

# **11i System Administrator Fundamentals**

**Student Guide**

Course Code D14411GC10

Edition 1.0

Month Year June 2002

Part Number D34840

**ORACLE<sup>®</sup>**

**Copyright © Oracle Corporation, 2002. All rights reserved.**

This documentation contains proprietary information of Oracle Corporation. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

**Restricted Rights Legend**

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of the Education Products group of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Worldwide Education Services, Oracle Corporation, 500 Oracle Parkway, Box SB-6, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

Oracle and all references to Oracle Products are trademarks or registered trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

**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

---

<b>Introduction.....</b>	<b>1-1</b>
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
<b>Managing Application Security .....</b>	<b>2-1</b>
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
<b>Managing Concurrent Programs and Reports.....</b>	<b>3-1</b>
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
<b>Administering Concurrent Managers .....</b>	<b>4-1</b>
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
<b>Managing Profile Options .....</b>	<b>5-1</b>
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
<b>Incorporating a Custom Program .....</b>	<b>6-1</b>
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

<b>Auditing System Resources .....</b>	<b>7-1</b>
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
<b>Managing Printers .....</b>	<b>8-1</b>
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
<b>Applications DBA Duties .....</b>	<b>9-1</b>
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
<b>Incorporating Custom Help Files .....</b>	<b>10-1</b>
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
<b>Administering Folders .....</b>	<b>11-1</b>
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
<b>Managing Document Sequences .....</b>	<b>12-1</b>
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
<b>Overview of Flexfields .....</b>	<b>13-1</b>
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
<b>Using Value Sets.....</b>	<b>14-1</b>
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
<b>Defining Key Flexfields.....</b>	<b>15-1</b>
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
<b>Using Additional Flexfield Options .....</b>	<b>16-1</b>
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
<b>Defining Descriptive Flexfields .....</b>	<b>17-1</b>
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
<b>Using Advanced Validation Capabilities.....</b>	<b>18-1</b>
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 \$PROFILE\$.....	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
<b>Overview of Workflow.....</b>	<b>19-1</b>
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
<b>Oracle Workflow Components .....</b>	<b>20-1</b>
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
<b>Viewing and Responding to Notifications .....</b>	<b>21-1</b>
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
<b>Monitoring Workflow Processes.....</b>	<b>22-1</b>
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
<b>Purging Workflow Data .....</b>	<b>23-1</b>
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
<b>Overview of Oracle Alert .....</b>	<b>24-1</b>
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
<b>Identifying Implementation Issues and Responsibilities for Oracle Alert .....</b>	<b>25-1</b>
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

---

## Typographic Conventions in Text

Convention	Element	Example
Bold italic	Glossary term (if there is a glossary)	The <i>algorithm</i> 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: <code>debug.set ('I', 300);</code> Directory: <code>bin</code> (DOS), <code>\$FMHOME</code> (UNIX) Filename: Locate the <code>init.ora</code> file. Password: User <code>tiger</code> as your password. Pathname: Open <code>c:\my_docs\projects</code> URL: Go to <code>http://www.oracle.com</code> User input: Enter <code>300</code> Username: Log on as <code>scott</code>
Initial cap	Graphics labels (unless the term is a proper noun)	Customer address ( <i>but</i> Oracle Payables)
Italic	Emphasized words and phrases, titles of books and courses, variables	Do <i>not</i> save changes to the database. For further information, see <i>Oracle7 Server SQL Language Reference Manual</i> . Enter <code>user_id@us.oracle.com</code> , where <i>user_id</i> 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 <code>SELECT</code> command to view information stored in the <code>LAST_NAME</code> column of the <code>EMP</code> 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 <i>role</i>
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.

# Defining Key Flexfields

## Chapter 15

# Defining Key Flexfields

ORACLE

### Objectives

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

- **Explain intelligent keys and provide examples**
- **Identify key flexfields that are required by Oracle Applications**
- **Explain the purpose of flexfield qualifiers and optional key flexfield features**
- **Design a key flexfield structure**
- **Define the key flexfield structure and segment attributes**
- **Define flexfield qualifiers and segment qualifiers**
- **Implement optional features as needed**

ORACLE

### Lesson Topics

At the end of this lesson, you should be able to:

- Explain intelligent keys and provide examples
- Identify key flexfields that are required by Oracle Applications
- Explain the purpose of flexfield qualifiers and optional key flexfield features
- Design a key flexfield structure
- Define the key flexfield structure and segment attributes
- Define flexfield qualifiers and segment qualifiers
- Implement optional features as needed

### Overview

- **Key flexfields as intelligent keys**
- **Key flexfields' use of code combinations**
- **Key flexfield qualifiers**
- **Additional key flexfield options**
- **Defining a key flexfield structure**
- **Specifying flexfield qualifiers and segment qualifiers if needed**
- **Using optional key flexfield features**

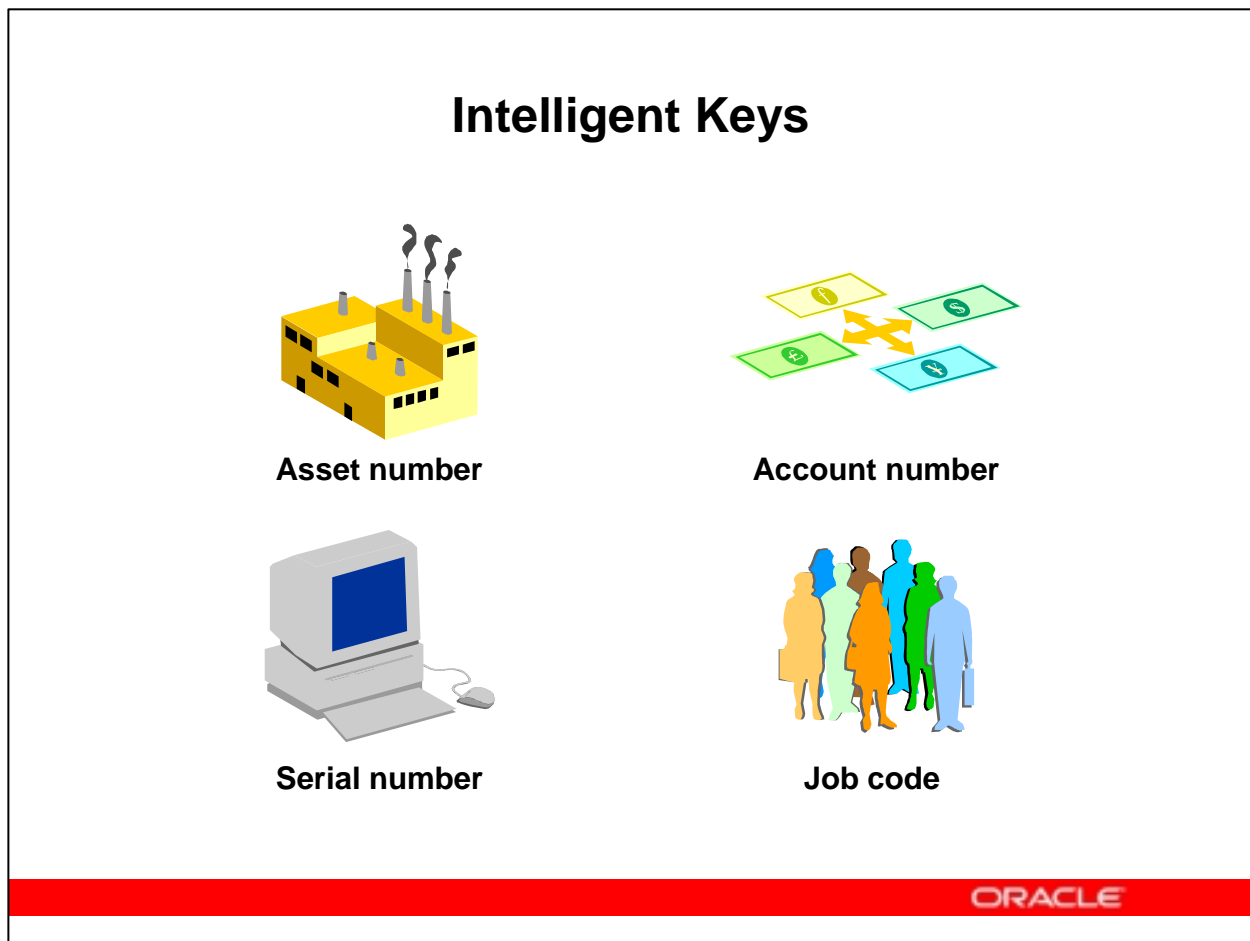
ORACLE

### Lesson Overview

Key flexfields create unique identifiers for use internally by Oracle Applications. The combinations of values for the segments of a key flexfield identify entities. These code combinations are used throughout Oracle Applications.

Depending on which key flexfield you are defining, you might need to specify flexfield qualifiers and segment qualifiers. There are also several optional features of key flexfields you may wish to implement.

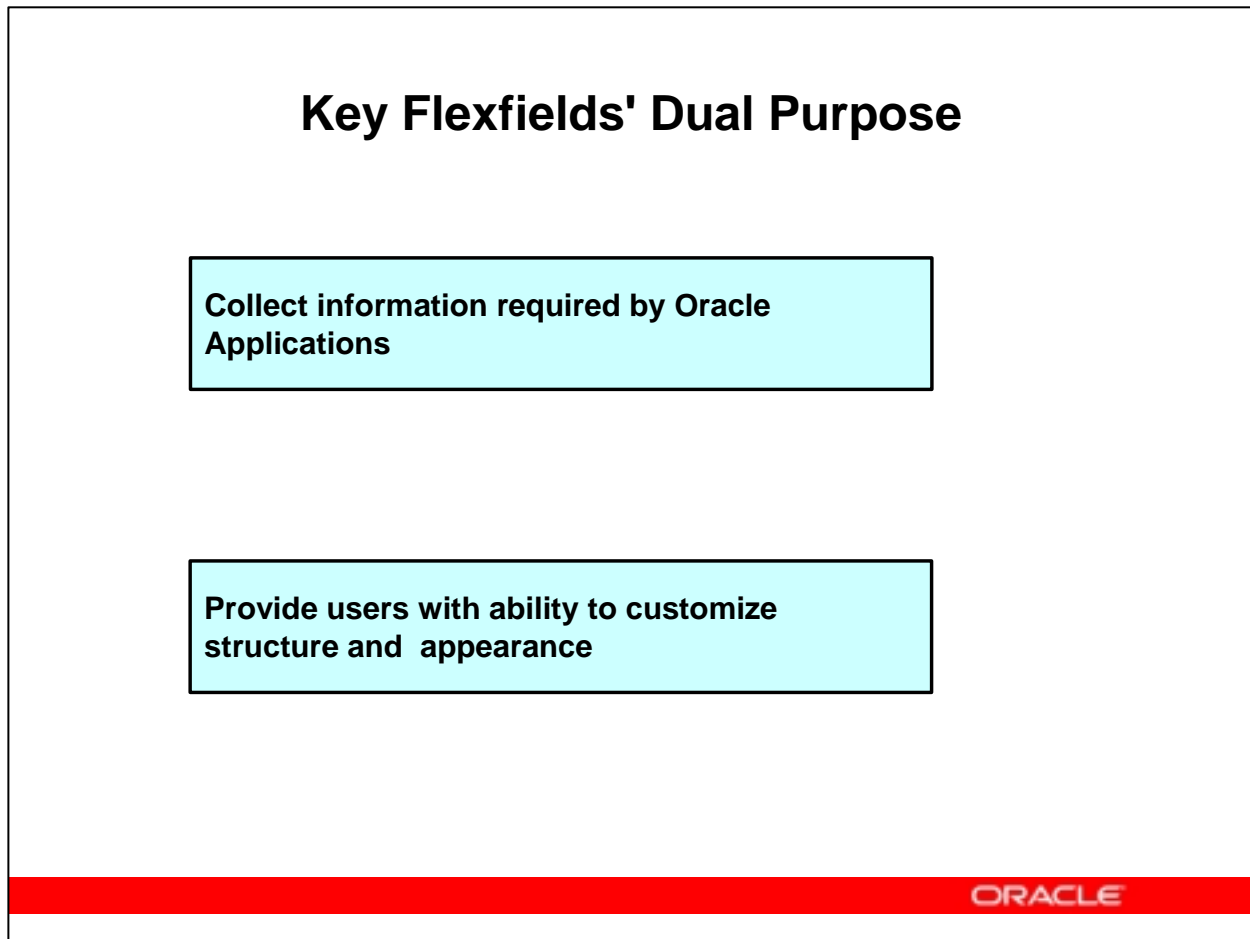
Finally, this lesson covers the mechanics of the definition process.



### Building Intelligent Keys for Oracle Applications

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

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



### Requirements for Key Flexfields

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

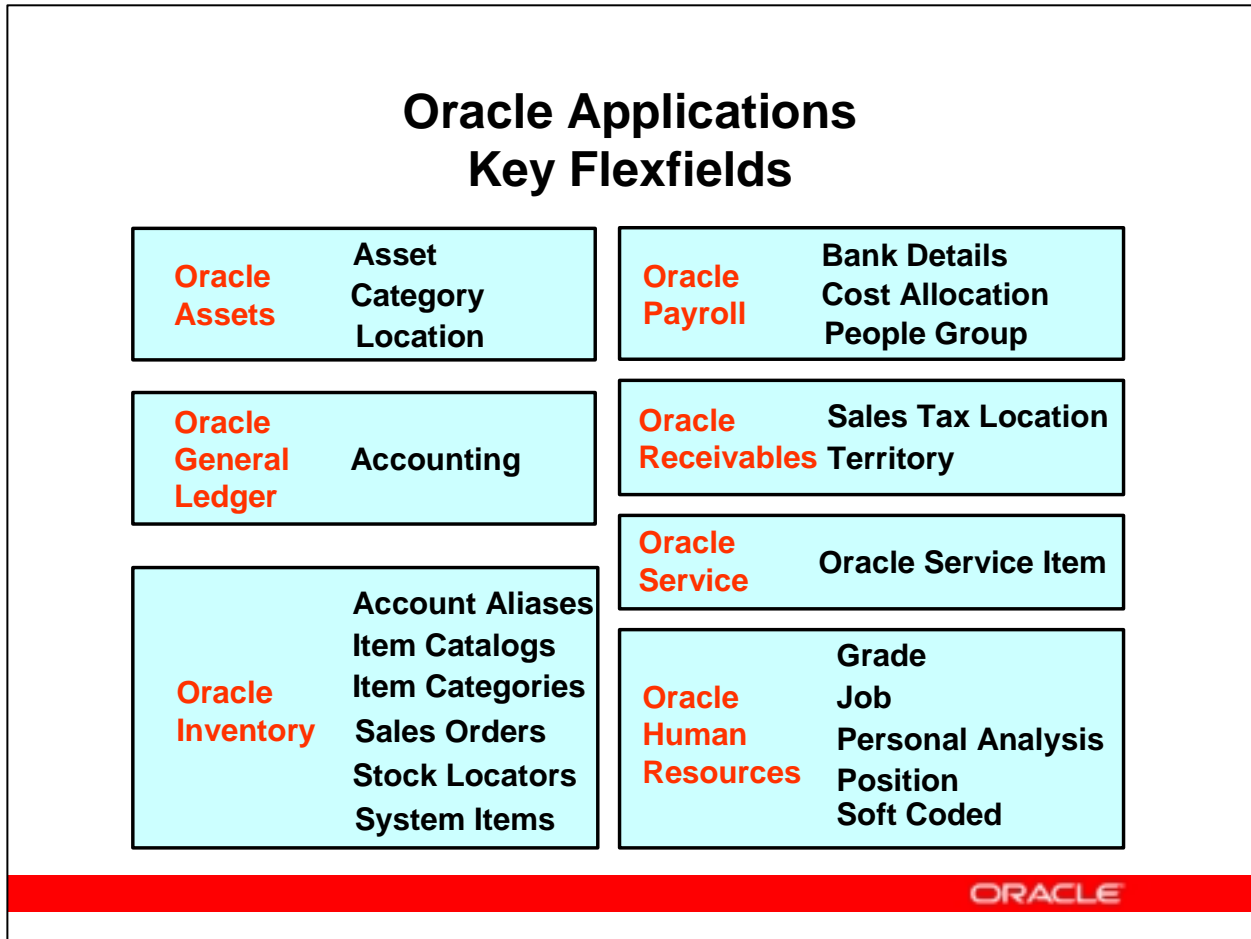
#### **Provide Required Information**

- Provide information needed for reports and processing. For some applications, particular items of information must be identified within the flexfield. For example, Oracle General Ledger requires the balancing segment of the Accounting Flexfield.
- Build unique IDs for use by the applications while giving users meaningful codes.

#### **Provide Customization Capability**

- Tailor the flexfield to the company's business practice instead of changing the practice to fit the flexfield.
- Retain the information the company already keeps.





## Key Flexfields Used by Oracle Applications

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

### Implementing Key Flexfields

- **Identify the target flexfield, the information required by Oracle Applications, and any qualifiers.**
- **Plan the flexfield structure, behavior, and appearance.**
- Define the key flexfield structures.
- Define any value sets required and their values.
- Define security rules when appropriate.
- Define cross-validation rules when appropriate.
- Define shorthand aliases as needed.

ORACLE

#### Implementing Key Flexfields: Procedure

The slide shows the steps involved in implementing a key flexfield. The first two steps are covered in this lesson. The remaining steps are covered in the remainder of the course.

## Key Flexfield Structures

### Key Flexfield Structures

**Business A**

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

**Business B**

Account Entry	
Company	10
Division	203
Department	3003
Account	025

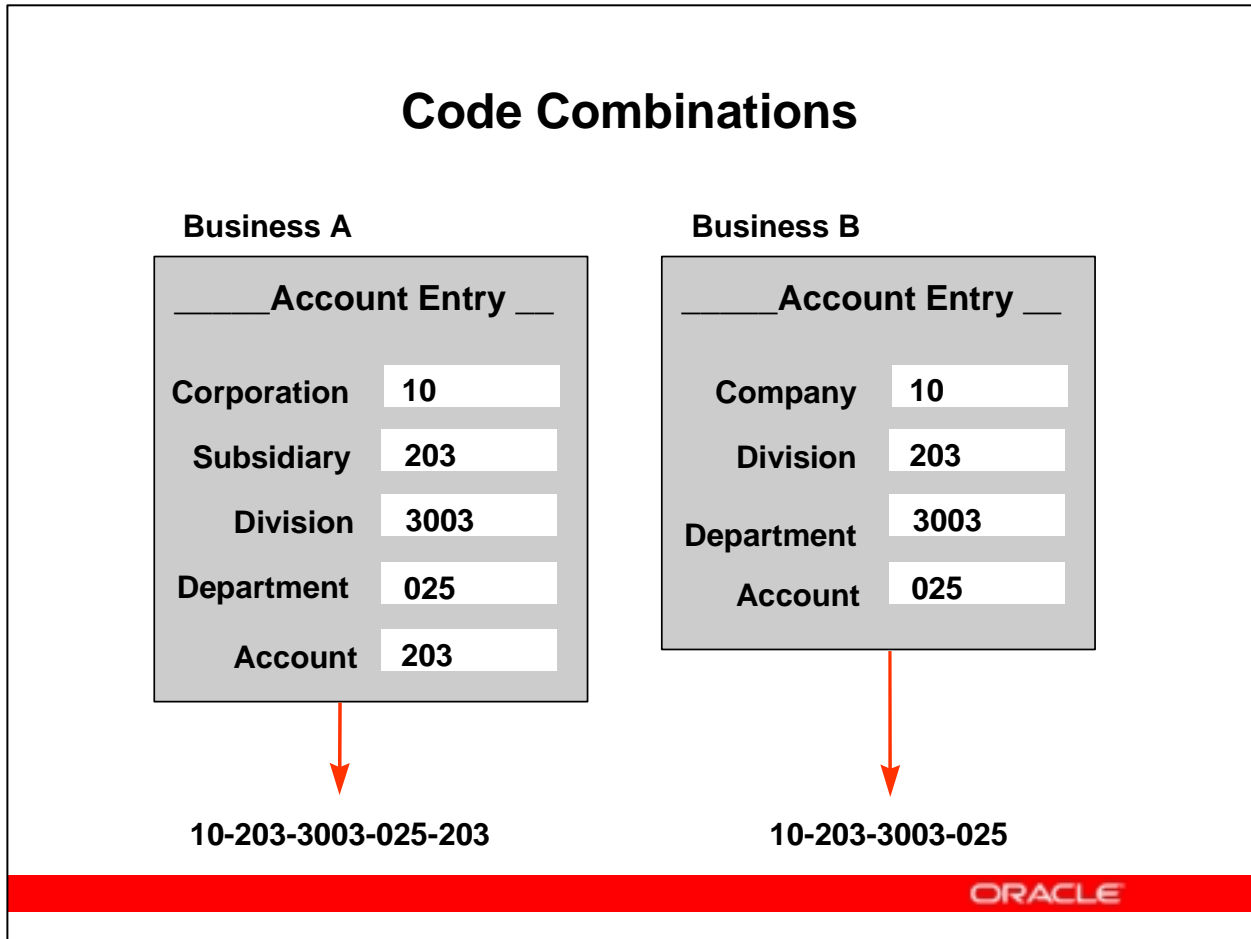
ORACLE

### Key Flexfield Structures

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

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

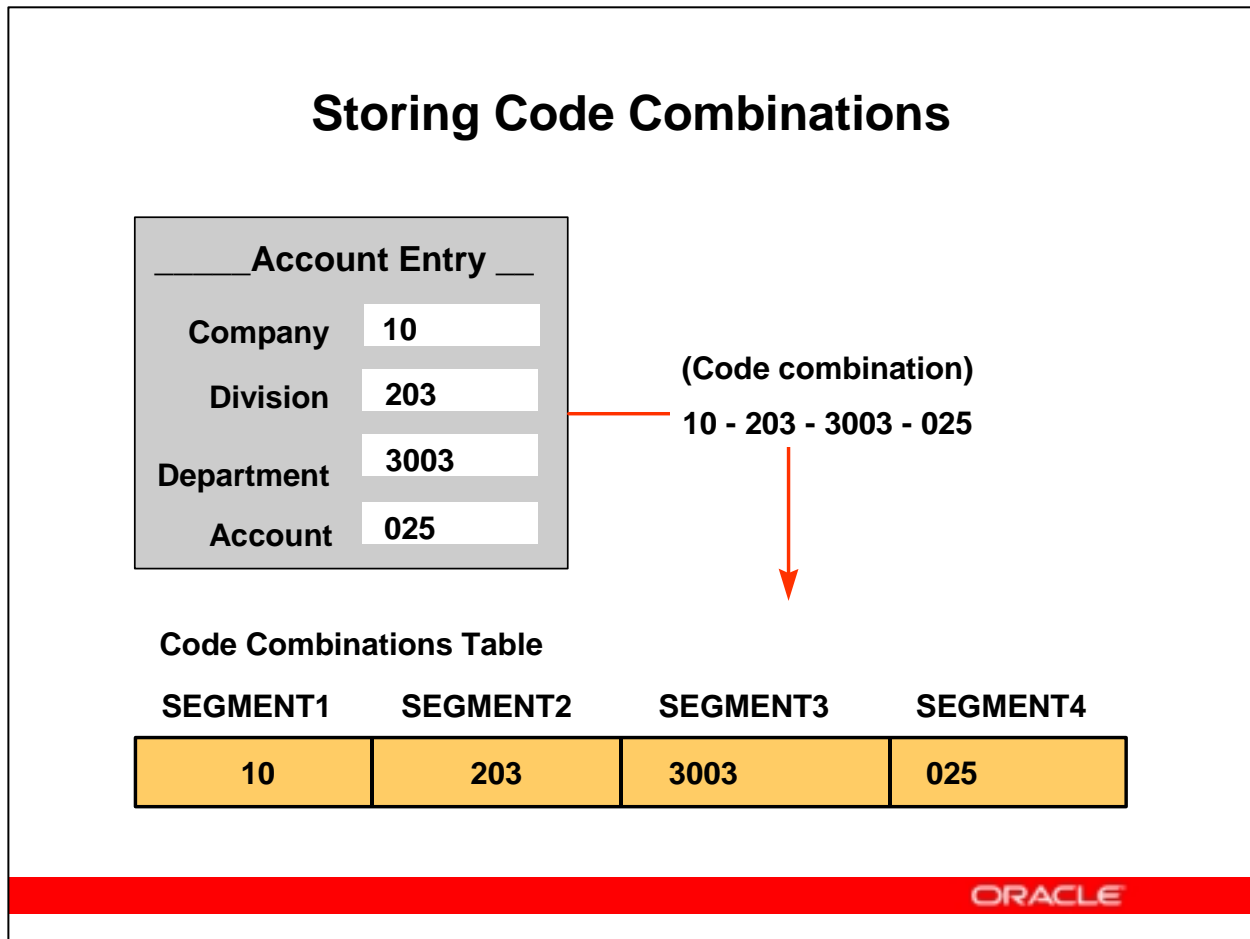
# Code Combinations



## Key Flexfield Code Combinations

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

## Storing Code Combinations



### Storing Code Combinations

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

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

## Key Flexfield Application Tables

---

### Key Flexfield Application Tables

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

ORACLE

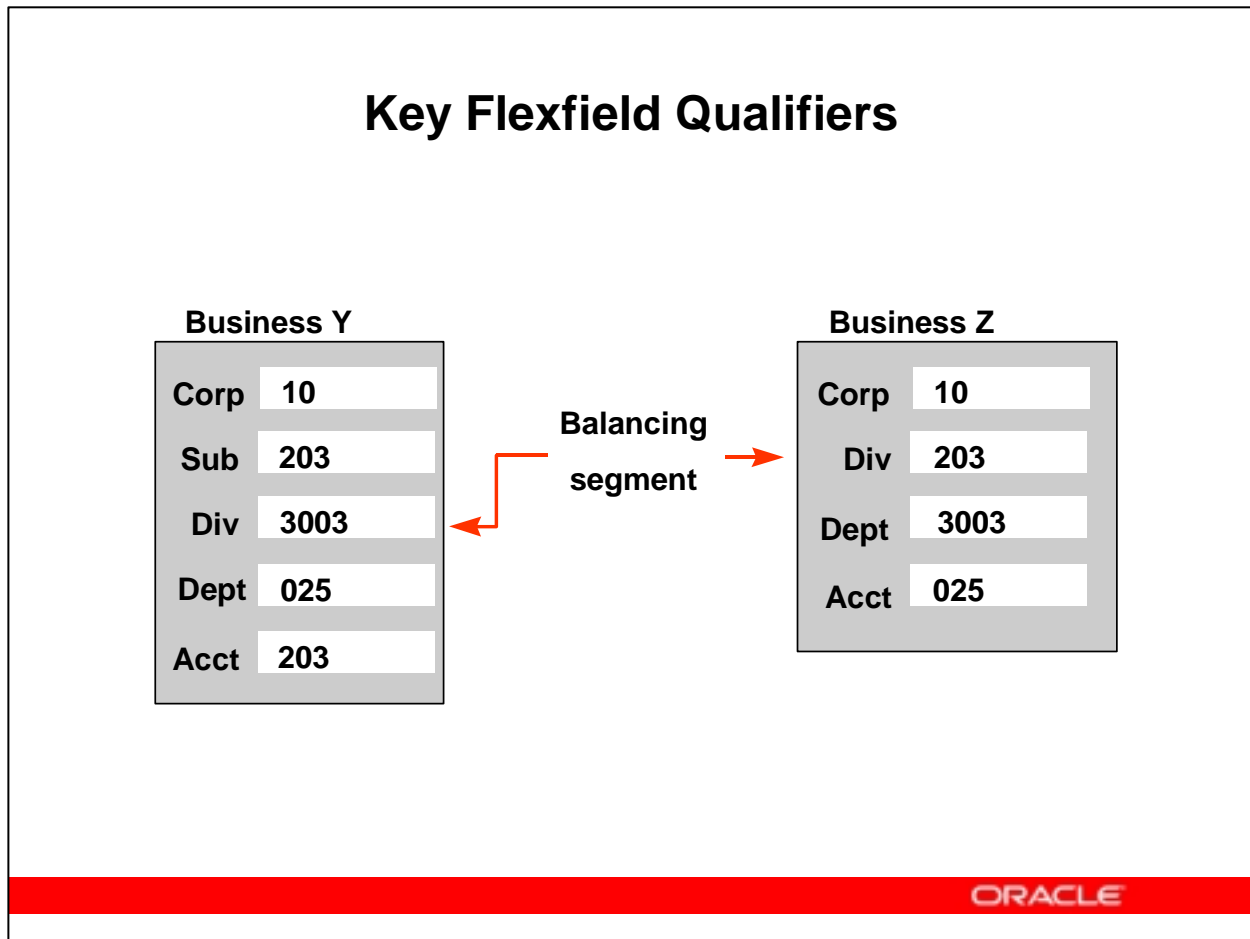
### Key Flexfield Application Tables

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

ID_FLEX_NAME	APPLICATION_TABLE_NAME
-----	-----
Accounting Flexfield	GL_CODE_COMBINATIONS
Category Flexfield	FA_CATEGORIES_B
Asset Key Flexfield	FA_ASSET_KEYWORDS
Location Flexfield	FA_LOCATIONS
Oracle Service Item Flexfield	MTL_SYSTEM_ITEMS_B
Territory Flexfield	RA_TERRITORIES
Sales Tax Location Flexfield	AR_LOCATION_COMBINATIONS
Item Categories	MTL_CATEGORIES_B
Account Aliases	MTL_GENERIC_DISPOSITIONS
Item Catalogs	MTL_ITEM_CATALOG_GROUPS

Sales Orders	MTL_SALES_ORDERS
System Items	MTL_SYSTEM_ITEMS_B
Stock Locators	MTL_ITEM_LOCATIONS
Grade Flexfield	PER_GRADE_DEFINITIONS
Job Flexfield	PER_JOB_DEFINITIONS
Personal Analysis Flexfield	PER_ANALYSIS_CRITERIA
Position Flexfield	PER_POSITION_DEFINITIONS
Soft Coded KeyFlexfield	HR_SOFT_CODING_KEYFLEX
Bank Details KeyFlexField	PAY_EXTERNAL_ACCOUNTS
Cost Allocation Flexfield	PAY_COST_ALLOCATION_KEYFLEX
People Group Flexfield	PAY_PEOPLE_GROUPS

## Key Flexfield Qualifiers



### Key Flexfield Qualifiers

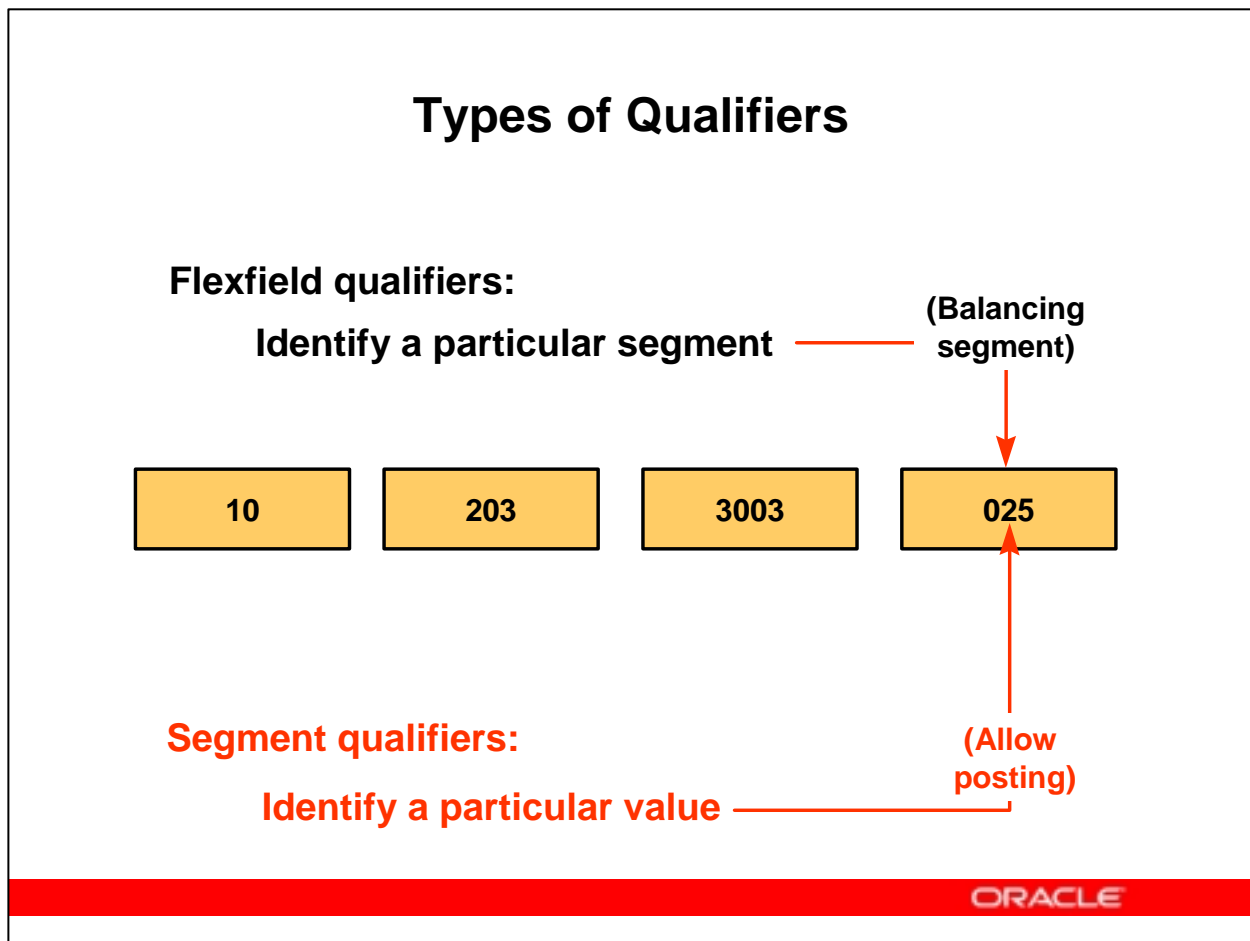
Both descriptive flexfields and key flexfields allow the user to design the flexfield structures and their segments. With descriptive flexfields, neither the information gathered nor the way the information is structured is used internally by Oracle Applications. Key flexfields, however, are different.

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

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



## Types of Qualifiers



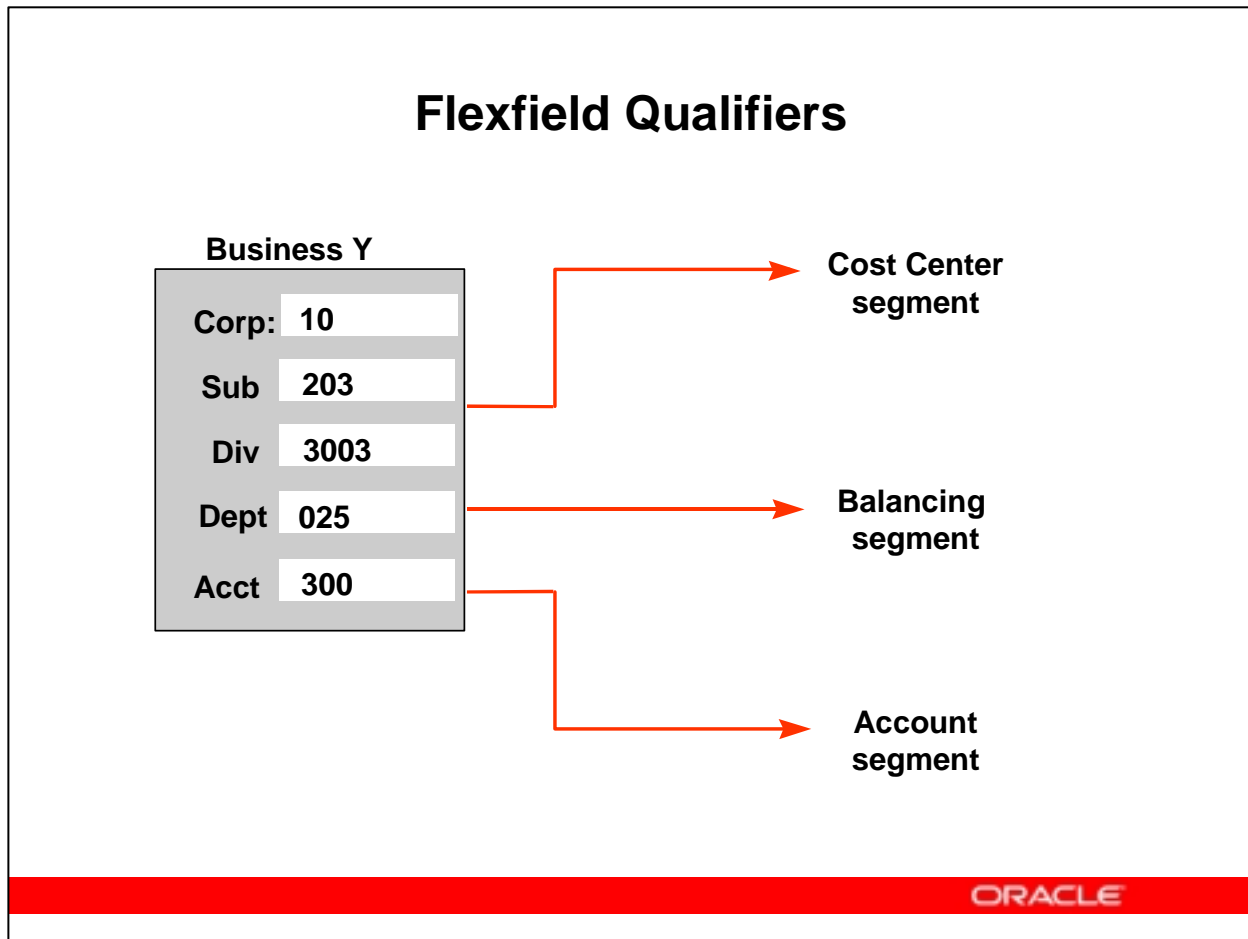
### Types of Key Flexfield Qualifiers

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

There are two types of qualifiers:

- Flexfield qualifiers identify a segment in a flexfield.
- Segment qualifiers identify a value in a segment.

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



### Flexfield Qualifiers Identify Key Flexfield Segments

The flexfield asks each segment a yes/no question.

Flexfield qualifiers may be unique, global, and required:

- Unique: “Is this the segment that this flexfield can have only one of?”
- Required: “Is this the segment this flexfield must have to do its work?”
- Global: “Is this a segment?” Global qualifiers exist as “carriers” for segment qualifiers.

### Assigning Flexfield Qualifiers to Segments

- Global qualifiers need not be assigned since they apply automatically to every segment in the flexfield.
- Assign flexfield qualifiers while defining segments.

### Oracle Applications Key Flexfields Using Qualifiers

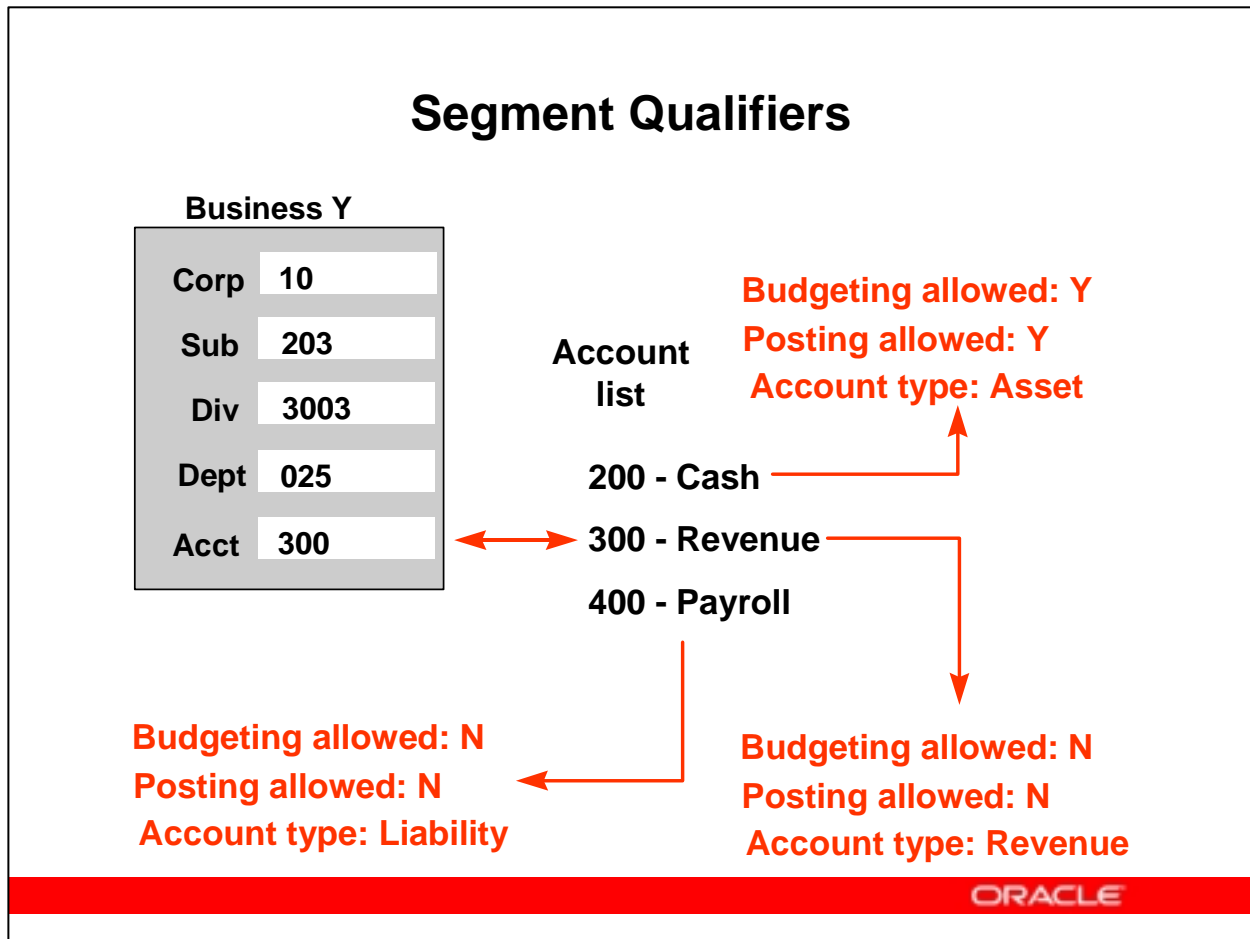
- **General Ledger** — Accounting flexfield
- **Oracle Assets** — Location flexfield, Asset Category flexfield
- **Oracle Human Resources** — SoftCoded Key flexfield
- **Oracle Payroll** — Cost Allocation flexfield
- **Oracle Receivables** — Sales Territory flexfield

ORACLE

#### Key Flexfields Using Qualifiers

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

## Segment Qualifiers



### Identifying Values in Flexfield Segments with Segment Qualifiers

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

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

Use the following segment qualifiers with the accounting flexfield:

- Allow Budgeting
- Allow Posting
- Account Type: Asset, Expense, Liability, Ownership/Stockholder’s Equity, or Revenue

### Other Key Flexfield Options

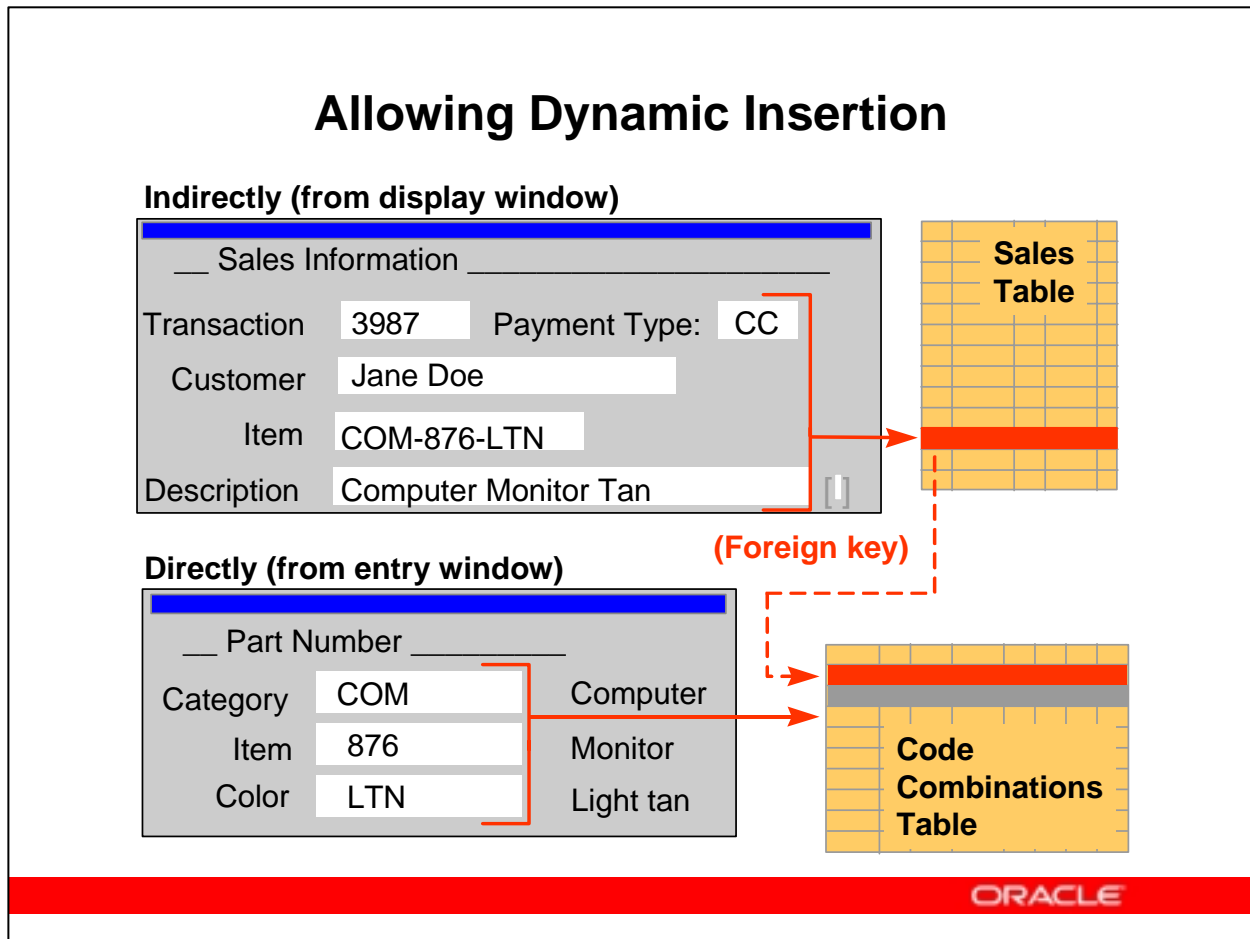
- **Dynamic insertion of new values**
- **Cross-validation of segment value combinations**
- **Security on value access**
- **Aliases to speed data entry**

ORACLE

### Other Key Flexfield Features

Some other capabilities of key flexfields are available for use. You should consider using these capabilities where appropriate:

- Dynamic insertion of new values
- Cross-validation of segment value combinations
- Security on values accessible
- Aliases to speed entry of frequently used value combinations

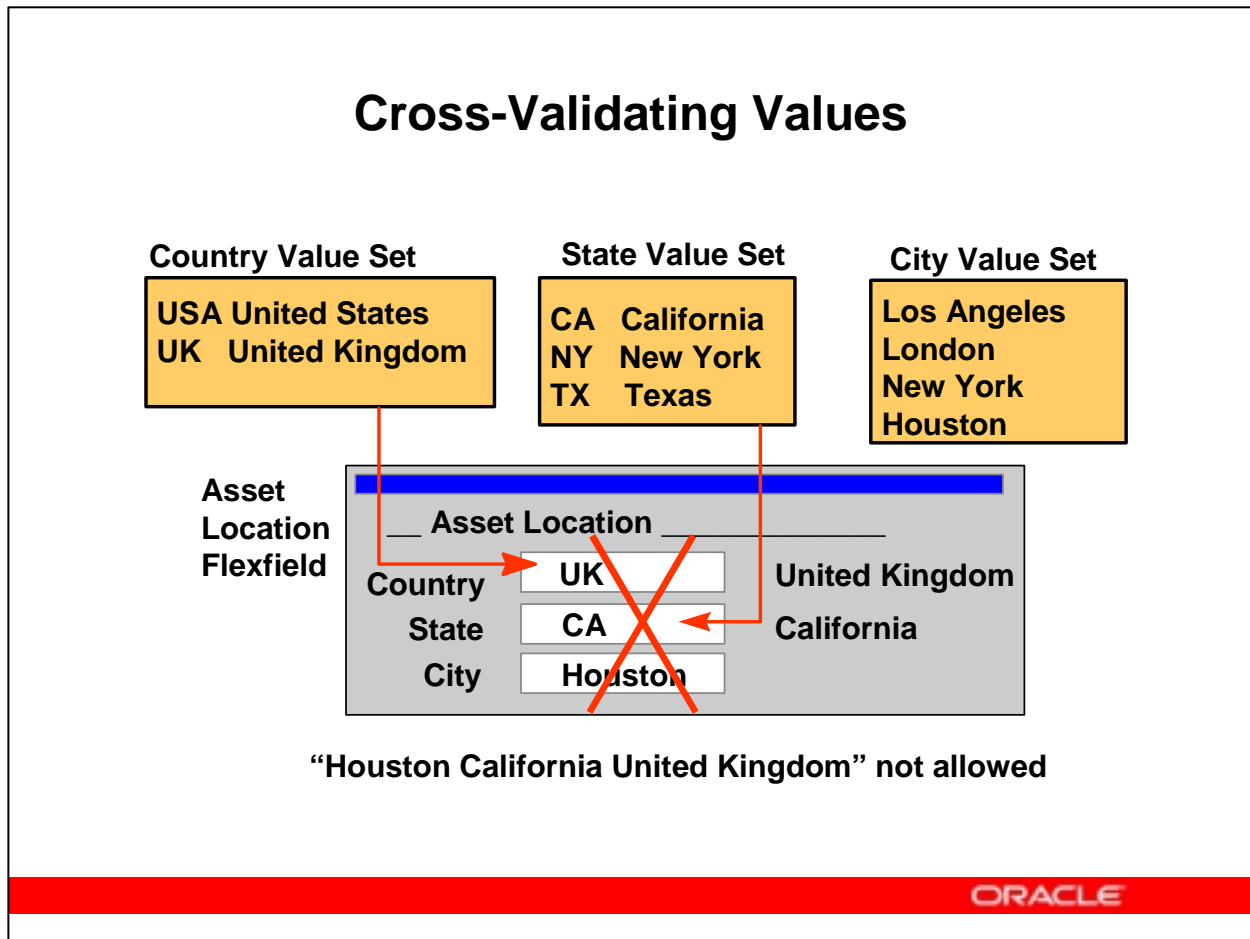


### Allowing Dynamic Insertion of New Code Combinations

Key flexfield code combinations appear on many types of windows. Typically, Oracle Applications use a particular form (called a combination form) for directly entering the new code combinations. These same code combinations then be displayed by many other windows. On these related windows, however, the fields are typically read-only and not updateable. Therefore new code combinations cannot be entered from these forms.

If you allow dynamic insertion, you can enter new code combinations from such display windows as well as from regular entry windows.

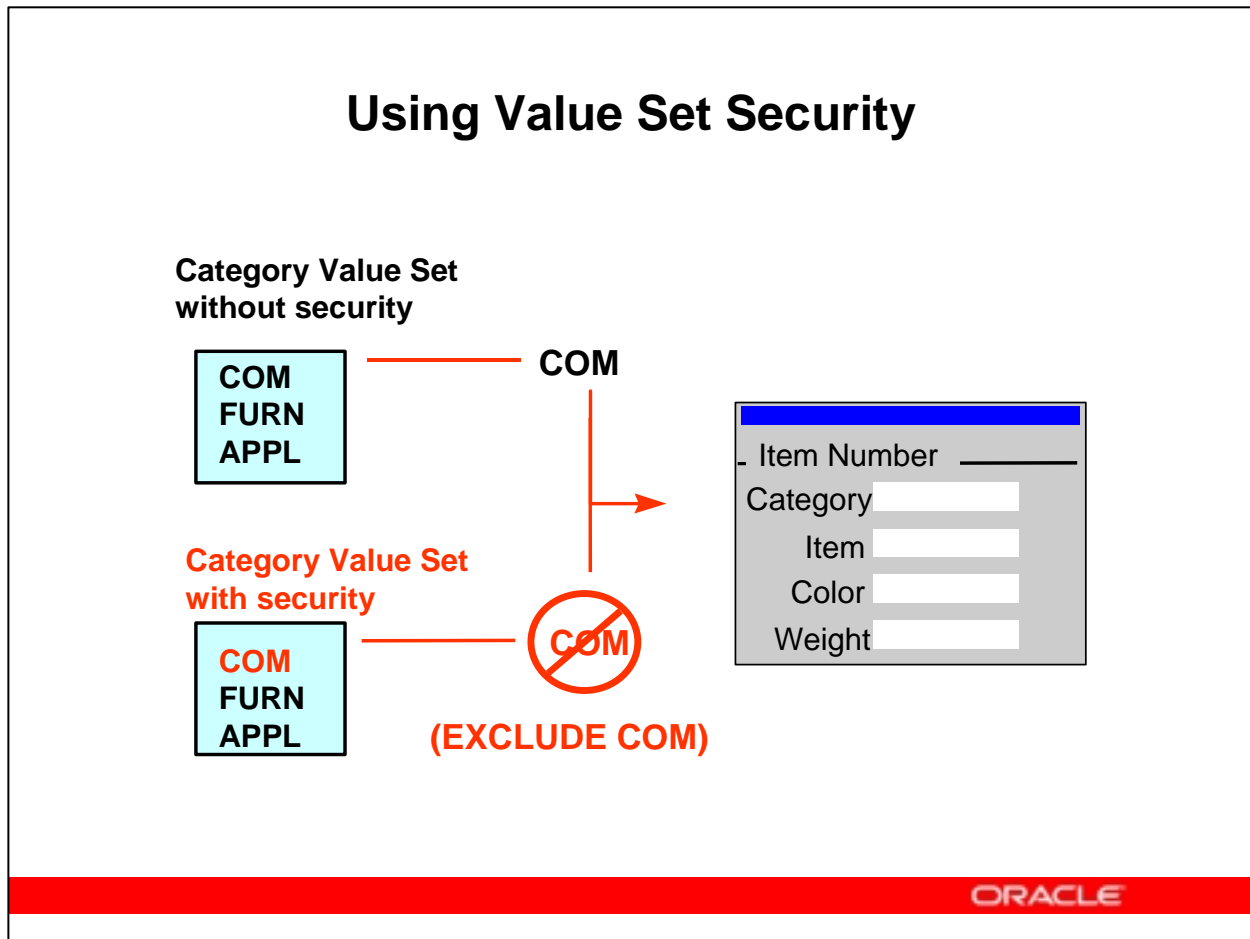
## Cross-Validating Values



### Validating the Combinations of Segment Values to Control Data Integrity

For key flexfields with multiple segments, you can define rules to cross-check value combinations entered. In this way, you can prevent combinations of segment values that are illogical or that should not be allowed from being entered.

The slide shows an illogical combination of values for the Asset key flexfield being disallowed.



### Using Value Security

You can specify who can use particular segment values by defining flexfield value security rules.

For example, the slide shows a security rule disallowing use of the value COM from the Category value set. The unsecured Category value set allows use of this value.

Define the rules for a particular value set and then associate the rule with the appropriate responsibility.



## Using Shorthand Aliases

### Using Shorthand Aliases

The screenshot shows a sales form with the following fields:

- Transaction: 3754
- Customer: Big Manufacturing
- Address: 1211 State
- City: Dallas, State: TX
- Zip: 75219, Country: USA
- Item: COM-876-LTN

An 'Item Alias' pop-up window shows 'Part Tan Monitor' selected. A table titled 'List of Aliases' is shown below:

Tan Monitor	COM-876-LTN
Hard Drive	COM-535-15G
Laser Printer	COM-788-630

ORACLE

### Using Shorthand Aliases to Speed Data Entry

Allow users to enter data faster and more easily with shorthand aliases. An alias is a label for a particular combination of key flexfield segment values. Give aliases to combinations that are entered frequently. Then just enter the alias into the flexfield to automatically populate the values for the segments.

### Planning Decisions

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

ORACLE

#### Planning Decisions

**Application Question:** Does the application support different segment structures?

**Related Question:** How many structures are needed?

**Application Question:** How many segment columns are available?

**Related Question:** What segments are needed?

**Application Question:** What flexfield qualifiers does this flexfield use or need?

**Related Question:** Do segments correspond to each needed qualifier?

**Application Question:** Are dynamic inserts feasible?

**Related Question:** Who can create new combinations?

**Application Question:** Should cross-validation be enabled?

**Related Question:** Is protection from invalid combinations needed?

**Application Question:** Should shorthand flexfield entry be enabled?

**Related Question:** Are many combinations used repeatedly?

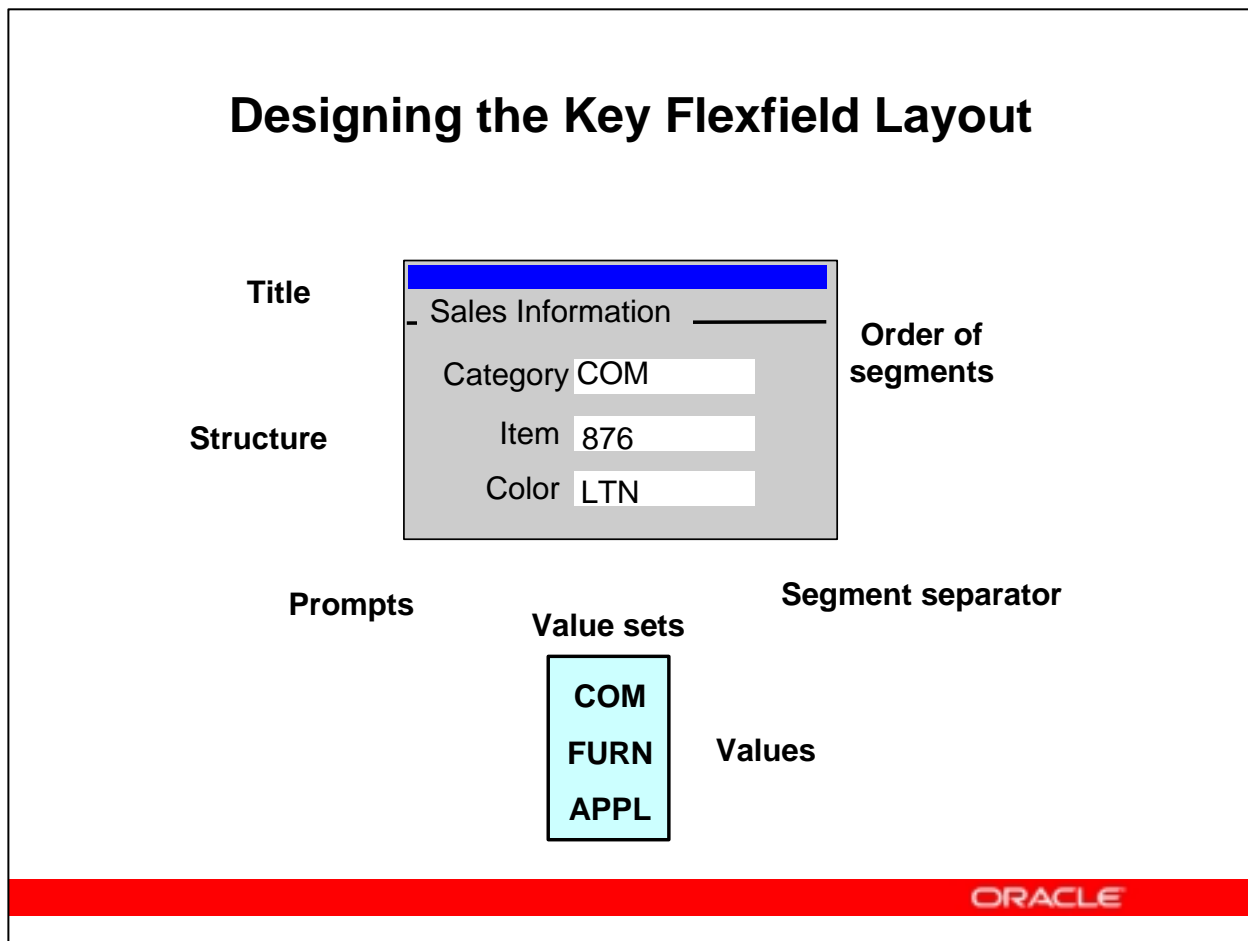
**Application Question:** Which value sets are available?

