oracle Applications and associate a printer type with it.

    - Printer: *Your Initials*Custom
    - Type: *Your Initials* - HPLJ5
    - Description: Printer for *Your Initials*'s office

16. Save your work. Your form should look similar to the following.



17. Close the form.
18. Navigate to (N) Requests > Run.
19. Click the “OK” button to submit a single request.
20. Select “Active Users” as the report to run.
21. Click the “Options…” button to open the “Upon Completion…” window.
22. Click the list of values icon for the Printer field to verity that your printer is available.
23. Close this form and cancel the request.

Summary

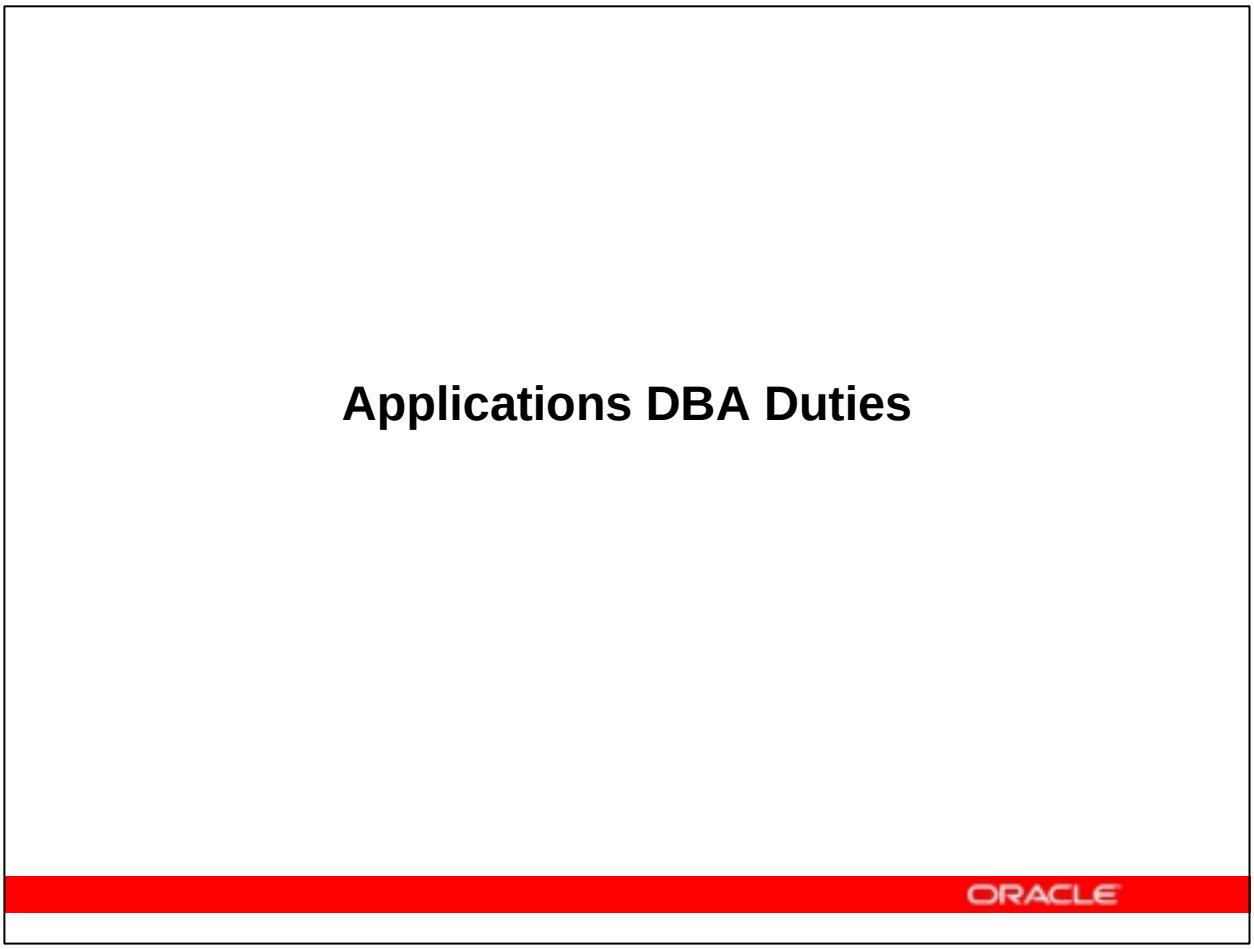
You should now be able to do the following:

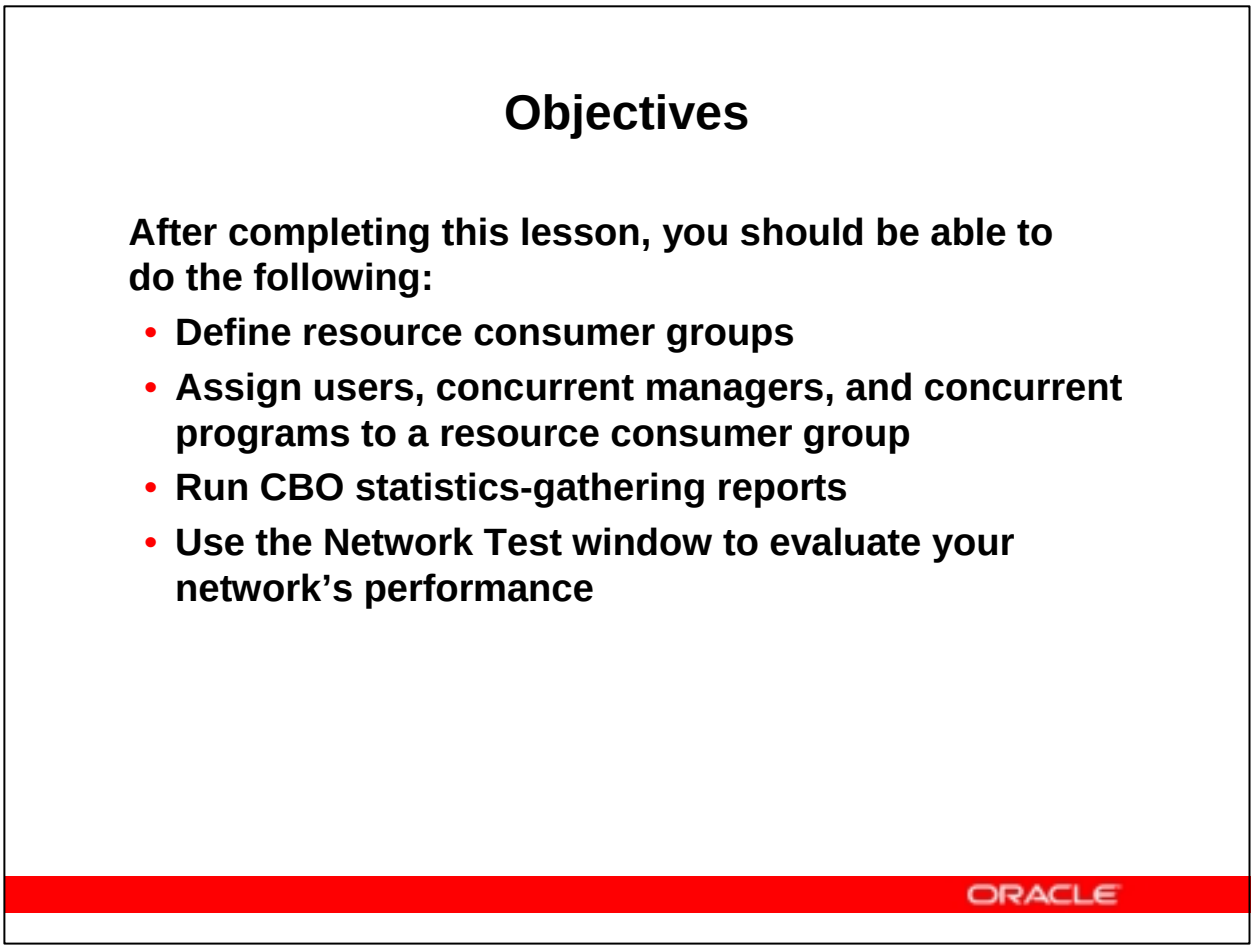
- **Register new printers**
- **Define new printer types**
- **Create a custom print style**
- **Create a custom printer driver**
- **Assign a print style and printer driver to a printer type**
- **Create a custom SRW driver for formatting text and page breaks**

ORACLE

Applications DBA Duties

Chapter 9

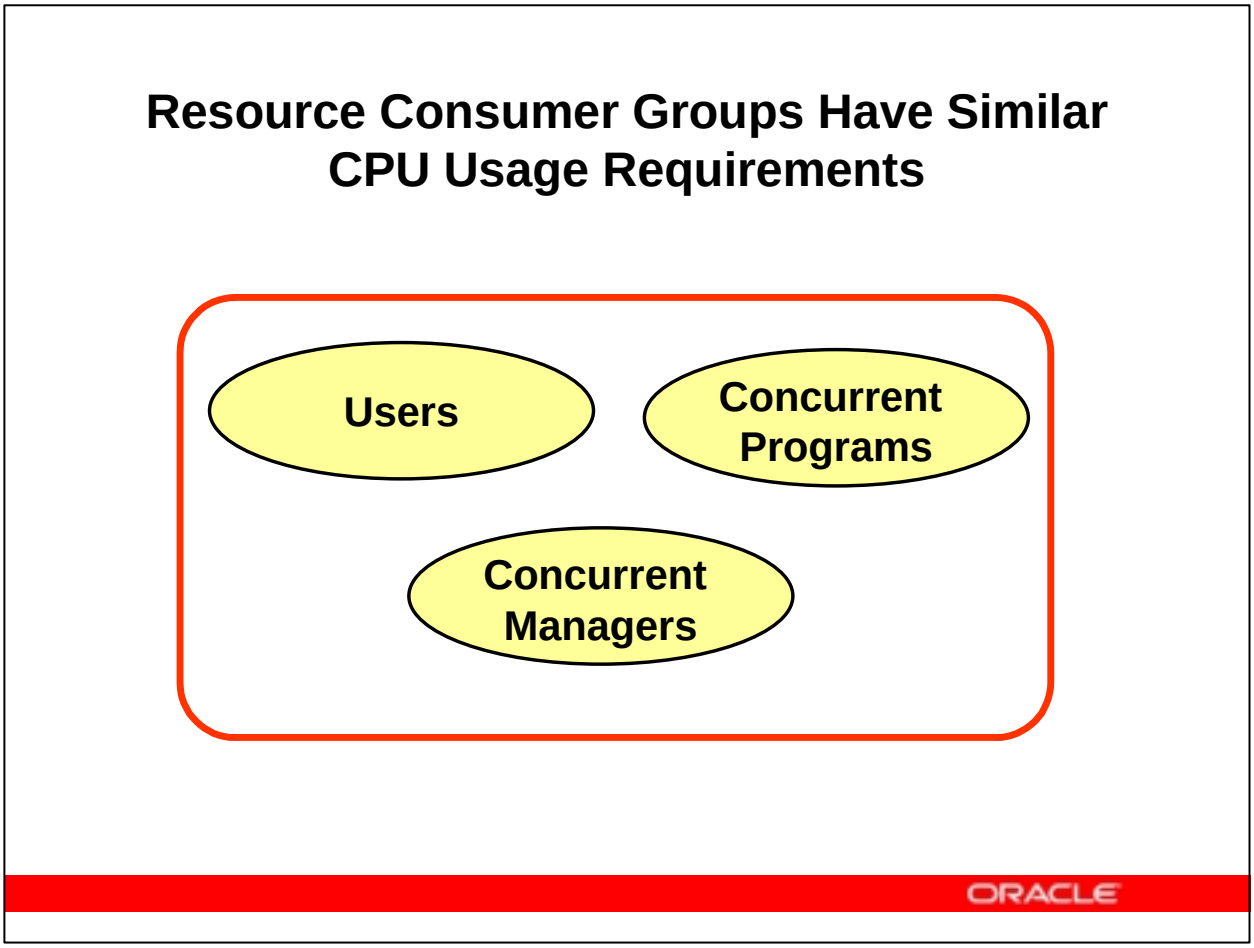




Lesson Topics

The administration of Oracle Applications includes some duties that can fall to either the Database Administrator or the System Administrator to perform. This lesson describes several security tasks from different functional areas, which encompass forms from various locations on the System Administrator menu tree. These tasks are referred to as Applications DBA duties.

Resource Consumer Groups Have Similar CPU Usage Requirements

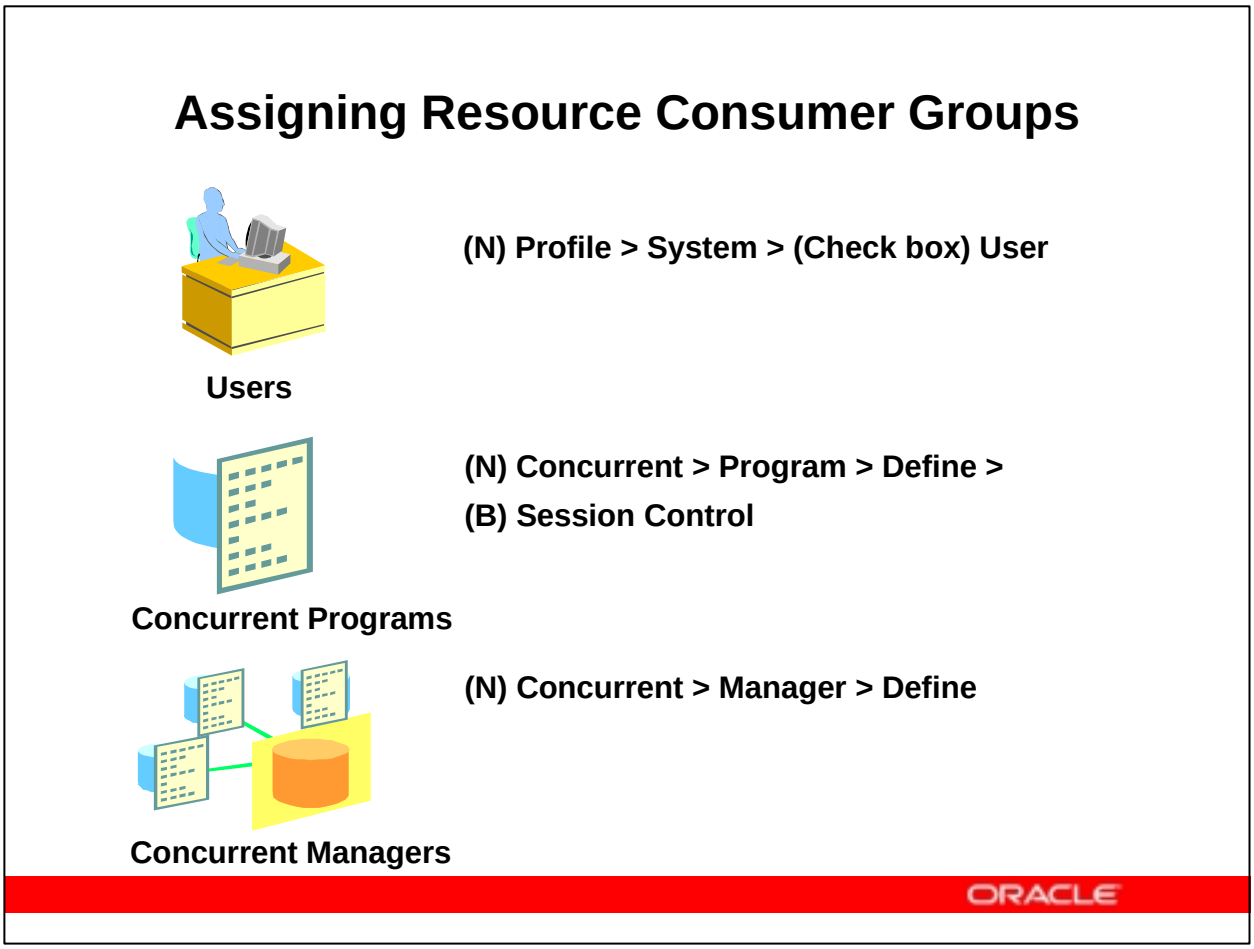


Resource Consumer Groups: Definition

Resource consumer groups and resource plans are tools used to allocate and manage resources among database users and applications. The database administrator establishes resource consumer groups to define users, concurrent managers, or concurrent programs that have similar CPU usage requirements. An overall resource plan is used to specify how resources are distributed among the different resource consumer groups.

System Administrator Assigns Resource Consumer Groups

Oracle Applications allows the system administrator to assign users, concurrent managers, and concurrent programs to existing resource consumer groups.



Assign a User

(N) Profile >System > (Check box) User

The system administrator can assign a user to a resource consumer group by setting the value of the user profile option FND:Resource Consumer Group for that particular user. The user can see this profile option but cannot update it.

Assign a Concurrent Program

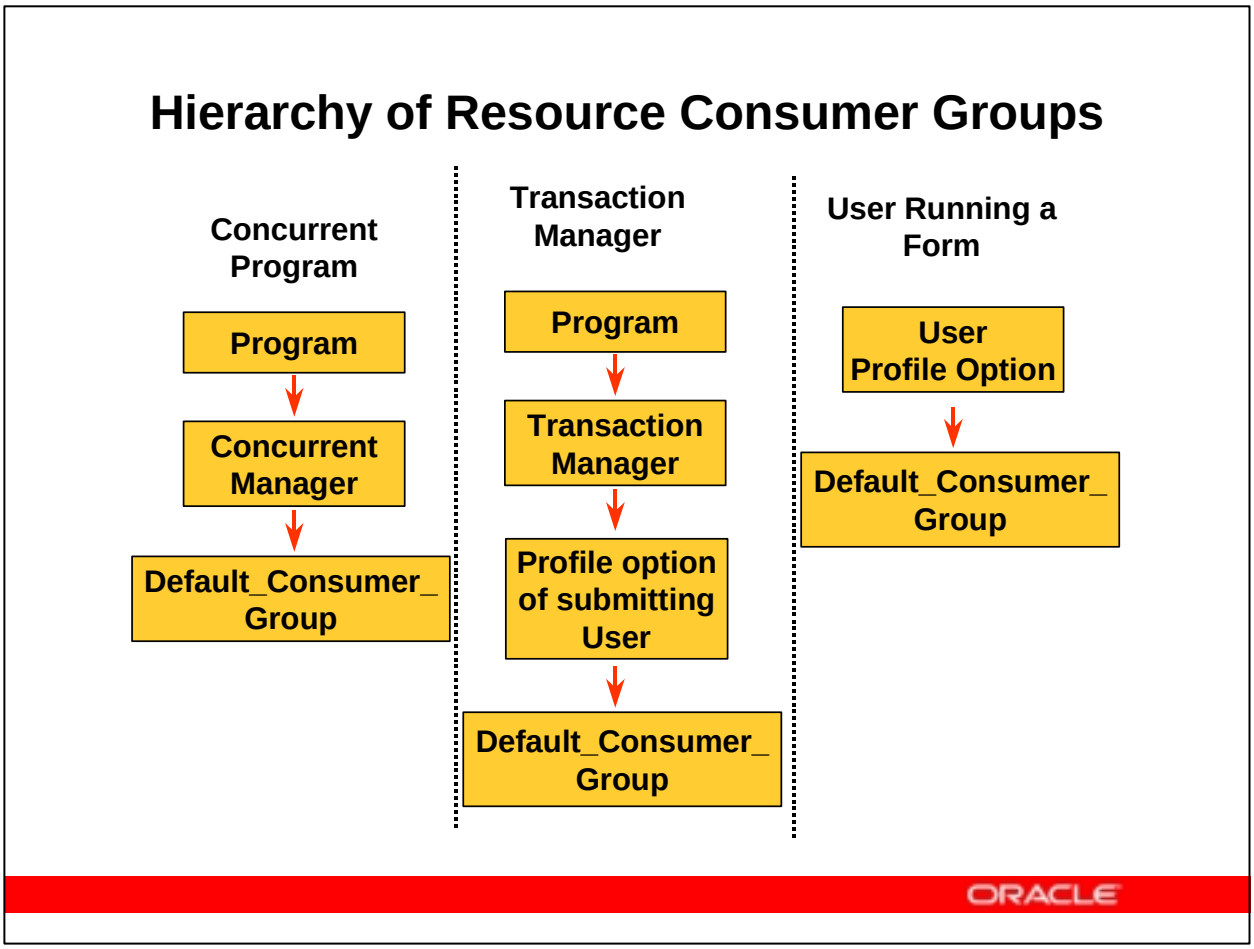
(N) Concurrent > Program > Define > (B) Session Control

The system administrator can assign a concurrent program to a resource consumer group in the Session Control window of the Concurrent Programs form.

Assign a Concurrent Manager

(N) Concurrent > Manager > Define

The system administrator can assign a concurrent manager to a resource consumer group in the Concurrent Managers form.



Hierarchy of Resource Consumer Groups

Conflicts can arise between the resource consumer groups associated with a single session. Oracle Applications uses a hierarchy to resolve these conflicts.

Concurrent Manager Running a Concurrent Program

When a concurrent program is submitted the system first checks for the resource consumer group assigned to the program. If none is assigned, the system will use the group assigned to the manager. If none is assigned to the manager, the system will use the default: Default_Consumer_Group.

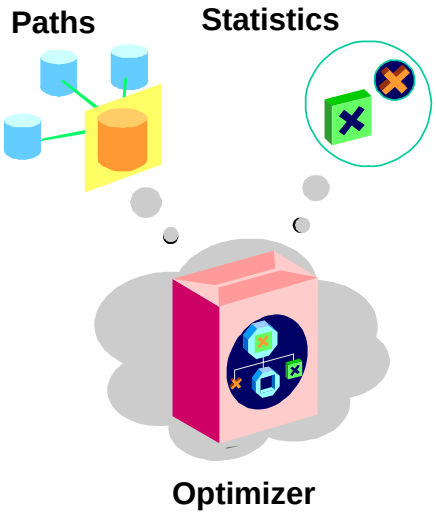
User Performing a Transaction

When a user performs a transaction, the system first checks the group assigned to the transaction. If none is assigned, the system checks the transaction manager. If the manager is not assigned to a group, the system checks the user's profile option. If the user does not belong to a group, then the system uses the Default_Consumer_Group.

User Running a Form

When a user runs a form, the system first checks the user's profile option. If the user is not assigned to a group, the system uses the Default_Consumer_Group.

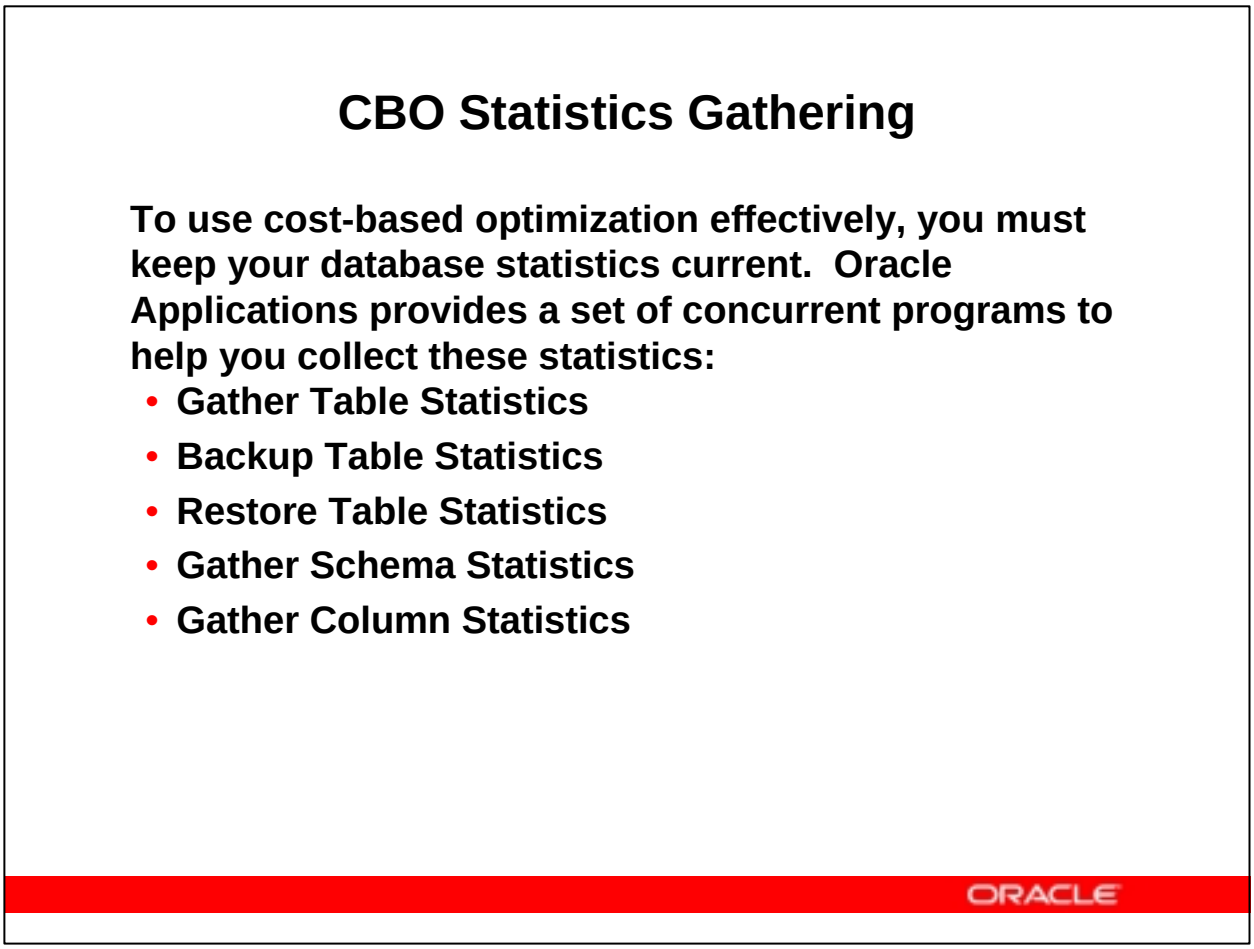
Cost-Based Optimization



Oracle Applications uses cost-based optimization (CBO) to choose the most efficient way to execute SQL statements.

The optimizer determines the most efficient execution plan by considering available access paths and factoring information based on statistics for the schema objects' SQL statement.

ORACLE



CBO Statistics Gathering Reports

Gather Table Statistics – gathers table statistics for a specified table.

Backup Table Statistics – stores the statistics of the given table and backs up related index and column statistics by default.

Restore Table Statistics – restores the previously backed up table statistics from a given statistics identifier. All index and column statistics associated with the specified table are imported as well.

Gather Schema Statistics – gathers specified schema-level statistics. Before gathering the statistics, this program creates a backup of the existing statistics so that if database performance is slowed after gathering new statistics, you can restore the system to its previous status.

Gather Column Statistics – this program is actually two procedures that gather column statistics. One procedure gathers the column statistics for all the columns specified in the SEED data table FND_HISTOGRAM_COLS. The second procedure gathers the column statistics for a specified column_name in a given table.

For further information on these reports see

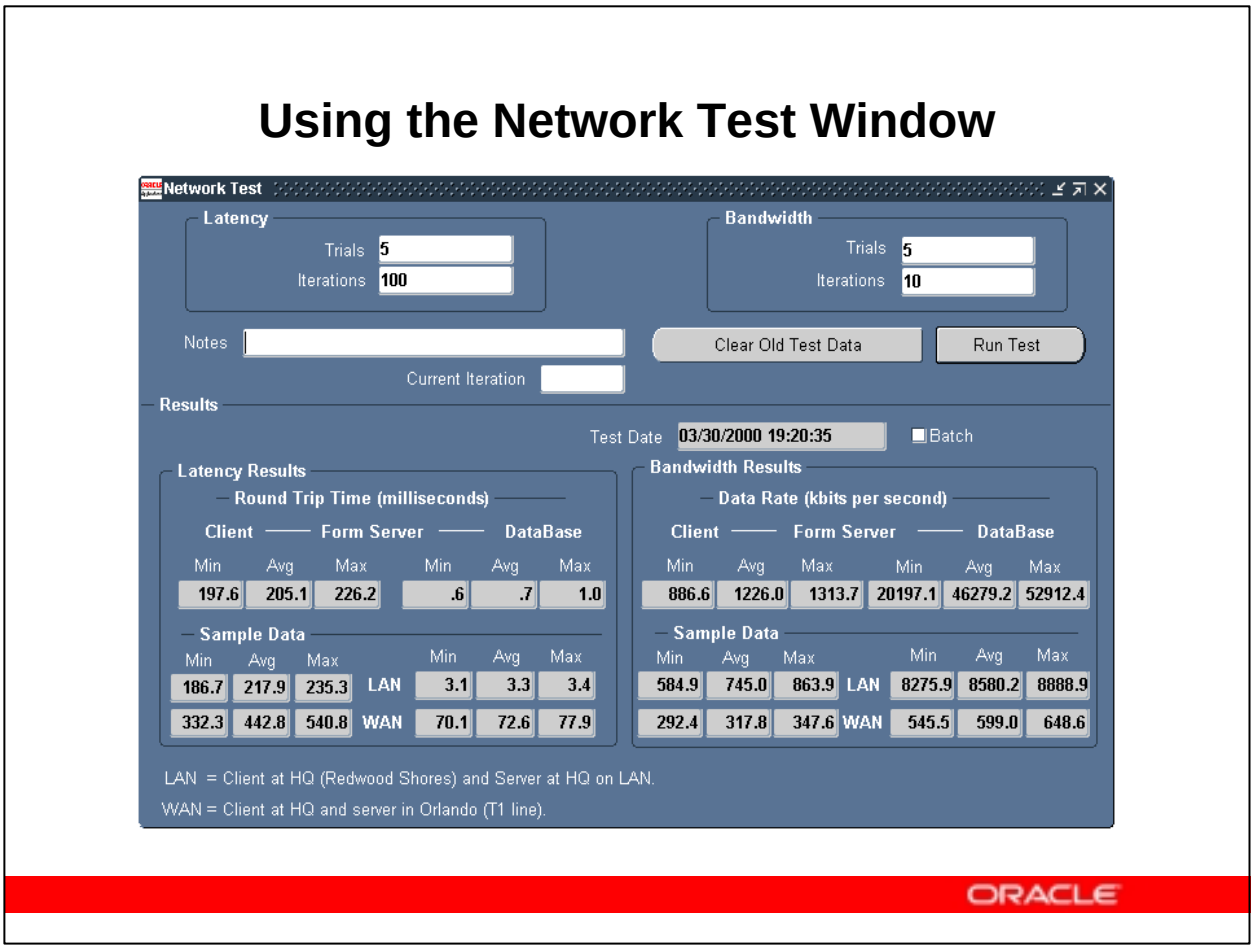
(Help) Applied Technology > Oracle Applications System Administration >

Using the Network Test Window

Use the Network Test window to evaluate the performance of your network with Oracle Applications. The window provides the following tests:

- **Latency Test - examines the time it takes for a single packet to make a round trip from your client side application to the server.**
- **Bandwidth Test - examines the data rate to see how many bytes per second your network can transfer from the server to the client.**

ORACLE



Running Network Tests

(N) Application > Network Test

To Test Latency

- Specify the number of Trials and the Iterations for each trial.
- For each iteration, a single packet is sent from the client application to the server and back. A trial consists of the specified number of iterations. The total time for all round trips in a trial is divided by the number of iterations to obtain their average latency.
- The default settings are 5 trials of 100 iterations each. Select the Run Test button to perform the test.

To Test Bandwidth

- Specify the number of Trials and the iterations for each trial. For each iteration, several kilobytes of data are sent from the client to the server and back. The form measures the average rate at which the data travels.
- The default settings are 5 trials of 10 iterations each.
- Select the Run Test button to perform the test.
- Use the Clear Old Test Data button to purge previous test results from your database.

Guided Demonstration - Consumer Groups

Responsibility: System Administration

Assigning a User to a Resource Consumer Group

1. Navigate to (N) Profile > System.
2. Check the User check box and select your user from the list of values.
3. Select “FND: Resource Consumer Group” from the Profile list of values.
4. Click the “Find” button.
5. The System Profile Values window will display.
6. Select “DEFAULT_CONSUMER_GROUP” from the list of values under the User column.
7. Save your work.

Assign a Concurrent Program to a Resource Consumer Group

1. Navigate to (N) Concurrent > Program > Define.
2. Query your program, *Your Initials*CPCQR. (Note: Any program will work.)
3. Click the “Session Control” button.
4. Use the list of values to select the consumer group to which you wish to assign the program.
5. Save your work.

Summary

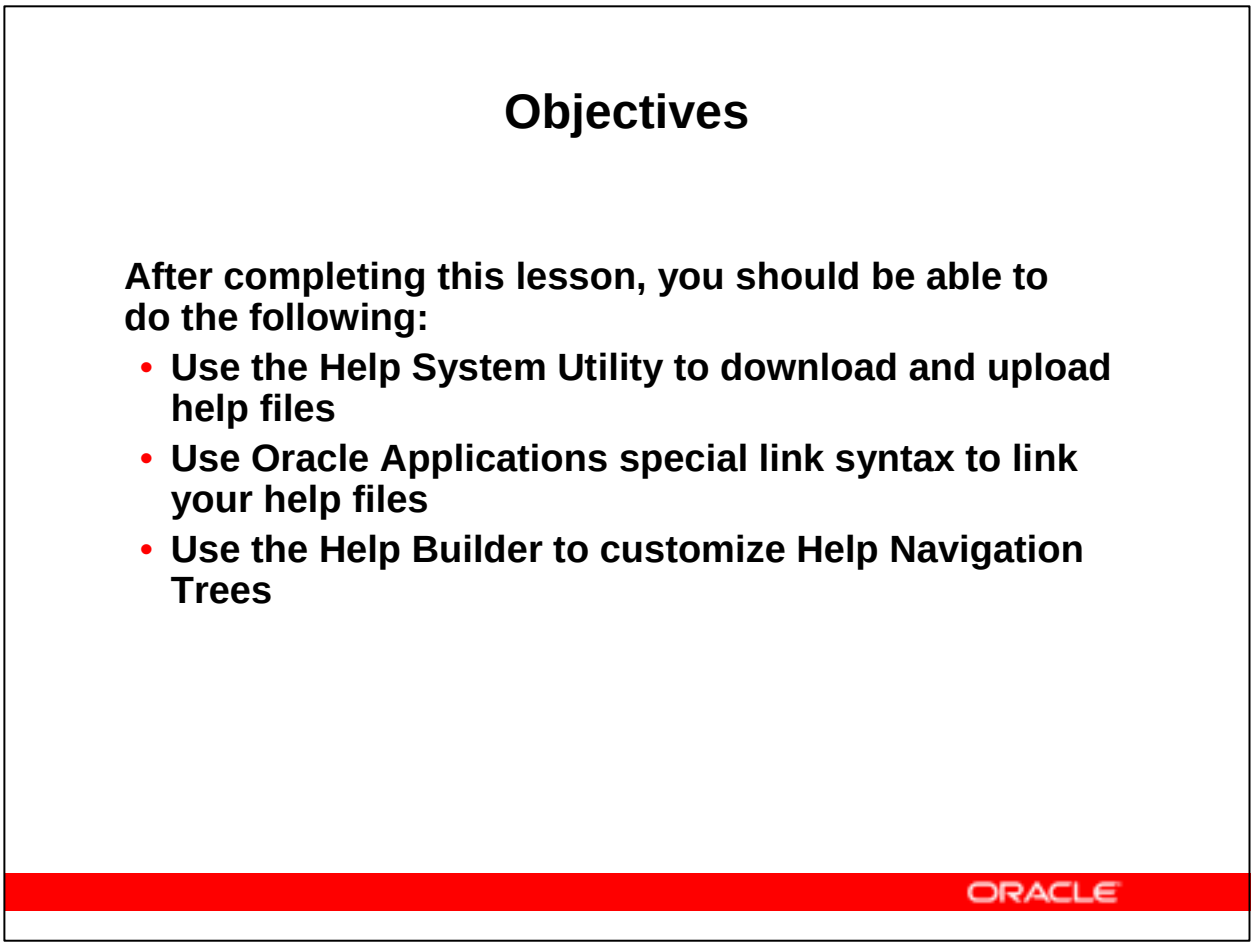
You should now be able to do the following:

- **Define resource consumer groups**
- **Assign users, concurrent managers, and concurrent programs to a resource consumer group**
- **Use the Network Test window to evaluate your network's performance**

ORACLE

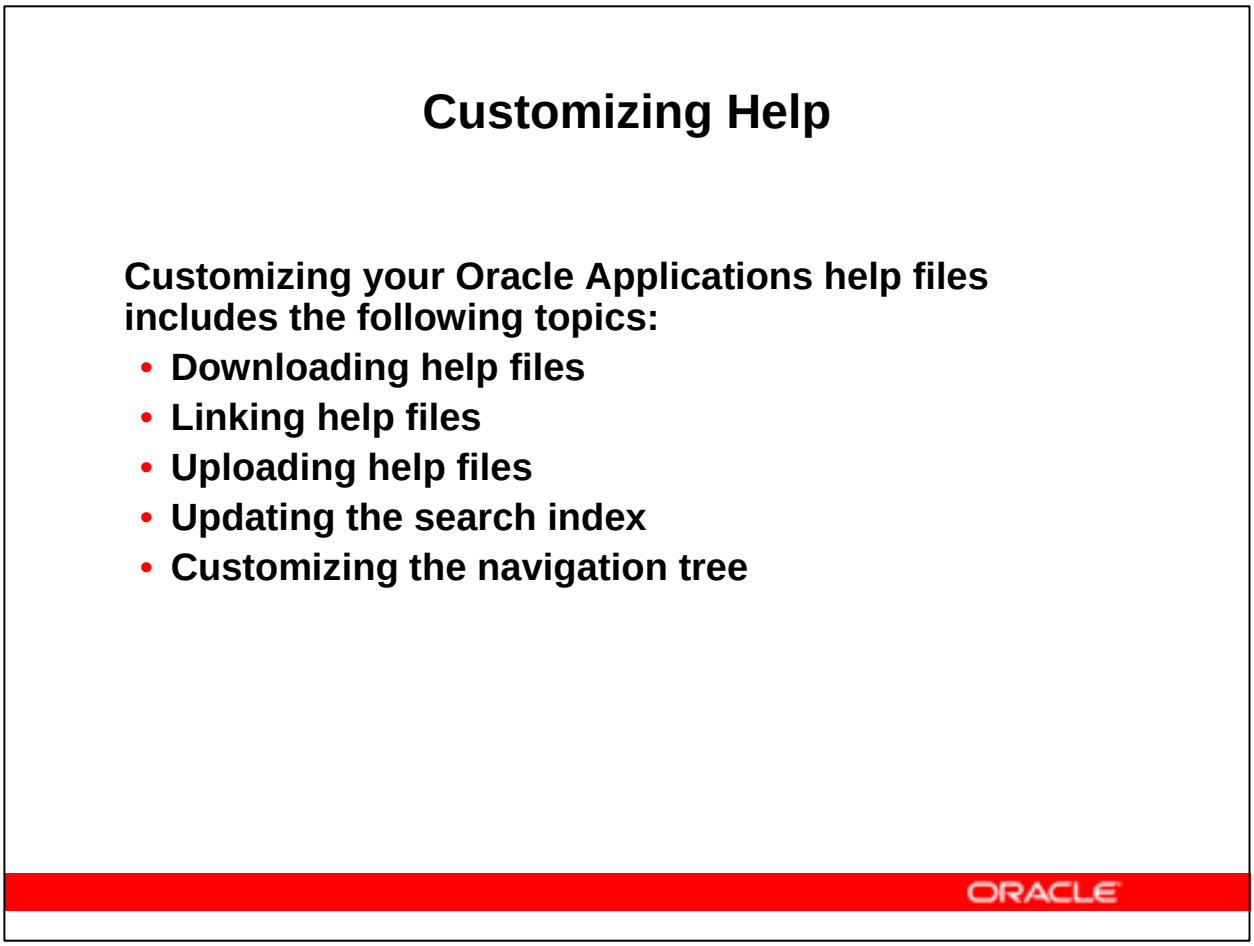
Incorporating Custom Help Files

Chapter 10



Lesson Aim

Oracle Applications uses an extensive network of help files to give users immediate online assistance. You can customize these help files by manipulating the files supplied by Oracle, or by adding your own to the system. This lesson covers how to integrate customized help files into Oracle Applications.



Overview of Customizing Help Files

In order to customize help files you first must download them into an area for manipulation. Oracle Applications provides the Help System Utility to download files. Oracle Applications help files are in HTML format, making them easy to modify using a commercial web browser/editor such as Netscape Navigator. You can also add HTML and GIF files of your own.

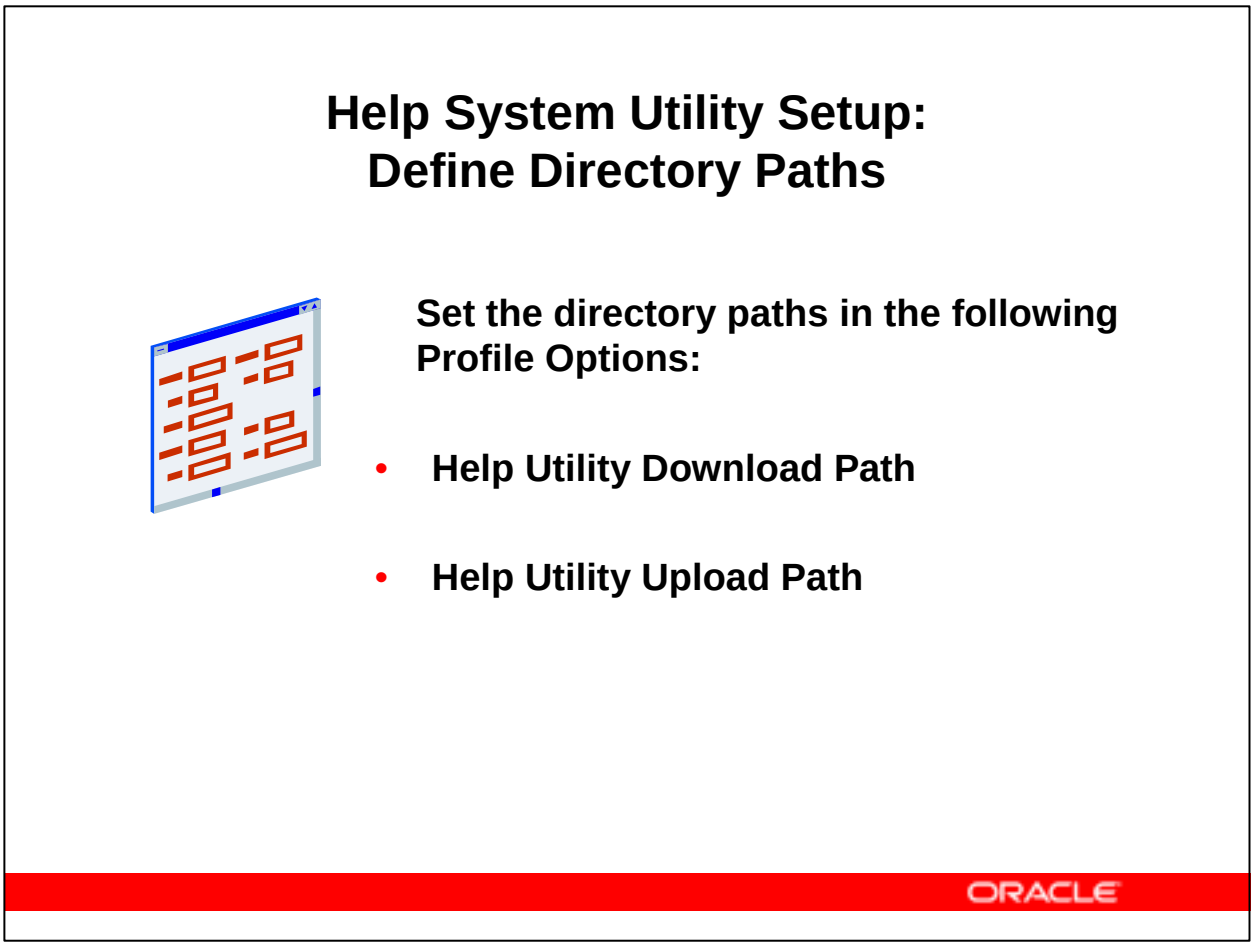
Oracle Applications help files use a special link syntax that points the links to anchors within files rather than to the files themselves. This method ensures that hypertext links will not be broken if files are renamed or split. This lesson explains how to use this syntax, although it is not required for your custom help files.

When your changes are complete, use the Help System Utility to upload the files back to the appropriate Oracle Applications directory.

Rebuild the Search Index and use the Help Builder applet to update the navigation tree with your new files.

Oracle Applications Help System Utility

- **Oracle Applications help files are stored in the database**
- **The Help System Utility is provided for retrieving and replacing them in the course of customization**



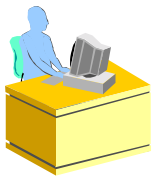
Set Profile Options

Before using the Help System Utility you must define the upload and download directory paths. Oracle Applications provides profile options for you to set these paths.

Use the profile option **Help Utility Download Path** to define the directory location to which the Help System Utility will download files.

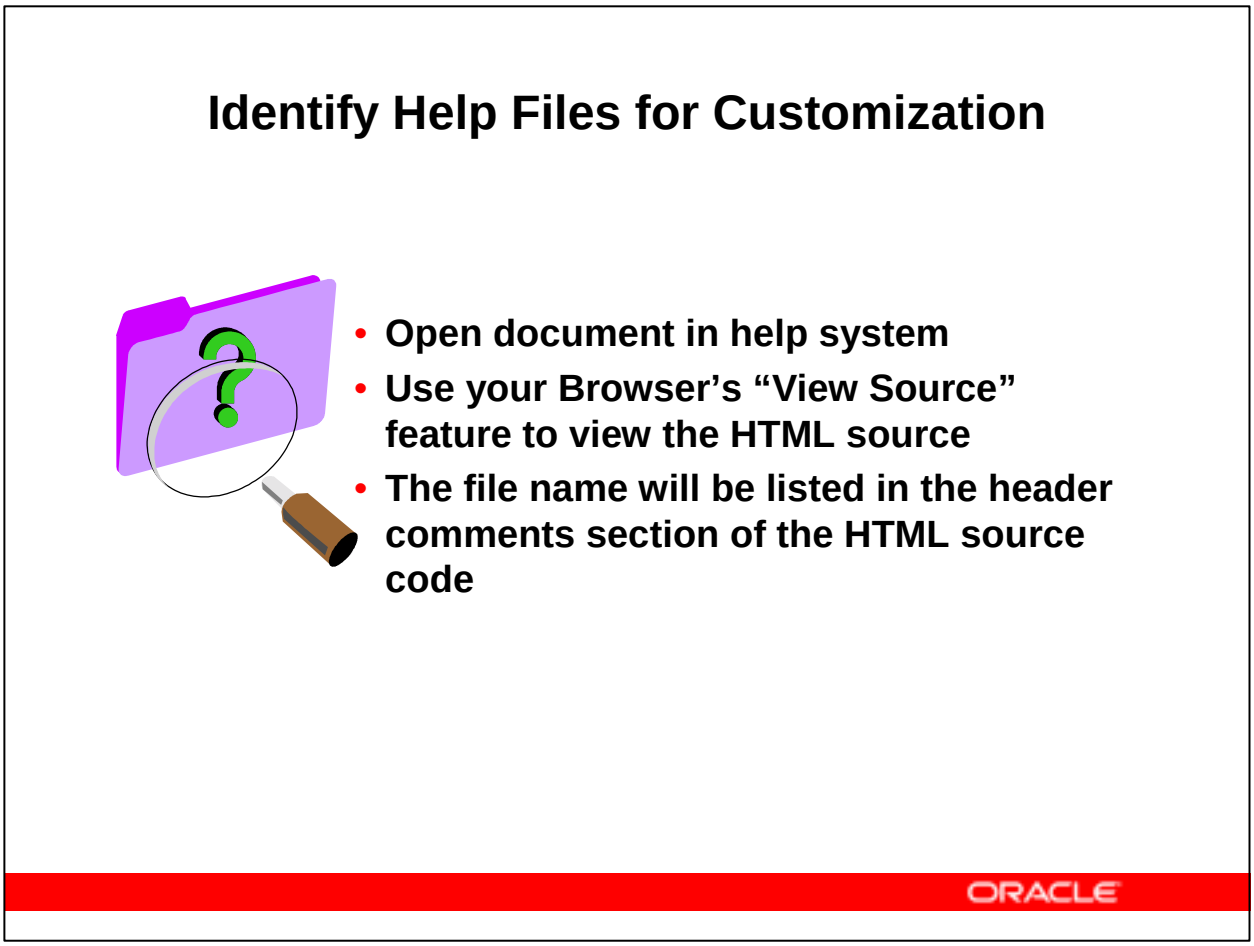
Use the **Help Utility Upload Path** to define the directory location from which your customized files will be transferred back into the Oracle Applications Help System.

Downloading Help Files



1. Identify the help system files you want to customize
2. Identify the language and product of the files
3. Use the Help System Utility to download the files

ORACLE



Identify Help Files

Help files are downloaded by file name. To identify the specific file that you want to customize, open the document in the Oracle Applications Help System. Use the view source function of your browser to view the HTML source code. The source information will include the file name.

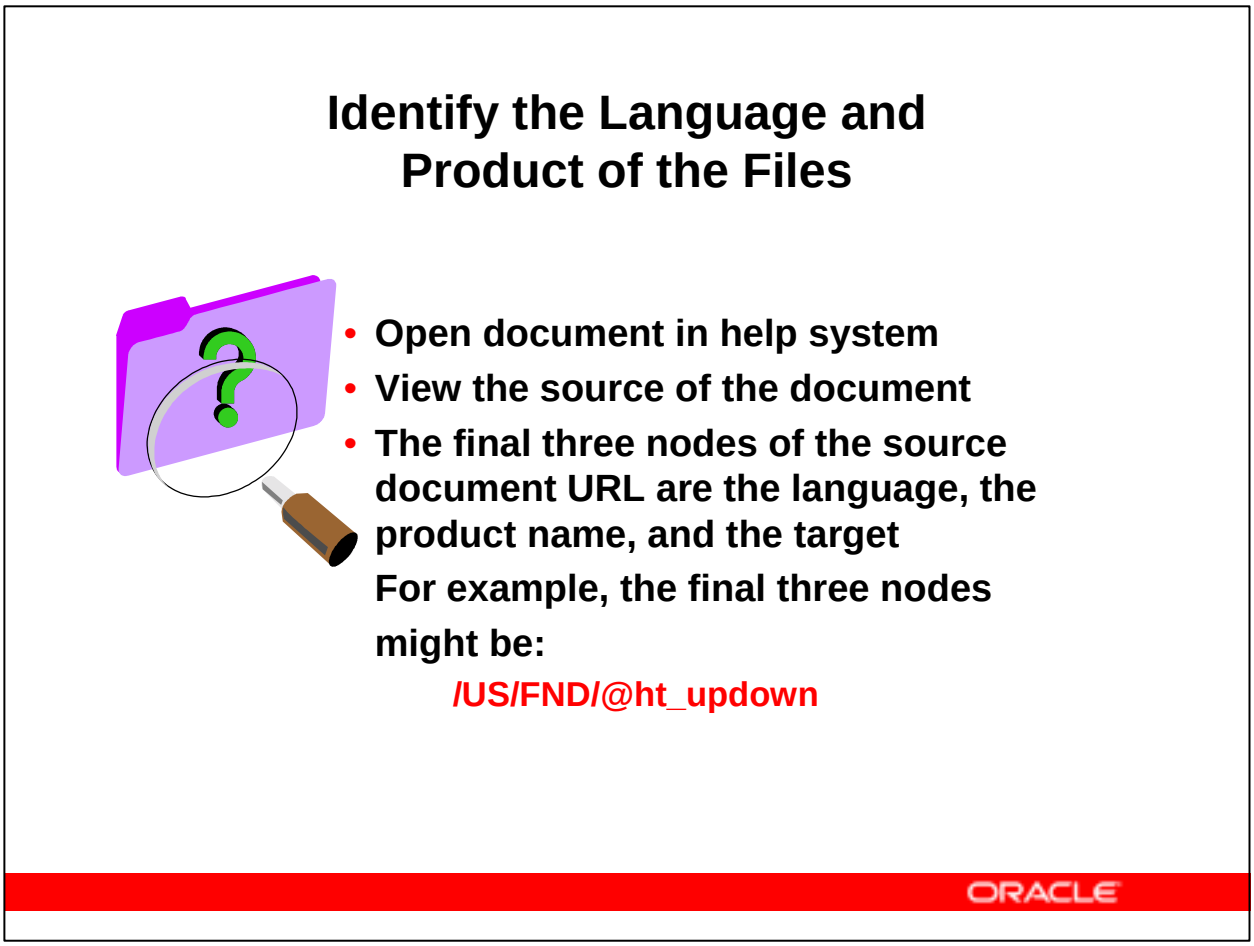
For example, this topic in the help system is found by navigating from the Help Library Tree to Applied Technology > System Administration > Customizing Help > Downloading and Uploading.