**Related Question:** How should the segments be validated?

**Application Question:** Which segments should use flexfield value security?

**Related Question:** Are some segment values privileged or applicable only for some users?

## Designing the Key Flexfield Layout



### Designing Key Flexfield Layout

Design the structures needed and the segments for each structure:

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

### Designing Segments

- **Enabled or Displayed**
- **Required**
- **Validated**
- **Secured**

ORACLE

#### Designing Key Flexfield Segments

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

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

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

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

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

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

### Specifying Default Values

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

ORACLE

#### Specifying Segment Defaults - Examples

**Default Type:** Constant

**Default Value:** The constant specified.

**Example:** USA

**Result:** USA

**Default Type:** Current Date

**Default Value:** The date at the time of entry.

**Example:**

**Result:** MAY 01, 2000

**Default Type:** Current Time

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

**Example:**

**Result:** 14:30:00 MAY 01, 2000

**Default Type:** Field

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

**Example:** ORDER:LINE

**Result:** 3

**Default Type:** Profile

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

**Example:** GL\_SET\_OF\_BOOKS\_ID

**Result:** 101

**Default Type:** Segment

**Default Value:** The value returned by the specified previous segment.

**Example:** Company

**Result:** 01

**Default Type:** SQL Statement

**Default Value:** The value returned by the specified SQL statement. The statement must return a single value. \$PROFILES\$ and \$FLEX\$ can be used in the statement.

**Example:** SELECT NAME FROM EMP WHERE JOB=CEO

**Result:** Jones

### Planning Key Flexfield Values

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

ORACLE

#### Planning Values for Use with Key Flexfields

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

#### Oracle General Ledger Values Information

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

- Oracle General Ledger applications require that segment qualifiers be assigned to some values used by the Accounting Flexfield.
- Oracle General Ledger applications can create hierarchies of values using rollup groups and parent-child relationships for processing and reporting.

The Oracle General Ledger courses provide more details.



### Definition Procedure

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

ORACLE

#### **Defining Key Flexfields: Procedure**

Use the following procedure to define a key flexfield:

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

### Defining Value Sets

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

(N) Application—>Validation—>Sets

ORACLE

### Accessing the Key Flexfield Definition

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

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

ORACLE

### Specifying Flexfield Behavior

Use the Key Flexfield Segments window to enter:

- **Enabled**
- **Segment Separator**
- **Cross-Validate Segments**
- **Freeze Rollup Groups**
- **Allow Dynamic Inserts**

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

ORACLE

### Specifying the Key Flexfield Behavior

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

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

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

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

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

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

### Defining Segment Attributes

Use the Segments Summary window to enter:

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

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

ORACLE

### Defining Segment Attributes

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

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

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

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

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

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

**Enabled** - This flags the segment as available for use.

### Defining Validation and Size Attributes

Use the Segments window to enter options for:

- Validation
- Sizes
- Prompts

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

ORACLE

### Defining Validation and Size Attributes

#### Validation Information

- Use the Validation block fields to specify value set information.
- Choose a predefined value set with a list of values or design one for this particular segment with the Value Sets window.
- Specify default information if you need to populate the segment with a default value.
- Determine whether security rules should apply to this value set for this segment.

#### Size Specifications

- Display Size determines the field size on the flexfield. Specify a display size the same as the maximum segment size to avoid scrolling.
- Keep prompts small for neater reports.

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

### Defining Flexfield Qualifiers

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

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

ORACLE

### Freezing and Compiling the Definition

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

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

ORACLE



### Defining Value Set Values

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

**(N) Application—>Validation—>Values**

ORACLE

### Defining Segment Qualifiers

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

(N) Application—>Validation—>Values

ORACLE

### Defining Segment Qualifiers

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

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

### Defining Value Hierarchies

- In the Values, Hierarchy, Qualifiers region of the Segment Values window, click the Define Child Ranges, Move Child Ranges, and View Hierarchies buttons to enter and view additional information about value hierarchies.
- Value hierarchies are used only by Oracle General Ledger and Oracle Public Sector General Ledger, and only with the Accounting Flexfield.

(N) Application—>Validation—>Values (B) Define Child Ranges

(N) Application—>Validation—>Values (B) Move Child Ranges

(N) Application—>Validation—>Values (B) View Hierarchies

ORACLE

### **Account Generator**

- **The Account Generator provides applications with the ability to construct Accounting Flexfield combinations automatically using customized construction criteria.**
- **Accounting Flexfield combinations are constructed using Oracle Workflow technology.**
- **The Account Generator can be called from a form or from a concurrent program.**

ORACLE

## Practice - Define a KFF

---

### Overview

Because many students access the system and create structures during this course, you need a way to distinguish between the structures created by you and by your classmates. Therefore, you will use your initials (e.g., WHS). Use your initials as a prefix wherever you need to define something. In this way, you can ensure that the definitions you create are unique.

Flexfield definitions can be created under many different responsibilities. However, the System Administrator responsibility has access to all functions needed to create the definitions in these practices.

In this practice you will create a key flexfield structure and values for your chart of accounts. The design of the structure has already been determined. The flexfield will have four segments: a company segment, a cost center segment, an account segment, and a segment labeled Future for possible future expansion, in that order. Before you define that structure, you must first create the value sets the structure will use. Then you will create the structure, identifying the segments by their respective flexfield qualifiers. Finally, you will create the valid values for each segment. After creating the values, you will test your structure by performing the first steps in the process for creating a new set of books.

### Tasks

#### Define your Value Sets

1. Use the Value Sets window to define four value sets.
  - Define an independent value set named *YourInitials\_CO*. Give the value set a description, a format type of Char, and a maximum size of 2. Enable security for the value set. Specify that the values should be right-justified and zero-filled.
  - Define an independent value set named *YourInitials\_COST*. Give the value set a description, a format type of Char, and a maximum size of 3. Enable security for the value set. Specify that the values should be right-justified and zero-filled.
  - Define an independent value set named *YourInitials\_ACCOUNT*. Give the value set a description, a format type of Char, and a maximum size of 4. Enable security and Longlist for the value set. Specify that the values should be right-justified and zero-filled.
  - Define an independent value set named *YourInitials\_FUTURE*. Give the value set a description, a format type of Char, and a maximum size of 4. Enable security for the value set. Specify that the values should be right-justified and zero-filled.

## Define your Structure

2. After defining your value sets, use the Key Flexfield Segments window to define a new flexfield structure for the Accounting Flexfield in the Oracle General Ledger application. Give your structure the code *YourInitials\_COA*, the title *YourInitials* Chart of Accounts, a description, and the view name *YourInitials\_AFF\_VIEW*. Enable the structure. Use a period (.) as the segment separator, specify that segments should be cross-validated, and allow dynamic inserts. Define four segments for your structure.
  - Define a segment named CO, with a prompt of Company. Assign the segment the number 1, the column SEGMENT1, and the value set *YourInitials\_CO*. Ensure that the segment is displayed and enabled. Enable security for the segment. Give the segment a display size of 2 and a description size of 30. Enable the Balancing Segment flexfield qualifier for the segment.
  - Define a segment named CC, with a prompt of Cost Center. Assign the segment the number 2, the column SEGMENT2, and the value set *YourInitials\_COST*. Ensure that the segment is displayed and enabled. Enable security for the segment. Give the segment a display size of 3 and a description size of 30. Enable the Cost Center Segment flexfield qualifier for the segment.
  - Define a segment named ACCT, with a prompt of Account. Assign the segment the number 3, the column SEGMENT3, and the value set *YourInitials\_ACCOUNT*. Ensure that the segment is displayed and enabled. Enable security for the segment. Give the segment a display size of 4 and a description size of 30. Enable the Natural Account Segment flexfield qualifier for the segment.
  - Define a segment named RFU, with a prompt of Future Use. Assign the segment the number 4, the column SEGMENT4, and the value set *YourInitials\_FUTURE*. Ensure that the segment is displayed and enabled. Specify a default type of Constant and a default value of 0000. Enable security for the segment. Give the segment a display size of 4 and a description size of 30.
  - When you finish defining the structure, freeze and compile your flexfield definition.

## Define your Values

3. After defining the value sets and segments for your flexfield, use the Segment Values window to define the values associated with each of the independent value sets.
4. Define and enable the following values for the *YourInitials\_CO* value set.
  - **Value:** 01
  - **Description:** Red Co.
  - **Value:** 02

- **Description:** Orange Co.
  - **Value:** 03
  - **Description:** Yellow Co.
  - **Value:** 04
  - **Description:** Green Co.
  - **Value:** 05
  - **Description:** Blue Co.
5. Define and enable the following values for the *YourInitials\_COST* value set.
- **Value:** 000
  - **Description:** Not Specified
  - **Value:** 100
  - **Description:** Sales
  - **Value:** 200
  - **Description:** Services
  - **Value:** 300
  - **Description:** Development
6. Define and enable the following values for the *YourInitials\_ACCOUNT* value set. Specify the account type segment qualifier for each value as shown.
- **Value:** 1110
  - **Description:** Cash
  - **Account Type:** Asset
  - **Value:** 1120
  - **Description:** Cash Clearing
  - **Account Type:** Asset
  - **Value:** 2110

- **Description:** Accounts Payable
  - **Account Type:** Liability
  - **Value:** 3110
  - **Description:** Salaries
  - **Account Type:** Liability
  - **Value:** 4110
  - **Description:** Travel Expense
  - **Account Type:** Expense
7. Define and enable the following value for the *YourInitials\_FUTURE* value set.
- **Value:** 0000
  - **Description:** Not Specified

### Test your Key Flexfield

8. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
9. In the General Ledger Super User, navigate to (N) Setup > Financials > Books > Define
10. Enter the name *Your Initials* Set of Books.
11. Select your chart of accounts (the Accounting Flexfield structure you defined).
12. Navigate to the Budgetary Control region.
13. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
14. Enter values in the Accounting Flexfield to test your work.
15. Cancel your entries and exit without saving.



## Solution - Define a KFF

### Define your Value Sets

#### Responsibility = System Administrator

1. Navigate to (N) Application > Validation > Set.
2. Enter the information for the first value set in the following fields:
  - **Value Set Name:** *YourInitials\_CO*
  - **Description:** *YourInitials* Company Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 2
  - **Right-justify and Zero-fill Numbers:** Selected
  - **Validation Type:** Independent
3. Save your work.
4. Enter the information for the second value set in the following fields:
  - **Value Set Name:** *YourInitials\_COST*
  - **Description:** *YourInitials* Cost Center Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 3
  - **Right-justify and Zero-fill Numbers:** Selected
  - **Validation Type:** Independent
5. Save your work.
6. Enter the information for the third value set in the following fields:
  - **Value Set Name:** *YourInitials\_ACCOUNT*

- **Description:** *YourInitials* Account Value Set
  - **Security Available:** Selected
  - **Enable Longlist:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 4
  - **Right-justify and Zero-fill Numbers:** Selected
  - **Validation Type:** Independent
7. Save your work.
8. Enter the information for the fourth value set in the following fields:
- **Value Set Name:** *YourInitials\_Future*
  - **Description:** *YourInitials* Reserved for Future Use Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 4
  - **Right-justify and Zero-fill Numbers:** Selected
  - **Validation Type:** Independent

9. Save your work.

### Define your Structure

1. Navigate to (N) Application > Flexfield > Key > Segments.
2. Query the application Oracle General Ledger and the flexfield title Accounting Flexfield.
3. In the Structures region, enter the information for the flexfield structure in the following fields:
  - **Code:** *YourInitials\_COA*
  - **Title:** *YourInitials* Chart of Accounts
  - **Description:** *YourInitials* Chart of Accounts

- **View Name:** XX\_AFF\_VIEW
  - **Enabled:** Selected
  - **Segment Separator:** Period (.)
  - **Cross-Validate Segments:** Selected
  - **Allow Dynamic Inserts:** Selected
4. Click the “Segments” button to navigate to the Segments Summary window.
  5. Enter the information for the first segment in the following fields:
    - **Number:** 1
    - **Name:** CO
    - **Window Prompt:** Company
    - **Column:** SEGMENT1
    - **Value Set:** *YourInitials\_CO*
    - **Displayed:** Selected
    - **Enabled:** Selected
  6. Click the “Open” button to navigate to the Segments window.
  7. Enter the information for the first segment in the following fields:
    - **Security Enabled:** Selected
    - **Display Size:** 2
    - **Description Size:** 30
  8. Save your work.
  9. Click the “Flexfield Qualifiers” button to navigate to the Flexfield Qualifiers window.
  10. Select the Enabled check box for the Balancing Segment flexfield qualifier.
  11. Save your work.
  12. Navigate back to the Segments Summary window.
  13. Enter the information for the second segment in the following fields:

- **Number:** 2
  - **Name:** CC
  - **Window Prompt:** Cost Center
  - **Column:** SEGMENT2
  - **Value Set:** *YourInitials\_COST*
  - **Displayed:** Selected
  - **Enabled:** Selected
14. Click the “Open” button to navigate to the Segments window.
  15. Enter the information for the second segment in the following fields:
    - **Security Enabled:** Selected
    - **Display Size:** 3
    - **Description Size:** 30
  16. Save your work.
  17. Click the “Flexfield Qualifiers” button to navigate to the Flexfield Qualifiers window.
  18. Select the Enabled check box for the Cost Center Segment flexfield qualifier.
  19. Save your work.
  20. Navigate back to the Segments Summary window.
  21. Enter the information for the third segment in the following fields:
    - **Number:** 3
    - **Name:** ACCT
    - **Window Prompt:** Account
    - **Column:** SEGMENT3
    - **Value Set:** *YourInitials\_ACCOUNT*
    - **Displayed:** Selected
    - **Enabled:** Selected

22. Click the “Open” button to navigate to the Segments window.
23. Enter the information for the third segment in the following fields:
  - **Security Enabled:** Selected
  - **Display Size:** 4
  - **Description Size:** 30
24. Save your work.
25. Click the “Flexfield Qualifiers” button to navigate to the Flexfield Qualifiers window.
26. Select the Enabled check box for the Natural Account Segment flexfield qualifier.
27. Save your work.
28. Navigate back to the Segments Summary window.
29. Enter the information for the fourth segment in the following fields:
  - **Number:** 4
  - **Name:** RFU
  - **Window Prompt:** Future Use
  - **Column:** SEGMENT4
  - **Value Set:** *YourInitials\_FUTURE*
  - **Displayed:** Selected
  - **Enabled:** Selected
30. Click the “Open” button to navigate to the Segments window.
31. Enter the information for the fourth segment in the following fields:
  - **Default Type:** Constant
  - **Default Value:** 0000
  - **Security Enabled:** Selected
  - **Display Size:** 4
  - **Description Size:** 30

32. Save your work.
33. Navigate back to the Key Flexfield Segments window.
34. Select the Freeze Flexfield Definition check box.
35. Click the “Compile” button to compile the flexfield definition.

### **Define your Values**

1. Navigate to (N) Application > Validation > Values.
2. In the Find window, select Value Set and find the *YourInitials\_CO* value set.
3. In the Values, Effective region of the Segment Values window, define the following values:
  - **Value:** 01
  - **Description:** Red Co.
  - **Enabled:** Selected
  - **Value:** 02
  - **Description:** Orange Co.
  - **Enabled:** Selected
  - **Value:** 03
  - **Description:** Yellow Co.
  - **Enabled:** Selected
  - **Value:** 04
  - **Description:** Green Co.
  - **Enabled:** Selected
  - **Value:** 05
  - **Description:** Blue Co.
  - **Enabled:** Selected
4. Save your work.
5. In the Segment Values window, select Value Set and find the *YourInitials\_COST* value set.

6. In the Values, Effective region, define the following values:
  - **Value:** 000
  - **Description:** Not Specified
  - **Enabled:** Selected
  - **Value:** 100
  - **Description:** Sales
  - **Enabled:** Selected
  - **Value:** 200
  - **Description:** Services
  - **Enabled:** Selected
  - **Value:** 300
  - **Description:** Development
  - **Enabled:** Selected
7. Save your work.
8. In the Segment Values window, select Value Set and find the *YourInitials\_ACCOUNT* value set.
9. In the Values, Effective region, define the following values:
  - **Value:** 1110
  - **Description:** Cash
  - **Enabled:** Selected
  - **Value:** 1120
  - **Description:** Cash Clearing
  - **Enabled:** Selected
  - **Value:** 2110
  - **Description:** Accounts Payable

- **Enabled:** Selected
- **Value:** 3110
- **Description:** Salaries
- **Enabled:** Selected
- **Value:** 4110
- **Description:** Travel Expense
- **Enabled:** Selected

10. In the Values, Hierarchy, Qualifiers region, define the following value attributes:

- **Value:** 1110
- **Account Type:** Asset
- **Value:** 1120
- **Account Type:** Asset
- **Value:** 2110
- **Account Type:** Liability
- **Value:** 3110
- **Account Type:** Liability
- **Value:** 4110
- **Account Type:** Expense

**Note:** Click in the Qualifiers field in the Values, Hierarchy, Qualifiers region to display the Segment Qualifiers window and enter a value in the Account Type field. Accept the default values for the other fields in the Segment Qualifiers window.

11. Save your work.

12. In the Segment Values window, select Value Set and find the XX\_FUTURE value set.

13. In the Values, Effective region, define the following value:

- **Value:** 0000
- **Description:** Not Specified



- **Enabled:** Selected

14. Save your work.

### **Test your Key Flexfield**

1. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
2. In the General Ledger Super User responsibility, navigate to (N) Setup > Financials > Books > Define.
3. Enter the name *YourInitials* Set of Books.
4. Select your chart of accounts (the Accounting Flexfield structure you defined).
5. Navigate to the Budgetary Control region.
6. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
7. Enter values in the Accounting Flexfield to test your work.
8. Cancel your entries and exit without saving.

# Practice - Define a KFF with a Dependent Segment

---

## Overview

In this practice, you will create a key flexfield with a dependent segment. Although you would not normally use a dependent segment in the Accounting Flexfield, this practice will illustrate how to use dependencies in key flexfields, a procedure similar to the use of dependencies in descriptive flexfields. Dependent segments can be used in many key flexfields, such as System Items or Item Categories. You will create another key flexfield structure for your chart of accounts. The design of this structure has also already been determined. The flexfield will have four segments: a company segment, an account segment, a subaccount segment, and a segment labeled Future for possible future expansion, in that order. The following flexfield qualifiers will be associated with this structure. The company segment will be the Balancing segment. The account segment will be the Natural Account segment. Cost Center is not a required flexfield qualifier, so you will not use that qualifier here.

## Tasks

### Define your Value Sets

1. Use the Value Sets window to define two value sets.
  - Define an independent value set named *YourInitials\_ACCT*. Give the value set a description, a format type of Char, and a maximum size of 4. Enable security for the value set. Specify that the values should be right-justified and zero-filled.
  - Define a dependent value set named *YourInitials\_SUB*. Give the value set a description, a format type of Char, and a maximum size of 2. Enable security for the value set. Specify that the values should be right-justified and zero-filled. Associate the dependent value set with the independent value set *YourInitials\_ACCT*. Assign the dependent value set a dependent default value of *YourInitials*, and give the default value a description.
  - In addition to these two value sets, you will use two of the value sets you created previously, *YourInitials\_CO* and *YourInitials\_FUTURE*. You do not need to redefine these value sets.

### Define your Structure

2. After defining your value sets, use the Key Flexfield Segments window to define another new flexfield structure for the Accounting Flexfield in the Oracle General Ledger application. Give your structure the code *YourInitials\_COA2*, the title *YourInitials Chart of Accounts 2*, a description, and the view name *YourInitials\_AFF\_VIEW\_2*. Enable the structure. Use a period (.) as the segment separator, specify that segments should be cross-validated, and allow dynamic inserts. Define four segments for your structure.

- Define a segment named CO, with a prompt of Company. Assign the segment the number 1, the column SEGMENT1, and the value set *YourInitials\_CO*. Ensure that the segment is displayed and enabled. Enable security for the segment. Give the segment a display size of 2 and a description size of 30. Enable the Balancing Segment flexfield qualifier for the segment.
- Define a segment named ACCT, with a prompt of Account. Assign the segment the number 2, the column SEGMENT2, and the value set *YourInitials\_ACCT*. Ensure that the segment is displayed and enabled. Enable security for the segment. Give the segment a display size of 4 and a description size of 30. Enable the Natural Account Segment flexfield qualifier for the segment.
- Define a segment named SUB, with a prompt of Subaccount. Assign the segment the number 3, the column SEGMENT3, and the value set *YourInitials\_SUB*. Ensure that the segment is displayed and enabled. Enable security for the segment. Give the segment a display size of 2 and a description size of 30.
- Define a segment named RFU, with a prompt of Future Use. Assign the segment the number 4, the column SEGMENT4, and the value set *YourInitials\_FUTURE*. Ensure that the segment is displayed and enabled. Specify a default type of Constant and a default value of 0000. Enable security for the segment. Give the segment a display size of 4 and a description size of 30.
- When you finish defining the structure, freeze and compile your flexfield definition.

## Define your Values

3. After defining the value sets and segments for your flexfield, use the Segment Values window to define the values associated with the independent and dependent value sets.
4. Define and enable the following values for the *YourInitials\_ACCT* value set. Specify the account type segment qualifier for each value as shown.
  - **Value:** 0000
  - **Description:** Not Specified
  - **Account Type:** Expense
  - **Value:** 1500
  - **Description:** Western Region
  - **Account Type:** Expense
  - **Value:** 2500

- **Description:** Eastern Region
  - **Account Type:** Expense
5. Define and enable the following values in the *YourInitials\_SUB* value set for the independent value 1500 (Western Region).
    - **Value:** 55
    - **Description:** Car expense
    - **Value:** 67
    - **Description:** Food expense
    - **Value:** 99
    - **Description:** Entertainment
  6. Define and enable the following values in the *YourInitials\_SUB* value set for the 2500 (Eastern Region).
    - **Value:** 55
    - **Description:** Car expense
    - **Value:** 87
    - **Description:** Hotel expense
    - **Value:** 89
    - **Description:** Training materials

### Test your Key Flexfield

8. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
9. In the General Ledger Super User, navigate to (N) Setup > Financials > Books > Define
10. Enter the name *Your Initials* Set of Books.
11. Select your chart of accounts (the Accounting Flexfield structure you defined).
12. Navigate to the Budgetary Control region.

13. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
14. Enter values in the Accounting Flexfield to test your work.
15. Cancel your entries and exit without saving.

## Solution - Define a KFF with a Dependent Segment

### Define your Value Sets

#### Responsibility = System Administrator

1. Navigate to (N) Application > Validation > Set
2. Enter the information for the first value set in the following fields:
  - **Value Set Name:** *YourInitials\_ACCT*
  - **Description:** *YourInitials* Account Value Set 2
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 4
  - **Right-justify and Zero-fill Numbers:** Selected
  - **Validation Type:** Independent
3. Save your work.
4. Enter the information for the second value set in the following fields:
  - **Value Set Name:** *YourInitials\_SUB*
  - **Description:** *YourInitials* Subaccount Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 2
  - **Right-justify and Zero-fill Numbers:** Selected
  - **Validation Type:** Dependent
5. Click the “Edit Information” button to navigate to the Dependent Value Set Information window.
6. In the Independent Value Set region, enter *YourInitials\_ACCT* in the Name field.

7. In the Dependent Default Value region, enter *YourInitials* in the Value field and *YourInitials* Subaccount Default in the Description field.
8. Save your work.

## Define your Structure

1. Navigate to (N) Application > Flexfield > Key > Segments.
2. Query the application Oracle General Ledger and the flexfield title Accounting Flexfield.
3. In the Structures region, enter the information for the flexfield structure in the following fields:
  - **Code:** *YourInitials\_COA2*
  - **Title:** *YourInitials* Chart of Accounts 2
  - **Description:** *YourInitials* Chart of Accounts 2
  - **View Name:** *YourInitials\_AFF\_VIEW\_2*
  - **Enabled:** Selected
  - **Segment Separator:** Period (.)
  - **Cross-Validate Segments:** Selected
  - **Allow Dynamic Inserts:** Selected
4. Click the Segments button to navigate to the Segments Summary window.
5. Enter the information for the first segment in the following fields:
  - **Number:** 1
  - **Name:** CO
  - **Window Prompt:** Company
  - **Column:** SEGMENT1
  - **Value Set:** *YourInitials\_CO*
  - **Displayed:** Selected
  - **Enabled:** Selected
6. Click the “Open: button to navigate to the Segments window.

7. Enter the information for the first segment in the following fields:
  - **Security Enabled:** Selected
  - **Display Size:** 2
  - **Description Size:** 30
8. Save your work.
9. Click the “Flexfield Qualifiers” button to navigate to the Flexfield Qualifiers window.
10. Select the Enabled check box for the Balancing Segment flexfield qualifier.
11. Save your work.
12. Navigate back to the Segments Summary window.
13. Enter the information for the second segment in the following fields:
  - **Number:** 2
  - **Name:** ACCT
  - **Window Prompt:** Account
  - **Column:** SEGMENT2
  - **Value Set:** *YourInitials\_ACCT*
  - **Displayed:** Selected
  - **Enabled:** Selected
14. Click the “Open” button to navigate to the Segments window.
15. Enter the information for the second segment in the following fields:
  - **Security Enabled:** Selected
  - **Display Size:** 4
  - **Description Size:** 30
16. Save your work.
17. Click the “Flexfield Qualifiers” button to navigate to the Flexfield Qualifiers window.
18. Select the Enabled check box for the Natural Account Segment flexfield qualifier.



19. Save your work.
20. Navigate back to the Segments Summary window.
21. Enter the information for the third segment in the following fields:
  - **Number:** 3
  - **Name:** SUB
  - **Window Prompt:** Subaccount
  - **Column:** SEGMENT3
  - **Value Set:** *YourInitials\_SUB*
  - **Displayed:** Selected
  - **Enabled:** Selected
22. Click the “Open” button to navigate to the Segments window.
23. Enter the information for the third segment in the following fields:
  - **Security Enabled:** Selected
  - **Display Size:** 2
  - **Description Size:** 30
24. Save your work.
25. Navigate back to the Segments Summary window.
26. Enter the information for the fourth segment in the following fields:
  - **Number:** 4
  - **Name:** RFU
  - **Window Prompt:** Future Use
  - **Column:** SEGMENT4
  - **Value Set:** *YourInitials\_FUTURE*
  - **Displayed:** Selected
  - **Enabled:** Selected

27. Click the “Open” button to navigate to the Segments window.
28. Enter the information for the fourth segment in the following fields:
  - **Default Type:** Constant
  - **Default Value:** 0000
  - **Security Enabled:** Selected
  - **Display Size:** 4
  - **Description Size:** 30
29. Save your work.
30. Navigate back to the Key Flexfield Segments window.
31. Select the Freeze Flexfield Definition check box.
32. Click the “Compile” button to compile the flexfield definition.

### **Define your Values**

1. Navigate to (N) Application > Validation > Values.
2. In the Find window, select Value Set and find the *YourInitials\_ACCT* value set.
3. In the Values, Effective region of the Segment Values window, define the following values:
  - **Value:** 0000
  - **Description:** Not Specified
  - **Enabled:** Selected
  - **Value:** 1500
  - **Description:** Western Region
  - **Enabled:** Selected
  - **Value:** 2500
  - **Description:** Eastern Region
  - **Enabled:** Selected
4. In the Values, Hierarchy, Qualifiers region, define the following value attributes:

- **Value:** 0000
- **Account Type:** Expense
- **Value:** 1500
- **Account Type:** Expense
- **Value:** 2500
- **Account Type:** Expense

**Note:** Click in the Qualifiers field in the Values, Hierarchy, Qualifiers region to display the Segment Qualifiers window and enter a value in the Account Type field. Accept the default values for the other fields in the Segment Qualifiers window.

5. Save your work.
6. In the Segment Values window, select Value Set and find the *YourInitials\_SUB* value set and the independent value 1500 (Western Region).
7. In the Values, Effective region, define the following values:
  - **Value:** 55
  - **Description:** Car expense
  - **Enabled:** Selected
  - **Value:** 67
  - **Description:** Food expense
  - **Enabled:** Selected
  - **Value:** 99
  - **Description:** Entertainment
  - **Enabled:** Selected
8. Save your work.
9. In the Segment Values window, select Value Set and find the *YourInitials\_SUB* value set and the independent value 2500 (Eastern Region).
10. In the Values, Effective region, define the following values:
  - **Value:** 55

- **Description:** Car expense
- **Enabled:** Selected
- **Value:** 87
- **Description:** Hotel expense
- **Enabled:** Selected
- **Value:** 89
- **Description:** Training materials
- **Enabled:** Selected

11. Save your work.

### **Test your Key Flexfield**

1. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
2. In the General Ledger Super User responsibility, navigate to (N) Setup > Financials > Books > Define.
3. Enter the name *YourInitials* Set of Books.
4. Select your chart of accounts (the Accounting Flexfield structure you defined).
5. Navigate to the Budgetary Control region.
6. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
7. Enter values in the Accounting Flexfield to test your work.
8. Cancel your entries and exit without saving.

### Defining Key Flexfields Summary

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

ORACLE

#### Summary

Key flexfields are used to build identifiers required by Oracle Applications. Key flexfields allow the user to provide information needed by Oracle Applications while still structuring that information to reflect a particular business environment.

The procedure for defining a key flexfield is similar to that for defining a descriptive flexfield. However, there are additional attributes for key flexfields that may also need to be defined. Additionally, key flexfields have several optional features that should be implemented where appropriate.

The additional requirements and options for key flexfields include:

- By using qualifiers in key flexfields you can require that certain segments and certain values be identified for processing.
- You can allow security checking and integrity checking.
- You can define aliases to speed data entry.
- You can enter key flexfield value combinations from different windows, if allowed.



# Using Additional Flexfield Options

## Chapter 16

# Using Additional Flexfield Options

ORACLE



### Objectives

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

- **Explain the purpose for cross-validation**
- **Define cross-validation rules**
- **Control the interaction of multiple cross-validation rules.**

ORACLE

### Lesson Objectives

At the end of this lesson, you should be able to:

- Explain the use of cross-validation to ensure data coherence
- Plan cross-validation rules for a key flexfield
- Implement cross-validation rules
- Review and control the interaction of cross-validation rules

### Objectives

- **Identify candidates for shorthand entry**
- **Plan useful aliases**
- **Enable shorthand entry**
- **Define a shorthand alias**

ORACLE

#### **Lesson Objectives (continued)**

- Identify candidates for shorthand entry
- Plan useful aliases
- Enable shorthand entry
- Define a shorthand alias

### Objectives

- **Explain how flexfield security is accomplished**
- **Identify which flexfields are candidates for security**
- **Design a security plan**
- **Control interactions between security rules**
- **Define security rules**
- **Assign security rules**
- **Enable security**

ORACLE

#### **Lesson Objectives (continued)**

- Explain how flexfield security is accomplished
- Identify which flexfields are candidates for security
- Design a security plan
- Control interactions between security rules
- Define security rules
- Assign security rules
- Enable security

### Overview

- **Designing cross-validation rules**
- **Defining and enabling cross-validation checking**
- **Identifying candidates for aliases**
- **Defining and enabling shorthand entry**
- **Using flexfield security**
- **Planning security rules**
- **Implementing security**

ORACLE

### Overview

Additional flexfield options include:

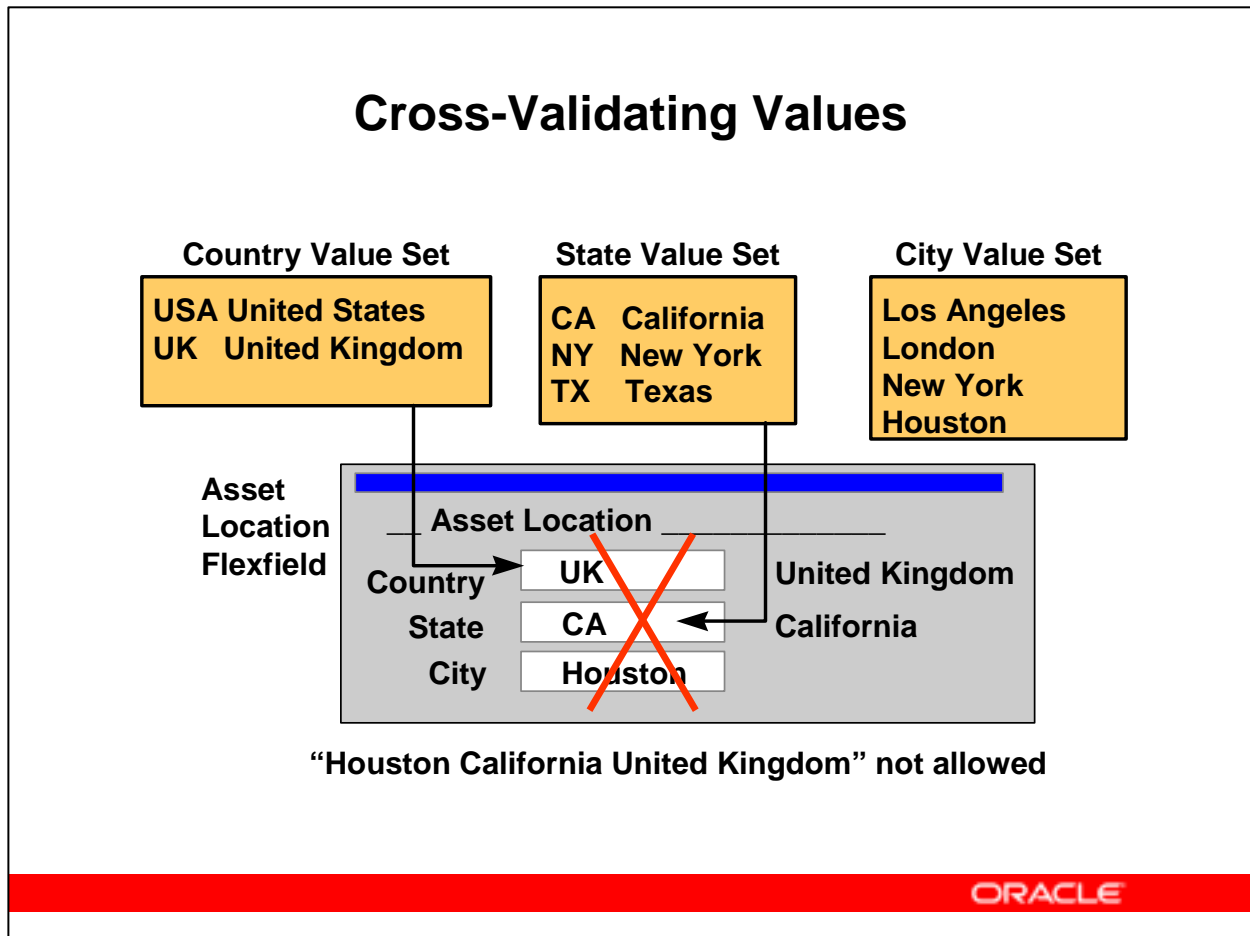
**Cross-validation (key flexfields only)** - Cross-validation rules let you control the combination of values entered for a key flexfield.

**Shorthand aliases (key flexfields only)** - Shorthand aliases let you label certain groupings of values for a multi-segment flexfield. Whenever you need one of these groupings of values, you can use the alias for that grouping to enter the entire group.

**Security rules (key and descriptive flexfields)** - Security rules let you control flexfield access to data. You create and tie security rules to a value set to specify which entries in a list of values are available to a user of a specific responsibility. This allows you to control that user's access to data.

You create security rules by defining inclusion or exclusion statements. You combine these statements to create a security rule. You must carefully consider the interaction of multiple rules when planning your security.

## Cross-Validating Values



### Using Cross-Validation to Enforce Data Integrity

You can create rules specifying the allowable value combinations for multiple segment key flexfields. In this way you can avoid the generation of illogical or inappropriate key values. Cross-validation rules apply only to key flexfields.

The slide shows an Asset Location flexfield with three fields for country, state, and city information. Since Asset Location is a key flexfield, you could define a cross-validation rule to be sure that no one tried to specify a US state value for the UK. Note that you could also specify cross-validation rules to ensure that each city value was placed in the correct state.

### Cross-Validation Rule Syntax

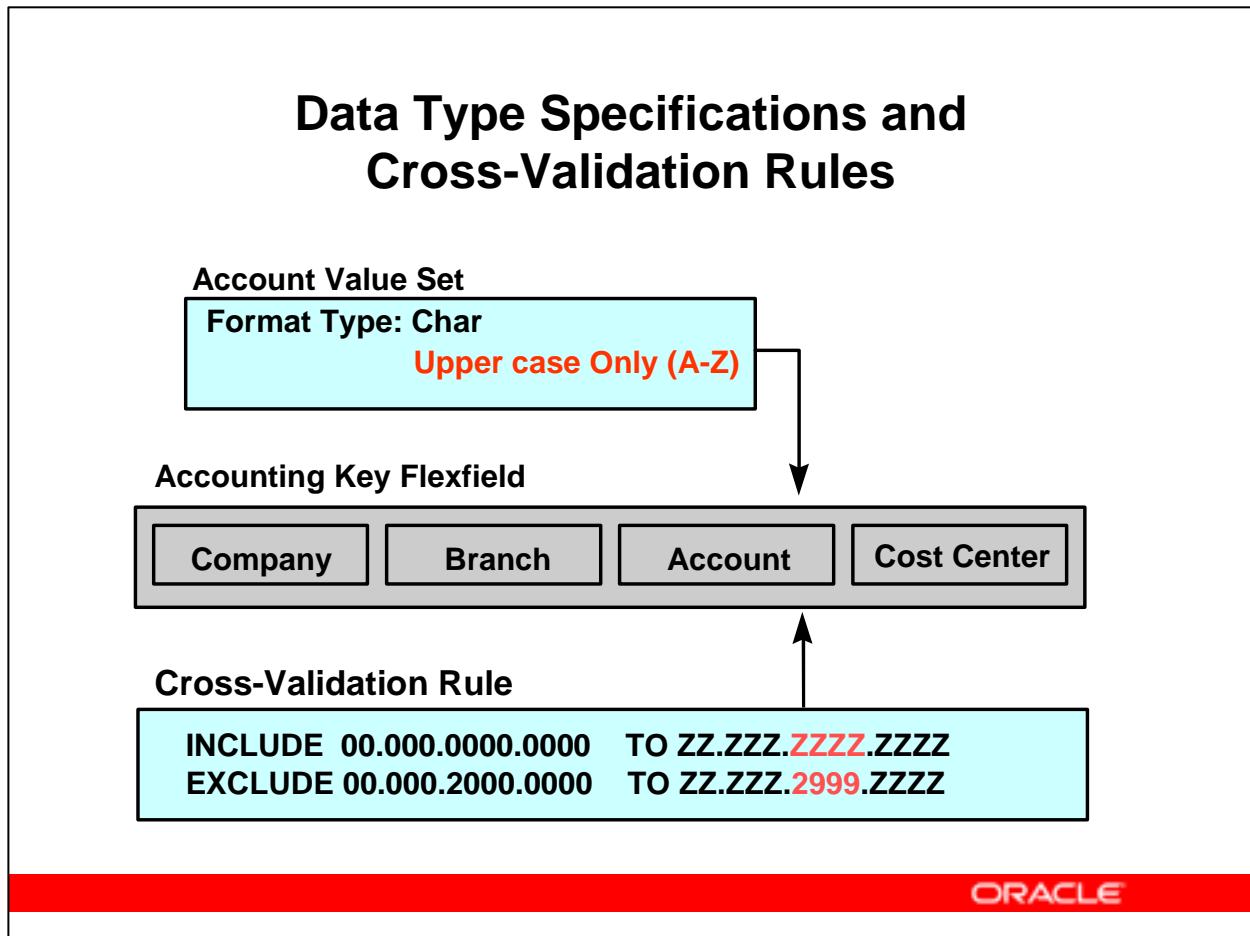
Type	From:	To:
INCLUDE	<i>value1.value2.value3</i>	<i>value4.value5.value6</i>
EXCLUDE	<i>value1.value2.value3</i>	<i>value4.value5.value6</i>

ORACLE

### Designing Rule Elements

Cross-validation rules consist of an Include element that includes all possible combinations, and one or more Exclude elements that restrict various segment value combinations. To make maintenance easier, use many simple rules rather than a few complex rules. To pass validation, a value combination must be included and not be excluded by any of the Exclude elements.

Rules are not retroactive, they apply only to segment value combinations entered after they are defined and enabled.



### Specifying Data Type and Format in Cross-Validation Rules

Cross-validation rules recognize any value characteristics specified for the value sets the segments use, such as Format Type, Right-Justify and Zero-Fill, Numbers Only, and Uppercase Only. Note that the Accounting flexfield does not use NULL values.

The Cross-Validation Rules window enforces the correct collating order for the platform.

- For most platforms (ASCII)  $0 < 000 < Z < ZZZ$
- For some platforms (EBCDIC)  $A < AAA < 9 < 999$

### Preparing for Cross-Validation

- **Before you can add cross-validation rules to an existing flexfield definition, you must unfreeze it in the Key Flexfield Segments window.**
- **After defining cross-validation rules, you must refreeze and recompile the flexfield definition in the Key Flexfield Segments window for the new validation rules to take effect.**

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

ORACLE



### Defining Cross-Validation Rules

Use the Cross-Validation Rules window to enter:

- **Name**
- **Description**
- **Error Message**
- **Error Segment**
- **Cross-Validation Rule Elements**

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

Because cross-validation rules can be defined from within multiple applications, the path to the Cross-Validation Rules window depends on which application you are using.

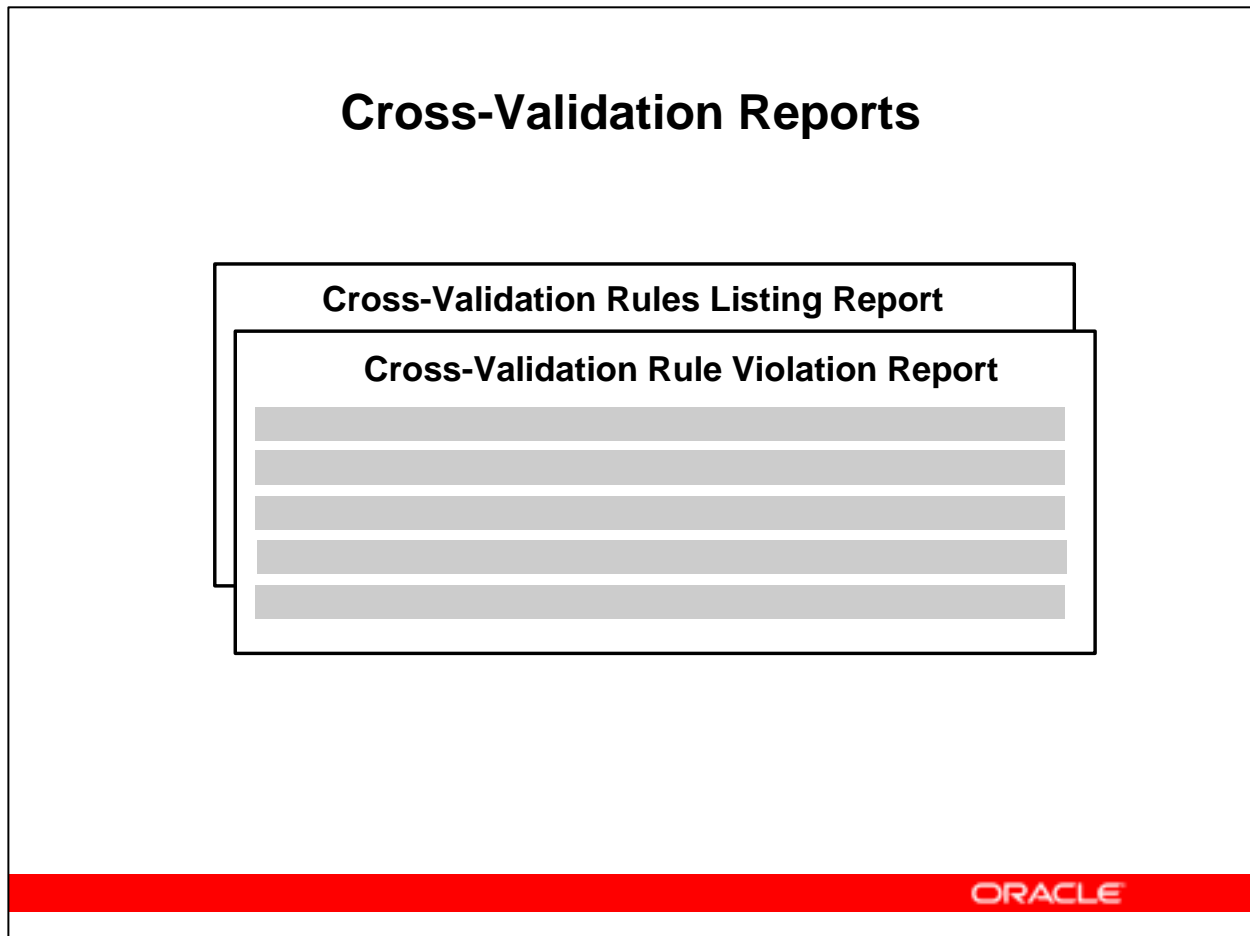
ORACLE

### Defining Cross-Validation Rules and Error Messages

**Define an Error Message and an Error Segment** - Enter a rule name and description for each cross-validation rule. Specify the error message to display and the segment where the cursor should move when a user enters an invalid combination. Use the segment where the error probably occurred. Enter optional start and end dates for your rule. You can also use these fields to disable a rule.

**Specify at Least One INCLUDE Element** - Unspecified values are always excluded, so every rule needs at least one Include statement. Otherwise no combinations will ever pass validation. As shown in the example, you can use the pop-up window to define which combinations to exclude. Exclude statements override Include statements. There can be multiple statements and each statement can specify values for multiple flexfield segments.

**Change the Flexfield Definition** - You must exit and sign on again or change responsibilities to see any changes.



### **Maintaining Cross-Validation Rules**

To minimize maintenance, plan cross-validation rules when first setting up the key flexfield. If later changes are necessary, review existing rules to ensure accurate and consistent validation. Disable previously existing combinations that are no longer valid according to the new rules.

#### **Using Reports to Maintain Cross-Validation Rules**

- The Cross-Validation Rule Violation Report offers a listing of all the previously created flexfield combinations that violate the cross-validation rules for a given structure. The report program can also disable the existing combinations that violate the new rules.
- The Cross-Validation Rules Listing report lists all the cross-validation rules that exist for a particular flexfield structure.

### Comparing Cross-Validation and Security Rules

Cross-Validation	Value Security
Applies to all users.	Applies only to users of the chosen responsibility.
Affects only key flexfields.	Affects key and descriptive flexfields as well as report parameters
Applies across an entire key flexfield structure.	Applies only to the value set used.

ORACLE

#### Interaction of Cross-Validation and Value Security Rules

Both cross-validation and value security control the data that a user can enter. The slide shows a comparison of the scope and action of each.

##### Cross-Validation

- Applies to all users regardless of responsibility
- Affects only key flexfields
- Applies across an entire key flexfield structure

##### Value Security Rules

- Apply only to users of the chosen responsibility
- Can affect key and descriptive flexfields as well as program parameters
- Apply only to the value set used by a flexfield segment or program parameter

# Practice - Cross-Validations

---

## Overview

Cross-validation rules enable you to restrict users from entering certain “account numbers,” or combinations of segment value codes. In this practice, you will use the Accounting Flexfield structure you defined earlier in the Define a KFF lesson. You will create a cross-validation rule to prevent the Orange company from using the Development cost center. Because cross-validation rules apply to all users, when you test your flexfield you will not be able to enter the combination of company and cost center excluded in the rule. Instead, the error message defined for the rule will appear, and the cursor will return to the segment specified as the error segment. Whenever you add or change cross-validation rules, you must recompile the flexfield structure to ensure that your changes are reflected in the applications.

## Tasks

### Define your Cross-Validation Rule

1. Use the Cross-Validation Rules window to define a cross-validation rule for the Accounting Flexfield structure you defined in the Define a KFF lesson, *YourInitials* Chart of Accounts. Name the rule *YourInitials\_CVR:01*, give the rule a description, and ensure that the rule is enabled. Enter an error message to explain the rule to users, and specify the CO segment as the error segment.

### Define your Cross-Validation Rule Elements

2. Define two cross-validation rule elements.
  - Define the first cross-validation rule element to include all the possible code combinations, from lowest to highest.
  - Define the second cross-validation rule element to exclude any code combinations that contain both the Orange company and the Development cost center.
  - After defining the cross-validation rule, recompile the flexfield structure definition.

### Test your Cross-Validation Rule

8. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
9. In the General Ledger Super User, navigate to (N) Setup > Financials > Books > Define
10. Enter the name *Your Initials* Set of Books.

11. Select your chart of accounts (the Accounting Flexfield structure you defined).
12. Navigate to the Budgetary Control region.
13. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
14. Enter values in the Accounting Flexfield to test your work.
15. Cancel your entries and exit without saving.

## Solution - Cross-Validations

### Define your Cross-Validation Rule

#### Responsibility = System Administrator

1. Navigate to (N) Application > Flexfield > Key > CrossValidation.
2. Query the *YourInitials* Chart of Accounts structure for the Accounting Flexfield in the Oracle General Ledger application.
3. In the Cross-Validation Rules region, enter the information for the rule in the following fields:
  - **Name:** *YourInitials\_CVR:01*
  - **Description:** *YourInitials* Cross-Validation Rule 01
  - **Enabled:** Selected
  - **Error Message:** The Orange company cannot use the Development cost center.
  - **Error Segment:** CO

### Define your Cross-Validation Rule Elements

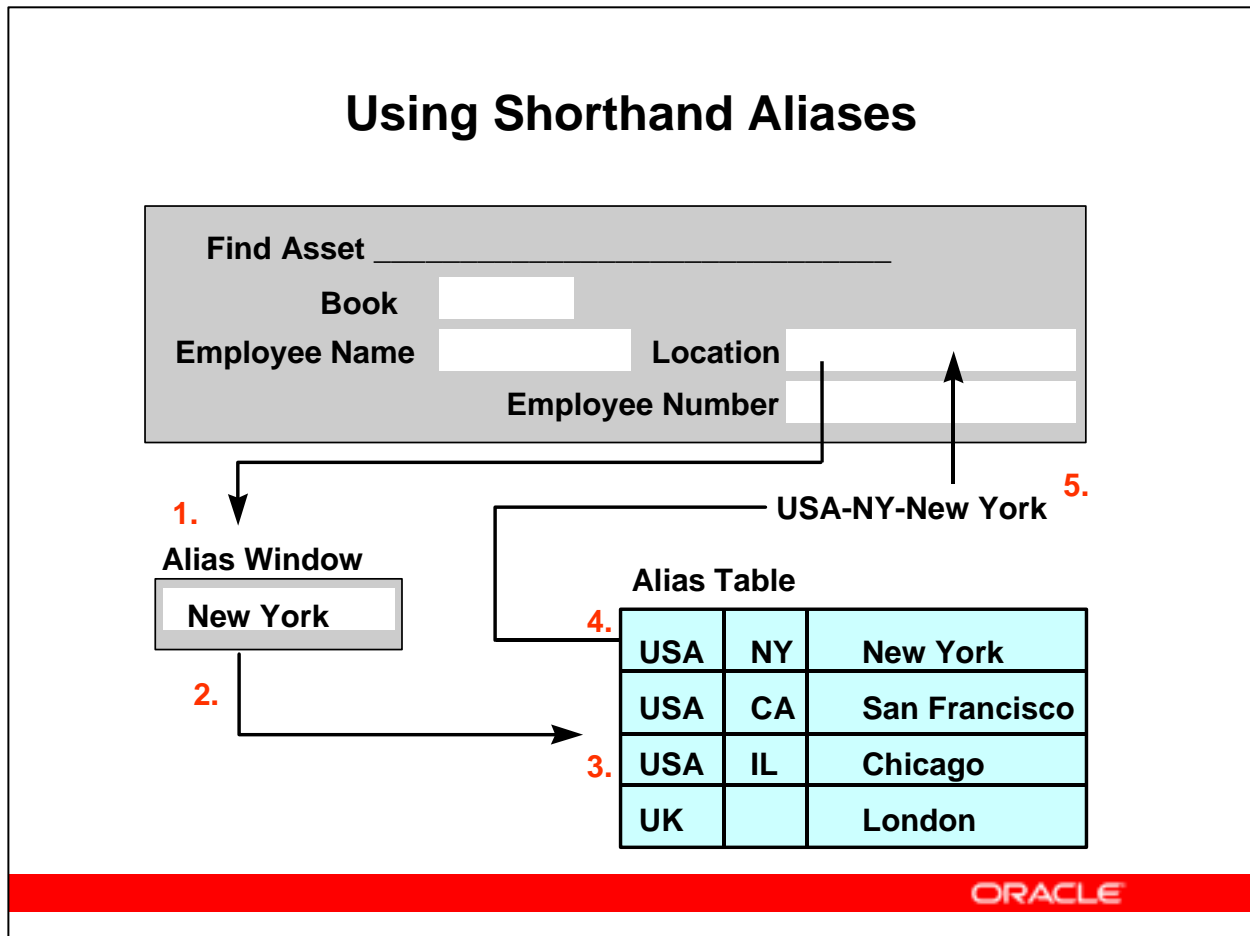
4. In the Cross-Validation Rule Elements region, enter the information for the first rule element in the following fields:
  - **Type:** Include
  - **From:** 00.000.0000.0000
  - **To:** *ZZ.ZZZ.ZZZZ.ZZZZ*
5. In the Cross-Validation Rule Elements region, enter the information for the second rule element in the following fields:
  - **Type:** Exclude
  - **From:** 02.300.0000.0000
  - **To:** 02.300.*ZZZZ.ZZZZ*
6. Save your work.
7. Navigate to (N) Application > Flexfield > Key > Segments.

8. Query the *YourInitials* Chart of Accounts structure for the Accounting Flexfield in the Oracle General Ledger application.
9. Click the “Compile” button to compile the flexfield definition.

### **Test your Cross-Validation Rule**

1. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
2. In the General Ledger Super User responsibility, navigate to (N) Setup > Financials > Books > Define.
3. Enter the name *YourInitials* Set of Books.
4. Select your chart of accounts (the Accounting Flexfield structure you defined).
5. Navigate to the Budgetary Control region.
6. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
7. Enter values in the Accounting Flexfield to test your work.
8. Cancel your entries and exit without saving.

## Using Shorthand Aliases



### Using Shorthand Aliases to Speed Data Entry

Whenever you notice that you are frequently entering the same combination of values into a multi-segment flexfield, you should consider defining an alias for that value combination. You can then use the alias to generate the complete set of values for the flexfield.

Select an alias from a pop-up list of possible aliases for the flexfield. You can also enter the alias name directly in the alias window.

Entering a shorthand alias automatically inserts the values for that alias into the key flexfield.



## Using Full and Partial Aliases

### Using Full and Partial Aliases

<p><b>Flexfield definition:</b></p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><p>Country State City</p></div> <p><b>Flexfield definition</b></p> <div style="border: 1px solid black; padding: 5px;"><p>Country State City Address</p></div>	<p><b>Full aliases (value for every segment)</b></p> <p><b>Partial aliases (some segments empty)</b></p>	<div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"><p>USA NY New York</p></div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"><p>USA CA San Francisco</p></div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"><p>USA NY New York</p></div> <div style="border: 1px solid gray; padding: 5px;"><p>USA CA San Francisco</p></div>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ORACLE

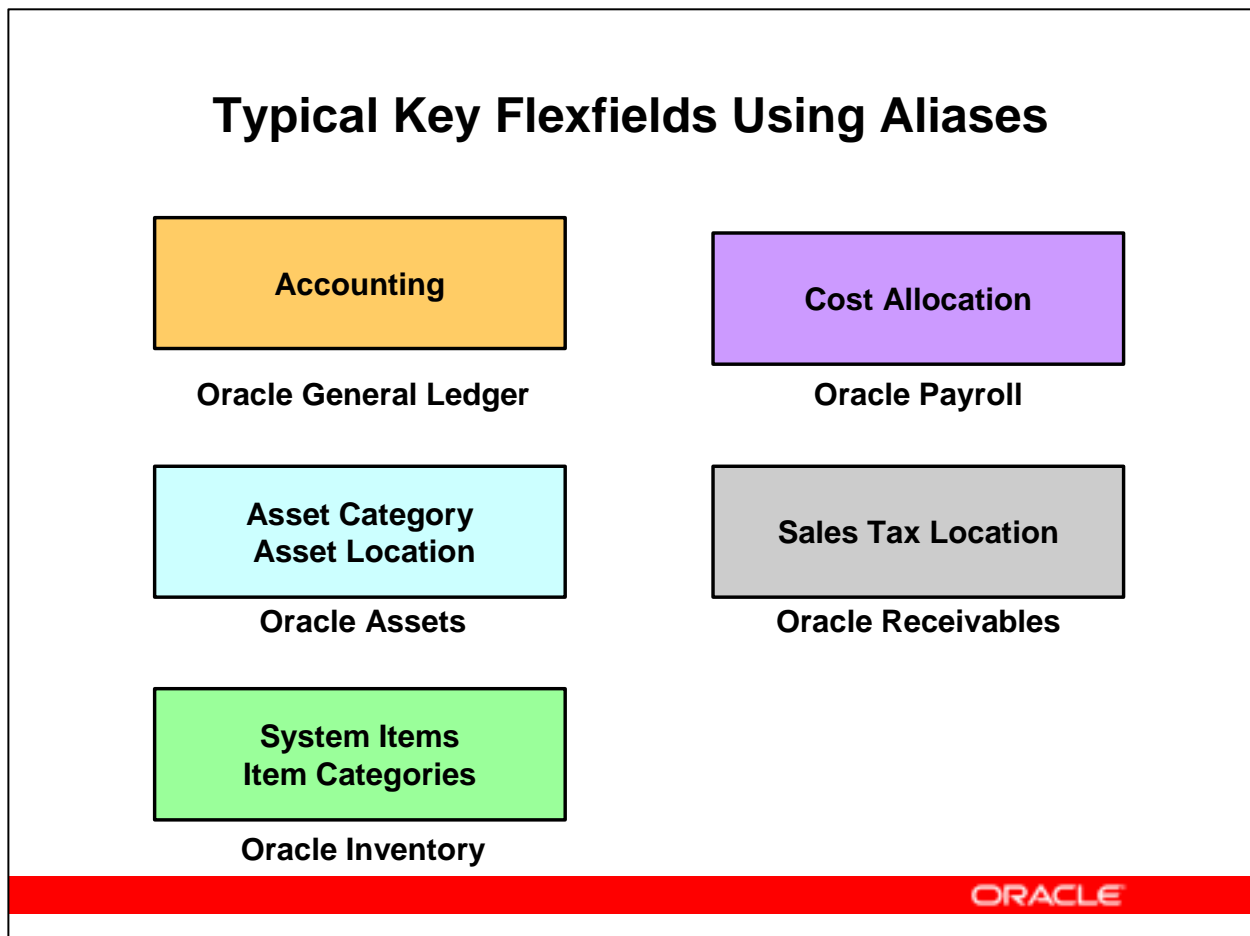
### Using Full and Partial Aliases

An alias can represent a complete flexfield combination or a set of partial segment values:

- If most combinations vary only in a single segment, create an alias with that segment blank and values specified for the remaining segments.
- If the alias represents a complete combination, once the alias name has been selected or entered, the shorthand window closes and the cursor moves to the next field.
- If the alias represents a partial flexfield value, the full flexfield window pops up with the cursor on the first empty segment.
- Pressing [Return] in a blank shorthand entry window also pops up the full flexfield window.
- Users can override values generated by aliases as appropriate.

## Typical Key Flexfields Using Aliases

---



### Typical Key Flexfields Using Aliases

Because aliases are used to speed up the entry of often-used value combinations, certain key flexfields in Oracle Applications are especially suited for alias entry.

### Defining an Alias

Use the Shorthand Aliases window to enter:

- Enabled
- Max Alias Size
- Prompt
- Alias
- Template
- Alias Description

Define multiple aliases for the same flexfield if necessary.

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

ORACLE

### Defining an Alias Template

- **If your alias needs a particular value at a particular location within the key value, enter the value at the appropriate place in the Template field.**
- **If your alias does not need a particular value for a location, you can simply leave that position empty.**
- **Enter the separators between positions.**
- **Having two separators next to each other with no intervening value is acceptable.**
- **Any values you define must pass any value set qualification as well as any cross-validation rules.**

ORACLE

### Recompiling the Flexfield Definition

**After you have specified any new aliases for a flexfield, you must recompile the flexfield definition to implement the new aliases. Use the Key Flexfield Segments window to:**

- **Unfreeze the definition**
- **Refreeze the definition**
- **Compile the definition**

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

ORACLE

### Flexfields: Shorthand Entry Profile Option

	Query	Add	Update
Not Enabled	–	–	–
New Entries Only	–	X	–
Query and New Entry	X	X	–
All Entries	–	X	X
Always	X	X	X

ORACLE

### Flexfields: Shorthand Entry Values

The Flexfields: Shorthand Entry profile option controls the behavior of aliases at the user level. The user has the ability to update this option as appropriate.

### **Flexfields: Show Full Value Profile Option**

If alias specifies values for all segments,

<b>No</b>	<b>Does not display full flexfield</b>
<b>Yes</b>	<b>Displays full flexfield with cursor in last segment</b>

ORACLE

### **Flexfields: Show Full Value Profile Option**

The Flexfields: Show Full Value Profile Option allows the user to turn off the full display of a key flexfield whenever the user specifies an alias with values for all segments of the flexfield.

### **Modifying Existing Alias Definitions**

**You may need to modify existing alias definitions after:**

- **Changing the order of flexfield segments**
- **Adding a new segment**
- **Disabling a segment**
- **Changing a segment length**

ORACLE

#### **Modifying Alias Definitions**

Since aliases are tied to the structure of the flexfield, any change to the flexfield structure must be reflected in any aliases providing values for that flexfield. Some of the typical modifications that also require changes to aliases are shown in the slide.\



# Practice - Shorthand Aliases

---

## Overview

Shorthand aliases speed data entry for end users. Aliases can be used either to define a complete key flexfield combination or to create a template that provides values only for some segments, letting users complete the flexfield entry. Users can determine whether to use aliases in the Flexfields: Shorthand Entry user profile option. In this practice, you will define both a complete shorthand alias and a partial alias template for the Accounting Flexfield structure you defined in the Define a KFF lesson.

## Tasks

### Define your Shorthand Alias

1. Use the Shorthand Aliases window to define shorthand aliases for the Accounting Flexfield structure, *YourInitials* Chart of Accounts, that you defined in the Define a KFF lesson. Enable aliases for this structure, specify a maximum size of 15, and specify a prompt of Shorthand.
2. Define an alias named *YourInitials\_SA\_1*, give the alias a description, and specify the complete combination 03.100.1110.0000.
3. Define an alias named *YourInitials\_SA\_2*, give the alias a description, and specify the partial combination 04...0000.

### Test your Shorthand Alias

8. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
9. In the General Ledger Super User, navigate to (N) Setup > Financials > Books > Define
10. Enter the name *Your Initials* Set of Books.
11. Select your chart of accounts (the Accounting Flexfield structure you defined).
12. Navigate to the Budgetary Control region.
13. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
14. Enter values in the Accounting Flexfield to test your work.
15. Cancel your entries and exit without saving.

## Solution – Shorthand Aliases

### Define your Shorthand Alias

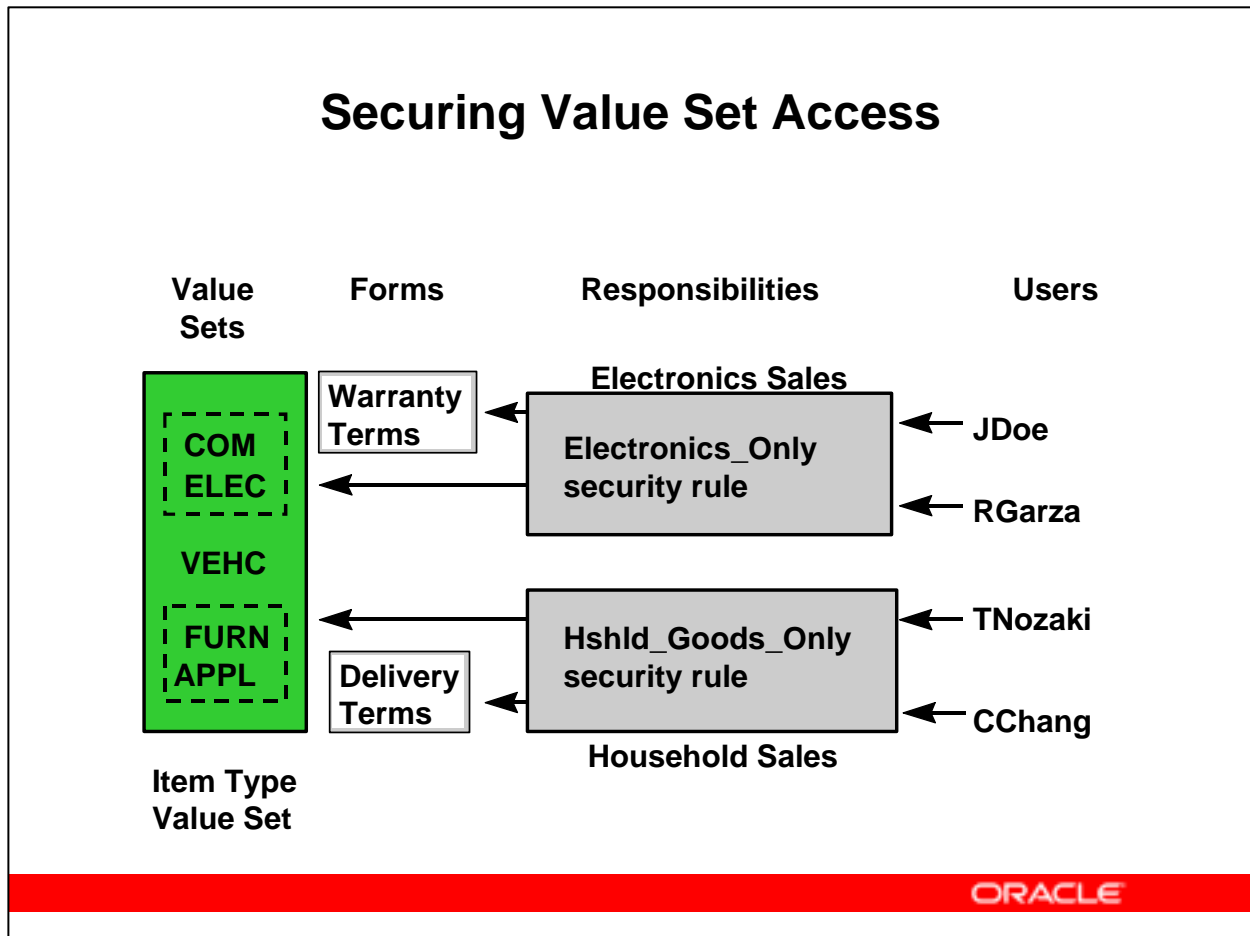
#### Responsibility = System Administrator

1. Navigate to (N) Application > Flexfield > Key > Aliases.
2. Query the *YourInitials* Chart of Accounts structure for the Accounting Flexfield in the Oracle General Ledger application.
3. In the Shorthand region, enter information in the following fields:
  - **Enabled:** Selected
  - **Max Alias Size:** 15
  - **Prompt:** Shorthand
4. In the Aliases, Descriptions region, enter information for the first alias in the following fields:
  - **Alias:** XX\_SA\_1
  - **Template:** 03.100.1110.0000
  - **Alias Description:** Yellow Sales Cash
5. Save your work.
6. In the Aliases, Descriptions region, enter information for the second alias in the following fields:
  - **Alias:** XX\_SA\_2
  - **Template:** 04...0000
  - **Alias Description:** Green
7. Save your work.
8. Navigate to (N) Application > Flexfield > Key > Segments.
9. Query the *YourInitials* Chart of Accounts structure for the Accounting Flexfield in the Oracle General Ledger application.
10. Click the “Compile” button to compile the flexfield definition.

## Test your Shorthand Alias

11. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
12. In the General Ledger Super User, navigate to (N) Setup > Financials > Books > Define
13. Enter the name *Your Initials* Set of Books.
14. Select your chart of accounts (the Accounting Flexfield structure you defined).
15. Navigate to the Budgetary Control region.
16. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
17. Enter values in the Accounting Flexfield to test your work.
18. Cancel your entries and exit without saving.

## Securing Value Set Access



### Controlling Access to Flexfield Values by Specifying Value Security Rules

Responsibilities are used within Oracle Applications to specify what operations a user can perform. Responsibilities limit the forms and reports a user can access. Responsibilities are equivalent to job descriptions and therefore usually map onto data access privileges as well. Users of the same responsibility generally need to see the same data.

You can define security rules to restrict users from using flexfields to enter or display certain data. Once you associate these rules with the appropriate responsibility, all users of that responsibility are restricted to the values allowed by that responsibility's security rule.

#### Example

In the example on the slide, a company has assigned its sales force to different responsibilities according to the type of items they sell. Two of these responsibilities are Electronic Sales and Household Sales. A value set called Item Type lists all the types of items the company sells. In this case, you can easily define a security rule to restrict the users of a particular responsibility from accessing item types not appropriate for their responsibility. For example the Hshld\_Goods\_Only security rule allows access only to the FURN and APPL item types.

### Security and Value Sets

#### **Flexfield security restricts values within a value set:**

- **Value sets used by key flexfields segments, descriptive flexfield segments, and request parameter are securable.**
- **Only Independent, Dependent, or Table-validated value sets are securable.**
- **List of values displays only approved values.**
- **Some forms allow display of restricted values but not update.**
- **Security does not restrict non-flexfield data entry.**

ORACLE

#### **How Value Security Works**

You can apply security to the value sets used by key flexfield segments, descriptive flexfields segments, and request parameters.

- Security rules have no effect for segments or parameters that use value sets with a validation type of None, Special, Pair, Translatable Independent, or Translatable Dependent, or for segments without a value set.
- Security rules provide data entry and querying control. Once the security rule is associated with a responsibility, a list of values displays only the values appropriate to that responsibility.
- Some forms permit querying of restricted values but do not permit updating of restricted values.
- Security rules apply only to data entered through the forms on which the flexfield appears. They do not apply to data created or copied automatically by the applications themselves. Therefore, be careful when loading data without using Oracle Application forms.

### Constructing Security Rules

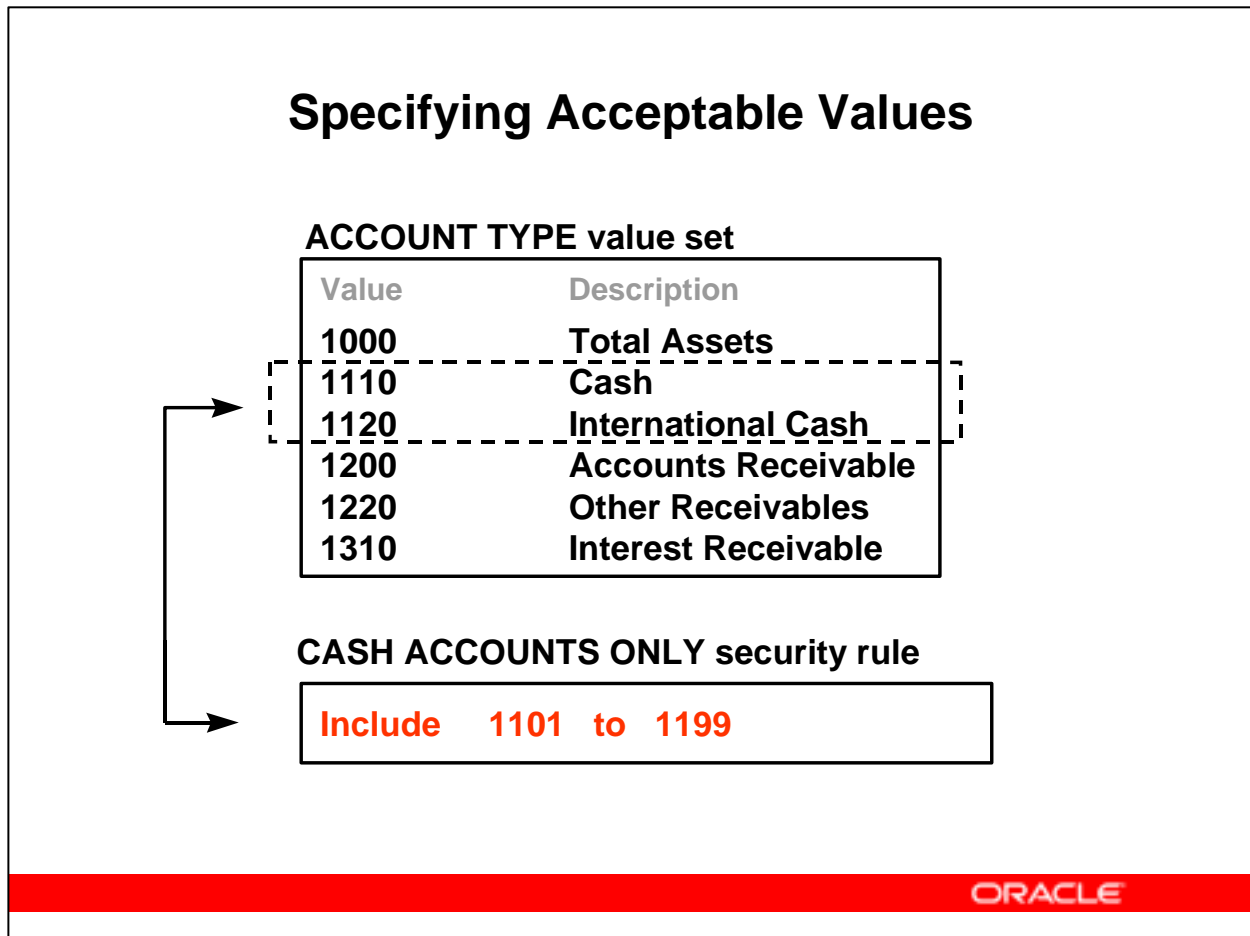
Element	From:	To:
<b>INCLUDE</b>	<i>beginning segment value</i>	<i>ending segment value</i>
<b>EXCLUDE</b>	<i>beginning segment value</i>	<i>ending segment value</i>

ORACLE

#### Security Rule Elements

- Security rules are defined in terms of the values in the value set.
- Certain values or value ranges are either “included” or “excluded” by a rule.
- The same rule can specify both values to be included as well as different values to be excluded.
- By default all values are excluded unless explicitly included, and exclusion tests override inclusion tests.

## Specifying Acceptable Values



### Specifying Security Rules

Whenever possible, restrict access to ranges of values. Plan values carefully to make security rules easy to define.

#### Example

Assume you have a value set ACCOUNT TYPE. Since the accounts are distinguished by the account type number, you can define a rule including all account types in the 1100 account type number series. This rule can later be assigned to an appropriate responsibility to restrict users from accessing anything but cash account types.

## Using Multiple Clauses

---

### Using Multiple Clauses

#### WESTERN REGION ONLY security rule

INCLUDE	CA	CA
INCLUDE	OR	OR
INCLUDE	WA	WA

#### INTEREST ACCOUNTS ONLY security rule

INCLUDE	000.00.200.00	000.00.299.00
---------	---------------	---------------

ORACLE

### Using Multiple Elements

As shown on the previous slides, you define security rules in terms of exclude or include clauses. You can combine these clauses as necessary to effect the restrictions you need. A security rule can therefore consist of one clause or many.

You can apply security rules to more than one responsibility, and a single responsibility can have more than one security rule associated with it. It is not unusual for several responsibilities to share a broadly defined security rule with some of the responsibilities also having more restrictive security rules.

#### Example

The examples on the slide show two rules: one containing multiple clauses and one consisting of only one clause.



## Rules with Overlapping Values

		Zip Code
<b>One Rule, Three Clauses</b>		
<b>INCLUDE 84200 to 84500</b> <b>INCLUDE 84400 to 84700</b> <b>EXCLUDE 84534 to 84534</b>	<b>(Union)</b>	<i>84201</i>
		<i>84312</i>
		<i>84423</i>
		84534
		<i>84678</i>
		84789
<b>Three Rules</b>		
<b>INCLUDE 84200 to 84600</b>	<b>(Intersection)</b>	84201
<b>INCLUDE 84400 to 84700</b>		84312
<b>EXCLUDE 84534 to 84534</b>		<i>84423</i>
		84534
		84678
		84789

**ORACLE**

### Interaction of Security Rules and Elements

Since a responsibility can have multiple security rules for the same value set, it is important to consider the interaction of security rules.

In a single security rule:

- Everything is excluded unless explicitly included.
- The clauses have a logical OR relationship to one another so that the result is a union of all included values.
- If a value satisfies the criteria for both an include and an exclude, it is excluded.

In multiple security rules:

- More rules restrict more not less. All values must pass all security rules to be acceptable.
- The different rules have a logical AND relationship to one another so that the intersection of the rule's included values applies.
- If a value satisfies the criteria for both an include and an exclude, it is excluded.

### Example

The slide shows two examples of security rules operating on a value set. The values approved by the different rules are shown in italics.

## Rules Without Overlapping Values

		Zip Code
<b>One Rule, Three Clauses</b>		
<b>INCLUDE 84200 to 84400</b> <b>INCLUDE 84500 to 84800</b> <b>EXCLUDE 84534 to 84534</b>	<b>(Union)</b>	<b>84201</b>
		<b>84312</b>
		84423 84534 <b>84678</b> <b>84789</b>
<b>Three Rules</b>		
<b>INCLUDE 84200 to 84400</b>	<b>(Intersection)</b>	84201
<b>INCLUDE 84500 to 84800</b>		84312
<b>EXCLUDE 84534 to 84534</b>		84423 84534 84678 84789

**ORACLE**

### Rule Results Without Overlapping Ranges

If the values specified by the clauses do not overlap, the result sets are different:

- The result set for a responsibility with a single security rule with multiple clauses contains all values that satisfy any of the include clauses. Any excluded values are removed.
- The result set for a responsibility with multiple rules contains no members if there are no values that satisfy all the rules.

### **Implementing Flexfield Security**

- 1. Enable security on the value set.**
- 2. Enable security on the flexfield segments and request parameters that use this value set.**
- 3. Define your security rule.**
- 4. Assign your security rule to a responsibility.**

ORACLE

### **Enabling Security for a Value Set**

**To enable security for a value set, check the Security Available check box in the Value Sets window. You must enable security for the value set before you can define security rules for that value set.**

**(N) Application—>Validation—>Set**

ORACLE

### Enabling Security for a Segment

- Check the **Security Enabled** check box in the **Segments** window to enable security for a flexfield segment.
- **Compile the flexfield structures** after making the change.

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

The path to the Segments window depends on which application you are using. This path is in the System Administrator responsibility.

ORACLE

### **Enabling Security for a Segment**

- **If you are changing an existing flexfield definition to add security, you must first unfreeze the flexfield definition. After you make the change, refreeze the definition and recompile.**
- **Users currently logged on will not see the change until they exit or change responsibilities.**

ORACLE

### **Enabling Security for a Program Parameter**

**To enable security for a concurrent program parameter, check the Enable Security check box in the Concurrent Program Parameters window.**

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

ORACLE

### Defining Security Rules

**Use the Define Security Rules window to enter:**

- **Name**
- **Description**
- **Message**
- **Security Rule Elements (Include/Exclude)**

**(N) Security—>Responsibility—>Value Set—>Define**

**The path to the Define Security Rules window depends on which application you are using. This path is in the System Administrator responsibility.**

ORACLE

### Defining Security Rules

- To define security information for an existing value set, search for that value set's definition by its value set name, or by any flexfield segment or concurrent program parameter that uses that value set. Specify how you wish to search by selecting the appropriate option. Security must be enabled for the value set, or it will not appear.
- Specify include and exclude ranges for values.
- An automatic exclude rule excludes all values not explicitly included. Rules without elements therefore exclude everything.
- Without any rules, anyone can see and use any segment value.
- On most computers,  $0 < Z$ .



### Assigning Security Rules

Use the Assign Security Rules window to enter:

- Application
- Responsibility
- Name

**(N) Security—>Responsibility—>Value Set—>Assign**

**The path to the Assign Security Rules window depends on which application you are using. This path is in the System Administrator responsibility.**

ORACLE

### Assigning Security Rules

- **Assign value security for value sets, flexfield segments, or concurrent programs by selecting the appropriate radio button.**
- **Multiple security rules can be applied to the same responsibility, and the same security rule can be applied to multiple responsibilities.**

ORACLE

# Practice - Security Rules

---

## Overview

Flexfield security rules control access to specific segment values by responsibility. These rules can be set up for any or all segments. Security rules are tied to a responsibility, which is in turn associated with users. In this practice, you will create a rule that prevents users from viewing an account. You will assign the rule to the General Ledger Super User responsibility. When you log in to this responsibility to test your key flexfield, this account will not appear in the list of values. **Note:** Before you can use security for a flexfield segment, you must enable security both at the value set level and at the segment level.

## Tasks

### Define your Security Rule

1. Use the Define Security Rules window to define a security rule for the ACCT segment of the Accounting Flexfield structure, *YourInitials* Chart of Accounts, that you defined in the Define a KFF lesson. Name the rule *YourInitials\_SR01*, and give the rule a description. Enter an error message to explain the rule to users.
  - Define the first security rule element to include all the possible account values, from 0000 to zzzz.
  - Define the second security rule element to exclude the account 4110.

### Assign your Security Rule

2. Use the Assign Security Rules window to assign your security rule to the General Ledger Super User responsibility in the Oracle General Ledger application.

### Test your Security Rule

3. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
4. In the General Ledger Super User responsibility, navigate to (N) Setup > Financials > Books > Define.
5. Enter the name *YourInitials* Set of Books.
6. Select your chart of accounts (the Accounting Flexfield structure you defined).
7. Navigate to the Budgetary Control region.

8. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
9. Enter values in the Accounting Flexfield to test your work.
10. Cancel your entries and exit without saving.

## Solution – Security Rules

### Define your Security Rule

#### Responsibility = System Administrator

1. Navigate to (N) Security > Responsibility > ValueSet > Define.
2. In the Find window, select Key Flexfield and find the ACCT segment of the *YourInitials* Chart of Accounts structure for the Accounting Flexfield in the Oracle General Ledger application.
3. In the Security Rules region of the Define Security Rules window, enter information in the following fields:
  - **Name:** *YourInitials\_SR01*
  - **Description:** *YourInitials* Security Rule 01
  - **Message:** GL responsibility cannot use account 4110.
4. In the Security Rule Elements region, enter information for the first rule element in the following fields:
  - **Type:** Include
  - **From:** 0000
  - **To:** zzzz
5. In the Security Rule Elements region, enter information for the second rule element in the following fields:
  - **Type:** Exclude
  - **From:** 4110
  - **To:** 4110
6. Save your work.

### Assign your Security Rule

7. Navigate to (N) Security > Responsibility > Value Set > Define (B) Assign
8. In the Security Rules region, enter information in the following fields:
  - **Application:** Oracle General Ledger

- **Responsibility:** General Ledger Super User
- **Name:** *YourInitials\_SR01*

9. Save your work.

### **Test your Security Rule**

10. After defining your value sets, segments, and values, navigate to the Accounting Flexfield to test the results of your work. To view the Accounting Flexfield, you will perform the beginning steps in the process for creating a new set of books. However, **you must not save your work!** You will not be completing the set of books definition.
11. In the General Ledger Super User responsibility, navigate to (N) Setup > Financials > Books > Define.
12. Enter the name *YourInitials* Set of Books.
13. Select your chart of accounts (the Accounting Flexfield structure you defined).
14. Navigate to the Budgetary Control region.
15. Navigate to the Reserve for Encumbrance field and display the list of values. The Accounting Flexfield appears.
16. Enter values in the Accounting Flexfield to test your work.
17. Cancel your entries and exit without saving.

### Summary

- **Cross-validation rules control combinations of segment values.**
- **Cross-validation rules specify the segment values to be either included or excluded.**
- **Frequently entered combinations of values are candidates for shorthand aliases.**
- **Define aliases and enable shorthand alias entry.**
- **Use security rules to control access through a flexfield.**
- **Security rules are applied to a value set.**
- **Security can be enabled for any flexfield segment or report parameter that uses that value set.**

ORACLE

### Summary

Cross-validation rules are used to restrict invalid combinations of segment values. Rules are defined by specifying the value combinations that are either allowed (included) or disallowed (excluded). Cross-validation rules are not tied to a responsibility like security rules. You can combine cross-validation rules as needed.

Short aliases provide a way to speed entry of frequently used value combinations. Any key flexfield is a candidate for shorthand alias entry.

You can control the access to data by defining value security rules. These rules list the allowable values for display or selection through a flexfield segment. The rules are applied to the value set defined for that flexfield segment. Once you have defined the rules, you assign them to a responsibility. All users of that responsibility are then restricted to the approved values when using the flexfield's segment.





# Defining Descriptive Flexfields

## Chapter 17

# Defining Descriptive Flexfields

ORACLE

### Objectives

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

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

ORACLE

### Lesson Topics

This lesson describes the process required to plan a descriptive flexfield.

At the end of this lesson, you should be able to:

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

### Overview

- **Identifying and organizing data**
- **Arranging different descriptive flexfield layouts**
- **Specifying the flexfield mechanics**
- **Specifying descriptive flexfield attributes**
- **Defining a descriptive flexfield with global segments**
- **Defining a descriptive flexfield with both global and context-sensitive segments**

ORACLE

### Lesson Overview

Descriptive flexfields allow the user to gather additional information beyond that gathered by Oracle Applications. The information gathered by a descriptive flexfield and the structure of that information can vary between different users. This lesson covers analyzing and organizing the additional data, designing the different flexfields layouts, and specifying how the descriptive flexfield will behave.

This lesson also covers the mechanics of defining a descriptive flexfield. Flexfields consisting of only global segments as well as both global and context-sensitive segments are covered.

### Planning a Descriptive Flexfield

- **Identify the flexfield to be implemented.**
- **Determine the resources available for the flexfield.**
- **Identify and organize the information to be collected.**
- **Design the layout of the flexfield to reflect how the information is used.**
- **Design the behavior of the flexfield.**
- **Define the flexfield, freeze, and compile.**

ORACLE

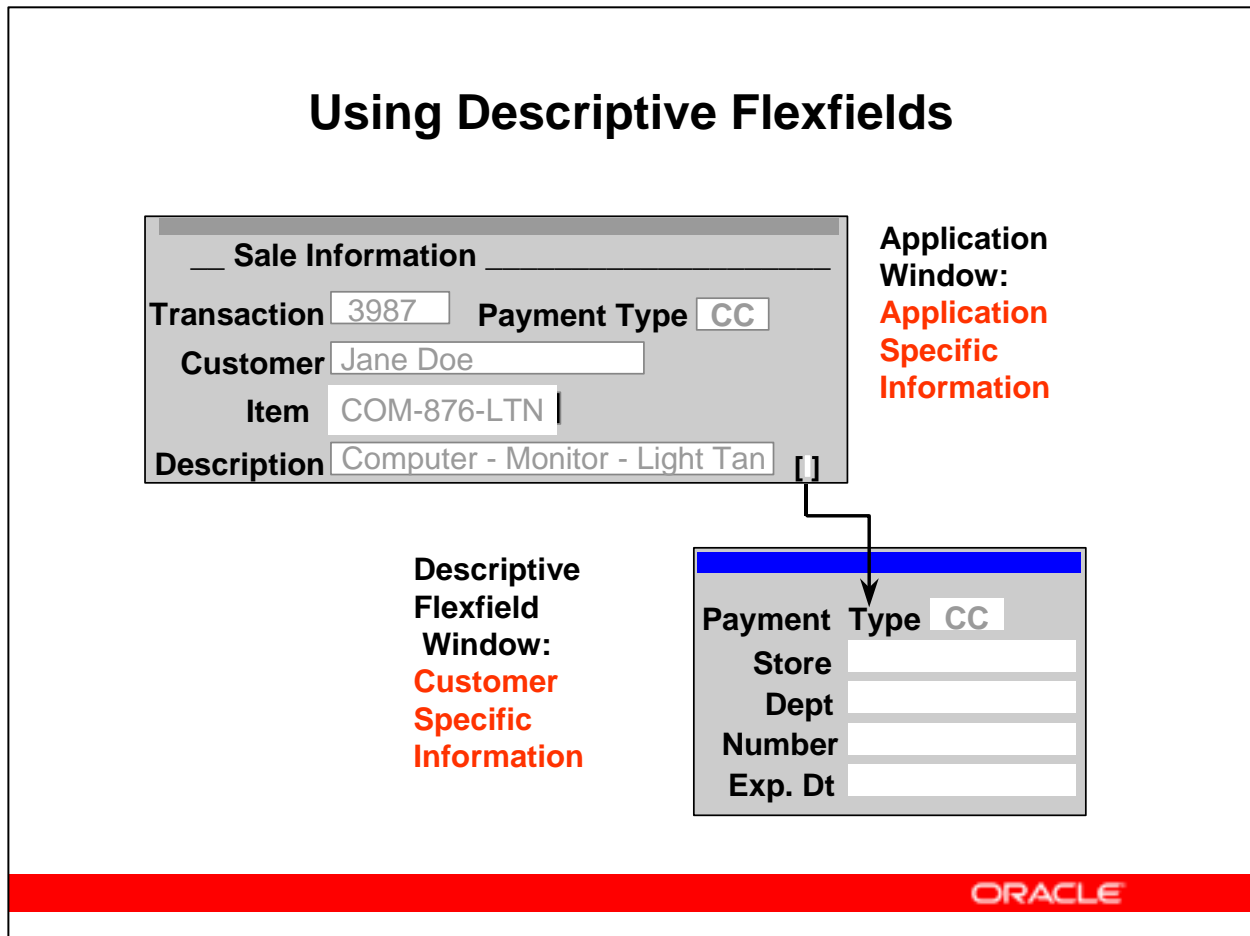
### Implementing a Descriptive Flexfield

Descriptive flexfields provide an easy and powerful way to extend the processing of Oracle Applications. However, since descriptive flexfield structures can be organized in many ways, you should spend some time in planning the new flexfield to ensure a successful result. Follow these steps to implement your descriptive flexfield:

- 1 Identify the flexfield to implement. Descriptive flexfields do not exist in a vacuum. They are logically related to the form on which they appear. Once you decide to implement the descriptive flexfield on a form, you must identify which descriptive flexfield it is.
- 2 Determine the system resources available to you. The number of segments available for use on your flexfield depends on the number of underlying columns in the base table. Since this will control the number of segments you have available for use, you must find this information before you can plan your design.
- 3 Identify the items of information you wish to gather. The information you gather should be logically related to the other information on the form.
- 4 Design the flexfield layout. Determine the number of segments and how they will be combined into structures.

- 5 Design the flexfield behavior. If you have multiple structures, determine how to process the different contexts.
- 6 Define the actual flexfield. Use the descriptive flexfield definition windows to enter your definition. Freeze and compile the definition to make the flexfield available to other users.

## Using Descriptive Flexfields



### Using Descriptive Flexfields

Descriptive flexfields have many uses in Oracle Applications:

- Flexfields expand Oracle Applications processing without programming.
  - Descriptive flexfields provide user-customizable expansion space in forms by enabling built-in blank fields to store extra data.
  - Each installation of Oracle Applications may use descriptive flexfields differently.
- You can use different structures for different contexts.
  - Use different segments depending on other information in the form or the descriptive flexfield.
  - Allow only the appropriate context-sensitive segments to appear.
- Flexfields save space.
  - Non-essential information resides in a descriptive flexfield that pops up only when the information becomes necessary.
  - If no extra fields are needed, the descriptive flexfield occupies little additional space on the form.

### Typical Descriptive Flexfield Information

- **Benefits information**
- **Calendar information**
- **Labor cost information**
- **Lease information**
- **Currency exchange information**
- **Payment information**
- **Credit information**
- **Budget information**
- **Distribution system information**

ORACLE

#### Typical Information Collected by Descriptive Flexfields

The slide shows typical business areas in which you may need to track additional or specialized information. You can easily define a descriptive flexfield to gather and store as much extra data as is required for your business needs.



## Some Oracle Applications Descriptive Flexfields

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

**ORACLE**

### Oracle Applications Descriptive Flexfields (Partial)

There are many descriptive flexfields available for use within Oracle Applications. The slide shows only some of the flexfields available.

### Identifying a Descriptive Flexfield

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

ORACLE

### **Determining the Descriptive Flexfield Name**

**Since descriptive flexfields are not labeled on the base form, once you recognize a descriptive flexfield you wish to implement, use the following procedure to determine the name of the flexfield:**

- 1. Click on a field in the same block in which the descriptive flexfield appears.**
- 2. Select Help—>Diagnostics—>Examine. This opens a window showing information on the selected field. Note the name of the block in which the field is located.**

**ORACLE**

### **Determining the Descriptive Flexfield Name**

- 3. Click the Block list of values button to display a list of the available blocks for this form. Select \$DESCRIPTIVE\_FLEXFIELD\$.**
- 4. Click the Field list of values button to display a list of the descriptive flexfields for this form. Each entry is prefixed by the name of the block in which the descriptive flexfield appears. Find the entries for the block whose name you determined in step 2.**
- 5. Select the flexfield you wish to implement from the entries for that block. The user name of the descriptive flexfield appears in the Value field.**

ORACLE

### Determining Available Resources

**Use the list of values for the Column field in the Segments Summary window to determine how many segments you can plan to use.**

**(N) Application—>Flexfield—>Descriptive—>Segments  
(B) Segments**

ORACLE

#### Determining the Available Resources

Once you have identified the descriptive flexfield you want to implement, you need to determine how many segments you can plan for. To do this, you need to know the number of ATTRIBUTE columns in the underlying table.

Find the flexfield definition and navigate to the Segments Summary window for that flexfield. Use the list of values on the Column field to display a list of the attribute columns. You will use this list of values again later to assign a segment to an underlying column, but you can also use it for planning now. The columns are numbered sequentially, so the highest numbered column tells you how many segments you can use.

### Planning Questions

- **What additional information needs to be captured?**
- **Is there any information you need to capture every time?**
- **Is there information you need to capture on an ad hoc basis?**
- **Can the need for capturing the ad hoc information be conditioned on a value in a base window?**
- **How much control over the window processing do you want to give to the user?**

ORACLE

### Identifying Your Information Needs

Once you know what resources you have available, you can begin to plan the layout of the flexfield.

First determine your information needs. Some of the questions to ask are shown on the slide. Before you can start designing the flexfield structure, you should know what information needs to be gathered by this flexfield, and how the information will be used.

## Identifying Your Information

---

**Identifying Your Information**

**Store number**                      **Check number**

**Credit card number**

**Expiration date**                      **Down payment**

---

**ORACLE**

### Identifying Needed Information

Assume you are planning a descriptive flexfield that will gather additional sales payment information. Some of the possible items of data in which you might be interested appear on the slide.

## Grouping Information By Usage

---

### Grouping Information By Usage

<b>Situation 1: (finance)</b>	<b>Store number</b>	<b>Down payment</b>	
<b>Situation 2: (check)</b>	<b>Store number</b>	<b>Check number</b>	
<b>Situation 3: (credit card)</b>	<b>Store number</b>	<b>Credit card number</b>	<b>Expiration Date</b>

**ORACLE**

### Organizing Information by Usage

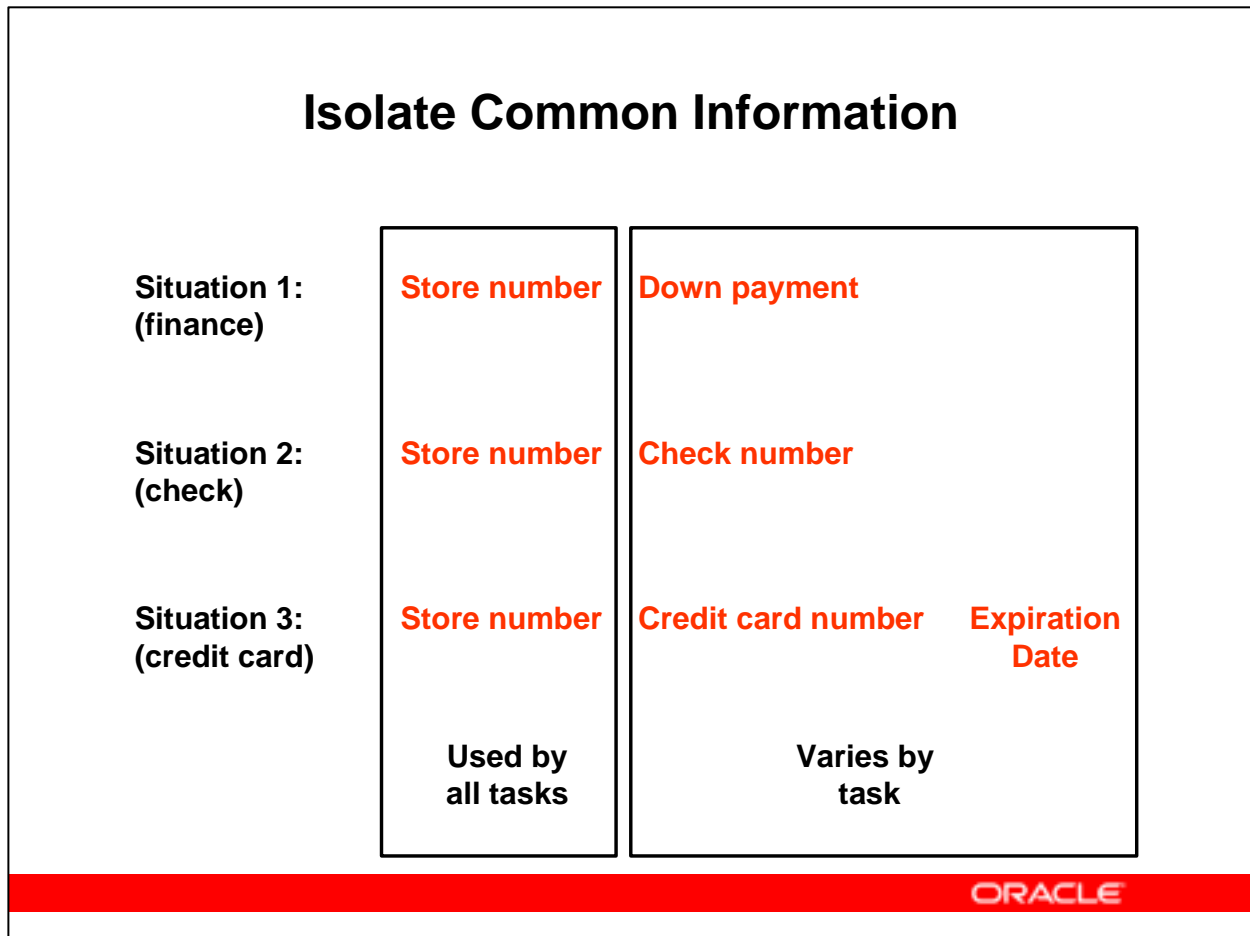
After you have identified all the items of information you want to gather, organize them by usage. Are all the items used all the time? Are all the items used in the same way?

The slide shows three different payment situations and the items of information appropriate for each situation.



## Isolate Common Information

---



### Isolate Common Information

After you have organized the items of information by usage, isolate any items that occur in all situations. You define the information used by all tasks in one structure and the information that varies by task in another, task-specific structure.



### Descriptive Flexfield Components

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

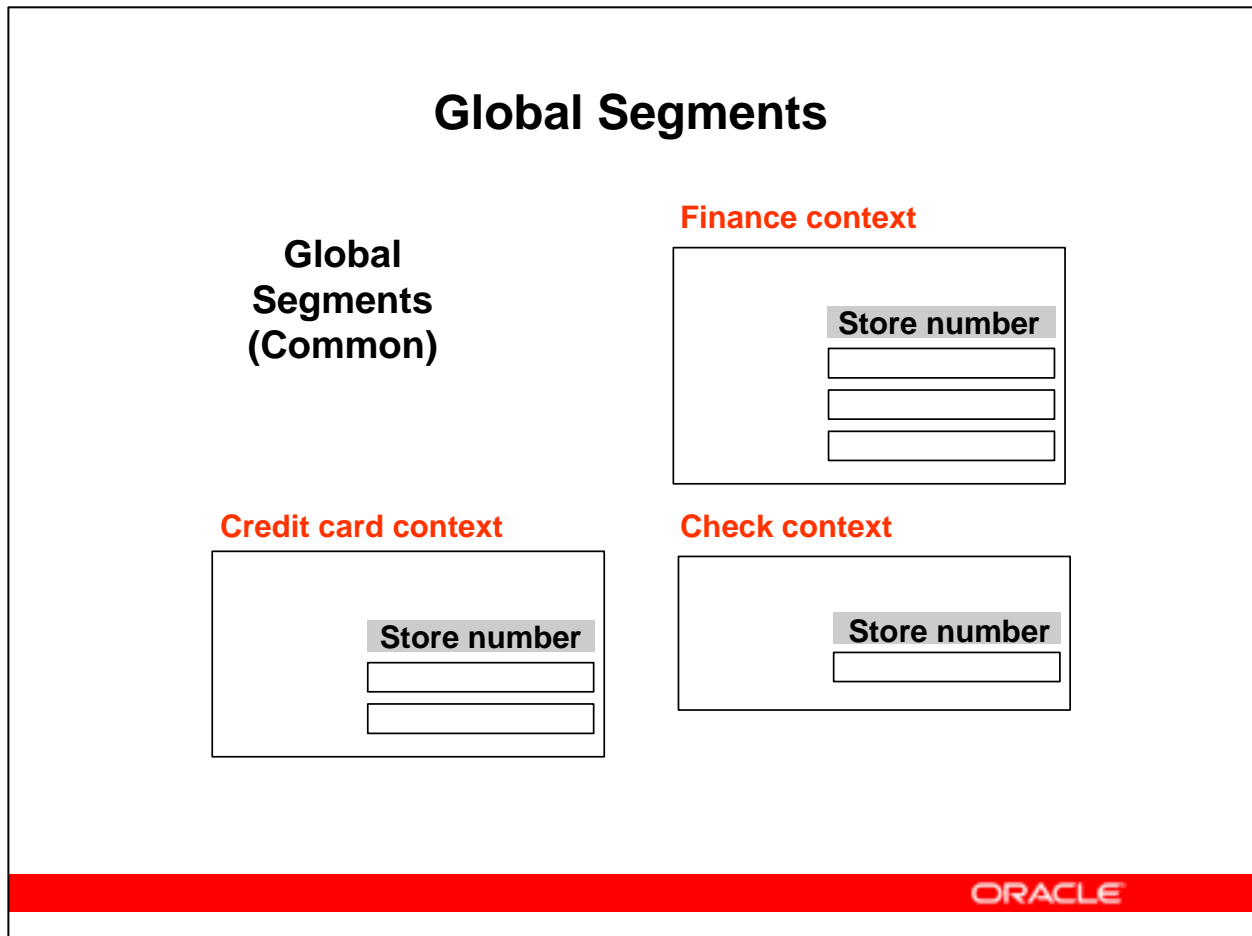
ORACLE

### Descriptive Flexfield Components

Descriptive flexfields are constructed from segments. Each segment contains one item of information. Since the same flexfield can be used by different contexts, and each context needs different items of information, you need to design different layouts for the same flexfield to support the different contexts.

Specify your layout in terms of global segments and context-sensitive segments.

# Global Segments



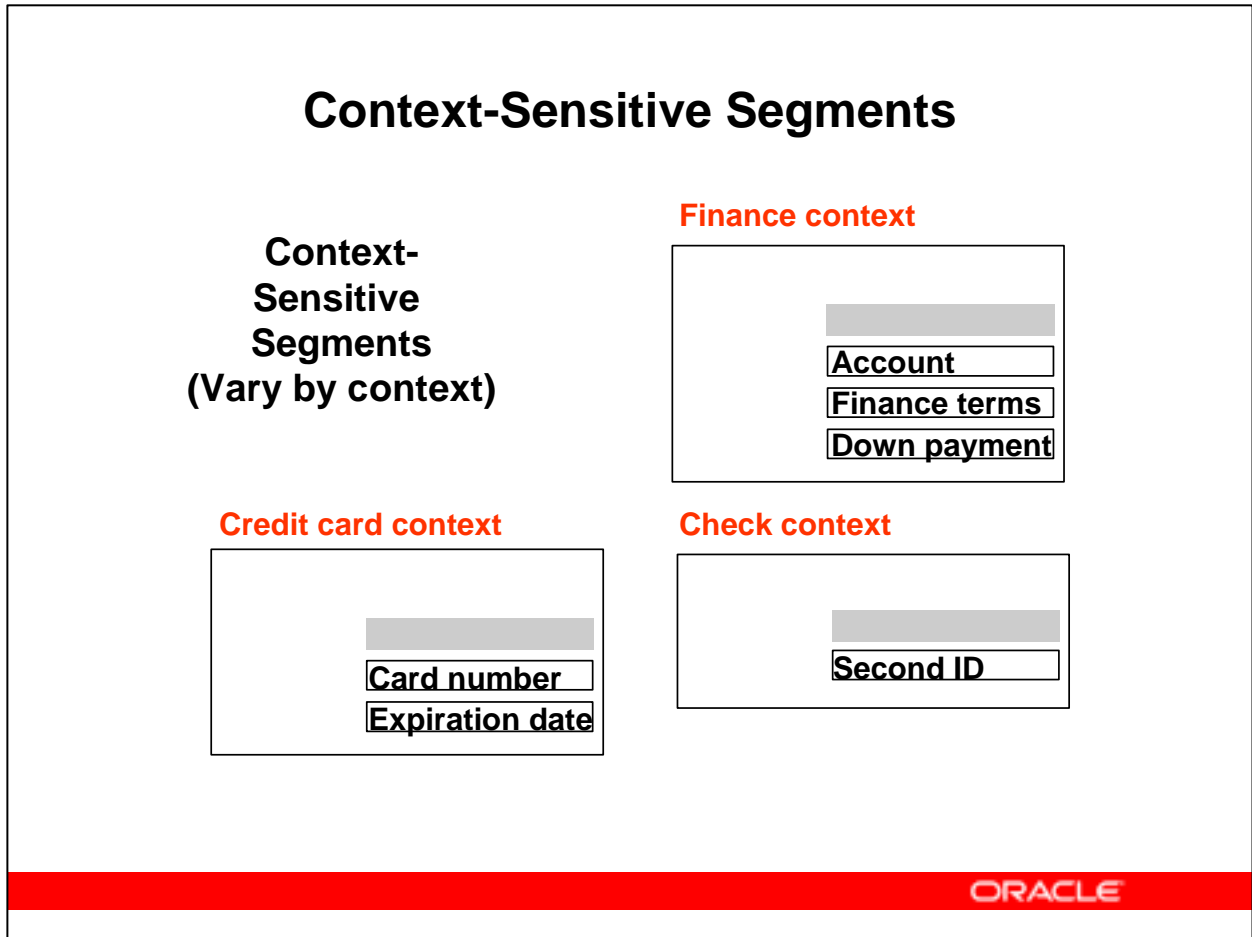
## Using Global Segments

Global segments are segments that appear regardless of context. Always plan your global segments first. Some descriptive flexfields use only global segments.

Continuing the payment information example, the slide shows that the store number is appropriate for all contexts. Thus it is an obvious global segment.

Global segments are the easiest to define. However, they may use up the allotted columns. Columns used for global segments cannot hold an context-sensitive segments.

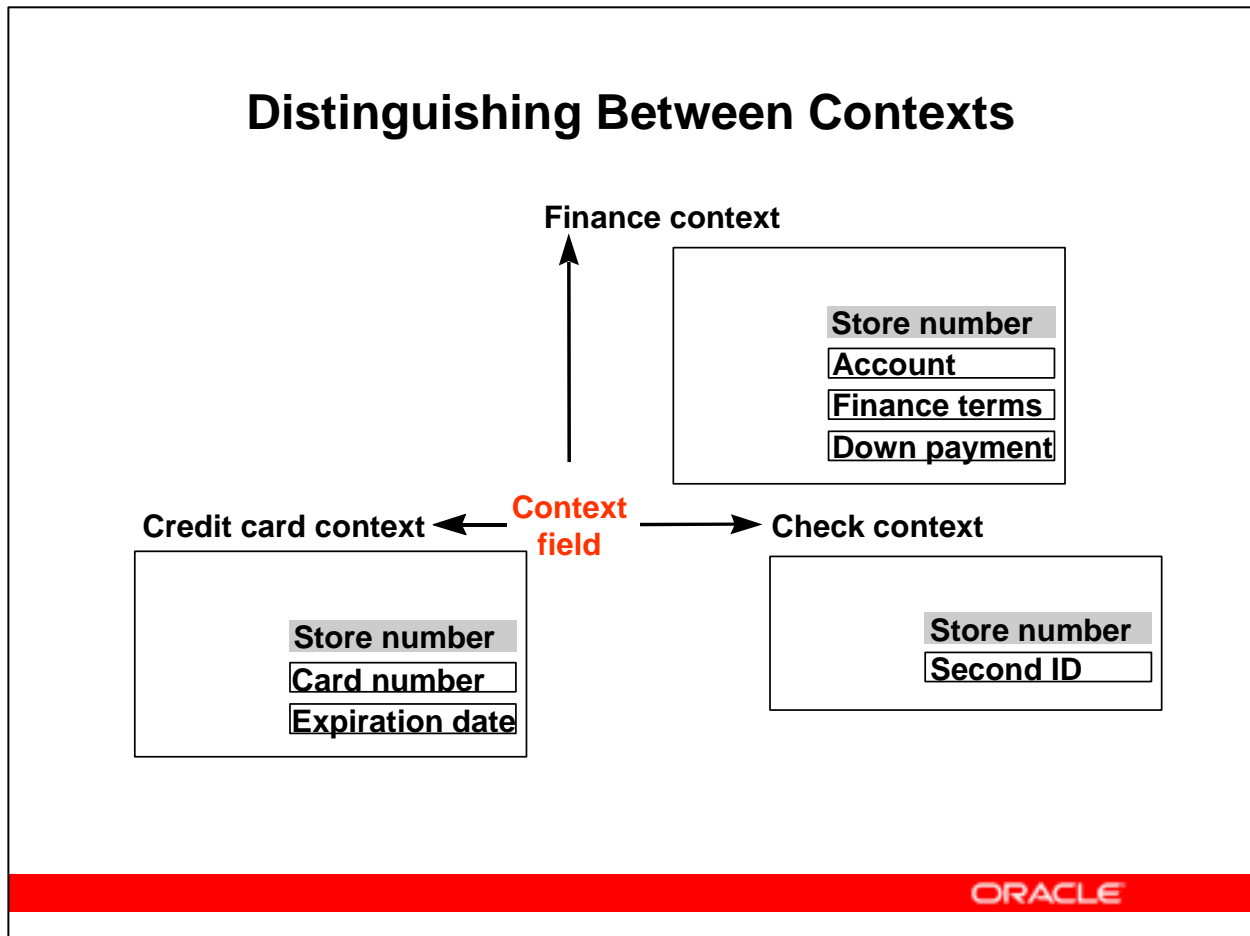
You can add context-sensitive segments later if columns are available, but enabled global segments always appear.



## Context-Sensitive Segments

Context sensitive segments occur depending on the context.

The slide shows sample contexts and the segments that are unique to each of them.



### Distinguishing Between Contexts

If your descriptive flexfield uses different contexts, you must decide how to distinguish between them. You must identify a field whose value can distinguish between contexts. This field is called the context field.

In some cases, you can use an existing field as the context field, in other cases, you must create a segment on the descriptive flexfield.

### Reference and Context Fields

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

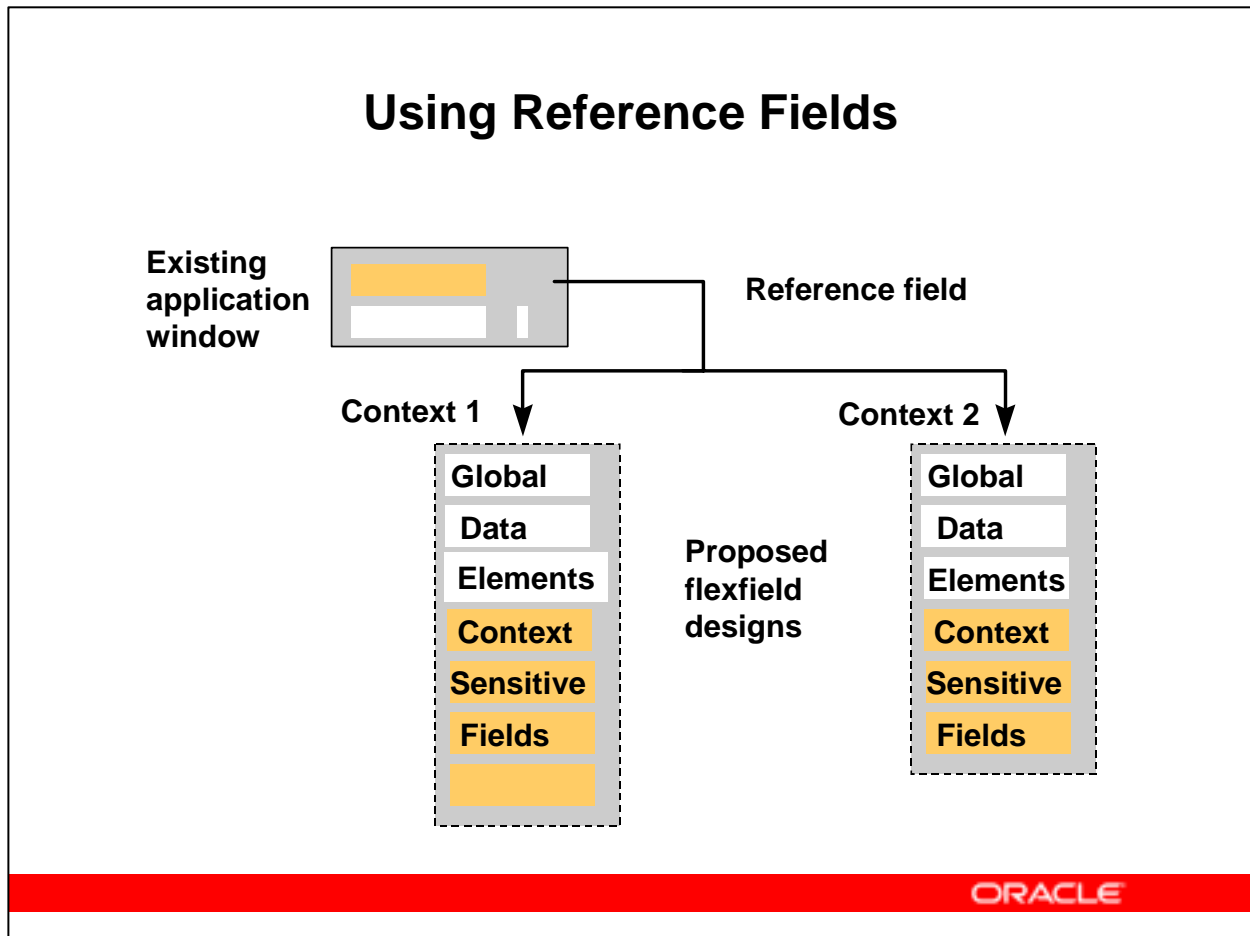
ORACLE

### Using Reference and Context Fields

There are two design options for distinguishing between contexts:

- If there is an existing field on the base window or an existing profile option whose value can be used to distinguish between contexts, it can be used as a reference field.
- If there is no existing field or profile option that can be used to automatically select the context, you may choose to allow users to manually select the context.

## Using Reference Fields



### Using Reference Fields

Reference fields are fields on the existing window whose values can determine which context a descriptive flexfield uses under the following conditions:

- The field must be defined to be referenceable. Not all fields on a window can be used as a reference field.
- The values appearing in the reference field should be known and predictable.
- Since the same descriptive flexfield can appear on different windows, any field used as a reference field for that descriptive flexfield must appear on the same windows. Also, the reference field must have the same internal name in all the forms where the flexfield is used.