View the source of this file and you will see the following header comments:

```
<HTML DIR="LTR">
<!-- Generated: 18/6/01, LeafOnline 4.482
     Source: updown.doc, 1
     File: SYS00032663.htm
     Context: nil
     Tiers: TIER-1-ONLY
     Pretrans: YES
     Label: 115.4   -->
```

```
<HEAD>
```

As you can see in the example, the file name is SYS00032663.htm.

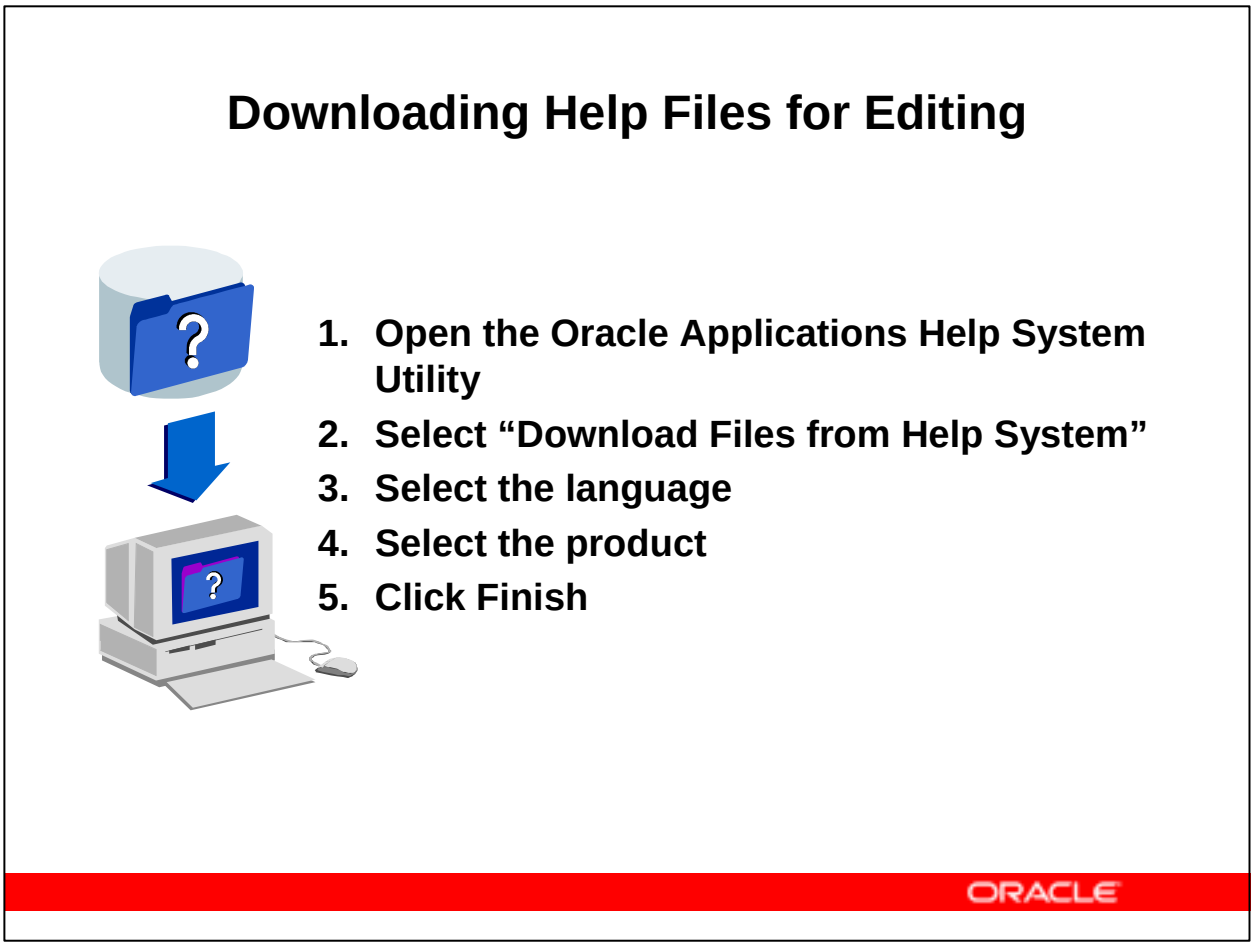


Identify the Language and Product of the Help Files

To identify the language and product of the help file, use the source document URL. The final three nodes of the source document URL are the language, the product name, and the anchor or target name.

Using this help file as an example again, the final three nodes of the URL are `/US/FND/@ht_updown`.

These nodes identify the language as `US`, the product name as `FND`, and the target name as `ht_updown`.



Downloading Help Files for Editing

You download help files by language and by product. That is, you select the language (for example, US for U.S. English) and you select the product (for example, FND for Oracle Foundation or AR for Oracle Receivables). It is important to note the code for the product because the product code determines the download directory.

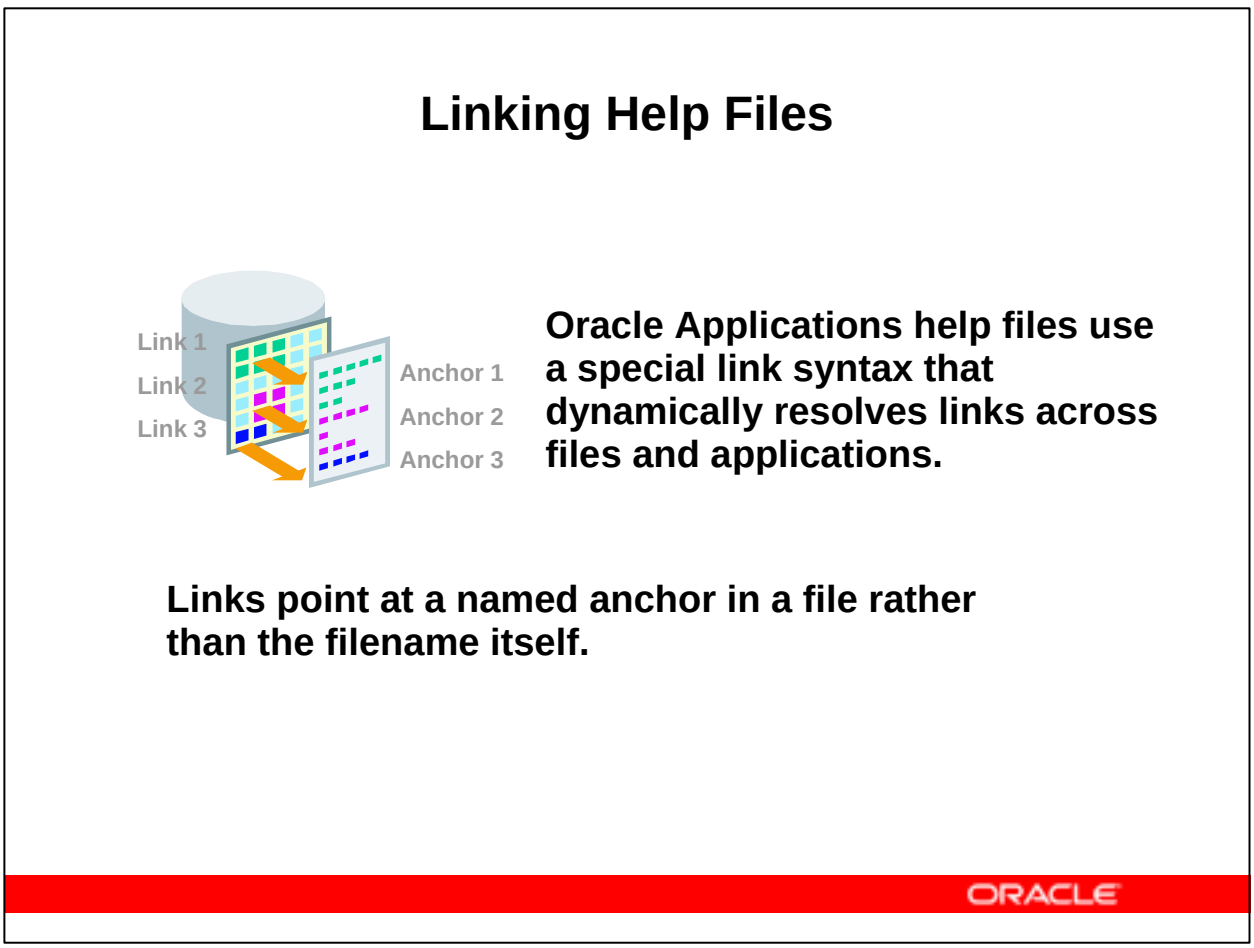
Follow these steps to download help files:

1. Open the Oracle Applications Help System Utility.

   From Oracle Self Service Web Applications click on System Administration under Web Based Applications. Under the Help Administration heading click on Help System Utility.
2. Select “Download Files from Help System” from the Choose Action option group.
3. Select the language from the Choose Language pop list.
4. Select the product code from the Choose product pop list. You can only select one product at a time.
5. Click Finish.

The Help System Utility will download the help files for the product you selected. All files will be downloaded including graphics files. The files are downloaded following the path defined in the profile option Help Utility Download Path, to a directory for the chosen language and product.

For example, if you selected the language US and the product FND, the files will be downloaded to <server location>\Help Utility Download Path\US\FND.



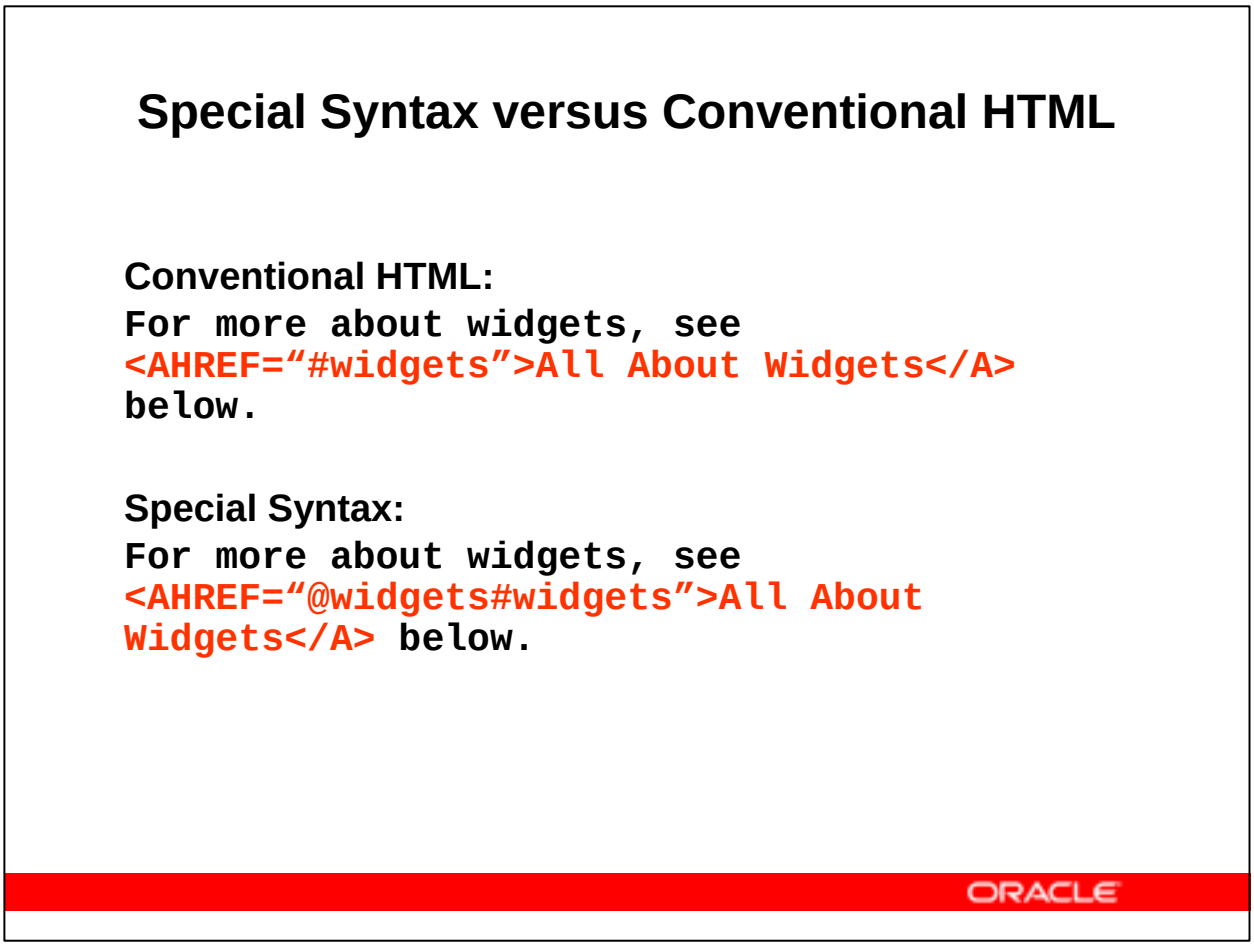
Oracle Applications Special Link Syntax: Overview

The Oracle Applications help system supports a special syntax for hypertext links that keeps them working even when files are renamed or split into parts.

Links in Oracle Applications help files point at a named anchor contained in the file rather than the filename itself. The help system resolves the anchorname to file link dynamically, every time a link is negotiated.

Information on which files contain which anchornames is put into the help system automatically on upload. Authors must ensure that anchornames are unique across an application's help files to prevent duplicate links.

It is not required that you use Oracle's special syntax for your custom help files. If you prefer, you can always use conventional hypertext links based on filename.

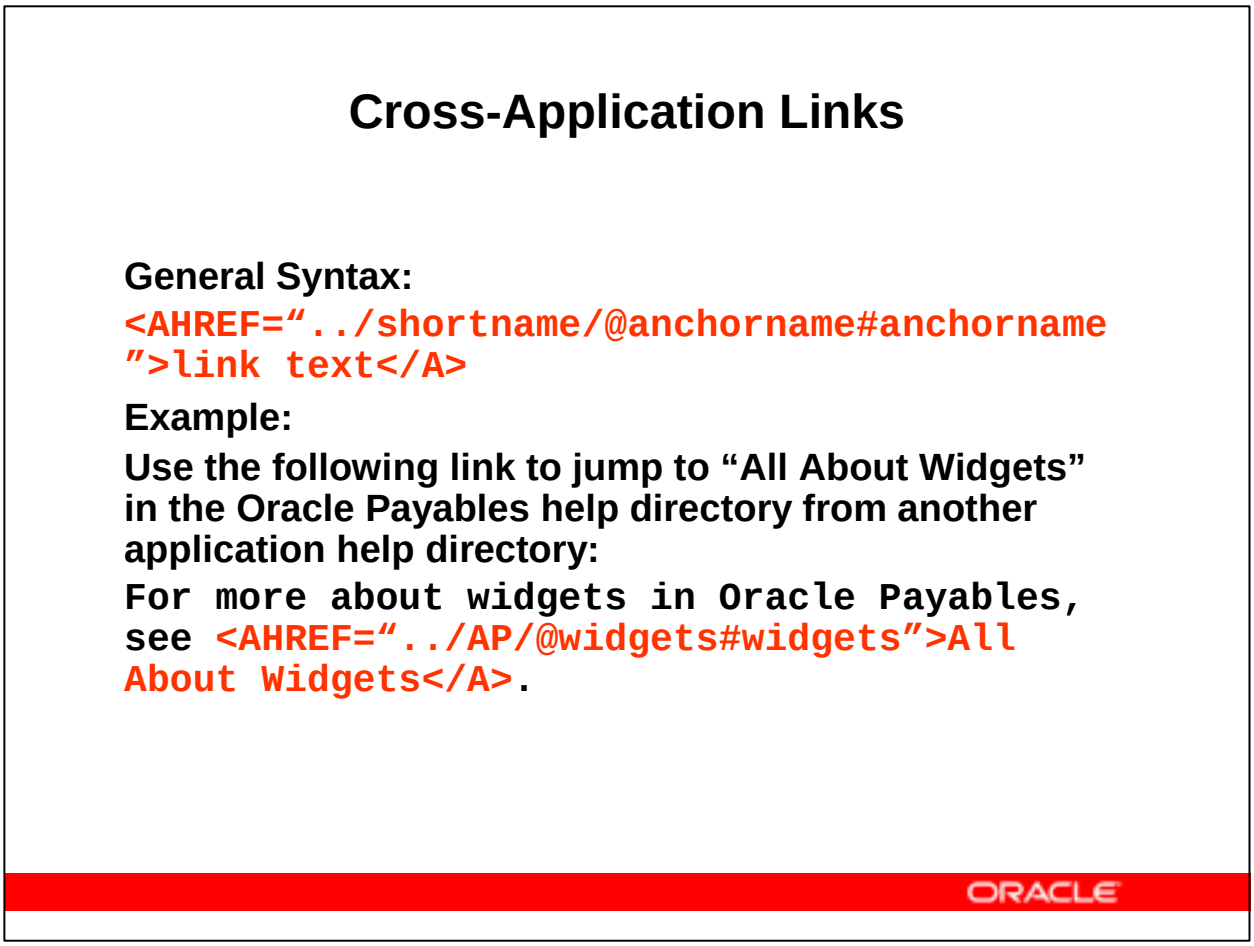


Using the Special Syntax

The conventional HTML example would allow the user to jump forward to a section with the anchorname "widgets."

Oracle Applications help files extend this HTML syntax to create links not only within, but also between help files.

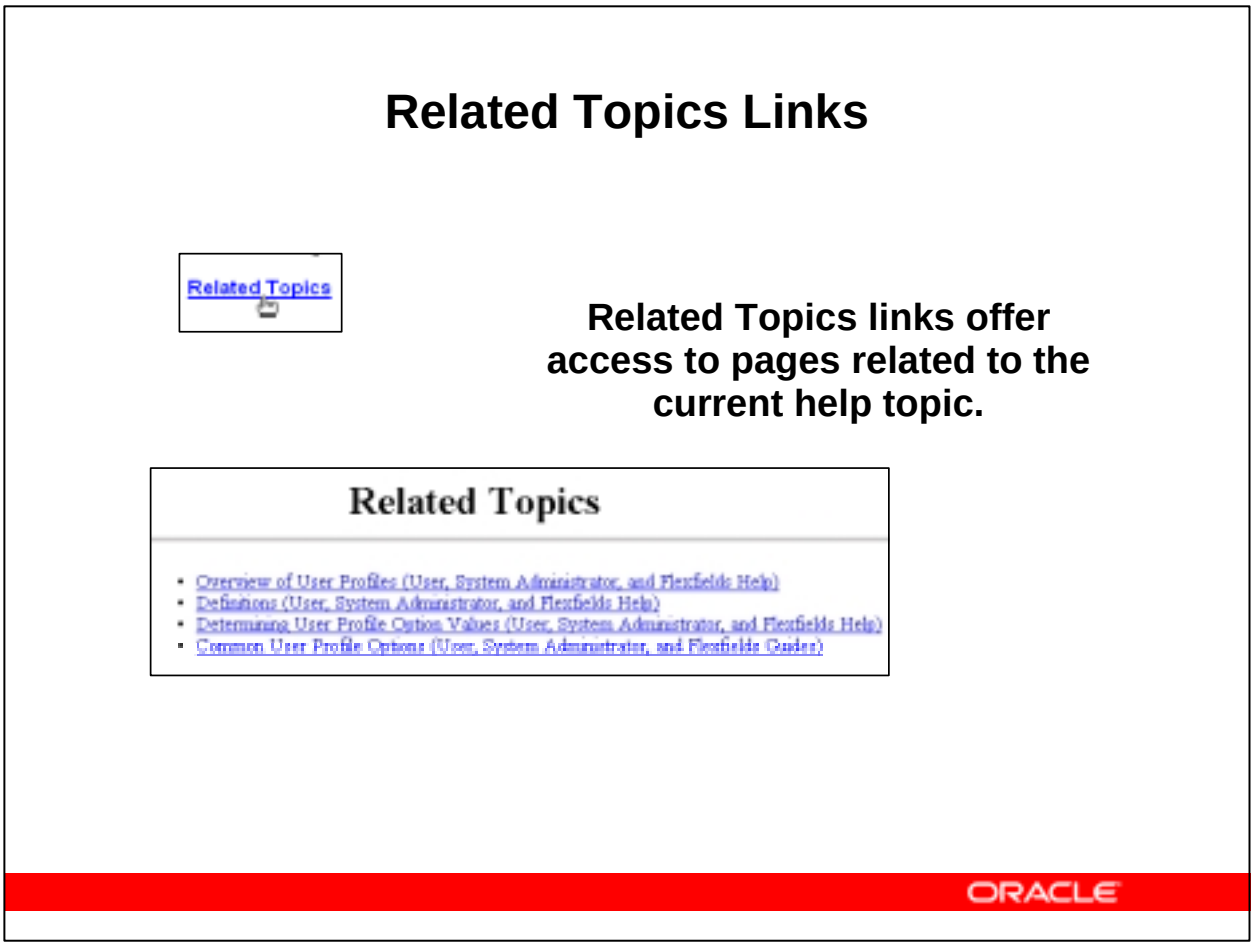
To link to a file that contains a particular named anchor, you simply place an "@" before the anchorname. To link to the precise spot within the file where the anchor appears, you append a pound sign followed by the anchorname, just as you would in conventional HTML.



Using the Special Links Syntax Across Applications

In the Oracle Applications help system, all help files associated with a particular application exist in the same directory, as far as their URL is concerned. Help files associated with other applications exist in directories named after the application's short name. All these application directories are at the same level in the help system.

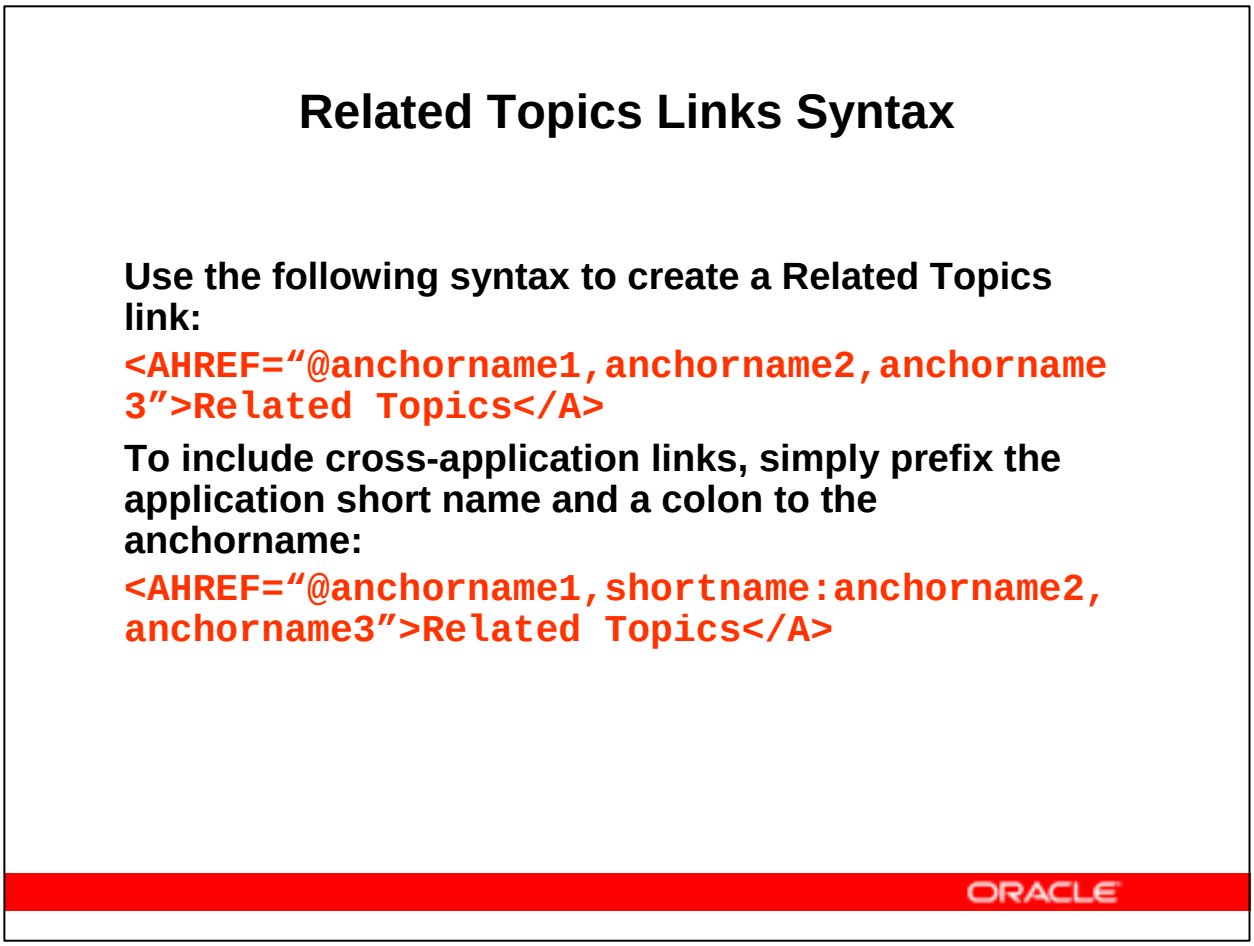
To create a link that goes to a help file associated with a different application, you create a relative link that goes up a level to the parent of all help application directories, and then back down through the other application's directory, before concluding with Oracle Applications' special link syntax.



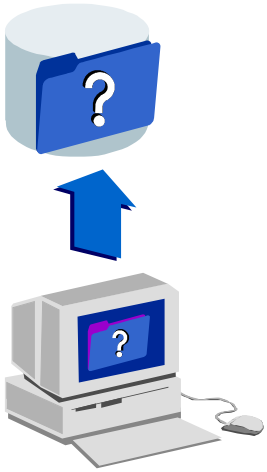
Related Topics Links

Links are not limited to a single target in the Oracle Applications help system. You can point your links at multiple topics and files by using Related Topics links.

When a user negotiates a Related Topics link, a page headed “Related Topics” appears containing a list of the related page links.



Uploading Your New or Changed Help Files



Once you have customized the help files, use the Help System Utility to upload the documents into the help system.

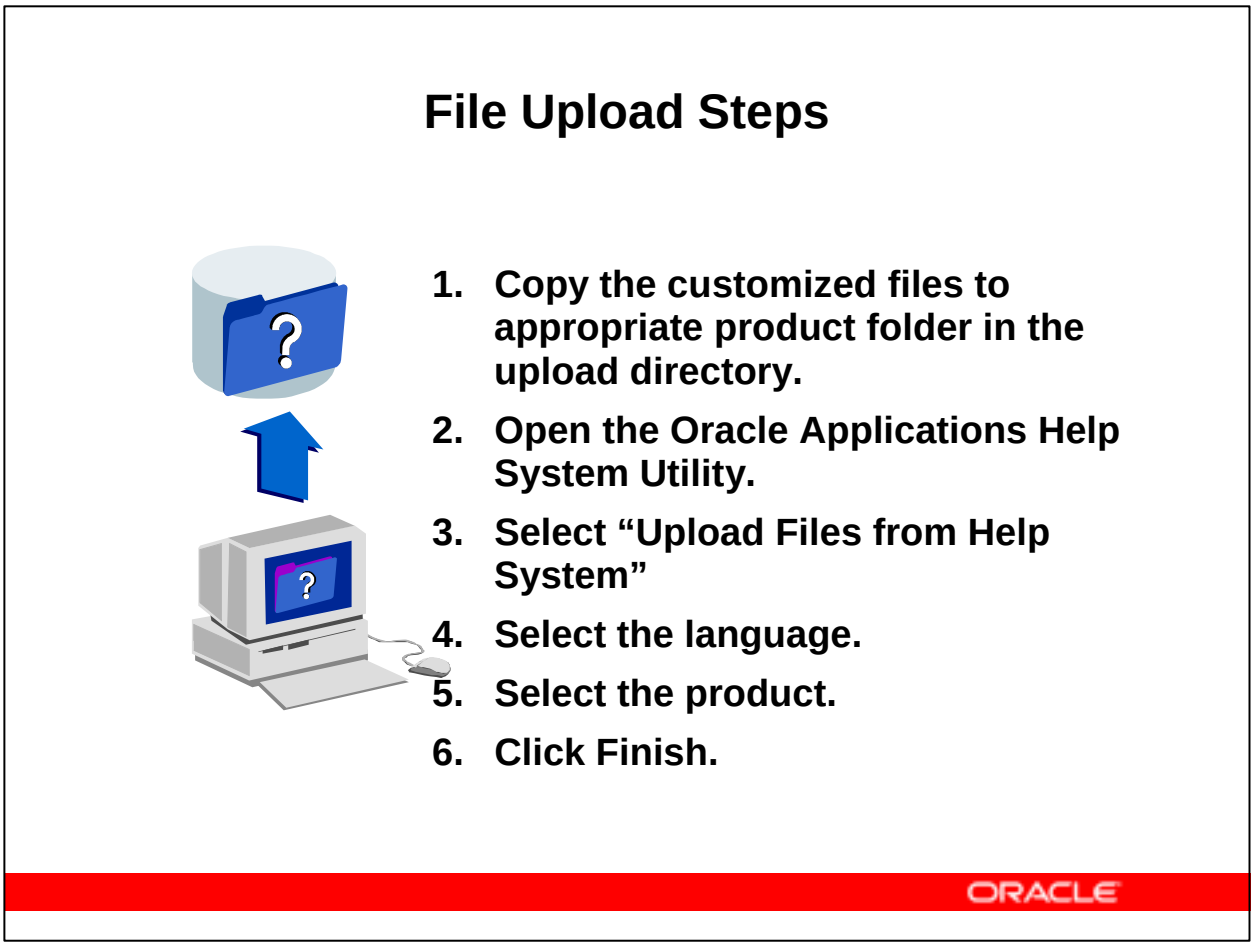
Your files are uploaded from the directory specified in the profile option Help Utility Upload Path.

ORACLE

Note:

If you have created a new application with new help files that you are uploading for the first time, you must use the Generic File Manager Access Utility (FNDGFU) to upload these files. See

Oracle Applications System Administration > Loaders > Generic File Manage Access Utility for additional information.



File Upload Steps

There are four types of files that can be uploaded to the help system. These are:

- HTML files (all HTML files must have a .htm extension)
- GIF graphics files (must have a .gif extension)
- Adobe Acrobat Files (must have a .pdf extension)
- Cascading Style Sheets (must have a .css extension)

Follow these steps to upload your customized help files:

1. Copy the customized files to the appropriate product folder in the upload directory.
   - For example, if you customized six help files for Accounts Receivable, copy the six files to the <server location><Help Utility Upload Path>\US\AR directory. You must copy the files to the correct product folder for the Oracle Applications Help System Utility to place the files in the correct location within the Oracle Applications Help System.
2. Open the Oracle Applications Help System Utility.

- From Oracle Self Service Web Applications click on System Administration under Web Based Applications. Under the Help Administration heading click on Help System Utility.

3. Select “Upload Files from Help System” from the Choose Action option group.
4. Select the language from the Choose Language pop list.
5. Select the product from the Choose Product pop list.
6. Click Finish.

The Help Utility uploads the help files for the product you selected. All files located in the directory for the selected language and product will be uploaded.

Creating Reports

The Help System Utility provides two reports for you to cross-reference help targets and file names.

- **Help Target to File Name Report – lists by target, each file that contains the target, the document title of the file, and the product.**
- **File Name to Help Target Report – lists every file name and document title by language and product and all the targets found within each file.**

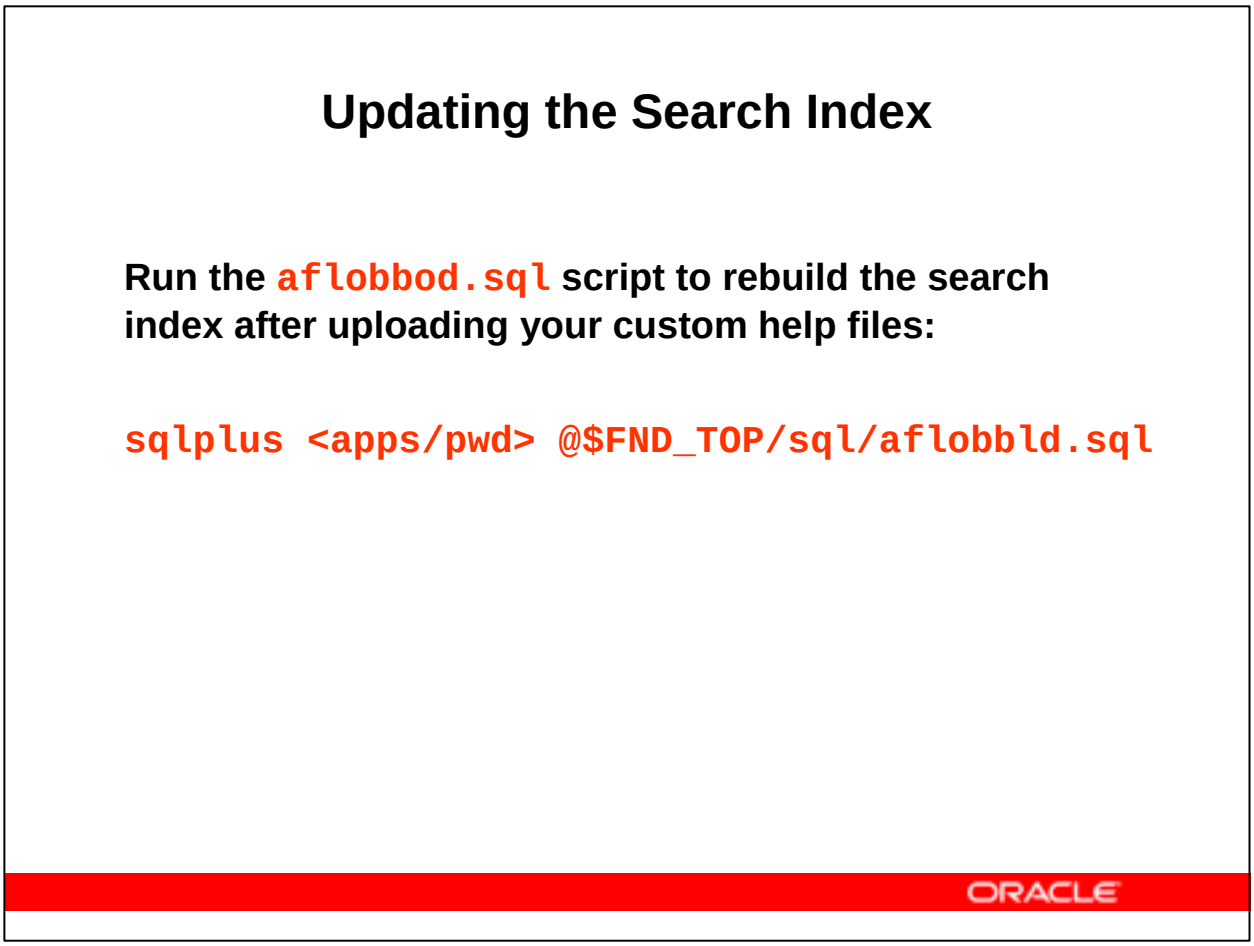
ORACLE

Running Reports

Follow these steps to run the Help System Utility reports:

1. **Open the Help System Utility**
2. **Select “Create Reports” from the Choose Action option group.**
3. **Select the appropriate report from the Create Reports pop list.**
4. **Select the language from the Choose Language pop list.**
5. **Select the product from the Choose Product pop list. To build reports for all products, choose “All Products.”**
6. **Click Finish.**

ORACLE



Rebuilding the Search Index

Oracle interMedia Text enables the search feature provided by the Oracle Applications help system. Rebuilding the search index ensures that your custom files will be included in any searches your users perform.

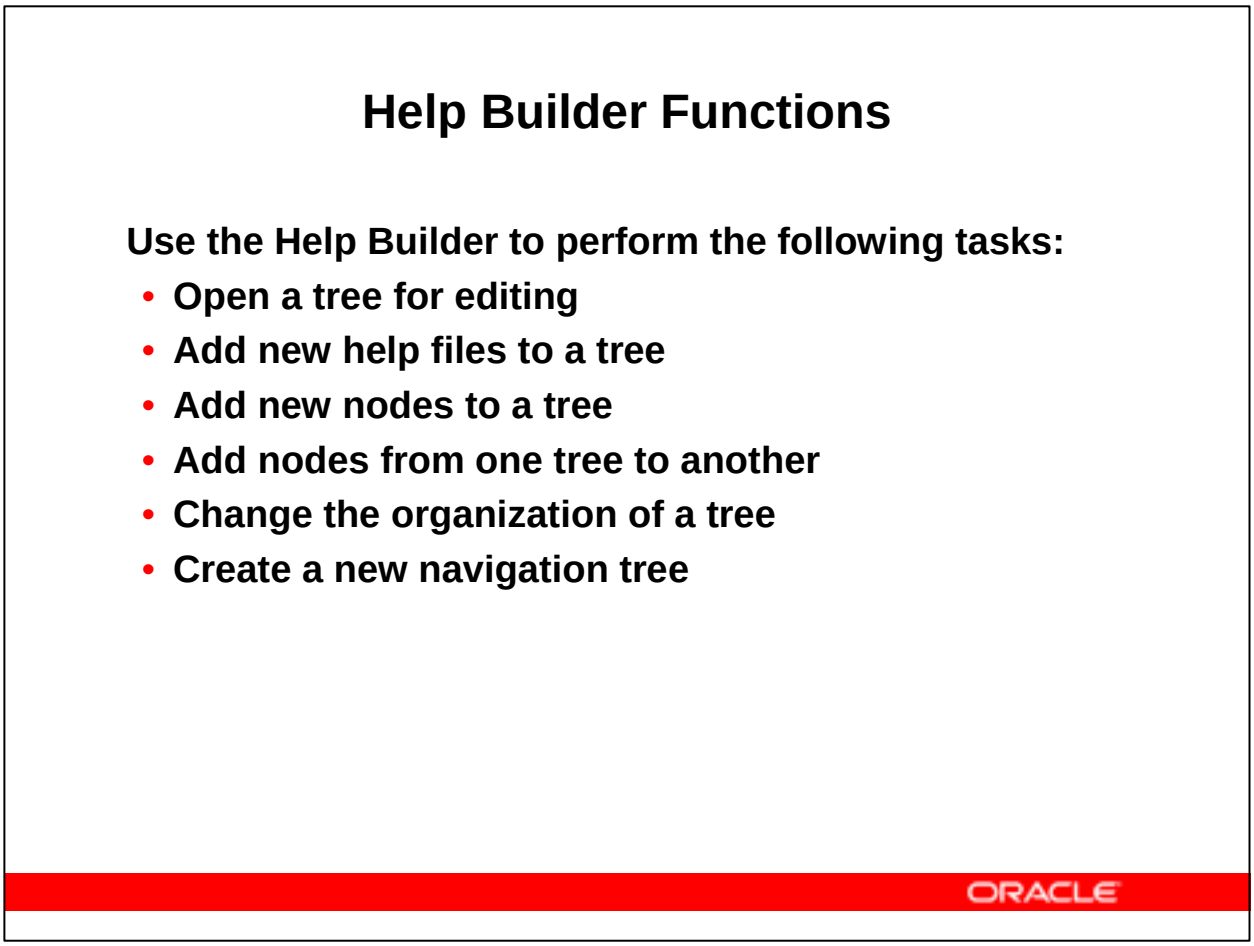
Customizing Help Navigation Trees

Use the Help Builder applet to customize the help navigation trees.

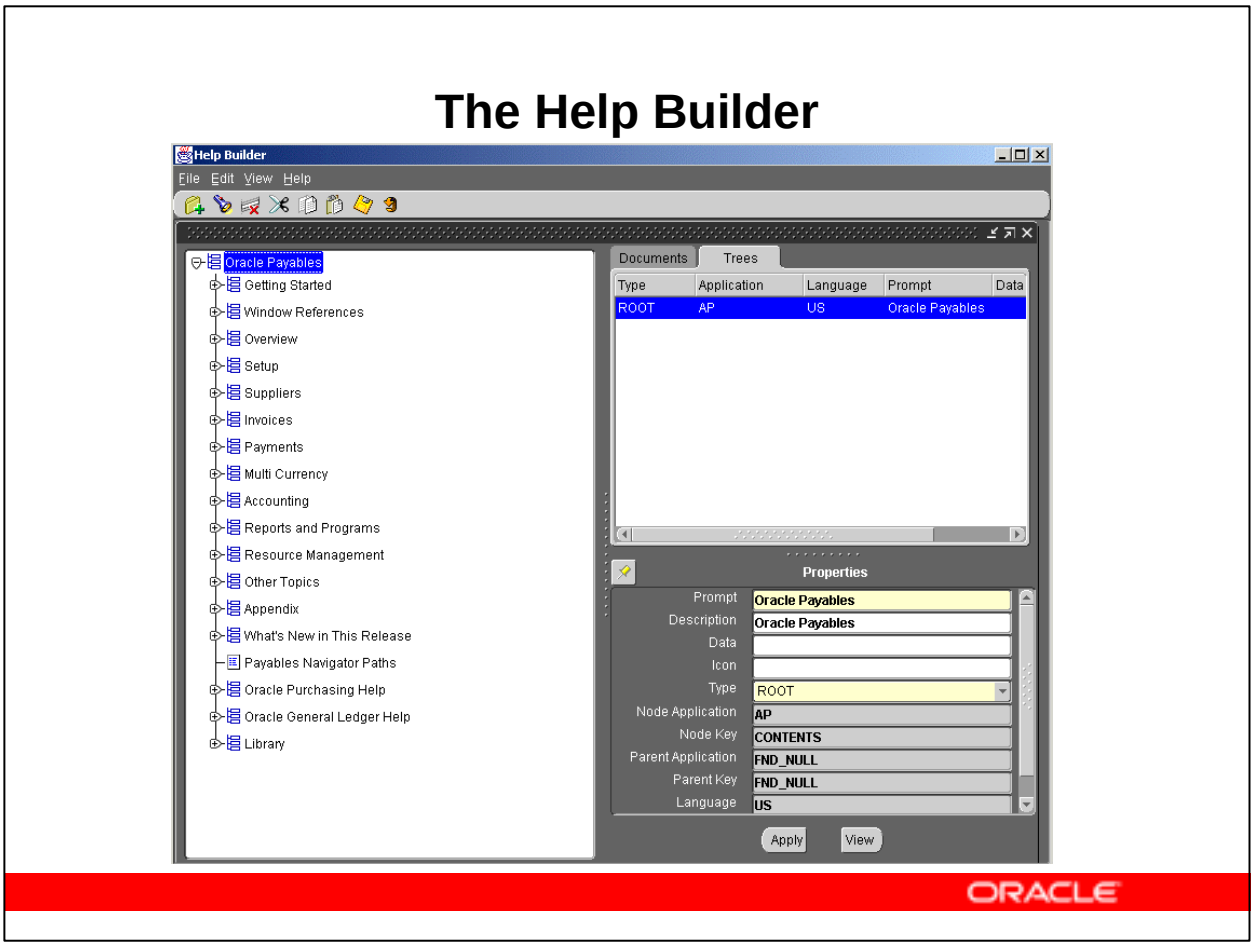
Trees are composed of a root, branches, and leaves:

- **Root - is the top-most level. When expanded, it reveals a collection of first-level nodes and documents under it.**
- **Branch- expands to reveal nodes and documents the next level down.**
- **Leaf - expands no further, but simply links to a document, terminating the hierarchy.**

ORACLE



To access the Help Builder navigate to OSSWA > System Administration > Help Builder.



The Help Builder User Interface

The Help Builder window default view consists of three panes. The left pane displays the tree that is currently selected. Use this area to manipulate your tree by adding nodes, deleting nodes, and dragging nodes into the positions desired.

The top right pane displays items matching the searches you have performed using the Find Documents or Find Trees functions. Switch from the Trees to the Documents display by clicking on the tabs.

The bottom right pane is the Properties Pane. This area displays the properties of the item (root, node, or document) currently selected. Those properties displaying enabled fields can be updated.

Opening a Tree for Editing

1. Enter information on the tree in the Find Trees window, and click Find.

   For example, enter `%Payables%` in the Prompt field to find the Oracle Payables tree.

2. Double-click the line in the Trees tab of the main Help Builder window.

   The tree's top-level node, or root, appears in the left pane. Expand and contract nodes to display the part of the tree you want to edit.

ORACLE

Adding New Help Files to a Tree

Documents are listed by Target name. Therefore if a file contains multiple targets the file will be listed multiple times. Be sure to choose the target or anchorname that corresponds to the topic you want to add.

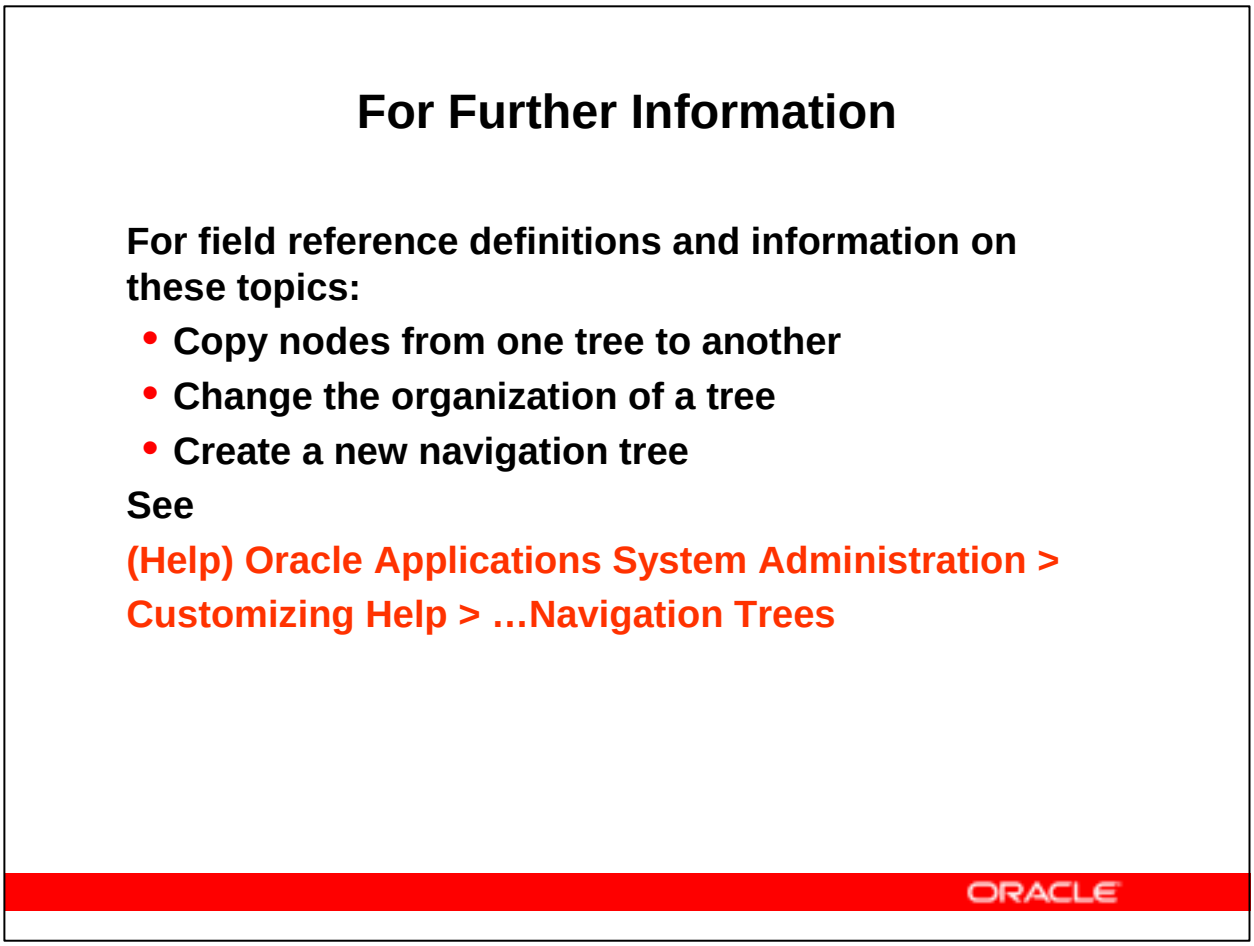
If the topic you want to add is not the header target of the file, but a target within the document, you must supply the special link syntax in the **Data** field of the Properties Pane.

For example, the target name might appear in the Data field as `@ht_updown`. To link directly to this anchor from the tree you will add `#ht_updown` to the end of the anchorname. The resulting entry in the Data filed will be `@ht_updown#ht_updown`.

Adding New Nodes to a Tree

1. After opening the tree for editing, select the node above the place where you want to add a new node.
2. Click New Node on the toolbar.
3. Enter information for the new node in the Properties Pane. Be sure to select Node from the Type list of values.
4. Click Apply.
5. To add a document to your new node, drag and drop it directly on top of the node.

ORACLE



Refer to Using the Help Builder [DEM00002]

Summary

You should now be able to:

- **Download and upload help files**
- **Use Oracle Applications special syntax to link help files**
- **Use the Help Builder to customize Help Navigation Trees**

ORACLE

Administering Folders

Chapter 11

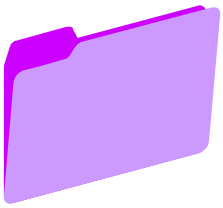
Objectives

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

- **Assign default folder definitions to a responsibility**
- **Assign default folder definitions to a user**
- **Assign ownership of a folder**

ORACLE

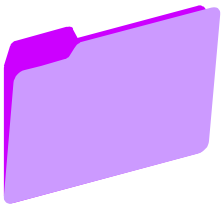
Folders: Definition



A folder is a flexible entry and display window in which you can choose the fields you want to see and where each appears in the window.