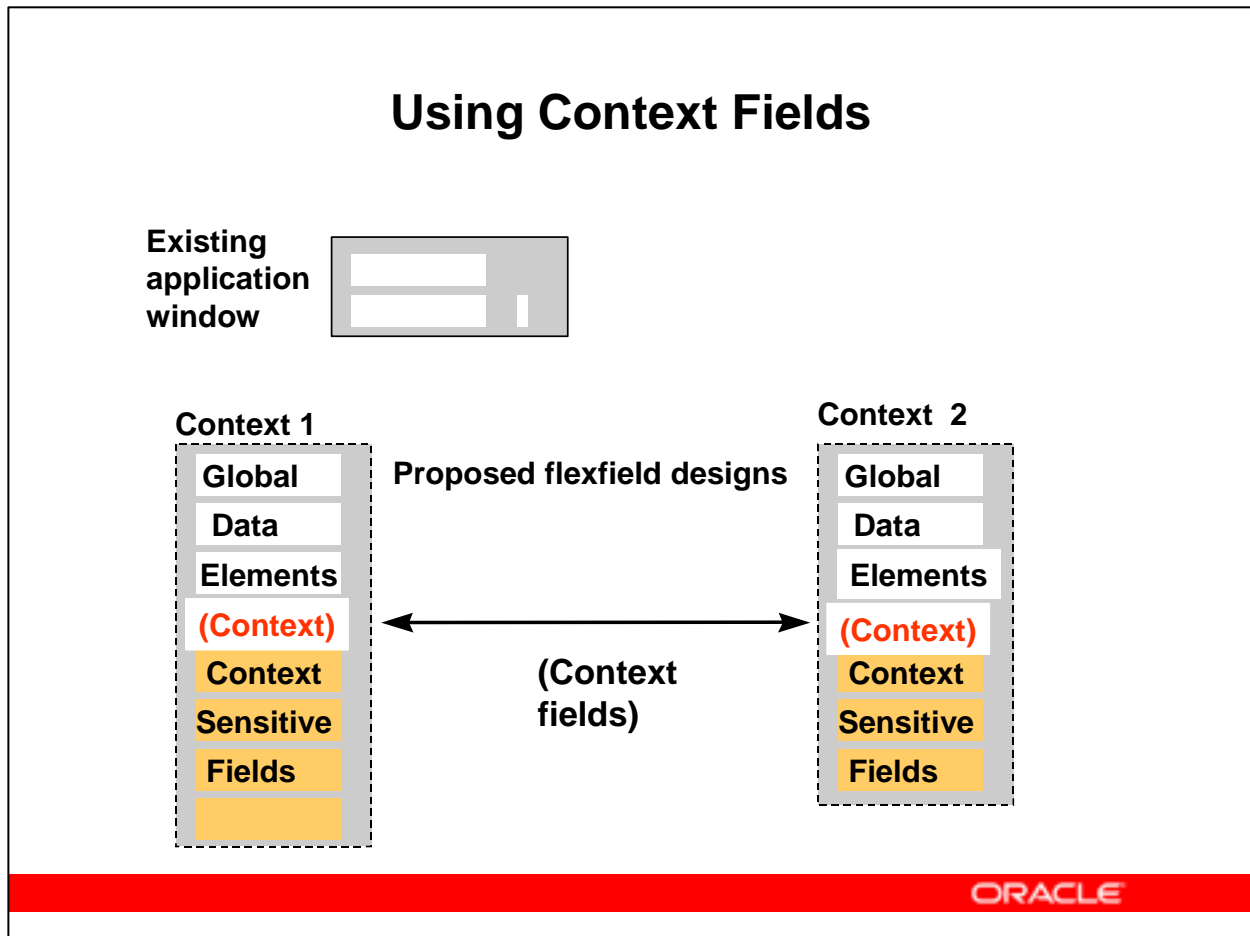
### Identifying Referenceable Columns

**Use the list of values for the Reference Field field in the Descriptive Flexfield Segments window to determine which fields are available to use as reference fields for this descriptive flexfield.**

**(N) Application—>Flexfield—>Descriptive—>Segments**

ORACLE

## Using Context Fields



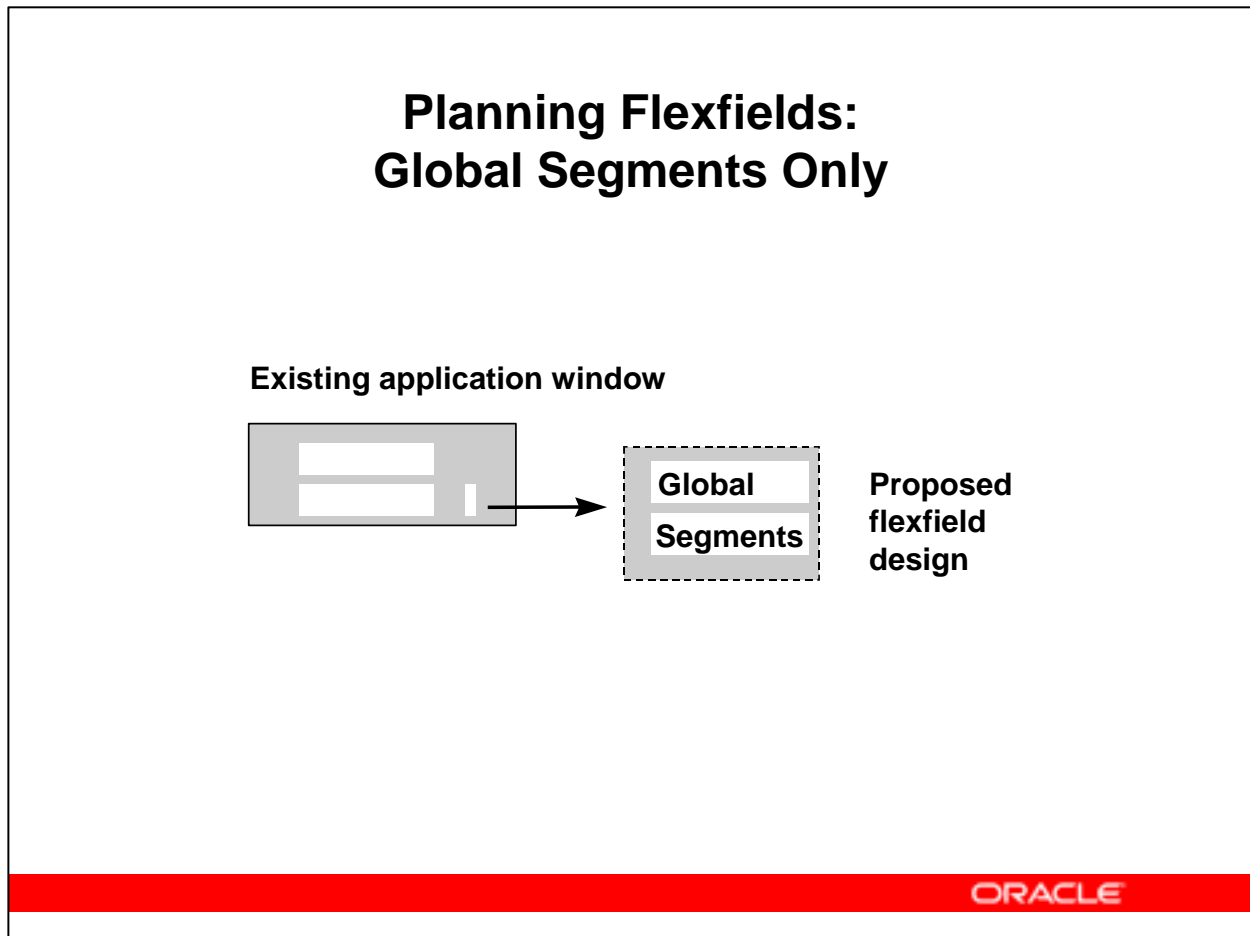
### Using a Context Field

Sometimes there is no field on the existing window that is appropriate for use as a reference field. In this case, you need to create a column on the descriptive flexfield itself to hold and display the different possible context values.

- A context field is an additional field appearing on the descriptive flexfield.
- The user can display the appropriate context by selecting a value from the pop-up list for the context field.

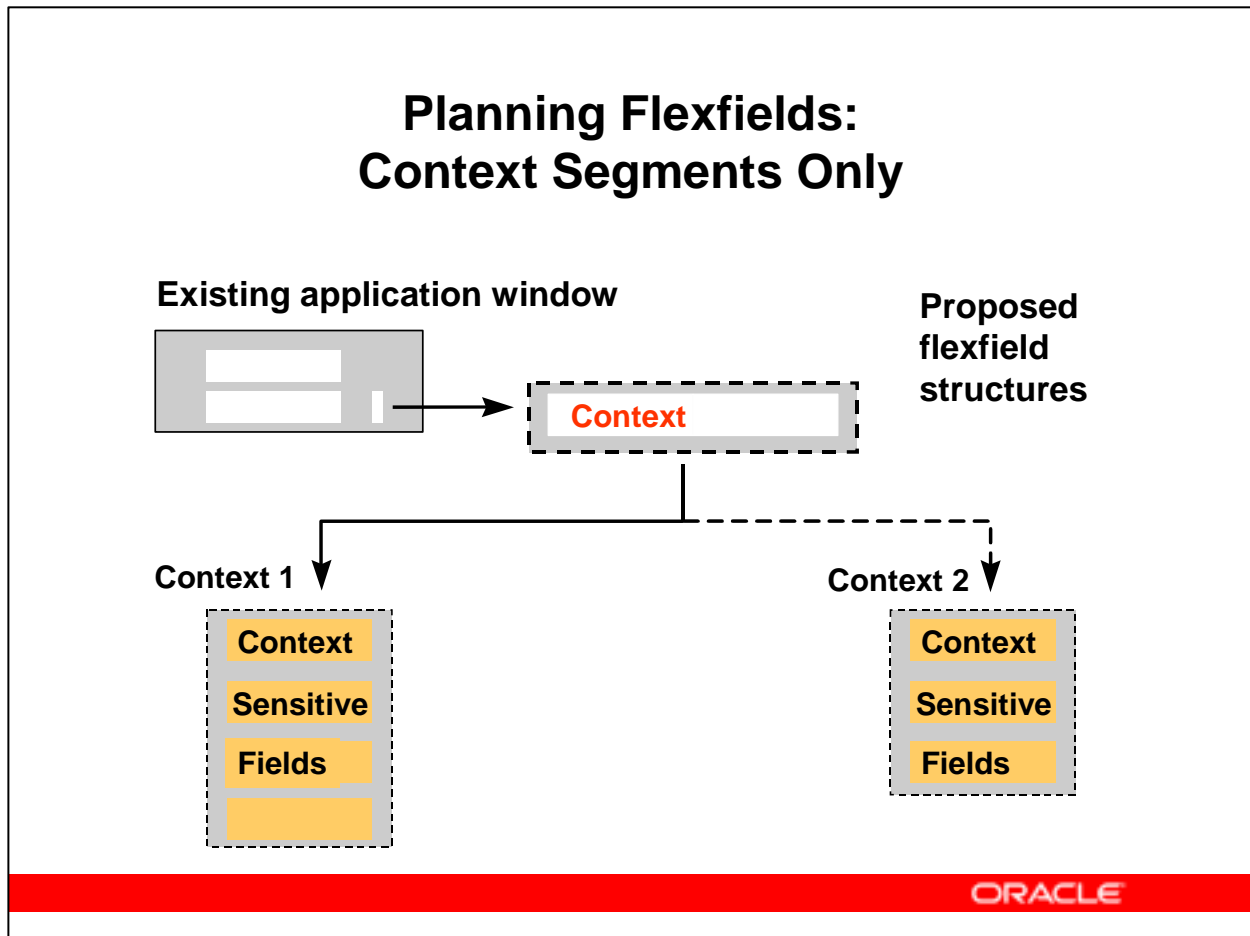
A context field is not a segment.

- The context field has a context field prompt.
- The response, called a context field value, determines which group of context-sensitive segments appears next.
- Each value for the context field can correspond to a separate context-sensitive structure.
- Context fields do not always display. Non-displayed context fields derive values from a default or from a reference field, and the user cannot change the context field value.



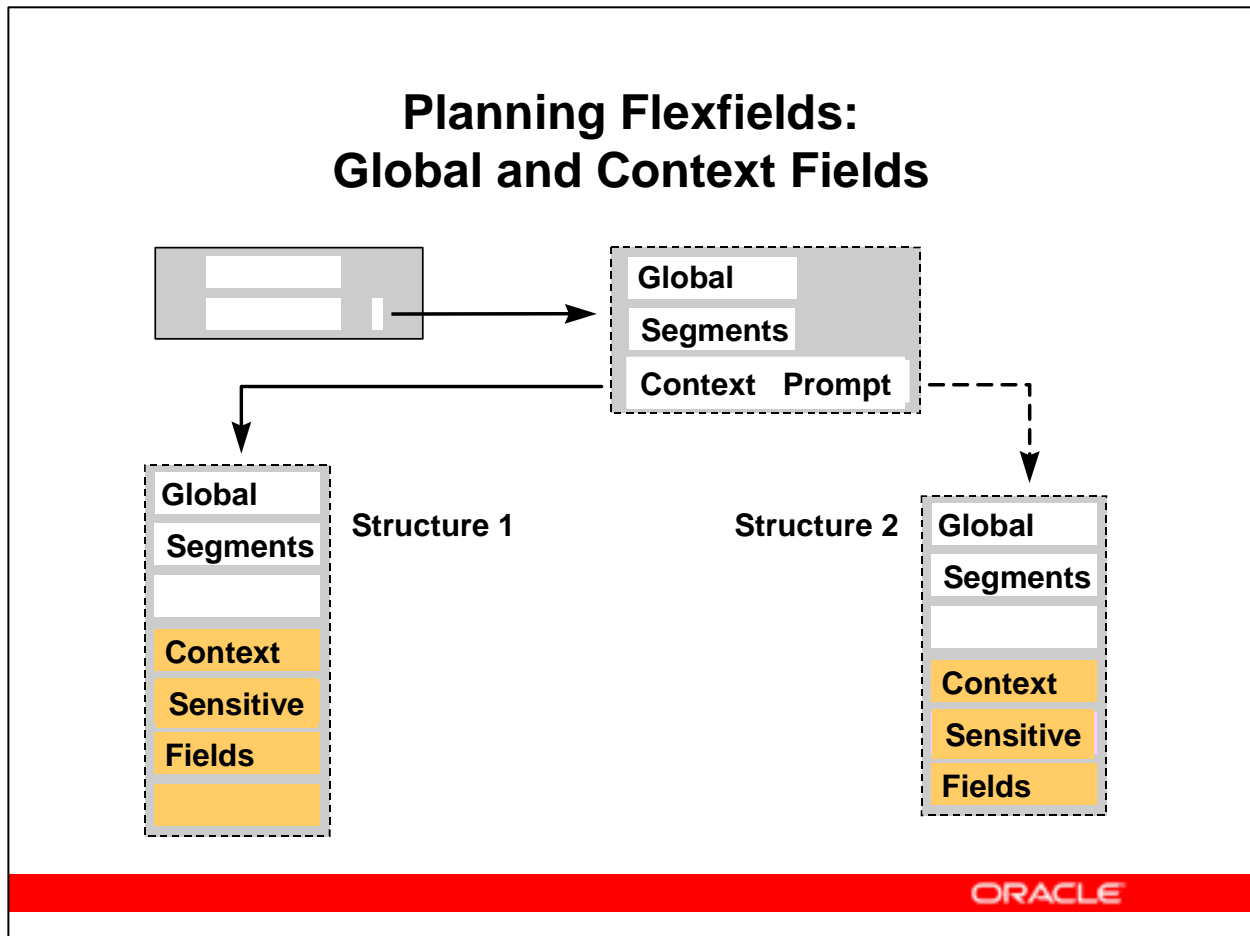
### Using Global Segments Only

Some descriptive flexfields use global segments only. The information gathered by that flexfield is valid for all users of the flexfield. All global segments display when the user enters the flexfield, and each segment prompts the user for one item of information.



## Using Context-Sensitive Segments Only

Some descriptive flexfields use only context-sensitive segments. They have no global segments. The flexfield contains only the context field prompt until a context is chosen. The context field value that is chosen determines which context is displayed in a second window.



### Using a Combination of Global and Context-Sensitive Segments

Most descriptive flexfields use a combination of global and context-sensitive segments. With a combination of global and context-sensitive segments, the processing is as follows:

- When the user opens the flexfield, all the global segments and a context prompt appear.
- Once the user chooses a context, the appropriate context-sensitive segments are added to the already visible global segments.

### Specifying the Flexfield Attributes

**In the Descriptive Flexfield Segments window, you can define:**

- **Title**
- **Segment Separator**
- **Context Field**
- **Context Field Values**

**(N) Application—>Flexfield—>Descriptive—>Segments**

ORACLE

### Specifying the Flexfield Attributes

Once you have the structure designed and the context behavior determined, you can plan the flexfield cosmetics. These include:

- The context prompt
- All the segment prompts
- The segment separator

### Specifying the Flexfield Structures

All of the segments contained on the flexfield will be defined to a structure. There is a default structure, Global Data Elements, to which you define all global segments. In addition to the Global Data Elements structure, you create a separate structure for each context. Use meaningful context names since they show up in the list of values for the contexts. Clear the enabled box if you ever need to mark this structure as unavailable. When you have specified your structure information, click the Segments button to define your structure segments.

### Specifying the Context Field Information

**In the Context Field region of the Descriptive Flexfield Segments window, you can define:**

- **Prompt**
- **Default Value**
- **Reference Field**
- **Value Required**
- **Override Allowed (Display Context)**

**(N) Application—>Flexfield—>Descriptive—>Segments**

ORACLE

#### Specify How the Context Is Determined

**Override Allowed (Display Context)** - Clear this check box if you want to restrict the user from choosing different contexts. If you use a reference field, you do not have to display the context field unless you want to allow the user to select a context different from the reference field value. If you do not display the context field and are not using a reference field, you should define a default context value as discussed below.

**Prompt** - This is the prompt context displayed on the flexfield. If you choose to display the context field, you must supply a value for this field.

**Default Value** - Specify a default context structure to use. Always define a value here if you are not using a reference field and not displaying the context field.

**Reference Field** - If you use a reference field, enter the name here in the format *Block.field*. Use the pop-up list on this field to display a list of the fields on the base window that are most likely to be referenced.

### Specifying the Segment Attributes

In the Segments Summary window, you can define:

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

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

ORACLE

### Specifying Segment Attributes

Specify the following attributes for each segment in the structure:

**Number** - The placement of the field on the window, from top to bottom.

**Name** - The internal name for this segment.

**Window Prompt** - The prompt to be displayed for this segment on the flexfield.

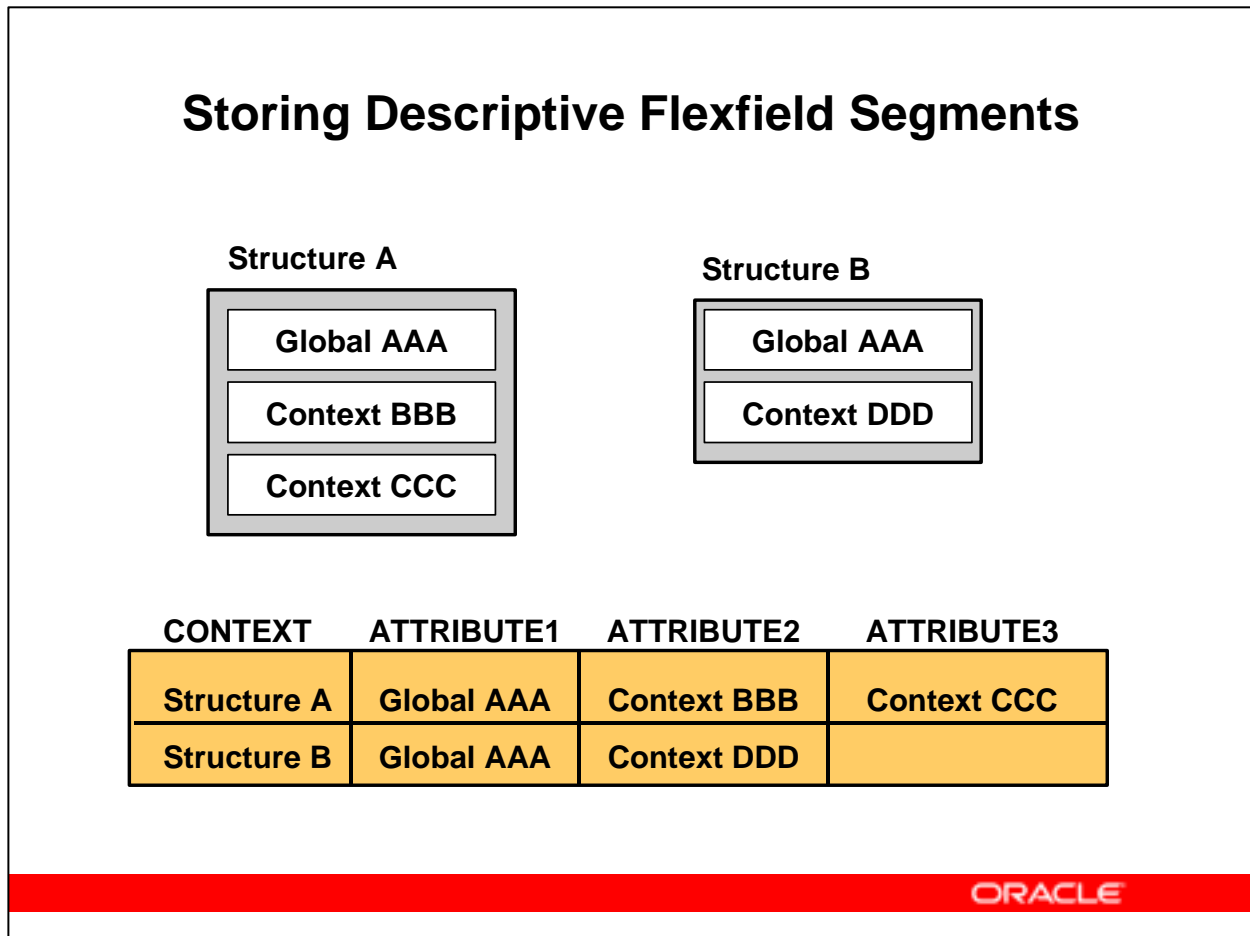
**Column** - The ATTRIBUTE column stores the data from this segment. Available columns are selectable from a pop-up list for this field. See the following page for an explanation of how ATTRIBUTE columns store descriptive flexfield data.

**Value Set** - If you wish to validate the data entered through this segment, specify a value set to perform validation checking.

**Displayed** - Clear this check box if you need to prevent the display of this field.

**Enabled** - Select this check box to mark this segment ready for processing.





### Storing Descriptive Flexfield Segments

As mentioned previously, the segments that make up a descriptive flexfield are stored in columns in the underlying tables. Each segment stores its data in one of the ATTRIBUTE columns. This does not mean, however, that every segment on the flexfield needs its own column. As shown on the slide, context-sensitive columns from different contexts can share the same column. The value in the CONTEXT column distinguishes between context segments.

### Descriptive Flexfield Application Tables

```
select fdftl.title,  
       fdf.application_table_name  
from   apps.fnd_descriptive_flexs fdf,  
       apps.fnd_descriptive_flexs_tl fdftl  
where  fdf.descriptive_flexfield_name =  
       fdftl.descriptive_flexfield_name  
and    fdf.application_id = '101'  
and    fdftl.title not like '$SRS$%'  
order by fdftl.title;
```

ORACLE

### Descriptive Flexfield Application Tables

This SQL\*Plus query can be used to determine the application base table where a descriptive flexfield stores its data. This is a partial report for one application (Oracle General Ledger - application ID 101) and does not show any information for flexfields used with SRS.

TITLE	APPLICATION_TABLE_NAME
Accounting Calendar: Calendar	GL_PERIOD_SETS
Accounting Calendar: Periods	GL_PERIODS
AutoAccounting Rules	GL_IEA_AUTOGEN_MAP
AutoAllocation Batch	GL_AUTO_ALLOC_BATCHES
AutoPost Criteria	GL_AUTOMATIC_POSTING_OPTIONS
AutoReversal Criteria	GL_AUTOREVERSE_OPTIONS
Automatic Posting Sets	GL_AUTOMATIC_POSTING_SETS
Budget Types	GL_BUDGET_TYPES
Budget Versions	GL_BUDGET_VERSIONS
Budgetary Control Group: Group	GL_BC_OPTIONS
Budgetary Control Group: Rules	GL_BC_OPTION_DETAILS
Common Stocks: Share Activity	GL_SHARES_ACTIVITY
Conversion Rate Types	GL_DAILY_CONVERSION_TYPES

### Specifying the Segment Detail Attributes

In the Validation region of the Segments window, you can define:

- Value Set
- Description
- Default Type
- Default Value
- Required
- Security Enabled

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

ORACLE

### Specifying the Segment Detail Attributes

You can use this window to specify any additional attributes for the segment.

#### Validation Block Entries

**Value Set** - The name of the value set used to validate data entered through this segment.

**Description** - The description of this value set.

**Default Type** - If you wish to define a default value for this segment, select the data type here.

**Default Value** - Define the actual default here.

**Required** - Select this checkbox if a value for this segment is mandatory.

**Security Enabled** - Select this checkbox to turn on any security rules for this value set.

### Specifying the Segment Detail Attributes

In the **Sizes** region of the **Segments** window, you can define:

- **Display Size**
- **Description Size**
- **Concatenated Description Size**

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

ORACLE

### Specifying the Segment Detail Attributes (continued)

#### Sizes

These sizes determine the width of the field on the flexfield that is used to display this segment. Especially consider the interaction between Display Size and the Maximum Size value defined to the value set used with this segment.

### Specifying the Segment Detail Attributes

**In the Prompts region of the Segments window, you can define:**

- **List of Values**
- **Window**

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

(B) Segments

ORACLE

### Specifying the Segment Detail Attributes (continued)

#### Prompts

**List of Values** - By default, this will be the same as the segment name. It should not be larger than the Display Size.

**Window** - Specify the label that will appear to the left of this segment on the flexfield.

## Specifying Default Values

---

### Specifying Default Values

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

ORACLE

#### Specifying Segment Defaults - Examples

**Default Type:** Constant

**Default Value:** The constant specified.

**Example:** USA

**Result:** USA

**Default Type:** Current Date

**Default Value:** The date at the time of entry.

**Example:**

**Result:** MAY 01, 2000

**Default Type:** Current Time

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

**Example:**

**Result:** 14:30:00 MAY 01, 2000

**Default Type:** Field

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

**Example:** ORDER:LINE

**Result:** 3

**Default Type:** Profile

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

**Example:** GL\_SET\_OF\_BOOKS\_ID

**Result:** 101

**Default Type:** Segment

**Default Value:** The value returned by the specified previous segment.

**Example:** Company

**Result:** 01

**Default Type:** SQL Statement

**Default Value:** The value returned by the specified SQL statement. The statement must return a single value. \$PROFILES\$ and \$FLEX\$ can be used in the statement.

**Example:** SELECT NAME FROM EMP WHERE JOB=CEO

**Result:** Jones

### **Defining a Descriptive Flexfield**

- 1. Define any value sets needed.**
- 2. Create the flexfield and specify context information.**
- 3. Define different structures.**
- 4. Specify segment cosmetics and behavior.**
- 5. Freeze and compile the flexfield definition.**
- 6. Define values for value sets if needed.**

**ORACLE**



### Defining Value Sets

- **Define value sets for each segment of the planned descriptive flexfield, including both global and context-sensitive segments.**
- **Use existing value sets when possible.**
- **Use the Value Sets window to create new value sets if necessary. These value sets can be used by other flexfields or report parameters as appropriate.**

**(N) Application—>Validation—>Sets**

ORACLE

### Locating the Flexfield Definition

- Use the **Descriptive Flexfield Segments** window to locate the target flexfield definition by finding the application that owns the definition and the flexfield title.
- A flexfield may appear on more than one window. However, defining the flexfield once automatically defines it for all locations.
- After you access the definition, you must unfreeze it to begin making changes.

**(N) Application—>Flexfield—>Descriptive—>Segments**

ORACLE

### Defining the Flexfield Header Attributes

Use the **Descriptive Flexfield Segments** window to enter:

- **Application**
- **Title**
- **Freeze Flexfield Definition**
- **Segment Separator**
- **Global Data Elements**

(N) **Application**—>**Flexfield**—>**Descriptive**—>**Segments**

ORACLE

### Defining the Header Attributes

The header attributes you specify include:

**Application** - The application that owns the flexfield.

**Title** - The title displayed at the top of the flexfield.

**Freeze Flexfield Definition** - Select this box when you have finished defining your flexfield. Freezing the definition enables the Compile button so you can compile your definition.

**Segment Separator** - Specify a separator to be used when displaying concatenated segment values.

**Global Data Elements** - If your structure has no context-sensitive structures, select Global Data Elements and click the Segments button to define the global segments.

Define the segments that always appear regardless of the context value. Define global segments first to facilitate efficient column use.

### Defining Segment Attributes

Use the Segments Summary window to enter:

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

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

ORACLE

### Defining Segment Attributes

**Number** - Specify the sequence number (from top to bottom) of this segment within the flexfield structure. The segments defined for Global Data Elements always appear first. They are followed by the segments defined for whatever other context-sensitive structure is selected.

**Name** - Enter the name for this segment.

**Window Prompt** - Enter a prompt to show up on the flexfield window. The default is to use the Name.

**Column** - Select the ATTRIBUTE column in the base table to hold this segment's data. A pop-up list for this field displays the columns available to you for definition.

**Value Set** - Specify an existing value set to validate the data for this segment.

Click the Open button to define additional attributes for your segment.

### Specifying Validation and Field Sizes

Use the Segments window to enter options for:

- Validation
- Sizes
- Prompts

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

ORACLE

### Specifying Validation and Field Sizes

Use this window to specify additional attributes for the segment.

If you are using a value set to validate the input in this segment, enter the name of the value set. You can also enter a default for the segment and enable security.

Use the Sizes fields to control the display of the segment in the flexfield window.

Specify the Display Size of the flexfield segment to be the same as the Maximum Size of the value set used by this segment to avoid scrolling.

**Index Flag (Accounting Flexfield Only)** - This is used with the General Ledger Optimizer feature.

### Freezing and Compiling the Definition

**Use the Descriptive Flexfield Segments window to:**

- **Freeze the flexfield definition by checking the Freeze Flexfield Definition check box**
- **Compile the flexfield definition by clicking the Compile button**

**(N) Application—>Flexfield—>Descriptive—>Segments**

ORACLE

### Freezing and Compiling the Definition

Freeze the flexfield information to notify the application to begin using the flexfield.

Compiling the flexfield stores the information efficiently. If the compile detects any problems, a message window displays a warning.

Flexfields automatically compile the flexfield definition at every commit on this form. The request for view generation automatically follows compilation.

You see your own changes immediately. Other users must exit or change responsibilities to see the new definitions take effect.

### Defining Context Field Information

**Use the Descriptive Flexfield Segments window to enter:**

- **Prompt**
- **Default Value**
- **Reference Field**
- **Value Required**
- **Override Allowed (Display Context)**
- **Context Field Values**

**(N) Application—>Flexfield—>Descriptive—>Segments**

ORACLE

### Defining Context Information

Define context information to determine how context-sensitive structures are chosen.

- If not using a reference field to obtain the context value, enter a context field prompt. This is the prompt that instructs the user to select a context.
- Optionally enter a default value. This should be the name of the most common context-sensitive structure. If you use a reference field, ensure that the default value can be obtained from the reference field.
- Specify whether the context field is required. Without a required response in the context field, users can avoid entering any information in any of the context-sensitive structures. If you are requiring a context value and not displaying a context prompt, either provide a default or ensure that a value can be obtained from a reference field value.
- If you are using a reference field, specify the field name in the Reference Field field. Not all fields on the window can be used as a reference field. Use the list of values for this field to set the possible reference fields for the window.
- Enter the context values in the Context Field Values region. The names of these structures must exactly match the values you expect to see in the context field.

### **Defining Context-Sensitive Segments**

**Use the Segments Summary, Segments, and Descriptive Flexfield Segments windows to define context-sensitive segments using the same procedure as for global segments:**

- **Specify all necessary segment attribute information.**
- **Enable the segment.**
- **Define any validation information.**
- **Modify the display size if necessary.**
- **Freeze and compile the definition.**

**(N) Application—>Flexfield—>Descriptive—>Segments**  
**(B) Segments (B) Open**

**ORACLE**



### Defining Values for a Value Set

**After you finish defining the descriptive flexfield structure, use the Segment Values window to define values for any independent or dependent value sets assigned to any segments of the descriptive flexfield.**

**(N) Application—>Validation—>Values**

ORACLE

## Practice - Define a DFF

---

### Overview

To perform the practices in this lesson (and the next 3 as well), you will need a descriptive flexfield to work on. Your instructor will assign you a descriptive flexfield that has not yet been used in the database you are accessing. You will use the same flexfield for practices throughout these four lessons. The list below assigns each team number a unique descriptive flexfield and shows the navigation path to the flexfield. The paths start from the General Ledger Super User responsibility. Your instructor will tell you which flexfield you are to design.

#### Team: 01

**Descriptive Flexfield Title:** Accounting Calendar: Periods

**Navigation Path:** Setup > Financials > Calendars > Accounting (Periods region)

#### Team: 02

**Descriptive Flexfield Title:** AutoPost Criteria

**Navigation Path:** Setup > Journal > AutoPost

#### Team: 03

**Descriptive Flexfield Title:** Budgetary Control Group: Rules

**Navigation Path:** Budgets > Define > Controls (Budgetary Control Rules region)

#### Team: 04

**Descriptive Flexfield Title:** Conversion Rate Types

**Navigation Path:** Setup > Currencies > Rates > Types

#### Team: 05

**Descriptive Flexfield Title:** Define Budget Organization: Organization

**Navigation Path:** Budgets > Define > Organization

#### Team: 06

**Descriptive Flexfield Title:** Define Recurring Journal: Batch

**Navigation Path:** Journals > Define > Recurring

**Team: 07**

**Descriptive Flexfield Title:** Journal Categories

**Navigation Path:** Setup > Journal > Categories

**Team: 08**

**Descriptive Flexfield Title:** Journal Sources

**Navigation Path:** Setup > Journal > Sources

**Team: 09**

**Descriptive Flexfield Title:** Open and Close Periods

**Navigation Path:** Setup > Open/Close

**Team: 10**

**Descriptive Flexfield Title:** Period Rates

**Navigation Path:** Setup > Currencies > Rates > Period

**Team: 11**

**Descriptive Flexfield Title:** Period Types

**Navigation Path:** Setup > Financials > Calendars > Types

**Team: 12**

**Descriptive Flexfield Title:** Summary Accounts

**Navigation Path:** Setup > Accounts > Summary

**Team: 13**

**Descriptive Flexfield Title:** Suspense Accounts

**Navigation Path:** Setup > Accounts > Suspense

The following two descriptive flexfields can be accessed through the General Ledger Super User responsibility but are owned by Application Object Library.

**Team: 14**

**Descriptive Flexfield Title:** Define Sequences

**Navigation Path:** Setup > Financials > Sequences > Define

## Team: 15

**Descriptive Flexfield Title:** Currencies

**Navigation Path:** Setup > Currencies > Define

### Scenario

Descriptive flexfields use two types of segments: global and context-sensitive. In this practice, you will create a descriptive flexfield using only global segments. You will also create value sets that provide lists of values for your user. The descriptive flexfield is designed to track the following additional information about orders that have been held:

- Who held the order
- The reason the order was held
- The amount of time the order was held

### Tasks

#### Define your Value Sets

1. Use the Value Sets window to define three value sets.
  - Define an independent value set named *YourInitials\_WHO\_HELD*. Give the value set a description, a format type of Char, and a maximum size of 2. Enable security for the value set.
  - Define an independent value set named *YourInitials\_WHY\_HELD*. Give the value set a description, a format type of Char, and a maximum size of 3. Enable security for the value set.
  - Define an independent value set named *YourInitials\_HOW\_LONG\_HELD*. Give the value set a description, a format type of Char, and a maximum size of 3. Enable security for the value set.

#### Define your Structure

2. After defining your value sets, use the Descriptive Flexfield Segments window to define the structure for the descriptive flexfield assigned to you. Do not allow overrides for the context. Define three segments for the Global Data Elements context.
  - Define a segment named Who Held, with a prompt of Who. Assign the segment the number 10, the column ATTRIBUTE1, and the value set *YourInitials\_WHO\_HELD*.
  - Define a segment named Why Held, with a prompt of Why. Assign the segment the number 20, the column ATTRIBUTE2, and the value set *YourInitials\_WHY\_HELD*.

- Define a segment named How Long, with a prompt of Length. Assign the segment the number 30, the column ATTRIBUTE3, and the value set *YourInitials\_HOW\_LONG\_HELD*.
- Ensure that all three segments are displayed and enabled. When you finish defining the structure, freeze and compile your flexfield definition.

## Define your Values

3. After defining the value sets and segments for your flexfield, use the Segment Values window to define the values associated with each of the independent value sets.
4. Define and enable the following values for the *YourInitials\_WHO\_HELD* value set.
  - **Value:** 01
  - **Description:** Helen Myers
  - **Value:** 02
  - **Description:** Mien Chan
  - **Value:** 03
  - **Description:** Michael Keller
  - **Value:** 04
  - **Description:** Luis Galvez
5. Define and enable the following values for the *YourInitials\_WHY\_HELD* value set.
  - **Value:** FIR
  - **Description:** Further information required
  - **Value:** MAR
  - **Description:** Manager authorization required
  - **Value:** AJR
  - **Description:** Additional justification required
6. Define and enable the following values for the *YourInitials\_HOW\_LONG\_HELD* value set.
  - **Value:** 100
  - **Description:** Less than one week

- **Value:** 200
- **Description:** One week to one month
- **Value:** 300
- **Description:** One month to one year
- **Value:** 400
- **Description:** More than one year

### **Test your Descriptive Flexfield**

7. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

## Solution – Define a DFF

### Define your Value Sets

**Responsibility = System Administrator or General Ledger Super User**

1. Navigate to (N) Application > Validation > Set.
2. Enter the information for the first value set in the following fields:
  - **Value Set Name:** *YourInitials\_WHO\_HELD*
  - **Description:** *YourInitials Who Held Value Set*
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 2
  - **Validation Type:** Independent
3. Save your work.
4. Enter the information for the second value set in the following fields:
  - **Value Set Name:** *YourInitials\_WHY\_HELD*
  - **Description:** *YourInitials Why Held Value Set*
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 3
  - **Validation Type:** Independent
5. Save your work.
6. Enter the information for the third value set in the following fields:
  - **Value Set Name:** *YourInitials\_HOW\_LONG\_HELD*
  - **Description:** *YourInitials How Long Held Value Set*
  - **Security Available:** Selected

- **Format Type:** Char
- **Maximum Size:** 3
- **Validation Type:** Independent

7. Save your work.

## Define your Structure

1. Navigate to (N) Application > Flexfield > Descriptive > Segments.
2. Query your descriptive flexfield.
3. In the Context Field region, ensure that the Override Allowed (Display Context) check box is cleared.
4. In the Context Field Values region, select the Global Data Elements context and click the “Segments” button to navigate to the Segments Summary window.
5. Enter the information for the first segment in the following fields:
  - **Number:** 10
  - **Name:** Who Held
  - **Window Prompt:** Who
  - **Column:** ATTRIBUTE1
  - **Value Set:** *YourInitials\_WHO\_HELD*
  - **Displayed:** Selected
  - **Enabled:** Selected
6. Save your work.
7. Enter the information for the second segment in the following fields:
  - **Number:** 20
  - **Name:** Why Held
  - **Window Prompt:** Why
  - **Column:** ATTRIBUTE2
  - **Value Set:** *YourInitials\_WHY\_HELD*



- **Displayed:** Selected
  - **Enabled:** Selected
8. Save your work.
  9. Enter the information for the third segment in the following fields:
    - **Number:** 30
    - **Name:** How Long
    - **Window Prompt:** Length
    - **Column:** ATTRIBUTE3
    - **Value Set:** *YourInitials\_HOW\_LONG\_HELD*
    - **Displayed:** Selected
    - **Enabled:** Selected
  10. Save your work.
  11. Navigate back to the Descriptive Flexfield Segments window.
  12. Select the Freeze Flexfield Definition check box.
  13. Click the “Compile” button to compile the flexfield definition.

### **Define your Values**

1. Navigate to (N) Application > Validation > Values.
2. In the Find window, select Value Set and find the *YourInitials\_WHO\_HELD* value set.
3. In the Values, Effective region of the Segment Values window, define the following values:
  - **Value:** 01
  - **Description:** Helen Myers
  - **Enabled:** Selected
  - **Value:** 02
  - **Description:** Mien Chan
  - **Enabled:** Selected

- **Value:** 03
  - **Description:** Michael Keller
  - **Enabled:** Selected
  - **Value:** 04
  - **Description:** Luis Galvez
  - **Enabled:** Selected
4. Save your work.
  5. Select Value Set and find the *YourInitials\_WHY\_HELD* value set.
  6. In the Values, Effective region, define the following values:
    - **Value:** FIR
    - **Description:** Further information required
    - **Enabled:** Selected
    - **Value:** MAR
    - **Description:** Manager authorization required
    - **Enabled:** Selected
    - **Value:** AJR
    - **Description:** Additional justification required
    - **Enabled:** Selected
  7. Save your work.
  8. Select Value Set and find the *YourInitials\_HOW\_LONG\_HELD* value set.
  9. In the Values, Effective region, define the following values:
    - **Value:** 100
    - **Description:** Less than one week
    - **Enabled:** Selected
    - **Value:** 200

- **Description:** One week to one month
- **Enabled:** Selected
- **Value:** 300
- **Description:** One month to one year
- **Enabled:** Selected
- **Value:** 400
- **Description:** More than one year
- **Enabled:** Selected

10. Save your work.

### **Test your Descriptive Flexfield**

1. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

# Practice - Define a DFF with None Validation

---

## Overview

To perform the practices in this lesson (and the next 2 as well), you will need a descriptive flexfield to work on. In the previous lesson, you were assigned a descriptive flexfield by your instructor. Please use that descriptive flexfield for this exercise. If you were not assigned a descriptive flexfield, please reference the previous lesson, and have your instructor assign one now.

This practice reinforces the concept of global segments for a descriptive flexfield and demonstrates the use of the None validation type. In this scenario, you will define a descriptive flexfield to track the following information about employees:

- The person who referred the employee
- Height
- Weight

## Tasks

### Define your Value Sets

1. Use the Value Sets window to define three value sets as follows:
  - Define an independent value set named *YourInitials\_WHO\_REFER*. Give the value set a description, a format type of Char, and a maximum size of 15. Enable security for the value set.
  - Define a value set of validation type None named *YourInitials\_EMP\_HEIGHT*. Give the value set a description, a format type of Number, a maximum size of 4, and a precision of 2.
  - Define a value set of validation type None named *YourInitials\_EMP\_WEIGHT*. Give the value set a description, a format type of Number, a maximum size of 3, a precision of 0, a minimum value of 90, and a maximum value of 999.

### Define your Structure

2. After defining your value sets, use the Descriptive Flexfield Segments window to define the structure for your descriptive flexfield. For the purposes of this practice, use the same descriptive flexfield that you worked on previously to enter your new structure. Unfreeze the flexfield definition first so that you can change the structure.
3. Delete the segments that you defined previously, and define three new segments for the Global Data Elements context.

- Define a segment named Who Referred, with a prompt of Who. Assign the segment the number 10, the column ATTRIBUTE1, and the value set *YourInitials\_WHO\_REFER*.
- Define a segment named Height, with a prompt of Height. Assign the segment the number 20, the column ATTRIBUTE2, and the value set *YourInitials\_EMP\_HEIGHT*.
- Define a segment named Weight, with a prompt of Weight. Assign the segment the number 30, the column ATTRIBUTE3, and the value set *YourInitials\_EMP\_WEIGHT*.
- Ensure that all three segments are displayed and enabled. When you finish defining the structure, freeze and compile your flexfield definition.

### Define your Values

3. After defining the value sets and segments for your flexfield, use the Segment Values window to define the values associated with the independent value set.
4. Define and enable the following values for the *YourInitials\_WHO\_REFER* value set.
  - **Value:** 100
  - **Description:** President
  - **Value:** 200
  - **Description:** Executive Vice President
  - **Value:** 300
  - **Description:** Senior Vice President

### Test your Descriptive Flexfield

5. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

## Solution – Define a DFF with None Validation

### Define your Value Sets

**Responsibility = System Administrator or General Ledger Super User**

1. Navigate to (N) Application > Validation > Set.
2. Enter the information for the first value set in the following fields:
  - **Value Set Name:** *YourInitials\_WHO\_REFER*
  - **Description:** *YourInitials Who Refer Value Set*
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 15
  - **Validation Type:** Independent
3. Save your work.
4. Enter the information for the second value set in the following fields:
  - **Value Set Name:** *YourInitials\_EMP\_HEIGHT*
  - **Description:** *YourInitials Employee Height Value Set*
  - **Format Type:** Number
  - **Maximum Size:** 4
  - **Precision:** 2
  - **Validation Type:** None

**Note:** You cannot enable security for a value set of validation type None.
5. Save your work.
6. Enter the information for the third value set in the following fields:
  - **Value Set Name:** *YourInitials\_EMP\_WEIGHT*
  - **Description:** *YourInitials Employee Weight Value Set*

- **Format Type:** Number
- **Maximum Size:** 3
- **Precision:** 0
- **Min Value:** 90
- **Max Value:** 999
- **Validation Type:** None

7. Save your work.

### **Define your Structure**

1. Navigate to (N) Application > Flexfield > Descriptive > Segments.
2. Query your descriptive flexfield.
3. Clear the Freeze Flexfield Definition check box.
4. In the Context Field Values region, select the Global Data Elements context and click the “Segments” button to navigate to the Segments Summary window.
5. Delete the segments that you defined previously.
6. Save your work.
7. Enter the information for the first segment in the following fields:
  - **Number:** 10
  - **Name:** Who Referred
  - **Window Prompt:** Who
  - **Column:** ATTRIBUTE1
  - **Value Set:** *YourInitials\_WHO\_REFER*
  - **Displayed:** Selected
  - **Enabled:** Selected

8. Save your work.

9. Enter the information for the second segment in the following fields:

- **Number:** 20
- **Name:** Height
- **Window Prompt:** Height
- **Column:** ATTRIBUTE2
- **Value Set:** *YourInitials\_EMP\_HEIGHT*
- **Displayed:** Selected
- **Enabled:** Selected

10. Save your work.

11. Enter the information for the third segment in the following fields:

- **Number:** 30
- **Name:** Weight
- **Window Prompt:** Weight
- **Column:** ATTRIBUTE3
- **Value Set:** *YourInitials\_EMP\_WEIGHT*
- **Displayed:** Selected
- **Enabled:** Selected

12. Save your work.

13. Navigate back to the Descriptive Flexfield Segments window.

14. Select the Freeze Flexfield Definition check box.

15. Click the “Compile” button to compile the flexfield definition.

### **Define your Values**

1. Navigate to (N) Application > Validation > Values.
2. In the Find window, select Value Set and find the *YourInitials\_WHO\_REFER* value set.
3. In the Values, Effective region of the Segment Values window, define the following values:
  - **Value:** 100



- **Description:** President
- **Enabled:** Selected
- **Value:** 200
- **Description:** Executive Vice President
- **Enabled:** Selected
- **Value:** 300
- **Description:** Senior Vice President
- **Enabled:** Selected

4. Save your work.

### **Test your Descriptive Flexfield**

1. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

# Practice - Define a DFF with Context-Sensitive Segment

---

## Overview

To perform the practices in this lesson (and the next 1 as well), you will need a descriptive flexfield to work on. In the previous lesson, you were assigned a descriptive flexfield by your instructor. Please use that descriptive flexfield for this exercise. If you were not assigned a descriptive flexfield, please reference the previous lesson, and have your instructor assign one now.

In this practice, you will set up a context-sensitive segment and create a table-validated value set. By using context sensitivity, you increase the number of questions you can ask without increasing the number of columns in the table. In this scenario, you will track additional information for a transaction based on whether or not the transaction is a project. If it is not a project, then you do not need to capture additional information. If it is a project, then the flexfield should track the salesperson, the salesperson's region, and the project name. To avoid some repetitive data entry, you will use a table that already exists within Oracle Applications to validate the project name.

## Tasks

### Define your Value Sets

1. Use the Value Sets window to define three value sets.
  - Define an independent value set named *YourInitials\_SALES\_REP*. Give the value set a description, a format type of Char, and a maximum size of 3. Specify that the values must be uppercase only, and enable security for the value set.
  - Define an independent value set named *YourInitials\_LOCATION*. Give the value set a description, a format type of Char, and a maximum size of 3. Specify that the values must be uppercase only, and enable security for the value set.
  - Define a table-validated value set named *YourInitials\_PROJECT*. Give the value set a description, a format type of Char, and a maximum size of 30. Enable security for the value set. Specify AR\_CUSTOMERS\_V in the Oracle Receivables application as the validation table. Assign the Value column the column name CUSTOMER\_NUMBER, a type of Char, and a size of 30. Assign the Meaning column the column name CUSTOMER\_NAME, a type of Char, and a size of 50.

### Define your Structure

2. After defining your value sets, use the Descriptive Flexfield Segments window to define the structure for your descriptive flexfield. For the purposes of this practice, use the same descriptive flexfield that you worked on previously to enter your new structure. Unfreeze the flexfield definition first so that you can change the structure.

3. For the context field, enter the prompt “Is this a project?” Specify that a value is required and context override is allowed.
4. Delete the segments that you defined previously for the Global Data Elements context.
5. Define and enable a new context named Yes with the code Yes. Define three new segments for the Yes context.
  - Define a segment named Sales Rep, with a prompt of Sales Rep. Assign the segment the number 10, the column ATTRIBUTE1, and the value set *YourInitials\_SALES\_REP*.
  - Define a segment named Location, with a prompt of Location. Assign the segment the number 20, the column ATTRIBUTE2, and the value set *YourInitials\_LOCATION*.
  - Define a segment named Project, with a prompt of Project. Assign the segment the number 30, the column ATTRIBUTE3, and the value set *YourInitials\_PROJECT*.
  - Ensure that all three segments are displayed and enabled.
6. Define and enable another new context named No with the code No.
7. When you finish defining the structure, freeze and compile your flexfield definition.

### **Define your Values**

8. After defining the value sets and segments for your flexfield, use the Segment Values window to define the values associated with each of the independent value sets.
9. Define and enable the following values for the *YourInitials\_SALES\_REP* value set.
  - **Value:** CJ
  - **Description:** Charles Jefferson
  - **Value:** AR
  - **Description:** Ana Rodriguez
  - **Value:** PL
  - **Description:** Philippe Lebeau
10. Define and enable the following values for the *YourInitials\_LOCATION* value set.
  - **Value:** NY
  - **Description:** New York
  - **Value:** MA

- **Description:** Madrid
- **Value:** PA
- **Description:** Paris

### **Test your Descriptive Flexfield**

11. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

## Solution – Define a DFF with a Context-Sensitive Segment

### Define your Value Sets

**Responsibility = System Administrator or General Ledger Super User**

1. Navigate to (N) Application > Validation > Set.
2. Enter the information for the first value set in the following fields:
  - **Value Set Name:** *YourInitials\_SALES\_REP*
  - **Description:** *YourInitials Sales Rep Value Set*
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 3
  - **Uppercase Only:** Selected
  - **Validation Type:** Independent
3. Save your work.
4. Enter the information for the second value set in the following fields:
  - **Value Set Name:** *YourInitials\_LOCATION*
  - **Description:** *YourInitials Location Value Set*
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 3
  - **Uppercase Only:** Selected
  - **Validation Type:** Independent
5. Save your work.
6. Enter the information for the third value set in the following fields:
  - **Value Set Name:** *YourInitials\_PROJECT*

- **Description:** *YourInitials* Project Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 30
  - **Validation Type:** Table
7. Click the “Edit Information” button to navigate to the Validation Table Information window.
  8. Enter Oracle Receivables as the table application and AR\_CUSTOMERS\_V as the table name.  
  
**Note:** AR\_CUSTOMERS\_V will not appear in the list of values because it has not been registered, but you can still use it for validation purposes.
  9. In the Table Columns region, enter the following information for the Value column:
    - **Name:** CUSTOMER\_NUMBER
    - **Type:** Char
    - **Size:** 30
  10. In the Table Columns region, enter the following information for the Meaning column:
    - **Name:** CUSTOMER\_NAME
    - **Type:** Char
    - **Size:** 50
  11. Save your work.

## Define your Structure

1. Navigate to (N) Application > Flexfield > Descriptive > Segments.
2. Query your descriptive flexfield.
3. Clear the Freeze Flexfield Definition check box.
4. In the Context Field region, enter information in the following fields:
  - **Prompt:** Is this a project?
  - **Value Required:** Selected

- **Override Allowed (Display Context):** Selected
5. In the Context Field Values region, select the Global Data Elements context and click the “Segments” button to navigate to the Segments Summary window.
  6. Delete the segments that you defined previously.
  7. Save your work.
  8. Navigate back to the Descriptive Flexfield Segments window.
  9. In the Context Field Values region, enter the following information for the first context:
    - **Code:** Yes
    - **Name:** Yes
    - **Description:** Yes
    - **Enabled:** Selected
  10. Click the “Segments” button to navigate to the Segments Summary window.
  11. Enter the information for the first segment in the following fields:
    - **Number:** 10
    - **Name:** Sales Rep
    - **Window Prompt:** Sales Rep
    - **Column:** ATTRIBUTE1
    - **Value Set:** *YourInitials*\_SALES\_REP
    - **Displayed:** Selected
    - **Enabled:** Selected
  12. Save your work.
  13. Enter the information for the second segment in the following fields:
    - **Number:** 20
    - **Name:** Location
    - **Window Prompt:** Location

- **Column:** ATTRIBUTE2
  - **Value Set:** *YourInitials\_LOCATION*
  - **Displayed:** Selected
  - **Enabled:** Selected
14. Save your work.
15. Enter the information for the third segment in the following fields:
- **Number:** 30
  - **Name:** Project
  - **Window Prompt:** Project
  - **Column:** ATTRIBUTE3
  - **Value Set:** *YourInitials\_PROJECT*
  - **Displayed:** Selected
  - **Enabled:** Selected
16. Save your work.
17. Navigate back to the Descriptive Flexfield Segments window.
18. In the Context Field Values region, enter the following information for the second context:
- **Code:** No
  - **Name:** No
  - **Description:** No
  - **Enabled:** Selected
19. Save your work.
20. Select the Freeze Flexfield Definition check box.
21. Click the “Compile” button to compile the flexfield definition.

## **Define your Values**

1. Navigate to (N) Application > Validation > Values.



2. In the Find window, select Value Set and find the *YourInitials\_SALES\_REP* value set.
3. In the Values, Effective region of the Segment Values window, define the following values:
  - **Value:** CJ
  - **Description:** Charles Jefferson
  - **Enabled:** Selected
  - **Value:** AR
  - **Description:** Ana Rodriguez
  - **Enabled:** Selected
  - **Value:** PL
  - **Description:** Philippe Lebeau
  - **Enabled:** Selected
4. Save your work.
5. Select Value Set and find the *YourInitials\_LOCATION* value set.
6. In the Values, Effective region, define the following values:
  - **Value:** NY
  - **Description:** New York
  - **Enabled:** Selected
  - **Value:** MA
  - **Description:** Madrid
  - **Enabled:** Selected
  - **Value:** PA
  - **Description:** Paris
  - **Enabled:** Selected
7. Save your work.

## Test your Descriptive Flexfield

1. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

# Practice - Define a DFF with a Dependent Segment

---

## Overview

To perform the practices in this lesson (and the next 1 as well), you will need a descriptive flexfield to work on. In the previous lesson, you were assigned a descriptive flexfield by your instructor. Please use that descriptive flexfield for this exercise. If you were not assigned a descriptive flexfield, please reference the previous lesson, and have your instructor assign one now.

In this practice, you will create a dependent value set to validate a descriptive flexfield segment. The descriptive flexfield will track the country and region for a salesperson. The list of values for the region segment will change depending on the country that is selected.

Remember that when defining a dependent value set, you must perform these steps in this order:

- Define the independent value set.
- Define the dependent value set.
- Define independent values.
- Define dependent values.

## Tasks

### Define your Value Sets

1. Use the Value Sets window to define two value sets.
  - Define an independent value set named *YourInitials\_COUNTRY*. Give the value set a description, a format type of Char, and a maximum size of 3. Specify that the values must be uppercase only, and enable security and Longlist for the value set.
  - Define a dependent value set named *YourInitials\_REGION*. Give the value set a description, a format type of Char, and a maximum size of 3. Specify that the values must be uppercase only, and enable security for the value set. Associate the dependent value set with the independent value set *YourInitials\_COUNTRY*. Assign the dependent value set a dependent default value of *YourInitials*, and give the default value a description.

### Define your Structure

2. After defining your value sets, use the Descriptive Flexfield Segments window to define the structure for your descriptive flexfield. For the purposes of this practice, use the same

descriptive flexfield that you worked on previously to enter your new structure. Unfreeze the flexfield definition first so that you can change the structure.

3. For the context field, change the prompt back to Context. Specify that no value is required and context override is not allowed. Disable the Yes and No contexts that you defined previously, and delete the segments that you defined for the Yes context.
4. Define two new segments for the Global Data Elements context.
  - Define a segment named Country Code, with a prompt of Country. Assign the segment the number 10, the column ATTRIBUTE1, and the value set *YourInitials\_COUNTRY*.
  - Define a segment named Region Code, with a prompt of Region. Assign the segment the number 20, the column ATTRIBUTE2, and the value set *YourInitials\_REGION*.
  - Ensure that both segments are displayed and enabled. When you finish defining the structure, freeze and compile your flexfield definition.

### Define your Values

5. After defining the value sets and segments for your flexfield, use the Segment Values window to define the values associated with the independent and dependent value sets.
6. Define and enable the following values for the *YourInitials\_COUNTRY* value set.
  - **Value:** USA
  - **Description:** United States
  - **Value:** CAN
  - **Description:** Canada
  - **Value:** UK
  - **Description:** United Kingdom
7. Define and enable the following values in the *YourInitials\_REGION* value set for the independent value USA.
  - **Value:** OK
  - **Description:** Oklahoma
  - **Value:** TX
  - **Description:** Texas
  - **Value:** WI

- **Description:** Wisconsin
8. Define and enable the following values in the *YourInitials\_REGION* value set for the independent value CAN.
    - **Value:** SAS
    - **Description:** Saskatchewan
    - **Value:** BC
    - **Description:** British Columbia
    - **Value:** YT
    - **Description:** Yukon Territory
  9. Define and enable the following values in the *YourInitials\_REGION* value set for the independent value UK.
    - **Value:** WA
    - **Description:** Wales
    - **Value:** SL
    - **Description:** Scotland
    - **Value:** EN
    - **Description:** England

### **Test your Descriptive Flexfields**

10. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

## Solution – Define a Descriptive Flexfield with a Dependent Segment

### Define your Value Sets

**Responsibility = System Administrator or General Ledger Super User**

1. Navigate to (N) Application > Validation > Set.
2. Enter the information for the first value set in the following fields:
  - **Value Set Name:** *YourInitials\_COUNTRY*
  - **Description:** *YourInitials* Country Value Set
  - **Security Available:** Selected
  - **Enable Longlist:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 3
  - **Uppercase Only:** Selected
  - **Validation Type:** Independent
3. Save your work.
4. Enter the information for the second value set in the following fields:
  - **Value Set Name:** *YourInitials\_REGION*
  - **Description:** *YourInitials* Region Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 3
  - **Uppercase Only:** Selected
  - **Validation Type:** Dependent
5. Click the “Edit Information” button to navigate to the Dependent Value Set Information window.
6. In the Independent Value Set region, enter *YourInitials\_COUNTRY* in the Name field.

7. In the Dependent Default Value region, enter *YourInitials* in the Value field and *YourInitials* Region Default in the Description field.
8. Save your work.

## Define your Structure

1. Navigate to (N) Application > Flexfield > Descriptive > Segments.
2. Query your descriptive flexfield.
3. Clear the Freeze Flexfield Definition check box.
4. In the Context Field region, enter information in the following fields:
  - **Prompt:** Context
  - **Value Required:** Cleared
  - **Override Allowed (Display Context):** Cleared
5. In the Context Field Values region, clear the Enabled check box for the Yes context and the No context.
6. Select the Yes context and click the “Segments” button to navigate to the Segments Summary window.
7. Delete the segments that you defined previously.
8. Save your work.
9. Navigate back to the Descriptive Flexfield Segments window.
10. In the Context Field Values region, select the Global Data Elements context and click the “Segments” button to navigate to the Segments Summary window.
11. Enter the information for the first segment in the following fields:
  - **Number:** 10
  - **Name:** Country Code
  - **Window Prompt:** Country
  - **Column:** ATTRIBUTE1
  - **Value Set:** *YourInitials\_COUNTRY*
  - **Displayed:** Selected

- **Enabled:** Selected
12. Save your work.
  13. Enter the information for the second segment in the following fields:
    - **Number:** 20
    - **Name:** Region Code
    - **Window Prompt:** Region
    - **Column:** ATTRIBUTE2
    - **Value Set:** *YourInitials\_REGION*
    - **Displayed:** Selected
    - **Enabled:** Selected
  14. Save your work.
  15. Navigate back to the Descriptive Flexfield Segments window.
  16. Select the Freeze Flexfield Definition check box.
  17. Click the “Compile” button to compile the flexfield definition.

### **Define your Values**

1. Navigate to (N) Application > Validation > Values.
2. In the Find window, select Value Set and find the *YourInitials\_COUNTRY* value set.
3. In the Values, Effective region of the Segment Values window, define the following values:
  - **Value:** USA
  - **Description:** United States
  - **Value:** CAN
  - **Description:** Canada
  - **Value:** UK
  - **Description:** United Kingdom
4. Save your work.



5. Select Value Set and find the *YourInitials\_REGION* value set and the independent value USA.
6. Define and enable the following values in the *YourInitials\_REGION* value set for the independent value USA.
  - **Value:** OK
  - **Description:** Oklahoma
  - **Value:** TX
  - **Description:** Texas
  - **Value:** WI
  - **Description:** Wisconsin
7. Save your work.
8. Select Value Set and find the *YourInitials\_REGION* value set and the independent value CAN.
9. Define and enable the following values in the *YourInitials\_REGION* value set for the independent value CAN.
  - **Value:** SAS
  - **Description:** Saskatchewan
  - **Value:** BC
  - **Description:** British Columbia
  - **Value:** YT
  - **Description:** Yukon Territory
10. Save your work
11. Select Value Set and find the *YourInitials\_REGION* value set and the independent value UK.
12. Define and enable the following values in the *YourInitials\_REGION* value set for the independent value UK.
  - **Value:** WA
  - **Description:** Wales

- **Value:** SL
- **Description:** Scotland
- **Value:** EN
- **Description:** England

13. Save your work.

### **Test your Descriptive Flexfields**

1. After defining your value sets, segments, and values, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

### Summary

- **Descriptive flexfields gather additional information.**
- **Design the descriptive flexfield to support the different needs of different users.**
- **Define flexfield level attributes.**
- **Define global segments for the Global Data Elements structure.**
- **Define a reference or context field if using different contexts.**
- **Define a structure for each context containing the segments appropriate for that context.**
- **Freeze and compile the finished definition.**

ORACLE

### Summary

This lesson discussed how to plan a descriptive flexfield. Descriptive flexfields gather additional information beyond that gathered by Oracle Applications. Not all users of the same descriptive flexfield need the same information. Descriptive flexfields can be customized so that each user sees only the information needed.

Once you have planned the structure of your descriptive flexfield, the actual definition process is straightforward. If this flexfield uses any new value sets, plan those first. Define the flexfield header attributes. Define all global segments for the Global Data Elements structure.

If you are using multiple contexts, define either a reference field on the base window, or a context field on the flexfield itself to control which context structure appears. For each context, define a structure containing the segments appropriate for that context.

When you have finished defining all your structures and segments, freeze your definition and compile the flexfield. This makes the flexfield available for others to use.



# Using Advanced Validation Capabilities

## Chapter 18

# Using Advanced Validation Capabilities

ORACLE

### Objectives

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

- **Use values retrieved from application tables for validating input**
- **Reference profile options when validating input**
- **Use another field on the same form for validation**
- **Use a value from a previously used value set for validation**

ORACLE

### Lesson Topics

A previous lesson discussed creating and using value sets to check user input. You performed practices in which you defined value sets of the validation types None, Independent, and Dependent.

In this lesson, you will learn how to use values stored in applications tables for validation. You will also learn how to restrict the set of values that is returned from the table.

At the end of this lesson, you should be able to:

- Use values retrieved from application tables for validating input
- Reference profile option values when validating input
- Use a value from another field on the same form for validation
- Use the value from a previously used value set for validation

### Overview

- **Using table validated value sets**
- **Using profile option values for validating**
- **Using the value from a prior field on a form**
- **Using the value from a prior value set**

ORACLE

### Overview

There are several advanced techniques you can use when defining value sets for validating input. These techniques use values obtained from a location other than a list specifically defined for the value set. These locations include:

- Values retrieved from an application table
- The current value for a particular profile option
- A value used earlier in a field on the form
- A value used earlier in another value set



### Advanced Validation Options

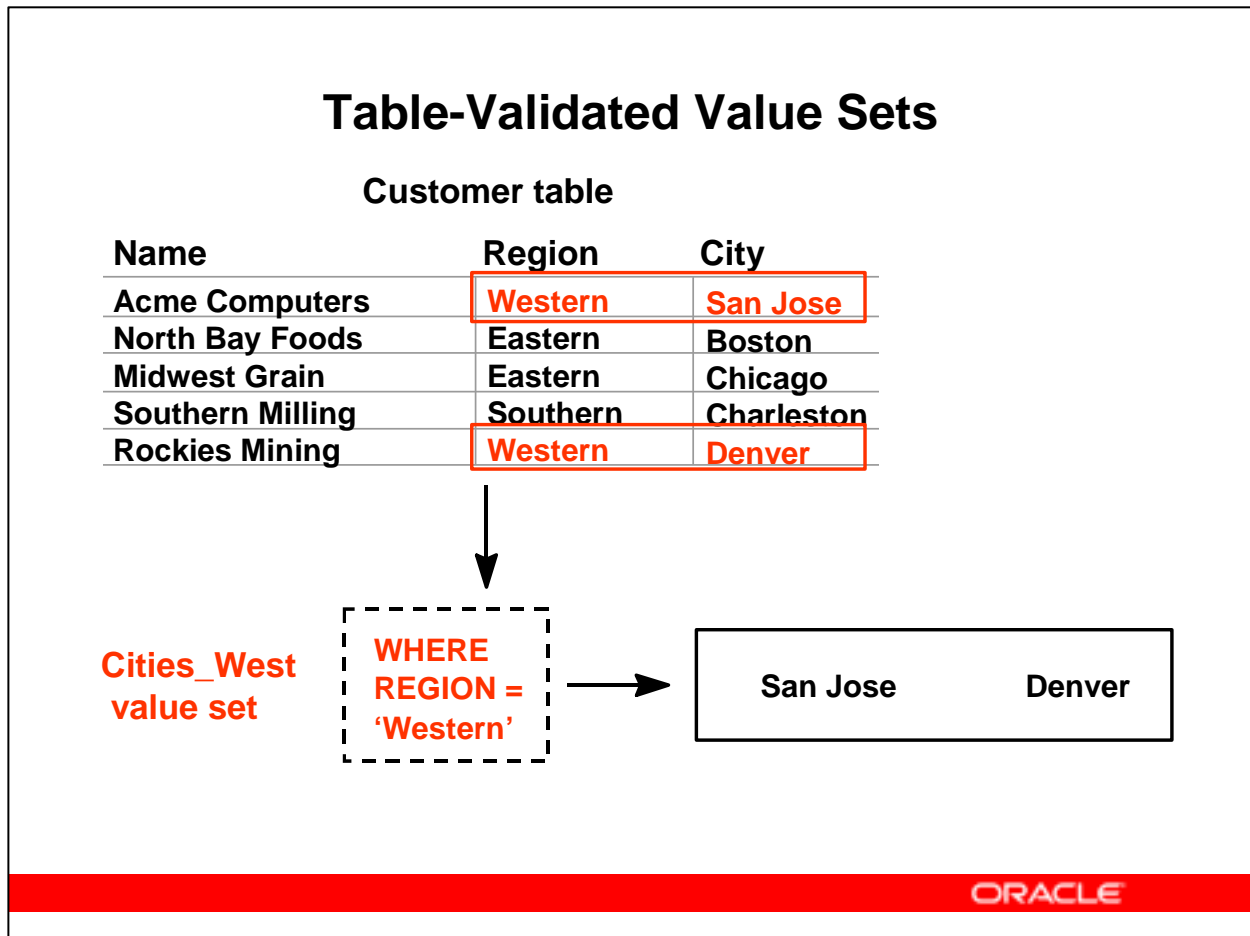
- **Use values from application tables.**
- **Use profile option values for validation.**
- **Use values from other fields on the same form.**
- **Use cascading dependencies.**

ORACLE

#### Using Advanced Validation Capabilities

Besides obtaining values from an application table, you can obtain other values for validation from:

- Current profile options
- A prior field on the same form
- A prior value set used by a segment on the same form



### Using Existing Values in Application Tables for Validation

- Use the values stored in application tables to provide values to flexfield segments and report parameters.
- You can use an SQL WHERE clause to restrict the set of values returned.
- Remember the Longlist feature for large tables.
- Dependent value sets cannot use a table-validated value set as their Independent value set.

### Avoiding Double Maintenance

- Values stored in application tables are maintained by the application.
- Table validation is especially useful when values already exist in feeder systems.

### Defining a Table-Validated Value Set

Use the Validation Table Information window to enter:

- Table Application
- Table Name
- Table Columns
- Value
- Meaning
- ID
- Where/Order By
- Additional Columns

(N) Application—>Validation—>Set (B) Edit Information

ORACLE

### Defining a Table-Validated Value Set

Match the format options to the values stored in the table when defining table-validated value sets. Specify which table and which columns to use for this value set in the Validation Table Information window.

**Value Column** - Specify the name, type, and size of the column containing the values to be used for validating. The list of values limits itself to columns matching the format choices made in the Value Sets window.

**Meaning Column** - This column contains the descriptions for the values in the Value column.

**Hidden ID Column** - This contains non-displayed values, for example, an application ID number corresponding to the displayed application name. Many Oracle Applications key flexfields do not allow the use of the hidden ID column.

**Additional Columns** - Any additional columns to display. You can use a column alias to provide a title for the column. Include a width indicator for the column. For example: user\_formname "Form Title"(30), application\_name "Application Name"(35)

### **Restricting Values Retrieved from a Table**

- **If not all the values stored in the table are appropriate for your value set, use a SQL WHERE clause to restrict the set of values returned from the table.**
- **You can also use an ORDER BY statement to control the order in which the values are returned.**

ORACLE

### Limitations on Using SQL WHERE Clauses

- Do not use **WHERE** clauses with the Accounting Flexfield.
- Do not use **HAVING** or **GROUP BY** clauses.
- Do not use **UNION**, **INTERSECT**, **MINUS**, **PLUS**, or other set operators except in a subquery.
- If you need a complex SQL clause to select your values from a table, you should instead first define a view over the table which selects the rows you need, and then define the value set over the view.

ORACLE

### Using Multiple Tables for Validation

- **Specify more than one name in the Table Name field.**

```
GL_SETS_OF_BOOKS SB, AR_SYSTEM_PARAMETERS SP
```

- **No value is required in the Table Application field.**
- **Use the JOIN statement in the WHERE clause to use columns from multiple tables. Use regular SQL syntax to specify the table join**

```
WHERE SB.SET_OF_BOOKS_ID = SP.SET_OF_BOOKS_ID
```

ORACLE

# Practice - Table Validation

---

## Overview

In this practice, you will use an application table to validate a descriptive flexfield segment. You will create a table-validated value set to validate employee numbers based on a Human Resources application table called PER\_ALL\_PEOPLE\_F. Then you will add an additional segment to the descriptive flexfield structure you defined in the Define a DFF lesson to track the salesperson's employee number.

## Tasks

### Define your Value Set

1. Use the Value Sets window to define a table-validated value set named *YourInitials\_EMP\_NUM*. Give the value set a description, a format type of Char, and a maximum size of 30. Enable security and Longlist for the value set. Specify the table PER\_ALL\_PEOPLE\_F in the Oracle Human Resources application as the validation table. Assign the Value column the column name EMPLOYEE\_NUMBER, a type of Varchar2, and a size of 30. Assign the Meaning column the column name FULL\_NAME, a type of Varchar2, and a size of 240. Restrict the value set to include only active employees by specifying the following Where/Order By clause:

```
WHERE CURRENT_EMPLOYEE_FLAG = 'Y'
```

### Define your Structure

2. After defining your value set, use the Descriptive Flexfield Segments window to define the structure for your descriptive flexfield. For the purposes of this practice, use the same descriptive flexfield that you worked on previously to enter your new structure. Unfreeze the flexfield definition first so that you can change the structure.
3. You can keep the two segments that you defined previously for the Global Data Elements context. Now define a new segment for the Global Data Elements context. Name the segment Employee Number, with a prompt of Employee Number. Assign the segment the number 30, the column ATTRIBUTE3, and the value set *YourInitials\_EMP\_NUM*.
4. Ensure that the new segment is displayed and enabled. When you finish defining the structure, freeze and compile your flexfield definition.

### Test your Descriptive Flexfield

5. After defining your value set and segment, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

## Solution – Table Validation

### Define your Value Set

**Responsibility = System Administrator or General Ledger Super User**

1. Navigate to (N) Application > Validation > Set.
2. Enter the information for the value set in the following fields:
  - **Value Set Name:** *YourInitials\_EMP\_NUM*
  - **Description:** *YourInitials* Employee Number Value Set
  - **Security Available:** Selected
  - **Enable Longlist:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 30
  - **Validation Type:** Table
3. Click the “Edit Information” button to navigate to the Validation Table Information window.
4. Use the lists of values in the Table Application and Table Name fields to select Oracle Human Resources as the table application and PER\_ALL\_PEOPLE\_F as the table name.
5. In the Table Columns region, use the lists of values in the following fields to select this information for the Value column:
  - **Name:** EMPLOYEE\_NUMBER
  - **Type:** Varchar2
  - **Size:** 30
6. In the Table Columns region, use the lists of values in the following fields to select this information for the Meaning column:
  - **Name:** FULL\_NAME
  - **Type:** Varchar2
  - **Size:** 240
7. In the Where/Order By field, enter



WHERE CURRENT\_EMPLOYEE\_FLAG = 'Y'

8. Save your work.

### Define your Structure

1. Navigate to (N) Application > Flexfield > Descriptive > Segments.
2. Query your descriptive flexfield.
3. Clear the Freeze Flexfield Definition check box.
4. In the Context Field Values region, select the Global Data Elements context and click the “Segments” button to navigate to the Segments Summary window.
5. Keep the two segments that you defined previously.
6. Enter the information for the third segment in the following fields:

- **Number:** 30
- **Name:** Employee Number
- **Window Prompt:** Employee Number
- **Column:** ATTRIBUTE3
- **Value Set:** *YourInitials\_EMP\_NUM*
- **Displayed:** Selected
- **Enabled:** Selected

7. Save your work.

8. Navigate back to the Descriptive Flexfield Segments window.

9. Select the Freeze Flexfield Definition check box.

10. Click the “Compile” button to compile the flexfield definition.

### Test your Descriptive Flexfield

1. After defining your value set and segment, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

### Other Advanced Validation Options

- **\$PROFILES\$** — References the current value of a profile option
- **:Block.field** — References the value of an earlier appearing field on the same form
- **\$FLEX\$** — References the value from a value set used earlier on the same form

ORACLE

### Other Advanced Validation Options

#### Using \$PROFILES\$

You can reference the current value of a profile option in a WHERE clause by prefixing the name of the profile option with \$PROFILES\$. For example:

`:$PROFILES$.profile_option_name`

#### Using :Block.field

You can test against the value of a prior field on the same form in a WHERE clause by referring to the field using the format *Block\_name.field\_name*. To find out the block and field names, click in the field and then select Help—>Tool—>Examine from the menu bar. This displays the block name and the field name for the selected field.

#### Using \$FLEX\$

You can refer to the current value of a previously used value set on the same form by using `$FLEX$.value_set_name`.

### Using \$PROFILE\$

- Use this keyword in a WHERE clause to reference a profile option value.
- Use the profile option internal name, not the end-user name.

```
WHERE SET_OF_BOOKS_ID =  
      :$PROFILE$.GL_SET_OF_BOOKS_ID
```

ORACLE

### Using :block.field

- **The keyword :block.field is the Oracle Forms internal name of a field on the base window.**
- **Using :block.field is different from using a descriptive flexfield reference field in that the flexfield structure does not change based on the different :block.field values.**
- **Use this value set only with flexfields on windows that have the same block.field available.**

ORACLE

#### Using :*block.field* to Refer to a Field on the Base Window

- Use this value set only with flexfields on windows that have the same *block.field* available. If the list of values cannot find the correct field, the segment cannot accept any value as valid. Do not use this value set with flexfields appearing on other windows or with report parameters.
- This argument requires the same *:block.field* on every window where a value set based on this validation table could be used. For example, if the flexfield whose segment uses *block.field* validation appears on seven different windows, the same *block.field* must also be present on those seven forms.
- If sharing value sets among multiple descriptive flexfields, all windows that use any of those flexfields must have this *block.field*.

### Using \$FLEX\$

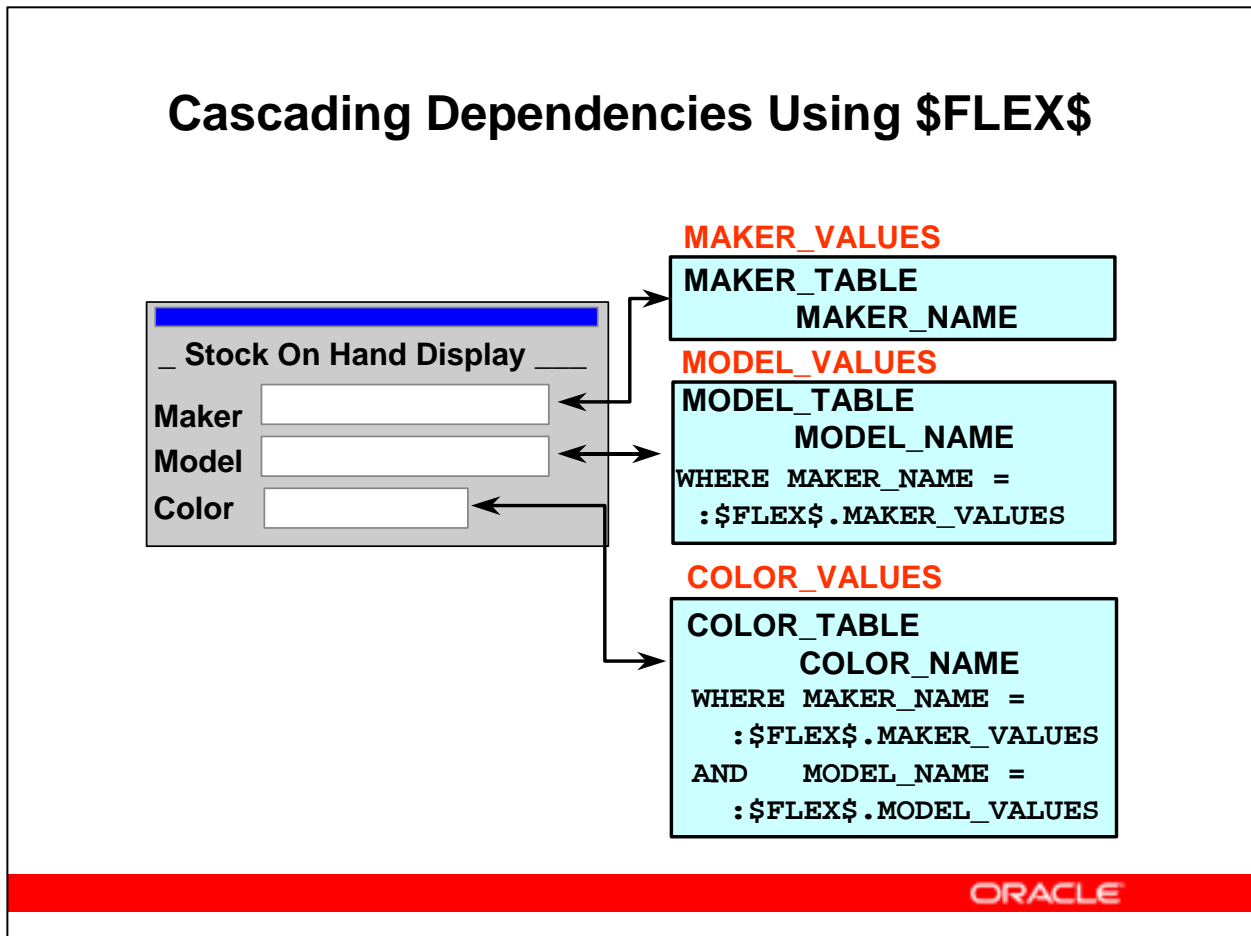
- Use `$FLEX$.value_set_name` to reference the value in a previous segment of the same flexfield structure.
- `Value_set_name` refers to the value set used by the previous segment.
- If two segments use the same value set, the value from the closest segment is used.

```
WHERE JOURNAL_TYPE =  
      :$FLEX$.GL_SRS_JOURNAL_TYPE
```

ORACLE

### Using \$FLEX\$ to Refer to a Value Used in a Previous Value Set

- The flexfield segment or report parameter always uses the value from the Value column (not the Meaning or Hidden columns).
- Because `Value_set_name` is case-sensitive, you must ensure that the name in the WHERE clause exactly matches the value set name defined.
- Use only letters, numbers, and underscores in value set names if you want to use them with \$FLEX\$.
- Do not use quotes, spaces, or other special characters in these value set names.
- The value set must point to a previous segment in the same flexfield structure. For descriptive flexfield context-sensitive segments, the value set must point to the previous segment in the same context-sensitive structure.
- Multiple `$FLEX$.value_set_name` statements can be used in the same WHERE clause.



### Using Cascading Dependencies

You can use the \$FLEX\$ keyword to create chains of validation checks called cascading dependencies.

## Practice - Cascading Dependencies with \$FLEX\$

---

### Overview

In this practice, you will create a descriptive flexfield that uses cascading dependencies to validate the values entered. The flexfield will track the manufacturer, model, and color of a vehicle. The possible model values will be restricted depending on the manufacturer value, and the possible color values will be restricted depending on the model value.

To set up the cascading dependencies, you will create table-validated value sets using the following tables:

#### CAR\_MAKERS

MANUFACTURER_ NAME	MANUFACTURER_ DESCRIPTION	MANUFACTURER_ ID
Ford	Ford	01
Nissan	Nissan	02
Chevrolet	Chevrolet	03
Chrysler	Chrysler	04

#### CAR\_MODELS

MODEL_ NAME	MODEL_ DESCRIPTION	MODEL_ ID	MANUFACTURER_ ID
Mustang	Mustang	011	01
Windstar	Windstar	012	01
240SX	240SX	013	02
Altima	Altima	014	02
Corvette	Corvette	015	03
Camaro	Camaro	016	03
Sebring	Sebring	017	04
Concorde	Concorde	018	04

## CAR\_COLORS

COLOR_	COLOR_	COLOR_	MODEL_	MANUFACTURER_
NAME	DESCRIPTION	ID	ID	ID
Red	Red	0101	011	01
Black	Black	0102	011	01
Green	Green	0103	011	01
Black	Black	0104	012	01
Blue	Blue	0105	012	01
White	White	0106	012	01
Champagne	Champagne	0107	013	02
Red	Red	0108	013	02
Sea Green	Sea Green	0109	013	02
Silver	Silver	0110	014	02
Gold	Gold	0111	014	02
Teal	Teal	0112	014	02
Black	Black	0113	015	03
Red	Red	0114	015	03
Yellow	Yellow	0115	015	03
Purple	Purple	0116	016	03
Blue	Blue	0117	016	03
Green	Green	0118	016	03
Maroon	Maroon	0119	017	04
Silver	Silver	0120	017	04
Green	Green	0121	017	04



White	White	0122	018	04
Blue	Blue	0123	018	04
Silver	Silver	0124	018	04

## Tasks

### Define your Value Sets

- Use the Value Sets window to define three value sets.
  - Define a table-validated value set named *YourInitials\_MFG*. Give the value set a description, a format type of Char, and a maximum size of 10. Enable security for the value set. Specify the table CAR\_MAKERS as the validation table. Assign the Value column the column name MANUFACTURER\_NAME, a type of Varchar2, and a size of 10. Assign the Meaning column the column name MANUFACTURER\_DESCRIPTION, a type of Varchar2, and a size of 30. Assign the ID column the column name MANUFACTURER\_ID, a type of Number, and a size of 2.
  - Define a table-validated value set named *YourInitials\_MODEL*. Give the value set a description, a format type of Char, and a maximum size of 10. Enable security for the value set. Specify the table CAR\_MODELS as the validation table. Assign the Value column the column name MODEL\_NAME, a type of Varchar2, and a size of 10. Assign the Meaning column the column name MODEL\_DESCRIPTION, a type of Varchar2, and a size of 30. Assign the ID column the column name MODEL\_ID, a type of Number, and a size of 3. Specify the following Where/Order By clause:

WHERE MANUFACTURER\_ID = :\$FLEX\$.XX\_MFG

- Define a table-validated value set named *YourInitials\_COLOR*. Give the value set a description, a format type of Char, and a maximum size of 10. Enable security for the value set. Specify the table CAR\_COLORS as the validation table. Assign the Value column the column name COLOR\_NAME, a type of Varchar2, and a size of 10. Assign the Meaning column the column name COLOR\_DESCRIPTION, a type of Varchar2, and a size of 30. Assign the ID column the column name COLOR\_ID, a type of Number, and a size of 4. Specify the following Where/Order By clause:

WHERE MODEL\_ID = :\$FLEX\$.XX\_MODEL

ORDER BY COLOR\_NAME

### Define your Structure

- After defining your value sets, use the Descriptive Flexfield Segments window to define the structure for your descriptive flexfield. For the purposes of this practice, use the same descriptive flexfield that you worked on previously to enter your new structure. Unfreeze the flexfield definition first so that you can change the structure.

3. Delete the segments that you defined previously for the Global Data Elements context. Then define three new segments for the Global Data Elements context.
  - Define a segment named Manufacturer, with a prompt of Manufacturer. Assign the segment the number 10, the column ATTRIBUTE1, and the value set *YourInitials\_MFG*.
  - Define a segment named Model, with a prompt of Model. Assign the segment the number 20, the column ATTRIBUTE2, and the value set *YourInitials\_MODEL*.
  - Define a segment named Color, with a prompt of Color. Assign the segment the number 30, the column ATTRIBUTE3, and the value set *YourInitials\_COLOR*.
  - Ensure that all three segments are displayed and enabled. When you finish defining the structure, freeze and compile your flexfield definition.

### **Test your Descriptive Flexfield**

4. After defining your value sets and segments, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

## Solution – Cascading Dependencies with \$FLEX\$

### Define your Value Sets

**Responsibility = System Administrator or General Ledger Super User**

1. Navigate to (N) Application > Validation > Set.
2. Enter the information for the first value set in the following fields:
  - **Value Set Name:** *YourInitials\_MFG*
  - **Description:** *YourInitials* Car Manufacturers Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 10
  - **Validation Type:** Table
3. Click the “Edit Information” button to navigate to the Validation Table Information window.
4. Enter CAR\_MAKERS as the table name.
5. In the Table Columns region, enter the following information for the Value column:
  - **Name:** MANUFACTURER\_NAME
  - **Type:** Varchar2
  - **Size:** 10
6. In the Table Columns region, enter the following information for the Meaning column:
  - **Name:** MANUFACTURER\_DESCRIPTION
  - **Type:** Varchar2
  - **Size:** 30
7. In the Table Columns region, enter the following information for the ID column:
  - **Name:** MANUFACTURER\_ID
  - **Type:** Number

- **Size:** 2
8. Save your work.
  9. Enter the information for the second value set in the following fields:
    - **Value Set Name:** *YourInitials\_MODEL*
    - **Description:** *YourInitials* Car Models Value Set
    - **Security Available:** Selected
    - **Format Type:** Char
    - **Maximum Size:** 10
    - **Validation Type:** Table
  10. Click the “Edit Information” button to navigate to the Validation Table Information window.
  11. Enter CAR\_MODELS as the table name.
  12. In the Table Columns region, enter the following information for the Value column:
    - **Name:** MODEL\_NAME
    - **Type:** Varchar2
    - **Size:** 10
  13. In the Table Columns region, enter the following information for the Meaning column:
    - **Name:** MODEL\_DESCRIPTION
    - **Type:** Varchar2
    - **Size:** 30
  14. In the Table Columns region, enter the following information for the ID column:
    - **Name:** MODEL\_ID
    - **Type:** Number
    - **Size:** 3
  15. In the Where/Order By field, enter  
  
WHERE MANUFACTURER\_ID = :\$FLEX\$.XX\_MFG

16. Save your work.
17. Enter the information for the third value set in the following fields:
  - **Value Set Name:** *YourInitials\_COLOR*
  - **Description:** *YourInitials* Car Colors Value Set
  - **Security Available:** Selected
  - **Format Type:** Char
  - **Maximum Size:** 10
  - **Validation Type:** Table
18. Click the “Edit Information” button to navigate to the Validation Table Information window.
19. Enter CAR\_COLORS as the table name.
20. In the Table Columns region, enter the following information for the Value column:
  - **Name:** COLOR\_NAME
  - **Type:** Varchar2
  - **Size:** 10
21. In the Table Columns region, enter the following information for the Meaning column:
  - **Name:** COLOR\_DESCRIPTION
  - **Type:** Varchar2
  - **Size:** 30
22. In the Table Columns region, enter the following information for the ID column:
  - **Name:** COLOR\_ID
  - **Type:** Number
  - **Size:** 4
23. In the Where/Order By field, enter  
  
WHERE MODEL\_ID = :\$FLEX\$.XX\_MODEL  
  
ORDER BY COLOR\_NAME

24. Save your work.

### Define your Structure

1. Navigate to (N) Application > Flexfield > Descriptive > Segments.
2. Query your descriptive flexfield.
3. Clear the Freeze Flexfield Definition check box.
4. Select the Global Data Elements context and click the “Segments” button to navigate to the Segments Summary window.
5. Delete the segments that you defined previously.
6. Save your work.
7. Enter the information for the first segment in the following fields:
  - **Number:** 10
  - **Name:** Manufacturer
  - **Window Prompt:** Manufacturer
  - **Column:** ATTRIBUTE1
  - **Value Set:** *YourInitials\_MFG*
  - **Displayed:** Selected
  - **Enabled:** Selected
8. Save your work.
9. Enter the information for the second segment in the following fields:
  - **Number:** 20
  - **Name:** Model
  - **Window Prompt:** Model
  - **Column:** ATTRIBUTE2
  - **Value Set:** *YourInitials\_MODEL*
  - **Displayed:** Selected

- **Enabled:** Selected
10. Save your work.
  11. Enter the information for the third segment in the following fields:
    - **Number:** 30
    - **Name:** Color
    - **Window Prompt:** Color
    - **Column:** ATTRIBUTE3
    - **Value Set:** *YourInitials\_COLOR*
    - **Displayed:** Selected
    - **Enabled:** Selected
  12. Save your work.
  13. Navigate back to the Descriptive Flexfield Segments window.
  14. Select the Freeze Flexfield Definition check box.
  15. Click the “Compile” button to compile the flexfield definition.

### **Test your Descriptive Flexfield**

1. After defining your value sets and segments, navigate to your descriptive flexfield and test the results of your work. When you are finished, exit without saving.

### Using Special Value Sets

**Many report parameters use special value sets to pass an entire key flexfield combination as the value for the report parameter.**

ORACLE



### **Using Pair Value Sets**

**Pair value sets use two segments to pass a range flexfield as the value.**

ORACLE

### Summary

- **Value sets can use values from application tables for validation.**
- **Value sets can use values from profile options for validation.**
- **Value sets can use values from a previously used field or value set for validation.**

ORACLE

### Summary

Some value sets (for example, Independent and Dependent) have lists of approved values explicitly defined. Other types of value sets, however, can use values from other locations for validation. Using values from an application table is one example of these alternate locations. The current value of a profile option can also be used. The value from a previously appearing field on a form can be referenced as well. Finally, the value from a previously used value set can be used by all subsequent value sets.

# Overview of Workflow

## Chapter 19

# Introduction of Workflow

ORACLE

### Objectives

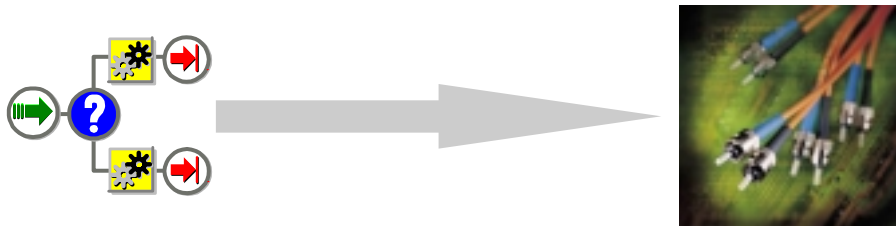
**After completing this lesson, you should be able to do the following:**

- **Explain the benefits of Oracle Workflow.**
- **Discuss the concept of a workflow process.**
- **Discuss the concept of business events.**

ORACLE

### Enabling E-Business

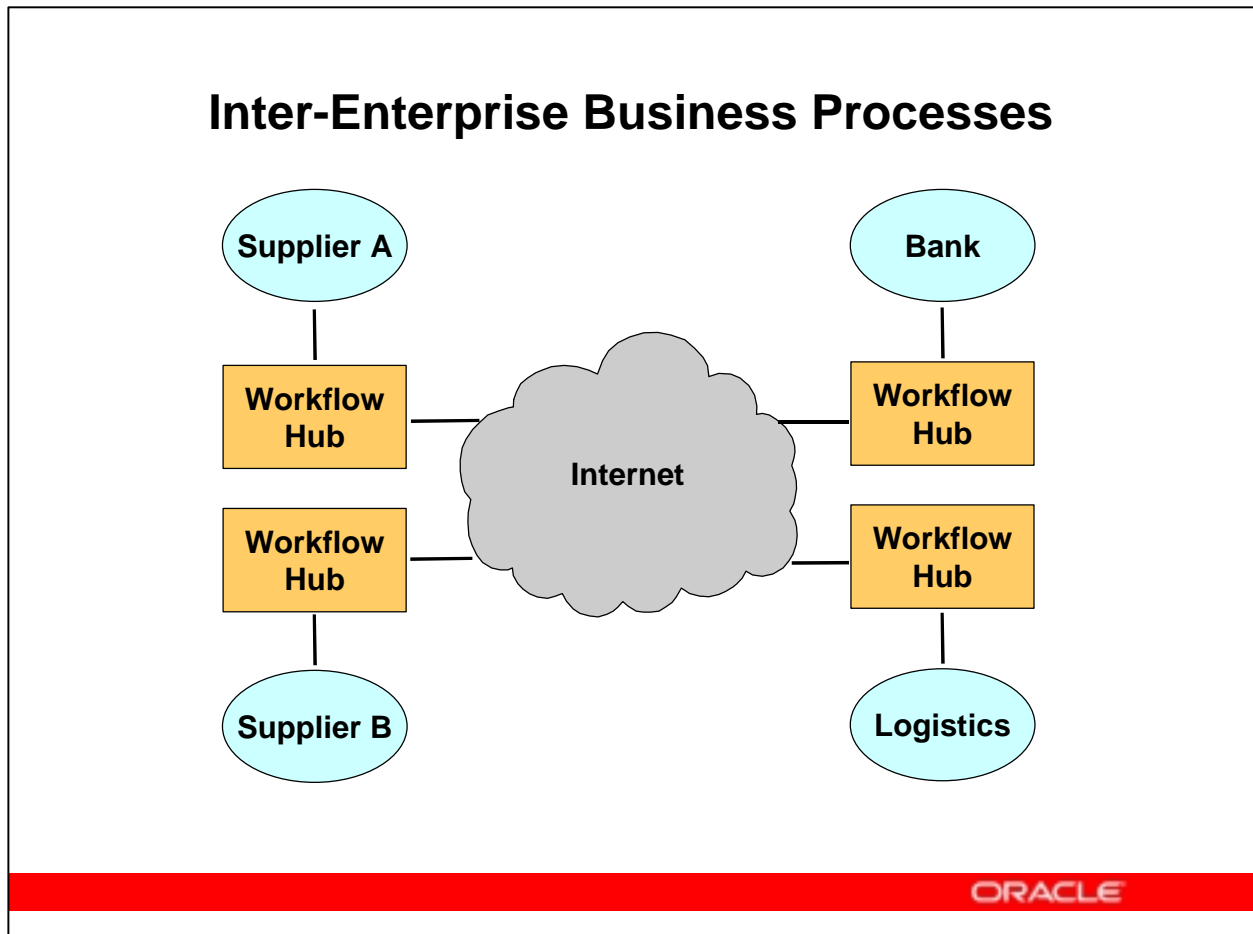
**Streamlined business processes play a critical role in the transformation to e-business. Oracle Workflow delivers a complete business process definition, automation, and integration solution.**



**Oracle Workflow: The wiring for e-business**

ORACLE

## Inter-Enterprise Business Processes

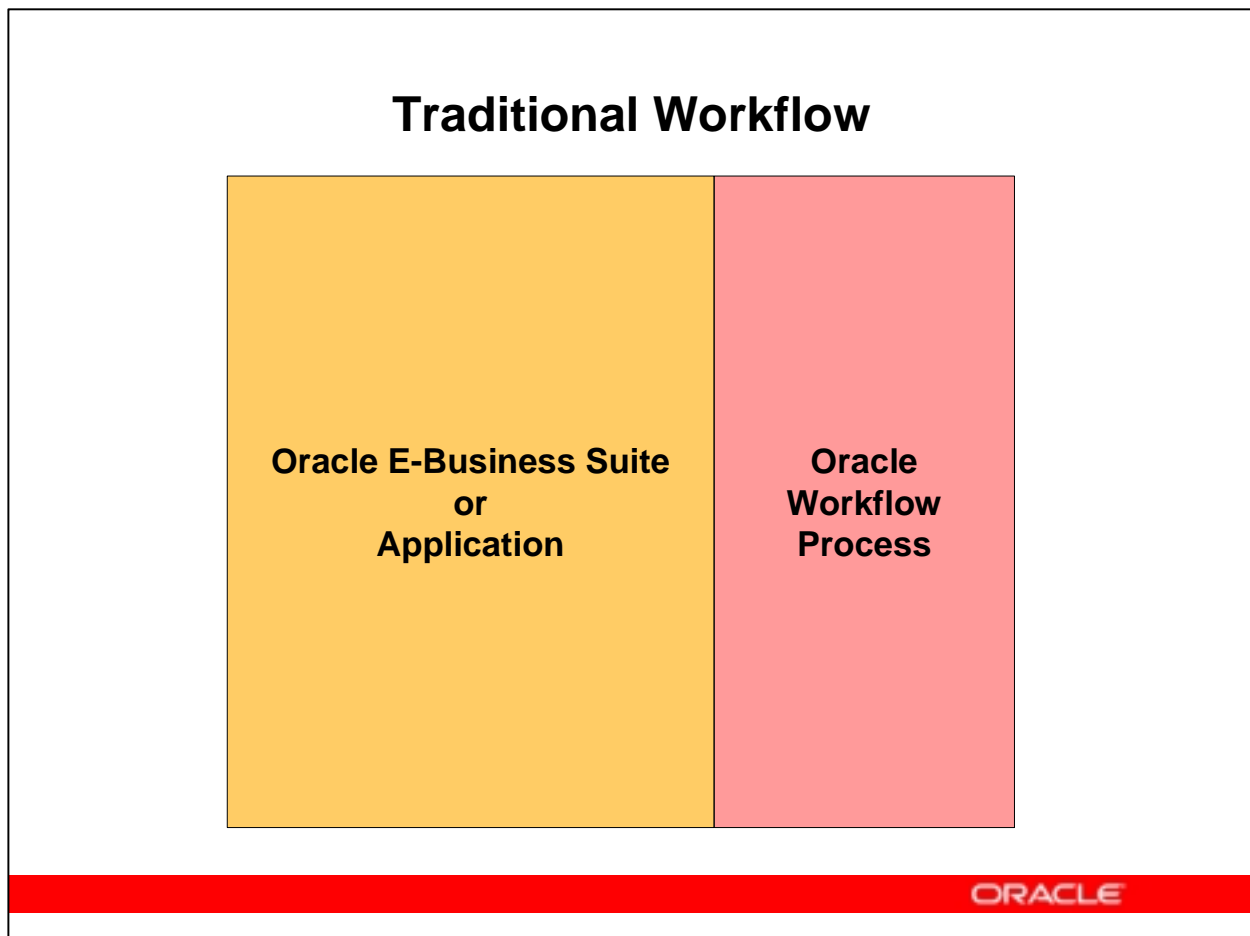


### Inter-Enterprise Business Processes

In e-business, different enterprises need to communicate with each other over the Internet. Oracle Workflow with the Business Event System can model business processes that span all the enterprises involved in an end-to-end process.

## Traditional Workflow

---



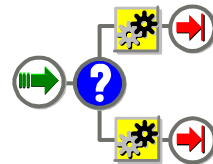
### Traditional Workflow

Traditional applications-based workflow processes are launched from a business application through APIs hard-coded within the application. These processes model the business rules in the individual local application and are made up of activities executed by the Workflow Engine only in that application's system. For example, the modeling of an approval hierarchy is a common use of Oracle Workflow in this scenario.



### Workflow-Driven Business Processes

- **Oracle Workflow automates and streamlines business processes contained within and between enterprises.**
- **For example, you can use workflow processes to:**
  - Add personalized trading partner rules
  - Validate self-service transactions
  - Achieve closed loop business intelligence
  - Approve standard business documents
  - Step through daily transaction flows
  - Integrate with trading partner systems

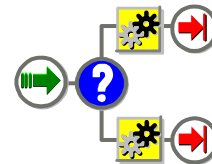


ORACLE

### Workflow-Driven Business Processes

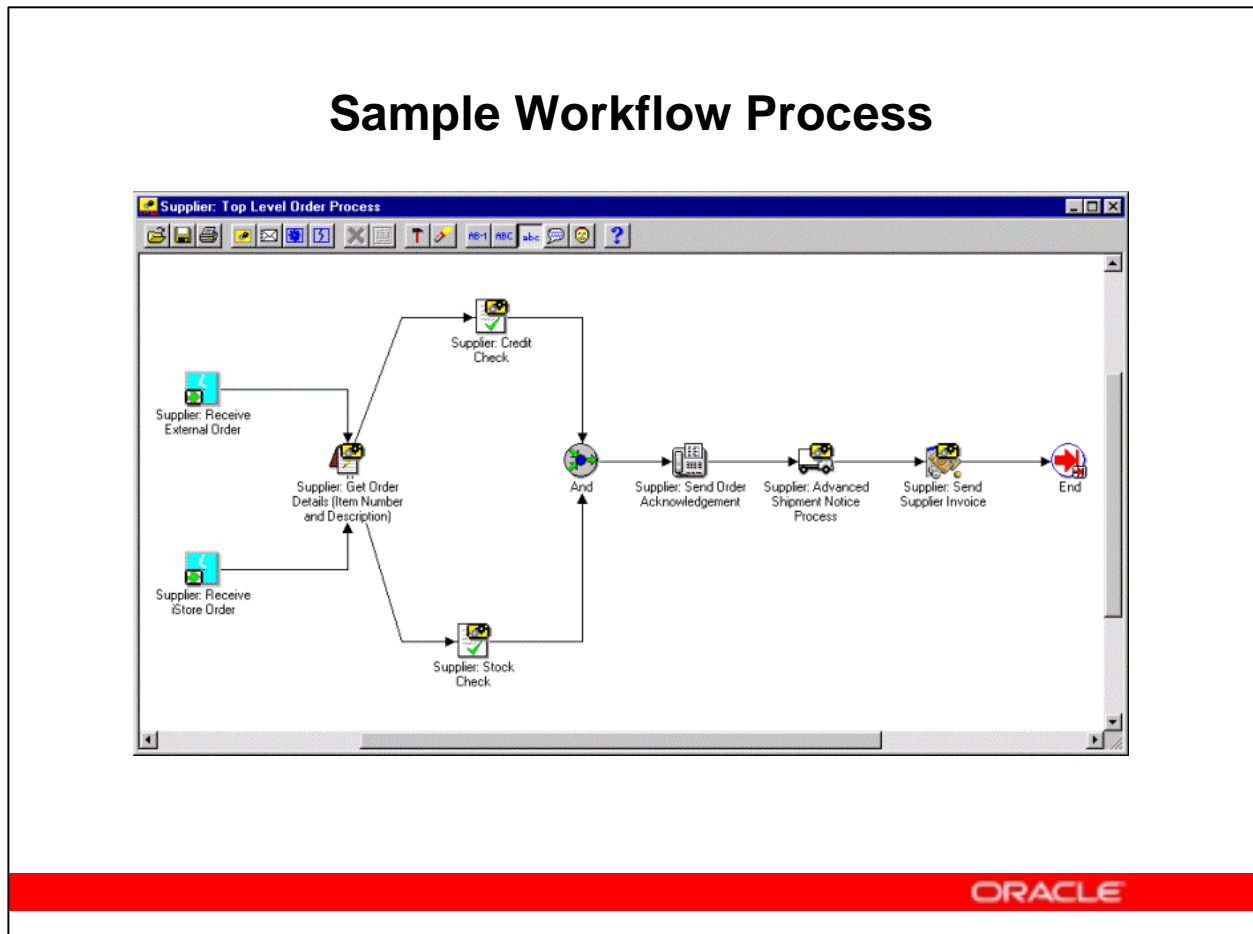
**Oracle Workflow lets you focus on managing the business process, not individual transactions.**

- **Define and implement your business policies**
- **Streamline the entire process**
- **Route information**
- **Capture exceptions and take action**
- **Build continuous improvements directly into the process definition**
- **Adapt your processes as your business changes**



ORACLE

## Sample Workflow Process



### Sample Workflow Process

A workflow process consists of a sequence of activities that together make up a business flow, expressing your organization's policies and rules. The activities can include significant business events, automated functions, notifications to users, or subprocesses.

This example shows a sample order processing workflow process that includes business events. The example comes from the Event System Demonstration workflow which is available with the standalone version of Oracle Workflow.

## Example: Expense Report Processing

---

### Example: Expense Report Processing

**Goal:** Fast and efficient payment of expense reports that conform to corporate policy

**Solution:** Workflow-enabled self-service expense reporting

**Accounts Payable professionals should spend their time:**

- Entering expense reports
- Responding to telephone queries regarding the status of an expense report
- **Defining and enforcing corporate policy, saving the company money**

ORACLE

### Example: Expense Report Processing

Oracle Workflow can help save time in expense report processing by reducing repetitive data entry tasks and by providing self-service monitoring capabilities.

## Example: New Hire Processing

---

### Example: New Hire Processing

**Goal:** Empower managers in hiring the best people

**Solution:** Workflow-enabled self-service hiring

#### Hiring managers should spend their time:

- Determining whose approval is required
- Chasing the HR department to mail offer letters
- Interviewing candidates and hiring the best people

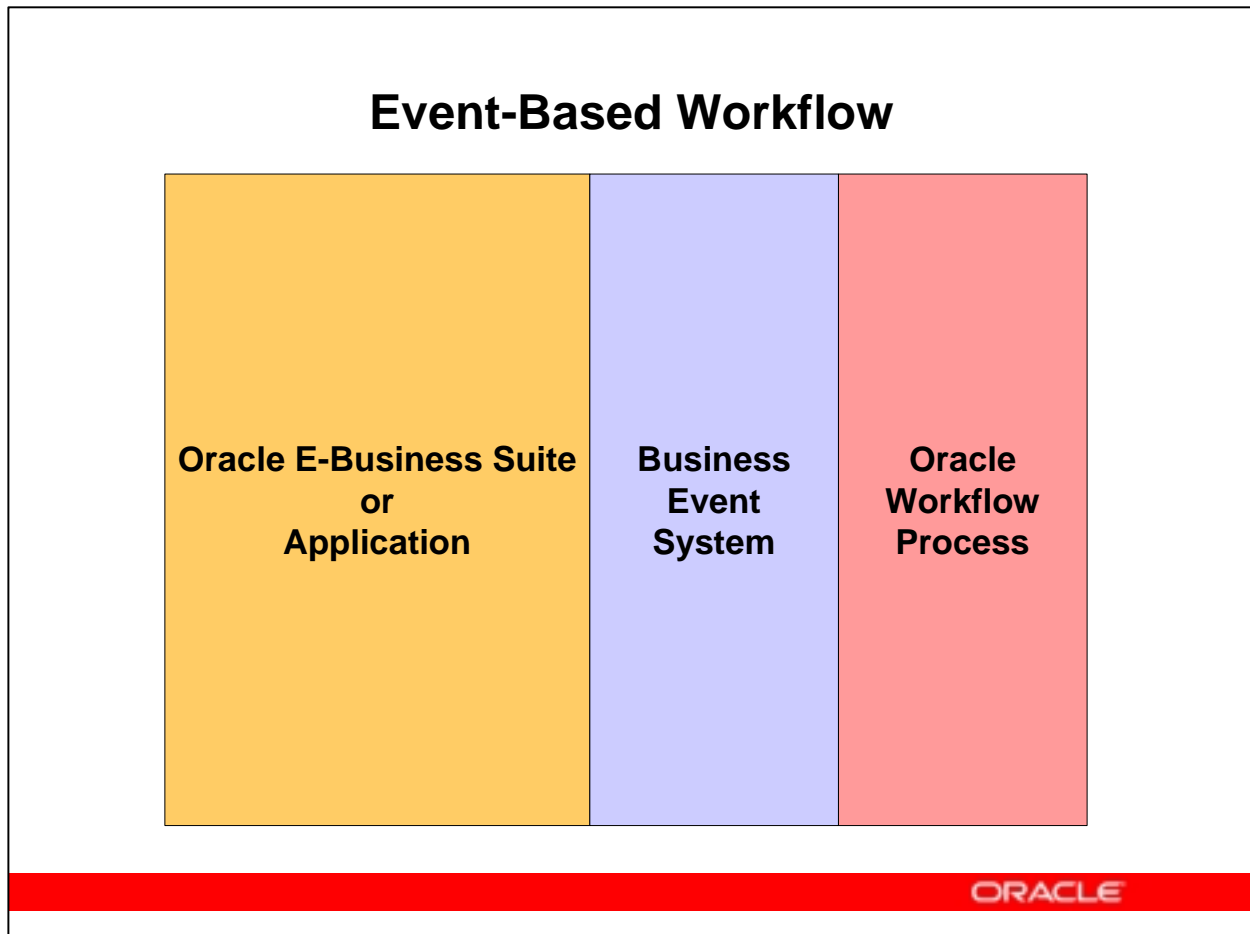
ORACLE

### Example: New Hire Processing

Oracle Workflow can help save time in new hire processing by automating approval hierarchies as well as automatically delivering notifications and reminders of work to be done.

## Event-Based Workflow

---



### Event-Based Workflow

With the Business Event System, Oracle Workflow supports both traditional applications-based workflows and event-based integration workflows.

For e-business, there is a requirement to integrate with external systems, such as sending a document to a business-to-business exchange, or other systems external to the local application. Oracle Workflow supports e-business integration workflows by allowing business analysts and developers to model business processes spanning different systems using a graphical drag-and-drop designer - the Workflow Builder - and run those processes using the Workflow Engine and the Business Event System. This support allows Oracle Workflow customers to deal with business objects in comprehensive e-business integration flows, with minimal intrusion into the core application.

The Business Event System and the Workflow Engine can function independently of each other. However, you can achieve the most powerful and flexible processing by using the Business Event System and the Workflow Engine together to execute cross-system processes for e-business integration.

### Subscription-Based Processing

#### In the Oracle Workflow Business Event System:

- **Business events in applications trigger event subscriptions in Oracle Workflow.**
- **Subscriptions can launch workflow processes or perform other processing.**
- **Multiple subscriptions can be defined to perform different processing for the same event.**
- **Subscriptions can be enabled, modified, or disabled as necessary without intruding into applications.**



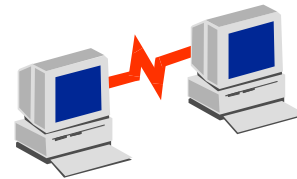
ORACLE

#### Subscription-Based Processing

The Business Event System provides increased flexibility through subscription-based processing: you raise a business event from an application, but specify the processing to perform for that event as a subscription in Oracle Workflow. For example, you can launch a workflow process when an event is raised by specifying that process in a subscription to the event. You can also define multiple subscriptions to the same event to perform additional processing for different purposes, without intruding any further on the core application.

### System Integration with Oracle Workflow

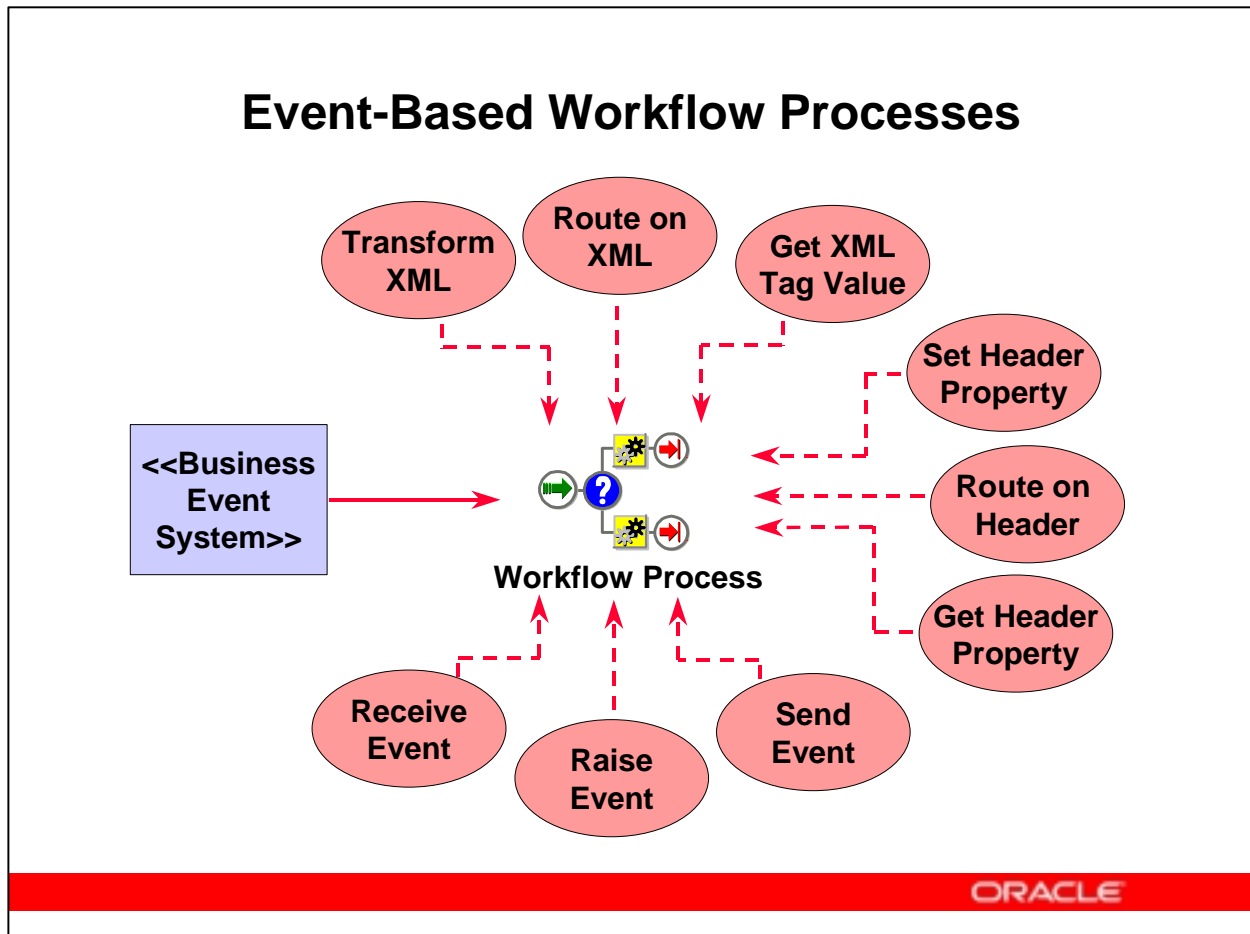
- **E-business accelerates the demand for system integration.**
- **Communication is required between systems both within and beyond the enterprise.**
- **Oracle Workflow supports e-business integration workflows through the Business Event System.**
- **Business event-based workflows allow modeling of cross-system processes, enabling business process-based integration.**



ORACLE



## Event-Based Workflow Processes



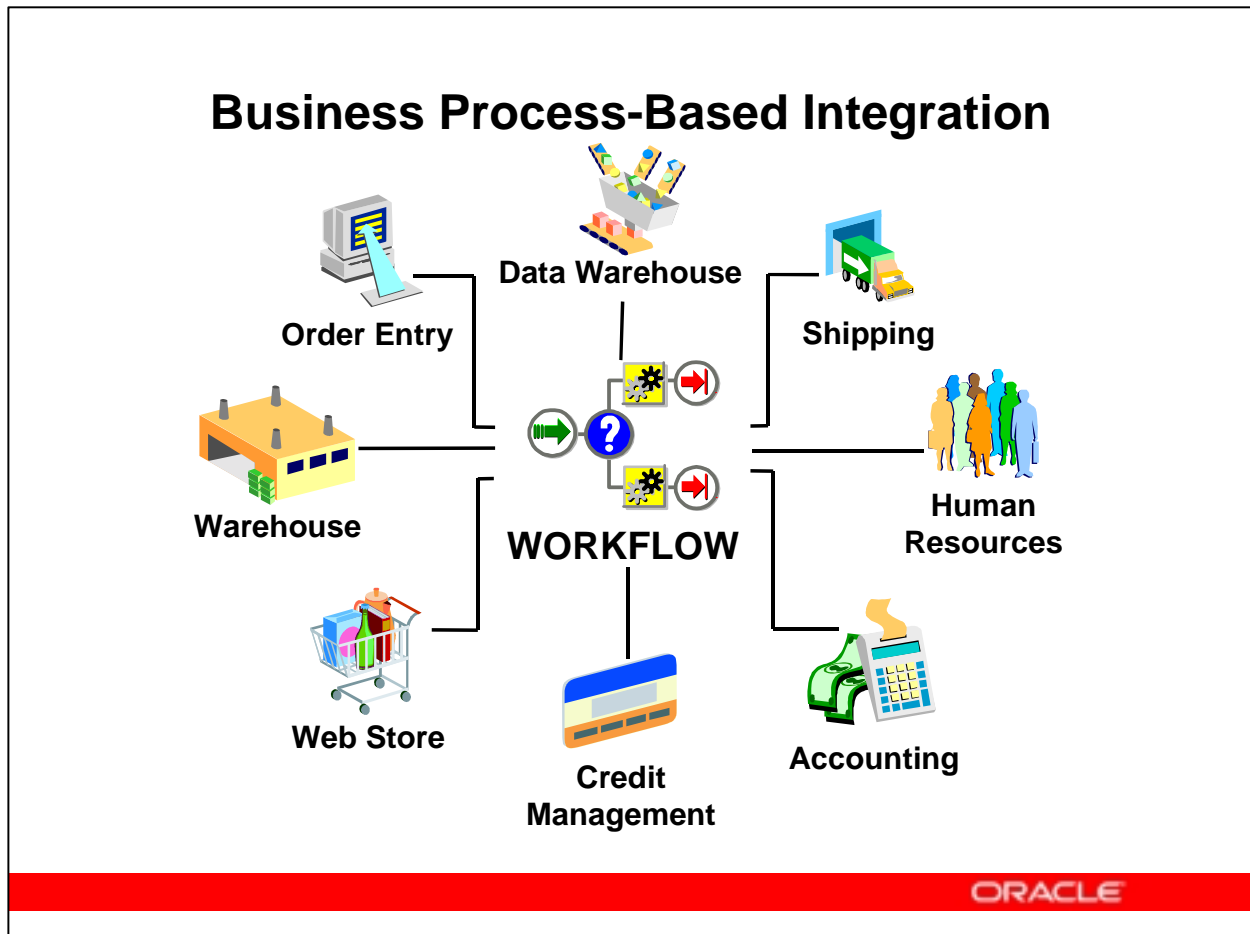
### Event-Based Workflow Processes

Event-based workflow processes control and route objects between applications according to business rules. These workflow processes support:

- Receiving business events to launch or continue processes
- Raising new business events
- Sending business event messages for inter-system communication
- Accessing and routing on header properties of event messages
- Accessing and routing on XML content within event messages  
(**Note:** This functionality is currently only available for the standalone version of Oracle Workflow. This functionality is not currently available for the version of Oracle Workflow embedded in Oracle E-Business Suite.)