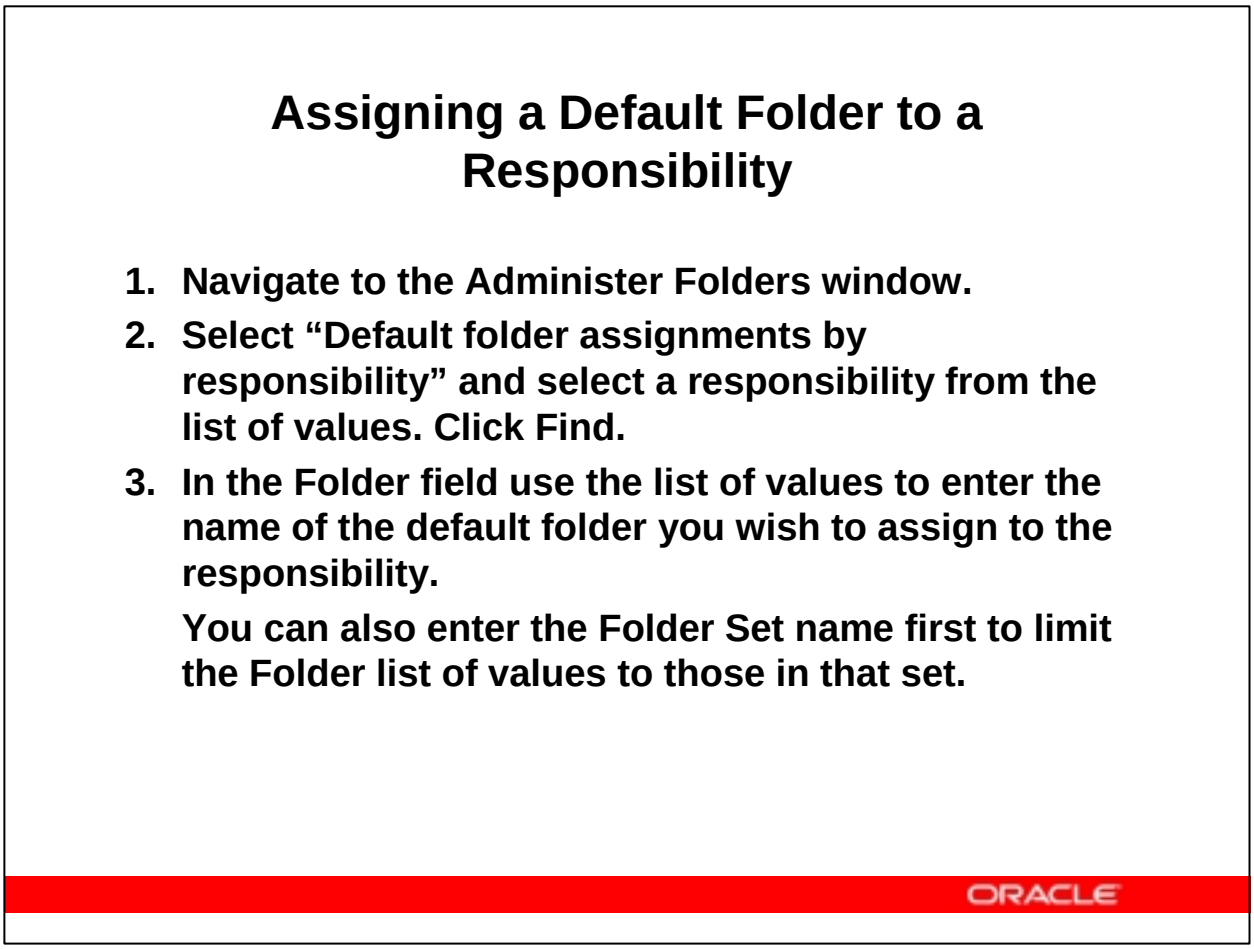
ORACLE

Administering Folders: Topics



- **Assigning default folder definitions**
  - **Responsibility**
  - **Specific user**
- **Assigning ownership of a folder**
- **Deleting a folder**

ORACLE



Assign a Default Folder to a Responsibility

(N) Application > Administer Folders

Note: You must create default folders before you can perform the following functions. For information on creating default folders see

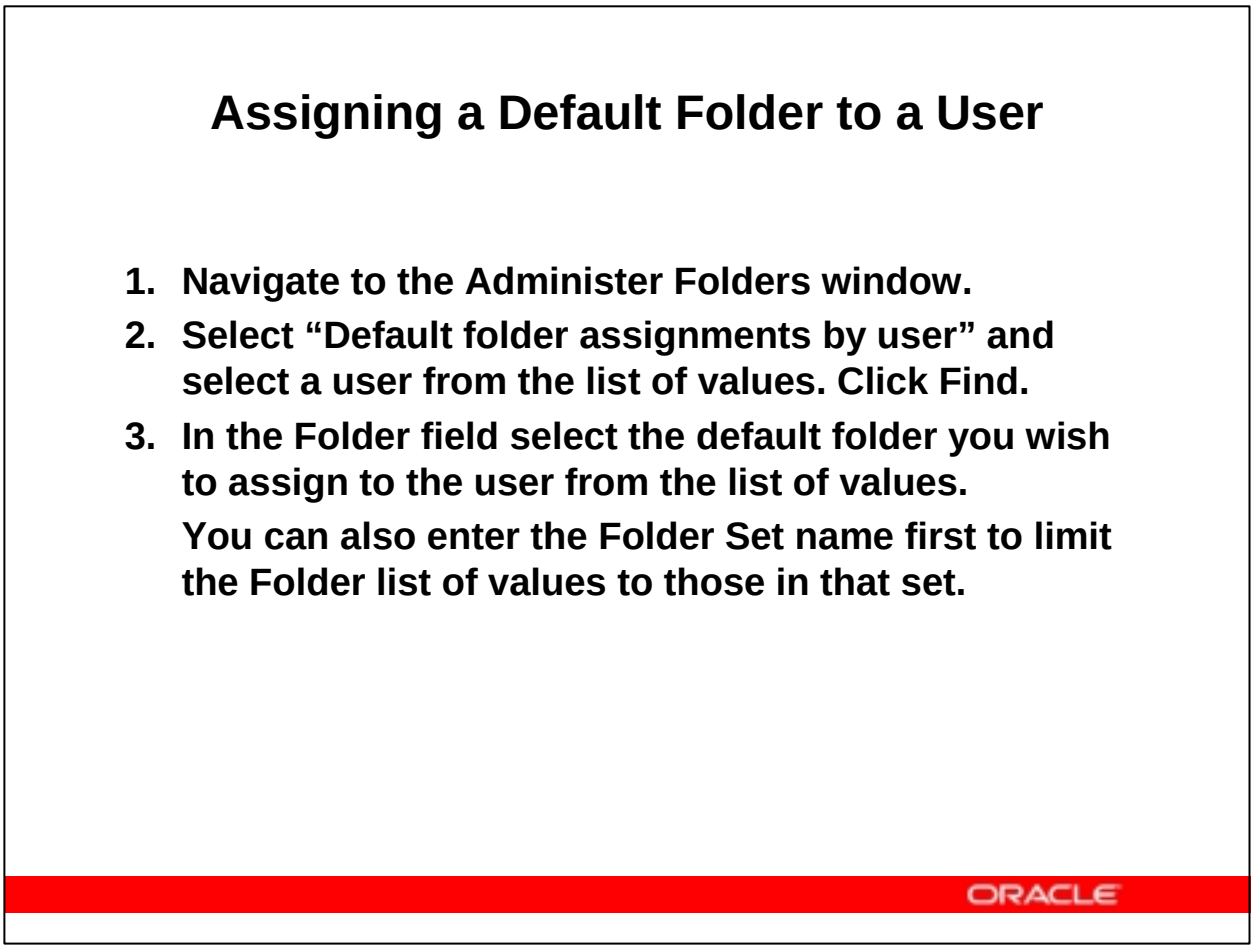
(Help) Oracle Applications User’s Guide > Folders >

Customizing the Presentation of Data in a Folder.

Note: After you save a default folder definition for a folder set, that folder set no longer appears in the list of values.

Form Fields

- Folder Set: Every folder set is associated with a particular folder block, and a user or responsibility can have one default folder within each folder set. The folder set name generally describes the records shown in the block; some blocks may have multiple sets of folders associated with them.



Assign a Default Folder to a User

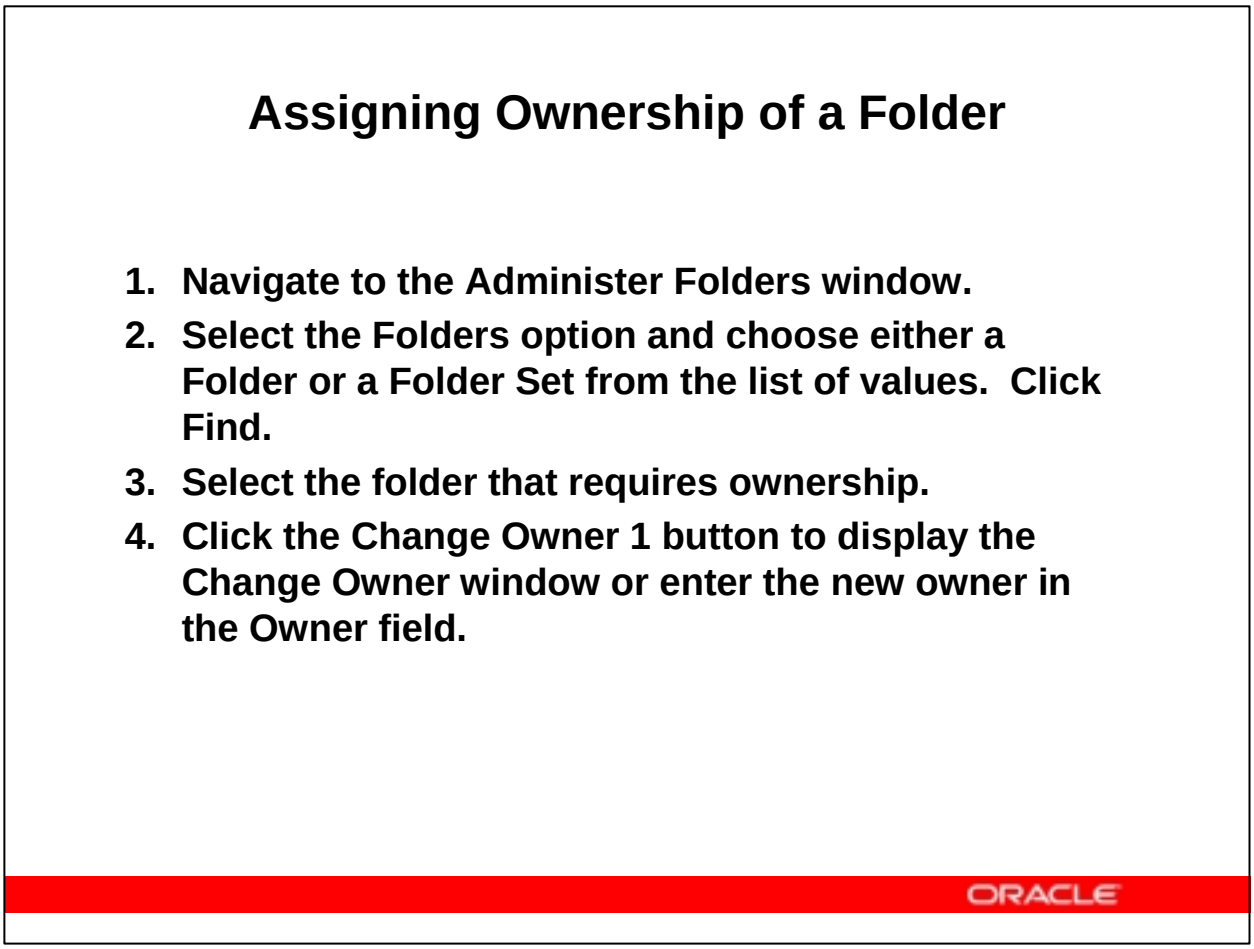
(N) Application > Administer Folders

Note: After you save a default folder definition for a folder set, that folder set no longer appears in the list of values.

Form Fields

- Source Type: Enter either User or Responsibility. Records entered in this window use the source type of User. If one of the current user’s responsibilities has default folders defined, the default folders are listed with a source type of Responsibility.

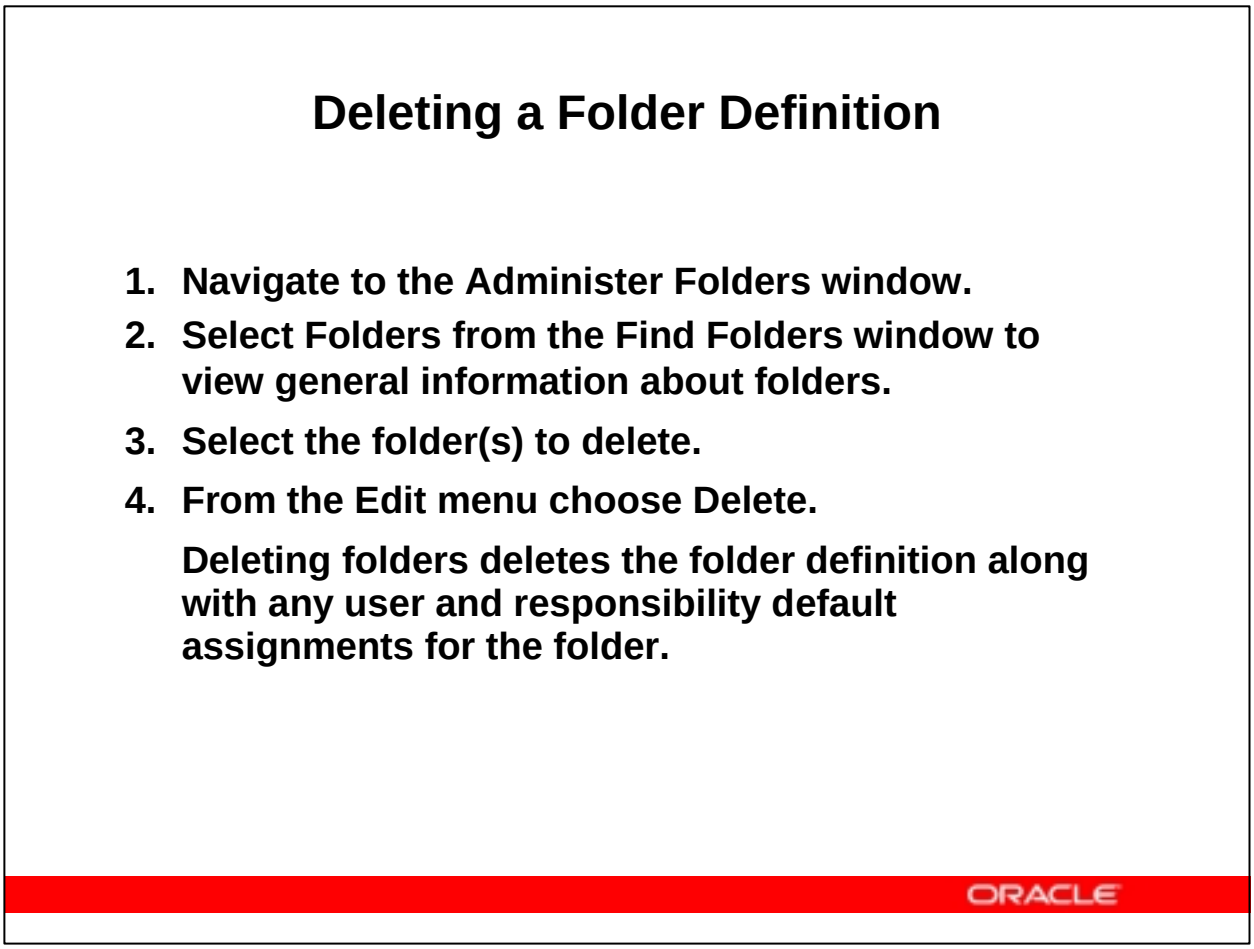
  User defaults override Responsibility defaults. You cannot delete Responsibility default folders in this window.
- Responsibility: The Responsibility that uses this default folder definition.



Form Fields

(N) Application > Administer Folders

- Public: This check box governs the availability of the folder to all users.
- Anyone's Default: Indicates whether this folder definition is used as a default by a user or a responsibility. If it is a default definition, use Default Assignments to view the users and responsibilities for which it is the default folder definition.
- Default Assignments: The users and responsibilities that use this folder definition as a default.



(N) Application > Administer Folders

Guided Demonstration - Folders

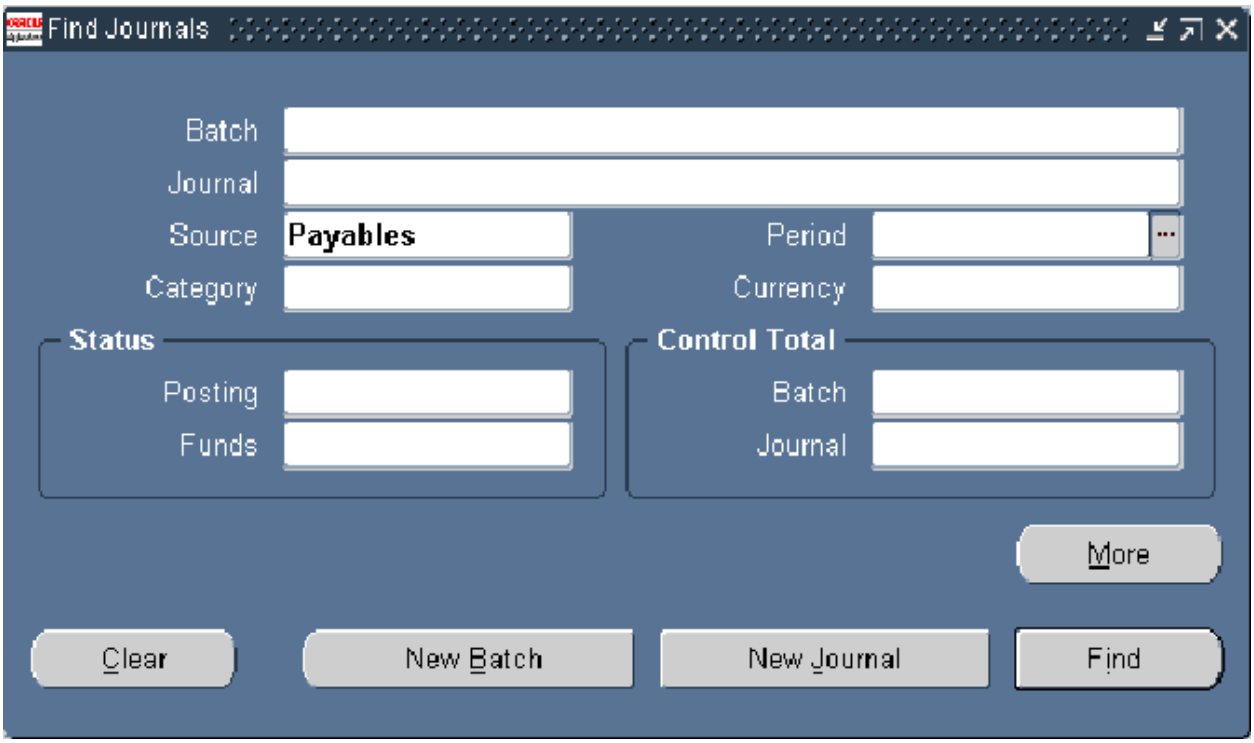
Overview

The committee wants you to test setting up and administering a Folder form. But, they are willing to do a single test. You will see the following demonstrated.

- Create the folder
- Assign a default folder definition to a user
- Set the profile option to control modifications of the folder by the user

Tasks

1. Using your General Ledger Super User responsibility, navigate to (N) Journals > Enter.
2. The first window you see is the Find Journals window. Note that once you create your folder and assign it to your user this window will not appear.
3. In the Find Journals window enter, **Source**: Payables.
4. Click the “Find” button.



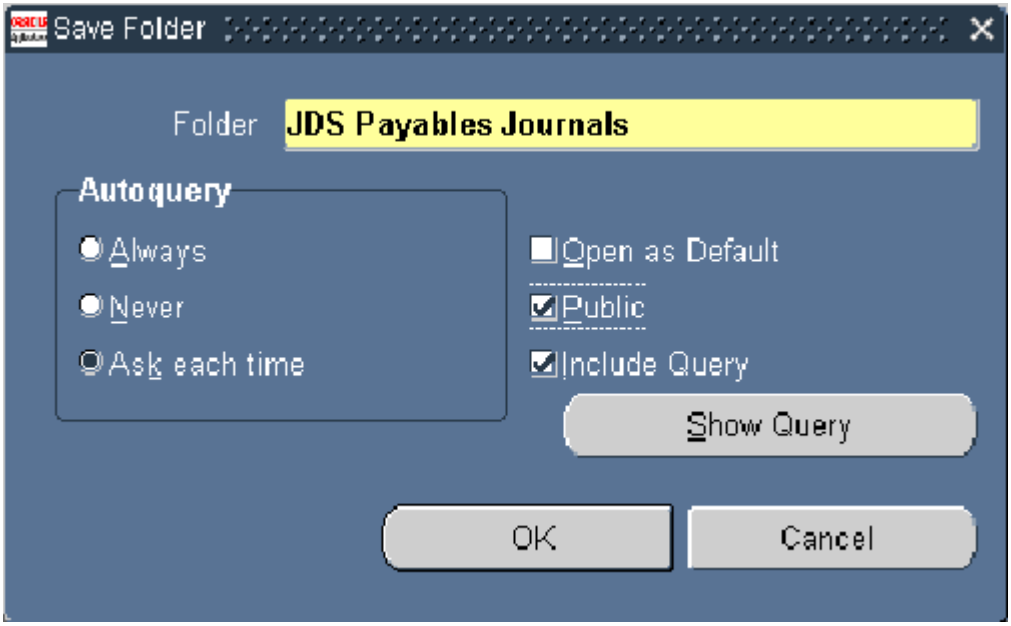
5. In the Enter Journals window, choose (M) Folder > Save As….
6. In the Save Folder window, enter the following:

- **Folder**: *Your Initials* Payables Journals
- **Autoquery**: Ask each time
- **Public**: enable check box
- **Include query**: enable check box

7. Click the “Show Query” button.

8. The Show Query window displays (JE_SOURCE=‘PAYABLES’). **Note:** you can view the query here, but you cannot change it. Click the “OK” button to close the window.

9. In the Save Folder window click the “OK” button



10. In the Enter Journals window, the upper left corner next to the Folder icon is the name you entered for your folder.

11. Click on the Folder icon to launch the Open Folder window. **Note**: the other folders that are available for your selection. These appear because they are defined as Public.

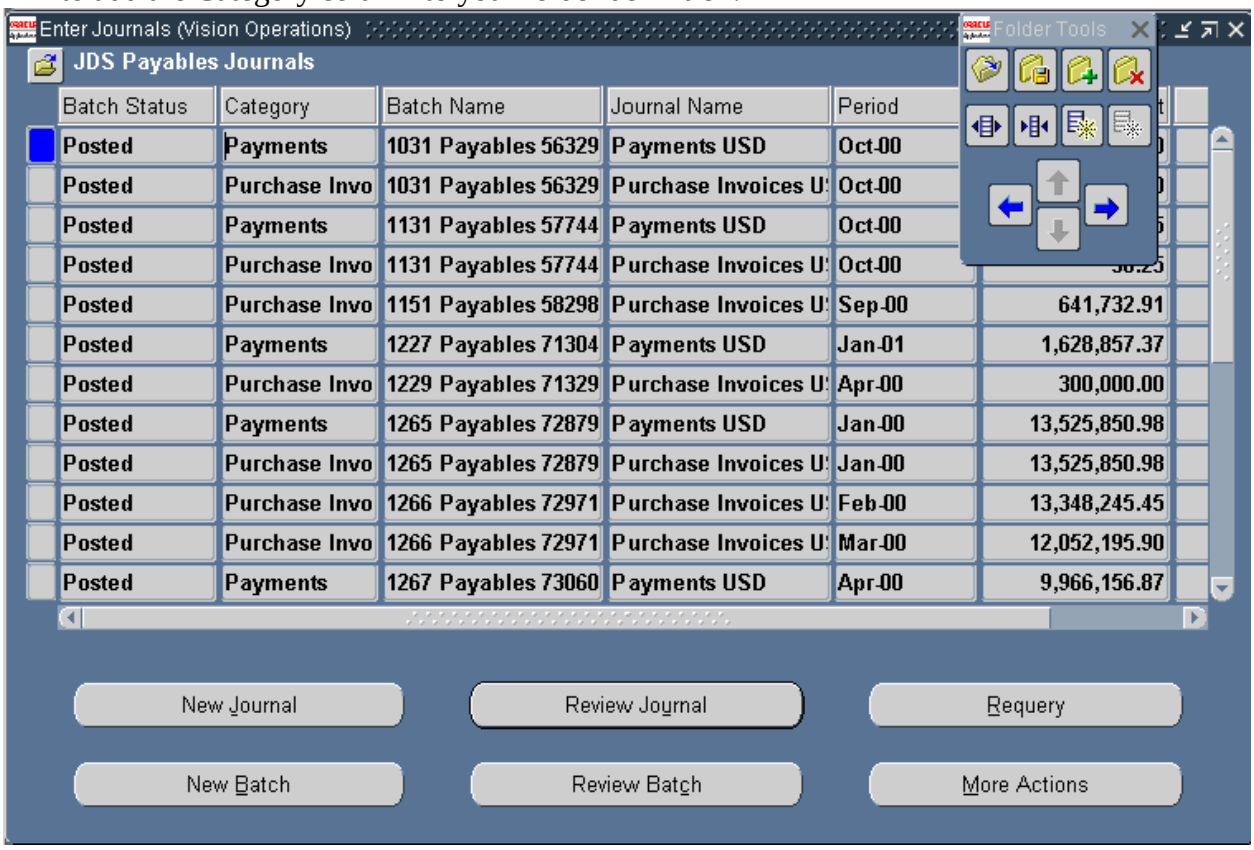
12. Click the “Cancel” button.

13. Click on the Folder Tools icon on the toolbar (the second icon from the right) to launch the Folder Tools window palette. Move your cursor over each icon to view its function.

14. Click the “Show Field…” icon to launch the Show Field selection window. This allows you to add any of the listed fields to the Enter Journals window display.

15. Choose “Category” from the Show Field window and click OK. **Note**: the “Category” field has been added to the display.

16. Click the “Save” icon on the Folder Tools window, or choose “Save” from the Folder menu to add the Category column to your folder definition.

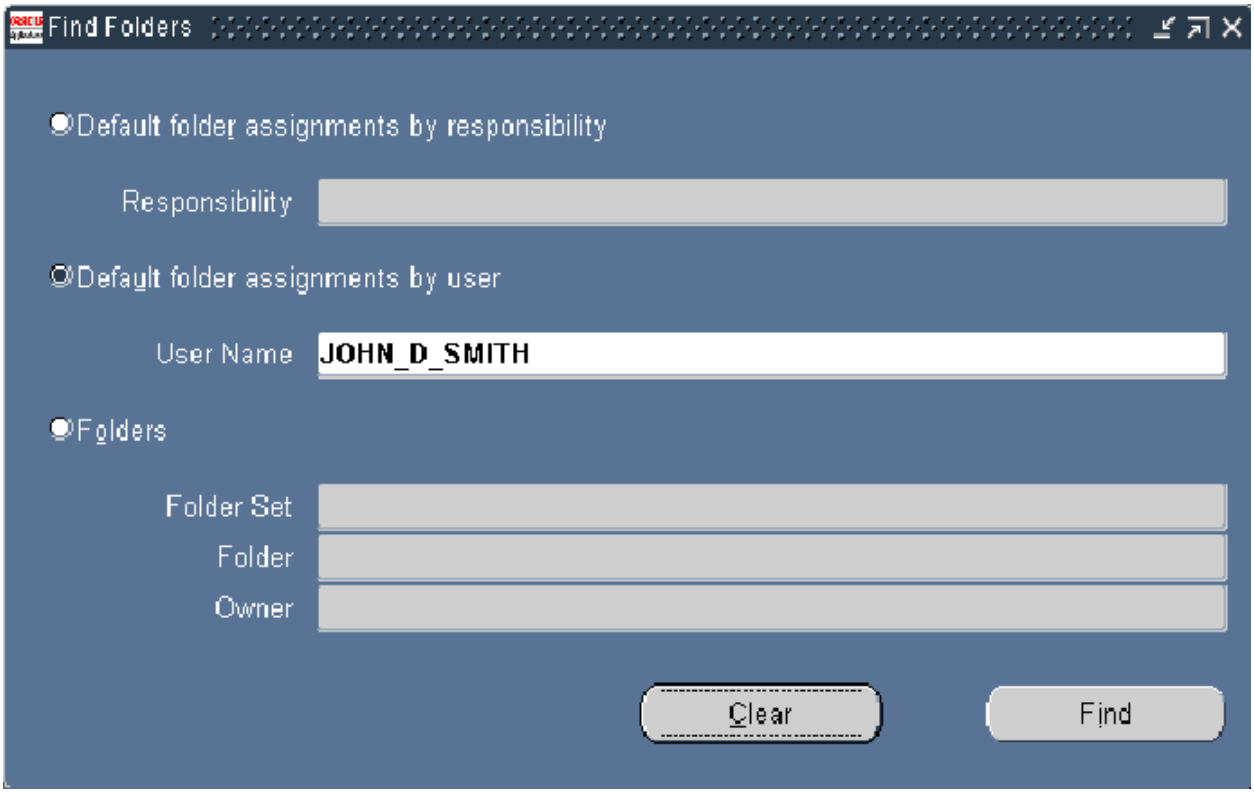


17. Switch Responsibilities to your System Administrator responsibility.

18. Navigate to (N) Application > Administer Folders.

19. In the Find Folders screen, select “Default folder assignments by user” and choose your user name from list of values.

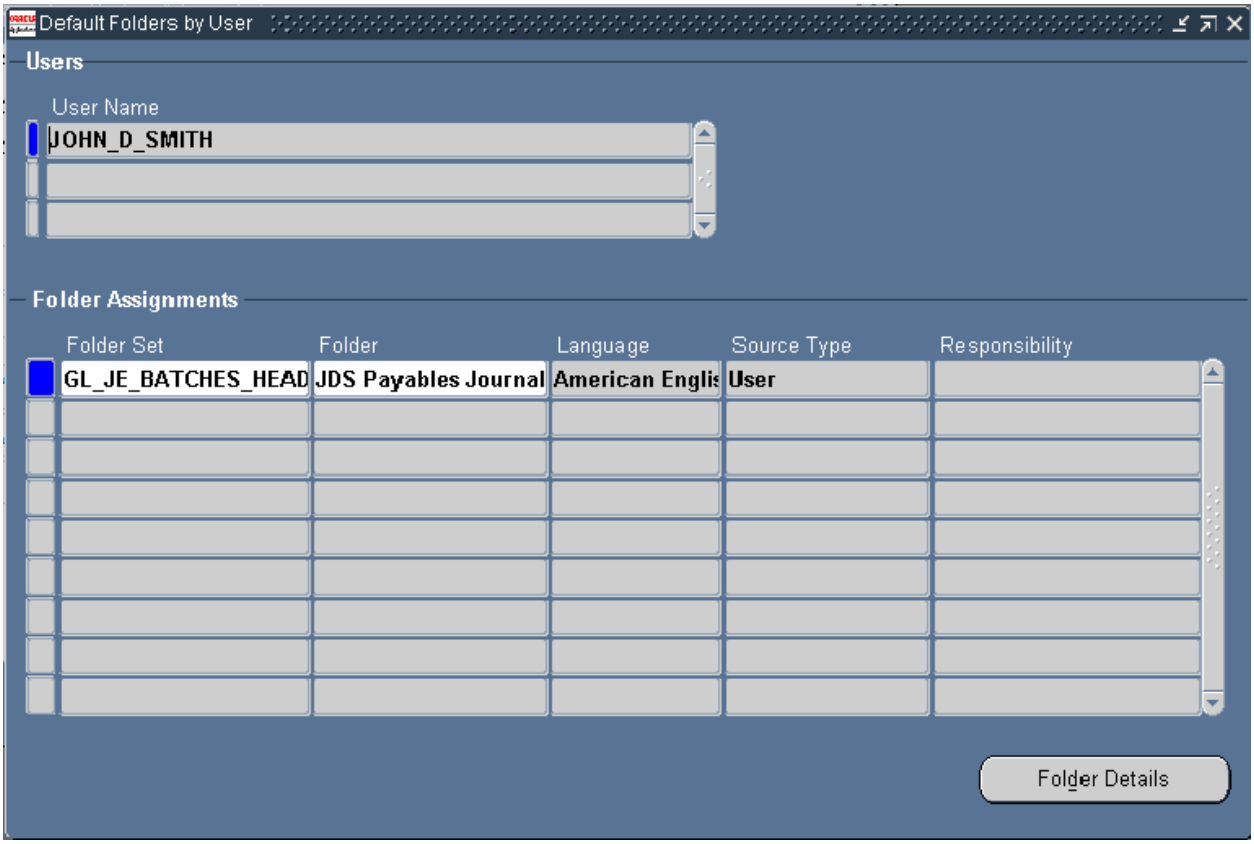
20. Click the “Find” button.



21. In the Default Folders by User window, enter the following.

    - **Folder Set**: GL_JE_BATCHES_HEADERS_V
    - **Folder:** *Your Initials* Payables Journals

22. Save your work.

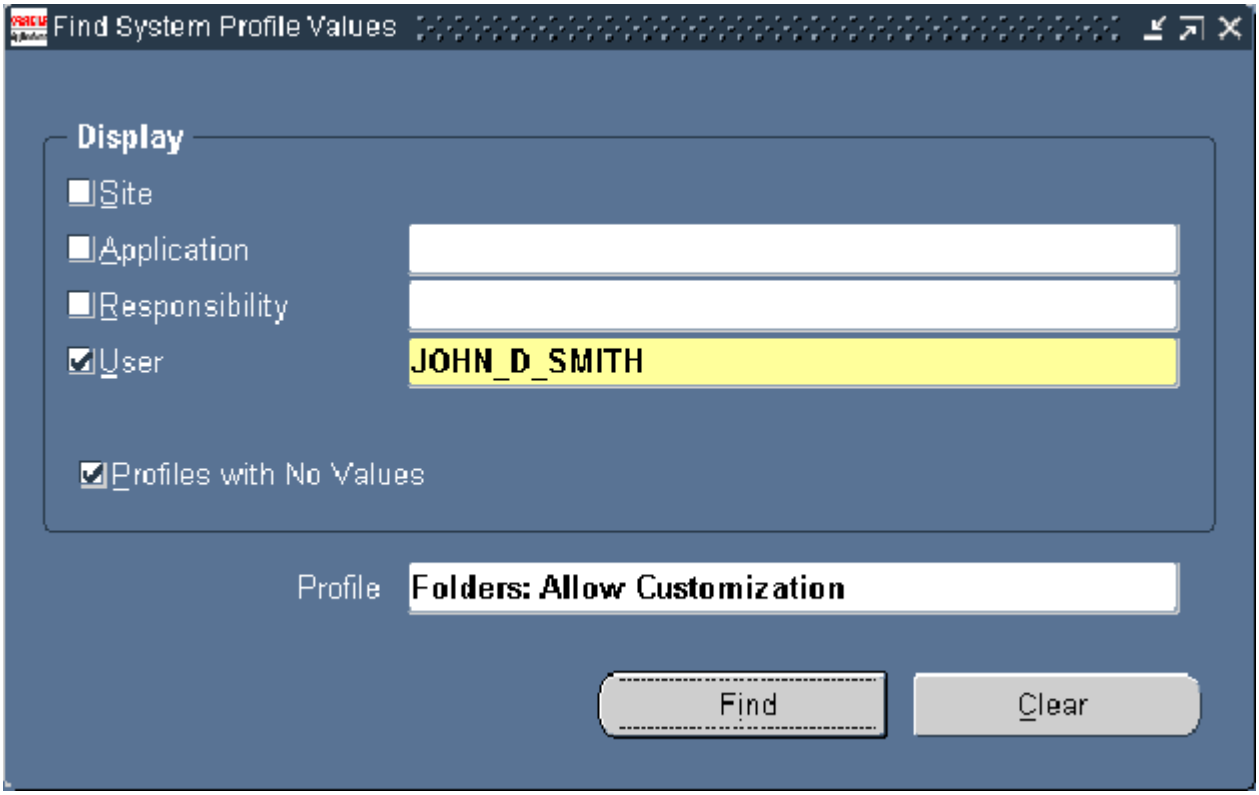


23. Navigate to (N) Profile > System.

24. In the Find System Profile Values window enter the following.

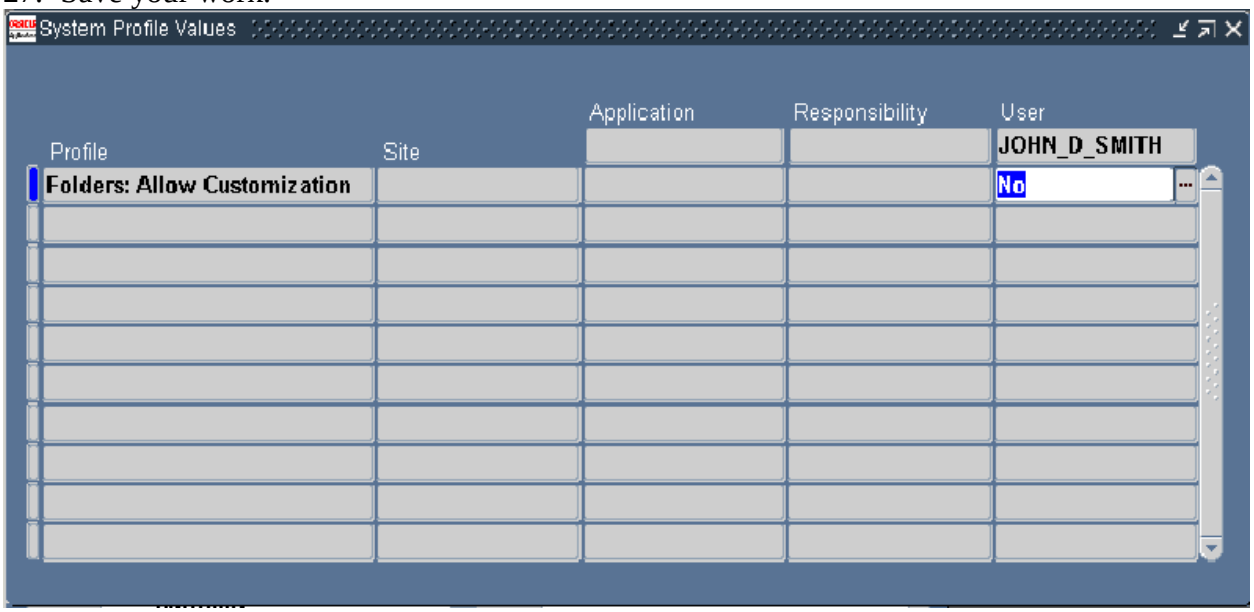
    - **User**: *Your Initials* User
    - **Profile**: Folders: Allow Customization

25. Click the “Find” button.



26. Set the Profile Option to disallow modification by your user. In the System Profile Values screen enter “No” as the User value for the profile option.

27. Save your work.



28. Switch Responsibilities back to your General Ledger Super User.

29. Navigate to (N) Journals > Enter.

30. Notice that you see a Decision window that says “Find all records in this folder?” This window displays because you chose “Ask each time” when you defined the folder.

31. Click the “Yes” button.

32. Notice that the Journal Entries window opens with your folder displayed.

33. Click on the “Folder Tools” icon on the toolbar. Notice that all the modification icons are disabled (because you set the Folders: Allow Customization profile option to No).

34. Click on the “Folder” menu and notice that the only enabled option is Open.

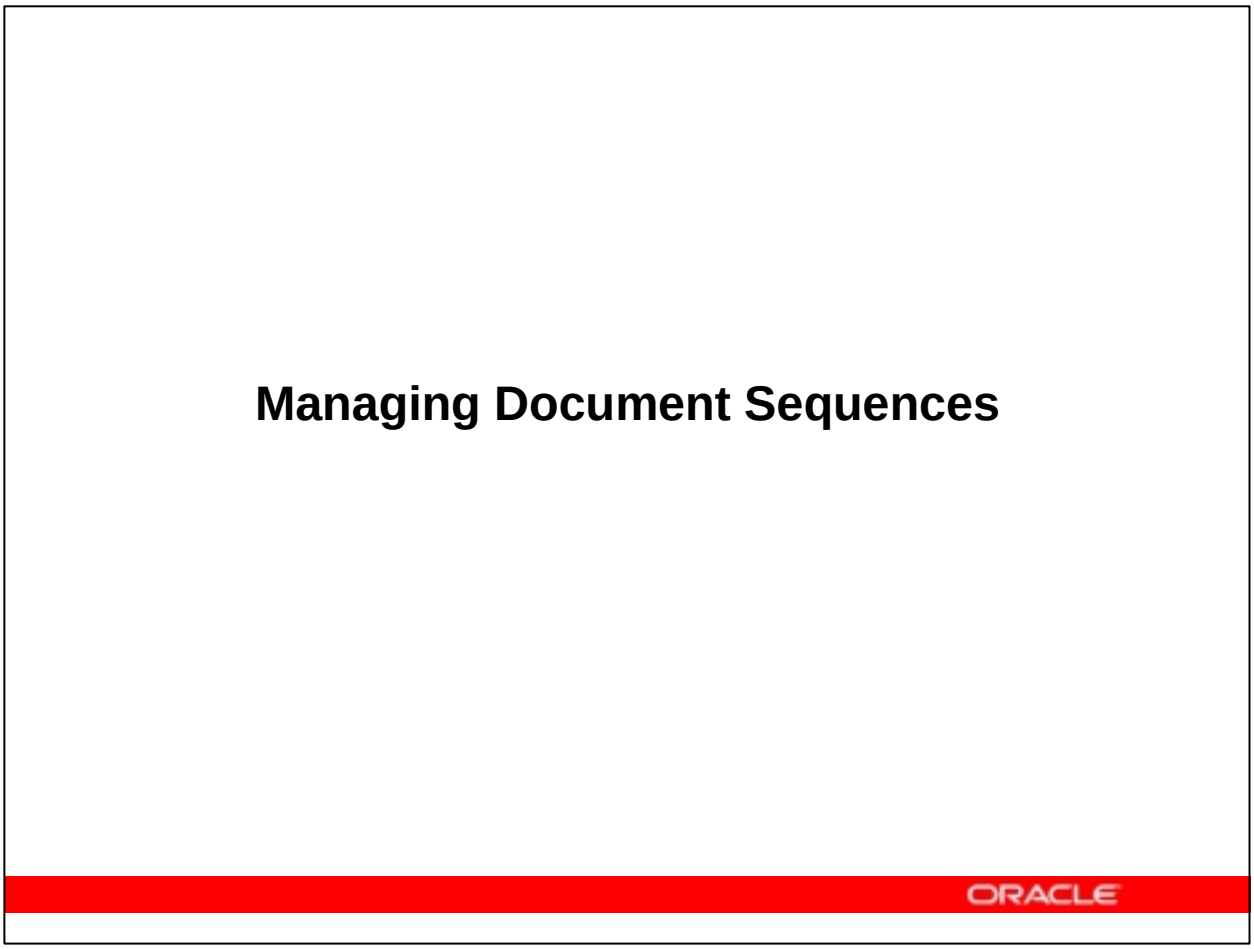
Summary

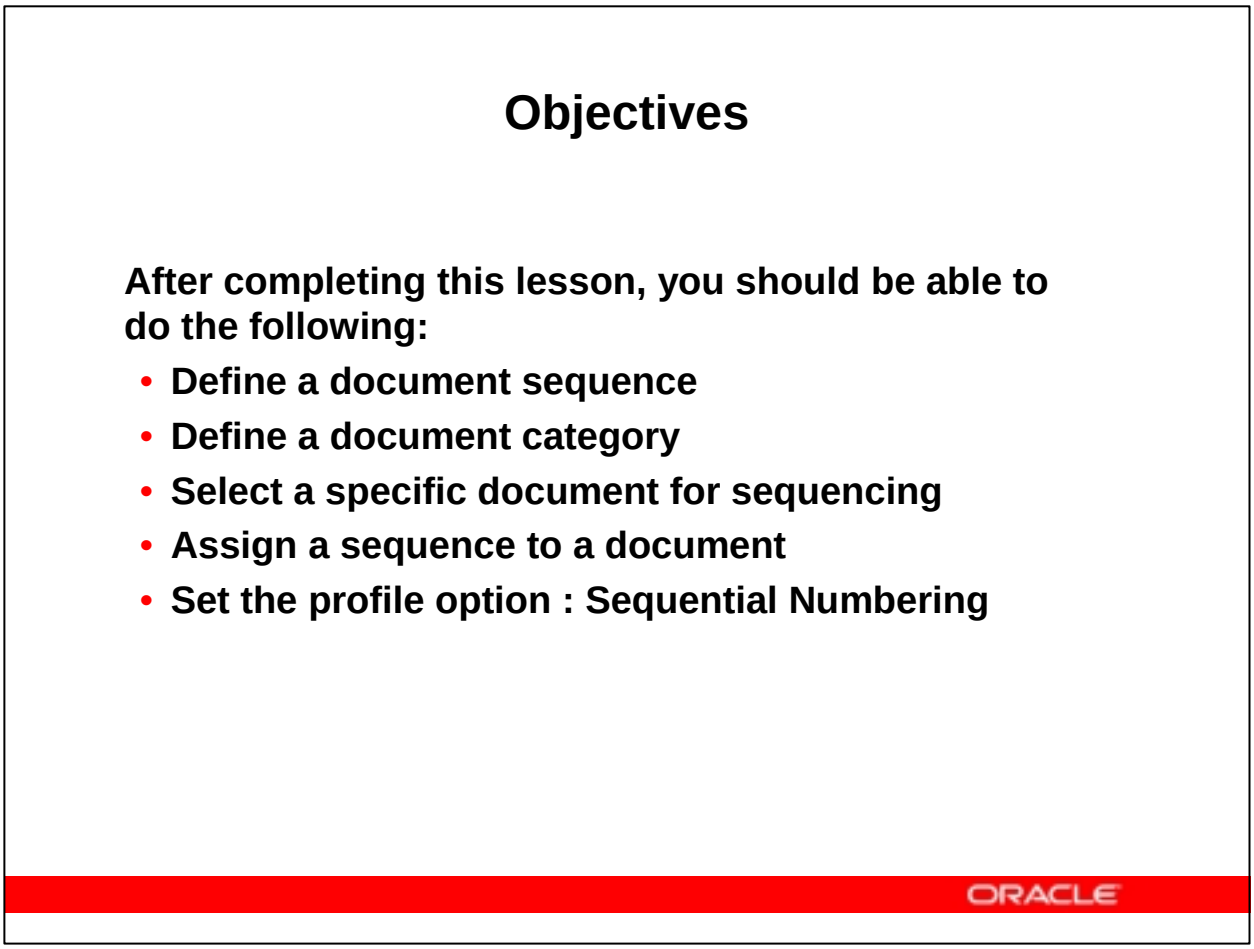
You should now be able to do the following:

- **Assign default folder definitions to a responsibility**
- **Assign default folder definitions to a user**
- **Assign ownership of a folder**

Managing Document Sequences

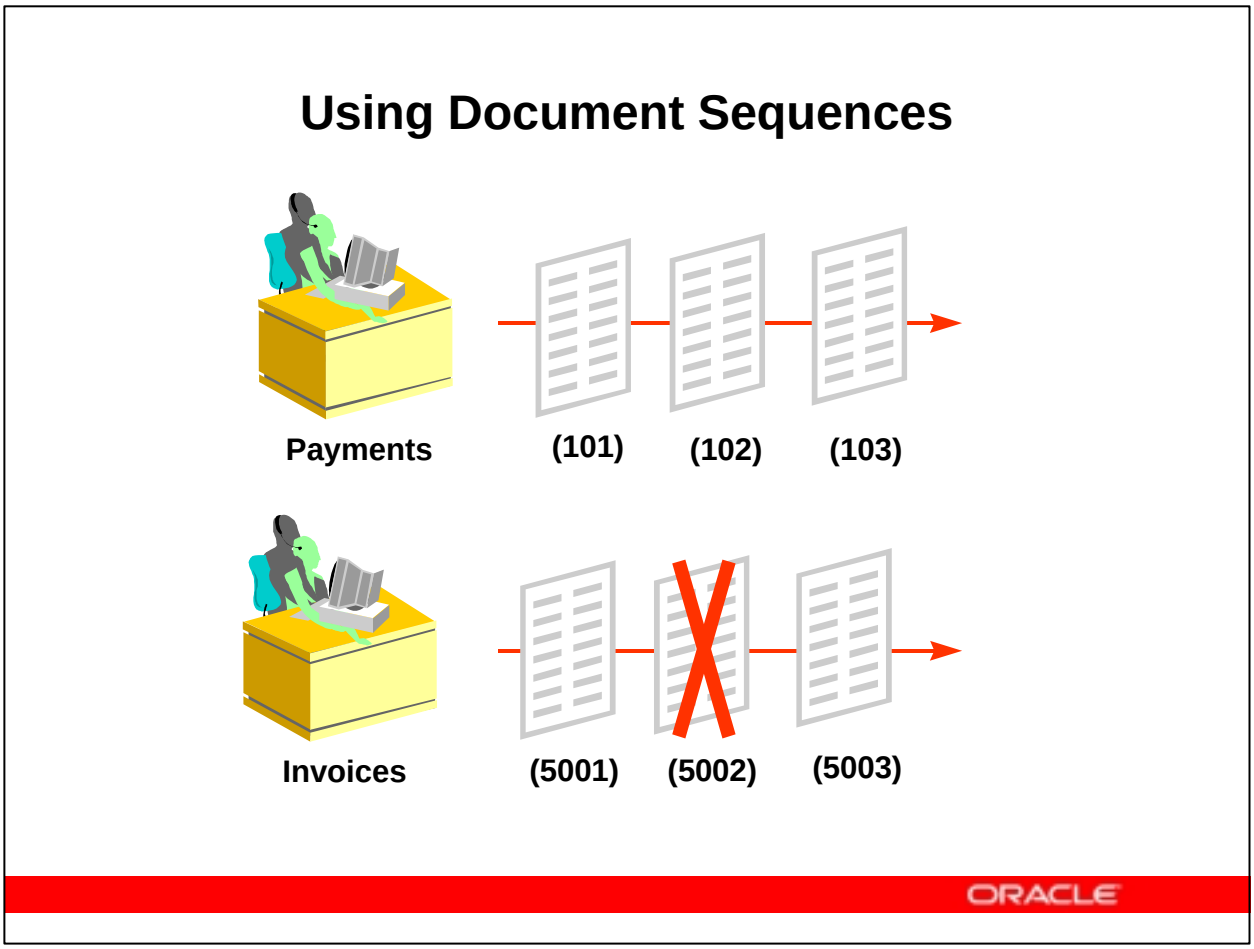
Chapter 12





Lesson Aim

A document sequence is an internally managed log of transactions against a particular table. Sequences enable complete auditing of all transactions, whether successful or not. This lesson covers creating and implementing document sequences.



Why Use Document Sequences?

Transactions Create Documents

- Whenever you enter a transaction, you create a document. For example, when you enter a payment, you create a payment document. Or when you enter an invoice, you create an invoice document.
- You can assign unique numbers to the documents that you create by using document sequences.

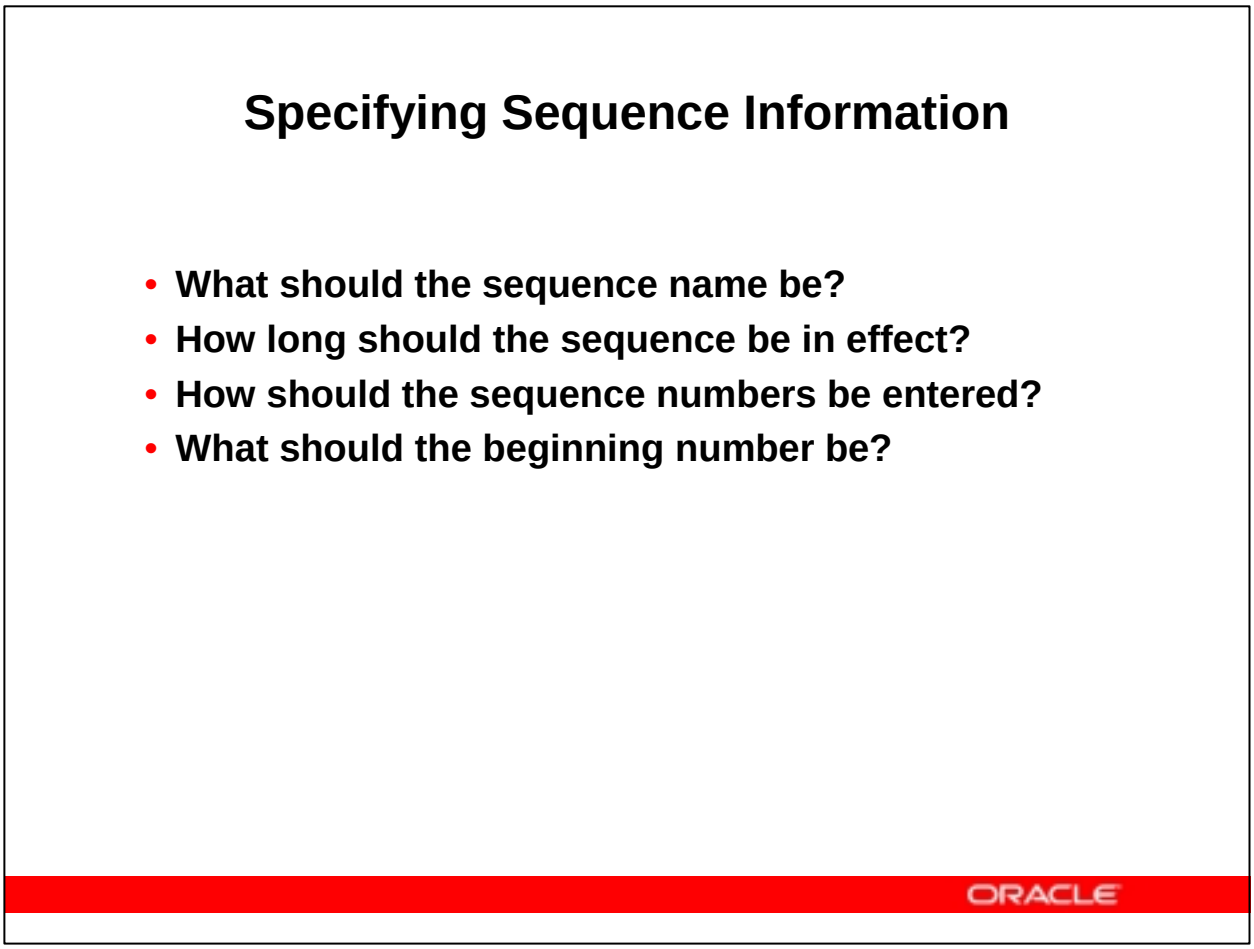
Document Sequences Provide Completeness

- Completeness is proof that every document created by a user's transaction can be accounted for.
- By assigning unique numbers to documents, you can use document sequences to account for all transactions, even those that fail.

Sequences, Categories, and Assignments

- Specify sequence information to determine how documents are numbered.
- Define document categories to group documents together.