By letting you model processes across different systems, event-based workflows enable business process-based integration.

## Business Process-Based Integration



### Business Process-Based Integration

Business process-based integration is model-driven.

- Business rules are expressed in a process model.
- These rules define the policy for each end-to-end process.
- The process model can encompass applications both within and beyond the enterprise.

Business process-based integration provides:

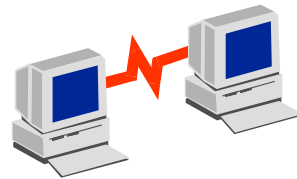
- A global, enterprise-level view of business objects
- Business process automation

Oracle Workflow supports business process-based integration through the Business Event System.

### Supported System Integration Types

The Business Event System supports integration in which applications are loosely coupled through asynchronous messaging.

- Point-to-point system integration—"Hardwired" communication between specified systems
- Messaging hub system integration—Intersystem communication routed through a central hub for more complex integration scenarios
- Distributed applications messaging—Master/copy replication of data for distributed applications



ORACLE

### Supported System Integration Types

The types of system integration supported by Oracle Workflow are message-based. By supporting the communication of messages between systems, Oracle Workflow lets you define processing across different systems encompassing both your own enterprise and your business partners. The power of this cross-system processing, together with the flexibility provided by subscription-based processing, enables you to use Oracle Workflow for e-business integration.

### Designing Applications for Change

- **Business processes change over time.**
- **Good design in an e-business environment requires:**
  - **Dynamic processes sympathetic to change**
  - **No artificial constraints on business processes**
  - **The ability to modify business processes without changing code**
  - **A visual overview of business processes**



ORACLE

### Designing Applications for Change

The benefits of good design include:

- Reduced cost of ownership through diminished development costs
- Ease of management and maintenance
- Visual documentation of business processes

### Designing Applications for Change

**Oracle Workflow helps you design applications for change using workflow processes, enabling continuous business process improvement.**

- **Complete process representation**
- **Graphical development tool**
- **Ease of management and maintenance**



ORACLE

### Designing Business Processes for Change

#### Complete process representation

- A workflow process can cross organizational and company boundaries to represent an end-to-end flow.
- You can provide alternatives within a process to accommodate different situations.
- You can build management metrics and performance goals into a process.

#### Graphical development tool

- The Oracle Workflow Builder separates business process definition and modeling from code development.
- Workflow diagrams provide a visual overview of your processes.
- You can easily modify a process definition.

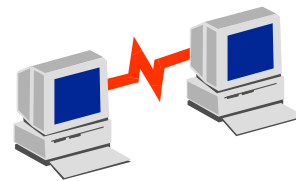
#### Ease of management and maintenance

- Oracle Workflow lets you analyze time and costs for entire business processes.
- You can refine your process definitions according to your analysis to streamline them and reduce time and costs.
- You can easily implement a modified process.

- Oracle Workflow empowers process participants by giving them access to review the progress and current status of their processes.
- Users can find the answers to many common questions themselves using these monitoring capabilities.

### Designing Applications for Integration

- **System integration is increasingly required for e-business.**
- **Good design in an e-business environment requires:**
  - **Availability of integration points in applications for immediate or future use**
  - **Noninvasive configuration**
  - **The ability to modify integration processing without changing code**



ORACLE

### Designing Applications for Integration

**Oracle Workflow helps you design applications for integration using the Business Event System, enabling business process-based integration.**

- **Business events as integration points**
- **Subscription-based processing**
- **Web-based business event and subscription management tool**



ORACLE

### Designing Applications for Integration

- Business events as integration points
  - Applications can raise business events at any point where further processing or integration might be required.
  - You can define the processing to be triggered by an event immediately after defining the event or at any later time.
- Subscription-based processing
  - Use subscriptions to specify the processing you want to perform for a business event.
  - You can define multiple subscriptions to the same event to perform additional processing for different purposes.
- Web-based business event and subscription management tool
  - The Event Manager separates event subscription definition from code development.
  - You can use the Event Manager web pages to define, update, or delete event subscriptions without intruding on the core application.



### Oracle Workflow Availability

**Oracle Workflow is available in two versions:**

- **Standalone**
  - With the Oracle Database Server (both Standard Edition and Enterprise Edition)
  - With the Oracle9i Application Server
- **Embedded in Oracle E-Business Suite**
  - Self-service applications
  - Professional applications



ORACLE

### Oracle Workflow Availability

A standalone version of Oracle Workflow is available with the Oracle database and Oracle9i Application Server. The standalone version of Oracle Workflow is leveraged by products such as JDeveloper, Oracle Warehouse Builder, and Oracle9iAS InterConnect, and is embedded in third party products.

Oracle Workflow is also available embedded in the Oracle E-Business Suite to enforce a common set of business rules. In the Oracle E-Business Suite, Oracle Workflow is incorporated in applications including Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Human Resources Management Systems (HRMS). Oracle Workflow is leveraged by both professional applications, which are typically Forms-based applications for power users, and self-service applications, which are typically HTML-based applications for more casual users.

### Summary

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

- **Explain the benefits of Oracle Workflow.**
- **Discuss the concept of a workflow process.**
- **Discuss the concept of business events.**

ORACLE

# Oracle Workflow Components

## Chapter 20

## Oracle Workflow Components

ORACLE

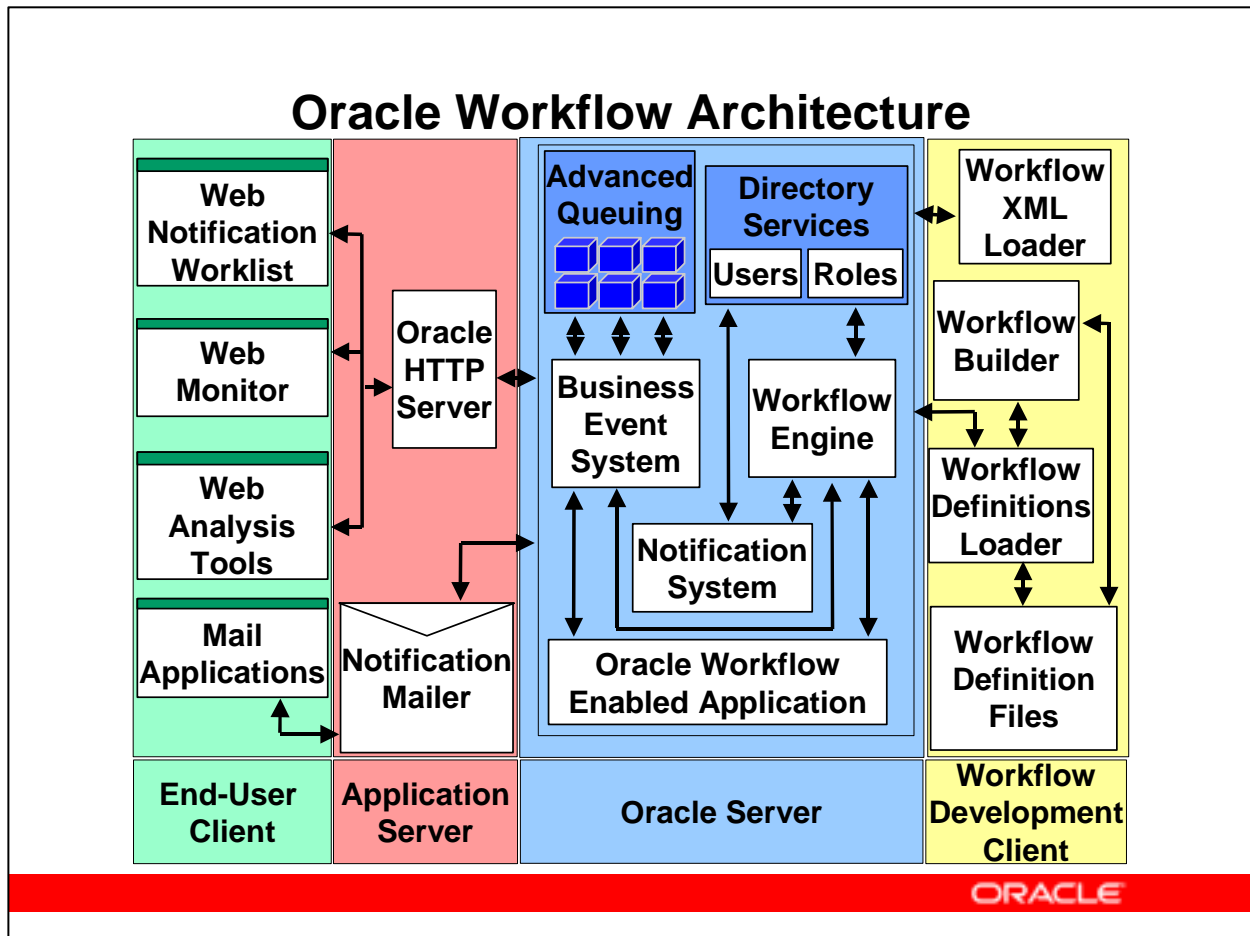
### Objectives

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

- **Describe the architecture and components of Oracle Workflow.**
- **Discuss how the Business Event System communicates events between systems.**
- **Describe how the Workflow Engine executes workflow processes.**

ORACLE

# Oracle Workflow Architecture



## Oracle Workflow Architecture

### Workflow Development Client

The development client is a PC running Windows 95, Windows 98, Windows 2000, or Windows NT 4.0 or higher. This platform is used to create and modify Oracle Workflow process definitions.

### Oracle Server

The Oracle Server platform is the Oracle RDBMS. This platform hosts the business application integrated with Oracle Workflow, the Workflow Engine, Business Event System, Notification System, and directory services.

### Application Server

The application server is the environment outside of the RDBMS. This environment includes ancillary services such as the Oracle9i Application Server (Oracle9iAS) as the web server, and the Notification Mailer.

### End-User Client

The end-user client is the workstation or PC that an end user uses to perform daily tasks. This client includes browser support for reviewing and responding to notifications in the

Notification Worklist, the Oracle Workflow Monitor, and web analysis tools, as well as mail applications for reviewing and responding to notifications by e-mail.

### Oracle Workflow Components

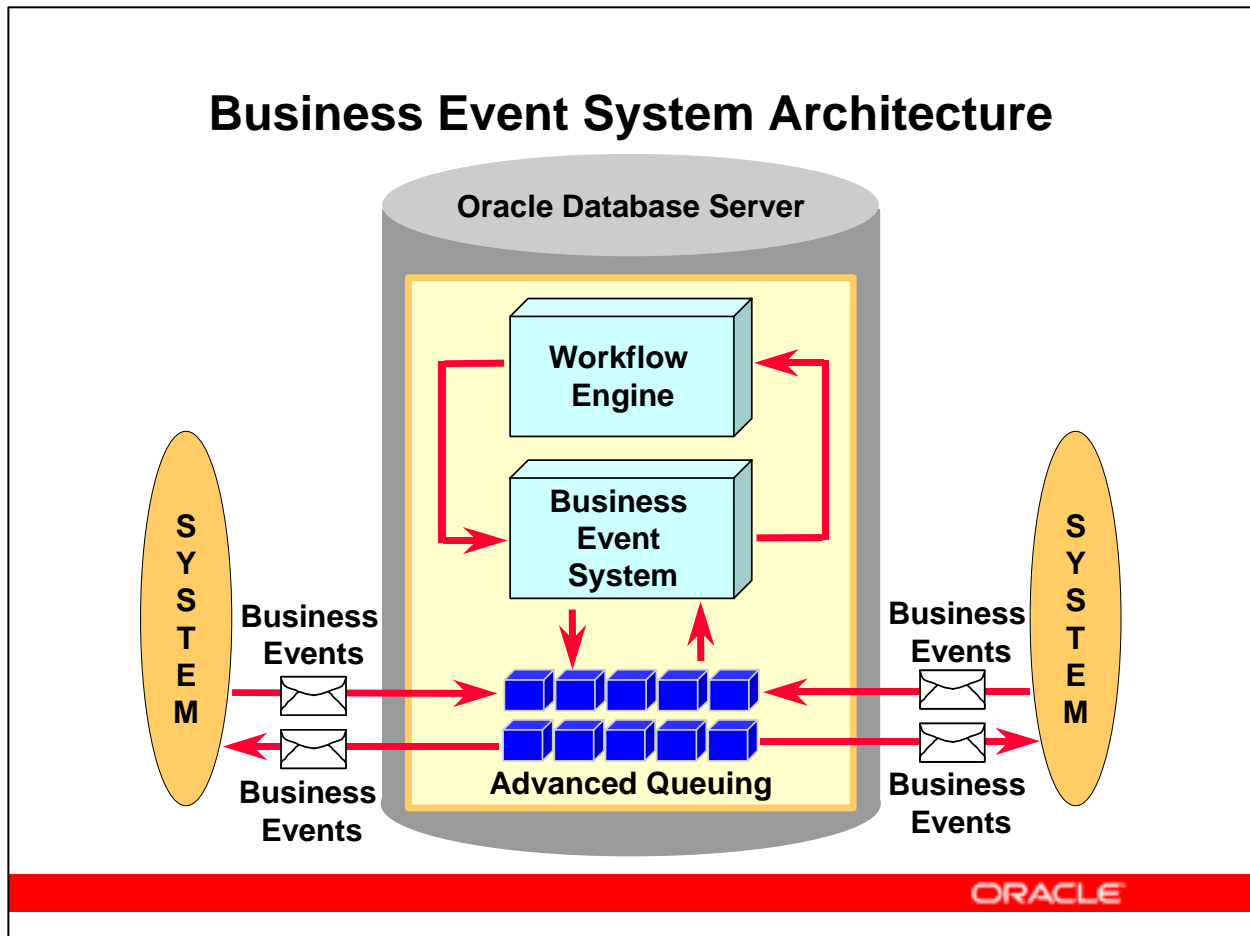
- **Business Event System**
- **Workflow Engine**
- **Oracle Workflow Builder**
- **Notification System**
- **Notification Worklist**
- **Directory Services**
- **Workflow Monitor**
- **Workflow Definitions Loader**
- **Workflow XML Loader**



ORACLE



## Business Event System Architecture



### Business Event System Architecture

The Business Event System is an application service that communicates business events between systems. Oracle Workflow with the Business Event System can act as a system integration messaging hub that relays business event messages among systems.

The Business Event System leverages Oracle Advanced Queuing (AQ) to send messages from one system to another.

For the greatest flexibility in routing and processing business events, you can model your business process logic in powerful cross-system workflow processes that are executed by the Workflow Engine. However, the Business Event System can also function independently of the Workflow Engine.

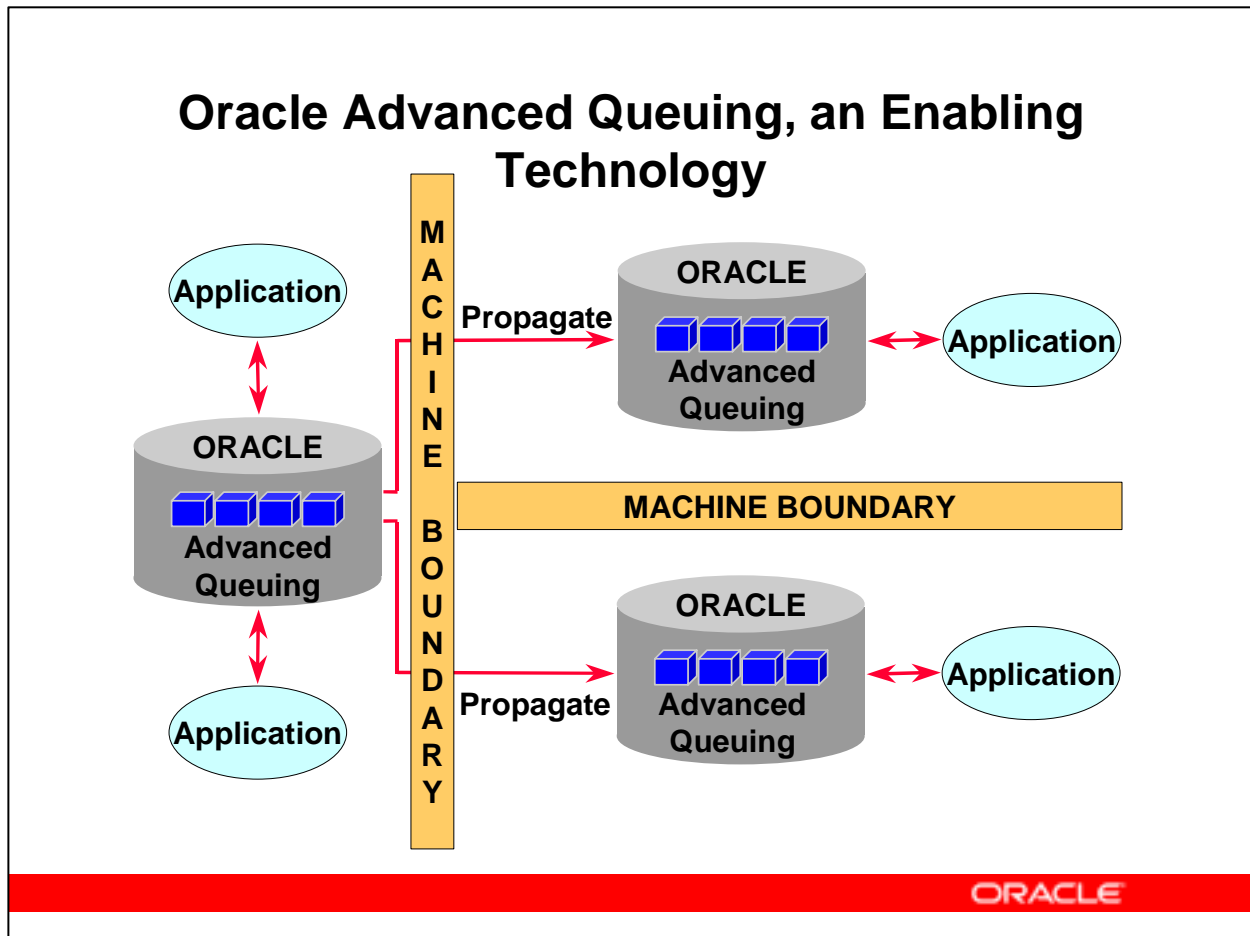
### **Business Event System Components**

**The Business Event System includes:**

- **The Event Manager—Lets you register:**
  - **Business events**
  - **Systems**
  - **Named communication agents within systems**
  - **Subscriptions to events that are significant to your systems**
- **Workflow Engine event activities—**  
**Let you model business events within workflow processes**



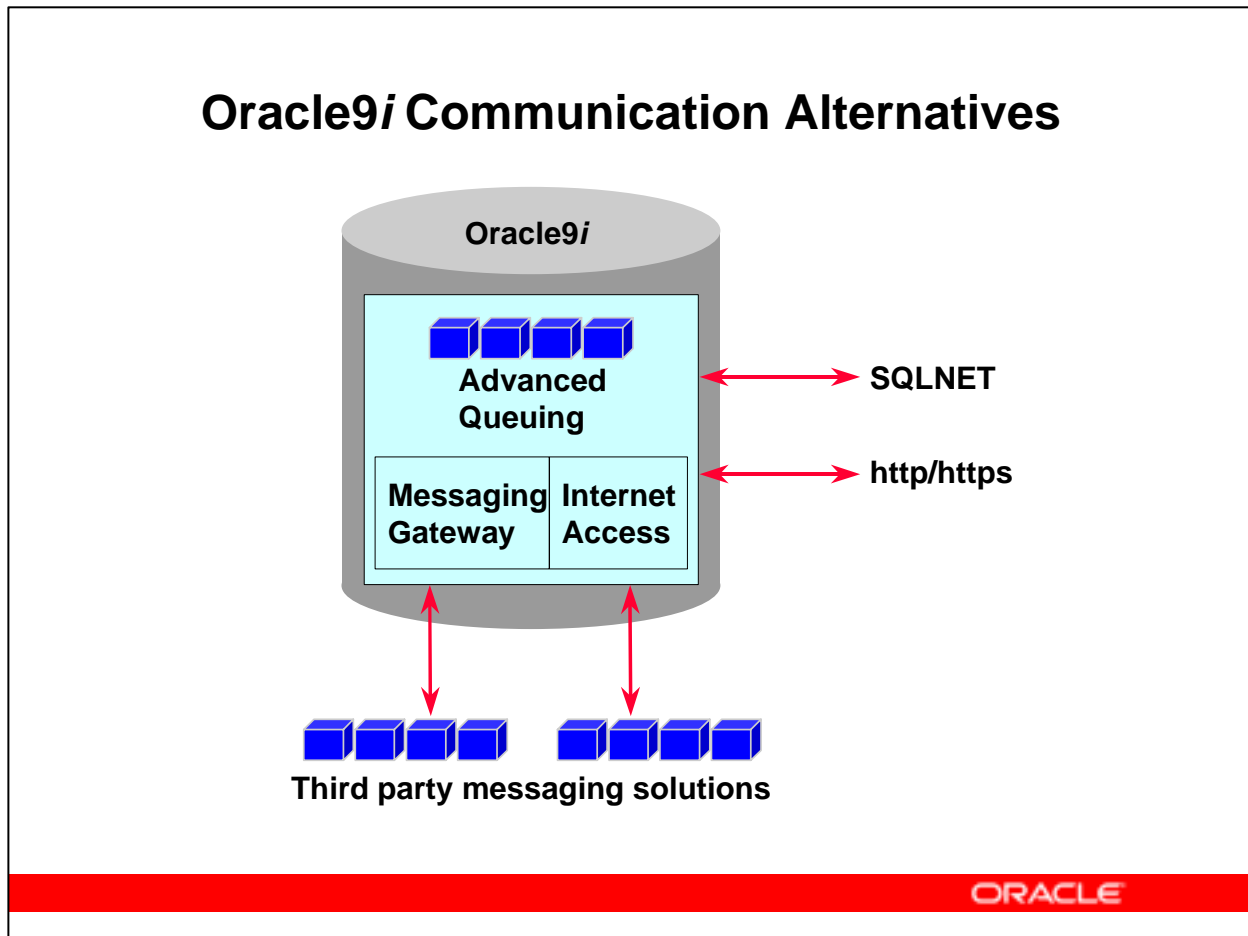
**ORACLE**



## Oracle Advanced Queuing, an Enabling Technology

Oracle Advanced Queuing (AQ) allows queue-to-queue propagation across machine boundaries. AQ is a feature of Oracle9i that provides database-integrated message queuing functionality, leveraging the functions of the Oracle database to store messages persistently and transmit them using various transport protocols.

For more information, refer to the *Oracle9i Application Developer's Guide - Advanced Queuing*.



### Oracle9i Communication Alternatives

In Oracle9i, you can use Oracle Advanced Queuing (AQ) for communication by SQLNET, HTTP, and HTTPS protocols, and for integration with third party messaging solutions.

You can use Oracle Net Services (formerly Net8) to propagate messages by the SQLNET protocol. Oracle Advanced Queuing's Internet access functionality lets you perform AQ operations over the Internet by using AQ's Internet Data Access Presentation (IDAP) for messages and transmitting the messages over the Internet using transport protocols such as HTTP or HTTPS.

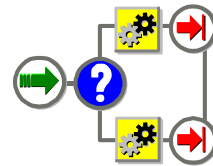
Messaging Gateway is a feature of AQ that enables communication between applications based on non-Oracle messaging systems and AQ. Standard AQ functionality provides propagation between two AQ queues; Messaging Gateway extends that propagation to legacy applications based on non-Oracle messaging systems.

For more information, refer to the *Oracle9i Application Developer's Guide - Advanced Queuing*.

### Workflow Engine

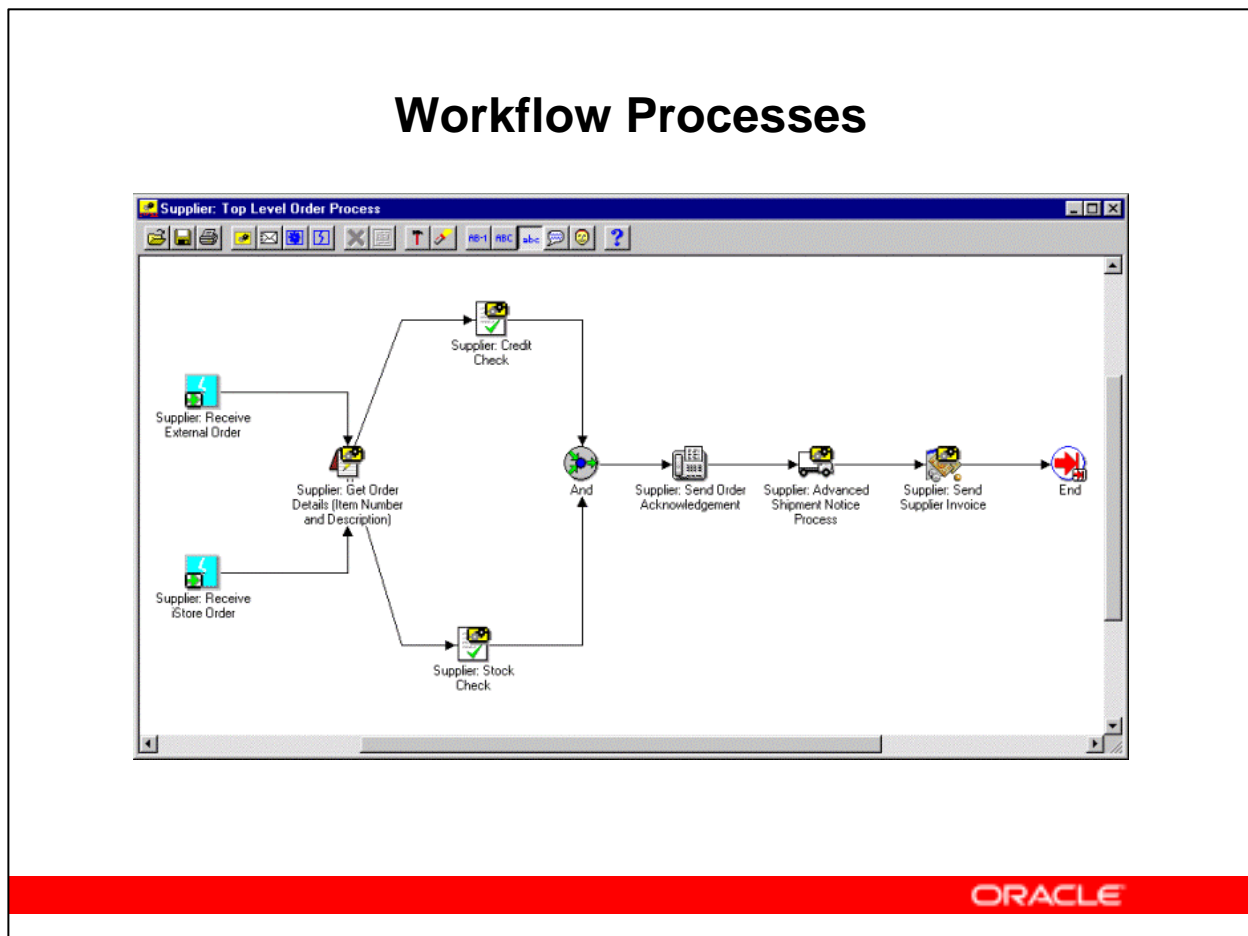
#### The Workflow Engine:

- Is embedded in the Oracle database server
- Uses the process definition created with Oracle Workflow Builder to coordinate the routing of activities for the process
- Monitors the state of each activity in a workflow process
- Signals any changes in the workflow state using calls to PL/SQL or Java APIs
- Guarantees consistency between the application and the workflow state because of Oracle Database Server transactional integrity



ORACLE

## Workflow Processes



### Workflow Processes

- A workflow process definition must be saved to the same database as the Workflow Engine.
- A process definition is composed of activities and the transitions between them.
- A completed application transaction or event can initiate a workflow process by raising an event or by calling a series of Workflow Engine APIs.
- The Workflow Engine locates the ‘Start’ activity in the process definition.
- The Workflow Engine drives through the process, performing all automated steps such as function activities and Raise and Send event activities, until an asynchronous activity such as a notification, Receive event activity, or blocking activity occurs.
  - The Workflow Engine calls the Notification System to deliver a notification message to an appropriate role. Once a user of that role completes the notification response, the Workflow Engine continues to drive through the remaining activities in the process.
  - If a blocking activity is encountered, the Workflow Engine waits for an external program to complete and call the appropriate Workflow Engine API before proceeding to the next activity.

- If a Receive event activity is encountered, the Workflow Engine waits to receive the event from the Business Event System before proceeding to the next activity.
- The process completes when the Workflow Engine encounters an 'End' activity.

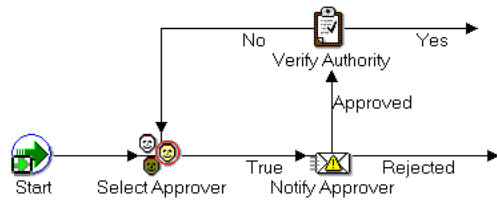
**Example: Order Processing**

This example shows a workflow process that includes business events. The example comes from the Event System Demonstration workflow which is available with the standalone version of Oracle Workflow.

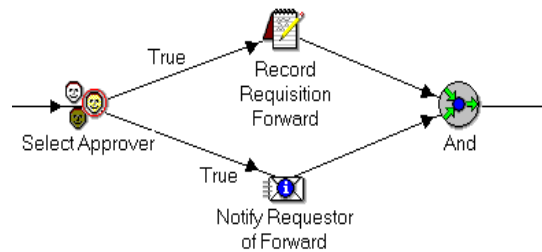
## Supported Process Constructs

### Supported Process Constructs

- **Looping**
- **Results-based branching**



- **Parallel flows**
- **Rendezvous**



ORACLE

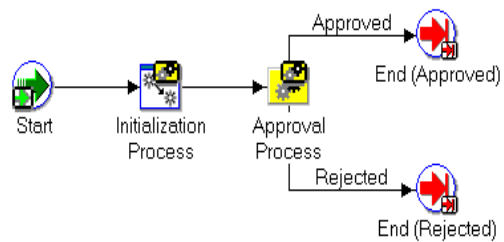
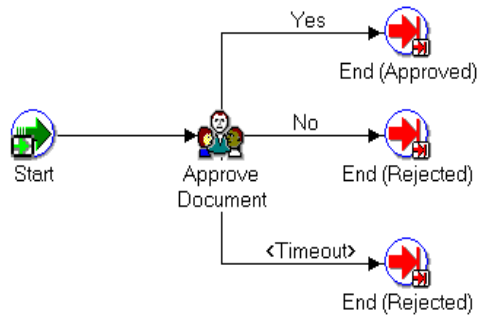
### Supported Process Constructs

The Workflow Engine supports sophisticated workflow rules to model your business logic.



## Supported Process Constructs

- **Voting**
- **Timeouts**
  - Escalation
  - Automatic forwarding
- **Subprocesses (unlimited hierarchy)**



ORACLE

### Oracle Workflow Builder

- **Oracle Workflow Builder is the development tool for Oracle Workflow.**
  - Lets you graphically define and customize workflow definitions
  - Requires a PC running Windows 95, 98, 2000 or Windows NT
- **You can save workflow definitions to a database or a flat file.**
- **Oracle Workflow Builder consists of two parts:**
  - Navigator Tree
  - Process Diagram



ORACLE

### Oracle Workflow Builder

Saving workflow definitions as flat files on the local file system enables designers to back up their work and use source control.

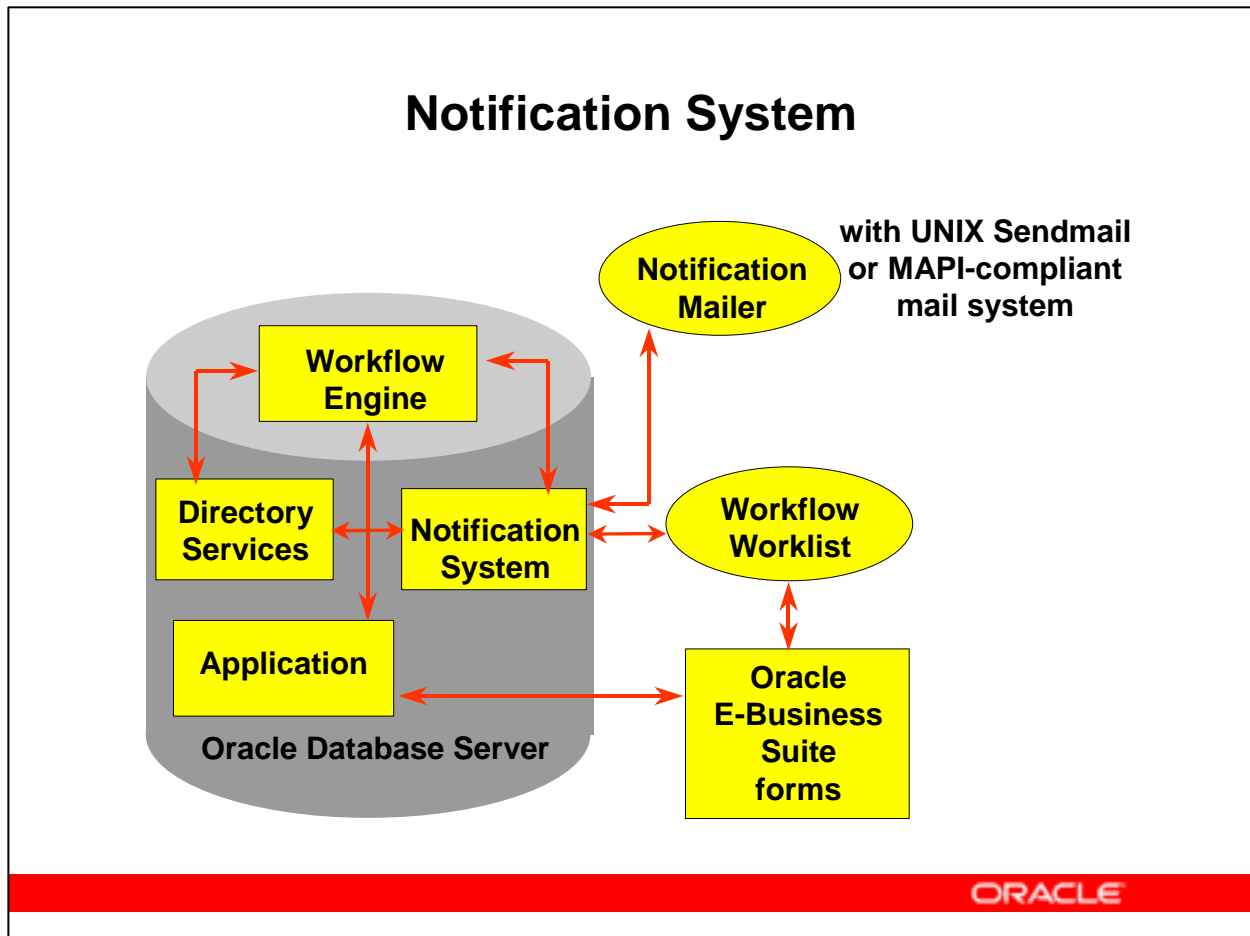
#### Navigator Tree

The navigator provides a tree structure for the workflow definition, with the highest level being the data store. Next is the item type, which is a grouping of workflow objects into a high level category. The lower levels are the workflow objects themselves, such as attributes, processes, notifications, functions, events, messages, and lookup types. All these objects are organized into their respective categories.

#### Process Diagram

The diagram is made up of icons representing workflow objects. A diagram is built by dragging the objects from the navigator window and dropping them into the process diagram window. This method is called bottom-up design. You can also create new objects as you design the diagram and complete the definitions of those objects later. This method is called top-down design.

# Notification System



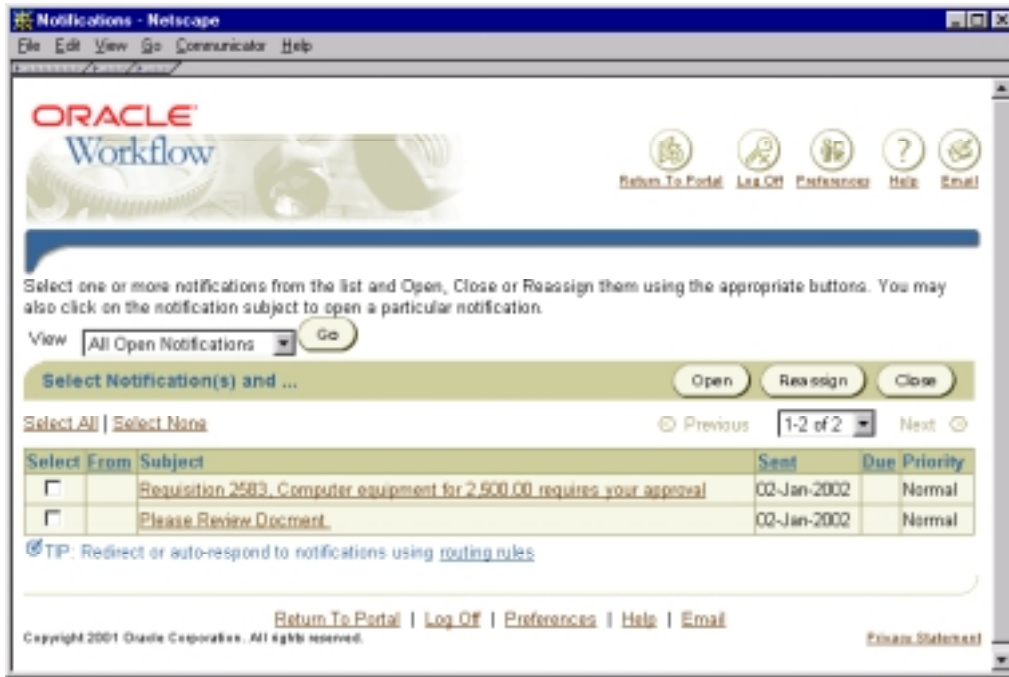
## Notification System

The Notification System:

- Routes notifications to a role, which can be a single user or group of users
- Enables users to receive and respond to notifications using an e-mail application or Web browser of choice
- Allows any users with access to the Internet to be included in a workflow process
- Provides access to the Notification Worklist from Oracle E-Business Suite
- Enables users to drill down to any URL or Oracle E-Business Suite form to respond to a notification from the Notification Worklist

## Notification Worklist Web Page

### Notification Worklist Web Page



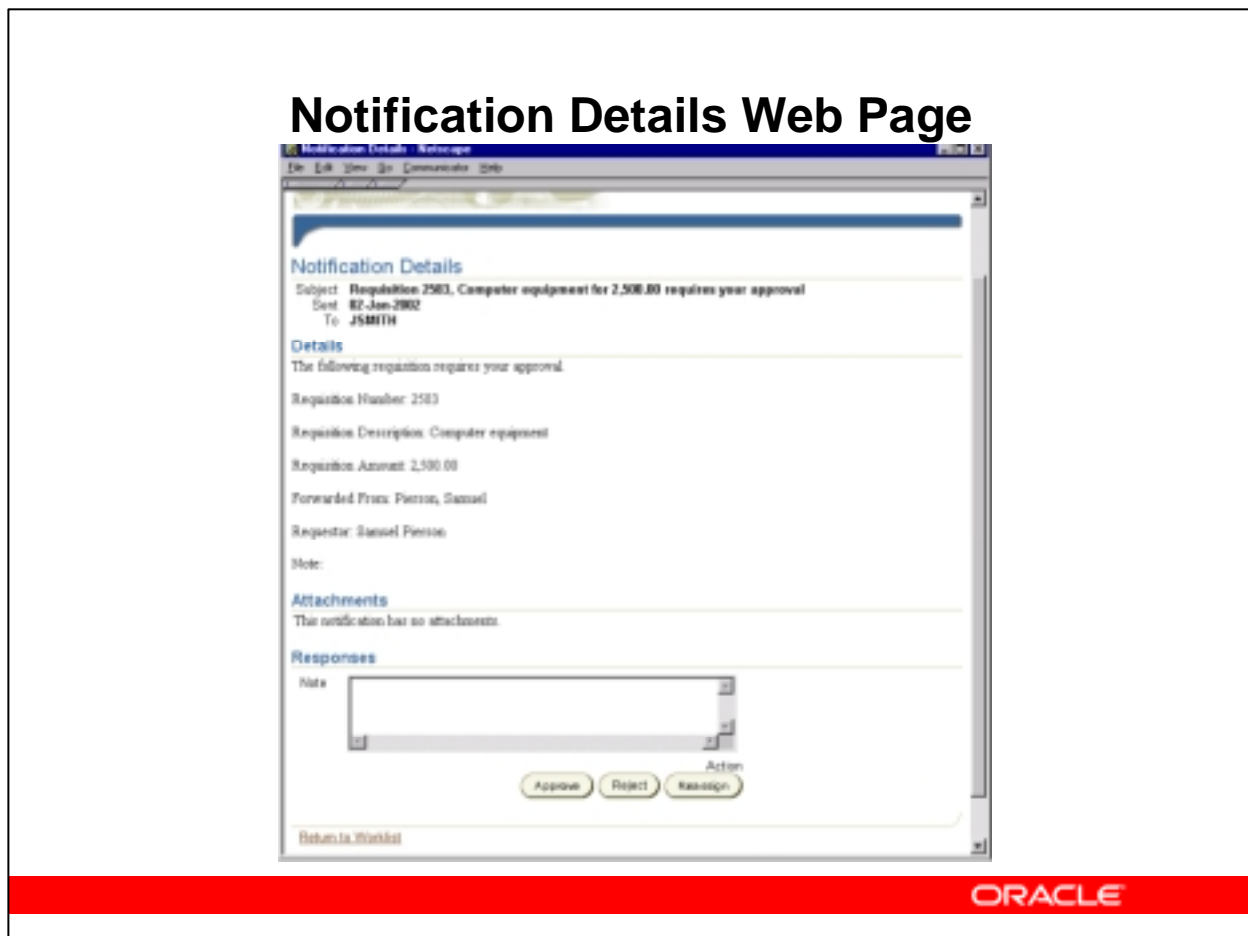
### Notification Worklist

The Notification Worklist is a feature that provides a list of open notifications for a particular user. The Worklist is viewable through the Oracle Workflow Web Agent. You can also configure the Notification Mailer to send an e-mail summarizing the outstanding notifications. From the Worklist, a user can view the notifications as well as respond to those that require a response.

### New Self-Service Web Page Format

The Oracle Workflow web pages are being converted to the new format for Oracle self-service web applications. Depending on your version of Oracle Workflow and which patches you have applied, you may see Oracle Workflow web pages in the previous format or in the new format. Currently, the Notifications Worklist and the Notification Rules web pages are available in the new format for the version of Oracle Workflow embedded in Oracle E-Business Suite.

## Notification Details Web Page



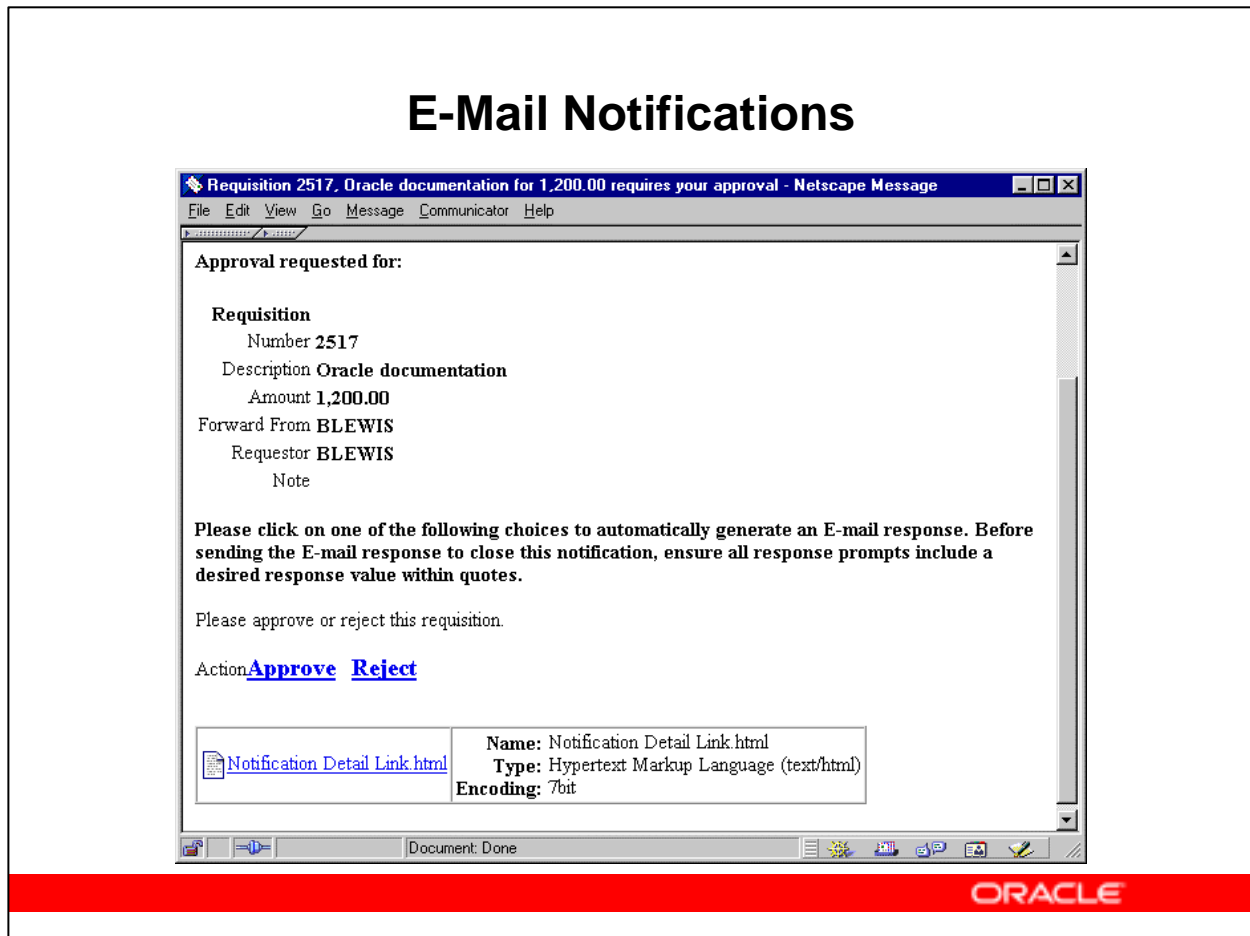
### Viewing Notifications from a Web Page

The Notification System has an Oracle Web Agent interface that dynamically generates each notification for a user.

To view notifications in the Notification Details web page, you must have Oracle HTTP Server installed as the web server for Oracle Workflow, and you must have a Web browser that supports Frames and JavaScript.

The Notification Details web page can display extended HTML message formats and can include links to URLs or Oracle E-Business Suite forms that let users research and make decisions through online inquiry.

## E-Mail Notifications



### E-Mail Notifications

- The Notification System interfaces with the Notification Mailer program to send e-mail notifications to users and roles. Users can reply to e-mail notifications using their e-mail client.
- There are two versions of the Notification Mailer program to integrate with UNIX Sendmail or Windows NT MAPI-compliant mail applications.
- E-mail notifications can be delivered to users on other mail systems if the appropriate UNIX gateway software is also installed.

### Directory Services

**The directory service for Oracle Workflow is implemented as a set of views that are mapped across the user tables of the underlying application.**

- **WF\_USERS:** Contains information on user names, display names, notification preferences and e-mail addresses
- **WF\_ROLES:** Contains information on the roles of which users can be members
- **WF\_USER\_ROLES:** Contains information on the association of users with roles

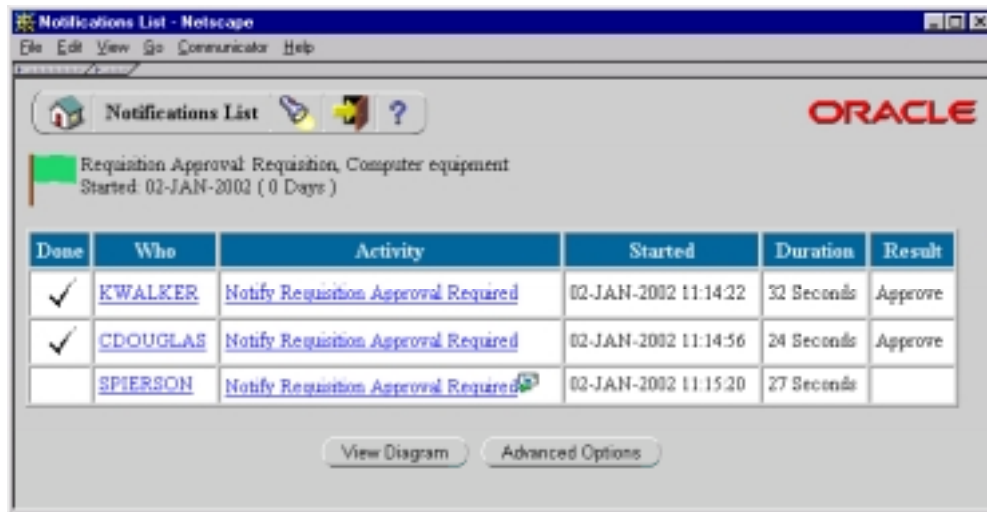


ORACLE

### Directory Services

Users can be associated with more than one role, and a role may contain more than one user. The Workflow Engine and Notification System use the directory service to determine who should receive notifications and in what format. Notifications can be delivered to an individual user or to all members of a particular role.

## Oracle Workflow Monitor



Notifications List - Netscape

Requestion Approval Requisition, Computer equipment  
Started: 02-JAN-2002 ( 0 Days )

Done	Who	Activity	Started	Duration	Result
✓	<a href="#">KWALKER</a>	<a href="#">Notify Requisition Approval Required</a>	02-JAN-2002 11:14:22	32 Seconds	Approve
✓	<a href="#">CDOUGLAS</a>	<a href="#">Notify Requisition Approval Required</a>	02-JAN-2002 11:14:56	24 Seconds	Approve
	<a href="#">SPIERSON</a>	<a href="#">Notify Requisition Approval Required</a>	02-JAN-2002 11:15:20	27 Seconds	

View Diagram    Advanced Options

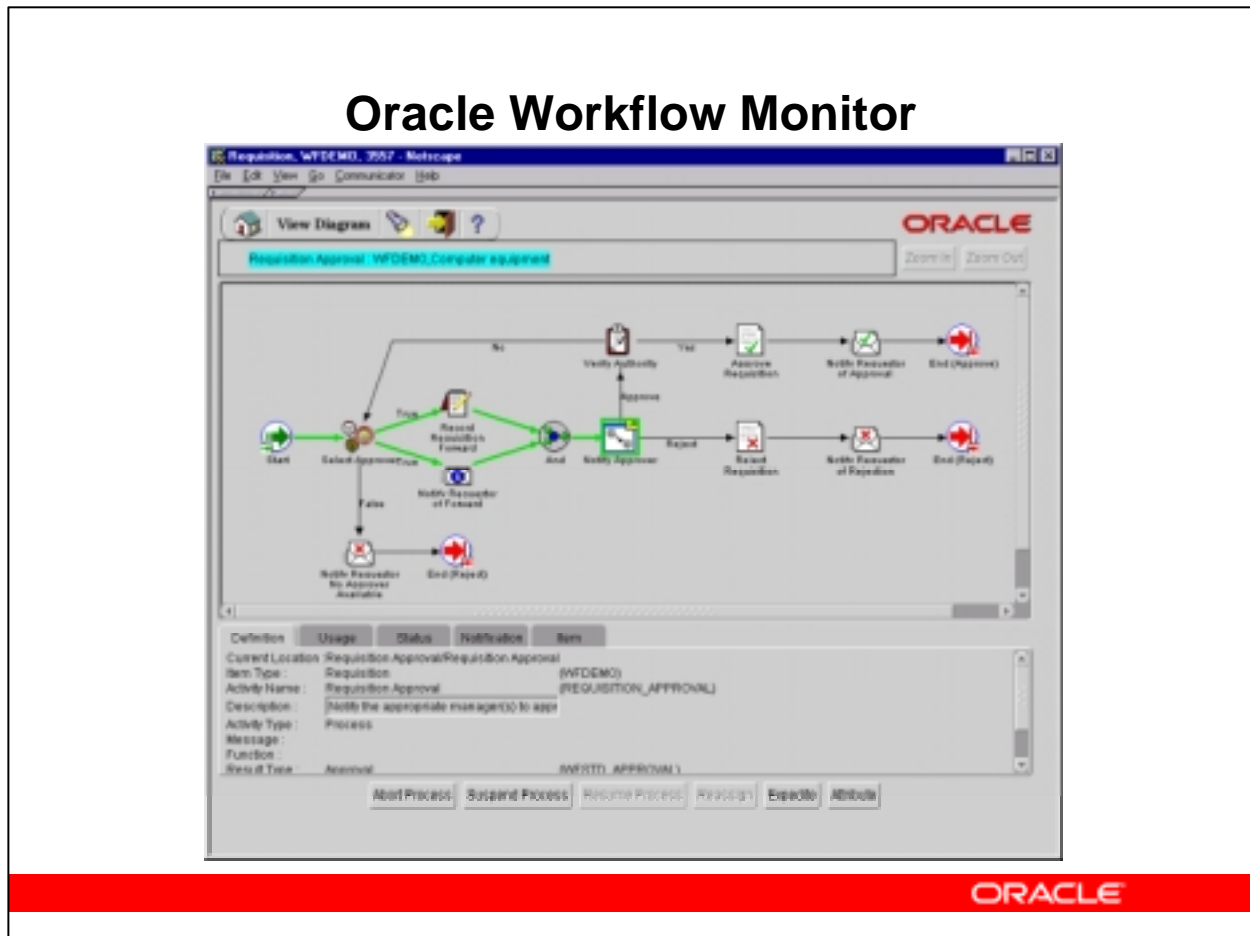
### Workflow Monitor

The Workflow Monitor Notifications List:

- Displays all the current notifications that have been sent for a process that require a response
- Shows the decision makers in the process as well as the current owner of any outstanding notifications

**Note:** The Find Processes web page lets you search for the process instance you want to view in the Workflow Monitor Notifications List.