- Associate a sequence with a document category to determine which documents the sequence will number.



Using Manual or Automatic Sequence Numbering

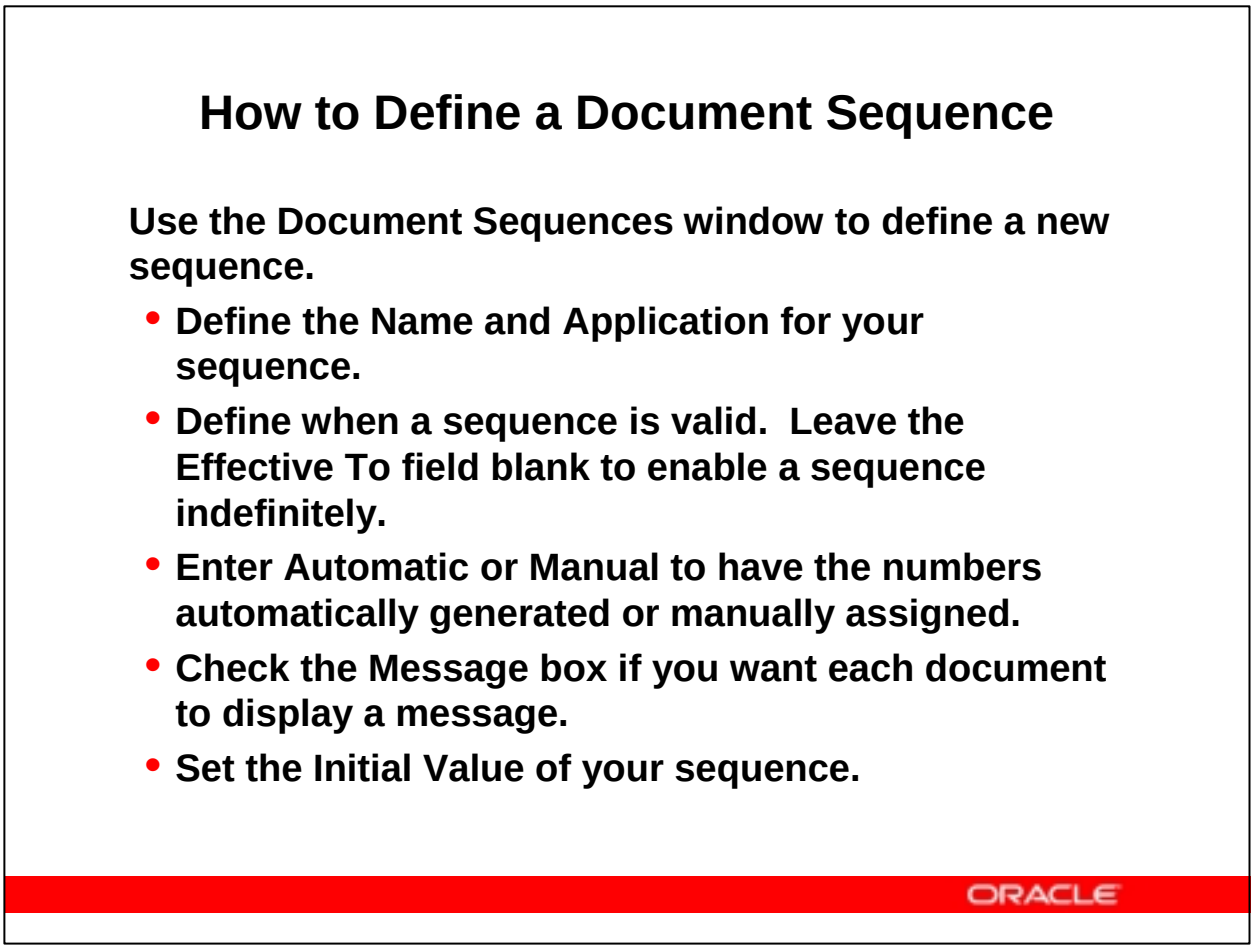
Certain information must be specified when a user creates a document sequence. This includes the document name, its effective dates, and the type of numbering used. The numbers associated with the documents can be entered manually or generated automatically.

Manual Numbering of Documents

- User assigns a unique number to each document before it is generated.
- Numerical ordering and completeness are not enforced.
- Users may skip or omit numbers when entering the sequence value.

Automatic Numbering of Documents

- Automatic numbering sequentially assigns a unique number to each document as it is generated.
- With automatic numbering, documents are numbered in order by date and time of creation.
- Numbers are in sequential order with no gaps or omissions.
- The user specifies the initial value for the sequence.
- A message can be sent to the user whenever a sequence number is assigned.



The Document Sequences Window

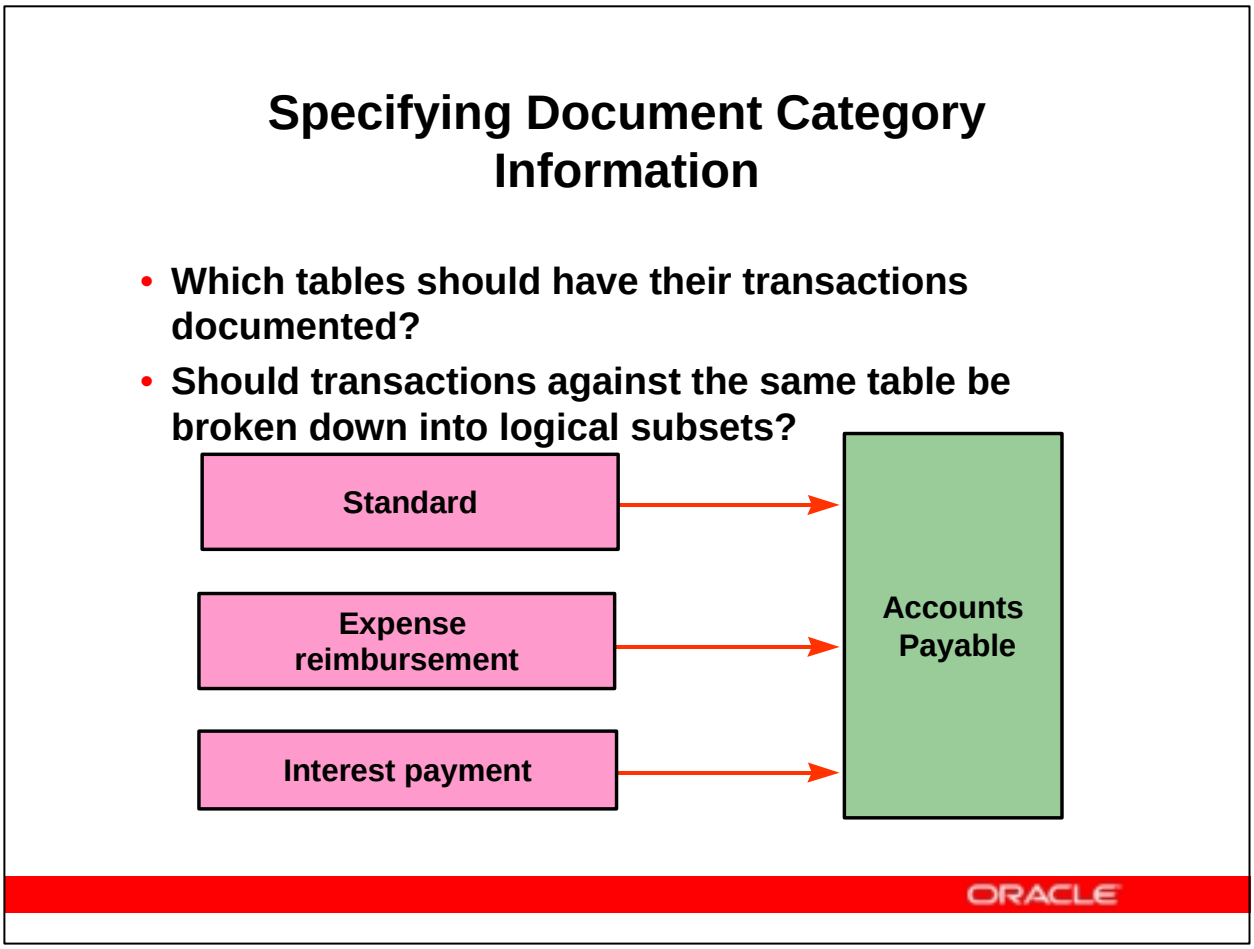
(N) Application > Document > Define

- **Name** – Once entered, the sequence name cannot be changed.
- **Application** – Once selected, the application associated with your sequence cannot be changed.
- **Effective From/To** – Enter the dates on which your document sequence takes effect/is no longer enabled. The From date automatically defaults to the current date. Once a sequence is defined, this date cannot be changed. If you leave the To field blank, your document sequence does not expire. Once you enter the To date, it cannot be modified.
- **Type** – Once defined, you cannot change the type of document numbering sequence.
  - **Automatic** – Sequentially assigns, by date and time of creation, a unique number to each document as it is generated.
  - **Manual** – Requires a user to assign a number to each document before it is generated. You must enter unique values. However, numerical ordering and completeness are not enforced.

Note: The Gapless numbering type is valid only in the context of certain localizations.

- **Message** – Check this box if you want each document to display a message (in the message line of the screen) informing the user of the sequence name and value. This applies only to sequences with the Automatic type of numbering. Once a message choice is defined, it cannot be changed.
- **Initial Value** – Enter a value for the first document in your sequence. This field applies only to sequences with Automatic or Gapless numbering type. If left blank, the first document is automatically assigned a value of 1. Once the sequence is defined, the initial value cannot be changed.

For further information about this fields in this window, see
(Help) Oracle Applications System Administration > Document Sequences >
Document Sequences Window

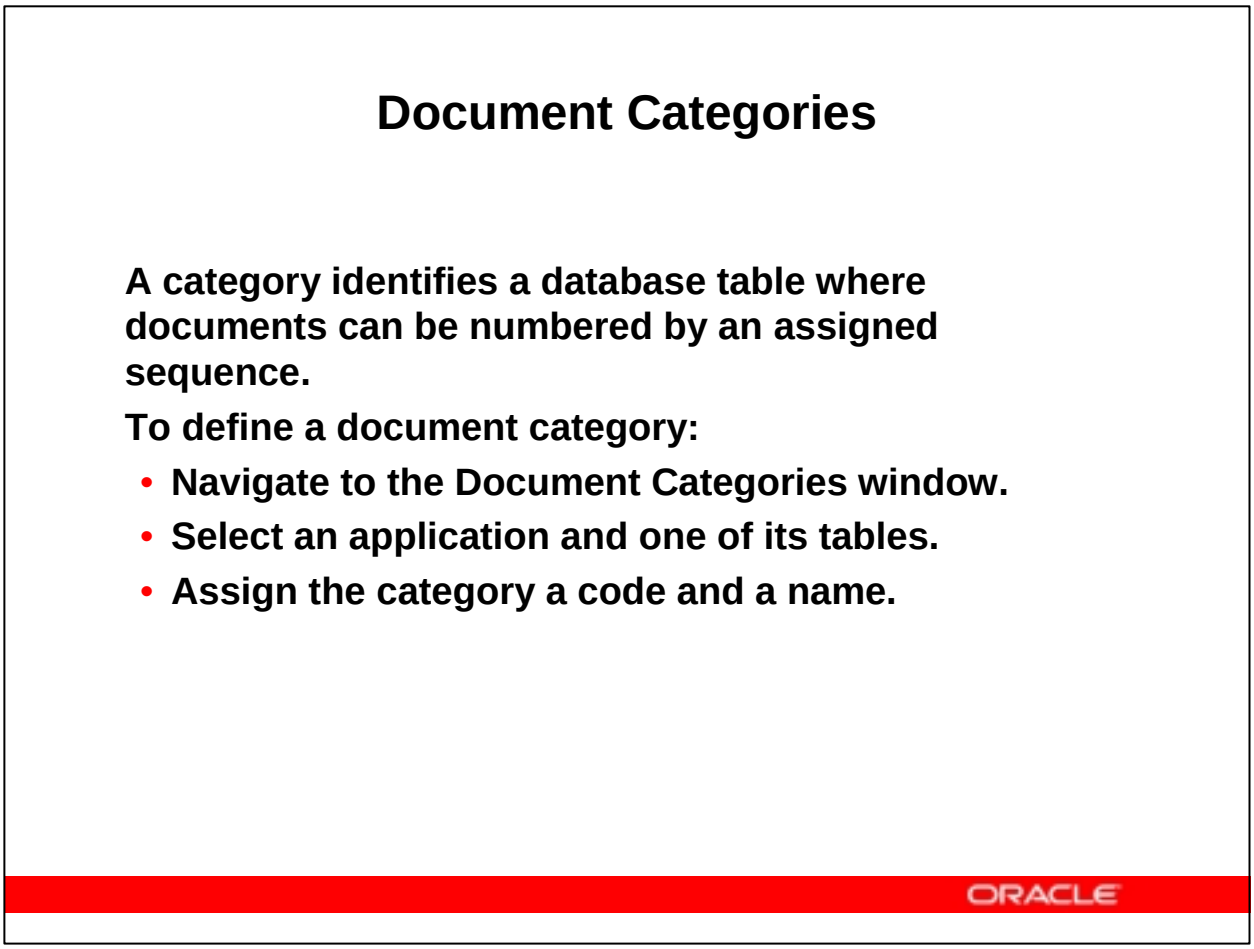


Specifying Document Category Information

Sequences track the transactions against a particular application table. The target table is identified by a document category. In some cases, transactions against a table may need to be classified into subgroups according to some criterion to obtain important information. For example, transactions against the Accounts Payable table may need to be divided according to what type of payment each transaction concerned.

Use categories to divide your documents into logical groups. For example, accounts payable invoices can be divided into the following types:

- Standard invoices
- Expense report invoices
- Prepayment invoices
- Interest invoices
- Credit memo invoices
- Debit memo invoices

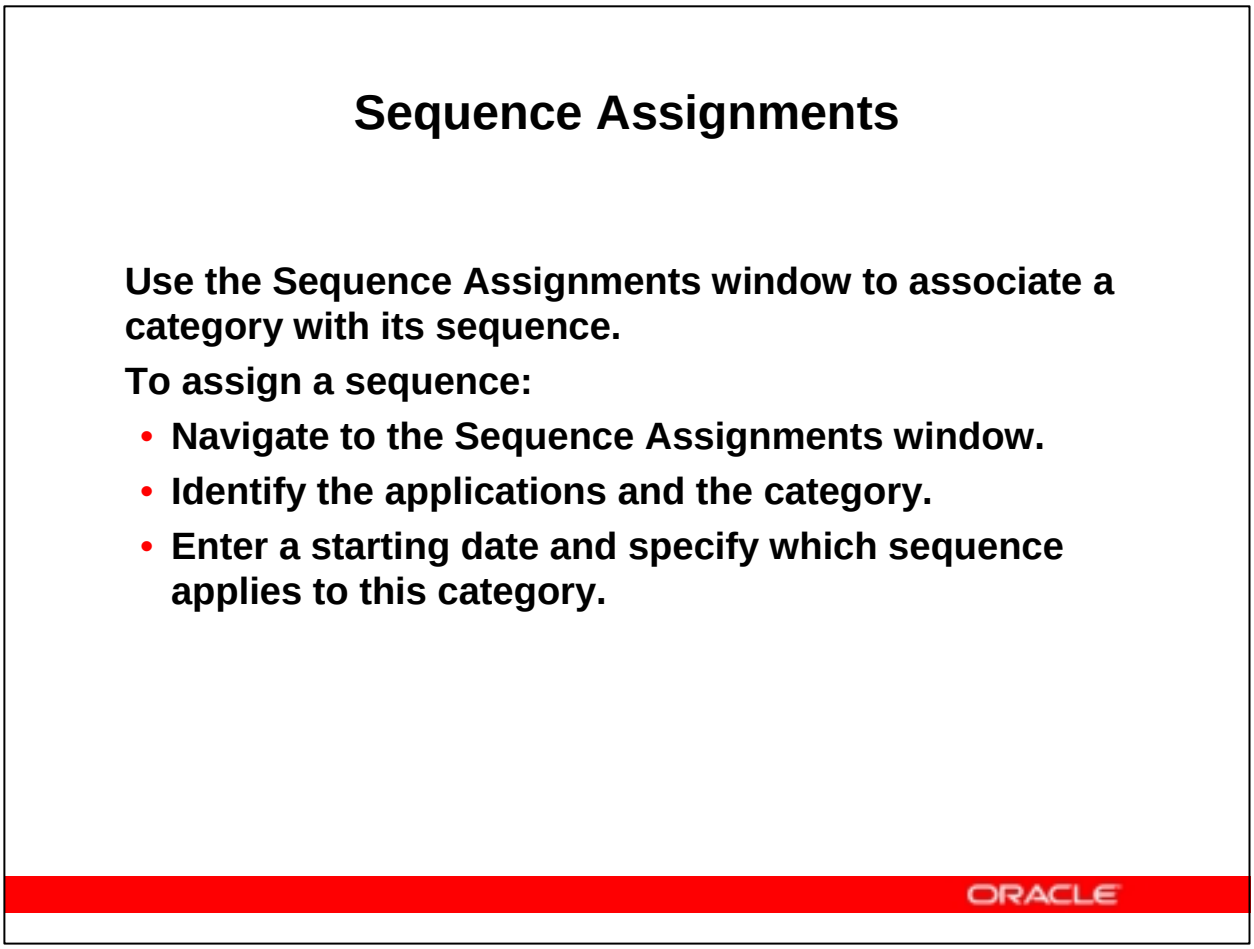


Category Code and Name

(N) Application > Document > Categories

- The combination of code and name must be unique within an application.
- Code must be unique. Name does not have to be unique.
- After a category is defined, you cannot change the choice of table.

For a complete explanation of the fields in the Document Categories window, see (Help) Applied Technology > Oracle Applications System Administration > Document Sequences > Document Categories Window.



(N) Application > Document > Assign

For a complete explanation of the fields in the Sequence Assignments window, see (Help) Applied Technology > Oracle Applications System Administration > Document Sequences > Sequence Assignments Window.

Using a Sequence in Your Application

After the document sequence has been defined, the category has been associated with a table, and the sequence has been assigned to the category, you can identify the document sequence to be used with a particular business transaction when you are creating the transaction.

Example:

(N) Invoices > Standard > Enter

ORACLE

Practice - Document Sequences

Overview

The committee wants to test Document Sequences. They have asked you to complete the following.

- Define a category
- Define a document sequence
- Assign the document sequence
- Enable sequential numbering

Tasks

Instructor Note

In order to enable "Sequential Numbering" for Oracle General Ledger, you must first make that profile updateable by the System Administrator. (This is a bug in Vision.) Go to the Application Developer Responsibility, query up the "Sequential Numbering" profile option and update the record to make it updateable by the System Administrator.

Define your Document Category

1. Use your General Ledger Super User responsibility.
2. Navigate to (N) Setup > Journal > Categories
3. Create a new document category. Remember to put your initials at the front of your data entries to keep your data unique.
   - **Category**: *Your Initials*DOCSEQ (e.g., WHSDOCSEQ)
   - **Description**: *Your Initials* Document Sequence

Define your Document Sequence

4. Navigate to (N) Setup > Financials > Sequences > Define
5. In the Document Sequences window enter the following.
   - **Name**: XXXSEQ
   - **Type**: Automatic
   - **Message**: enabled

- **Initial Value**: 1001

Assign your Document Sequence to the Category

6. Navigate to (N) Setup > Financials > Sequences > Assign

7. In the Sequence Assignments window enter the following.

   - **Application**: Oracle General Ledger
   - **Category**: *Your Initials*DOCSEQ (e.g., WHSDOCSEQ)
   - **Set of Books**: Vision Operations
   - **Method**: Manual
   - **Start Date**: 01-JAN-1950
   - **End Date**: (leave blank)
   - **Sequence**: *Your Initials*SEQ (e.g., WHSSEQ)

Enable Sequential Numbering for Oracle General Ledger

8. Switch responsibilities to the System Administrator.

9. Set the profile option “Sequential Numbering” to “Partially Used” for the application General Ledger. **Note**: Once one student sets this value, it will be set for everyone.

10. Switch back to the General Ledger Super User responsibility.

11. Navigate to (N) Journals > Enter.

12. In the Find Journals window, select New Journal.

13. In the Journals window, change the value in the category field to *Your Initials*DOCSEQ. A message will display indicating that the number 1001 has been assigned.

Solution – Document Sequences

Define your Document Category

Responsibility = General Ledger Super User

1. If necessary, log in to Oracle Applications and select the General Ledger Super User responsibility.

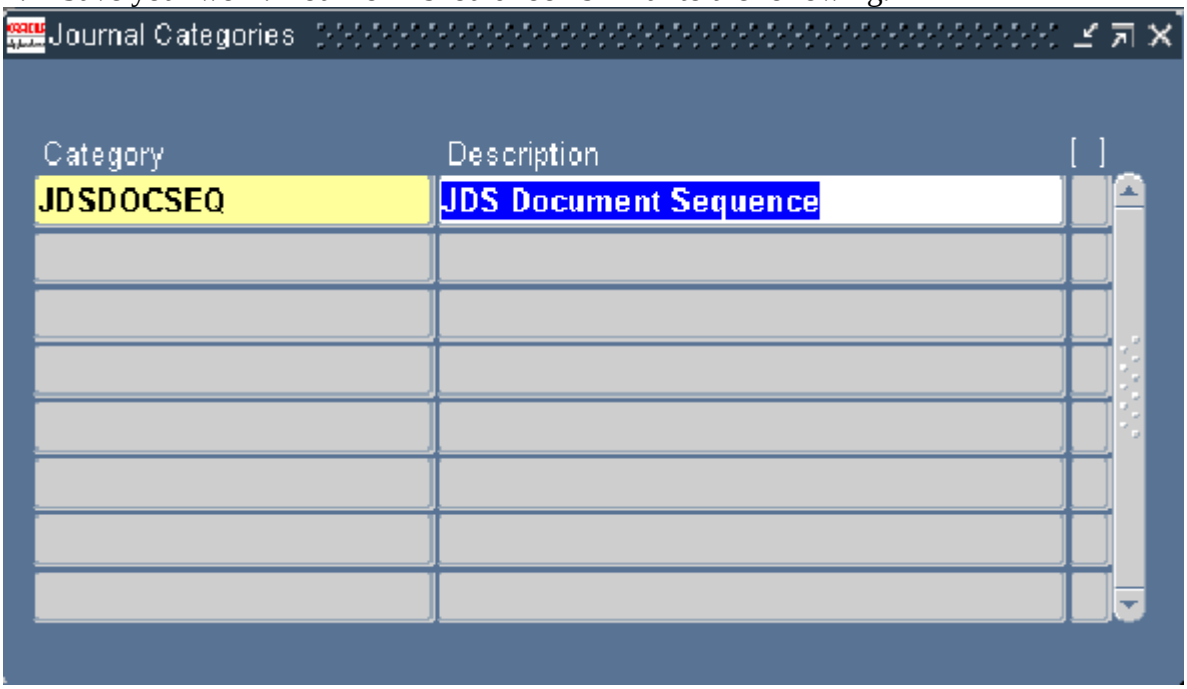
2. Navigate to (N) Setup > Journal > Categories

3. Create a new document category. Remember to put your initials at the front of your data entries to keep your data unique.

   - **Category**: *Your Initials*DOCSEQ (e.g., WHSDOCSEQ)

   - **Description**: *Your Initials* Document Sequence

4. Save your work. Your form should look similar to the following.



5. Close the form.

Define your Document Sequence

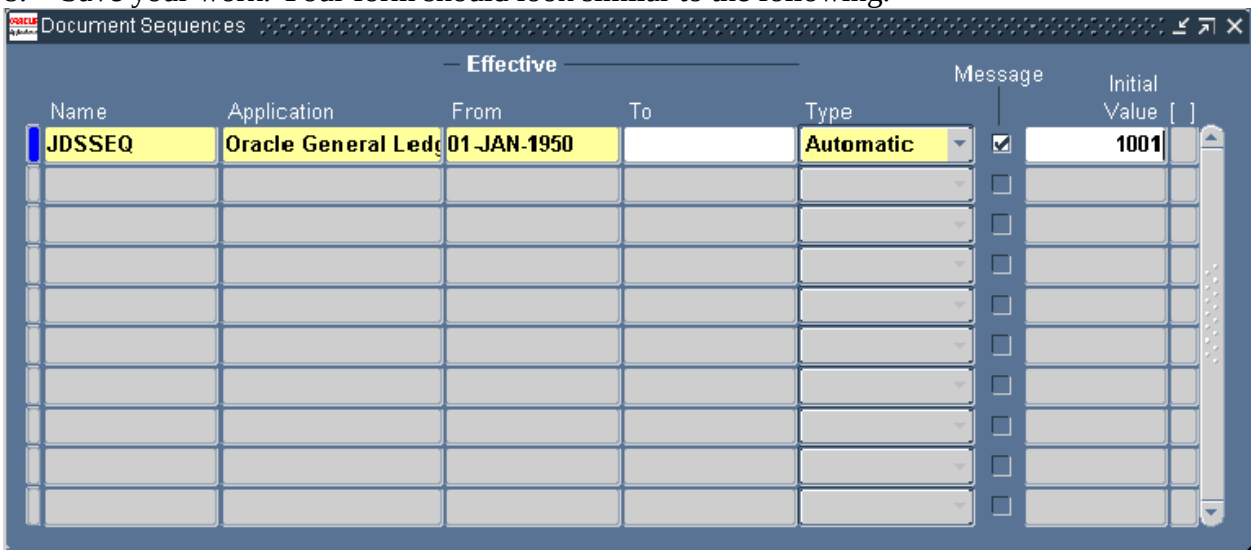
6. Navigate to (N) Setup > Financials > Sequences > Define

7. In the Document Sequences window enter the following.

   - **Name**: XXXSEQ

- **Type**: Automatic
- **Message**: enabled
- **Initial Value**: 1001

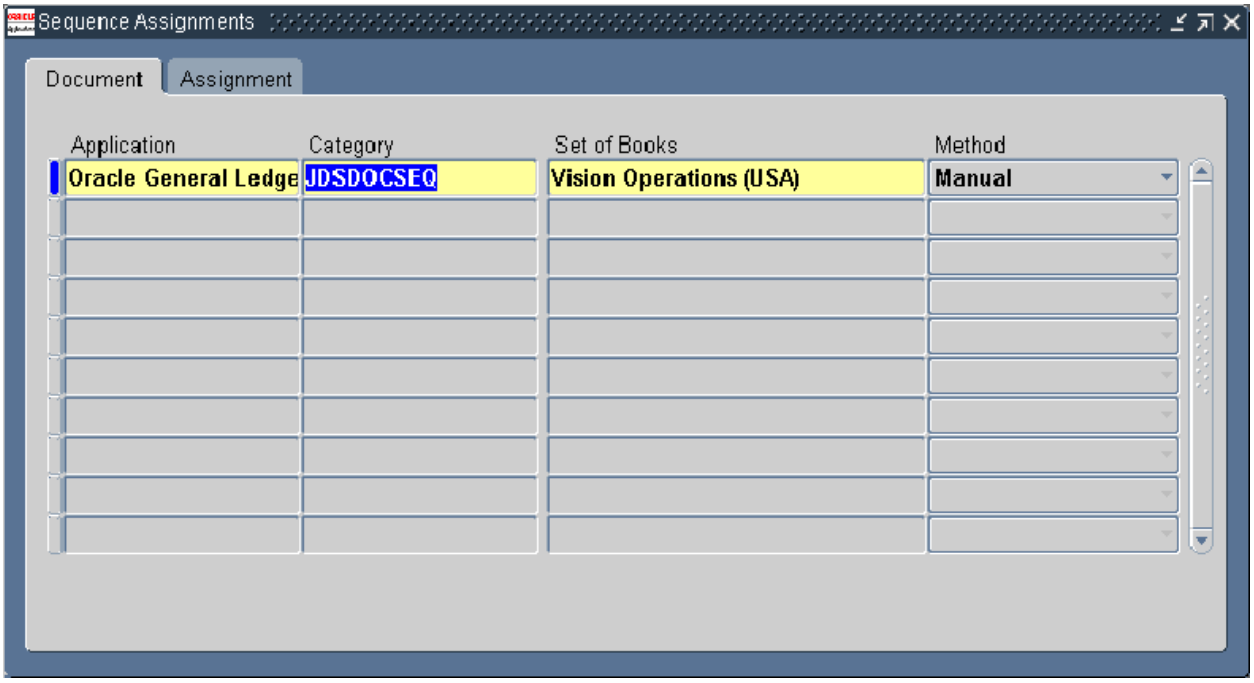
8. Save your work. Your form should look similar to the following.

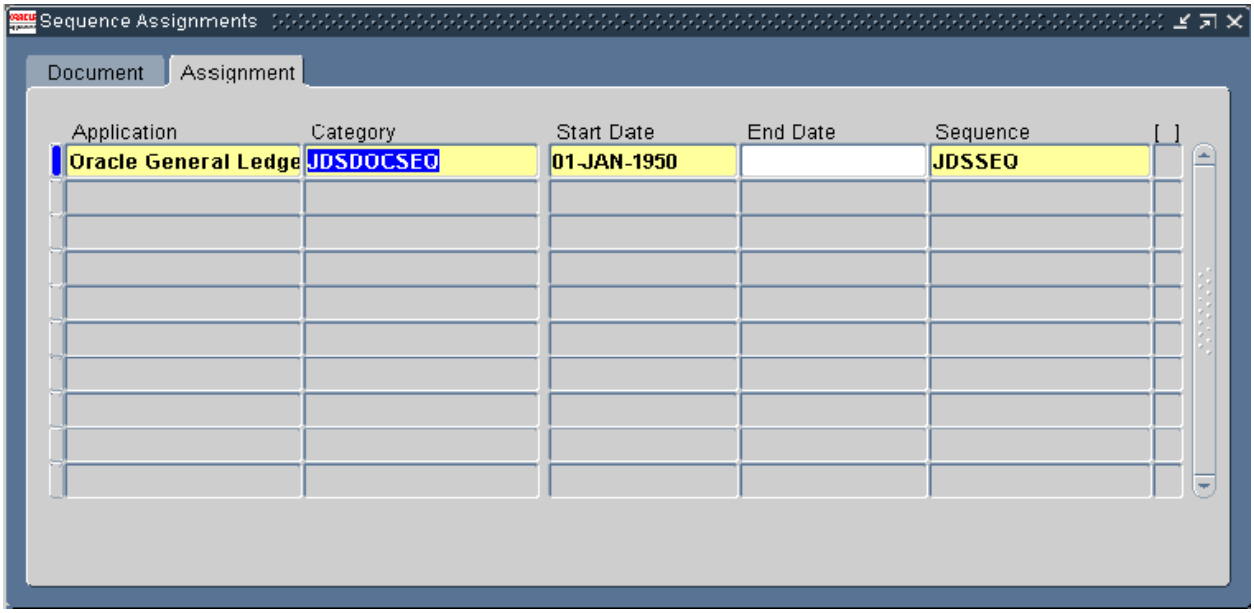


9. Close the form.

Assign your Document Sequence to the Category

10. Navigate to (N) Setup > Financials > Sequences > Assign
11. In the Sequence Assignments window enter the following.
    - **Application**: Oracle General Ledger
    - **Category**: *Your Initials*DOCSEQ (e.g., WHSDOCSEQ)
    - **Set of Books**: Vision Operations
    - **Method**: Manual
    - **Start Date**: 01-JAN-1950
    - **End Date**: (leave blank)
    - **Sequence**: *Your Initials*SEQ (e.g., WHSSEQ)
12. Save your work. Your form should look similar to the following.





13. Close the form.

Enable Sequential Numbering for Oracle General Ledger

14. Switch responsibilities to the System Administrator.

15. Navigate to (N) Profile > System.

16. Query the “Sequential Numbering” profile option.

17. Set the profile option “Sequential Numbering” to “Partially Used” for the application General Ledger. **Note**: Once one student sets this value, it will be set for everyone.\

18. Switch back to the General Ledger Super User responsibility.

19. Navigate to (N) Journals > Enter.

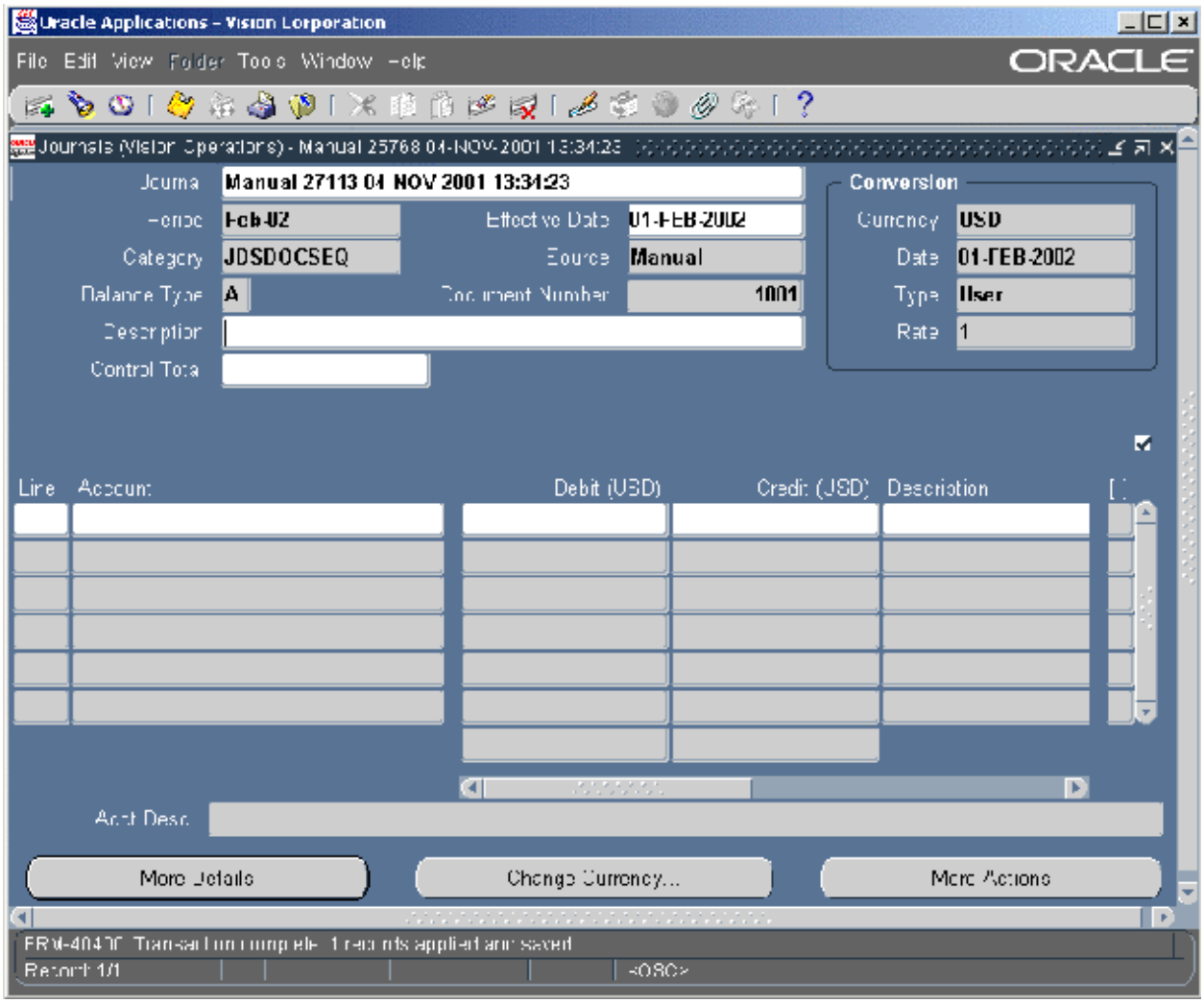
20. In the Find Journals window, click on the “New Journal” button.

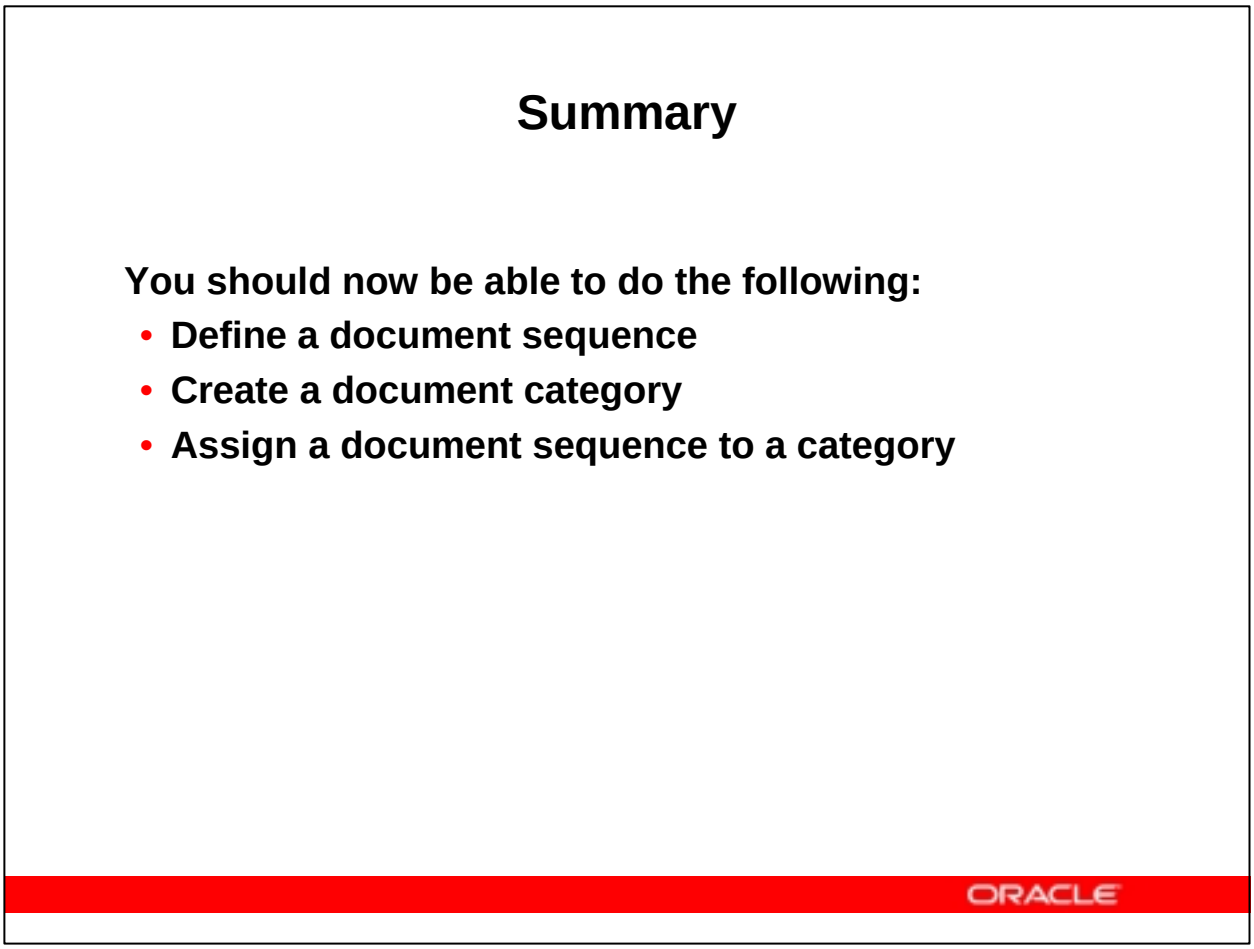
21. A message will display indicating that no sequence exists for this category. That is because the default category (set by a profile option called Journals: Default Category) is Adjustment, for which no sequence has been defined.

22. Change the value in the category field to *Your Initials*DOCSEQ.

23. Save your work.

24. A message will display stating that the number 1001 has been assigned. **Note** that on the Journals window the Document Number field has been filled with 1001.



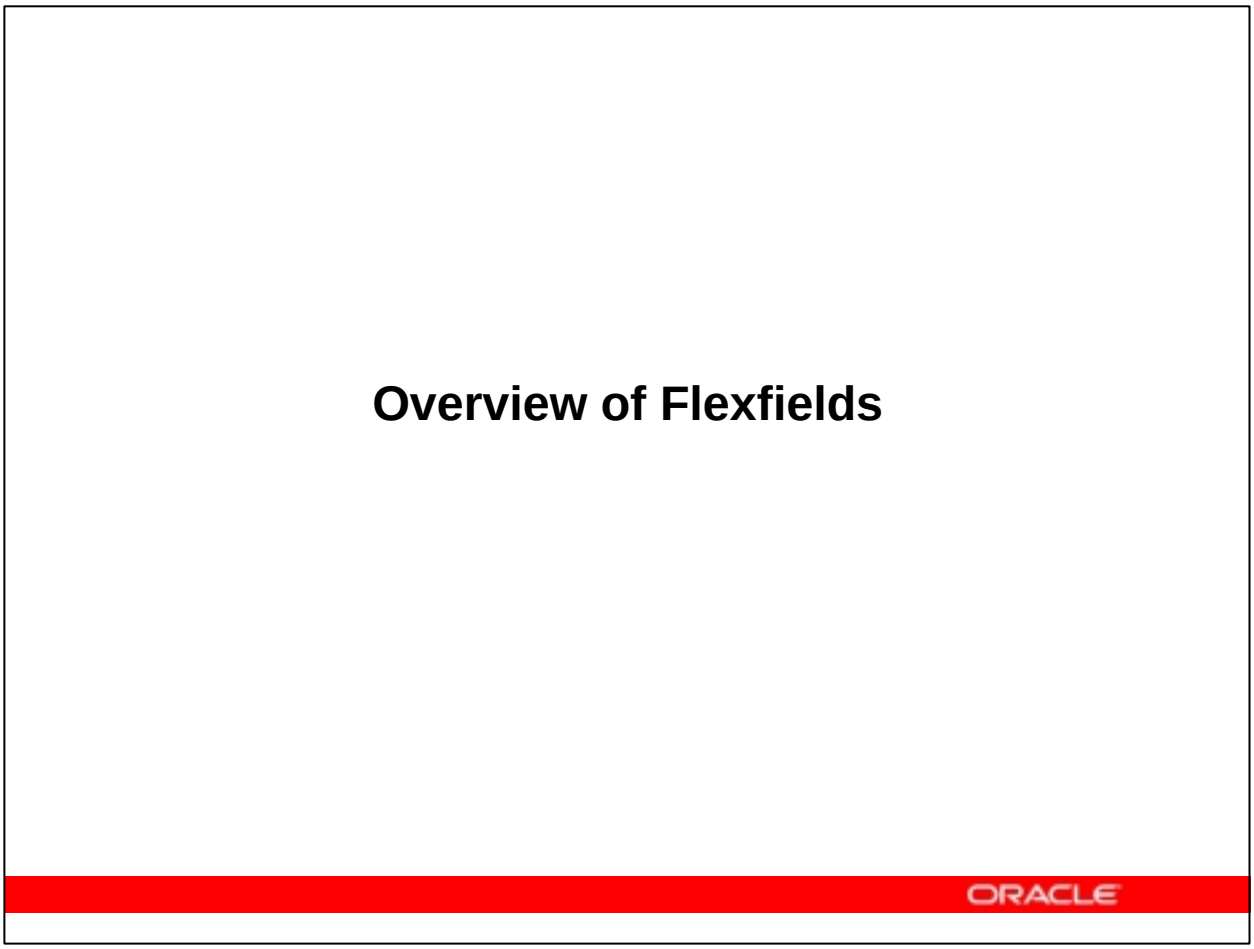


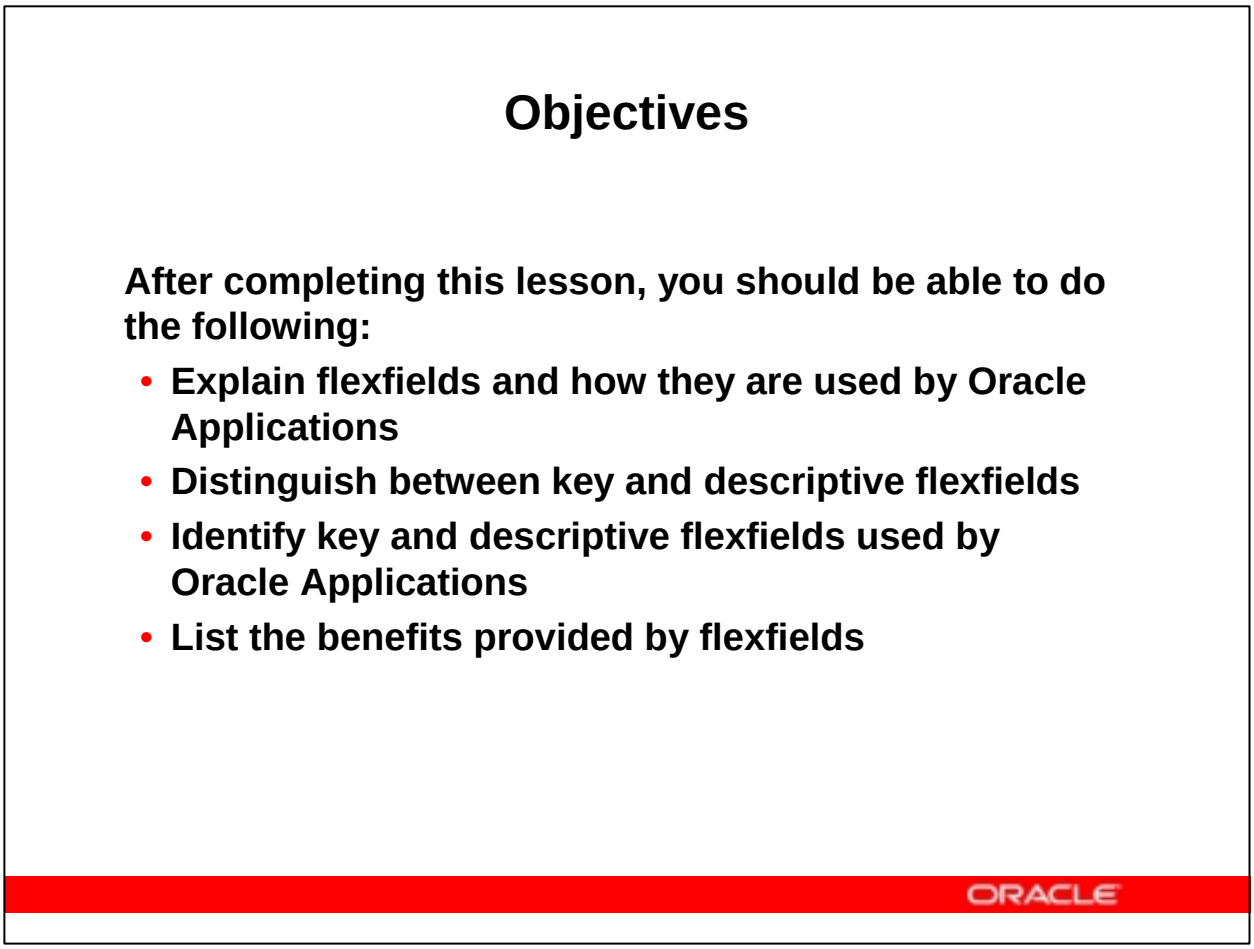
Summary

Document sequences can be used to track all transactions against a table regardless of transaction completion. These sequences can provide an audit trail of transactions against a table. A particular table can have multiple categories defined to it. In this way, different types of transactions can all be tracked.

Overview of Flexfields

Chapter 13





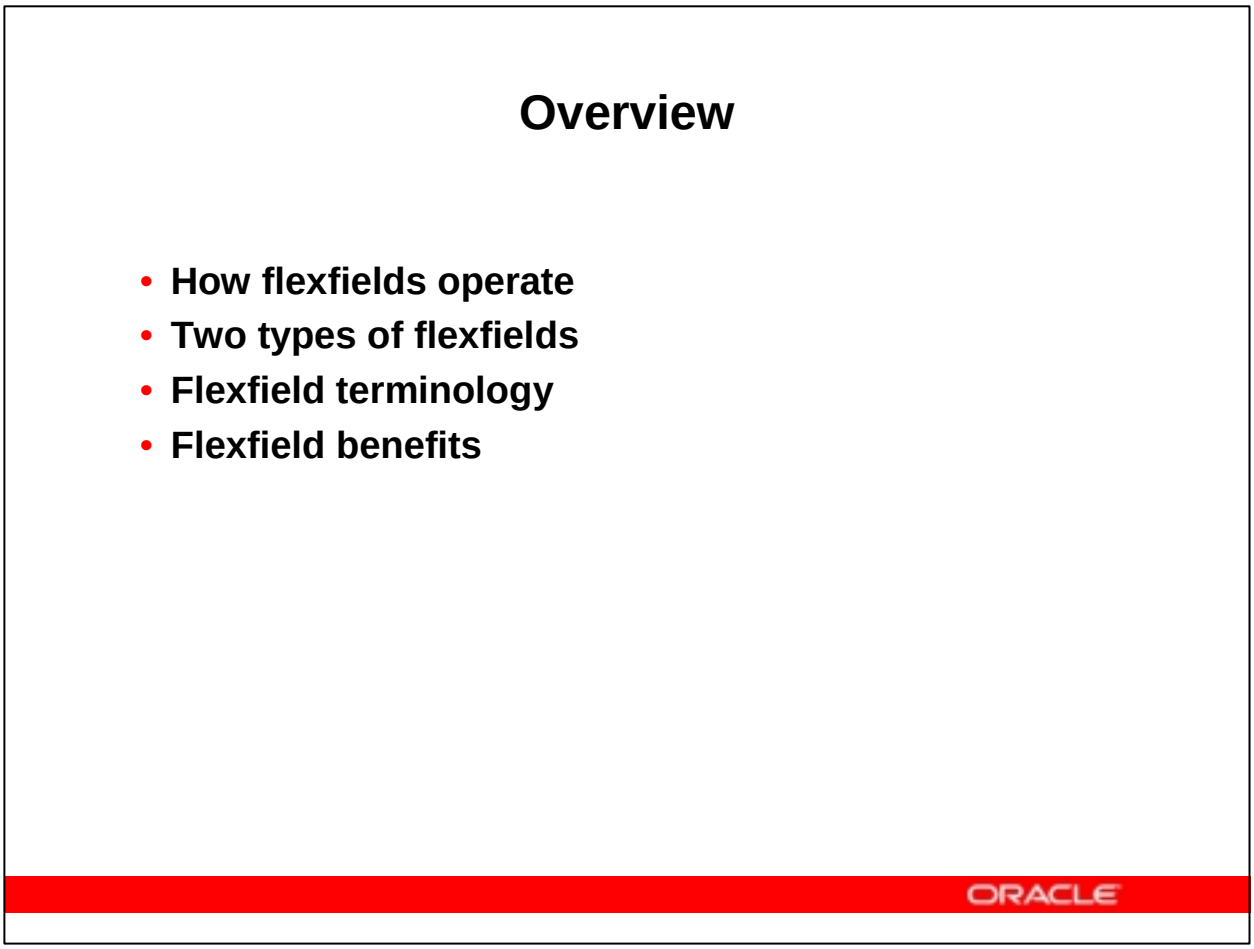
Lesson Topics

This lesson describes how to use modifiable fields called flexfields to tailor Oracle Applications to reflect your business needs and practices.

You can use flexfields to define the structure of certain data identifiers required by Oracle Applications. You can also gather, store and process additional data that may be required by your business.

After completing this lesson, you should be able to:

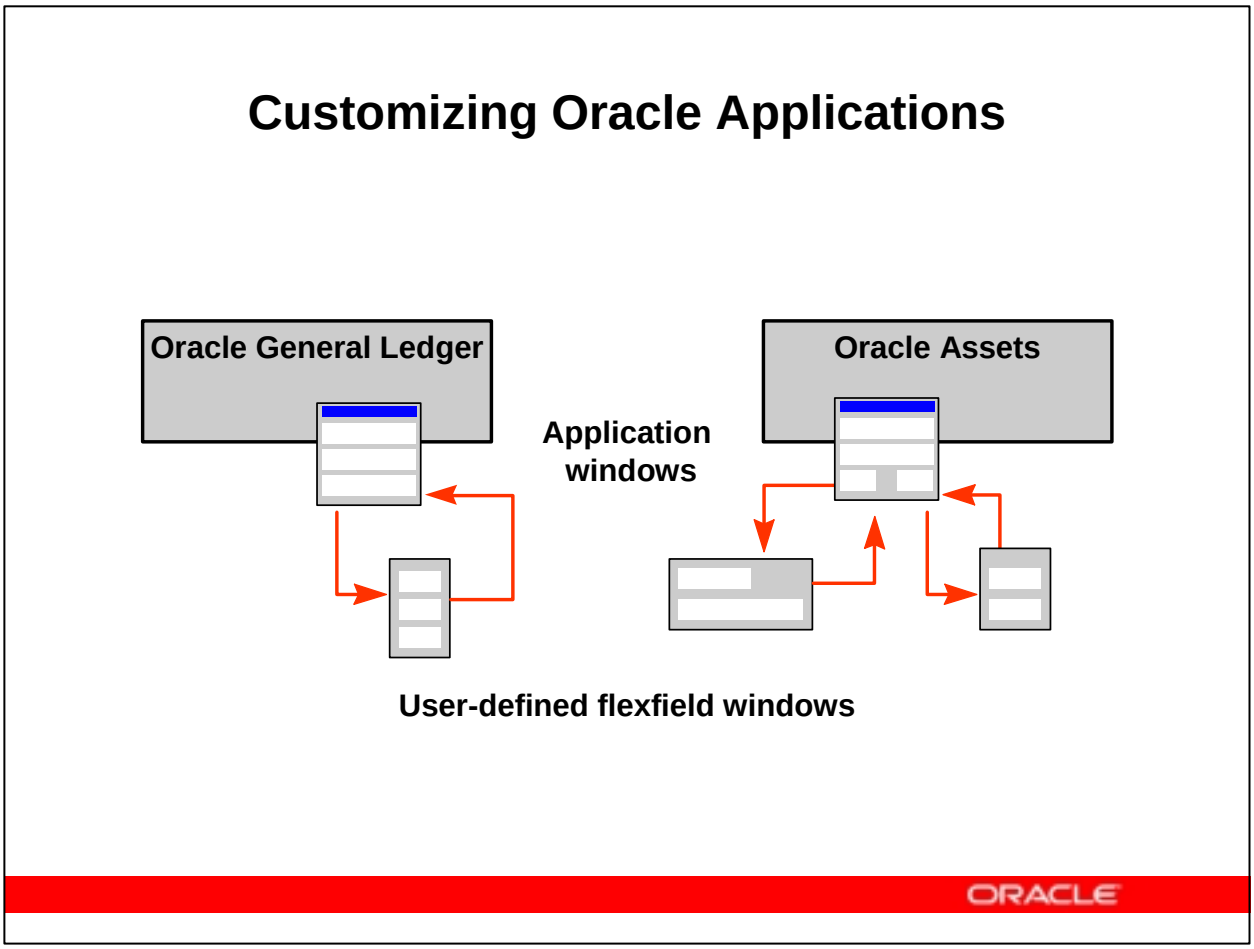
- Explain flexfields and how they are used by Oracle Applications
- Distinguish between key and descriptive flexfields
- Identify key and descriptive flexfields used by Oracle Applications.
- List the benefits provided by flexfields



Lesson Overview

This lesson describes how Oracle Applications uses customizable fields called flexfields to modify and expand processing. Specifically it covers:

- How flexfields operate
- The two types of flexfields
- Flexfield terminology
- Flexfield benefits

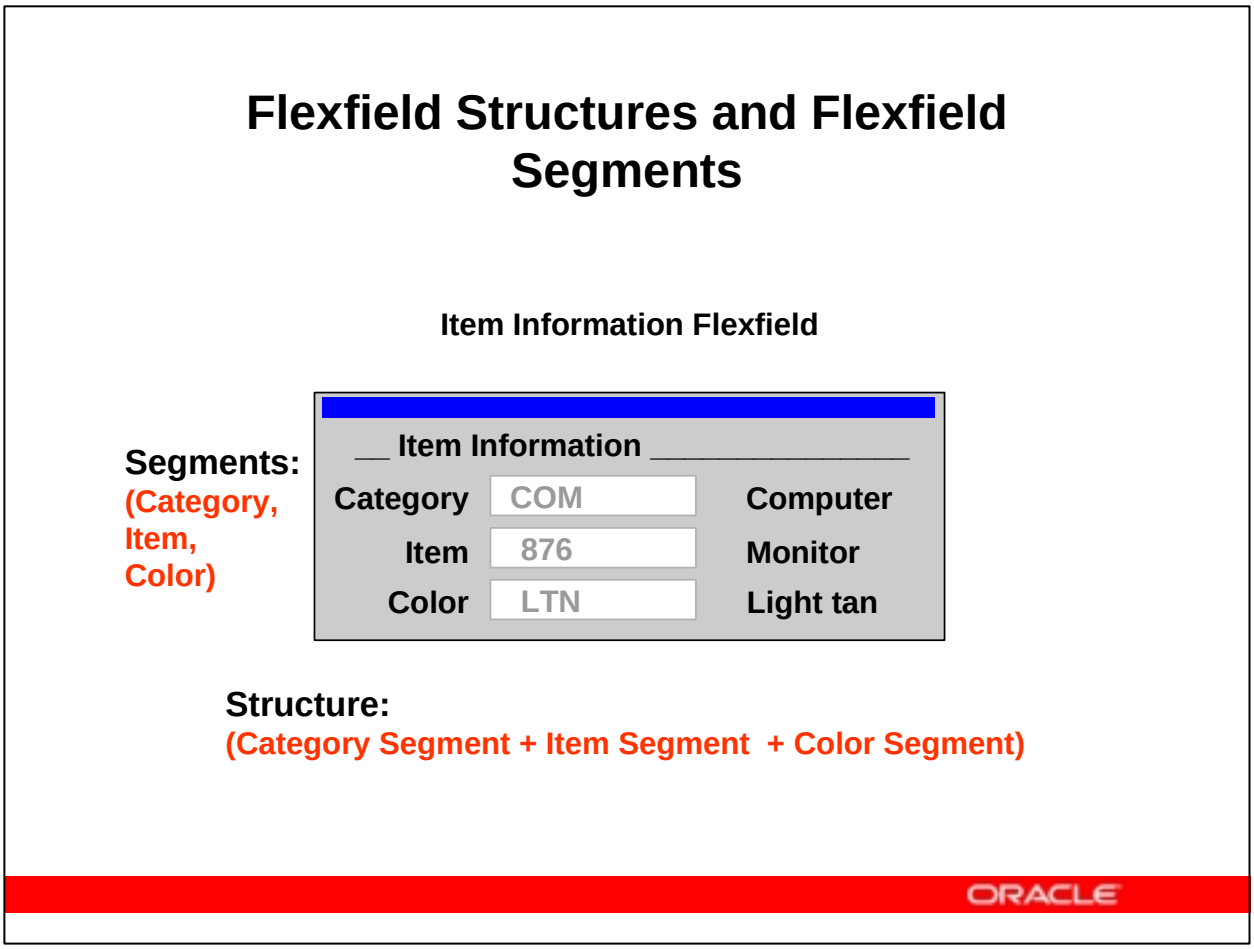


Using Flexfields to Customize Applications

A flexfield is a customizable field that opens in a window from a regular Oracle Applications window. Defining flexfields enables you to tailor Oracle Applications to your own business environment. You can easily define flexfields to modify or extend Oracle Applications without programming. By using flexfields within Oracle Applications, you can:

- Structure certain identifiers required by Oracle Applications according to your own business environment
- Collect and display additional information for your business as needed

Flexfields are important because they are used throughout Oracle Applications. Flexfields provide many opportunities for easy modification and customization of standard Oracle Applications processing. Members of the system implementation team as well as system administration personnel should be familiar with the concepts and procedures of flexfields to design and support an Oracle Applications environment that meets the needs of all its users.



Flexfield Components

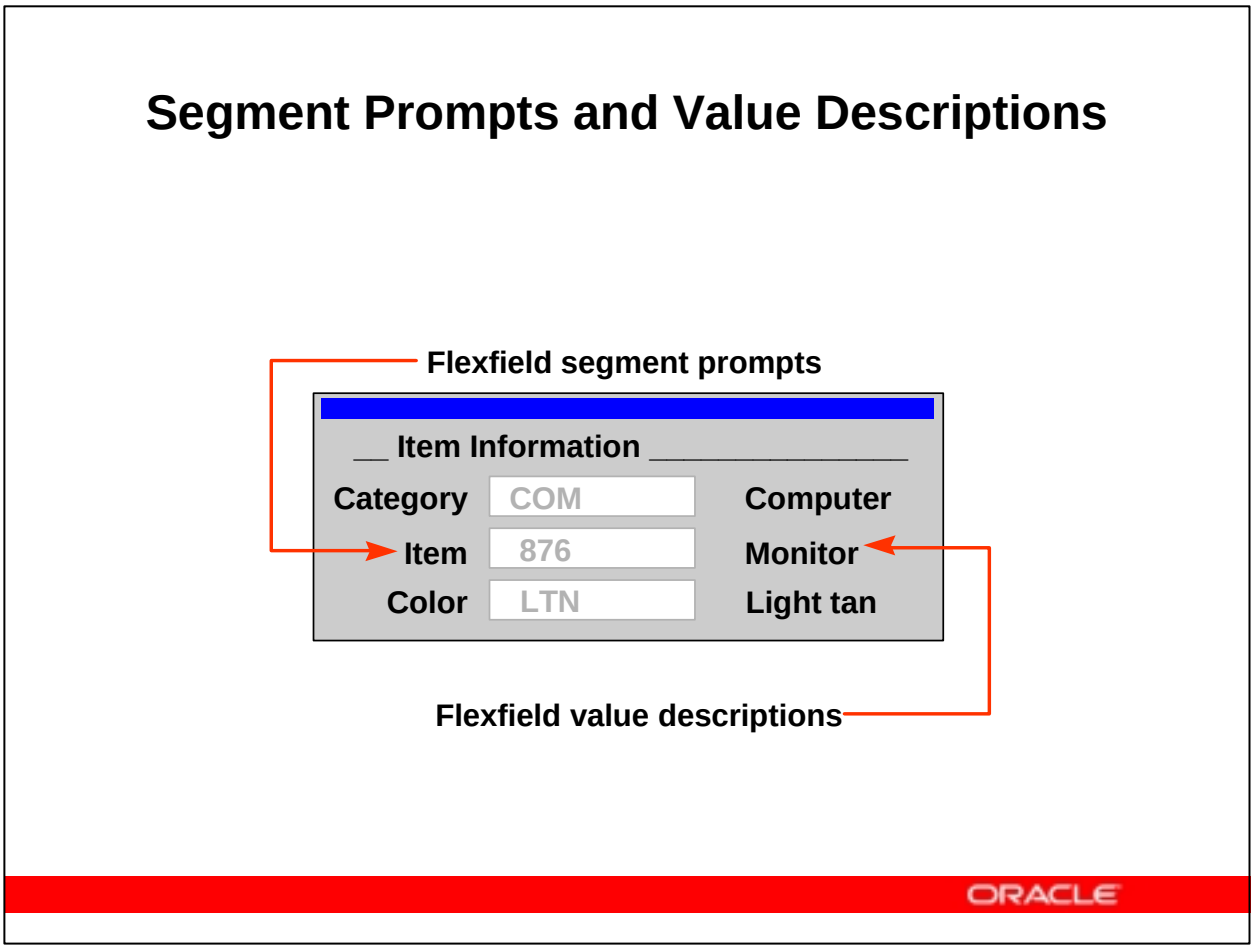
A flexfield is composed of segments. The segments of a flexfield contain the flexfield values. Flexfield segments are combined in a grouping called a structure. The segments that make up a particular structure are logically or functionally related. A single flexfield can have more than one structure.

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:

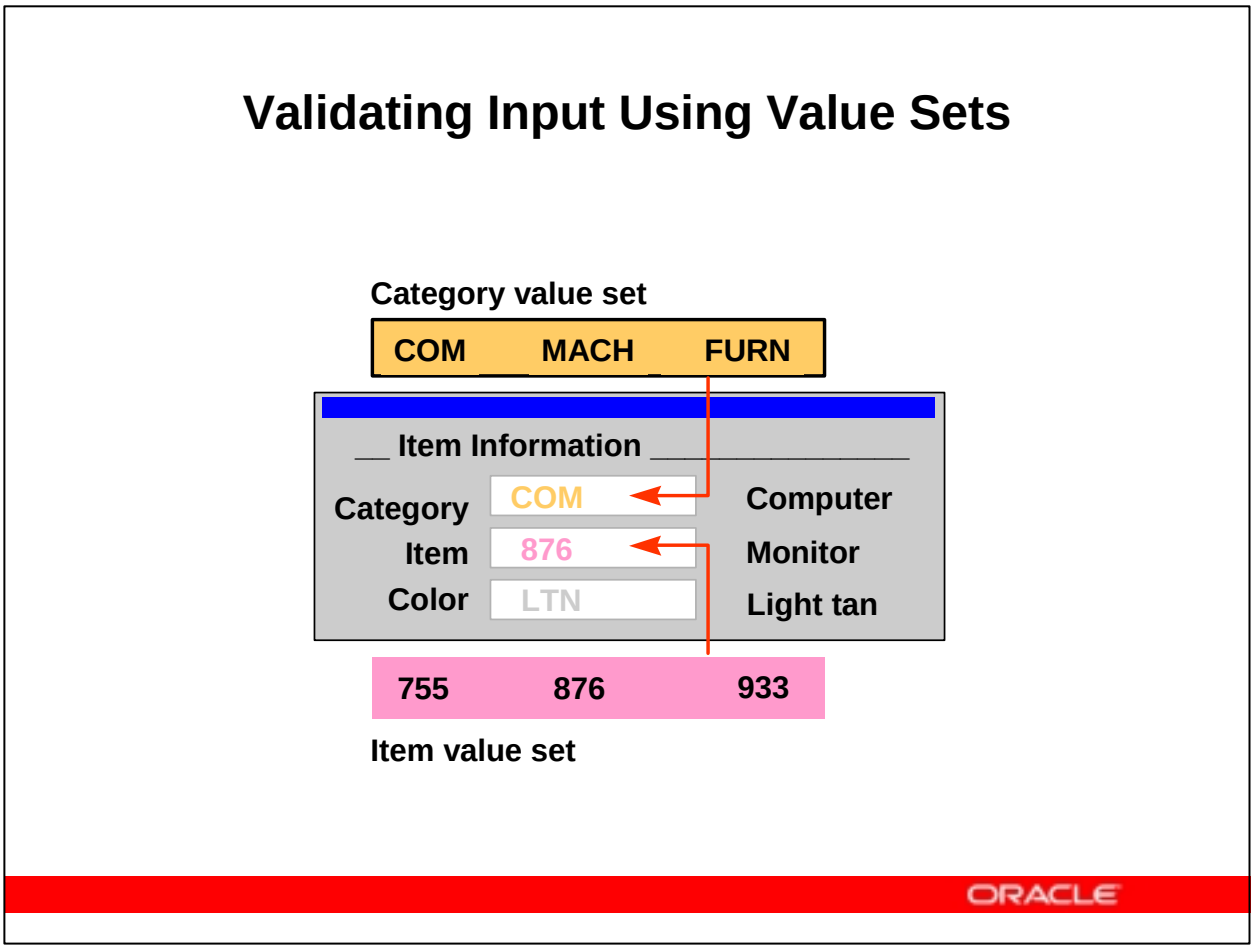
- Category
- Item
- Color

These three segments together form a structure.



Customizing the Flexfield Appearance

In addition to defining the segments and structures that make up the flexfield, you can define the appearance of the flexfield. You can specify names and descriptions for the segments appearing on the window. You can also specify the display size for the fields.

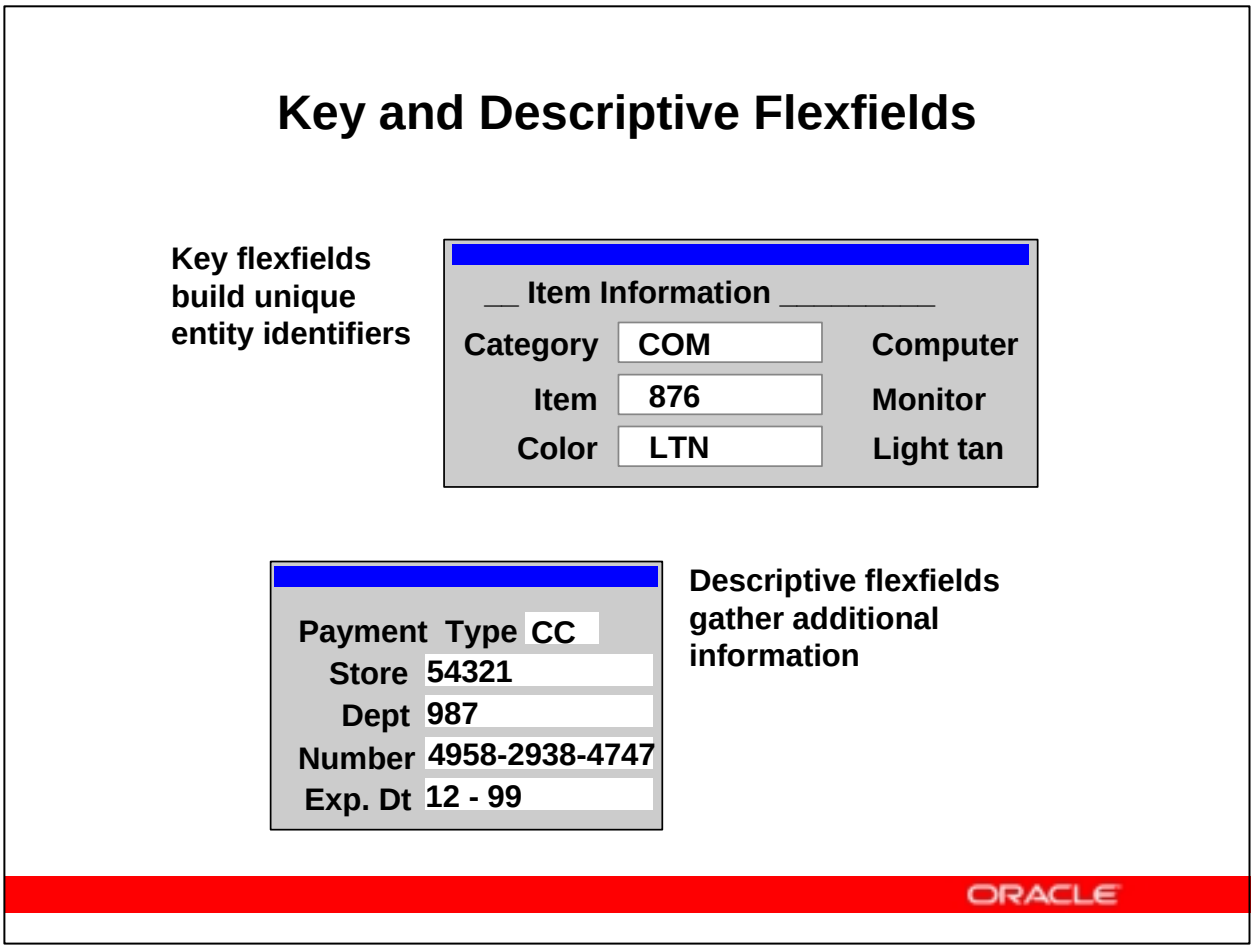


Using Value Sets with Flexfields

Value sets allow you to control the values for a segment or a report parameter. A value set is a definition of the values approved for entry or display by a particular flexfield segment. A value set may also contain a list of actual approved values although this is not required. You may be able to define the appropriate values after you define the value set.

Value Sets Describe Acceptable Types of Values

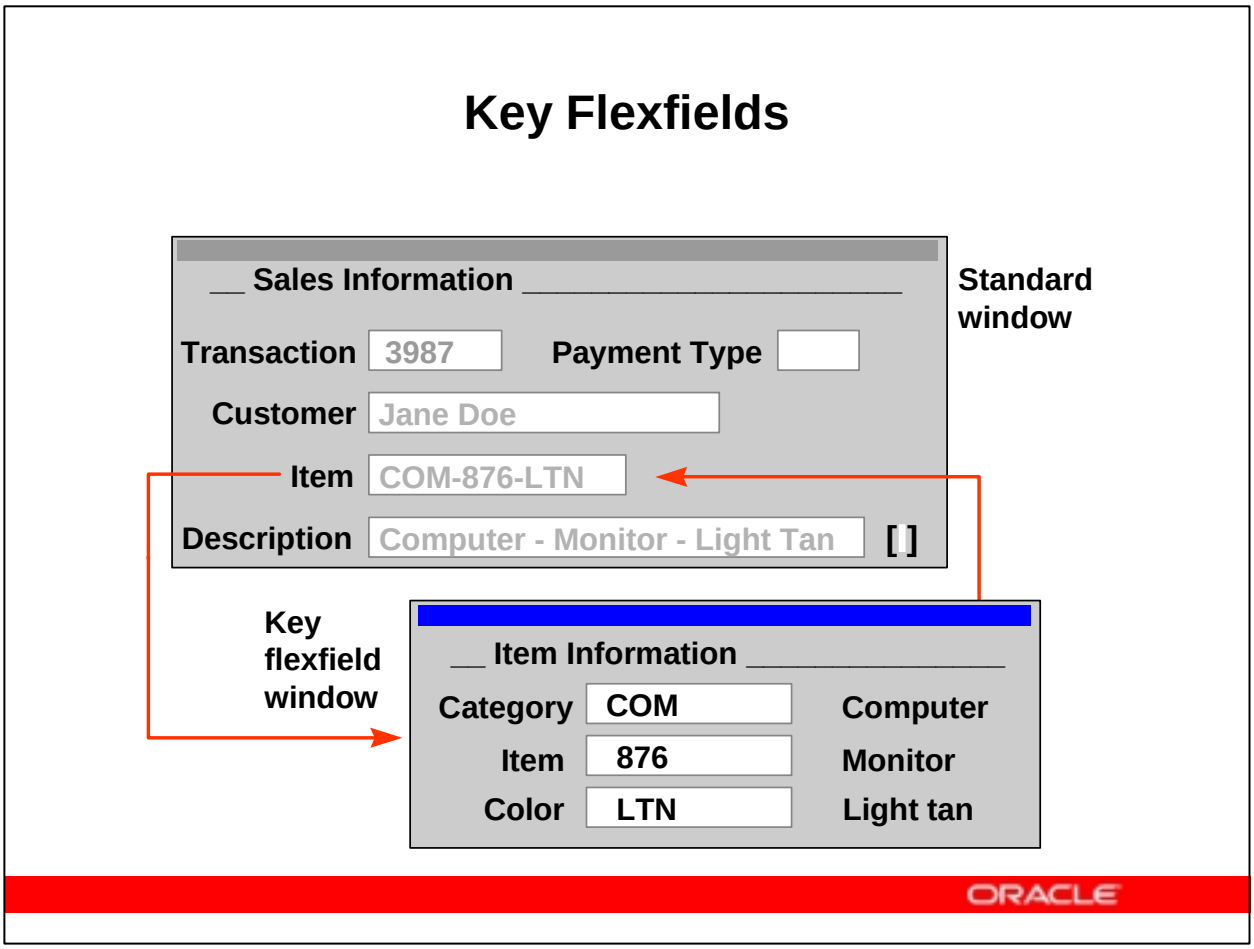
- Some value sets permit a limited range of values; others permit only certain values; others have minimal restrictions.
- Different flexfields can share the same value set. For example, a value set containing the names of regional offices could be used by many different flexfields.
- Different segments of the same flexfield can use the same value set, for example a date value set. Segments defined to different structures of the same flexfield can share value set. Many of the report parameters used with SRS forms are tied to shared value sets.
- Value sets do not have to have the actual values defined for them.



Different Types of Flexfields

There are two types of flexfields, key flexfields and descriptive flexfields. Each type is discussed in more detail on the following slides. The main differences are:

- 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.



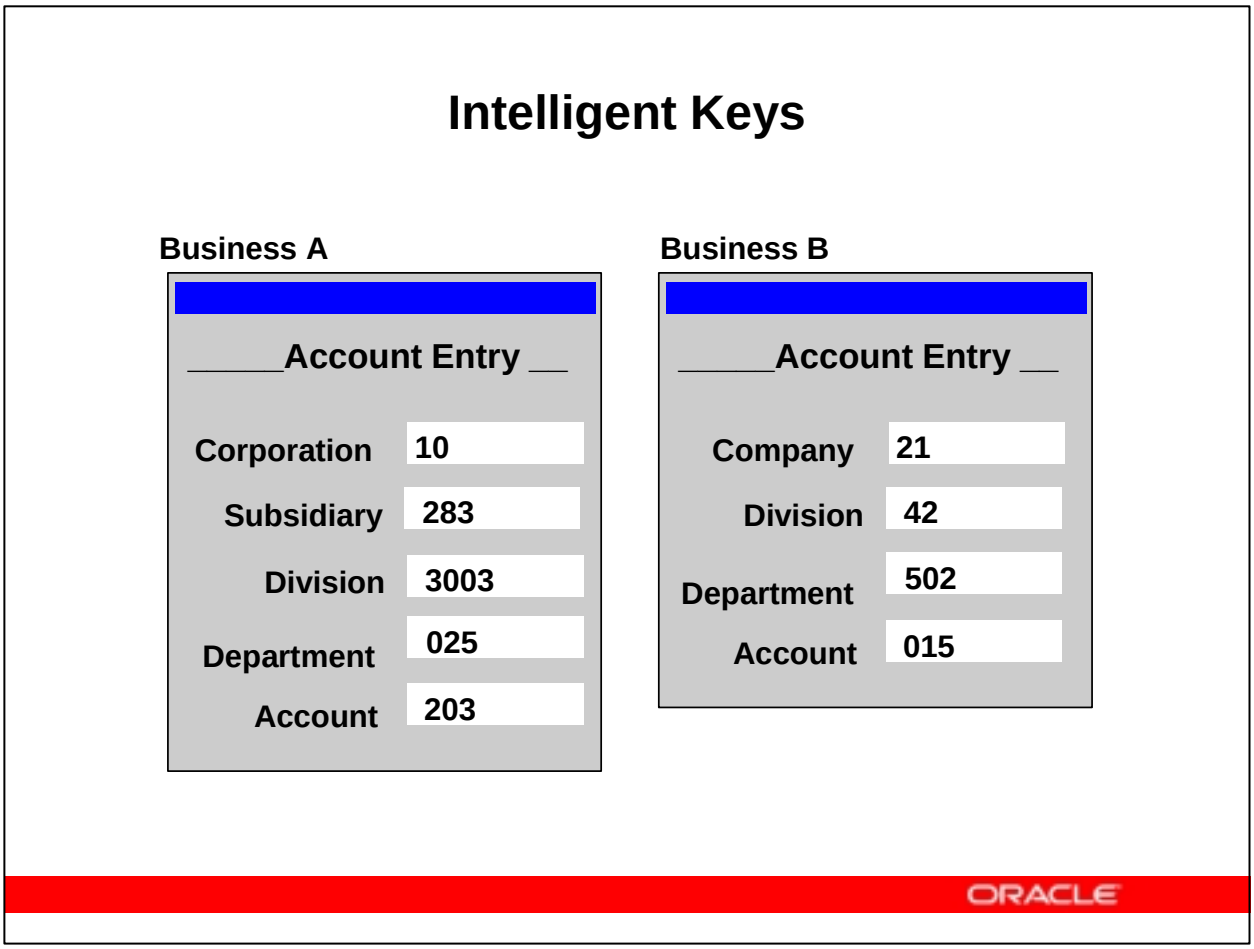
Key Flexfields

In Oracle Applications you use key flexfields as identifiers for entities. Generally, the identifier you create using a key flexfield is required by the application (for example, the Accounting Flexfield builds the account number used by Oracle General Ledger).

A key flexfield appears as a normal field on a form. Any existing value for the key appears in the field as a concatenated value having segment separators.

You can use the Flexfields: Open Key Window profile option to specify whether you want the key flexfield window to be opened automatically when you navigate to the key flexfield on the base form. This profile option is visible and updatable at the user level.

A key flexfield structure usually consists of multiple segments, each of which contains meaningful information. The resulting combinations of values from these segments therefore function as intelligent keys.



Using Key Flexfields to Build Intelligent Keys

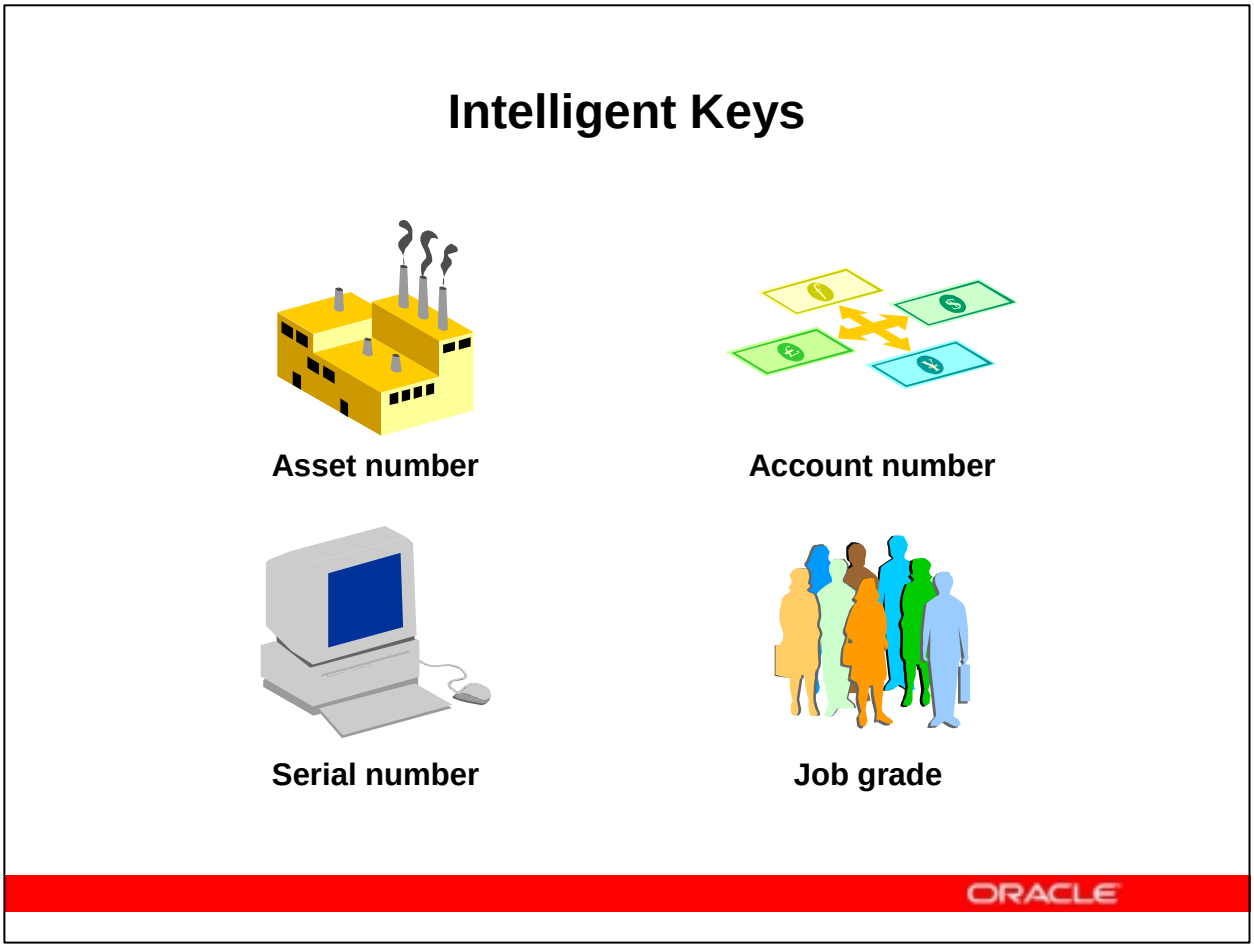
Intelligent keys are multipart codes where the value in each individual part contains meaningful information. Each combination of values can therefore identify a particular business entity or class of entities. Such intelligent key structures are used throughout Oracle Applications. Because you define your key flexfield structure yourself, it reflects the organization of your business.

Accounting Flexfield Example

One important key flexfield is the Accounting Flexfield. The example on the slide shows how two hypothetical businesses could define Accounting Flexfield structures that reflected their different accounting structures. Business A's accounting structure has five segments, while business B's accounting structure has four segments.

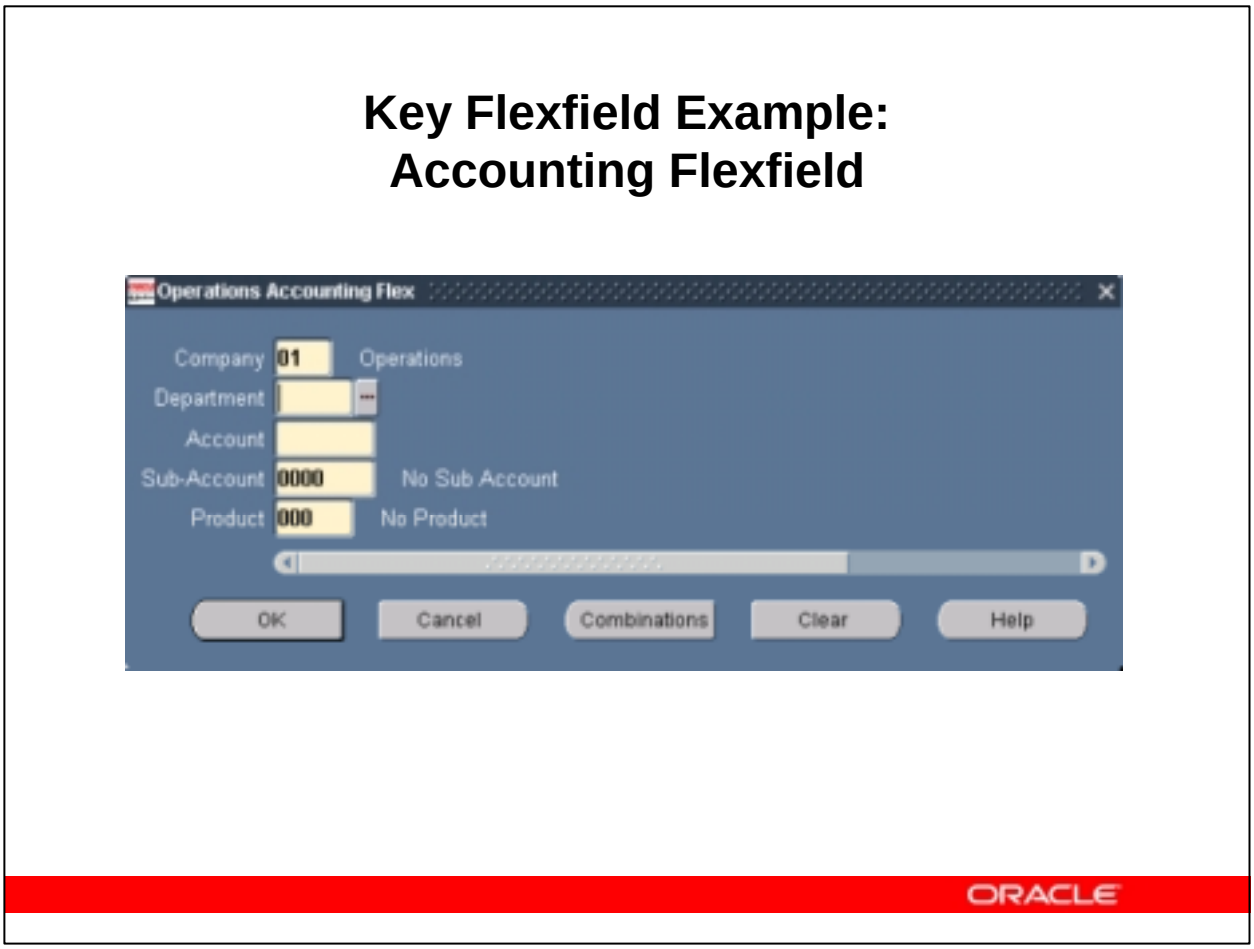
Additional Key Flexfield Features

You can define value sets to control the allowable values for each segment of the key. You can also define cross-validation rules to control the allowable combinations of segment values within the key.



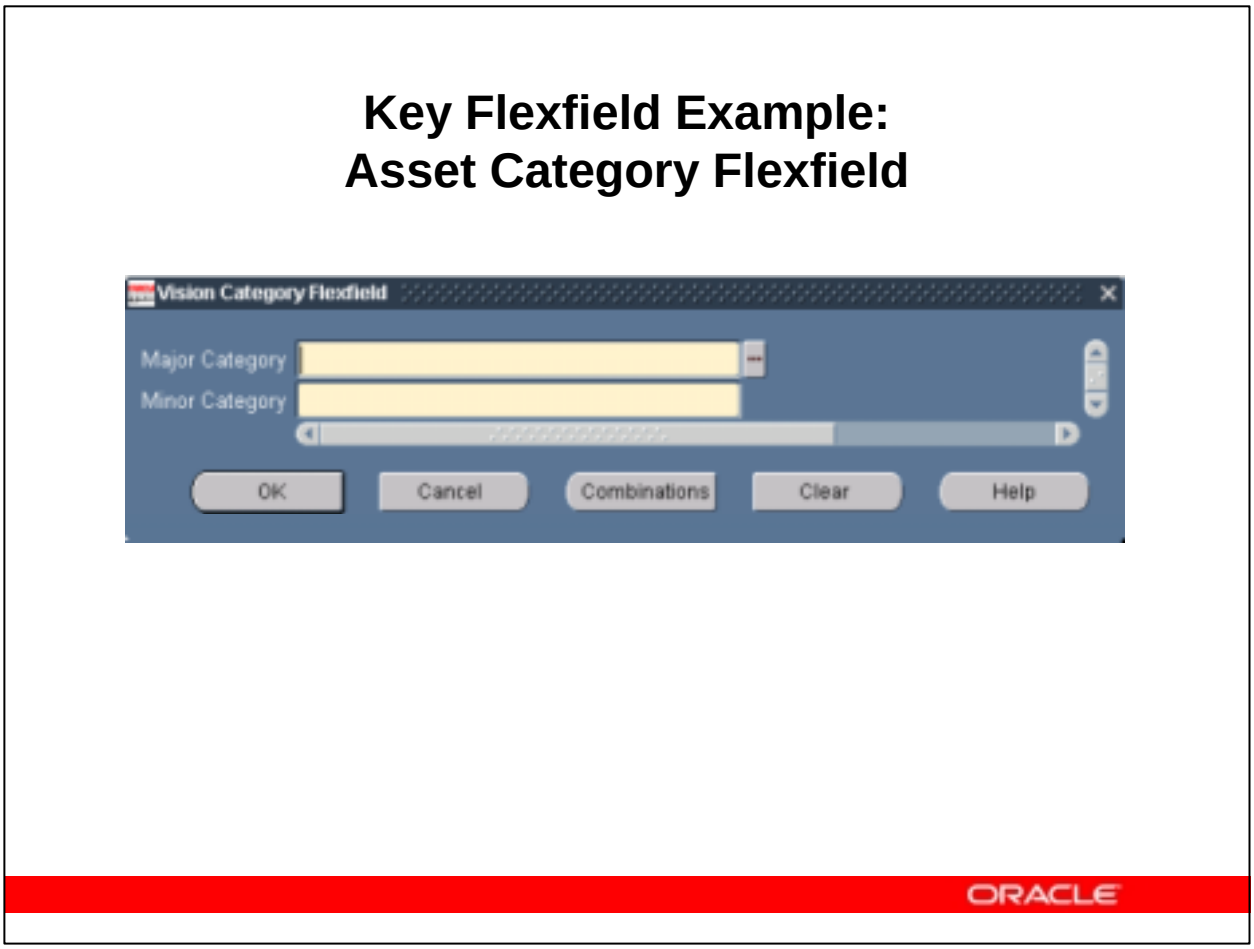
Intelligent Key Examples

There are many areas in business where you can use intelligent keys. Some examples are shown. Many such intelligent keys are used in Oracle Applications to represent such entities.



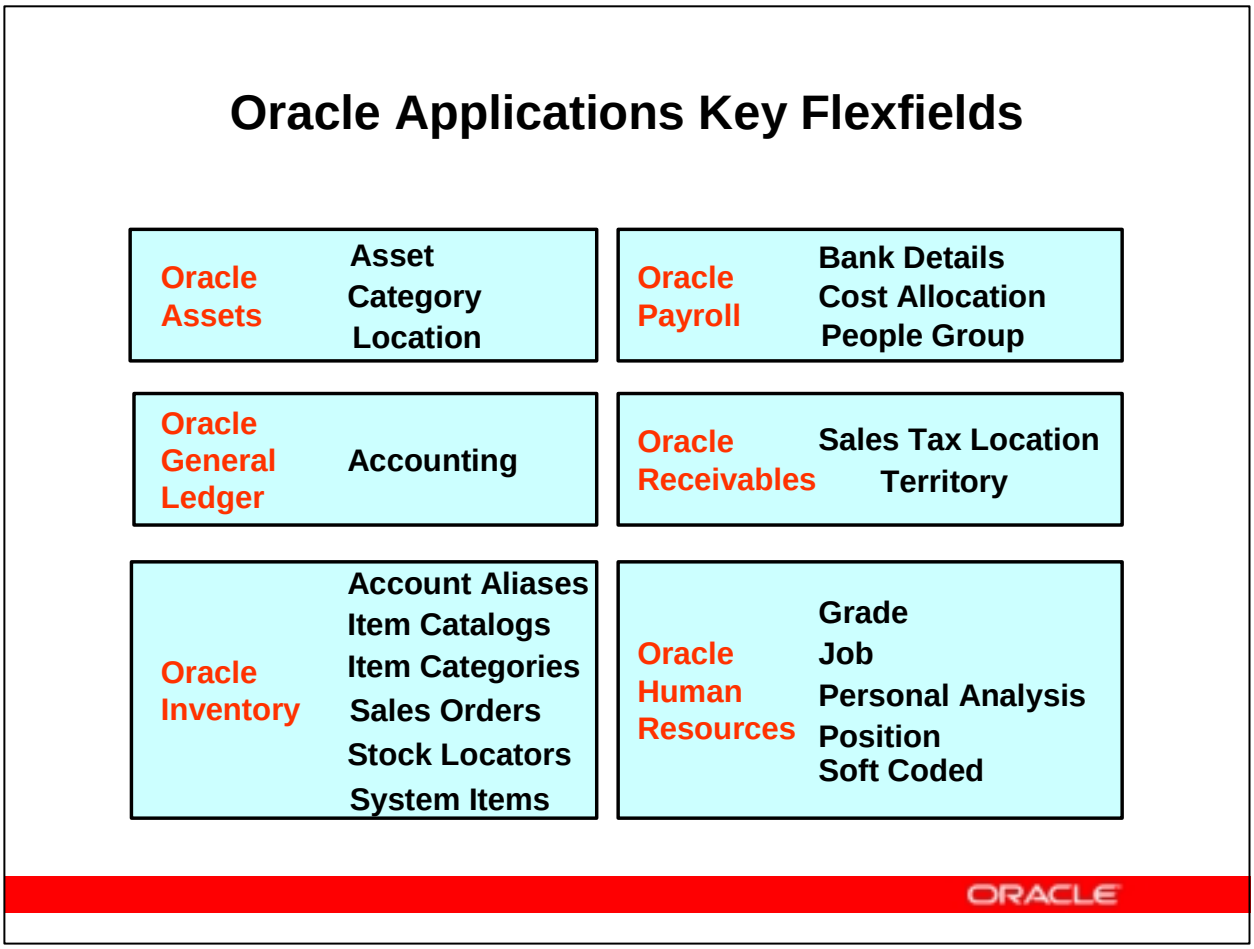
The Accounting Flexfield

The Accounting Flexfield is used to create and display account numbers. The Accounting Flexfield is owned by Oracle General Ledger, but Accounting Flexfield values are used by many of the financial applications. The example shows an Accounting Flexfield structure consisting of five segments.



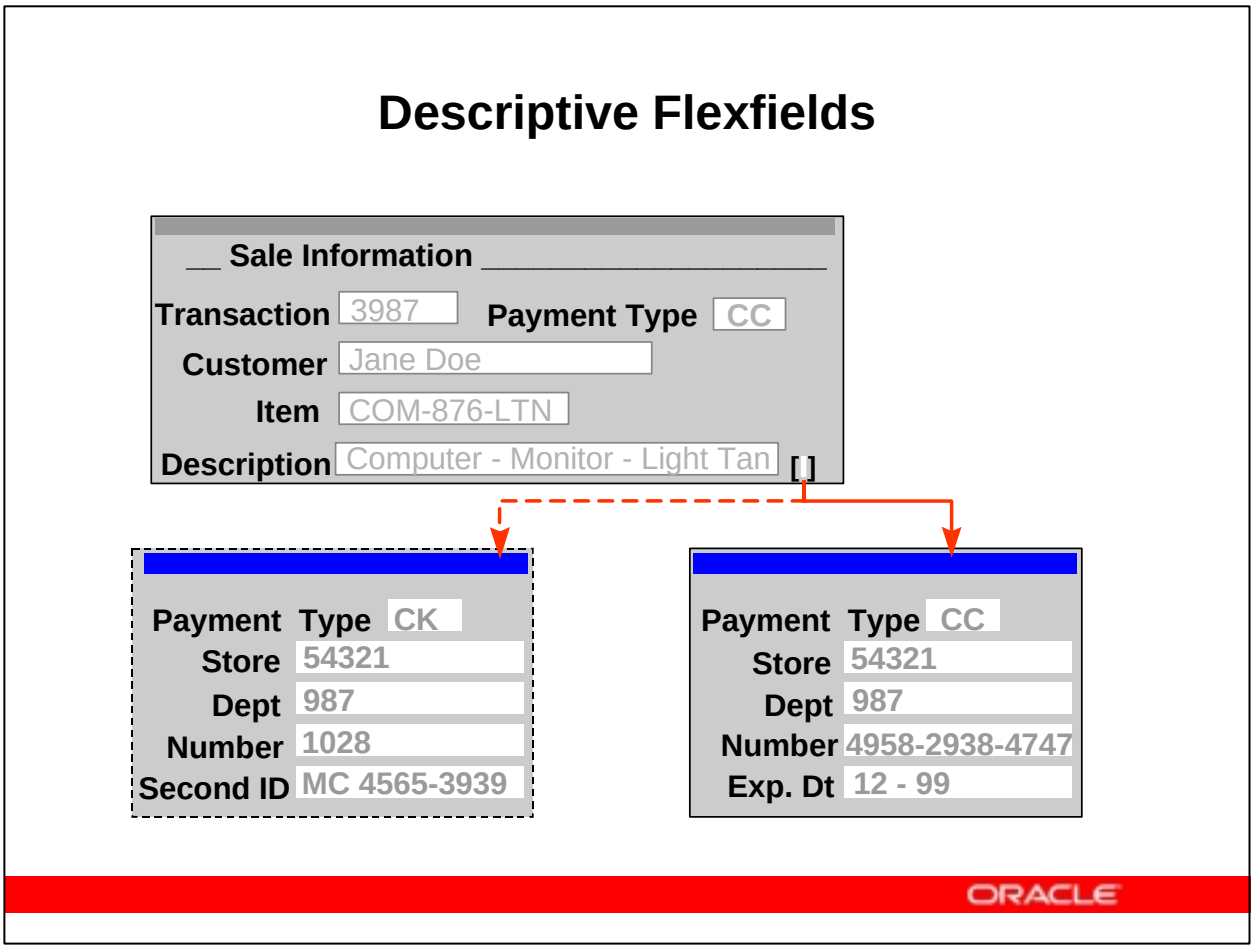
The Asset Category Flexfield

You use the Asset Category flexfield in Oracle Inventory to identify major groupings of inventory. The example shows a key flexfield structure that has two segments.



Key Flexfields Within Oracle Applications

Many Oracle applications have key flexfields. The slide shows some of the applications and the key flexfields they use. In many cases, you must implement the key flexfield to implement the application. Many key flexfields are used by multiple applications. For example, the Accounting Flexfield is used by both Oracle General Ledger and Oracle Payables.



Descriptive Flexfields

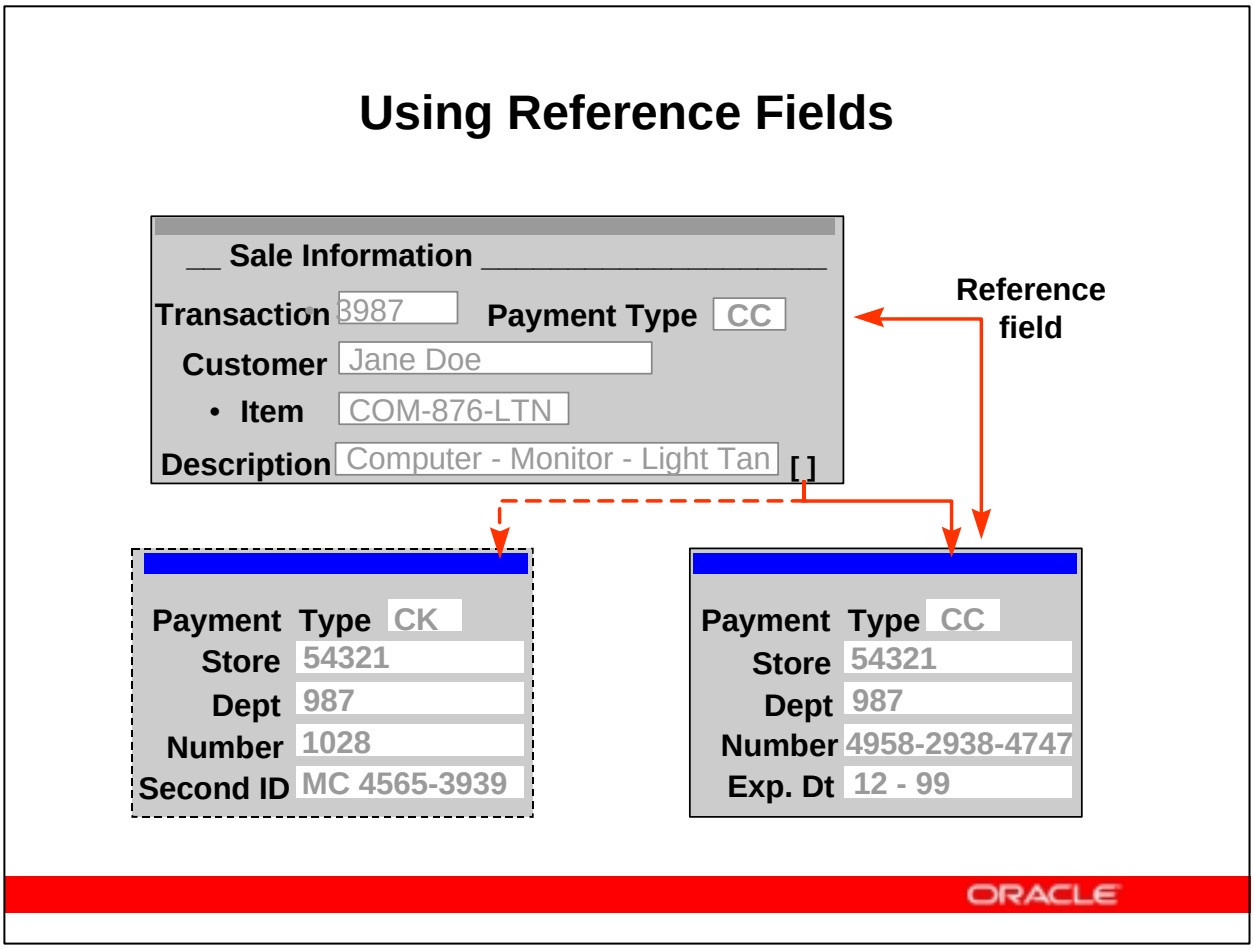
You use descriptive flexfields to collect information beyond that collected by Oracle Applications. Using descriptive flexfields allows you to gather additional specialized information required by your business. The use of descriptive flexfields is optional.

A descriptive flexfield appears on a form as a single-space field enclosed by brackets. You can use the Flexfields: Open Descr Window profile option to specify whether you want the descriptive flexfield window to be opened automatically when you navigate to the bracketed field, if the flexfield is enabled. This profile option is visible and updatable at the user level.

A descriptive flexfield typically uses multiple structures. You can:

- Define a basic structure that gathers additional information for all entities.
- Define several different structures that gather specialized information for different types of the same general entity.
- Define a combination of the preceding two. This structure can gather general information about all entities and then optionally gather certain information about certain types of entities.

The example shows a descriptive flexfield that gathers different payment information based on the type of payment: check (CK) or credit card (CC).