## Workflow Monitor

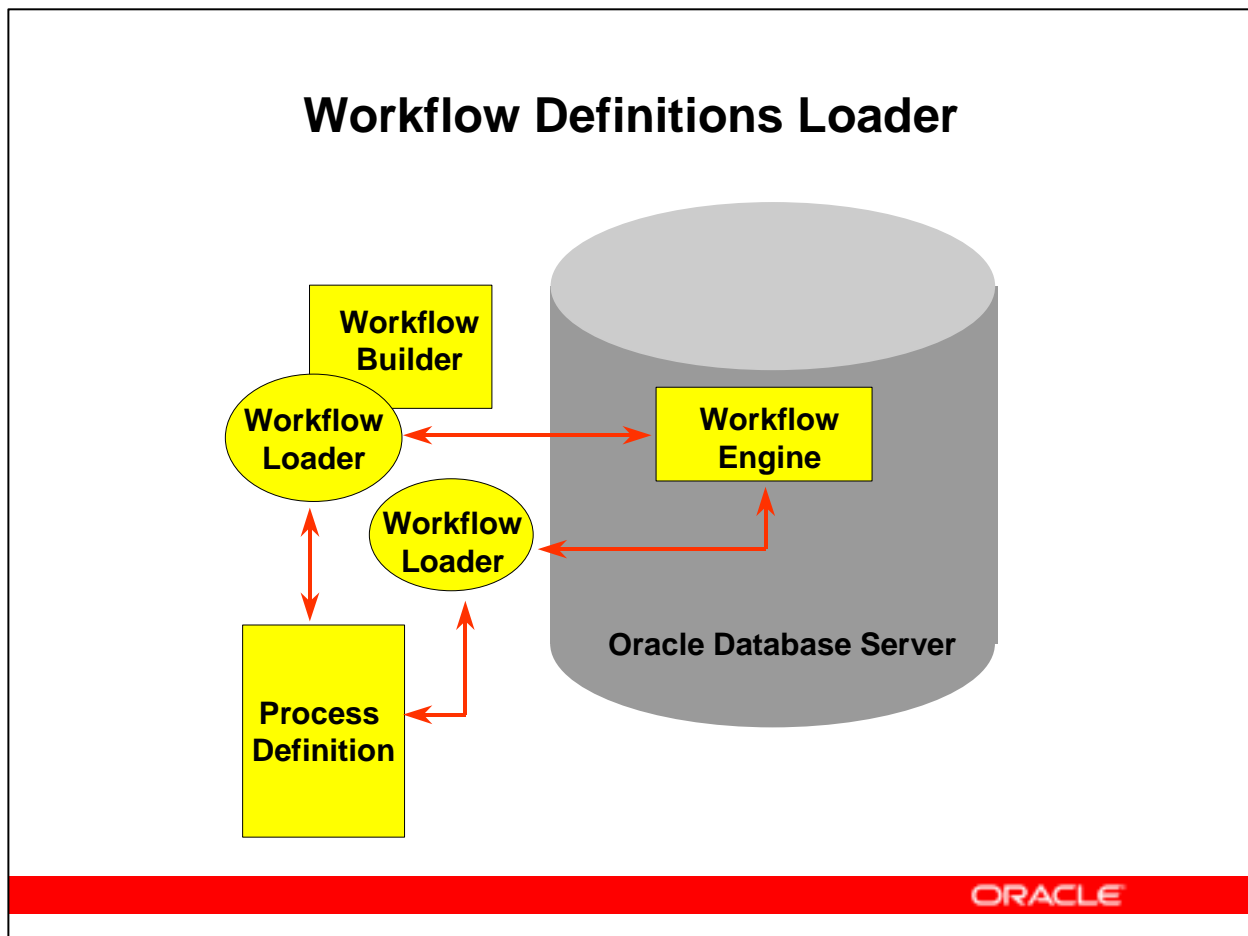
The Workflow Monitor diagram is a Java applet that displays status information for a single workflow process instance. The Workflow Monitor diagram:

- Graphically depicts the status of a workflow process instance in its upper frame
- Displays detailed information about individual activities, as well as about the process as a whole, in the lower tabbed frame
- Runs in USER mode to display end user information or ADMIN mode to display administrator information and functionality

**Note:** The Find Processes web page lets you search for the process instance you want to view in the Workflow Monitor.

To view a process in the Workflow Monitor, you must have Oracle HTTP Server installed as the web server for Oracle Workflow, and you must have a web browser that supports Java Development Kit (JDK) Version 1.1.8 or higher and Abstract Windowing Toolkit (AWT), such as Netscape Communicator version 4.76 or a higher version of 4.7x, or Microsoft Internet Explorer version 5.0x or 5.5.

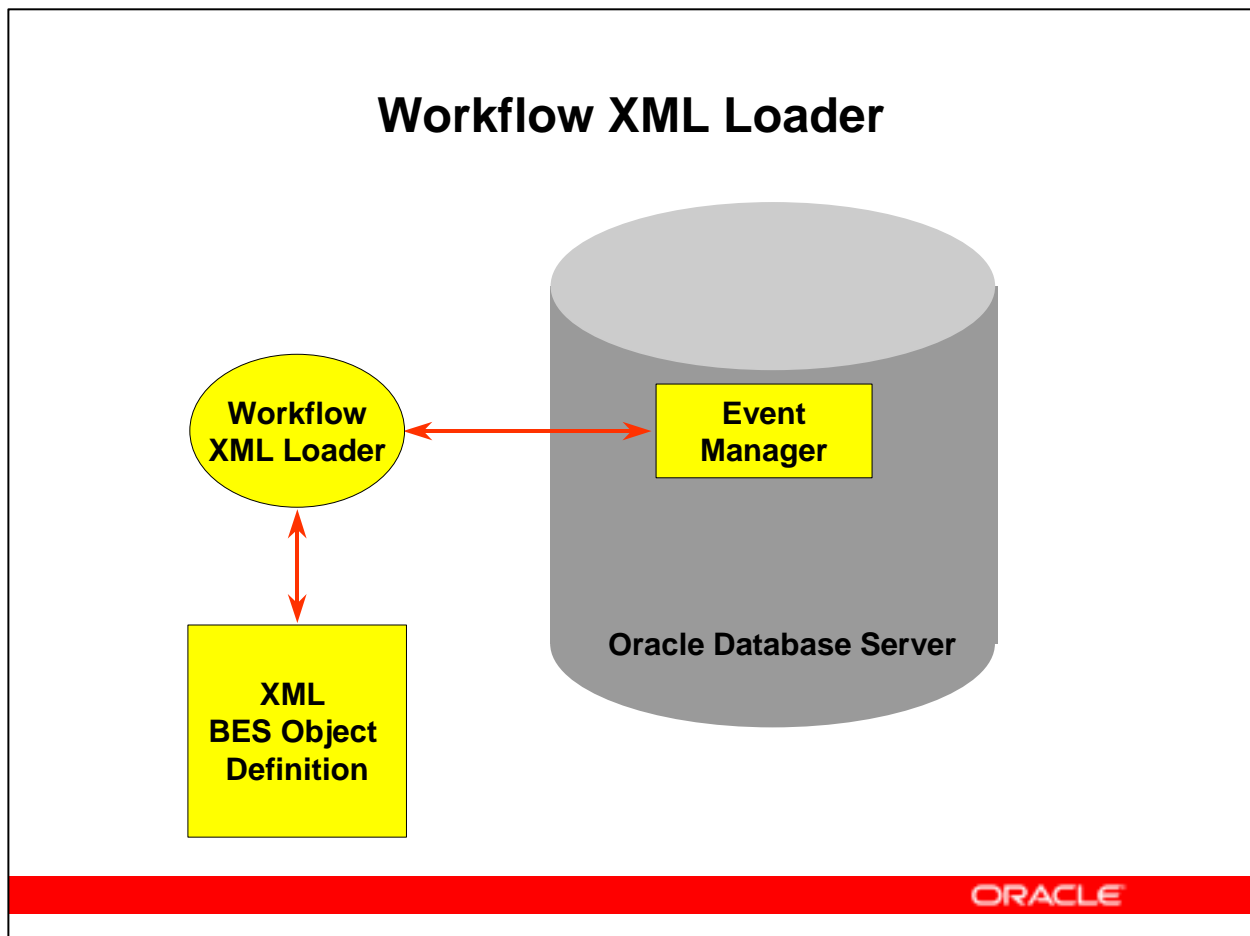
## Workflow Definitions Loader



### Loading Workflow Definitions

The Workflow Definitions Loader is a utility program that lets you transfer process definitions between a database and a flat file. The Workflow Definitions Loader:

- Runs on the server machine
- Lets you upgrade a database with new versions of process definitions or upload existing process definitions after a database upgrade
- Is also integrated into Oracle Workflow Builder
- Allows process definitions to be source-controlled as flat files



### Workflow XML Loader

The Workflow XML Loader is a utility program that lets you transfer XML definitions for Business Event System objects between a database and a flat file.

- When you download Business Event System object definitions from a database, Oracle Workflow saves the definitions as an XML file.
- When you upload object definitions to a database, Oracle Workflow loads the definitions from the source XML file into the Business Event System tables in the database, creating new definitions or updating existing definitions as necessary.

**Refer to Oracle Workflow Demo [DEM00004]**

### Summary

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

- **Describe the architecture and components of Oracle Workflow.**
- **Discuss how the Business Event System communicates events between systems.**
- **Describe how the Workflow Engine executes workflow processes.**

ORACLE

# Viewing and Responding to Notifications

## Chapter 21

# Viewing and Responding to Notifications

ORACLE

### Objectives

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

- **Access the Oracle Workflow web pages.**
- **Respond to notifications from the Notifications Worklist web page.**
- **Respond to notifications using e-mail.**
- **Define notification routing rules.**

ORACLE

## Accessing Oracle Workflow Web Pages

### Accessing Oracle Workflow Web Pages



### Accessing the Oracle Workflow Web Pages

#### Standalone Oracle Workflow

If you are using the standalone version of Oracle Workflow, use the Oracle Workflow home page to access Oracle Workflow's web-based features.

To access the standalone Oracle Workflow home page, use a web browser to connect to the following URL:

`<webagent>/wfa_html.home`

Replace `<webagent>` with the base URL of the web agent configured for Oracle Workflow in your Web server.

#### Embedded Oracle Workflow

If you are using the version of Oracle Workflow embedded in Oracle E-Business Suite, use a Workflow responsibility to access Oracle Workflow's web-based features.

#### New Self-Service Web Page Format

The Oracle Workflow web pages are being converted to the new format for Oracle self-service web applications. Depending on your version of Oracle Workflow and which patches you have applied, you may see Oracle Workflow web pages in the previous format or in the new format.



Currently, the Notifications Worklist and the Notification Rules web pages are available in the new format for the version of Oracle Workflow embedded in Oracle E-Business Suite.

### Viewing Notifications from a Web Browser

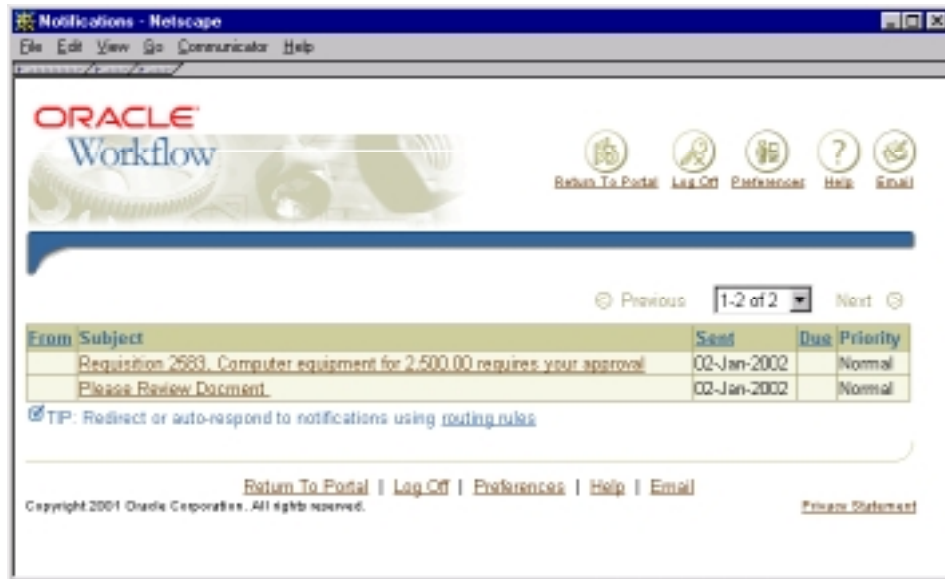
- **The Oracle Workflow Notifications web pages let you view and respond to your notifications.**
  - Find Notifications page
  - Notifications Worklist page
  - Notification Details page
- **All users can view their notifications in these web pages, regardless of their notification preference.**
- **If your notification preference is set to 'Do not send me mail' in the User Preferences web page, then you can only access your notifications through the Notifications web pages.**



ORACLE

## Simple Worklist

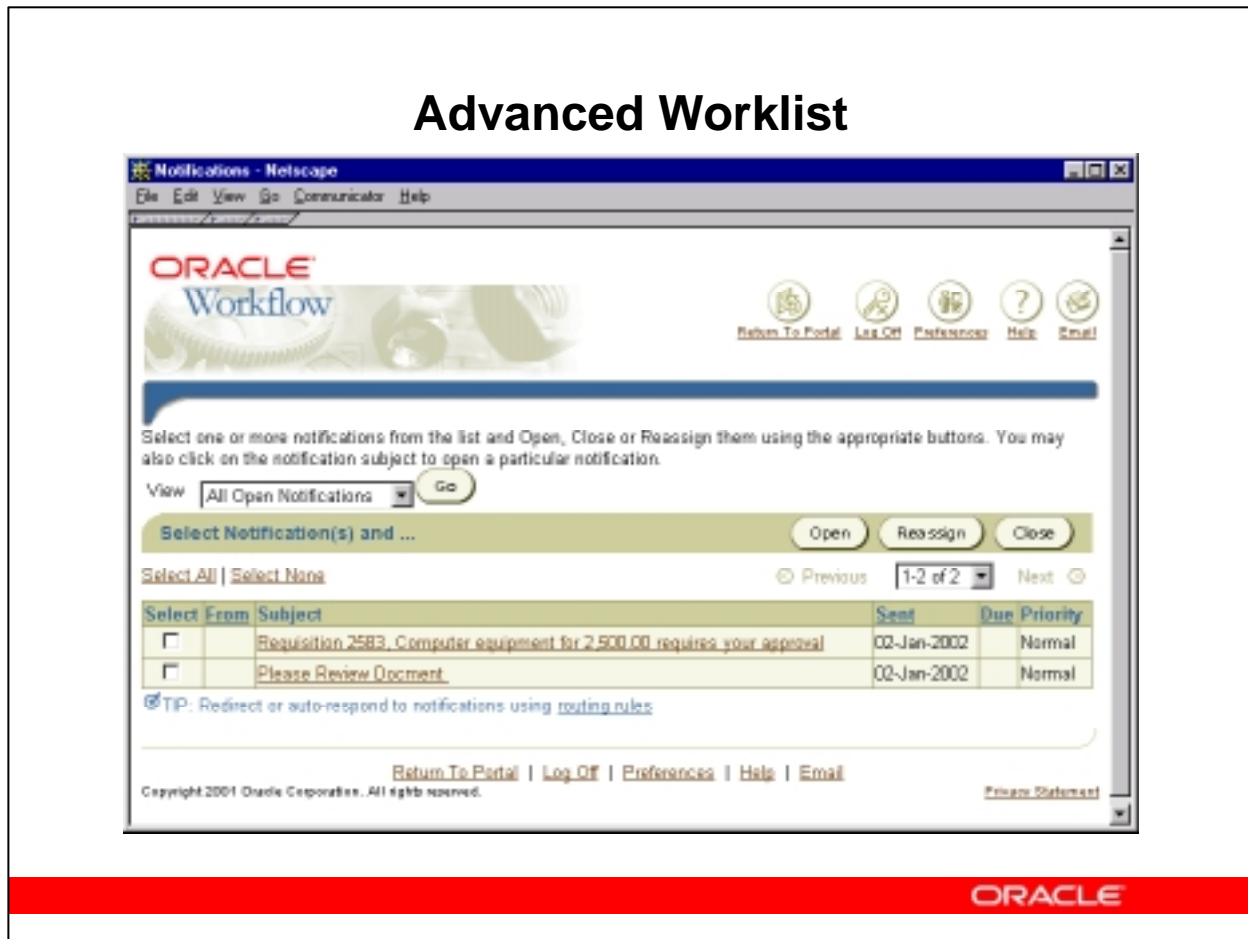
### Simple Worklist



ORACLE

### Simple Worklist

- The Simple Worklist in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch. Choose Simple Worklist from a Workflow responsibility to access this page.
- The Simple Worklist lists all your open notifications, displaying the from role, subject, sent date, due date, and priority for each notification.  
**Note:** The due date is determined by the timeout value set for the notification.
- Click the From, Sent, Due, or Priority column heading to sort your notifications by that column in ascending or descending order.
- To navigate to the full details of any notification, click the notification's Subject link.



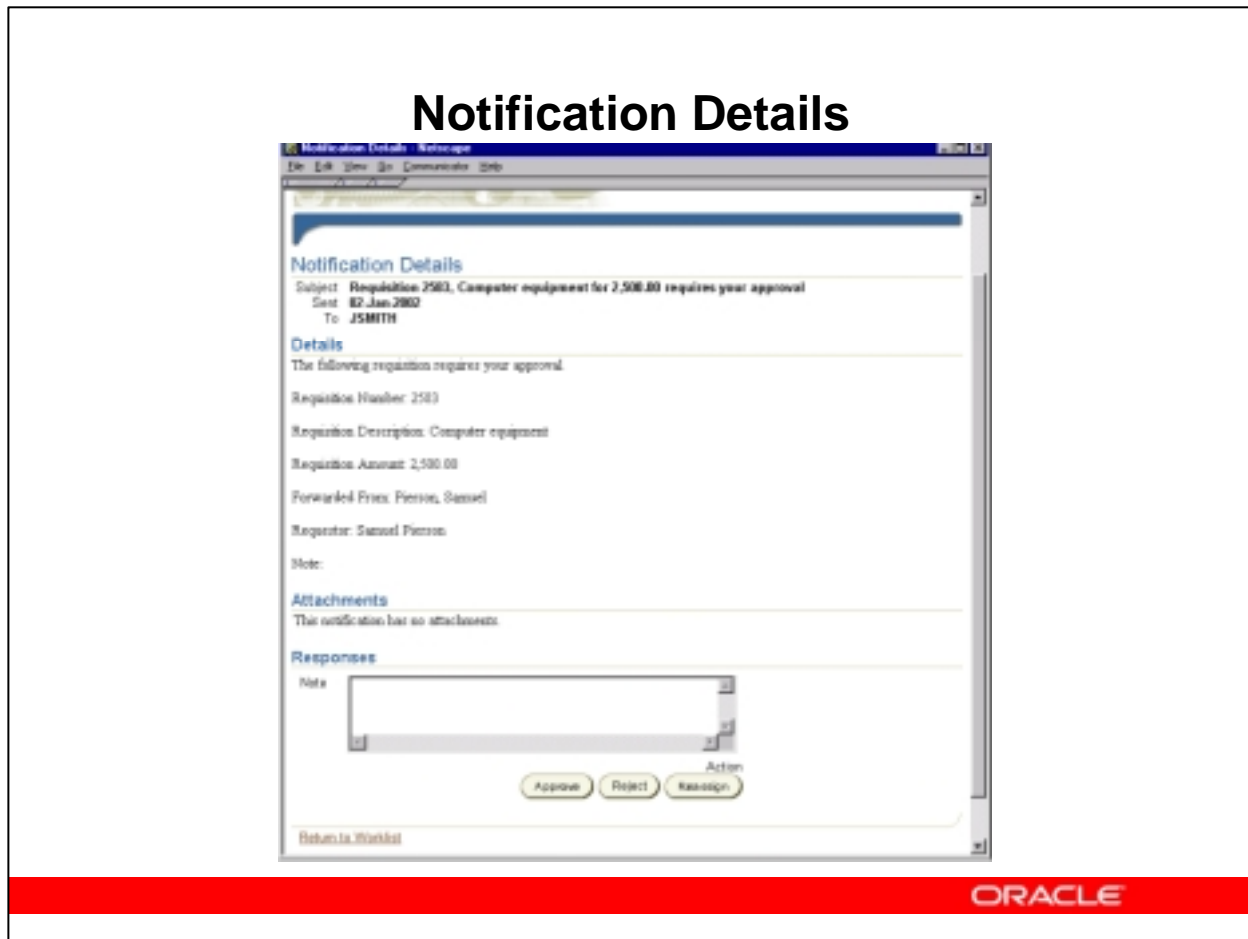
### Advanced Worklist

- The Advanced Worklist in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch. Choose Advanced Worklist from a Workflow responsibility to access this page.
- The Advanced Worklist lists all your open notifications, displaying the from role, subject, sent date, due date, and priority for each notification.
- Click the From, Sent, Due, or Priority column heading to sort your notifications by that column in ascending or descending order.
- A View poplist lets you specify the types of notifications to display in the Worklist. You can choose to view:
  - All Open Notifications
  - FYI Notifications
  - All To Do Notifications
  - All Notifications

After selecting the set of notifications to display, choose Go to display those notifications in the Worklist.

- The Advanced Worklist lets you simultaneously close multiple FYI-type notifications that do not require a response. Check Select for each FYI-type notification you want to close, and then choose Close.
- You can also collectively reassign a group of notifications. Check Select for the notifications you want to reassign, and then choose Reassign. A Reassign page appears that lets you specify to whom and how you wish to reassign the notifications.
- To navigate to the full details of any notification, click the notification's Subject link or select the notification and then choose Open.

## Notification Details



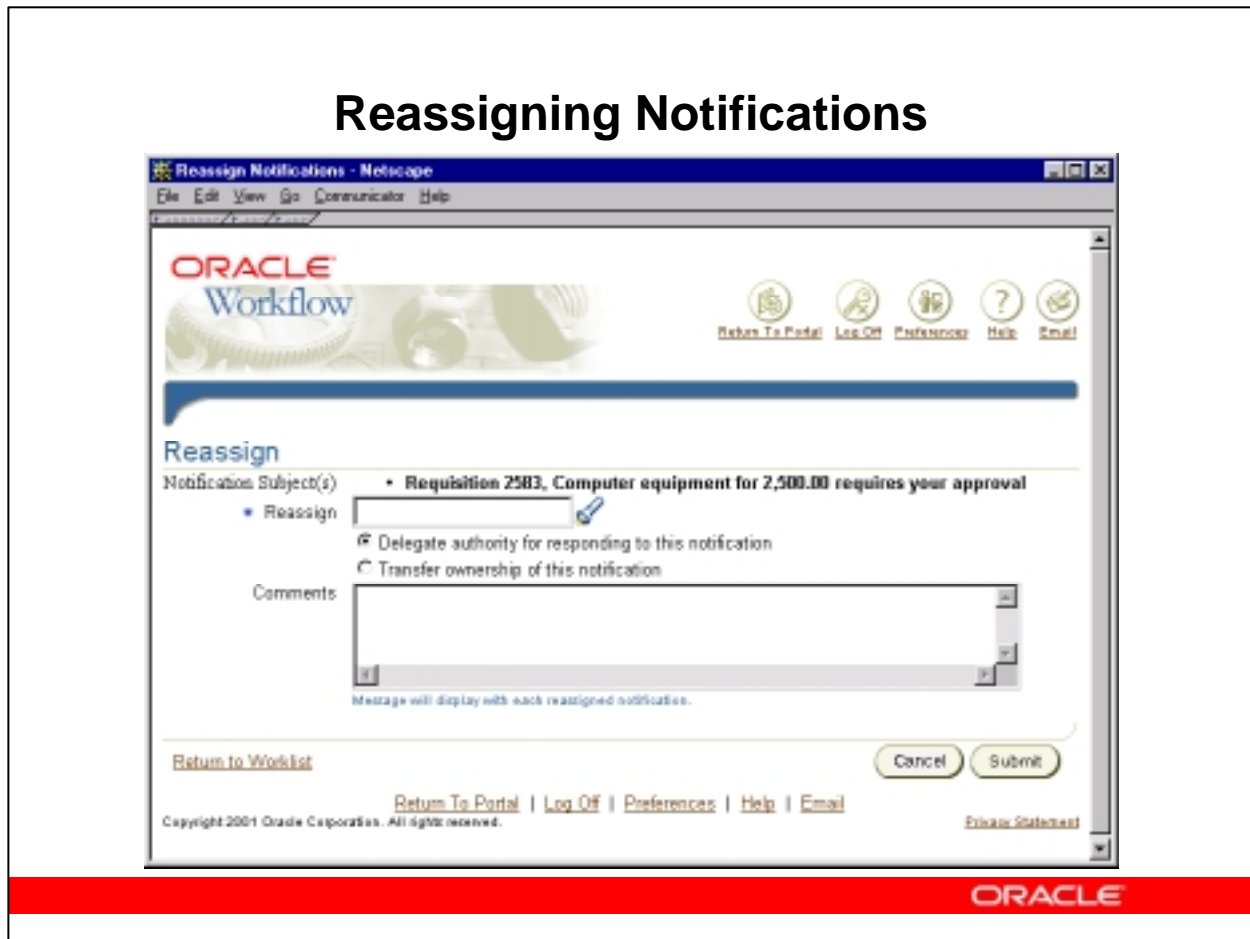
### Notification Details

- The Notification Details page in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch.
- You can access the Notification Details page from either the Simple Worklist or the Advanced Worklist.
- The Notification Details page shows the full details of the notification and may also include attachments and a response section.
- The Attachments section may include icons that let you drill down to URL, document, or form attachments.
- The Response section appears as follows:
  - If a notification requires a response, but none of the responses affect the result of the notification activity, the response prompts all appear as fields and/or poplists. When you are done entering your response values, submit your response by choosing the Submit button.
  - If a notification requires a response, and one of the responses becomes the result of the notification activity, then that determining response will appear last as a set of buttons to choose from. The buttons represent the possible choices to the response prompt. All

other response prompts, if any, appear as fields or poplists above that prompt. When you choose a button for that last response prompt, you also submit your response for the notification.

- If a notification does not require a response, the response section indicates that. Choose Close in the Response section to close the notification so that it does not appear in your worklist the next time you query for open notifications.
- The Response section may also include links that let you drill down to an Oracle E-Business Suite form to complete your response.

## Reassigning Notifications

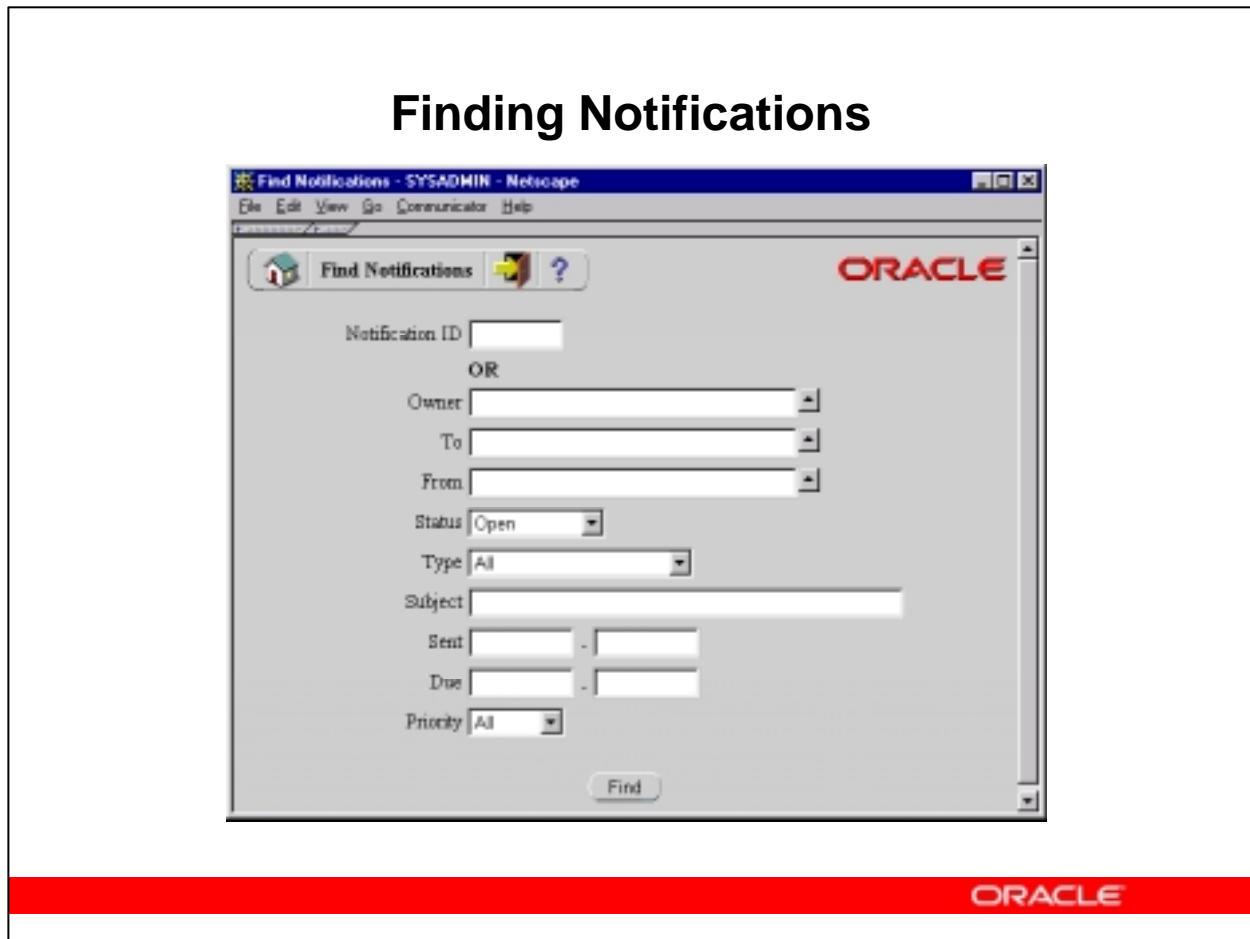


### Reassigning Notifications

- The Reassign page in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch.
- You can reassign notifications either by selecting notifications in the Advanced Worklist and choosing Reassign or by viewing a notification in the Notification Details page and choosing Reassign in the response section of the notification. Select the user to whom you want to reassign the notification, the reassign option you want to use, and any optional comments you want to include. Then choose Submit.
- Use the following options to reassign notifications:
  - Delegate authority for responding to this notification: Grant permission to someone else to act on a notification on your behalf while you remain listed as the owner and performer of the notification activity. The delegate function is WF\_NOTIFICATION.Forward.
  - Transfer ownership of this notification: Transfer responsibility to act on a notification to someone else and transfer the ownership and performance of the activity to that person. The transfer function is WF\_NOTIFICATION.Transfer.



## Finding Notifications



### Finding Notifications

To access the Find Notifications web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

`<webagent>/wfa_html.find`

Replace `<webagent>` with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Find Notifications web page from the Oracle Workflow home page, `<webagent>/wfa_html.home`.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Notifications option from a Workflow responsibility.

### Searching for Notifications

- Use the Find Notifications web page to search for a specific set of notifications.
- Enter search criteria to locate the notifications you want to view.
  - Users who have workflow administrator privileges can search for notifications that belong to any other user or role by specifying an internal role name in the Owner field.

- Users who do not have workflow administrator privileges can only search for their own notifications.
- Choose Find to locate notifications that meet your specified criteria and display those notifications in the Notifications Worklist web page.

## Notifications Worklist



### Notifications Worklist

You can access the Notifications Worklist web page either directly or from the Find Notifications web page.

To access the Notifications Worklist web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

`<webagent>/wfa_html.worklist`

Replace `<webagent>` with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Notifications Worklist web page from the Oracle Workflow home page, `<webagent>/wfa_html.home`.

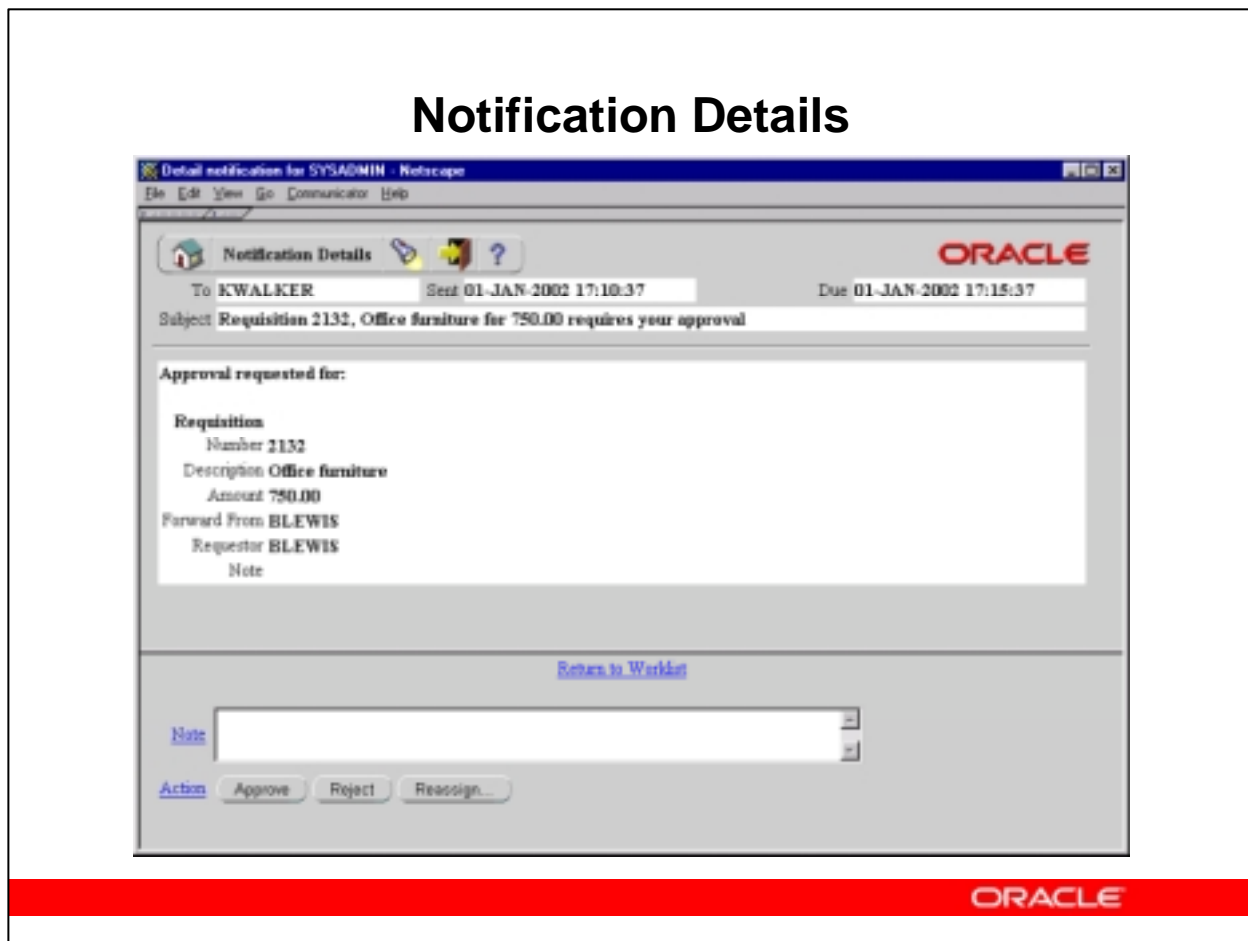
For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Notifications option from a Workflow responsibility.

### Reviewing Notifications in the Notifications Worklist

- The Notifications Worklist presents a summary list of notifications.
  - When you access the Notifications Worklist directly, it displays the list of all your open notifications.

- If you use the Find Notifications page to search for specific notifications, the Notifications Worklist displays only the notifications that match your search criteria.
- Click any column heading to sort the list by that column in ascending order.
- Click a notification subject link to display the details of the notification in the Notification Details web page.
  - In the Notification Details page, you can respond to or reassign a notification. After you respond to or reassign a notification, it no longer appears in your Worklist when you return to the Worklist page.
  - If a notification does not require a response, you can choose Close to remove it from your Worklist.
- You can simultaneously close multiple notifications that do not require a response by selecting those notifications in the Worklist and choosing Close.
- You can collectively reassign a group of notifications by selecting those notifications in the Worklist and choosing Reassign.

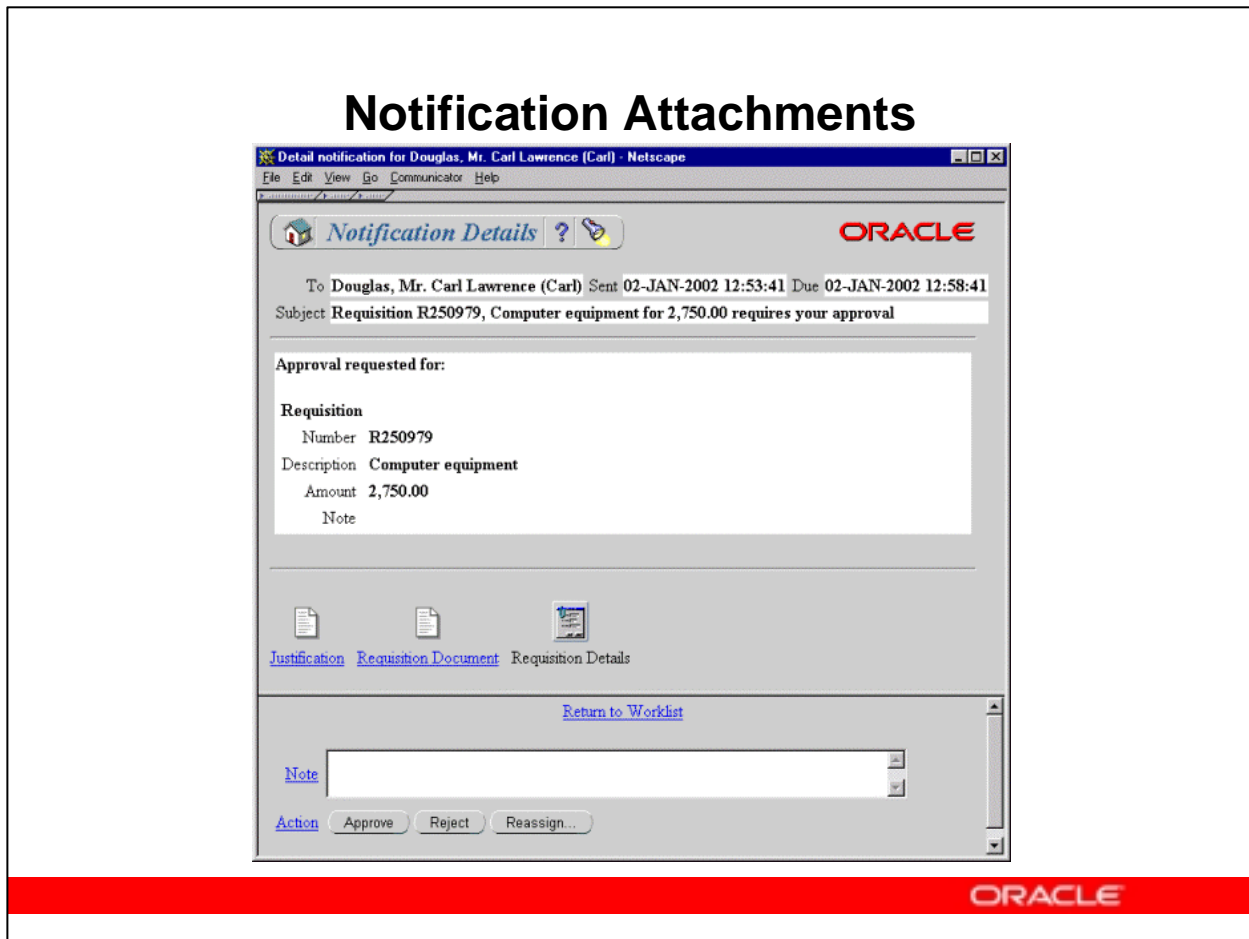
## Notification Details



### Notification Details

The Notification Details web page lets you view details for a single notification and respond to, reassign, or close the notification.

# Notification Attachments



## Notification Attachments

### Attached URLs

The Notification Details web page supports message attributes of type URL. These attributes appear in a notification message body as a hypertext link or below the message as an attachment icon. When you open your notification, you can click the link or attachment icon to launch the URL in the web browser according to the frame target specified for the attribute.

### Attached Forms

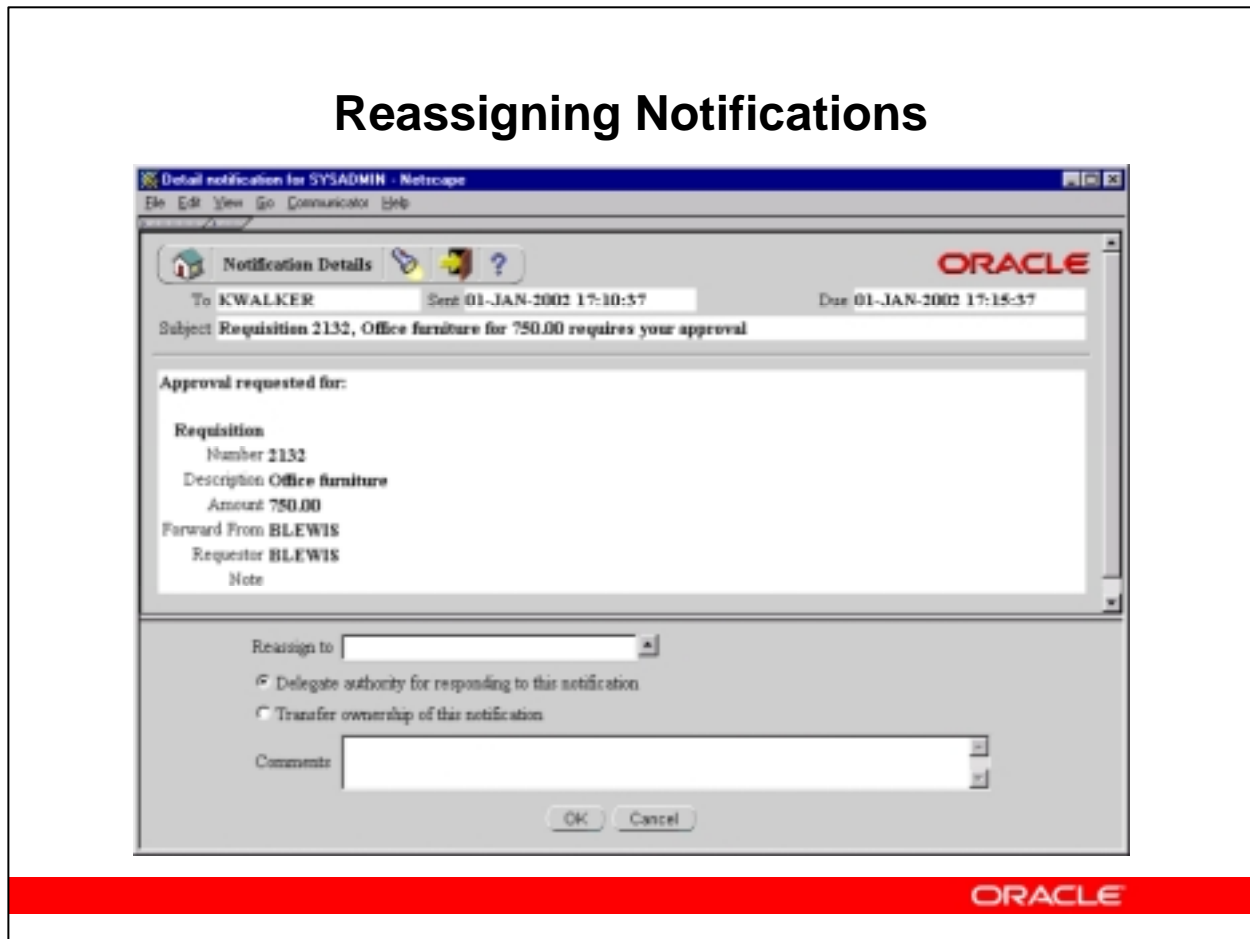
The Notification Details web page supports message attributes of type form. These attributes appear in a notification message as a form icon. When you open your notification, you can click the attached form icon to launch the referenced Oracle E-Business Suite form.

**Note:** This attribute type is not relevant for the standalone version of Oracle Workflow.

### Attached Documents

The Notification Detail web page supports message attributes of type document. These attributes appear in a notification message as an inline link or as an attachment icon. When you open your notification, you can click a link or an attachment icon to open the referenced document.

## Reassigning Notifications



### Reassigning Notifications

You can reassign notifications either by selecting notifications in the Notifications Worklist and choosing Reassign or by viewing a notification in the Notification Details page and choosing Reassign in the response section of the notification. Select the user to whom you want to reassign the notification, the reassign option you want to use, and any optional comments you want to include. Then choose OK.

Use the following options to reassign notifications:

- Delegate: Grant permission to someone else to act on a notification on your behalf while you remain listed as the owner and performer of the notification activity. The delegate function is `WF_NOTIFICATION.Forward`.
- Transfer: Transfer responsibility to act on a notification to someone else and transfer the ownership and performance of the activity to that person. The transfer function is `WF_NOTIFICATION.Transfer`.

The following Oracle Workflow areas use these notification reassignment values:

- The Workflow Monitor Diagram web page
- The Notifications web pages

If you view your notifications through e-mail, you can only reassign a notification by using the Forward function, which performs similarly to the Delegate option.



### Viewing Notifications through E-mail

- **You can receive e-mail notifications if your workflow administrator sets up the Notification Mailer to run and your notification preference is set to one of the following in the User Preferences web page:**
  - Plain text mail
  - HTML mail
  - Plain text mail with HTML attachments
- **An e-mail notification contains all the details of the notification, including instructions on how to respond to the notification.**



ORACLE

### Viewing Notifications through E-mail

- **Plain text mail:** The notification message appears as plain text, with no attachments. If the message requires a response, you can respond by replying to the e-mail message.
- **HTML mail:** The notification message appears as HTML-formatted text, with at least one other attachment that is a link to the notification in the Notification Details web page. If the notification message has 'Content-Attached' message attributes, these attributes appear as additional attachments to the message. If the message requires a response, you can respond by clicking one of the response links or you can navigate to the Notification Details page and respond there.
- **Plain text mail with HTML attachments:** The notification message appears as plain text, with at least two other attachments. One attachment is an HTML-formatted version of the message, and the other is a link to the notification in the Notifications web page. If the notification message has 'Content-Attached' message attributes, these attributes appear as additional attachments to the message. If the message requires a response, you can respond by replying to the e-mail message or by opening the HTML-formatted version of the message and clicking one of the response links, or you can navigate to the Notification Details page and respond there.

### E-mail Response Methods

- **Your workflow administrator determines the response method for plain text e-mail notifications when setting up the Notification Mailer.**
  - **Direct response:** Enter your response values directly as the first lines of your reply.
  - **Templated response:** Reply using the template of response prompts provided in the notification and enter your response values between the quotes following each prompt.
- **HTML-formatted notifications always use the templated response method.**



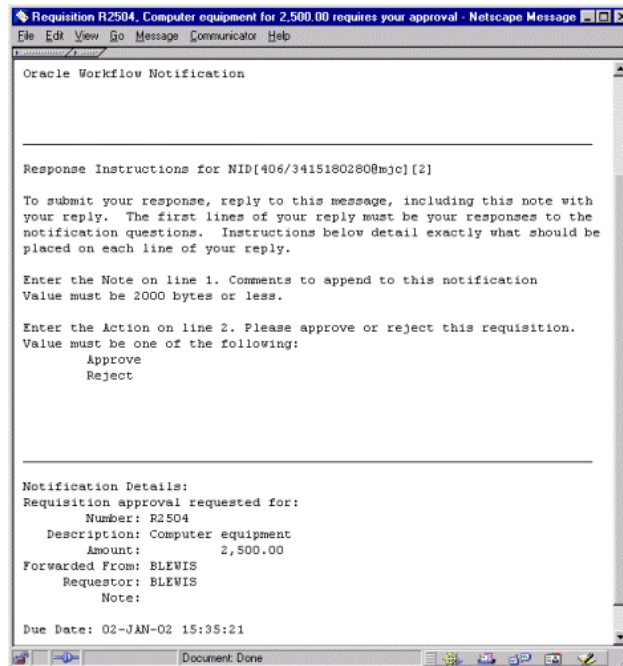
ORACLE

### E-mail Response Methods

The Notification System interprets your response values literally, so uppercase values are distinct from lowercase values.

## Plain Text E-mail Notifications Using Direct Response

### Plain Text E-mail Notifications Using Direct Response



ORACLE

### Plain Text E-mail Notifications Using Direct Response

Include the text of the original notification in your reply to ensure that you include the special notification ID and access key that the Notification Mailer requires to identify the notification you are responding to. The first lines of your reply are interpreted as your notification response, where each line represents a separate response value listed in the same order as its corresponding response prompt. If a response value requires more than one line, enclose the entire response value in double quotes. If a response prompt provides a default value, then you can accept the default value by leaving the appropriate response line blank.

#### Attachments

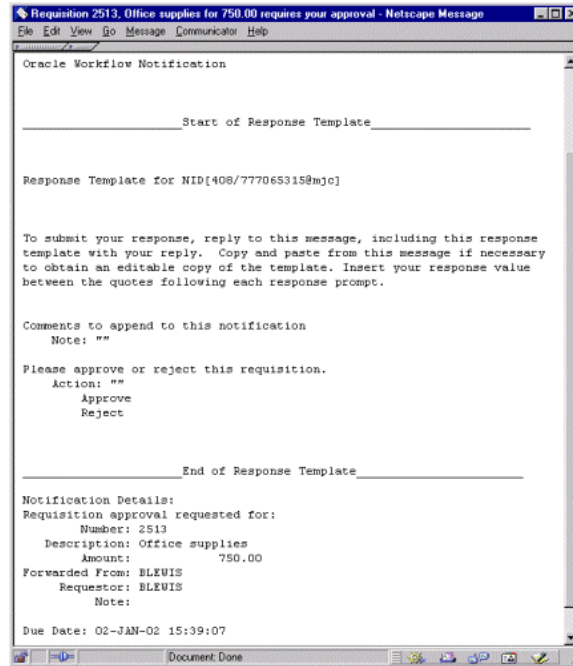
In a plain text e-mail notification, the URL location value for an attached URL attribute is included as plain text. PL/SQL or PL/SQL CLOB document attributes are attached as plain text to their parent notification. Note that this may render some attachments unreadable if the attachment includes special formatting or your plain text e-mail reader does not recognize attachments. To view these attachments, you should display your notifications in the Notification Details web page.

**Note:** Form attributes cannot be attached to e-mail notifications. To view form attachments, you must view the notification in the Notification Details web page.

## Plain Text E-mail Notifications Using Templated Response

---

### Plain Text E-mail Notifications Using Templated Response



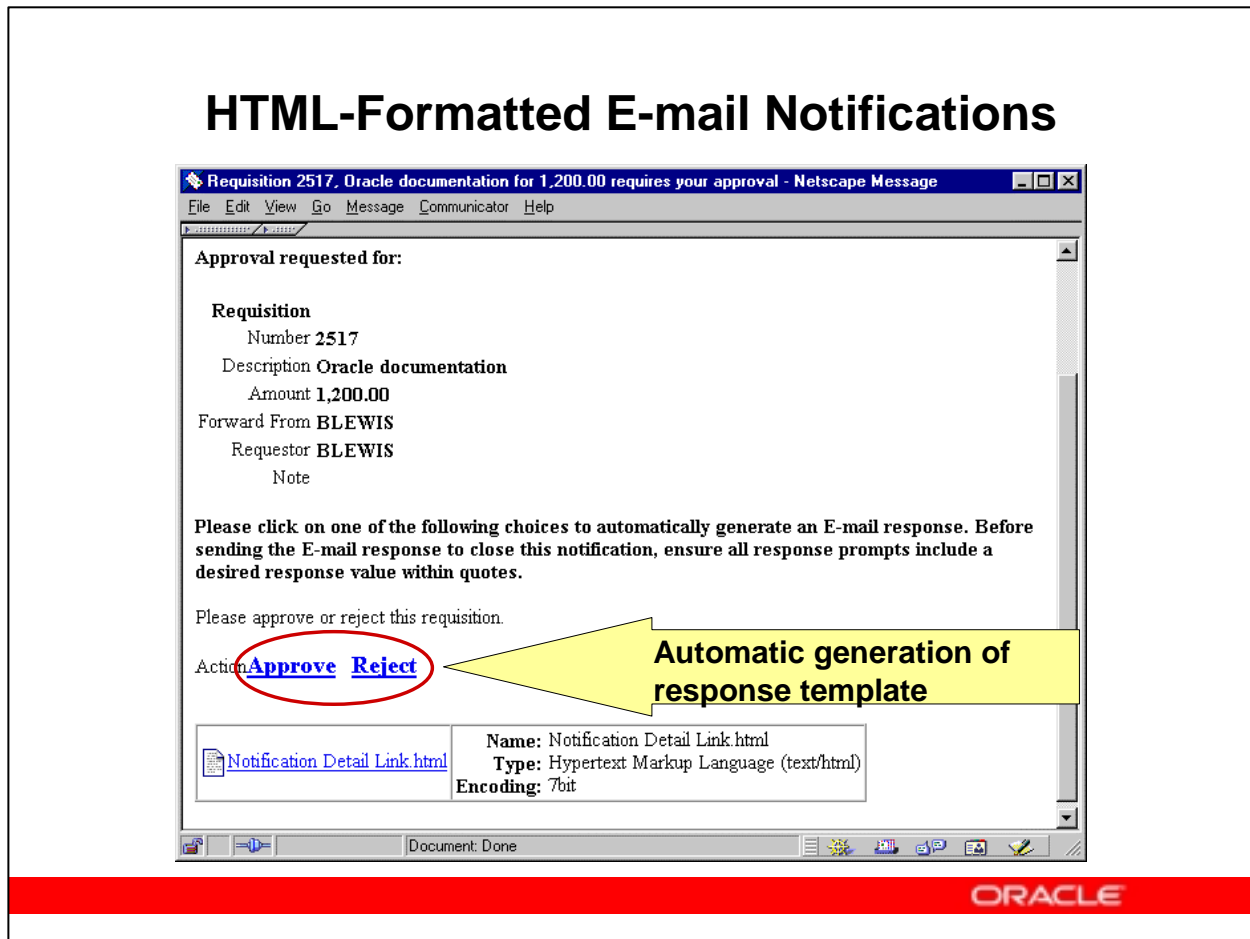
ORACLE

### Plain Text E-mail Notifications Using Templated Response

Include the response template from the original notification in your reply. In addition to the response prompts, the response template includes the special notification ID and access key that the Notification Mailer requires to identify the notification you are responding to. If your mail application includes an editable copy of the original message when it generates the reply message, you can use that copy to enter your response values. Otherwise, copy and paste from the original message to obtain a copy of the response template that you can edit.

Follow the response template instructions and insert your response values between the quotes (“”) following each response prompt.

# HTML-Formatted E-mail Notifications



## HTML-Formatted E-mail Notifications

HTML e-mail notifications always use the templated response method.

The response section of an HTML e-mail notification includes links for the possible result response values at the end of the notification. After reviewing the message body of the notification, click one of the response links.

Each response link automatically generates a plain text e-mail reply that contains the correct Reply To: e-mail address as well as a response template in the message body. The response template consists of the required notification ID and access key that identify the notification you are responding to and a response prompt edited with your selected result response.

Depending on the notification, the auto-generated e-mail response template may also prompt you for other information in addition to your selected result response.

An HTML-formatted notification always includes at least one attachment. The attachment is called Notification Detail Link. When you select this attachment, your e-mail reader opens a browser window that displays your notification in the Notification Details web page. You can respond directly to your notification from this web page, bypassing the need to process your response through the Notification Mailer.

**Note:** If your system administrator has set the `SEND_ACCESS_KEY` parameter in the Notification Mailer configuration file to N, and you are not already logged in, you will be prompted to log in when you select the Notification Detail Link before you can access the Notification Details web page.

### **Attachments**

In an HTML-formatted e-mail notification, a URL attribute appears as a hypertext link in the message body. For attached URL attributes, an attachment called Notification References is appended to the message. This attachment includes a link to each attached URL attribute for the message. You can navigate to a URL by choosing its link.

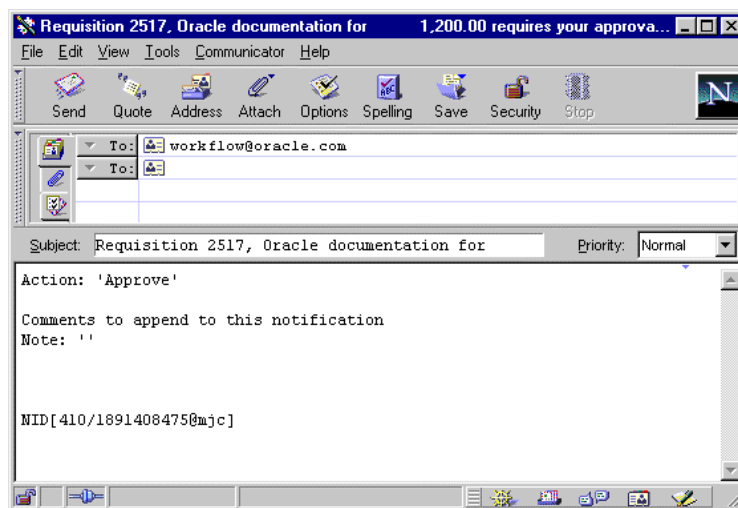
If an HTML-formatted e-mail notification has an attached PL/SQL or PL/SQL CLOB document attribute, the fully generated PL/SQL document is fetched and attached to the message.

**Note:** Form attributes cannot be attached to e-mail notifications. To view form attachments, you must view the notification in the Notification Details web page.

## Response Template

---

### Response Template



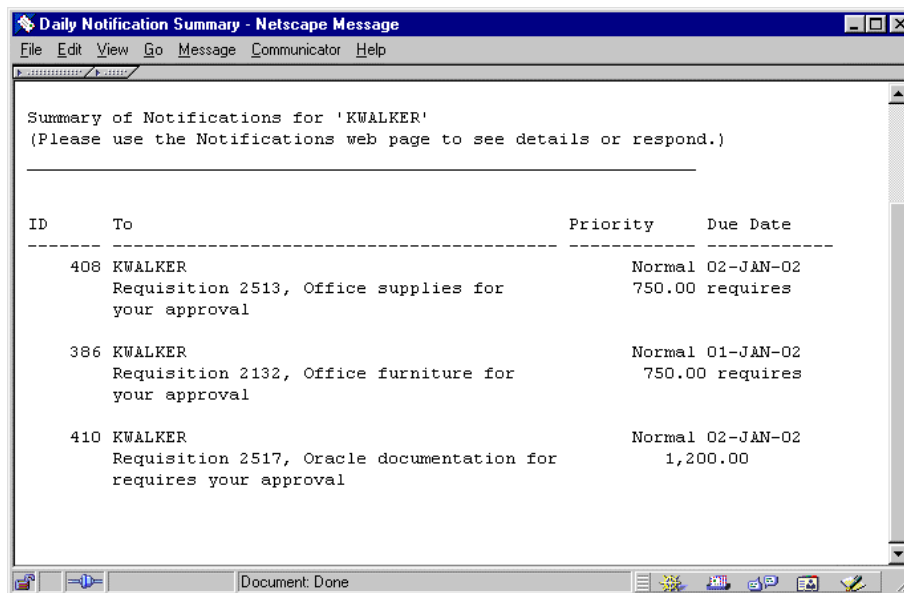
ORACLE

### Response Template

When responding to a notification with an automatically generated template, supply your responses by editing the response value text between the quotes following each prompt.

## Viewing an E-mail Summary of Notifications

### Viewing an E-mail Summary of Notifications



ORACLE

### Viewing an E-mail Summary of Notifications

- You can receive an e-mail summary of notifications if your workflow administrator sets up the Notification Mailer to run and your notification preference is set to 'Plain text summary mail' in the User Preferences web page.
- The Notification Mailer delivers a single e-mail message summarizing your current list of open notifications.
- To respond to individual notifications listed in the summary, you must use the Notification Details web page.
- How often you receive e-mail summaries depends on how frequently the Notification Mailer that handles e-mail summaries is scheduled to run.