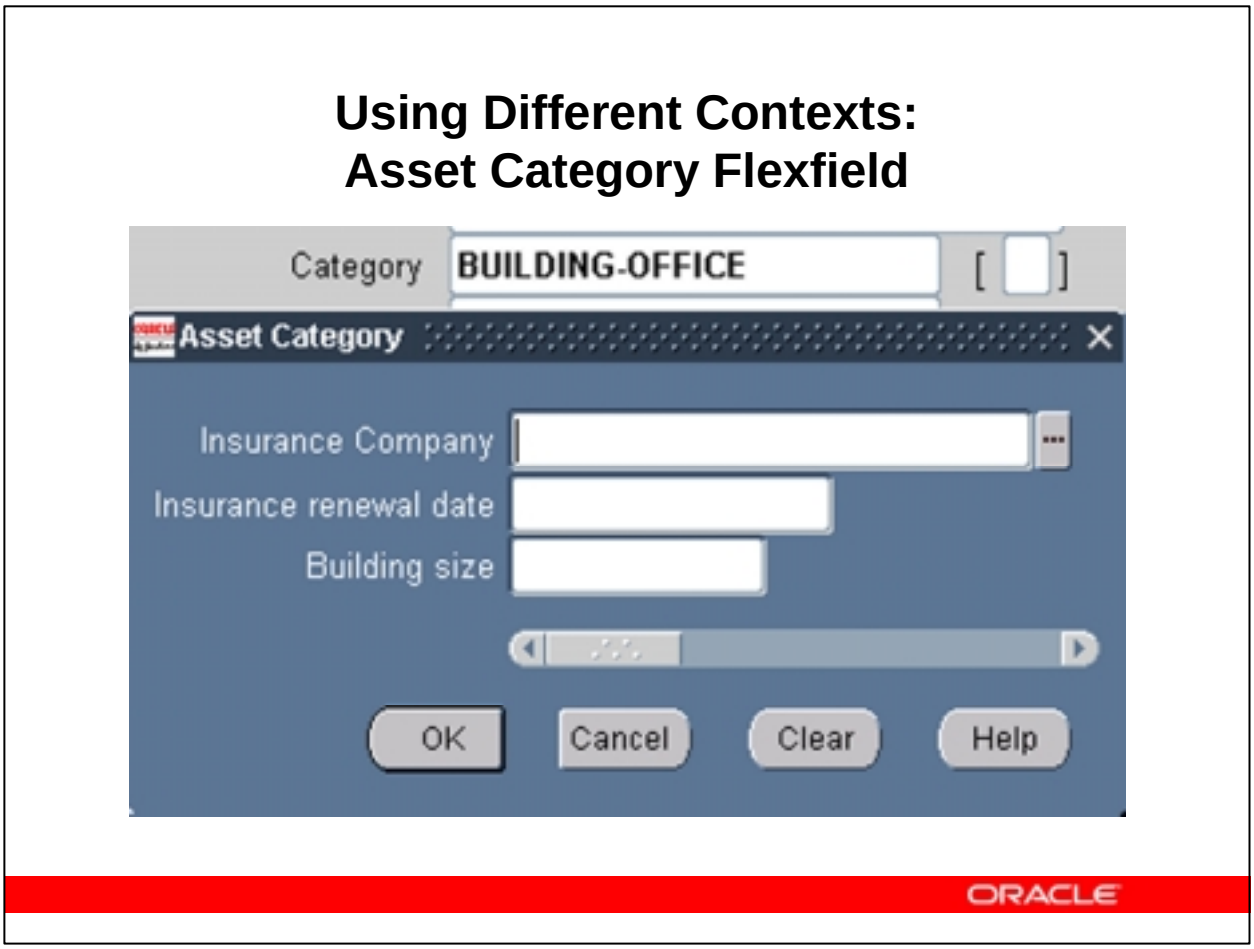
Using Context and Reference Fields to Control Descriptive Flexfields

The same descriptive flexfield often uses multiple structures. Each of these structures can have different segments and gather different data. When you define multiple structures for a descriptive flexfield, you can make the choice of structure either dependent on the value of another field or selectable by the user:

- If the value in a preceding field elsewhere on the form determines which structure is used, the preceding field is known as a reference field. However, if the descriptive flexfield is used in another form, then the two forms must have the same name for the reference field.
- If the value in a field within the flexfield itself determines which structure is used, that field is called a context field.

The example on the slide shows a hypothetical window containing sales information. One of the fields on this window, Payment Type, is used as a reference field for a descriptive flexfield that contains information about how the customer will pay.

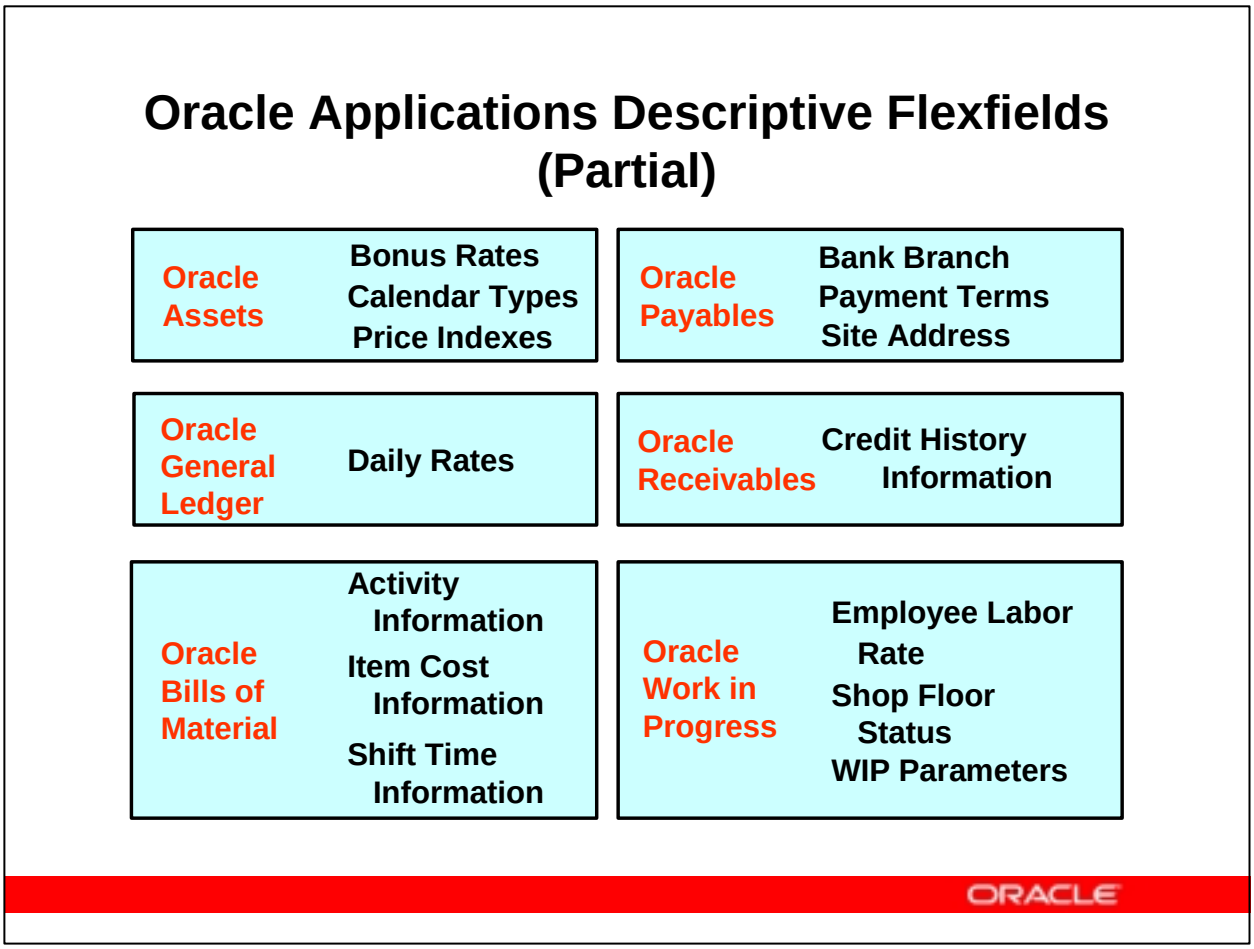
In this example, the buyer is using a credit card so the credit card information is gathered by using the descriptive flexfield structure shown on the right. If payment is by check, you could gather check-related information by using the descriptive flexfield structure shown on the left.



The Asset Category Flexfield

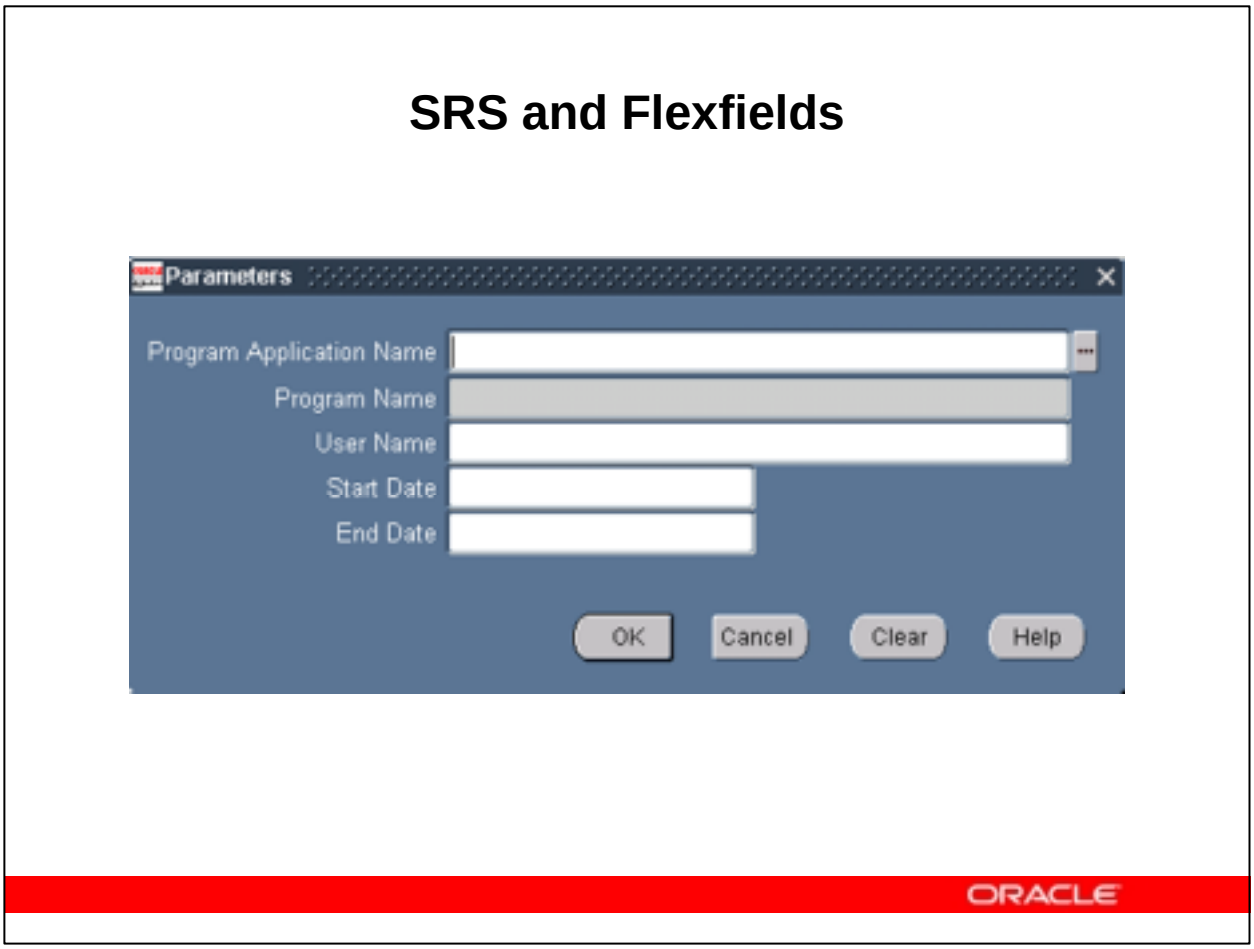
The example on the slide shows the Asset Category descriptive flexfield. The descriptive flexfield is gathering further information about an office building asset. You can see this both by the value in the reference field, and by the type of information being requested on the flexfield.

Note: In this example, the reference field is itself a key flexfield.



Partial List of Oracle Applications Descriptive Flexfields

Numerous descriptive flexfields are available throughout Oracle Applications. The slide shows only a partial list. In contrast to key flexfields, descriptive flexfields are not shared between applications.

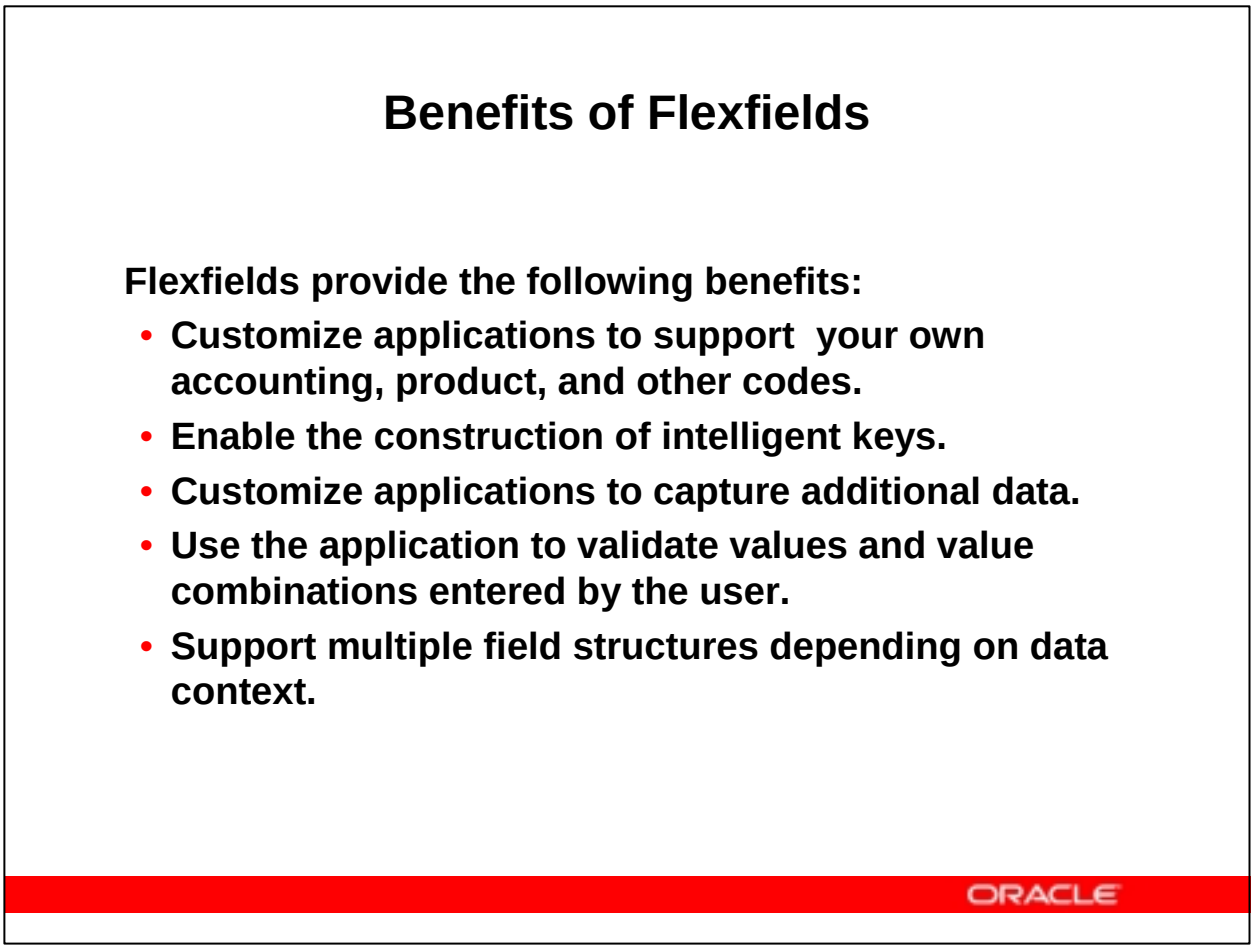


Using Flexfields as Parameters Within Standard Request Submission

Reports and concurrent programs submitted with Standard Request Submission often have parameter windows that pop up. These parameter windows behave similarly to descriptive flexfields.

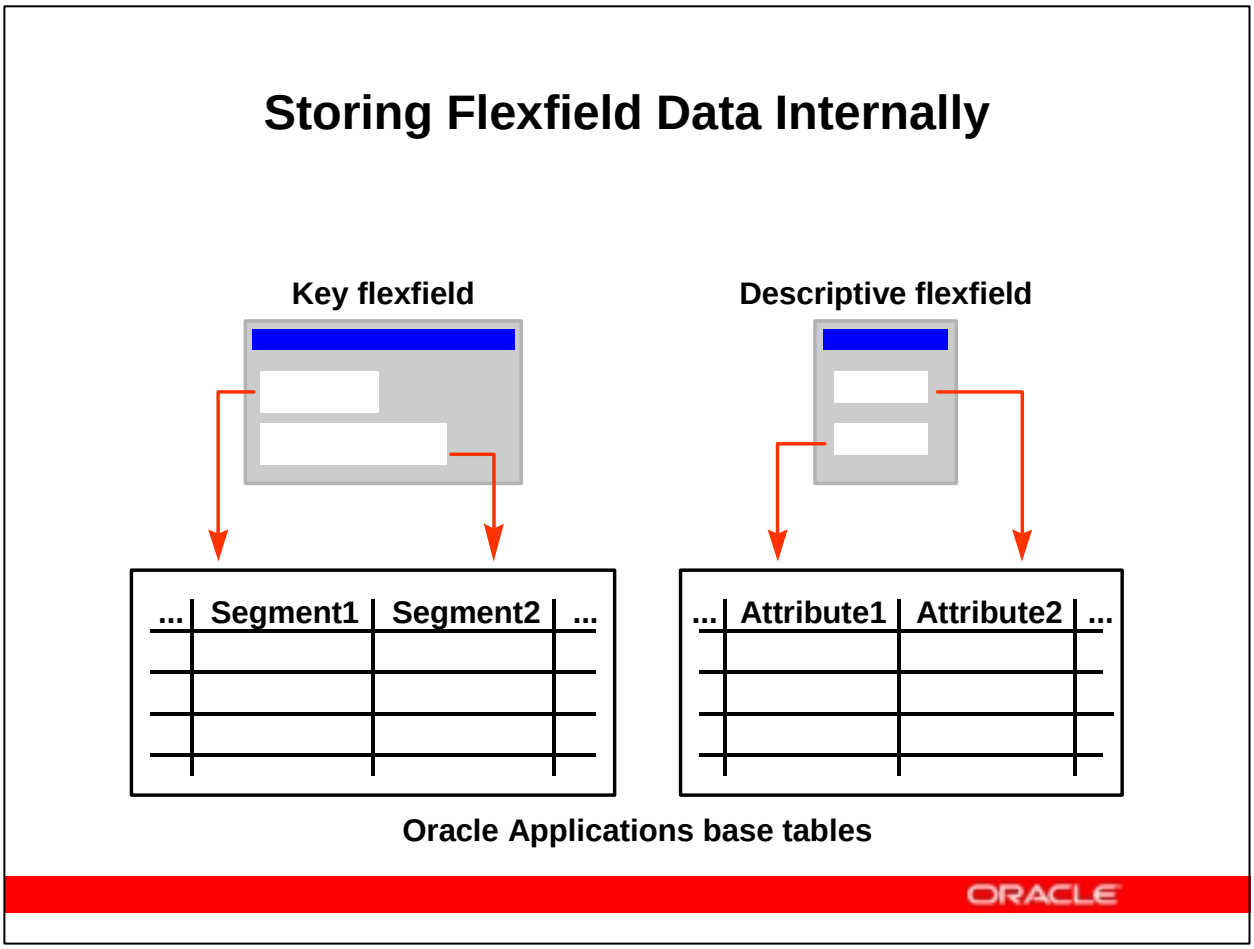
Parameter Windows Are Context Sensitive

- The window that appears and the segments that are contained in it depend on which request is being run.
- The request name acts as a reference field.
- Many request parameters have value sets associated with them to control the allowable values for that parameter. Such value sets are often shared by many other flexfields. You should always be careful when modifying any value sets that are also used by SRS processing.



Benefits of Flexfields

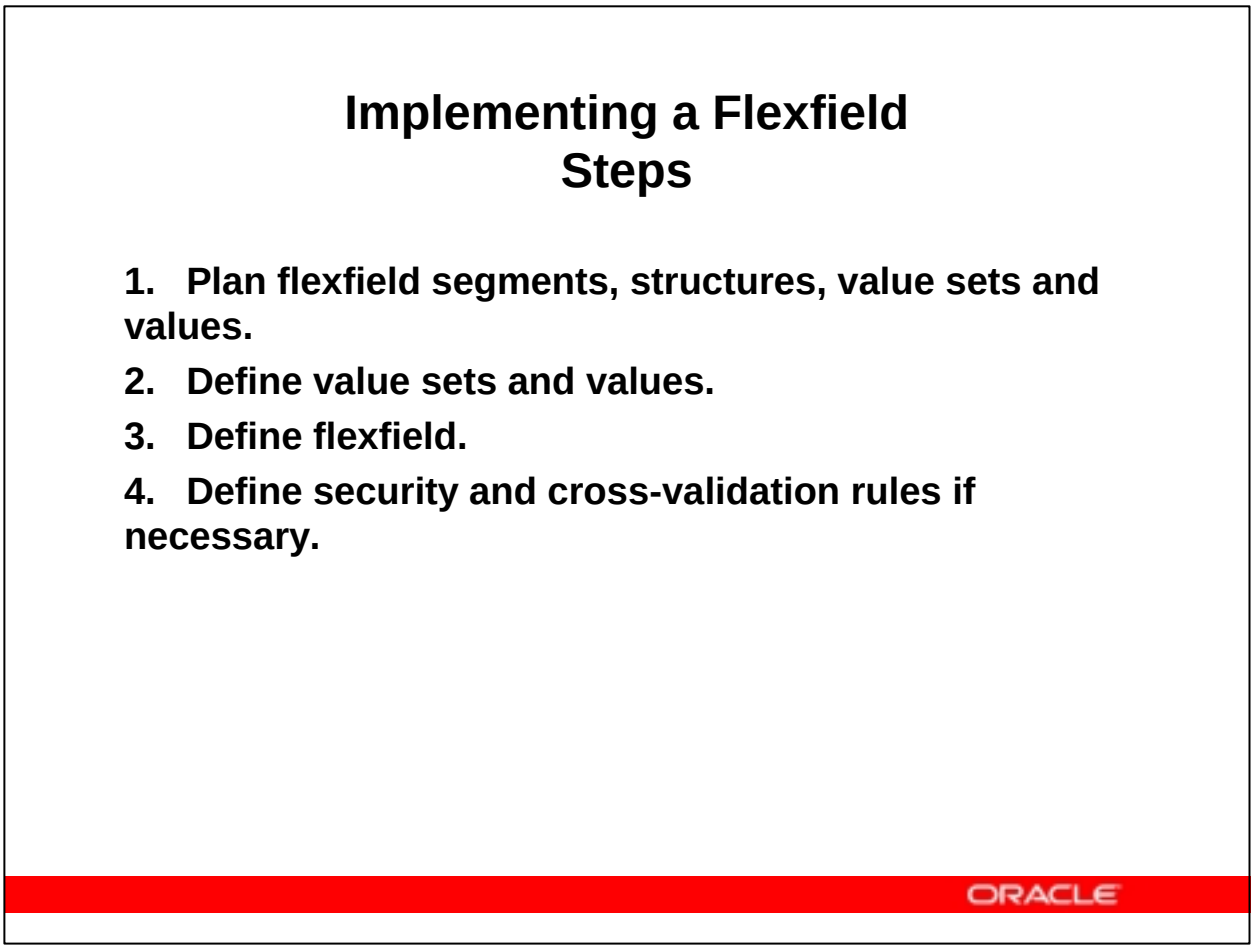
You should now have a basic understanding of how flexfields are used by Oracle Applications. The ease of customizing flexfields provides many benefits to users. The main benefits are listed on the slide.



Storing Flexfield Values in Database Tables

The values entered through an application's flexfields are stored internally in that application's database tables:

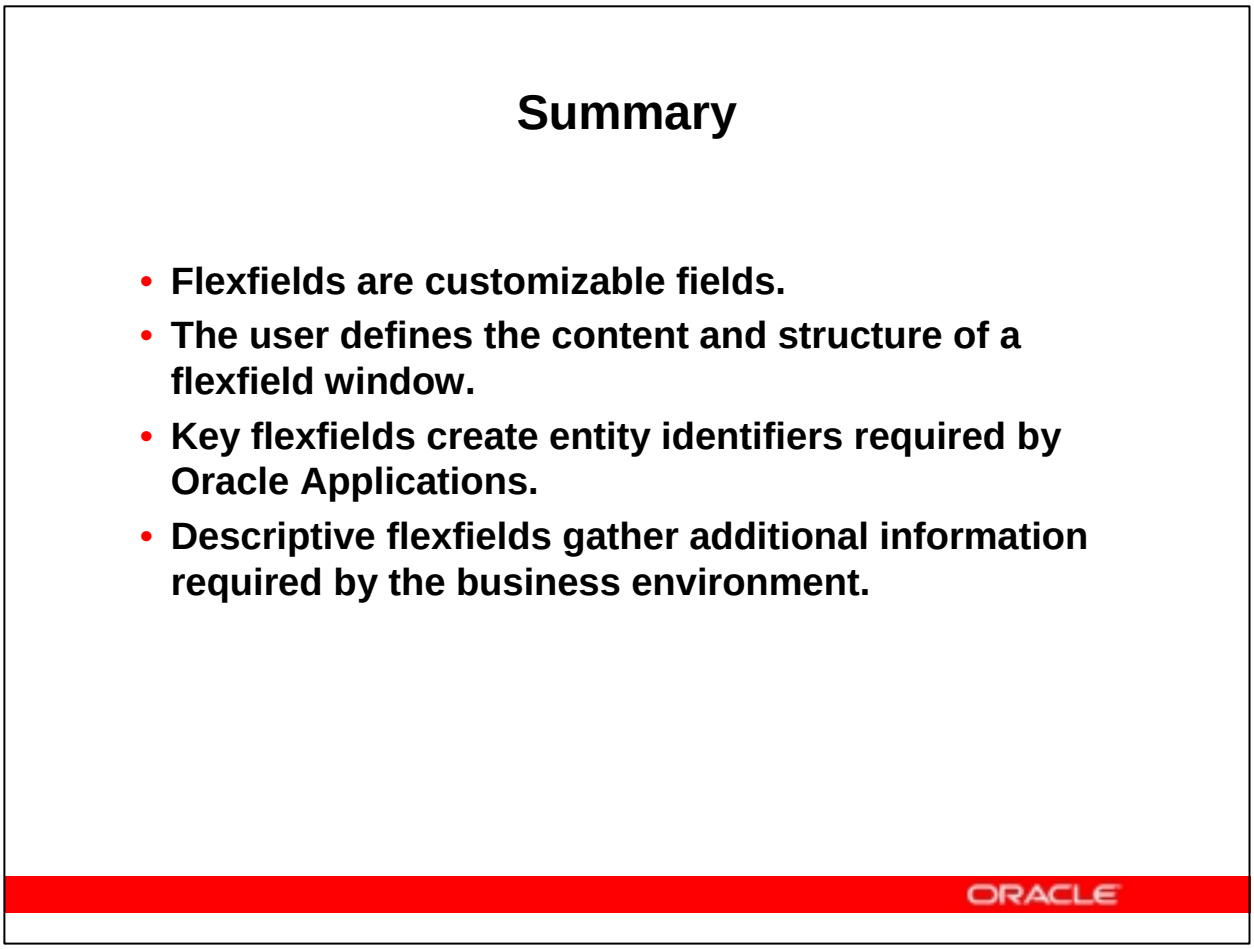
- Each segment's value is stored in a column in one of the base tables.
- The column name reflects the type of flexfield data it holds. In general, key flexfields store their data in columns called SEGMENTn, where n is a number. In general, descriptive flexfields store their data in columns called ATTRIBUTEn, 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.



Implementing a Flexfield: General Procedure

Use the following steps when defining flexfields, both key and descriptive:

- Plan your flexfield structure and layout. Remember that many flexfields use more than one structure, and that each structure can consist of different segments. Also plan any value sets and their values.
- Define flexfield value sets. Value sets describe the allowable values for the flexfield segment. If you create your value sets first, you can refer to them while your are defining your flexfield segments in the following step.
- Define flexfield segments and structures. Use the plan you designed earlier.
- Define values for your value sets.
- For key flexfields, define security and cross-validation rules as necessary.



Summary

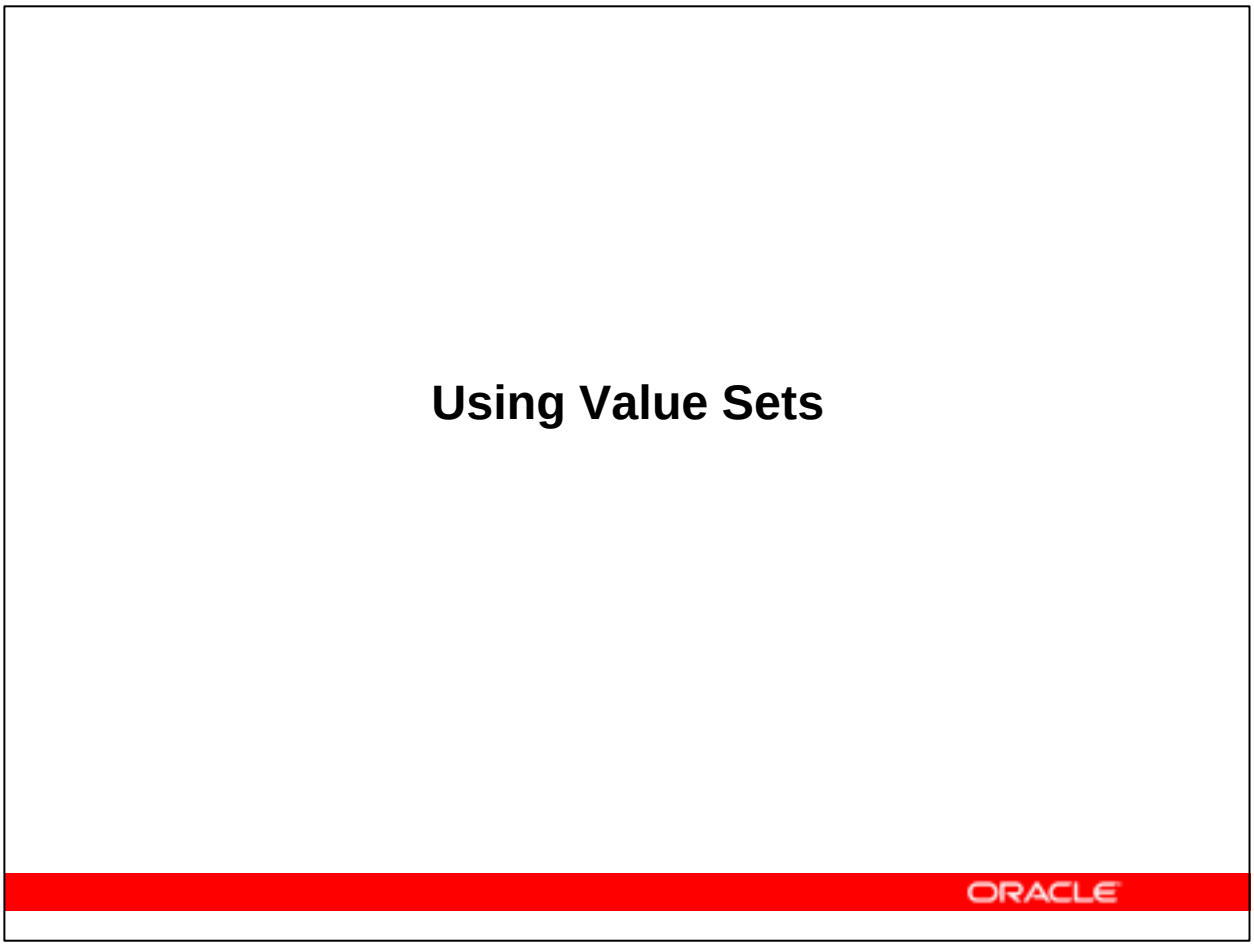
Flexfields are Oracle Applications fields that you can customize. You can define the content and layout of the flexfield to reflect your own business needs.

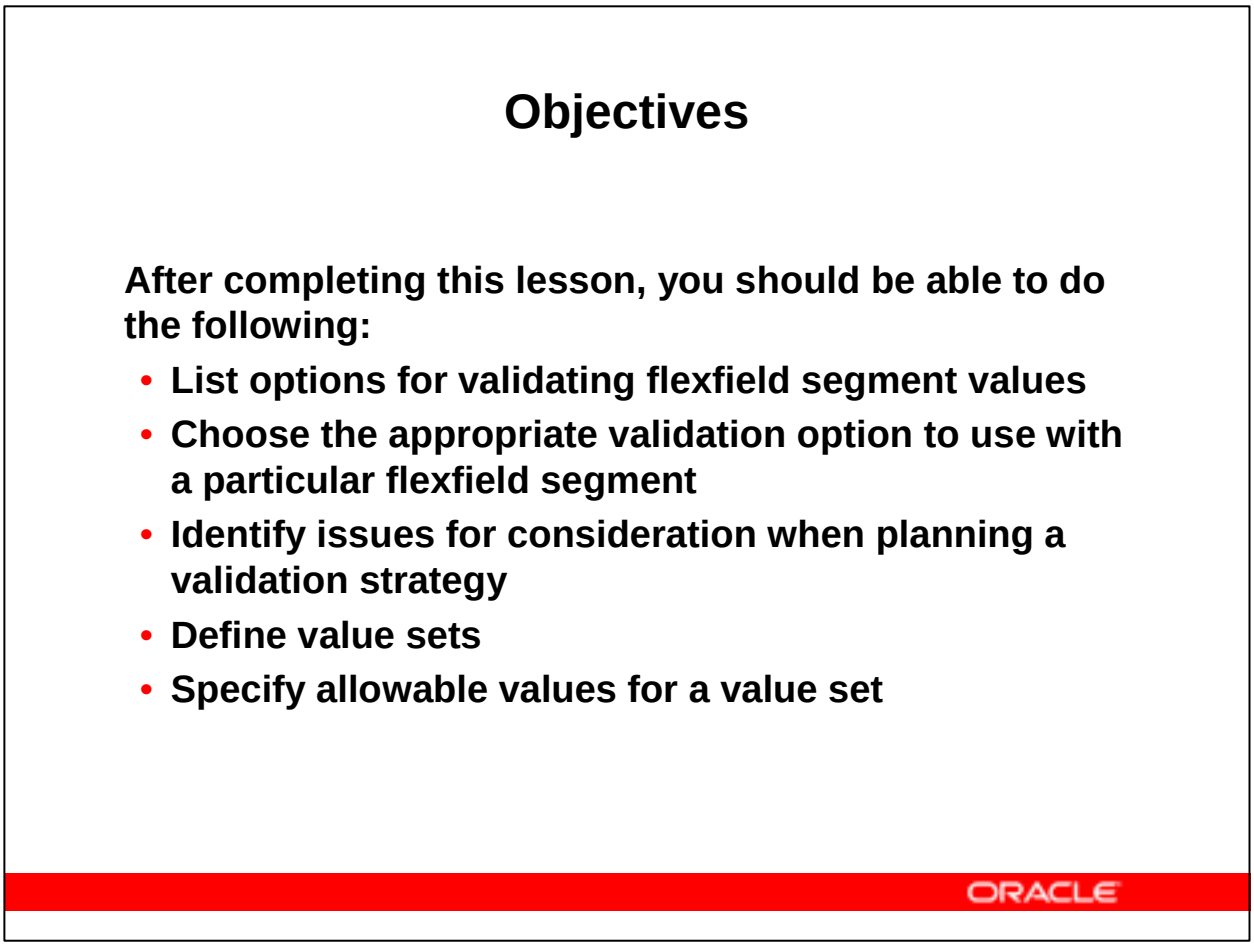
There are two types of flexfields:

- Key flexfields are used to create entity identifiers required by Oracle Applications. You can use key flexfields to structure these identifiers according to your business environment. The definition of many key flexfields is required by the application.
- You use descriptive flexfields to gather additional information for storage or processing. You can define descriptive flexfields in multiple ways to accommodate the needs of many different groups of users. The use of descriptive flexfields is optional.

Using Value Sets

Chapter 14



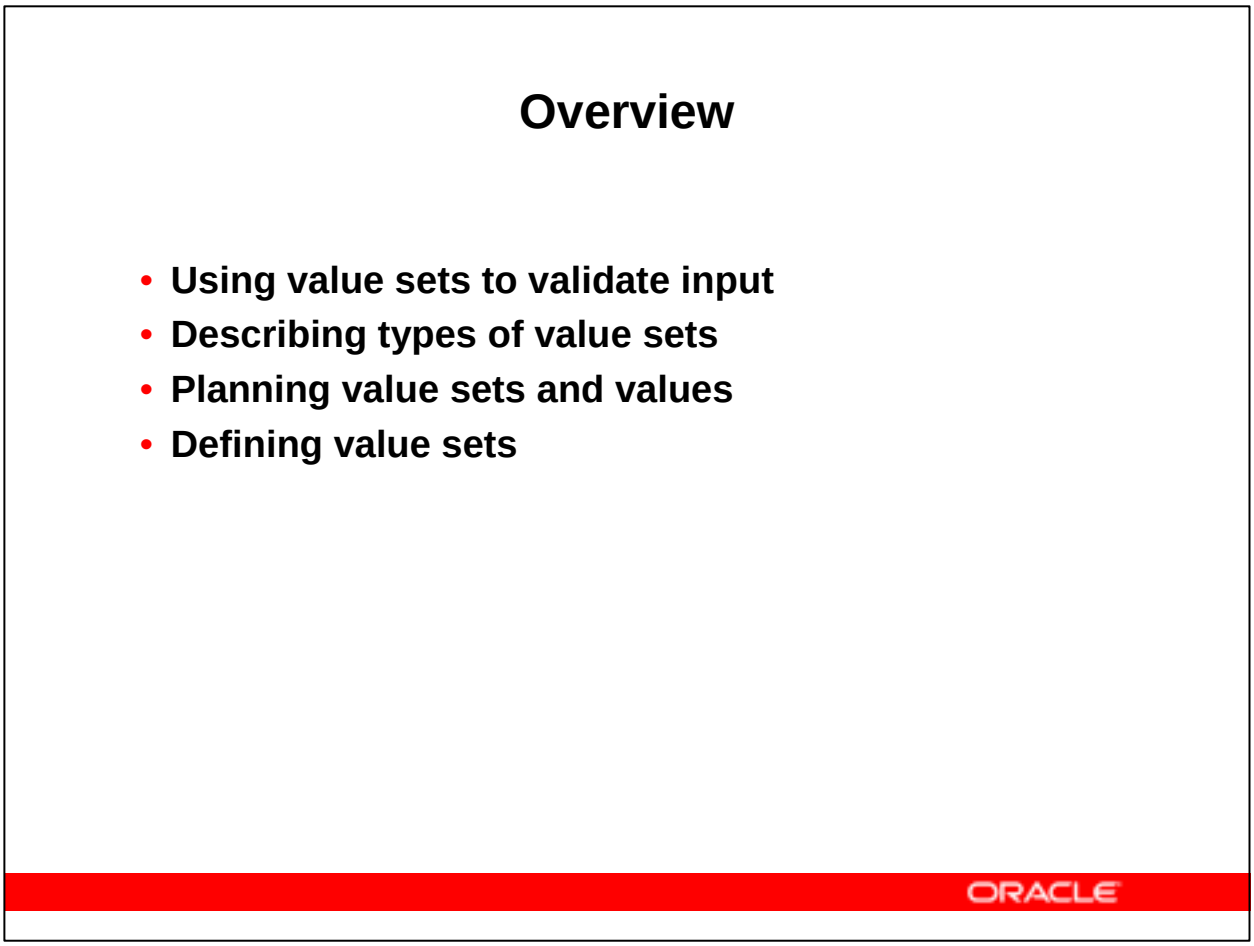


Lesson Topics

This lesson describes planning and using value sets to control input with flexfields.

At the end of this lesson, you should be able to:

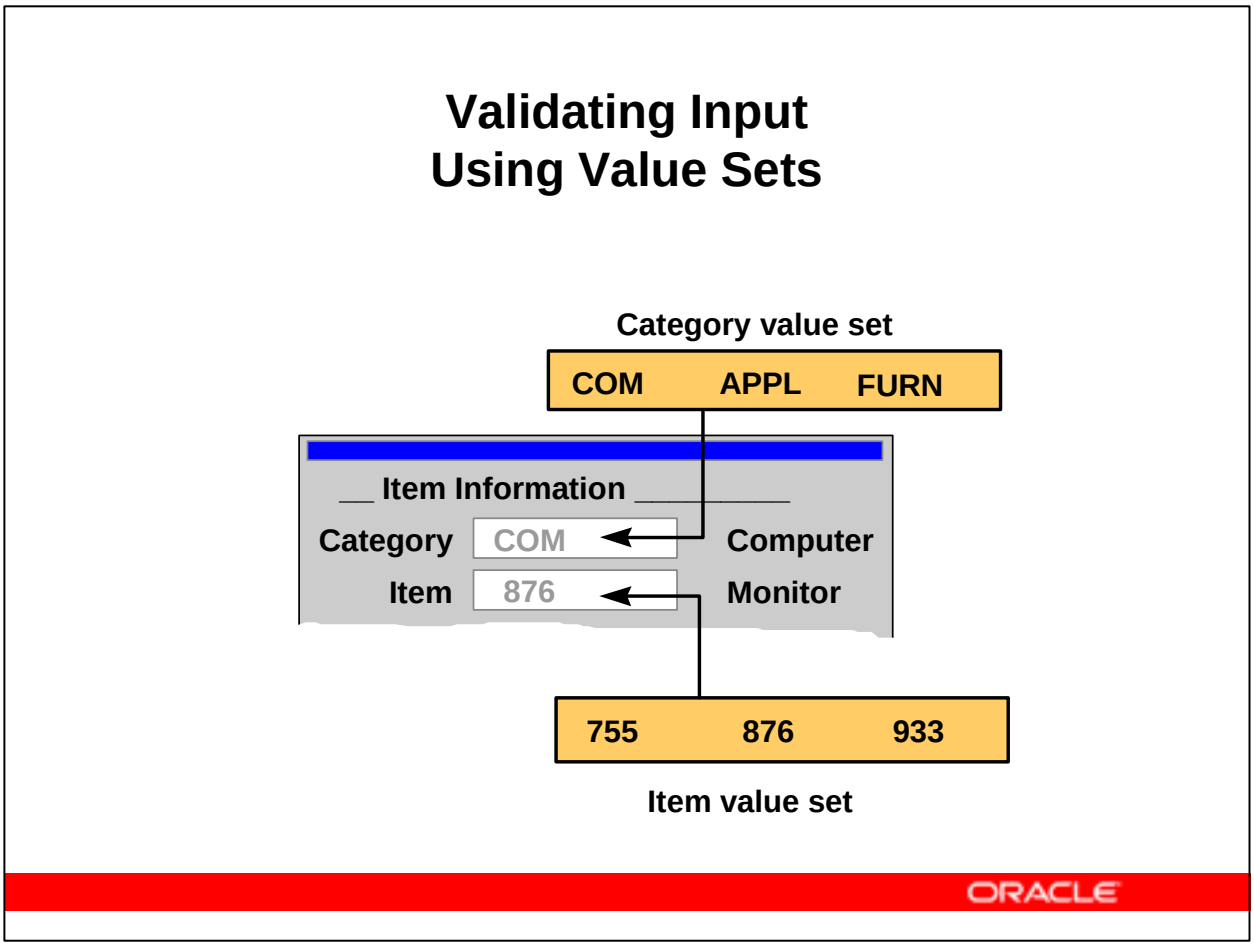
- List options for validating flexfield segment values and report parameters
- Choose the appropriate validation option to use with a particular segment
- Identify issues for consideration when planning a validation strategy
- Define new value sets
- Define allowable values for a value set



Lesson Overview

Value sets are definitions of allowable values for use with a flexfield segment or report parameter. This lesson covers the different types of value sets and how each is used. It also describes how to plan the value set as well as the actual values for those sets that require a list of values. Finally, it covers the definition process itself.

Validating Input Using Value Sets



Value Sets Define Allowable Values

You can create a value set and assign it to a flexfield segment or report parameter to control what values that segment can use.

Value Sets Check User Input

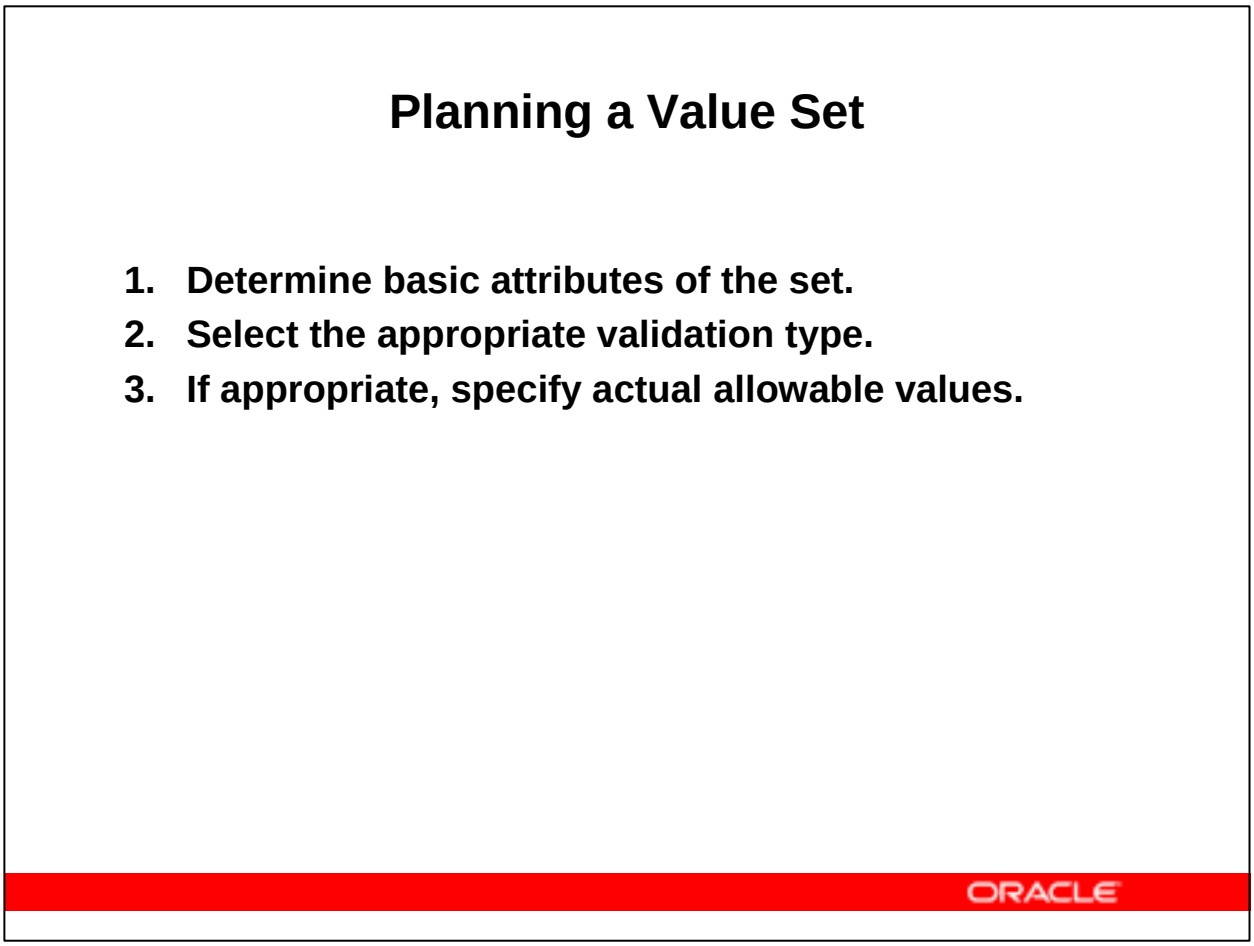
Define a value set for each flexfield segment. A value set is a general definition of the values that are acceptable for a particular flexfield segment or report parameter. A value set can also have a list of actual values available to the segment, although this is not required.

Other Advantages of Value Sets

Value sets control which segments can use special features such as Longlist, value security, and segment value combination checking.

The example in the slide shows two value sets:

- The Category value set specifies the allowable values for the Category segment.
- The Item value set specifies the allowable entries for the Item segment.



Planning Your Value Set

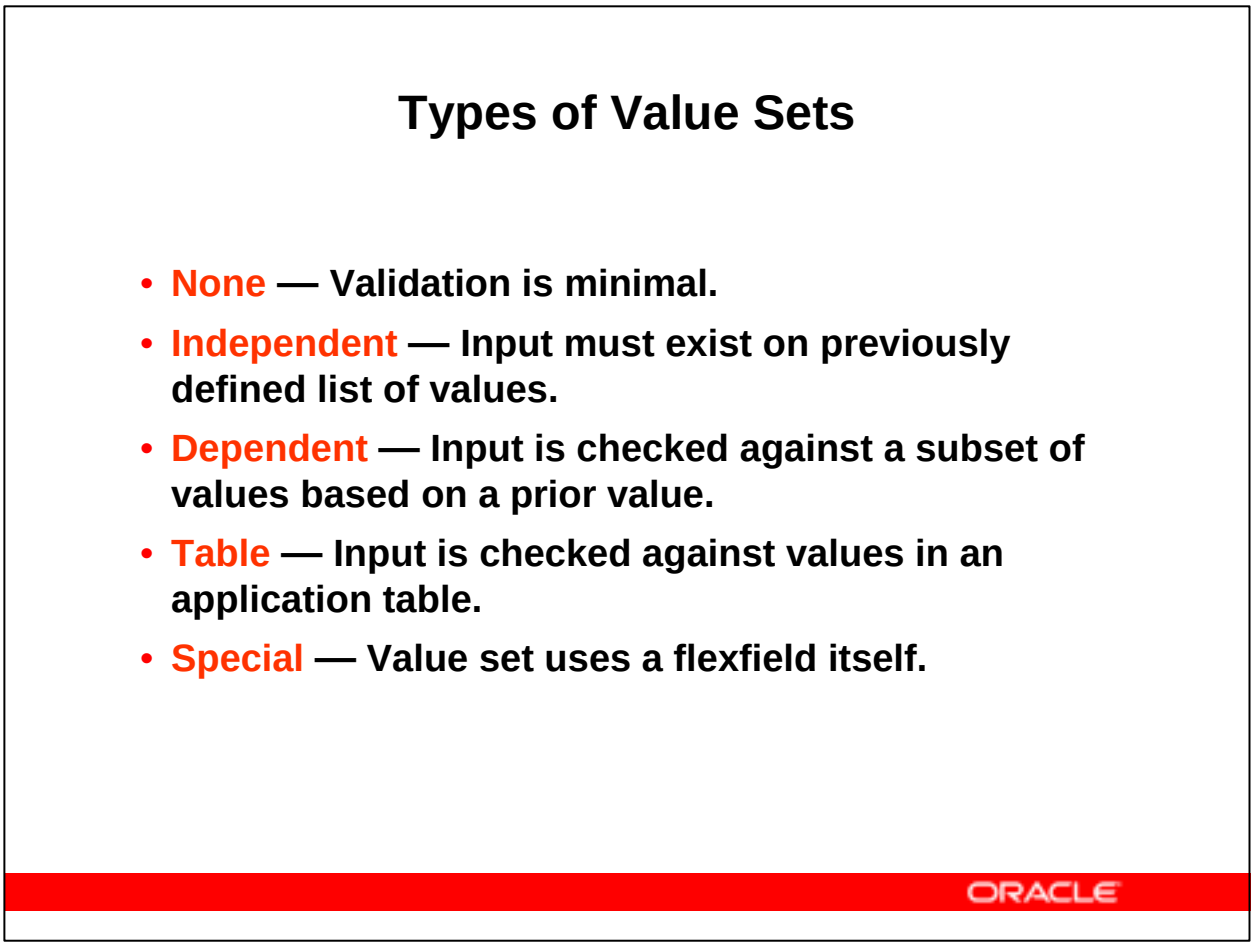
There are several steps to planning your value set:

1 Determine the basic attributes for the set. Choose what conditions all values must fit to be considered valid values. Such criteria include data type (Character, Numeric), value length, and minimum and maximum values, if appropriate.

2 Select the appropriate validation strategy. Choose the type of validation most appropriate for the data. Validation types are discussed in the following pages.

3 Identify approved values, if appropriate. For some value sets, you know ahead of time the acceptable values. In these cases, in addition to defining the value set, you can also specify the values that appear when the user enters the flexfield segment. The segment accepts only values explicitly defined for that value set.

For some value sets, it does not make sense to try to provide a complete list of all the approved values. For example, a segment containing customer telephone numbers probably does not need a list of values, because each new customer will have a new telephone number which you do not know at design time.



Types of Value Sets

You can define several types of value sets depending on how you need your values to be checked. All value sets perform minimal checking; some value sets also check against the actual values, if you have provided any.

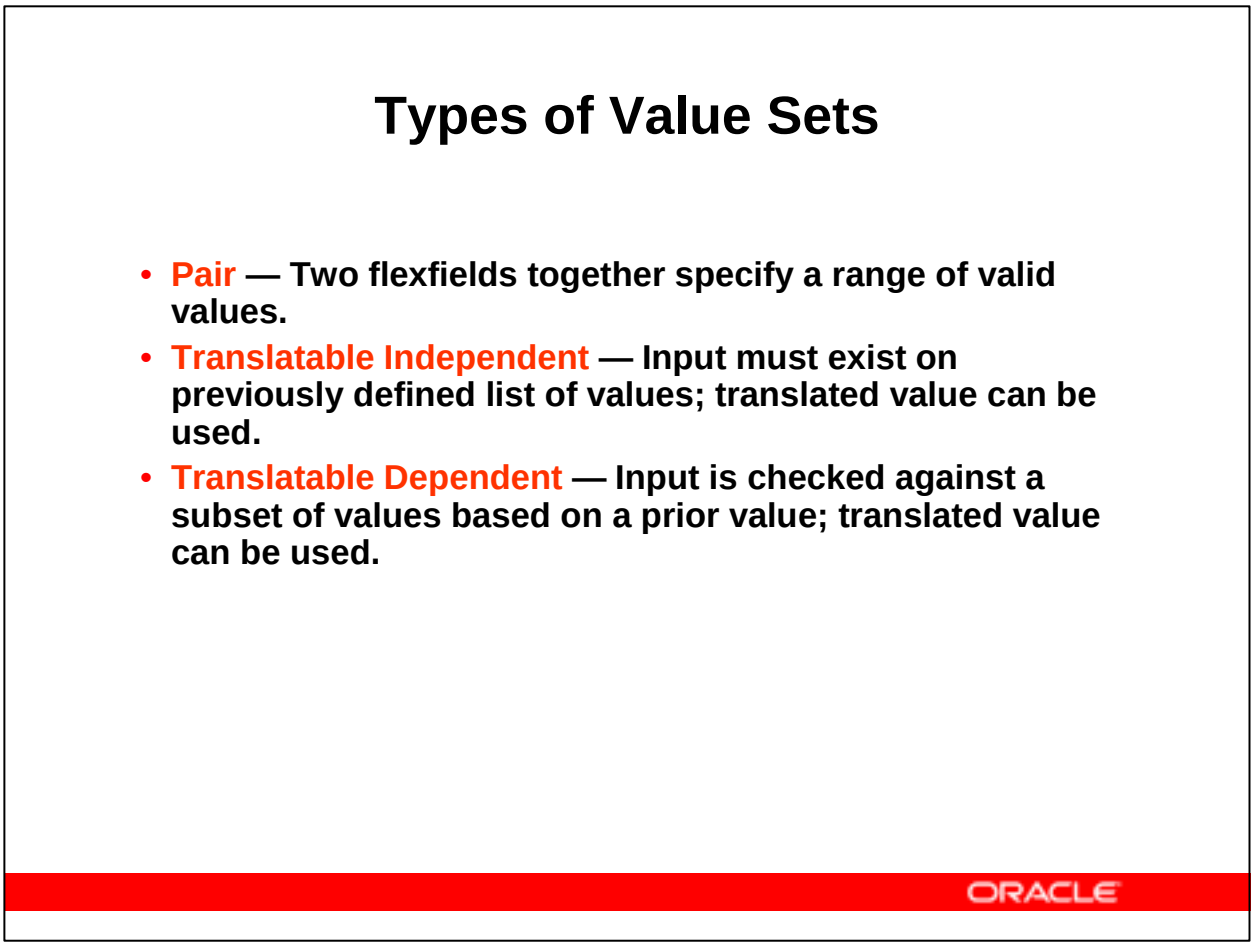
None - A value set of the type None has no list of approved values associated with it. A None value set performs only minimal checking of, for example, data type and length.

Independent - Independent type value sets perform basic checking but also check a value entered against the list of approved values you define.

Dependent - A Dependent value set is associated with an Independent value set. Dependent value sets ensure that all dependent values are associated with a value in the related Independent value set.

Table - Table value sets obtain their lists of approved values from existing application tables. When defining your table value set, you specify a SQL query to retrieve all the approved values from the table.

Special - This specialized value set provides another flexfield as a value set for a single segment.

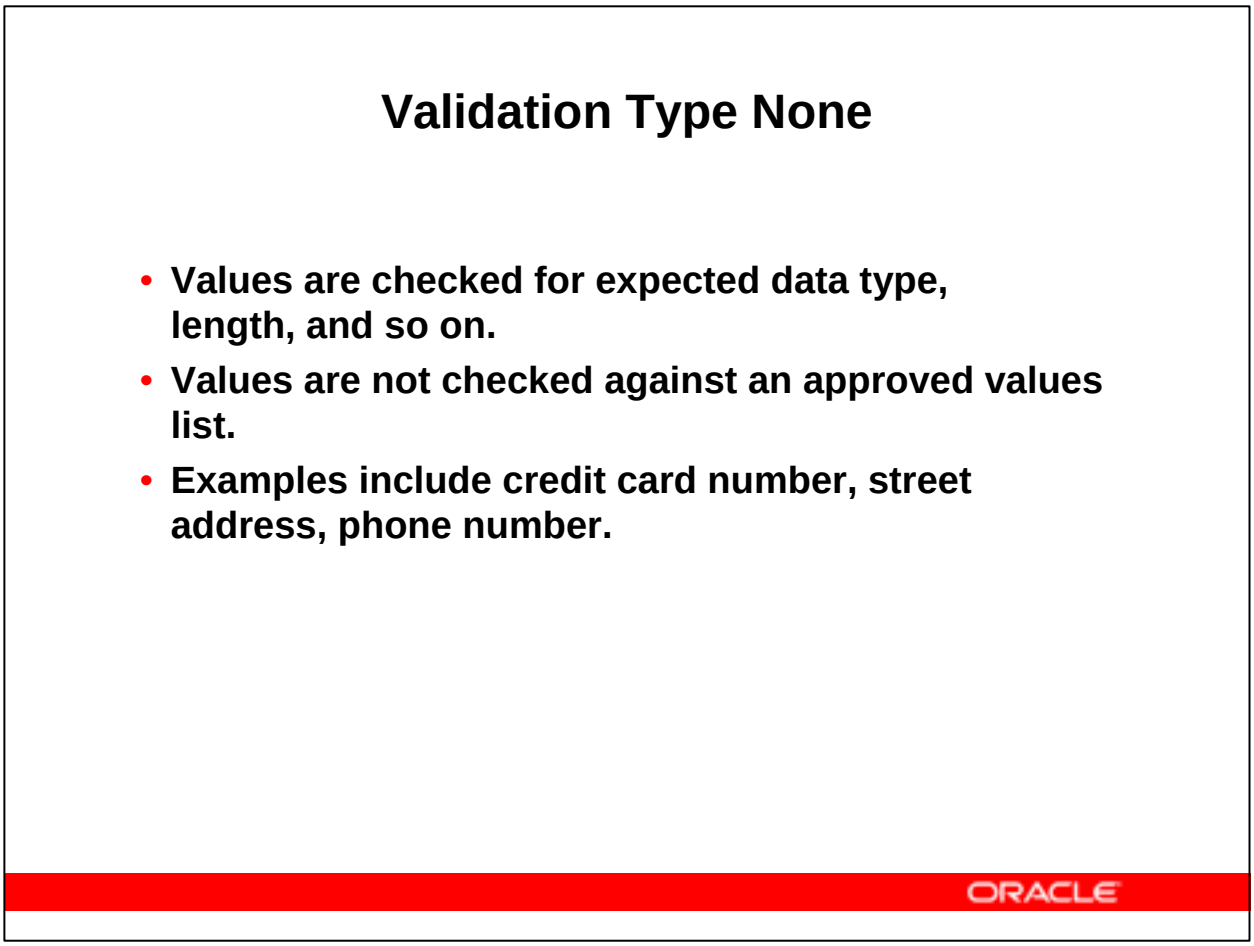


Types of Value Sets (continued)

Pair - This specialized value set provides a range flexfield as a value set for a pair of segments.

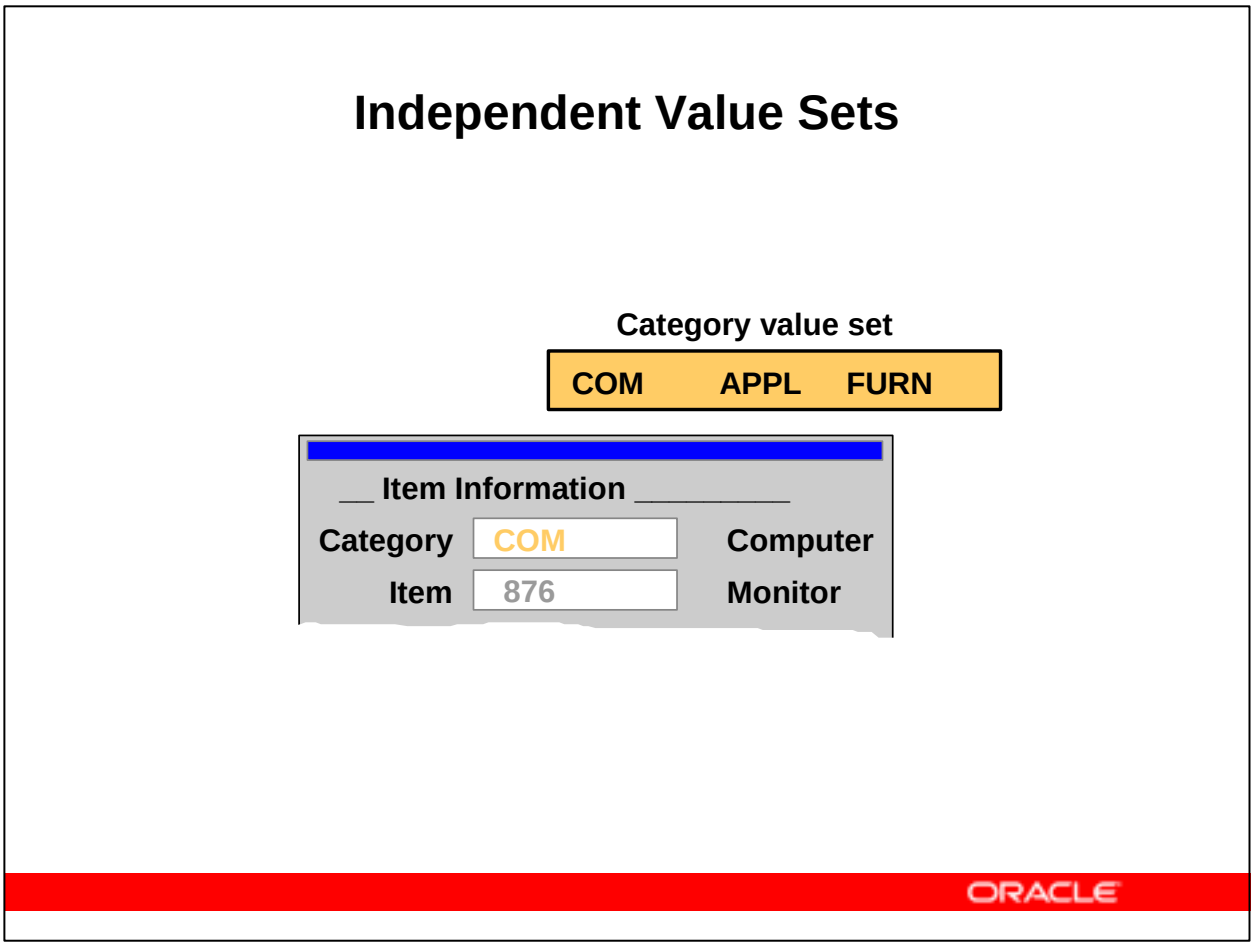
Translatable Independent - A Translatable Independent value set is similar to an Independent value set in that it provides a predefined list of values for a segment. However, a Translatable Independent value set can contain display values that are translated into different languages.

Translatable Dependent - 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. However, a Translatable Dependent value set can contain display values that are translated into different languages.



Validation Type None

A validation of type None performs only minimal checking of the data input through this flexfield segment. Use the validation type None when the actual values to be input cannot be anticipated in advance but the data entered must conform to a particular data type, length, or any other formatting criteria. Examples of such values include credit card numbers, street addresses, and phone numbers.

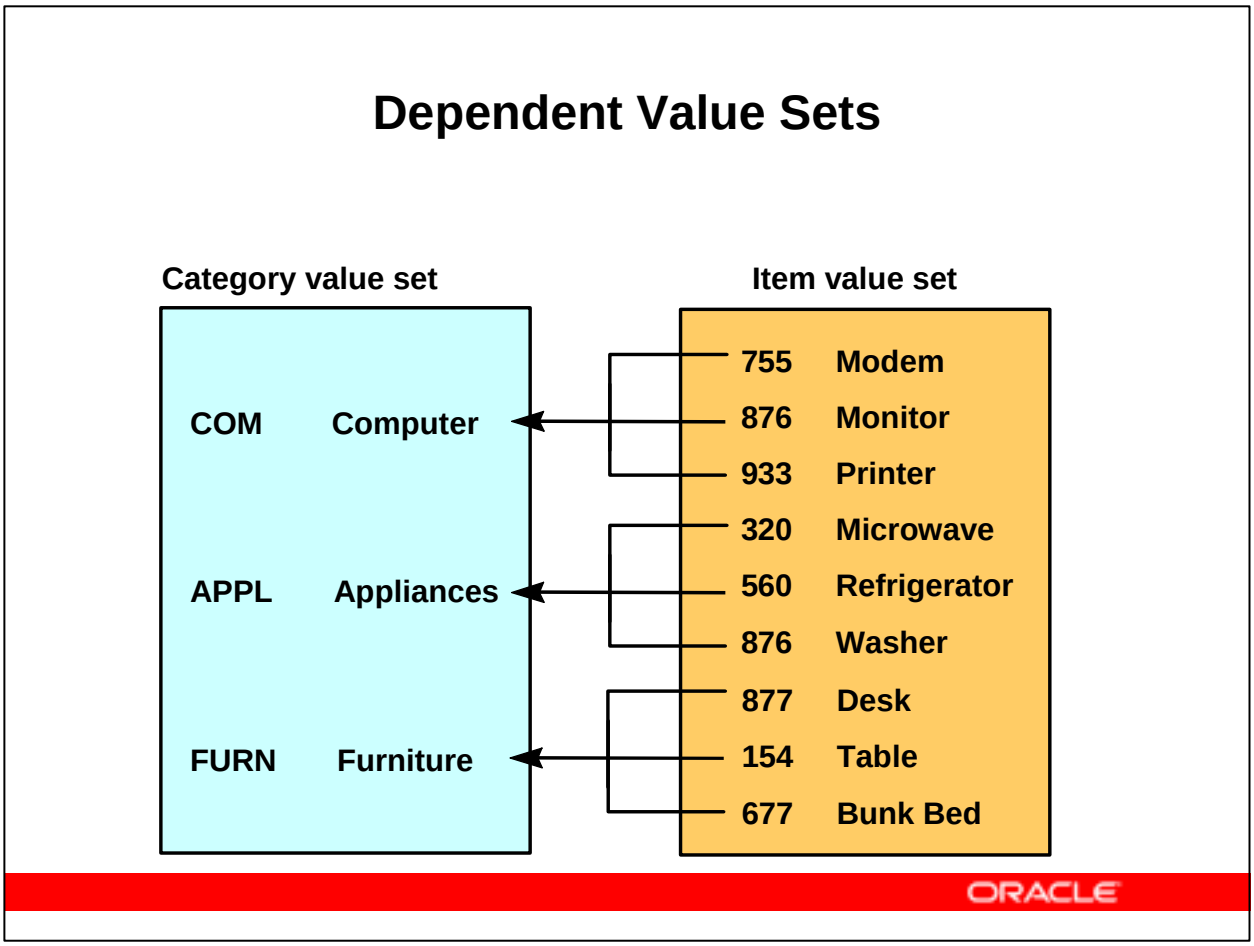


Validation Type Independent

Use the validation type Independent when you know the allowable values ahead of time. With an Independent validation type, you typically also define a list of the actual values that are approved for use. The user selects these values from a pop-up list.

In the example in the slide, the Category value set contains the only three allowable values for the Category segment of a hypothetical Item Information flexfield. The category segment value must indicate a computer item, an appliance item, or an item of furniture.

Dependent Value Sets

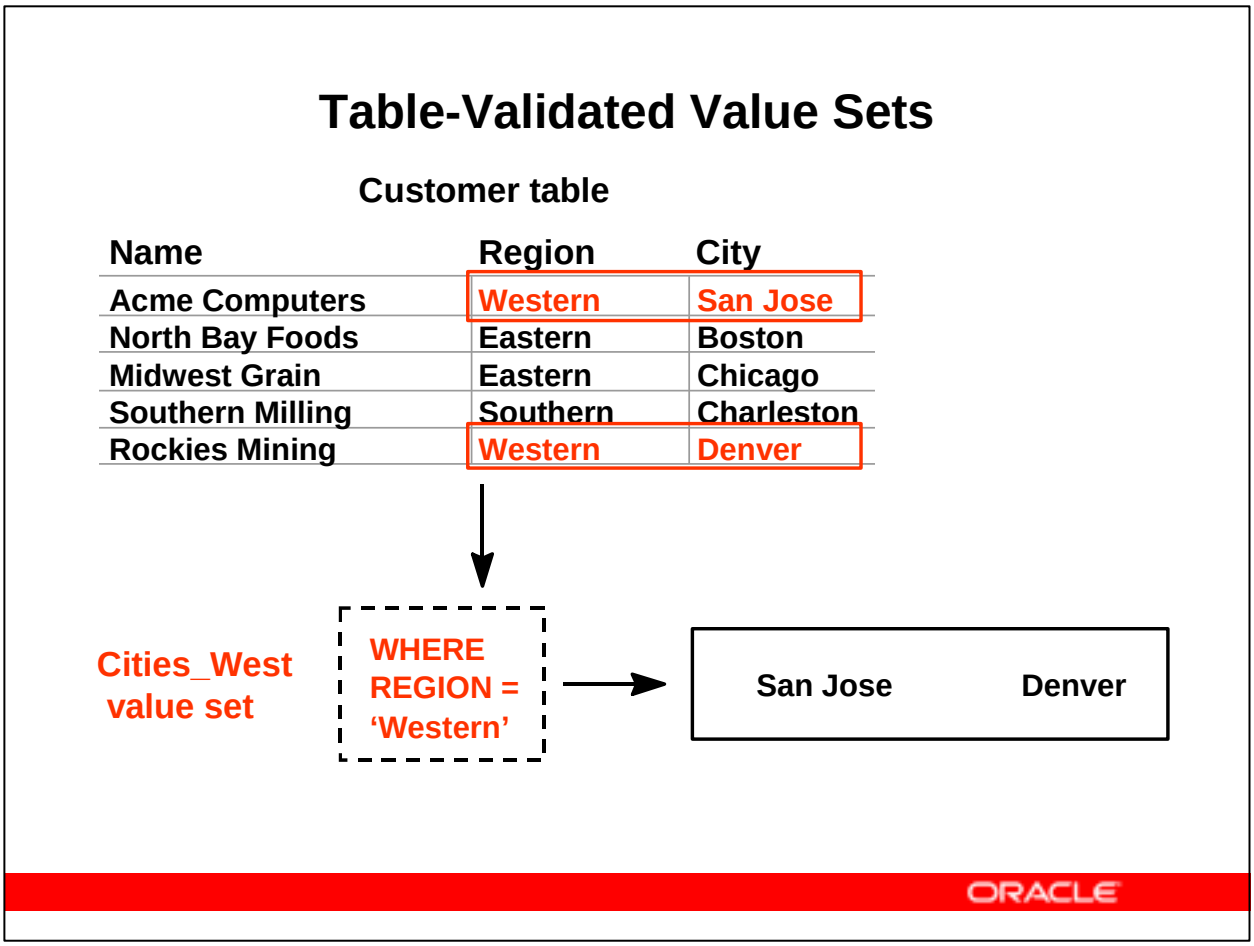


Validation Type Dependent

A Dependent value set is also associated with a list of approved values. In this case however, the values on the list can be grouped into subsets of values. Each subset of values is then associated with a value from an Independent value set. Once a value from the Independent value set has been specified, the list of values for the Dependent value set displays only the values that are approved for the value selected from the Independent value set.

In the example in the slide, once a value from the Category value set has been specified, only the appropriate values from the Item value set are displayed.

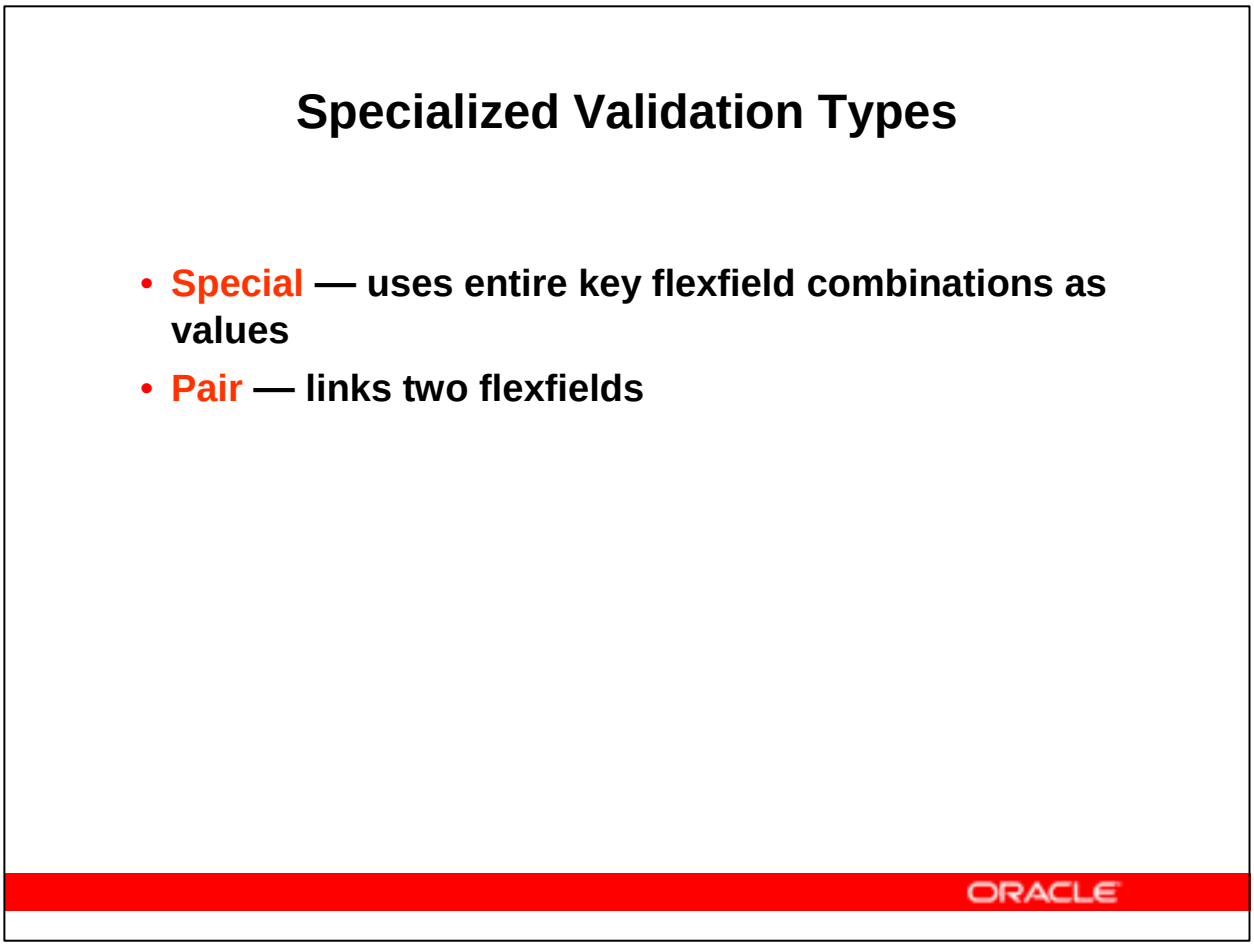
The segment or parameter using the dependent value set must appear after the segment or parameter using the independent value set.



Validation Type Table

If an application table already contains all the allowable values, you can define a table-validated value set to retrieve them from the table. You can specify selection criteria in the form of an SQL WHERE clause if you need to restrict the values for some reason.

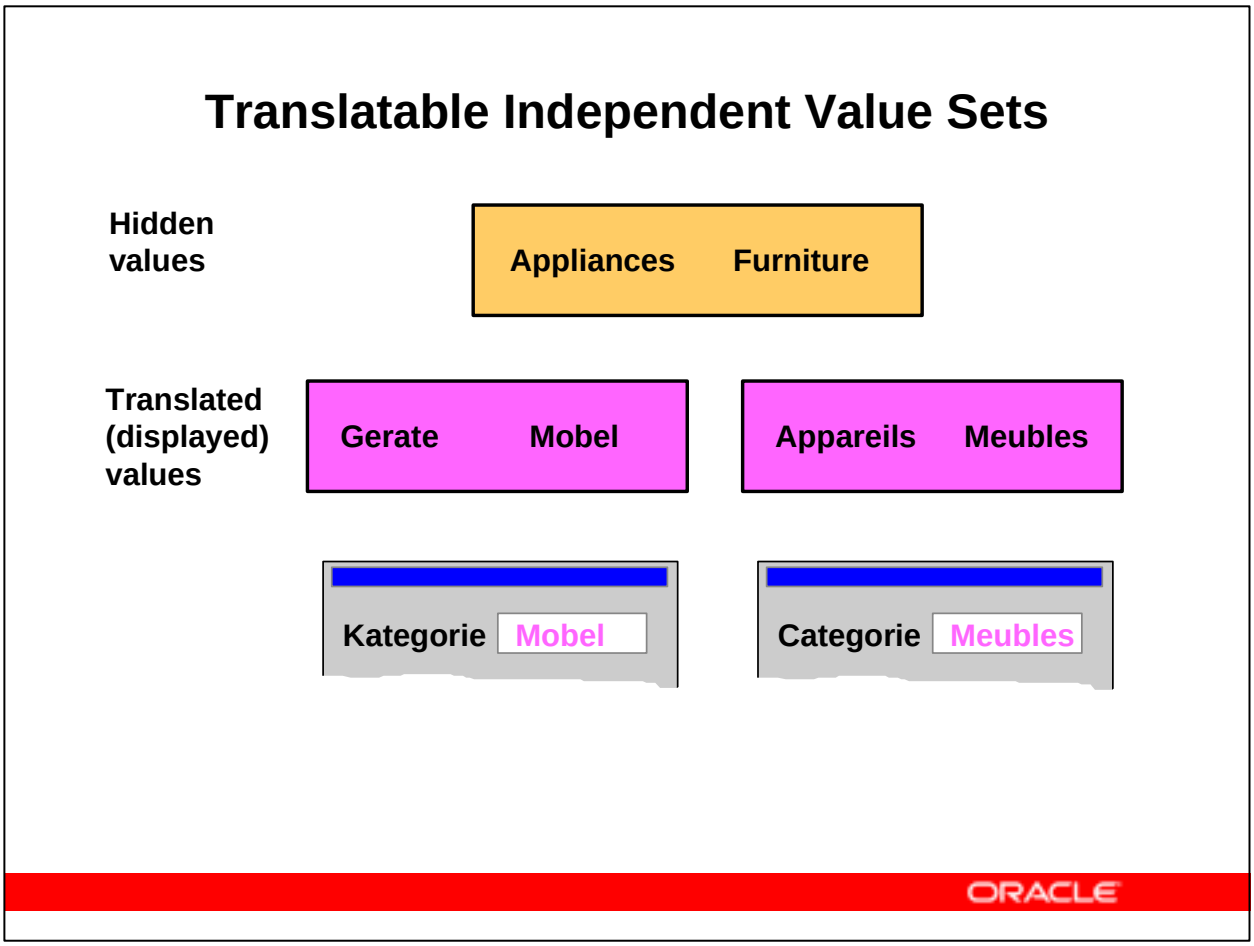
Table-validated value sets are very useful when the set of approved values is very large, changes frequently, and is maintained by the application.



Specialized Validation Types

Two additional validation types are used for more specialized validation needs:

- 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.

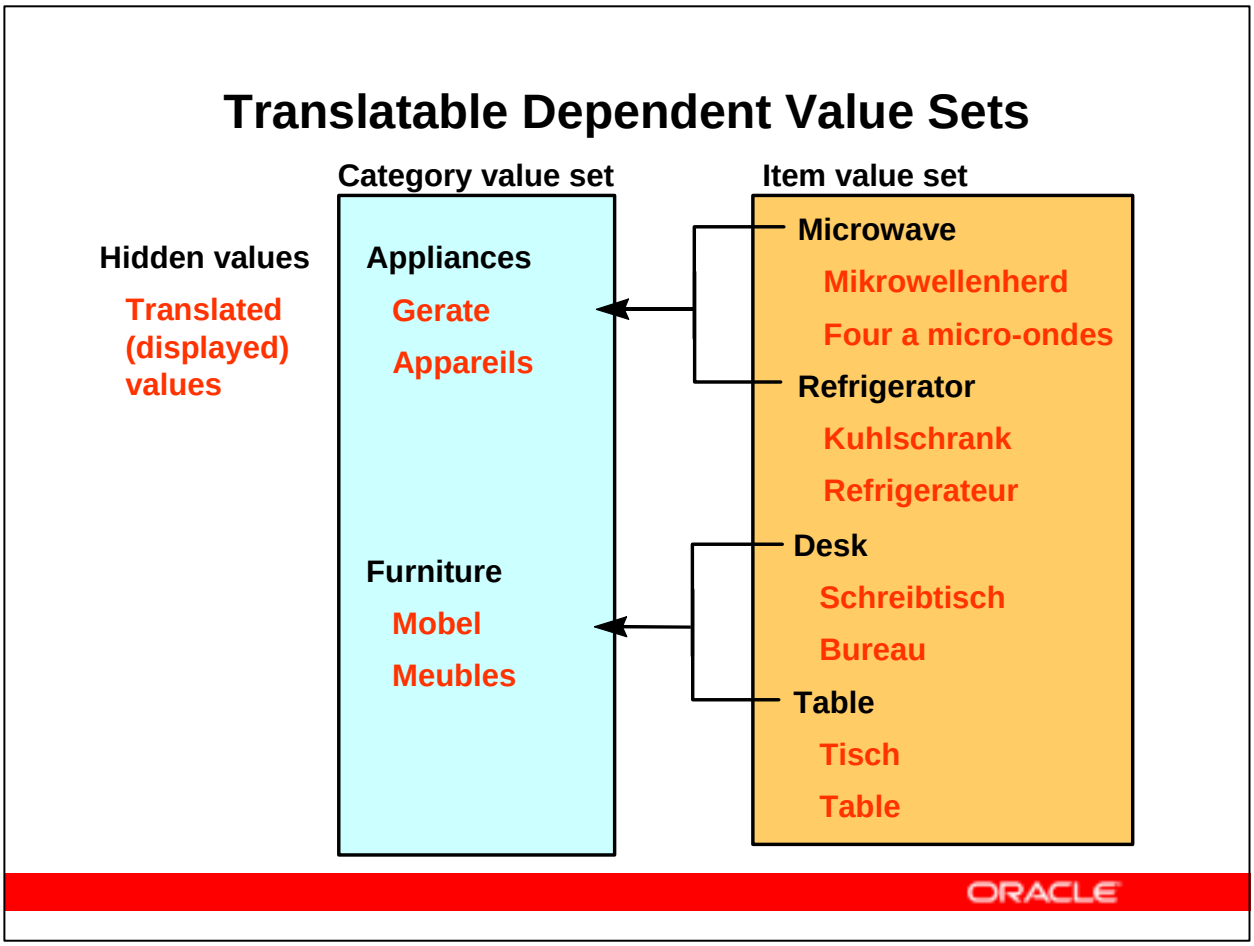


Validation Type Translatable Independent

Translatable Independent value sets are similar to Independent value sets except that translated values can be displayed to the user. Translatable Independent value sets enable you to use hidden values and displayed (translated) values in your value sets. In this way your users can see a value in their preferred languages, yet the values will be validated against a hidden value that is not translated.

A Translatable Independent value set can have only Translatable Dependent value sets dependent on it.

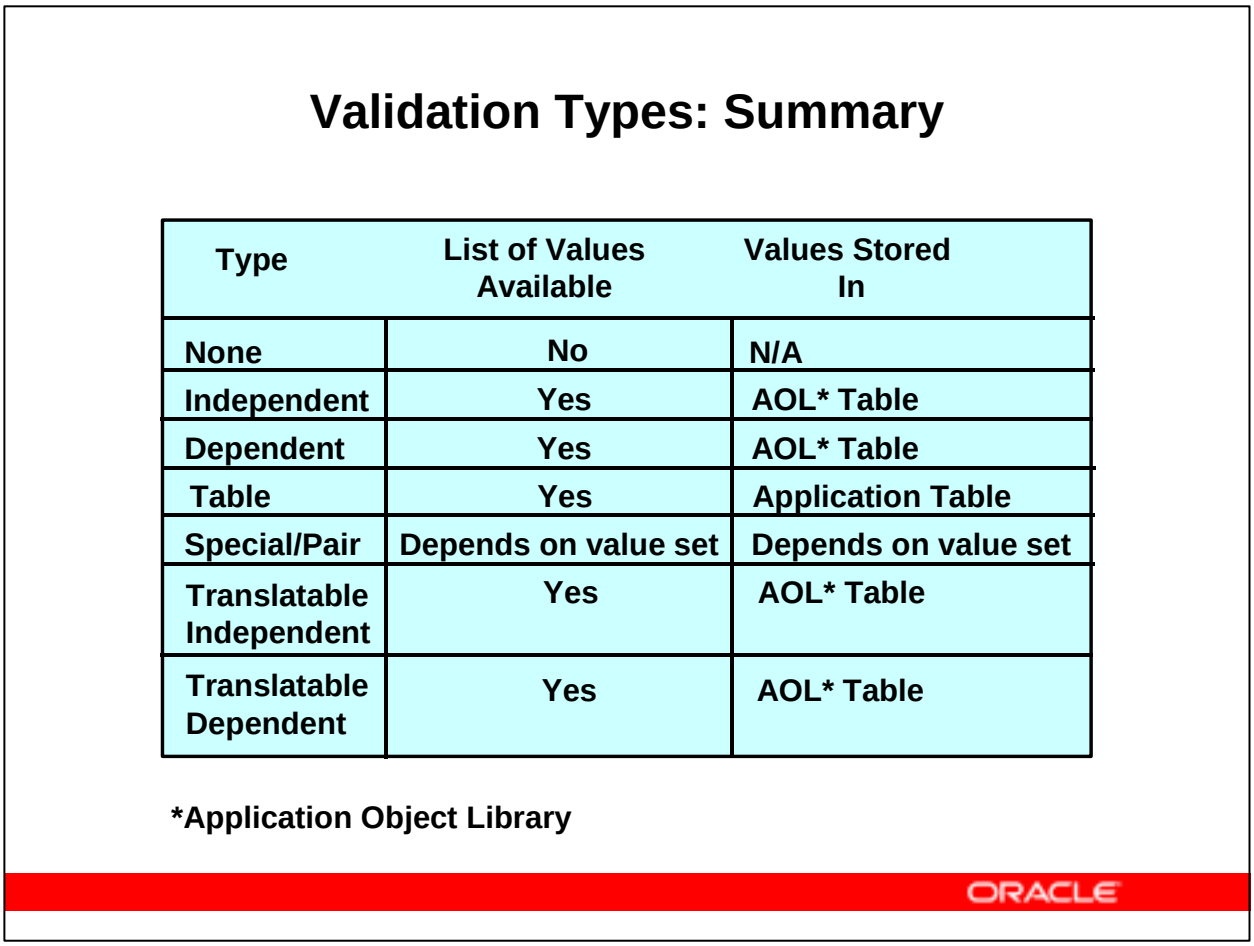
Translatable Dependent Value Sets



Validation Type Translatable Dependent

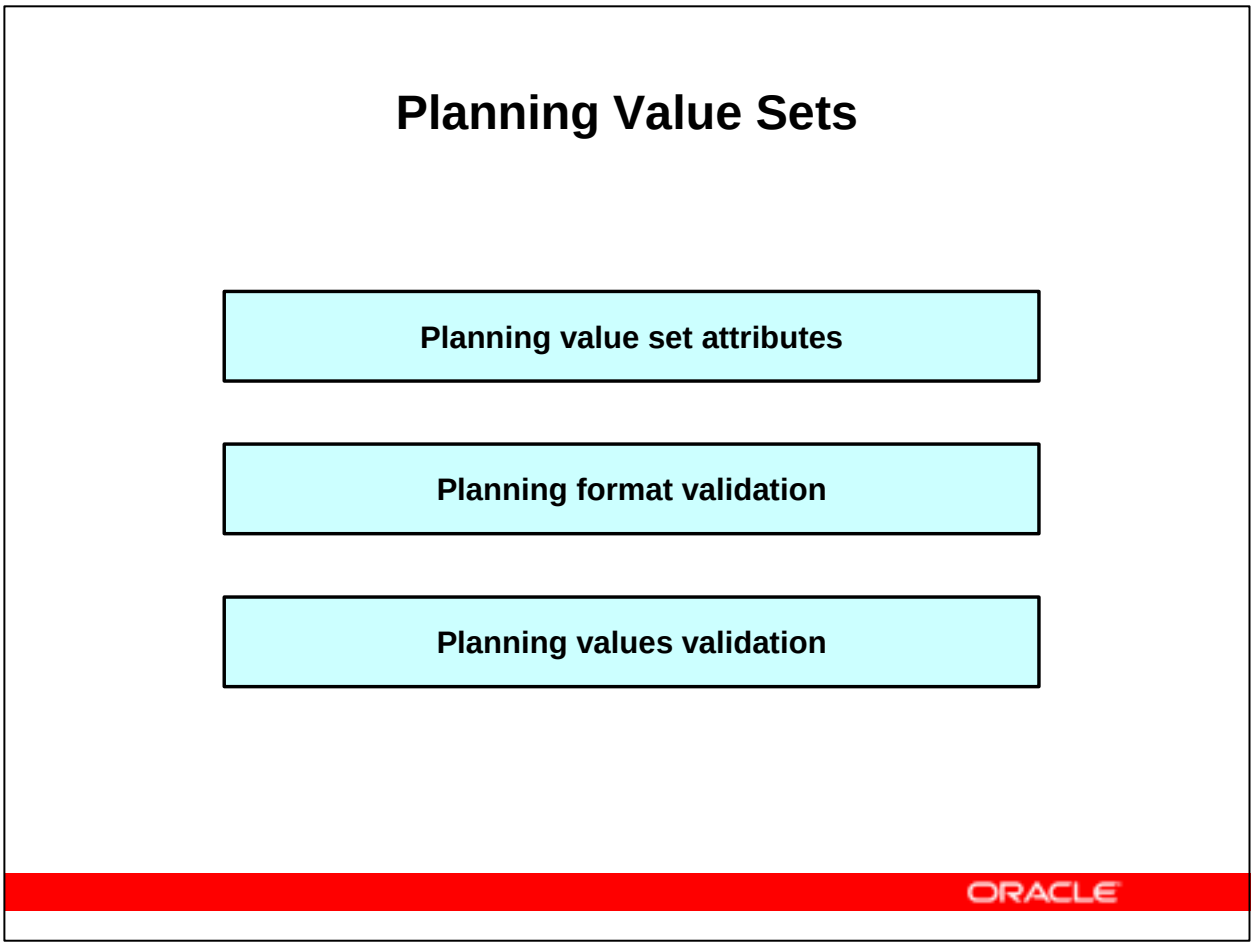
Translatable Dependent value sets are similar to Dependent value sets except that translated values can be displayed to the user. Translatable Dependent value sets enable you to use hidden values and displayed (translated) values in your value sets. In this way your users can see a value in their preferred languages, yet the values will be validated against a hidden value that is not translated.

Translatable Dependent value sets must be dependent on a Translatable Independent value set.



Summary of Validation Types

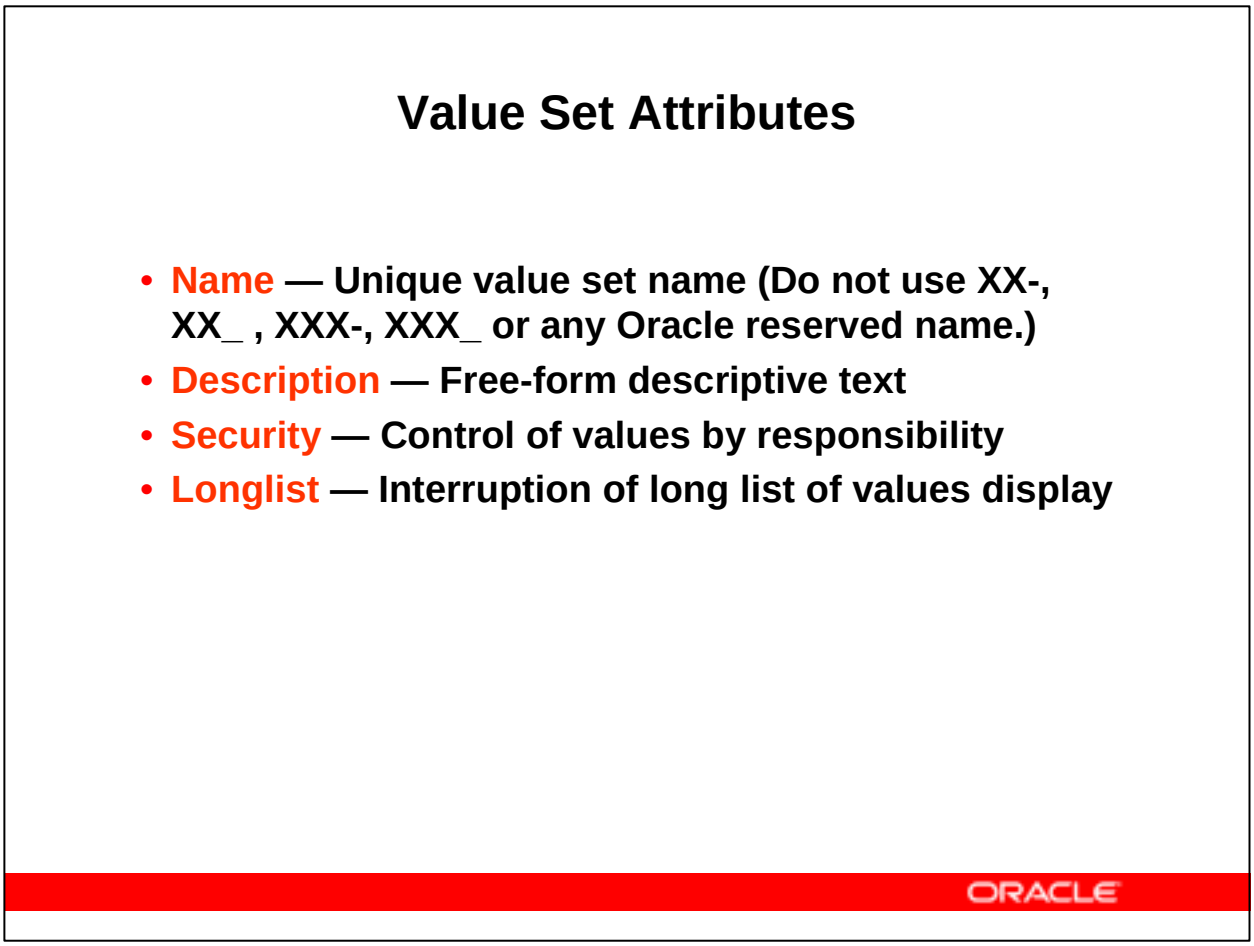
The chart in the slide shows a comparison of the various validation types. It also shows where the lists of acceptable values are stored.



Planning Your Value Set

There are three major areas you should consider when planning your value sets.

- Planning the value set attributes, including the value set name, description, and other attributes
- Planning the format validation to determine how all input must be formatted to be acceptable
- Planning the value validation, including specifying the validation type and any additional validation information, if required by the validation type



Value Set Level Attributes

Value Set Name - If an Oracle application's predefined value set has the same name as a user-defined value set within the same application, the user-defined value set is overridden during an upgrade. Therefore, follow these naming guidelines:

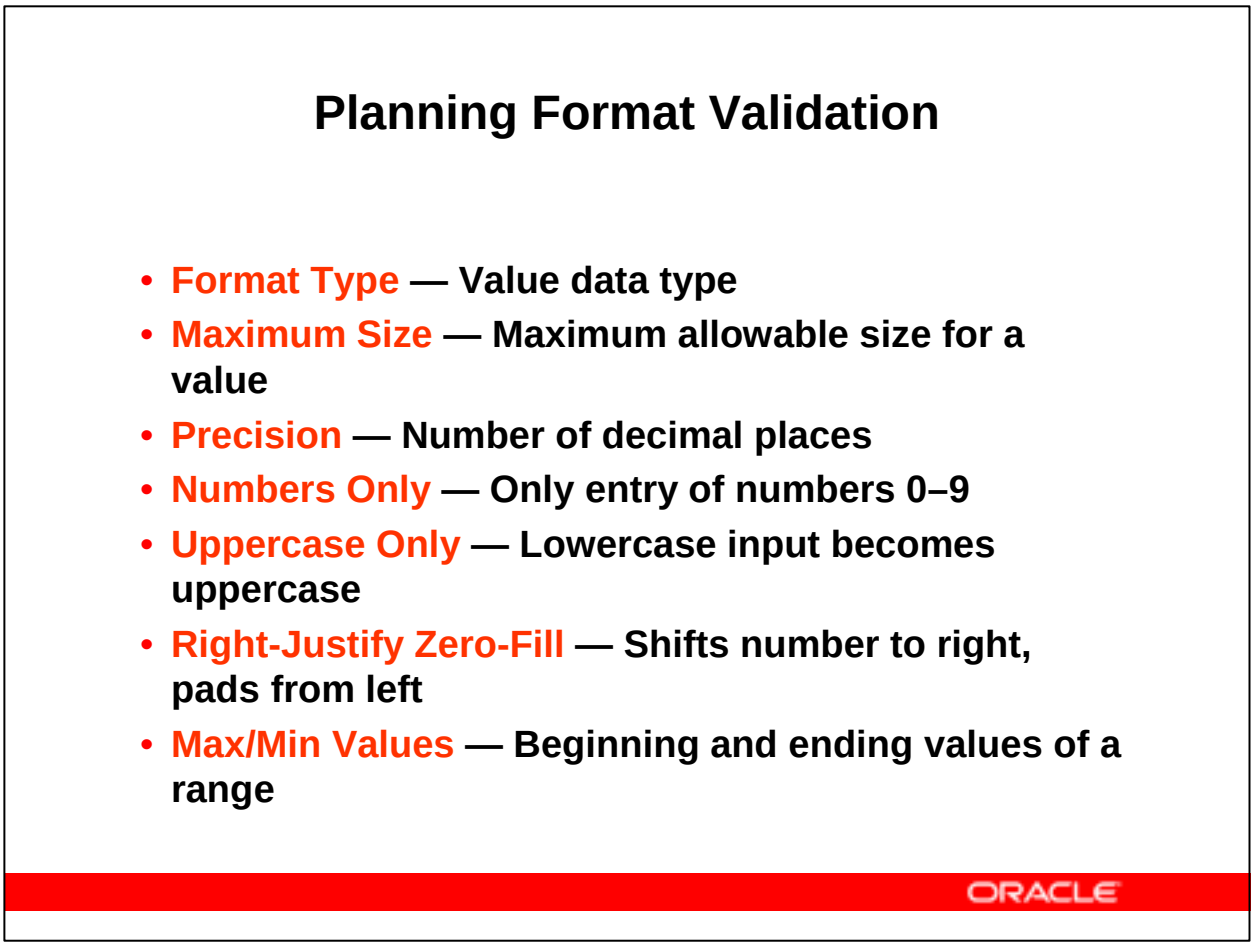
- Do not use the patterns of either two or three characters followed immediately by a hyphen or an underscore. These patterns are reserved by Oracle Applications.
- Do not use spaces in your value set name.
- Include a custom or site prefix in the value set name to make it unique.

Oracle always provides a list of reserved value set names before an upgrade. Be sure to check this list against your existing value sets.

Description - You can give your value sets descriptive text information.

Security - You can define security rules that further control the values available from this value set.

Longlist Option - You can enable the user to interrupt the display of very long lists of values.



Planning Data Format Validation

Format Type - Select the available data types from the list of values.

Maximum Size - Specify the maximum size of the value. The maximum size must be less than or equal to the size of the underlying column in the base application.

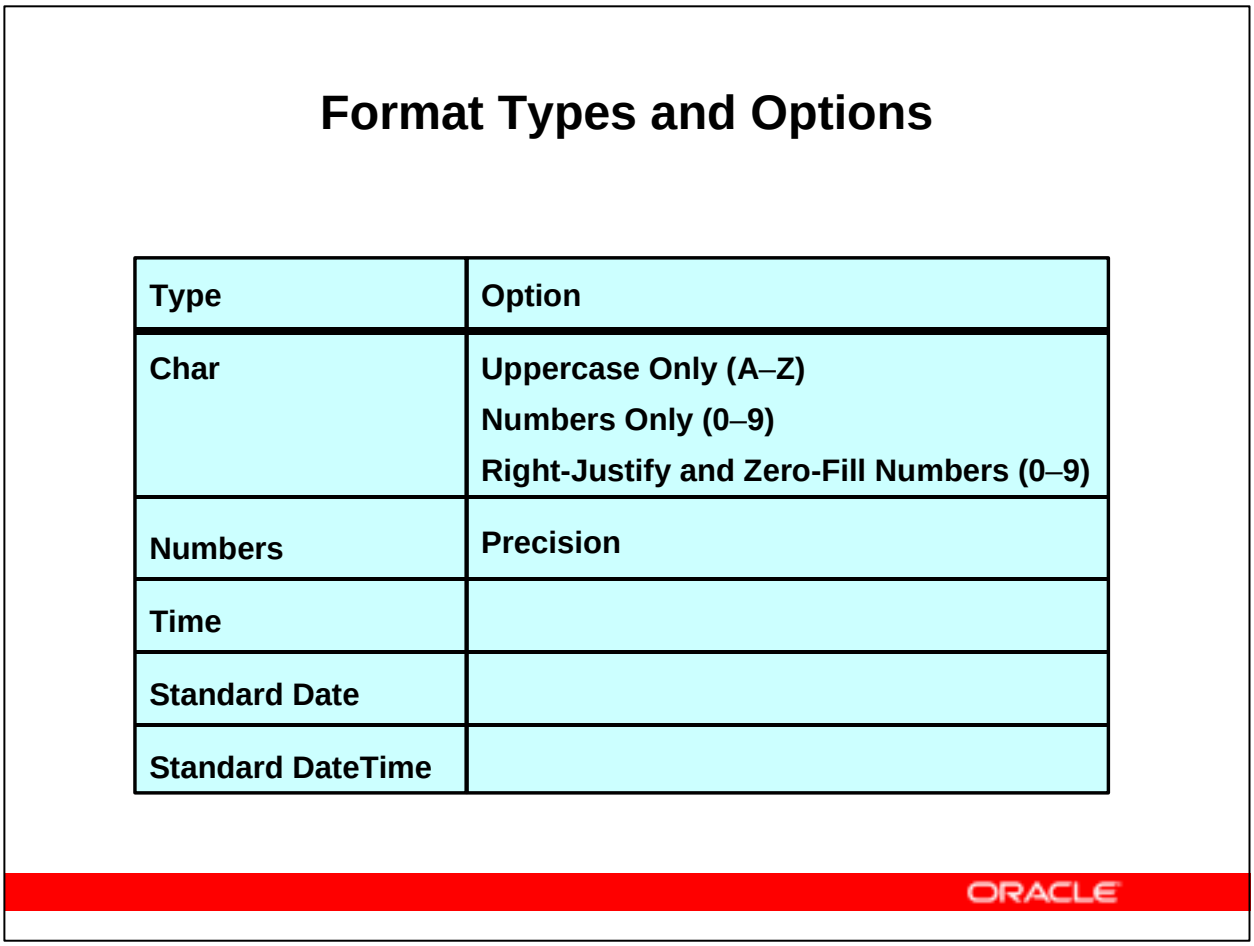
Precision - For numeric data, specify the number of decimal places.

Numbers Only - Select Numbers Only to accept only digits.

Uppercase Only - Select Uppercase Only to force any lowercase input to uppercase.

Right-Justify and Zero-Fill - Select these options to shift number to the right and then pad from the left with zeros. This is an alternate format for alphanumeric numbers.

Maximum/Minimum Value - To define a range of values for this value set, specify a beginning value and an ending value.

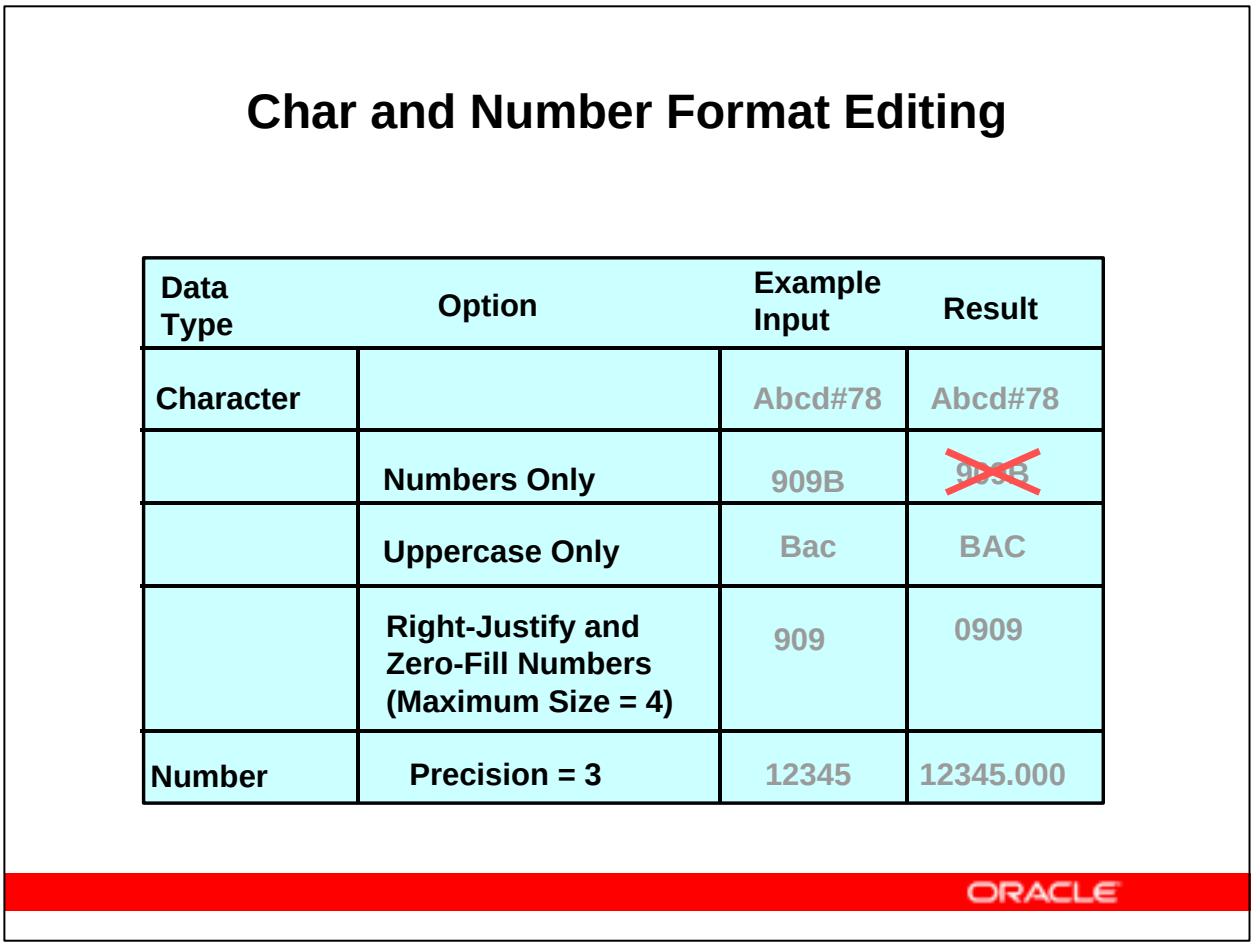


Formatting Types

You can require one of several types of data formatting. When you declare a value set a particular format type, all data entered in the segment that uses this value set must match the requirements of the format.

For some format types, there are additional options.

The various format types and options are discussed on the following pages.



Char and Number Formatting Rules and Their Results

You have several choices about the type of data you can allow in a value set and the editing that is done to that data. Two of the choices are shown in the slide.

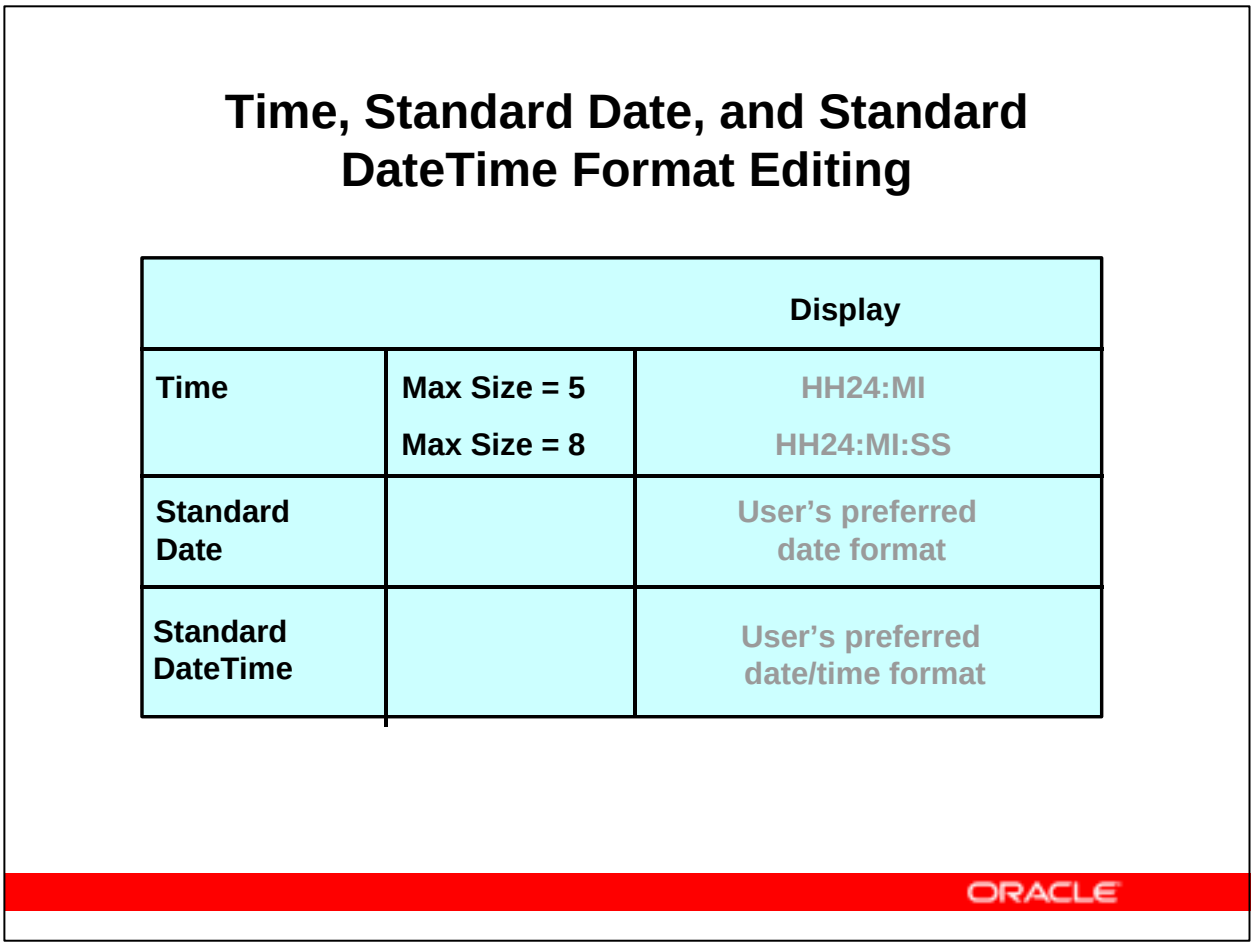
Char Format - Plain character format type accepts uppercase, lowercase, and special characters.

Numbers Only - This format type allows only the digits 0–9 to be input into a field. You will receive an error message if you attempt to enter anything else.

Uppercase Only - This format type converts any lowercase letters entered to uppercase.

Right-Justify and Zero-Fill Numbers - Alphanumeric data is ordinarily left-justified. This option enables you to right-justify numeric values and pad with zeros. Any number entered is converted to the full width of the field size.

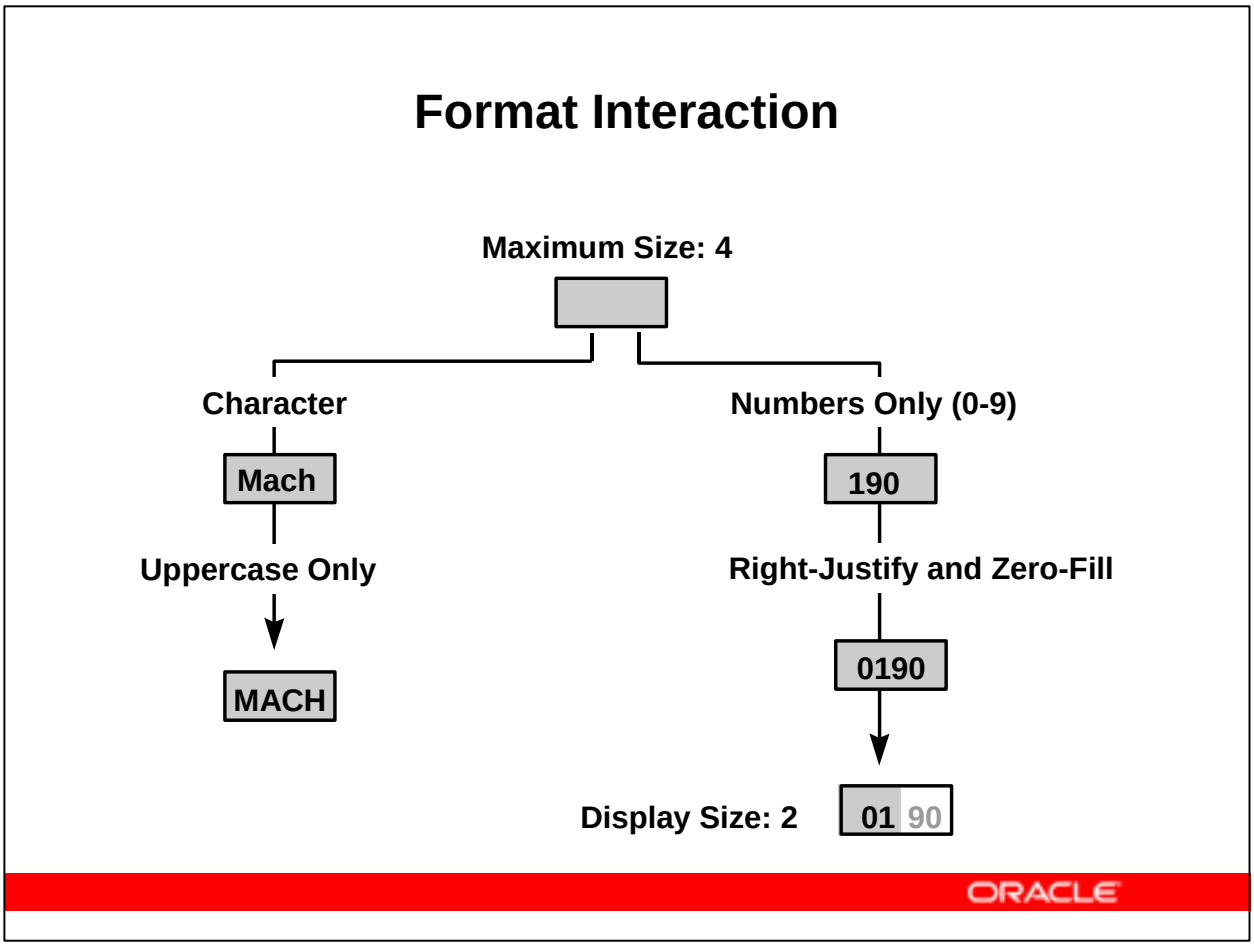
Precision - For data, this numeric format type enables you to specify the number of decimal positions available. If the data entered has more decimals than there are precision positions, the data will be rounded.



Date and Time Format Results

The Max Size attribute controls the display of the output for the Time format as shown in the slide. The Standard Date and Standard DateTime formats enforce the user's preferred format.

There are two additional format types: Date and DateTime. However, these exist for backward compatibility only. Any new value sets you create should use Standard Date and Standard DateTime.

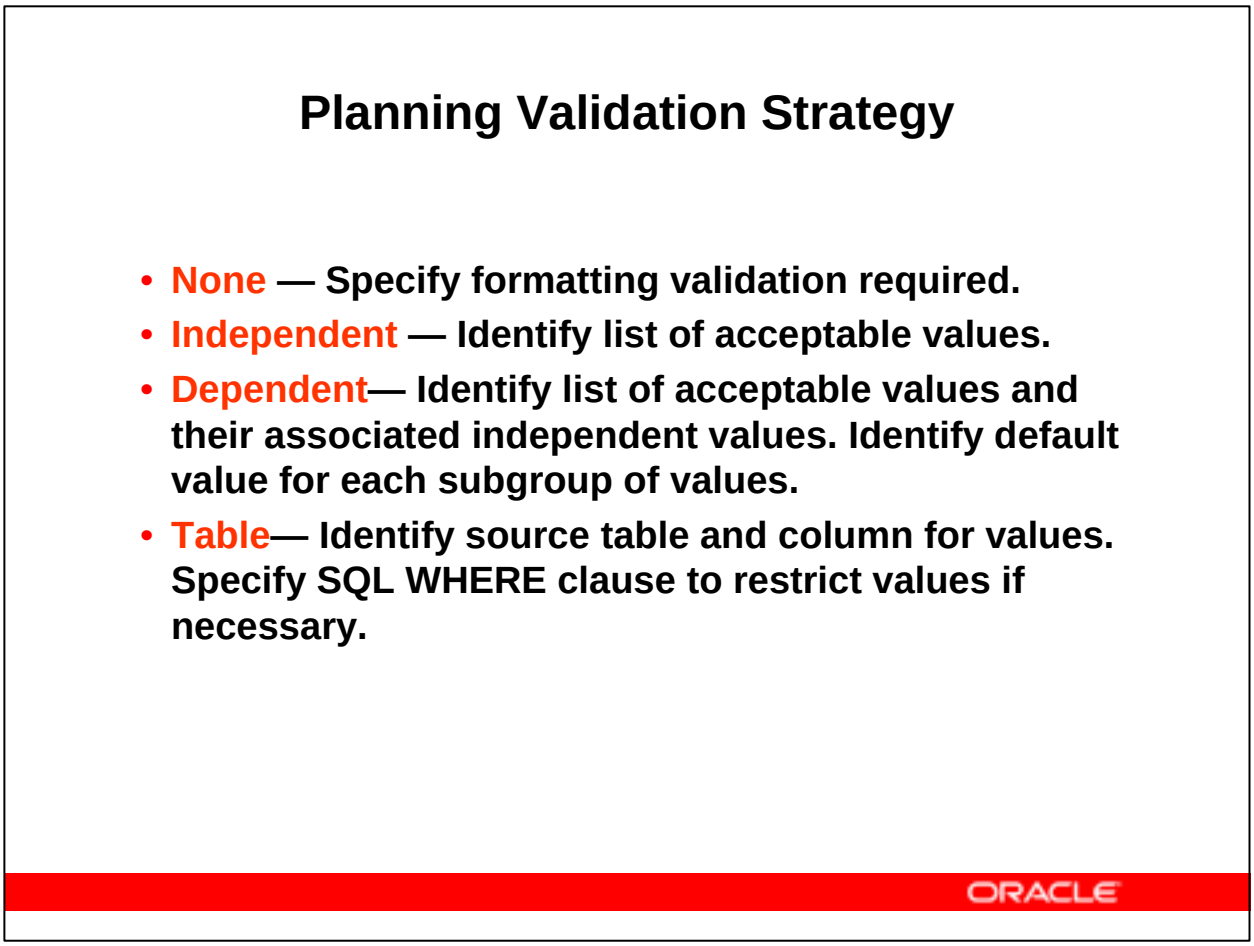


Considering the Interaction of Formatting

When planning your value set, consider the interaction between the value set and the segment that will eventually use it. Optionally, try to keep identical the maximum size defined for the value set and the display size defined for the flexfield segment. Otherwise, the user may have to scroll to see the full value. This is especially important with values formatted with Right-Justify and Zero-Fill.

Formatting Requirements for Translatable Value Sets

- **Translatable value sets must use the format type Char.**
- **The maximum size must be no greater than 150.**
- **The Numbers Only option and the Right-Justify and Zero-Fill Numbers option cannot be used with translatable value sets.**



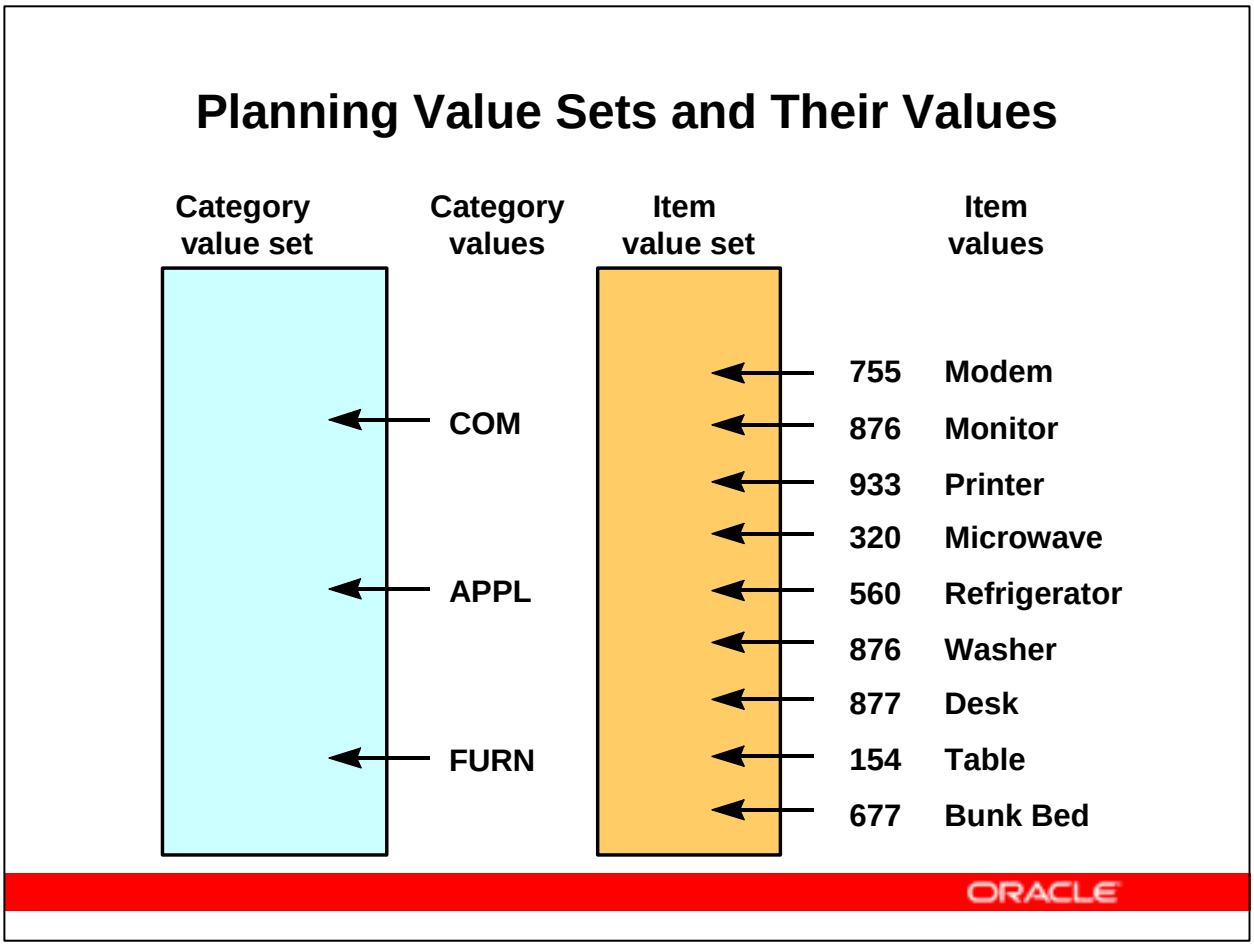
Planning How to Validate Your Values

Which type of validation you use depends on the variability of the data and how the list of acceptable values is processed.

Planning Validation Strategy

- **Special — Identify key flexfield to be used for source.**
- **Pair — Identify two flexfields involved and their interaction.**
- **Translatable Independent — Identify list of acceptable values with their corresponding translated values.**
- **Translatable Dependent — Identify list of acceptable values with their corresponding translated values and associated independent values. Identify default value for each subgroup of values.**

ORACLE



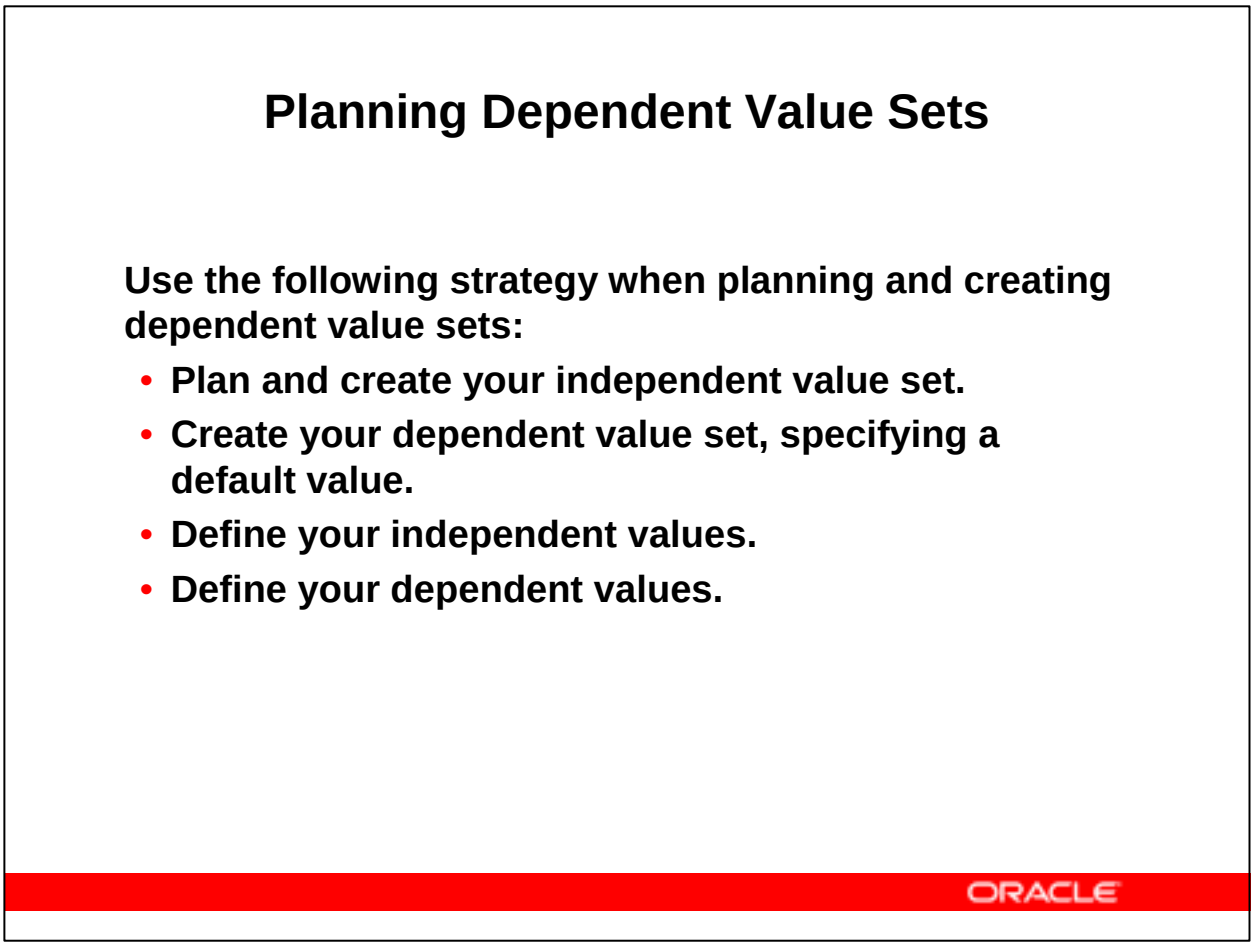
Planning Your Values When Planning Value Sets

For validation types that provide lists of values, plan the values they will process when you are planning the value set itself:

- Consider how the actual values should be formatted. For example, if the values are 01, 02, instead of 1,2, define the value set with Right-Justify Zero-Fill set to Yes.
- Consider the segment size and the display sizes you define.
- When using existing value sets, disable values that have become obsolete (values cannot be deleted), or change the description and reuse the value if possible.

Using Range Features

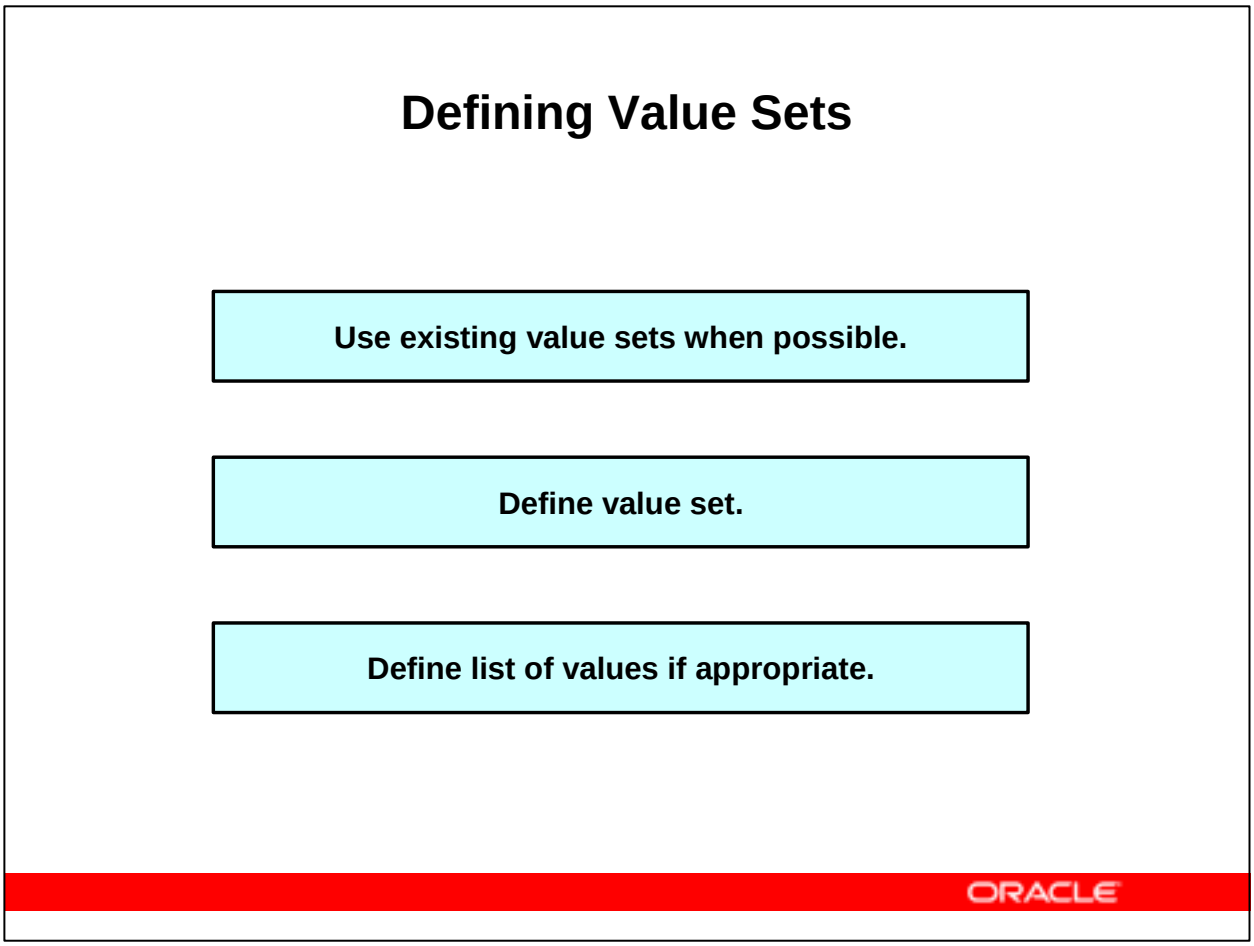
- Group related values in ranges to simplify implementing range-based features such as cross-validation and security rules.
- Allow for future expansion by using large values. For example, use three digits instead of two for a segment.



Planning Values for Dependent Value Sets

Because the values for a dependent value set interact with the values from an independent value set, use the procedure outlined on the slide when planning your dependent value set. Notice that you must define a default value for each group of dependent values.

Translatable dependent value sets behave like dependent value sets except that they must be dependent on a translatable independent value set. A translatable independent value set can have only translatable dependent value sets dependent on it.



Defining Your Value Sets

The procedure for defining value sets is shown in the slide. You should always check the existing value sets to see if there is an existing value set that you can use.

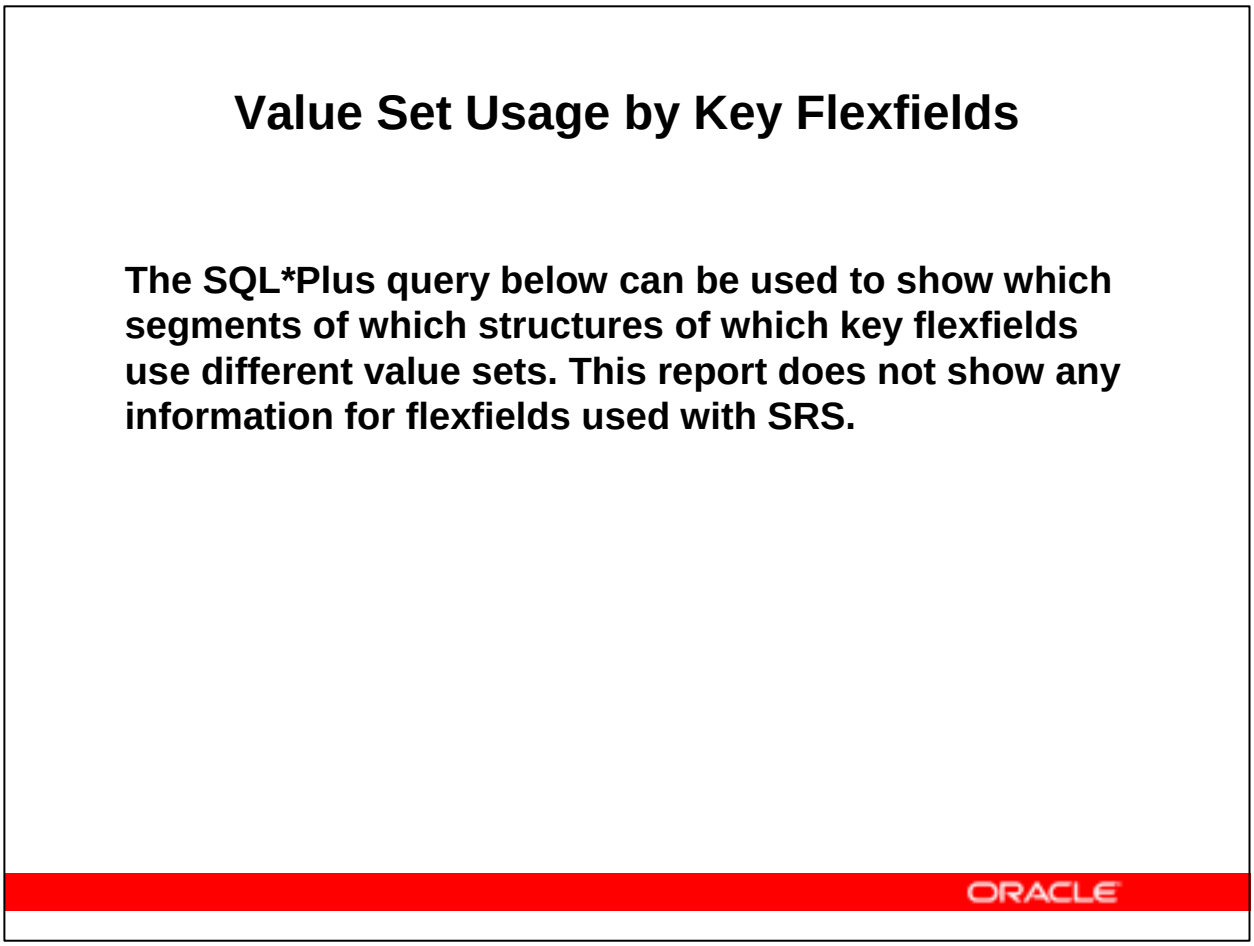
Predefined Value Sets

- **Choosing a predefined value set limits the necessary maintenance of values.**
- **Most predefined value sets are table-validated value sets.**
- **A useful value set is Yes_No, containing the values Yes and No.**
- **Never alter value sets provided by Oracle Applications, especially the SRS value set.**

Using Existing Value Sets

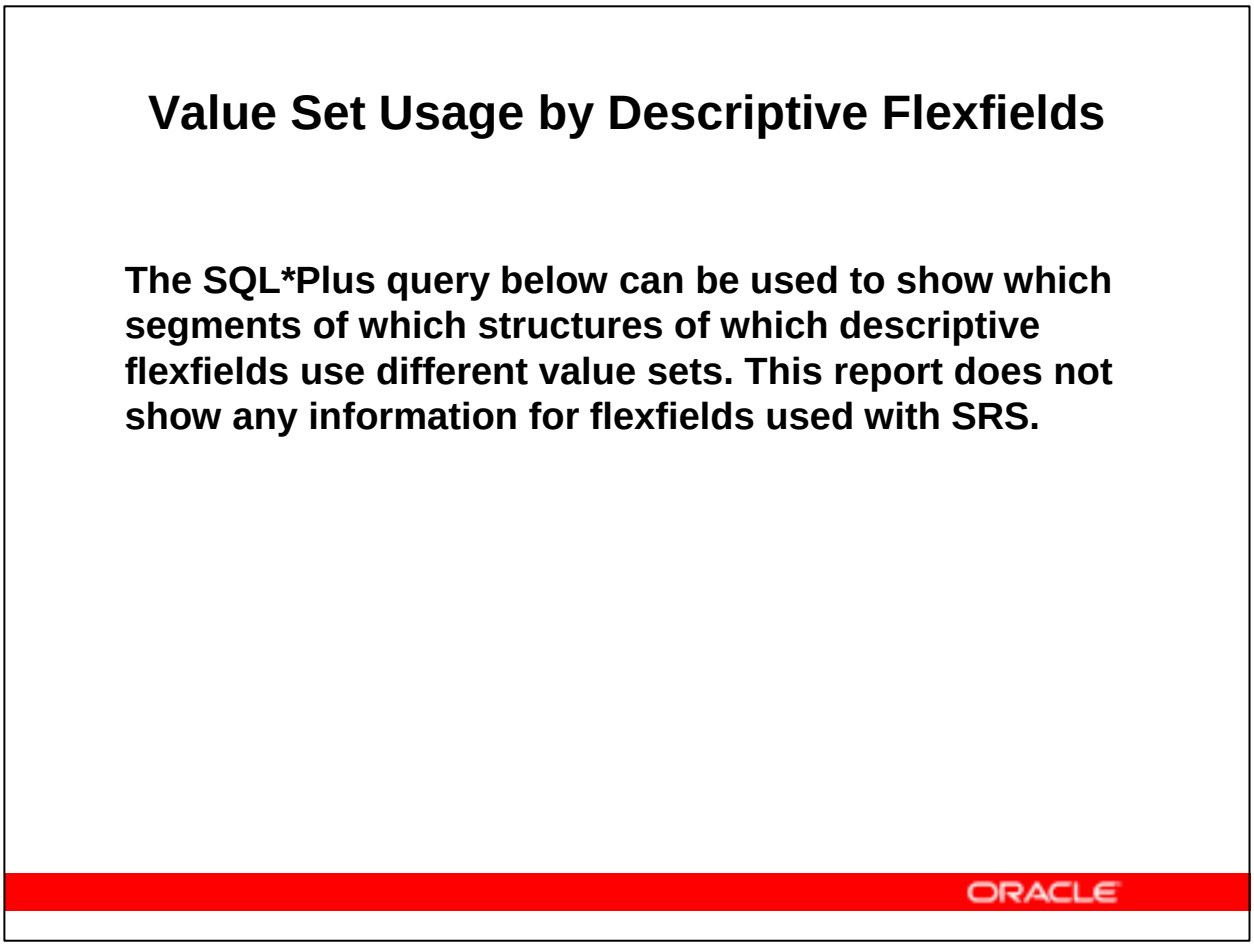
Use the Value Sets window to display a list of existing value sets.

(N) Application—>Validation—>Set



Value Set Usage by Key Flexfields

```
select   flex_value_set_name "Value Set",
         descriptive_flexfield_name "Flexfield",
         descriptive_flex_context_code "Structure",
         end_user_column_name "Column"
from     apps.fnd_descr_flex_column_usages a,
         apps.fnd_flex_value_sets b
where    a.flex_value_set_id = b.flex_value_set_id
         and descriptive_flexfield_name not like '$SRS$%'
order by flex_value_set_name,
         descriptive_flexfield_name,
         descriptive_flex_context_code;
```



Value Set Usage by Descriptive Flexfields

```
col 'value set' for a40
col flexfield for a40
col structure for a30
col column for a32

select   d.flex_value_set_name "Value Set",
         a.id_flex_name "Flexfield",
         e.id_flex_structure_name "Structure",
         c.segment_name "Segment"
from     apps.fnd_id_flexs a,
         apps.fnd_id_flex_structures b,
         apps.fnd_id_flex_segments c,
         apps.fnd_flex_value_sets d,
         apps.fnd_id_flex_structures_tl e
```

```
where    a.application_id  = b.application_id
         and a.id_flex_code = b.id_flex_code
         and b.application_id = c.application_id
         and b.id_flex_code = c.id_flex_code
         and b.id_flex_num = c.id_flex_num
         and c.flex_value_set_id = d.flex_value_set_id
         and c.application_id = e.application_id
         and c.id_flex_code = e.id_flex_code
order by d.flex_value_set_name, a.id_flex_name,
         e.id_flex_structure_name;
```

Defining a New Value Set

Use the Value Sets window to enter:

- **Value Set Name and attributes**
- **Format Validation options**
- **Validation Type**

Click the Edit Information button to enter information required for validation types Dependent, Translatable Dependent, Table, Special, or Pair.

(N) Application—>Validation—>Set

ORACLE

Defining a Dependent or Translatable Dependent Value Set

Use the Dependent Value Set Information window to enter:

- **Independent Value Set**
- **Dependent Default Value**

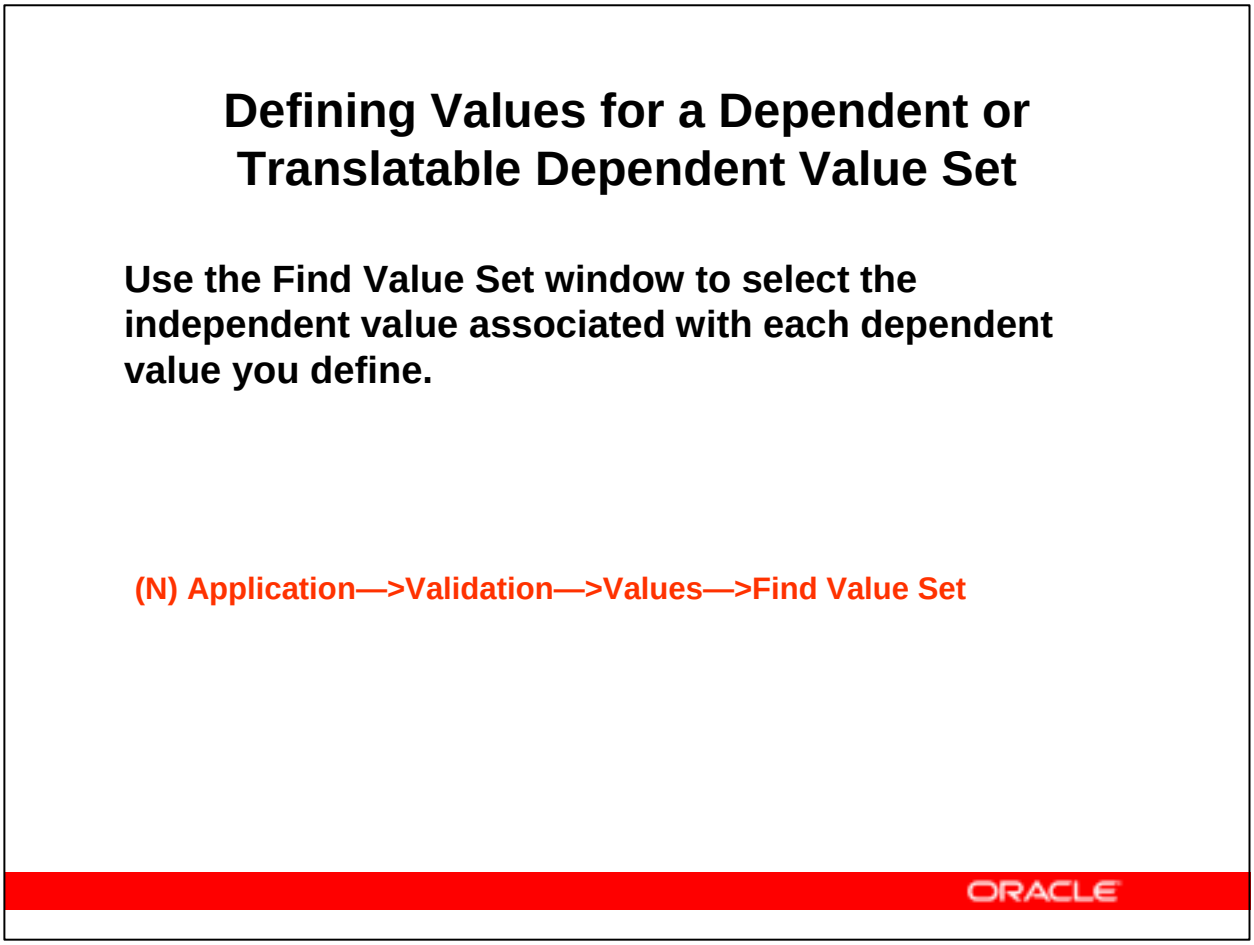
(N) Application—>Validation—>Set (B) Edit Information

Selecting a Value Set for Value Definition

Use the Find Value Set window to select the value set for which you want to define values in the Segment Values window.

(N) Application—>Validation—>Values—>Find Value Set

ORACLE



Defining Values for a Value Set

Use the Values, Effective region of the Segment Values window to enter:

- **Value**
- **Translated Value (translatable value sets only)**
- **Description**
- **Enabled**
- **From/To effective dates**

(N) Application—>Validation—>Values—>Find Value Set (B) Find

ORACLE

Modifying Value Definitions

- **You cannot change or delete values after they are defined.**
- **You can change the value description or the translated value to reuse an existing value.**
- **You can disable or limit the effective dates of a value that is no longer in use.**

ORACLE

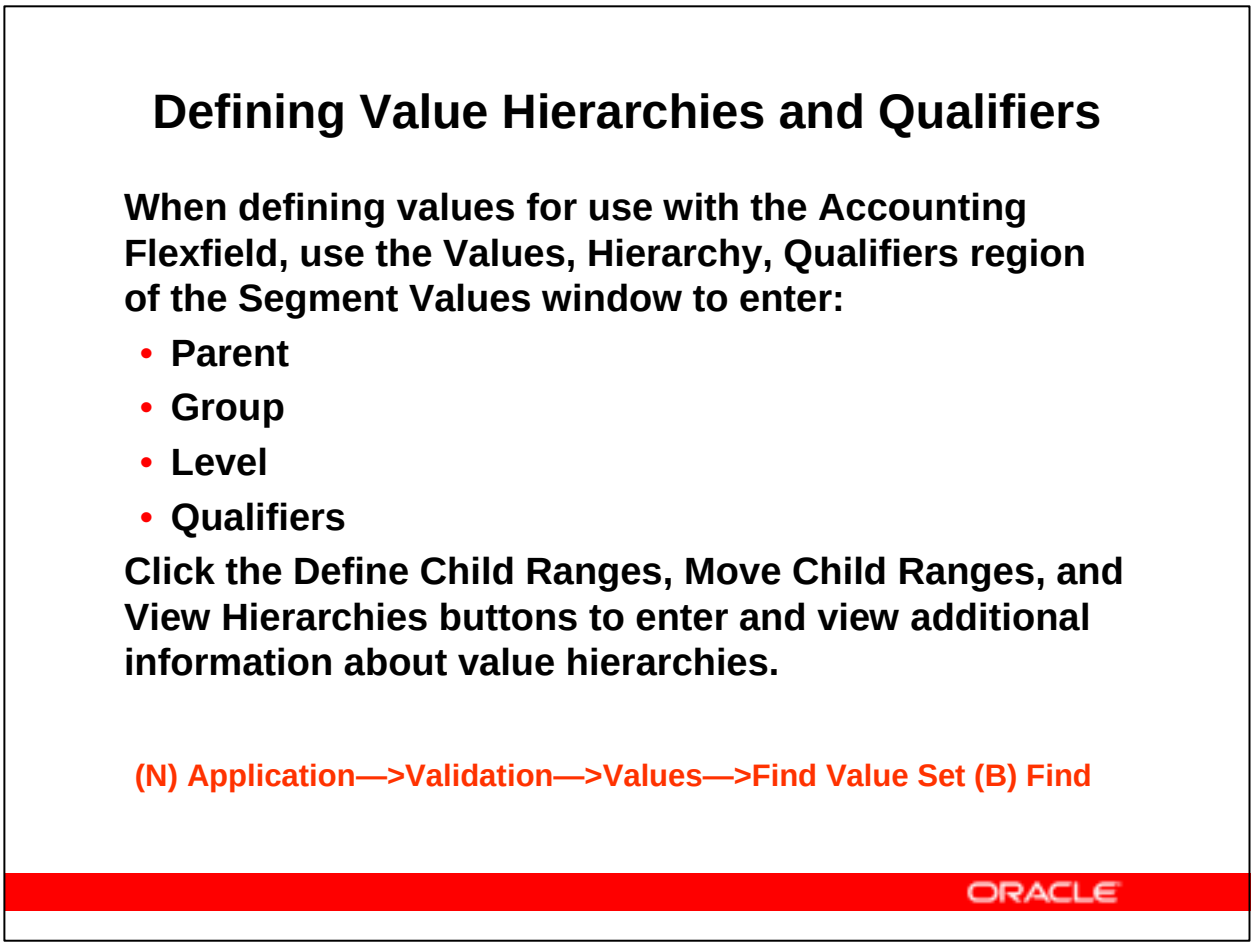
Value Hierarchies

- **Only Oracle General Ledger and Oracle Public Sector General Ledger use value hierarchies, and only with the Accounting Flexfield.**
- **You can use parent-child relationships for reporting and other application purposes.**
- **You create parent-child relationships by defining a range of child values that belong to a parent value.**
- **You can use rollup groups to group related parent values for creating summary templates for reporting and other application purposes.**

ORACLE

Value Hierarchies

- **You define rollup groups in the Rollup Groups window and assign parent values to the rollup groups when you define the parent values.**
- **You can also use the Account Hierarchy Editor in the Applications Desktop Integrator to define rollup groups, graphically create parent-child relationships, and assign parent values to rollup groups.**



Defining Value Hierarchies and Qualifiers

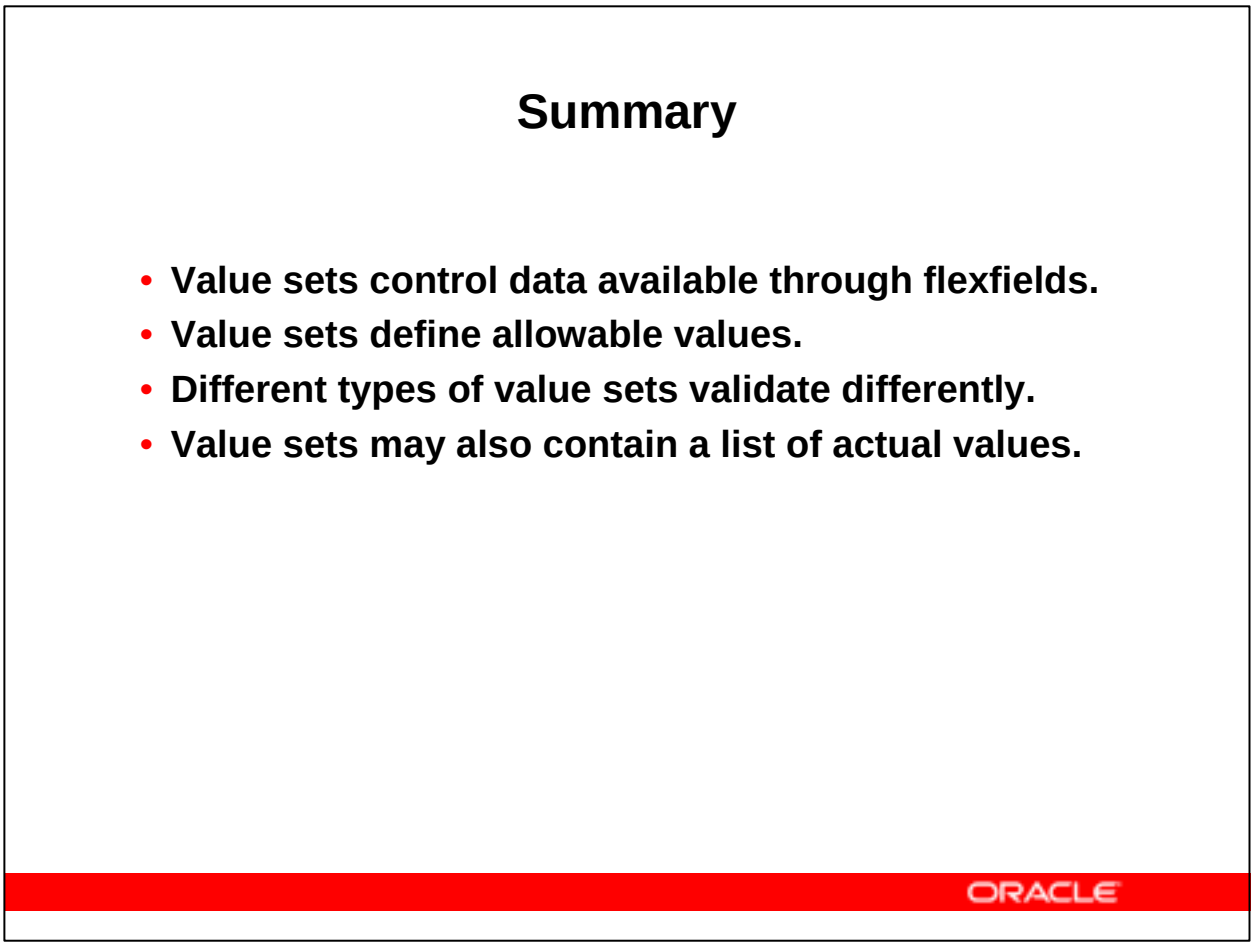
Parent - Select this check box for parent values. Clear this check box for values that are not parents of any other value.

Group - For parent values, select the rollup group to which you want to assign the value.

Level - Enter a description of this value's relative level in your hierarchy structure. This level description is for your purposes only.

Qualifiers - Navigate to the Qualifiers field to display the Segment Qualifiers window and assign segment qualifiers to your values as appropriate. Examples of segment qualifiers for the Accounting Flexfield include:

- Allow Budgeting
- Allow Posting
- Account Type



Using Value Sets: Summary

Use value sets to control the data entered and displayed through flexfield segments. A value set is a definition of allowable values. There are several types of value sets. Each type differs in the amount and type of validation processing it performs. Some types of value sets present the user with a list of actual values from which to choose. You can define this list yourself or retrieve it from an application table.