### Automatic Notification Processing

- **You can automatically forward or respond to incoming notifications during a planned absence by defining automatic notification processing rules.**
- **Use the Notification Rules web pages to define your rules.**
- **Each rule is specific to a role and can optionally apply to messages of a specific item type or message name.**
- **A rule can result in one of three actions:**
  - Reassign the notification to another role
  - Respond to or close the notification
  - Deliver to the original recipient with no further action



ORACLE

### Automatic Notification Processing

- When a notification is sent or reassigned to a role, Oracle Workflow tests the notification against that role's list of automatic processing rules for the most specific match based on the following order of criteria:

ROLE = <role> and:

1. ITEM\_TYPE = <type> and MESSAGE\_NAME = <name>
2. ITEM\_TYPE = <type> and MESSAGE\_NAME is null
3. ITEM\_TYPE is null and MESSAGE\_NAME is null

- If a rule reassigns a notification, Oracle Workflow again performs rule matching against the new recipient role's list of rules.
- A count of the number of times that a notification is forwarded helps detect perpetual forwarding cycles. If a notification is automatically forwarded more than ten times, Oracle Workflow assumes a forwarding cycle has occurred, ceases all further automatic forwarding, and marks the notification as being in error.

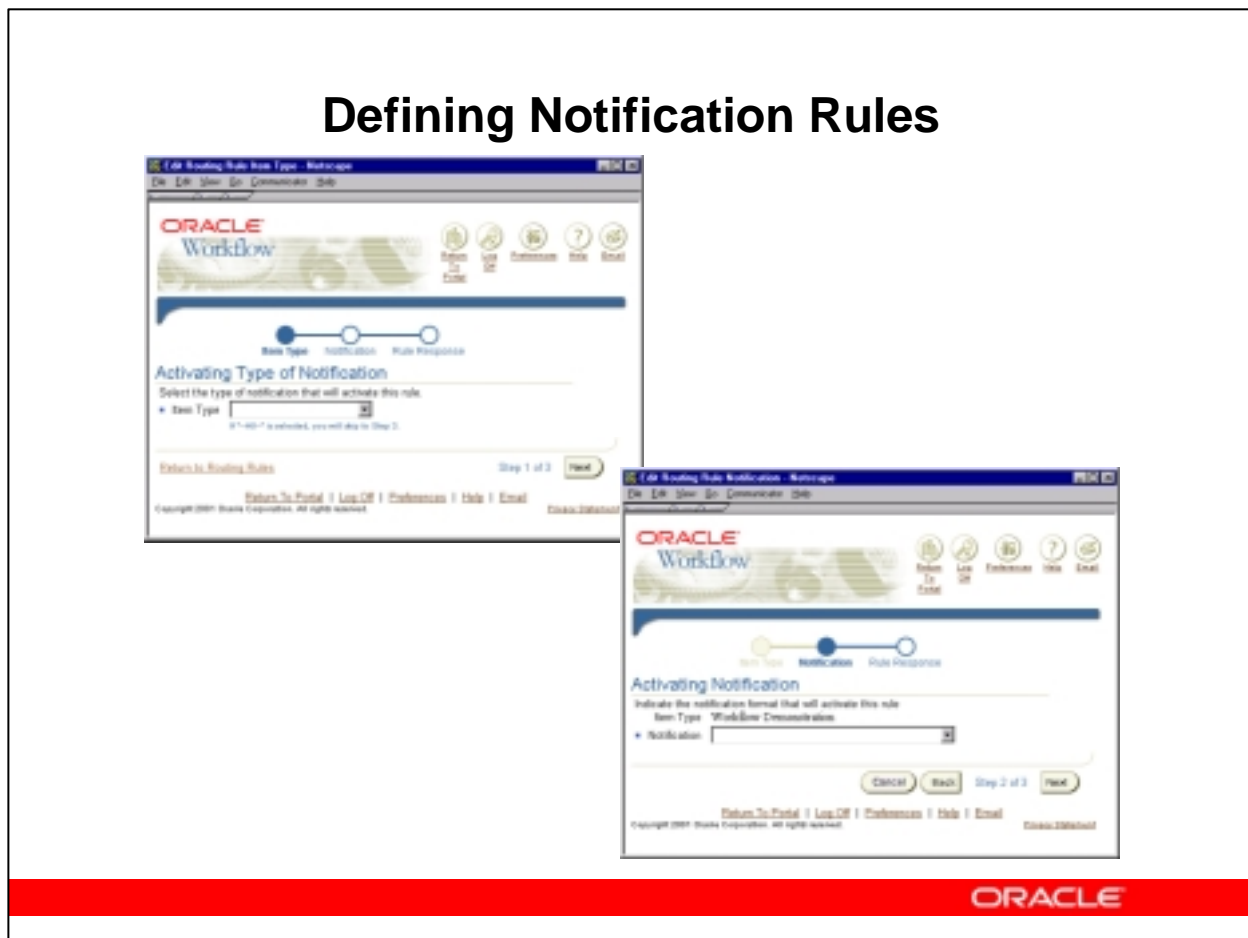
## Viewing Notification Rules



### Viewing Notification Rules

- The Notification Routing Rules page in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch.
- You can access the Notification Routing Rules page by clicking the routing rules link in the tip at the bottom of either the Simple Worklist or the Advanced Worklist.
- The Notification Routing Rules page displays a list of the existing rules for the current role. The following information is listed for each rule:
  - Rule name describing the action the rule performs
  - Item type
  - Notification name, or <ALL> for all notifications within that item type
  - Active status
- Click the rule link in the Rule Name column or the edit icon in the Update column to edit a rule.
- Click the delete icon in the Delete column to delete a rule.
- Choose the Create Rule button to define a new rule.

## Defining Notification Rules



### Defining New Notification Rules

1. Choose Create Rule in the Notification Routing Rules page.
2. In the Activating Type of Notification page, specify the item type to which you want the rule to apply, and choose Next.
  - Rules apply to notification messages, and all messages are associated with a specific item type.
  - Choose <ALL> if you want the rule to apply to all item types.
3. In the Activating Notification page, specify the name of the notification message to which you want the rule to apply, and choose Next.
  - Messages are listed shown by subject name.
  - Choose <ALL> if you want the rule to apply to all messages.
  - Skip this step if you selected <ALL> as the item type.

## Defining Notification Rules

### Defining Notification Rules

**Rule Response**

Complete the form below to set up the proper response for the rule.

Item Type: **Workflow Demonstration**

Notification: **All**

Start Date:  (example: dd-MMM-yyyy)

End Date:  (example: dd-MMM-yyyy)

Message:

Work Item  Reassign to:   Delegate authority for responding to this notification  
 Transfer ownership of this notification  
 Deliver notifications to me regardless of any general rules

Return To Portal | Log Off | Preferences | Help | Email

Copyright 2001 Oracle Corporation. All rights reserved. [Privacy Statement](#)

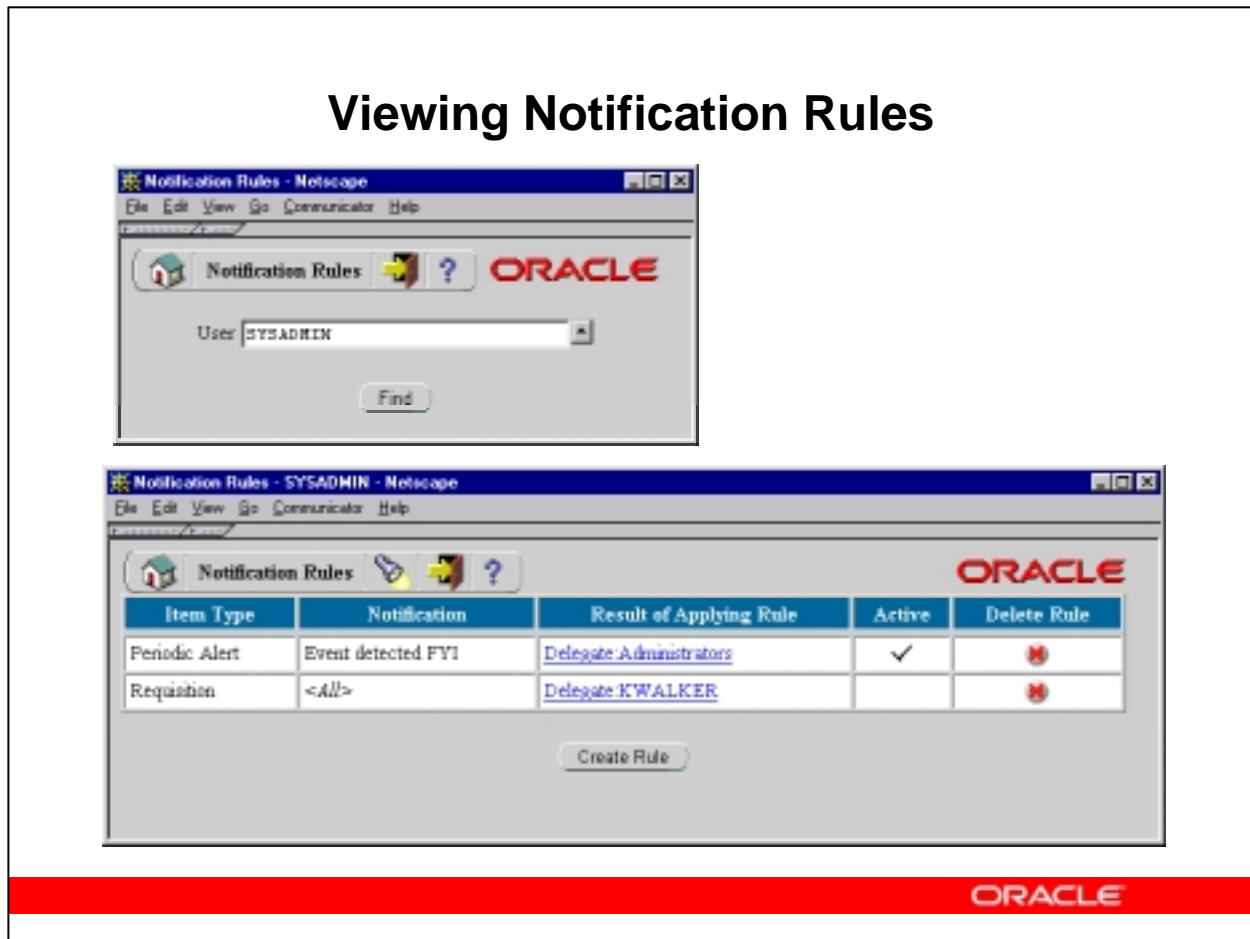
**ORACLE**

### Defining New Notification Rules

4. In the Rule Response page, specify the period during which you want the rule to be active by entering a start and end date and time.
    - Use the default date format of your database.
    - Use the time format HH24:MI:SS.
    - A blank Start Date means the rule is effective immediately.
    - A blank End Date means the rule is effective indefinitely.
- Note:** Because you can define multiple rules for the same notifications, make sure your rules for the same notifications do not overlap in their effective dates.
5. Use the Message field to specify text that you want to append to the notification in the “Prior comments” field when the rule is applied.
  6. Select the action for the rule.
    - Reassign to: Forward the notification to the specified role. In this case you must also specify a role and select either ‘Delegate Authority for Responding to Notifications’ or ‘Transfer Ownership of Notifications’.
    - Respond: Respond to the message with a set of predefined response values.

- Deliver Notifications to me, regardless of any general rules: Leave the notification in your inbox and do nothing. Use this action to exclude a subset of notifications from a more encompassing rule.
7. Click the Submit button to save the rule. You can also choose Cancel if you do not want to save the rule.

## Viewing Notification Rules



### Viewing Notification Rules

To access the Notification Rules web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

`<webagent>/wf_route.find`

Replace `<webagent>` with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Notification Rules web page from the Oracle Workflow home page, `<webagent>/wfa_html.home`.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Notification Rules option from a Workflow responsibility.

In the Notification Rules search page, enter a role name in the User field and choose Find to display the list of rules for that role. You can view rules for roles other than your own only if you have workflow administrator privileges.

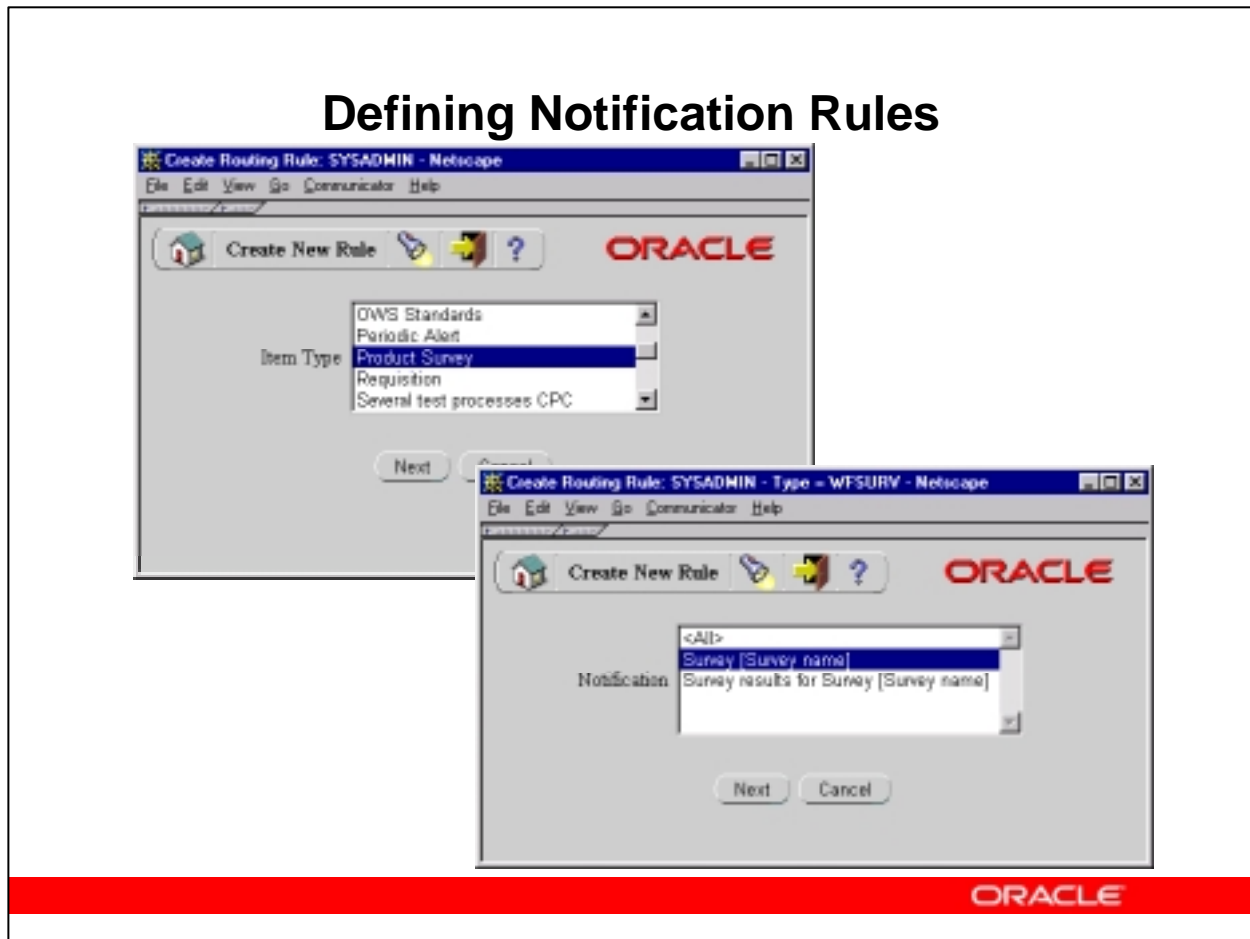
To access the Notification Rules list page directly for standalone Oracle Workflow, use a web browser to connect to the following URL:

`<webagent>/wf_route.list`

## Notification Rules List

- When a notification is sent or reassigned to a role, Oracle Workflow tests the notification against that role's list of automatic processing rules for the most specific match based on the following order of criteria:
- The Notification Rules list page displays the existing rules for the role. The following information is listed for each rule:
  - Item type
  - Notification name, or <ALL> for all notifications within that item type
  - Result of applying the rule
  - Active status
- Click the action link in the "Result of Applying Rule" column to edit a rule.
- Choose the delete icon in the Delete Rule column to delete a rule.
- Choose the Create Rule button to define a new rule.

## Defining Notification Rules



### Defining New Notification Rules

1. Choose Create Rule in the Notification Rules list page.
2. Specify the item type to which you want the rule to apply, and choose Next.
  - Rules apply to notification messages, and all messages are associated with a specific item type.
  - Choose <ALL> if you want the rule to apply to all item types.
3. Specify the name of the notification message to which you want the rule to apply, and choose Next.
  - Messages are listed shown by subject name.
  - Choose <ALL> if you want the rule to apply to all messages.
  - Skip this step if you selected <ALL> as the item type.



## Defining Notification Rules

### Defining Notification Rules

The screenshot shows a web-based form for creating a notification rule. The form is titled "Create New Rule" and is part of the Oracle NetSuite interface. It includes the following fields and options:

- Item Type:** Product Survey
- Notification:** Survey [Survey name]
- Start Date:** 01-JAN-2002 00:00:00
- End Date:** (empty)
- Automatically:**
  - Comments to include in notification:** (text area)
  - Reassign to:** (dropdown menu)
    - Delegate Authority for responding to Notifications
    - Transfer Ownership of Notifications
  - Respond:** (radio button)
  - ranking 1 - 10:** (text area)
  - Comments:** (text area)
- Deliver Notifications to me, regardless of any general rules

Buttons: OK, Cancel

### Defining New Notification Rules

- Specify the period during which you want the rule to be active by entering a start and end date and time.
  - Use the default date format of your database.
  - Use the time format HH24:MI:SS.
  - A blank Start Date means the rule is effective immediately.
  - A blank End Date means the rule is effective indefinitely.

**Note:** Because you can define multiple rules for the same notifications, make sure your rules for the same notifications do not overlap in their effective dates.

- Use the “Comments to include in notification” field to specify text that you want to append to the notification in the “Prior comments” field when the rule is applied.
- Select the action for the rule.
  - **Reassign to:** Forward the notification to the specified role. In this case you must also specify a role and select either ‘Delegate Authority for Responding to Notifications’ or ‘Transfer Ownership of Notifications’.
  - **Respond:** Respond to the message with a set of predefined response values.

- Deliver Notifications to me, regardless of any general rules: Leave the notification in your inbox and do nothing. Use this action to exclude a subset of notifications from a more encompassing rule.
7. Click the OK button to save the rule. You can also choose Cancel if you do not want to save the rule.

### Summary

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

- **Access the Oracle Workflow web pages.**
- **Respond to notifications from the Notifications Worklist web page.**
- **Respond to notifications using e-mail.**
- **Define notification routing rules.**

ORACLE



# Monitoring Workflow Processes

## Chapter 22

# Monitoring Workflow Processes

ORACLE

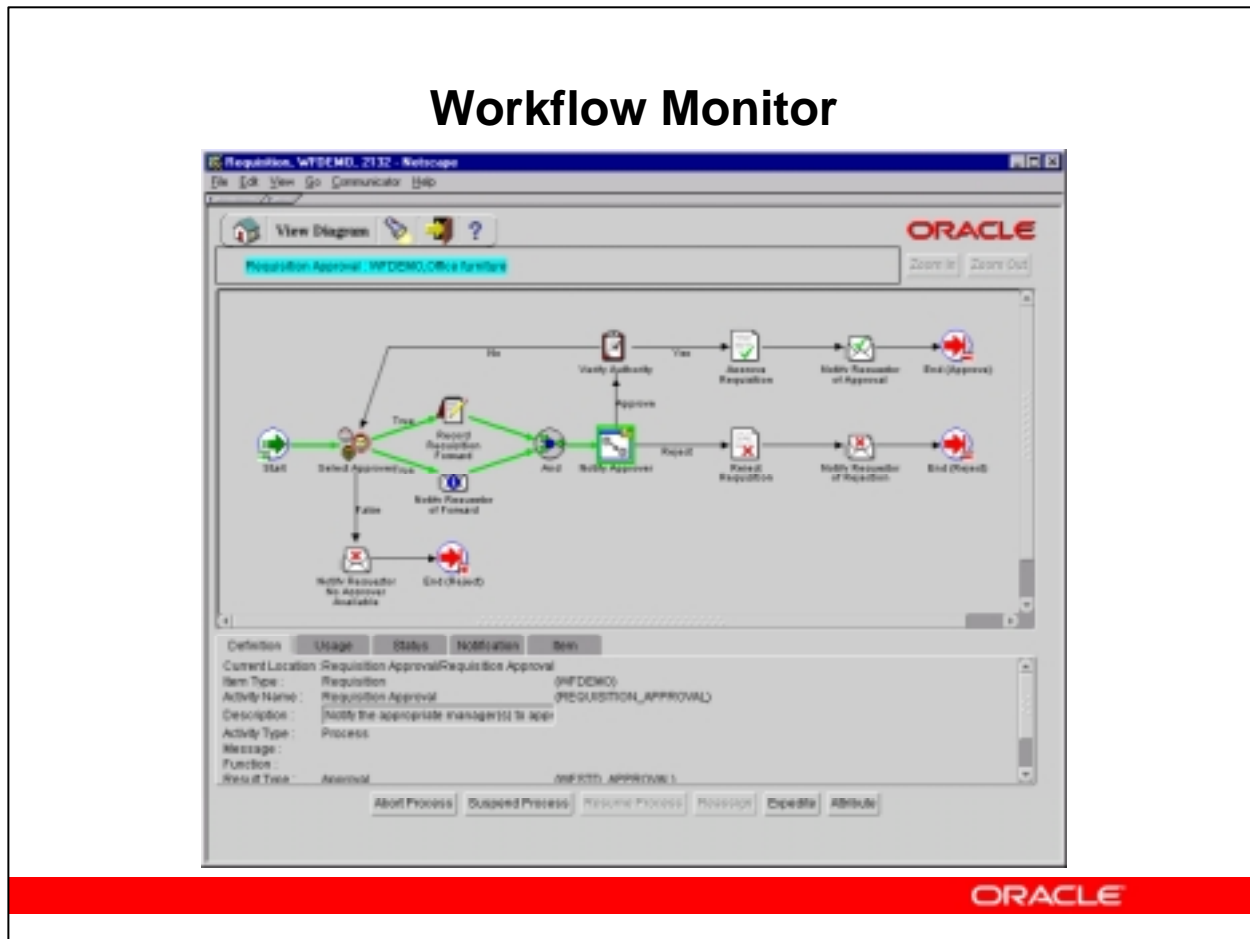
### Objectives

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

- **Check the progress of a workflow using the Workflow Monitor.**
- **Launch a test process.**

ORACLE

# Workflow Monitor



## Workflow Monitor

The Workflow Monitor is a Java-based tool that lets you view and administer the progress of a workflow process instance. Workflow Monitor features include:

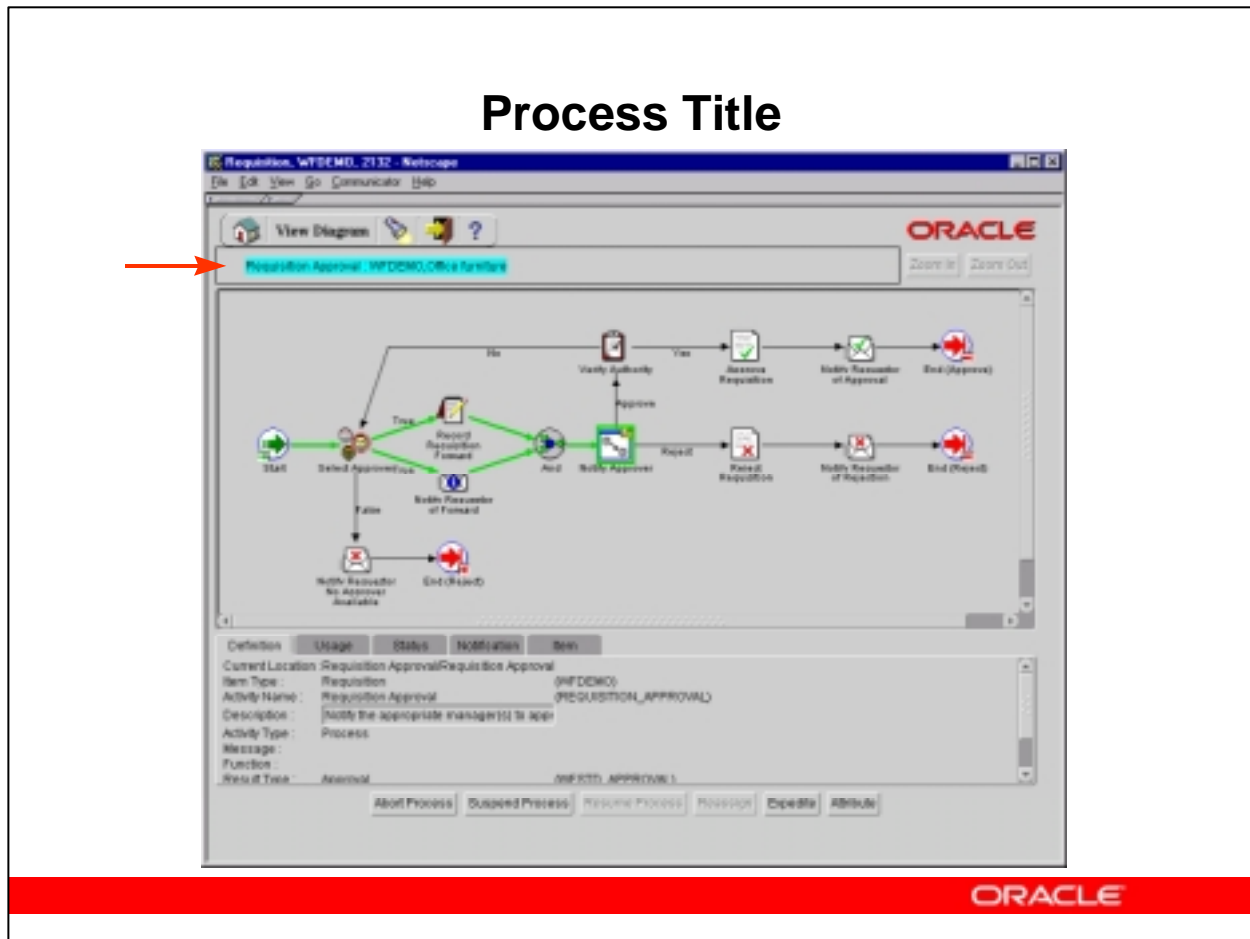
- A point-and-click interface
- A display for detailed status information for individual activities as well as for the whole process
- The ability to run in USER mode or in ADMIN mode, which provides additional details and functionality pertinent only to a workflow administrator

User interface components include:

- Process title
- Process diagram window
- Detail tab window
- Administration buttons



## Process Title



## Process Title

The process title displays the process name, followed by the item type and user key (or item key) that uniquely identifies the process instance. If you drill down to a subprocess in the process diagram window, the process title shows the name of the subprocess.



## Detail Tab Window

**Detail Tab Window**

The screenshot shows the Oracle Workflow Monitor interface. At the top, the title bar reads 'Requestion Approval, WFDEMO, 21:32, Netscape'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Go', 'Communicator', and 'Help'. A toolbar contains icons for 'View Diagram', a magnifying glass, and a question mark. The main area displays a flowchart for the 'Requestion Approval' process. The flow starts with 'Start', followed by 'Select Approver(s)', which branches into 'True' and 'False'. The 'True' path leads to 'Requestion Forward', which then branches into 'Approve' and 'Notify Approver'. The 'Approve' path leads to 'Notify Approver', which then branches into 'Approve' and 'Reject'. The 'Reject' path leads to 'Notify Approver', which then branches into 'Notify Responder of Approval' and 'End (Reject)'. The 'Notify Approver' path leads to 'Notify Responder of Approval', which then branches into 'Notify Responder of Approval' and 'End (Approve)'. The 'False' path leads to 'Notify Responder No Approval Available', which then branches into 'Notify Responder No Approval Available' and 'End (Error)'. Below the flowchart is a table with columns for 'Definition', 'Usage', 'Status', 'Notification', and 'Item'. The 'Definition' tab is selected, showing the following information:

Definition	Usage	Status	Notification	Item
Current Location	Requestion Approval	Requestion Approval		
Item Type	Requestion Approval			WFDEMO
Activity Name	Requestion Approval			REQUESTION_APPROVAL
Description	Notify the appropriate manager(s) to app			
Activity Type	Process			
Message				
Function				
Result Tree	Approval REQUESTION_APPROVAL			

Below the table are buttons for 'About Process', 'Suspend Process', 'Resume Process', 'Refresh', 'Delete', and 'Archive'. An orange arrow points to the 'Definition' tab.

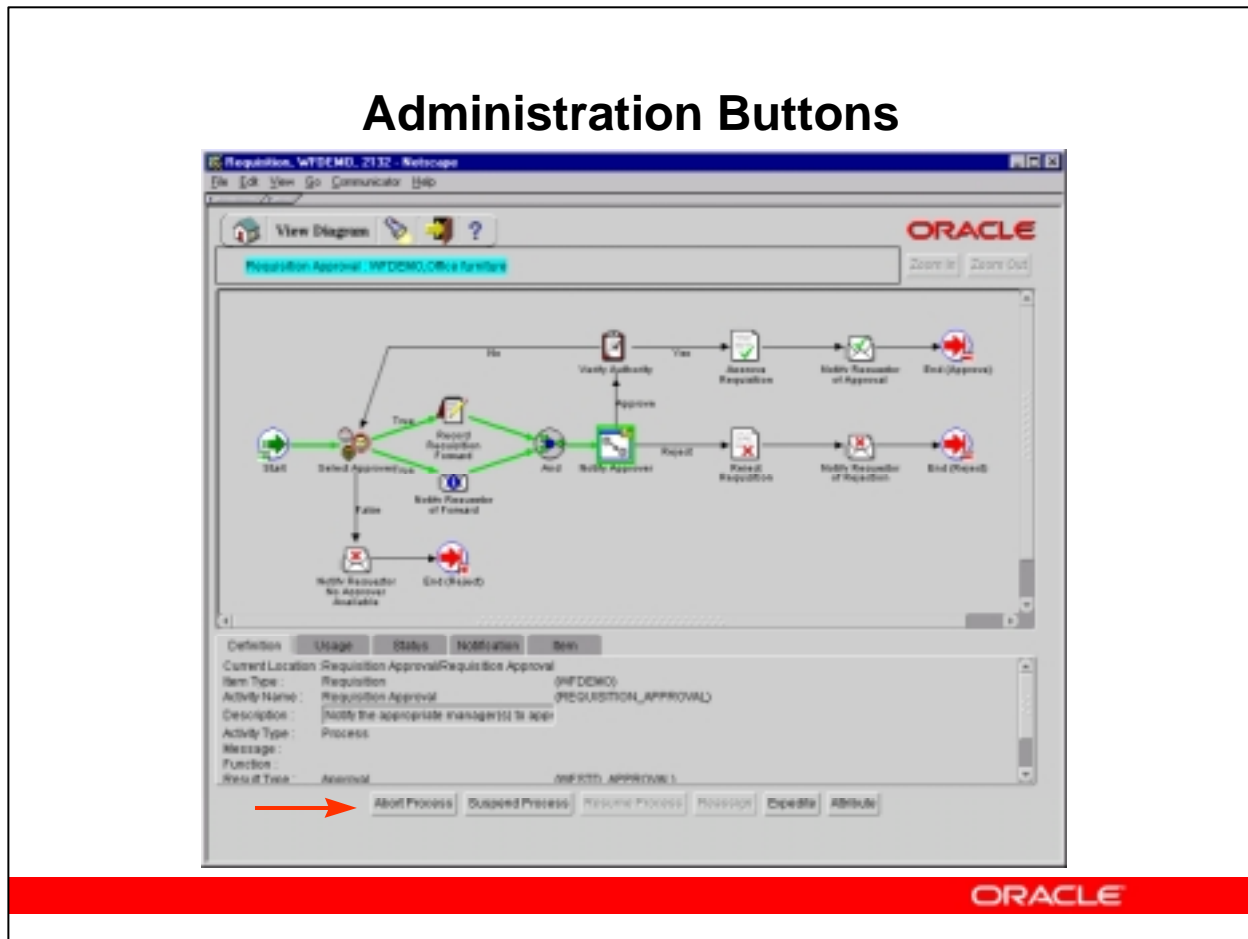
**ORACLE**

### Detail Tab Window

The detail tab window displays detailed information about the selected activity or process. Additional information may appear in these tabs if you are accessing the Workflow Monitor in ADMIN mode. Select from the following tabs:

- **Definition:** Displays the properties of the activity or process.
- **Usage:** Displays the properties for the activity as a node in the process.
- **Status:** Displays status and result information about the activity. Also shows error information if the activity status is ERROR.
- **Notification:** Displays notification details for the selected notification activity.
- **Item:** Displays item type and item type attribute information.

## Administration Buttons



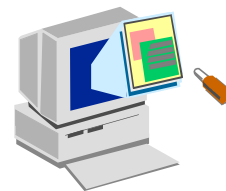
### Administration Buttons

The administration buttons appear only when you run the Workflow Monitor in ADMIN mode. The buttons and their behavior are as follows:

- **Abort Process:** Calls `WF_ENGINE.AbortProcess` to abort the selected process and cancel any outstanding notifications. Prompts for a result to assign to the aborted process.
- **Suspend Process:** Calls `WF_ENGINE.SuspendProcess` to suspend the selected process.
- **Resume Process:** Calls `WF_ENGINE.ResumeProcess` to resume the selected suspended process to normal execution status.
- **Reassign:** Calls `WF_ENGINE.AssignActivity` to reassign the selected notification activity to a different performer. Prompts for a role name.
- **Expedite:** Calls `WF_ENGINE.HandleError` to alter the state of an errored activity, or to undo the selected activity and all other activities following it to rollback part of the process. Prompts you to select one of two values:
  - Skip, to skip the activity and assign it a specified result.
  - Retry, to reexecute the activity.
- **Attribute:** Lets you change the value of an item attribute.

### **Application-Controlled Access to the Workflow Monitor**

- **A calling application can launch a Web browser and pass a Workflow Monitor URL.**
- **You can monitor a specific item type and key through the following URLs:**
  - **Process Diagram URL**
  - **Notifications List URL**
  - **Activities List URL**



ORACLE

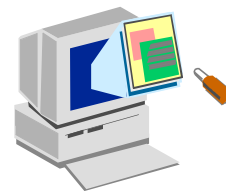
### **Workflow Monitor**

The Workflow Monitor requires a Web browser that supports Java Development Kit (JDK) Version 1.1.8 or higher and Abstract Windowing Toolkit (AWT), such as Netscape Communicator version 4.76 or a higher version of 4.7x, or Microsoft Internet Explorer version 5.0x or 5.5.

From an Oracle E-Business Suite form, use `FND_UTILITIES.OPEN_URL( )` to launch a browser and pass a Workflow Monitor URL.

### Application-Controlled Access to the Workflow Monitor

- To generate Workflow Monitor URL strings, call the following functions:
  - Process Diagram: `WF_MONITOR.GetDiagramURL()`
  - Notifications List: `WF_MONITOR.GetEnvelopeURL()`
  - Activities List:  
`WF_MONITOR.GetAdvancedEnvelopeURL()`
- The calling application must supply the Web agent string, item type, and item key, and specify whether to run the monitor in ADMIN or USER mode.



ORACLE

### Application-Controlled Access to the Workflow Monitor

#### Workflow Web Agent String

Use the function `wf_core.Translate('WF_WEB_AGENT')` to retrieve the `<webagent>` string used in creating the Find Processes and Monitor URLs.

#### Workflow Monitor URLs

The Workflow Monitor Process Diagram URL is composed as follows:

```
<webagent>wf_monitor.html?x_item_type=<item_type>
&x_item_key=<item_key>&x_admin_mode=<YES_or_NO>
&x_access_key=<access_key>
```

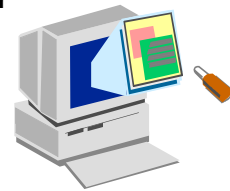
The Notifications List and Activities List URLs are similar to this URL.

#### User Authentication

The Workflow Monitor functions for access to the Process Diagram, Notifications List, and Activities List all return a hidden password in the URL string that provides the user access to the pages in either ADMIN or USER mode.

### Direct Access to the Workflow Monitor

- **The Find Processes web page in Oracle Workflow lets you search for processes you want to monitor.**
- **The Find Processes web page is a secured web page that requires user authentication.**
  - **If you have Workflow administrator privileges, then you can search for and monitor any workflow process instance in ADMIN mode.**
  - **If you do not have Workflow administrator privileges, then you can only search for and monitor processes that you own.**



ORACLE

### Direct Access to the Workflow Monitor

To access the Find Processes web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

`<webagent>/wf_monitor.find_instance`

Replace `<webagent>` with the base URL of the web agent configured for Oracle Workflow in your Web server.

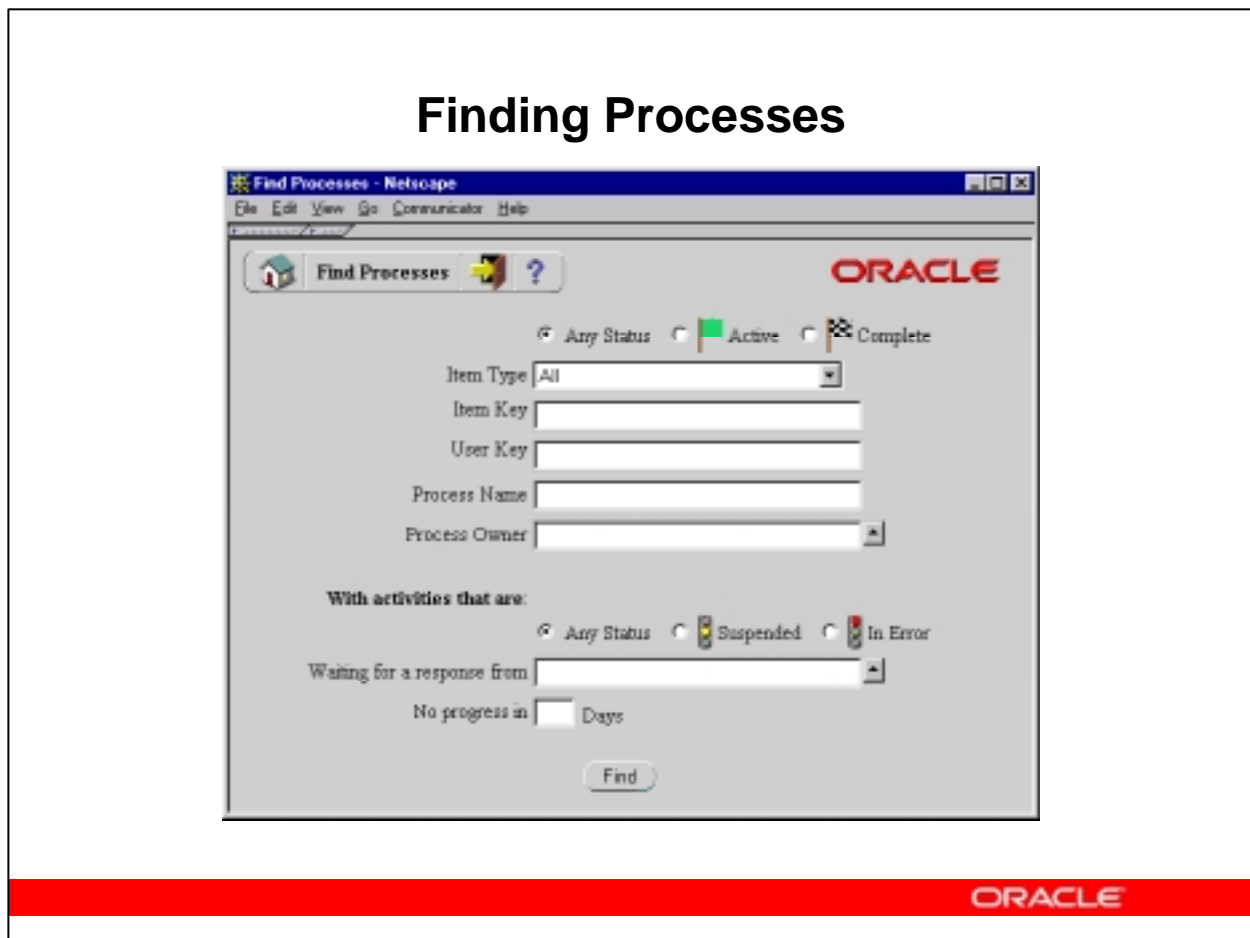
You can also access the Find Processes web page from the Oracle Workflow home page, `<webagent>/wfa_html.home`.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Processes option from a Workflow responsibility.

### Workflow Administrator Privileges

Users have workflow administrator privileges if they belong to the Oracle Workflow administration role, which is defined in the Global Workflow Preferences web page.

## Finding Processes



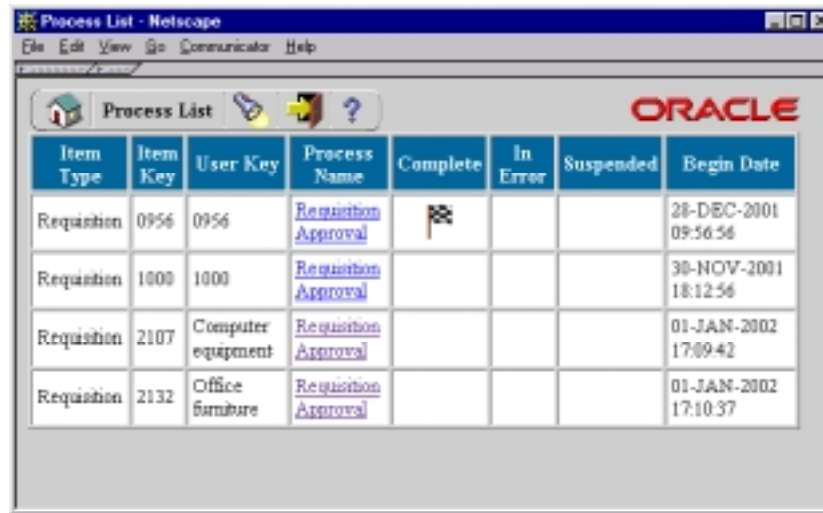
### Finding Processes

- The Find Processes Web page lets you specify the following search criteria to locate workflow process instances:
  - Process Status
  - Item Type
  - Item Key
  - User Key
  - Process Name
  - Process Owner (if you have workflow administrator privileges; otherwise you can only search for process instances that you own)
- You can also locate workflow process instances with activities that:
  - Have any status, are suspended, or in error
  - Are waiting for a response from a specified role
  - Have had no progress in a specified number of days
- Choose Find to locate the process instances that meet your specified criteria and display those processes in the Process List.



## Reviewing the Process List

### Reviewing the Process List



Item Type	Item Key	User Key	Process Name	Complete	In Error	Suspended	Begin Date
Requisition	0956	0956	<a href="#">Requisition Approval</a>	🚩			28-DEC-2001 09:56:56
Requisition	1000	1000	<a href="#">Requisition Approval</a>				30-NOV-2001 18:12:56
Requisition	2107	Computer equipment	<a href="#">Requisition Approval</a>				01-JAN-2002 17:09:42
Requisition	2132	Office furniture	<a href="#">Requisition Approval</a>				01-JAN-2002 17:10:37

### Reviewing the Process List

- The Process List provides a summary of all process instances that match your Find Processes search criteria.
- Process instances are listed in ascending order, first by item type, then by item key.
- The Process List summarizes the status of each process instance.
- Choose a Process Name link to display the Notifications List, which shows the notification activities that have been initiated for that process instance.

## Reviewing the Notifications List

### Reviewing the Notifications List

Done	Who	Activity	Started	Duration	Result
	<a href="#">KWALKER</a>	<a href="#">Notify Requisition Approval Required</a>	01-JAN-2002 17:10:37	1 Hours 57 Minutes	

ORACLE

### Reviewing the Notifications List

- The Notifications List displays all the current notifications that have been sent that require a Result response. It summarizes what each notification activity is, who it is assigned to, when it was sent, whether it has been completed, how many days have passed before completion and what the result is.

**Note:** If the process instance is in an error state and the cause of the error was from a notification, you can click the link in the Result column, if any, to display the cause of the error.

- Choose a user link in the Who column to send an e-mail to the user to whom a notification is assigned.
- Choose a notification activity link in the Activity column to view the full definition of a notification activity.
- If a notification is still open and requires a response, an icon will appear after the notification activity name. You can click this icon to navigate to the Notification Details page, where you can respond to the notification if you are logged in as the notification recipient or if you are logged in with administrator privileges.

- Choose Advanced Options to navigate to the Activities List web page, where you can specify advanced criteria to search for and display specific activities of interest for the process.
- Choose View Diagram to display the selected process instance in the Workflow Monitor for a graphical representation of the process status. When connected to the current session with workflow administrator privileges, the Workflow Monitor displays the process in ADMIN mode; otherwise the process is displayed in USER mode.

## Filtering Activities in the Activities List

### Filtering Activities in the Activities List

Status	Who	Parent Activity	Activity	Started*	Duration	Result
Active	<a href="#">KWALKER</a>	<a href="#">Notify Approver</a>	<a href="#">Notify Requisition Approval Required</a>	01-JAN-2002 17:10:37	1 Hours 59 Minutes	

### Filtering Activities in the Activities List

The Activities List lets you use different criteria to filter for specific activities of interest in the current process instance.

- Activity Status Options:
  - Active (includes Notified, Deferred, and Waiting states)
  - Complete
  - Error
  - Suspended
- Activity Types:
  - Response Notifications
  - FYI Notifications
  - Functions
  - Standard Workflow Items
  - Events

Choose Filter Activities to display the activities that match your criteria.

## Filtering Activities in the Activities List

### Filtering Activities in the Activities List



The screenshot shows the Oracle Activities List interface. At the top, there is a title bar 'Activities List - Netscape' and an 'ORACLE' logo. Below the title bar, there is a navigation bar with icons for Home, Search, and Help. The main content area displays the following information:

- Request Approval, Requestor: Office Services  
Started: 01-JAN-2002 (0 Days)
- Activity Status Options:  Active,  Complete,  Error,  Suspended
- Activity Type:  Response Notifications,  FYI Notifications,  Function Workflow Items,  Standard Workflow Items,  Event
- Filter Activities button

Status	Who	Parent Activity	Activity	Started*	Duration	Result
Active	Workflow Engine		Request Approval	01-JAN-2002 17:10:37	2 Hours 1 Minutes	
Complete	Workflow Engine	Request Approval	Select Approver	01-JAN-2002 17:10:37	0 Seconds	True
Complete	BLEWIS	Request Approval	Notify Requestor of Forward	01-JAN-2002 17:10:37	0 Seconds	
Complete	Workflow Engine	Request Approval	Record Requestor Forward	01-JAN-2002 17:10:37	0 Seconds	
Active	Workflow Engine	Request Approval	Notify Approver	01-JAN-2002 17:10:37	2 Hours 1 Minutes	
Active	KWALKER	Notify Approver	Notify Requestor Approval Request	01-JAN-2002 17:10:37	2 Hours 1 Minutes	

View Diagram button

### Filtering Activities in the Activities List

You can sort the Activities List based on any column by clicking the column heading.

- An asterisk (\*) to the left of the sort column title indicates an ascending sort order.
- An asterisk (\*) to the right of the sort column title indicates a descending sort order.
- Choose a user link in the Who column to send an e-mail to the user to whom a notification is assigned.
- Choose an activity link in the Activity column to view the full definition of that activity.
- If a notification activity is still open and requires a response, an icon appears after the notification activity name. You can click this icon to navigate to the Notification Details page, where you can respond to the notification if you are logged in as the notification recipient or if you are logged in with administrator privileges.
- Choose View Diagram to display the selected process instance in the Workflow Monitor for a graphical representation of the process status. When connected to the current session with workflow administrator privileges, the Workflow Monitor displays the process in ADMIN mode; otherwise the process is displayed in USER mode.

## Reviewing Parent/Child Instances

### Reviewing Parent/Child Instances

The screenshot displays the Oracle Activities List interface. The main content area shows a table of activities for a 'Survey - Master/Detail Process'. The table includes columns for Status, Who, Parent Activity, Activity, Started, Duration, and Error. The activities listed are 'Wait For Flow', 'Start Individual Requests', 'Start', and 'Survey - Master/Detail Process'. The 'Wait For Flow' activity is currently active, while the others are complete.

Status	Who	Parent Activity	Activity	Started	Duration	Error
Active	Workflow Engine	<a href="#">Survey - Master/Detail Process</a>	<a href="#">Wait For Flow</a>	01-JAN-2002 19:13:36	5 Seconds	
Complete	Workflow Engine	<a href="#">Survey - Master/Detail Process</a>	<a href="#">Start Individual Requests</a>	01-JAN-2002 19:13:35	0 Seconds	
Complete	Workflow Engine	<a href="#">Survey - Master/Detail Process</a>	<a href="#">Start</a>	01-JAN-2002 19:13:35	0 Seconds	
Active	Workflow Engine		<a href="#">Survey - Master/Detail Process</a>	01-JAN-2002 19:13:35	6 Seconds	


### Reviewing Parent/Child Instances

If the selected process is a member of a parent/child process, a parent/child hierarchy list appears in the left pane of the Activities List page. The hierarchy list shows links to related parent and child instances of the current process. Each link lets you navigate to the Notifications List for the selected parent or child instance.

## Testing Processes

---

### Testing Processes



The screenshot shows a Netscape browser window titled 'Initiate Workflow - WFPING'. The browser's address bar is empty. The page header includes the Oracle logo and a question mark icon. The form fields are as follows:

Item Key	TST01
User Key	Test_01
Process Name	Master Ping Process
Process Owner	SYSADMIN
To Agent	
Event Name	
Out Agent	
Event Key	
Event Message	

Buttons: OK, Cancel

---

**ORACLE**

### Testing Processes

You can use the Launch Processes web page to launch a workflow process for testing purposes.

To access the Launch Processes web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

`<webagent>/wf_initiate.ItemType`

Replace `<webagent>` with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Launch Processes web page from the Oracle Workflow home page, `<webagent>/wfa_html.home`.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Launch Processes option from a Workflow responsibility.

In the Launch Processes page, choose the item type you want to test. The Initiate Workflow page appears. In this page, specify:

- A unique item key for the process instance
- A user-defined key that you want to use to identify the process

- The name of the process to test
- An optional process owner
- Values for any required item type attributes

Choose OK. The Workflow Monitor Activities List page appears, displaying the activities for the process you launched. You can review the progress of the process in the Workflow Monitor and respond to any notifications using the Notifications Worklist.



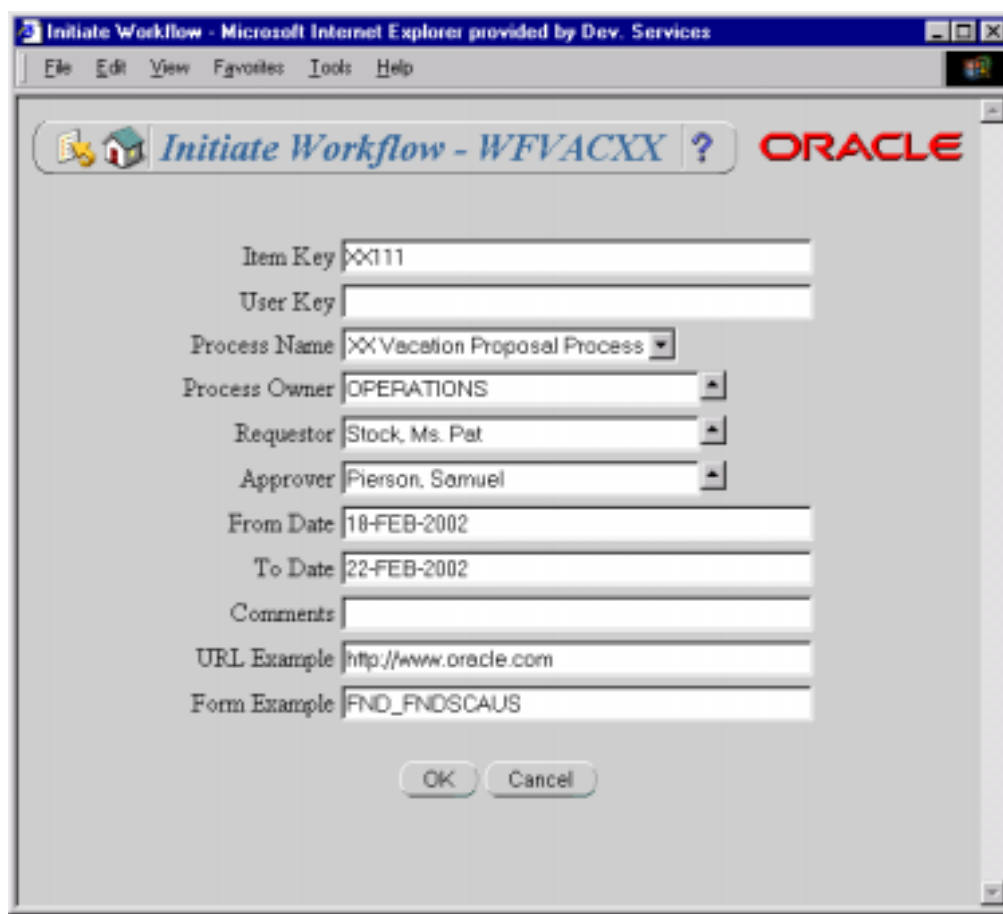
## Guided Demonstration - Workflow

---

### Run your Workflow

#### Responsibility = System Administration

1. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.
2. Choose the Launch Processes link.
3. In the Launch Processes page, choose your *XX* Vacation Proposal item type.
4. In the Initiate Workflow page, enter a unique item key such as *YourInitials111* in the Item Key field.
5. In the Process Owner field, select the role that you want to use as the requestor for the vacation proposal.
6. In the Process Name field, select your *XX* Vacation Proposal Process.
7. In the Requestor field, select the role that you want to use as the requestor for the vacation proposal.
8. In the Approver field, select the role that you want to use as the approver for the vacation proposal.
9. In the From Date field, enter the vacation from date.
10. In the To Date field, enter the vacation to date.
11. Leave the Comments field blank.



12. Choose OK. Oracle Workflow launches the workflow process with the values you entered.
13. Review the process activities in the Activities List page that appears. If an error occurred in your process, you can use the Result Exception link in the Activities List to determine the cause of the error.
14. Choose View Diagram to review the status of the process in the Workflow Monitor.
15. Log off and log in again as the approver.
16. Choose the Worklist link.
17. Open the Vacation Proposal notification sent by your process and review the notification message.

**Note:** If you attached the URL Example attribute to the Vacation Proposal message, Oracle Workflow displays the URL link at the end of the notification. If you attached the Form Example attribute to the Vacation Proposal message, Oracle Workflow displays an attached form icon at the end of the notification. If you are viewing the notification from a responsibility that has access to that form, you can click the form icon to drill down to the form. Otherwise, a message is displayed stating that your responsibility does not have access to the form.

18. Enter comments if you want and choose **Approve** or **Reject** to respond to the notification.
19. Log off and log in as the requestor.
20. Choose the **Worklist** link.
21. Open the **Vacation Approved FYI** or **Vacation Rejected FYI** notification sent by your process and review the notification message.
22. You can review the status of the process in the **Workflow Monitor** by choosing the **Find Processes** link from the **Workflow** home page and searching for the process with the **XX Vacation Proposal** item type and your item key.

# Guided Demonstration - Workflow Builder

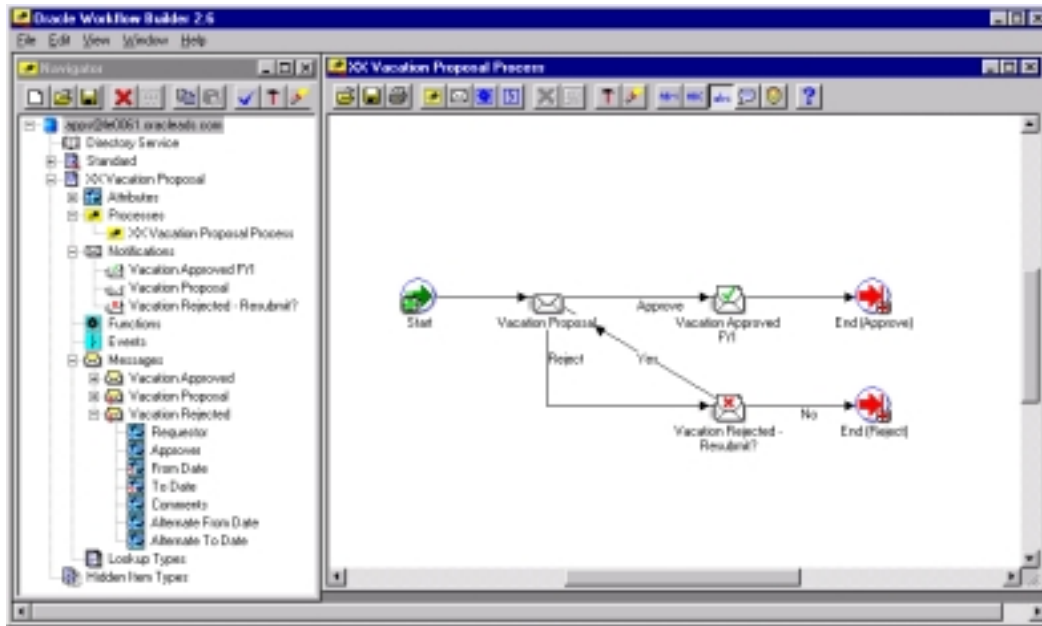
---

## Modify your Workflow

### Responsibility = System Administration

1. Start the Oracle Workflow Builder.
2. From the File menu, choose Open to open the wfvacXX.wft data store.
3. In the navigator tree, select your item type and select the Vacation Proposal message.
4. Open the property pages for the Vacation Proposal message and choose the Body tab. Within the Body tab, choose the HTML Body tab.
5. Enter an HTML-formatted version of your message. You can use the Import button to import a sample HTML body from the wfvacxx.html sample solution file on your file system. Choose OK.
6. In the Navigator window, click the “Verify” button to verify your workflow.
7. In the navigator tree, select your item type.
8. From the Edit menu, choose (M) New > Attribute.
9. Define the following properties for the item attribute:
  10. Internal Name: ALT\_FROM\_DATE
    - **Display Name:** Alternate From Date
    - **Type:** Date
    - **Format:** DD-MON-RRRR
  11. Click the “OK” button.
  12. From the Edit menu, choose (M) New > Attribute.
  13. Define the following properties for the item attribute:
    - **Internal Name:** ALT\_TO\_DATE
    - **Display Name:** Alternate To Date
    - **Type:** Date
    - **Format:** DD-MON-RRRR

14. Click the “OK” button.
15. Drag and drop the Alternate From Date and Alternate To Date item attributes onto the Vacation Proposal message to create the corresponding message attributes. Set the Source field for both alternate date attributes to Respond.
16. Open the property pages for the Vacation Rejected message. Choose the Body tab and add the alternate from and to dates into the message body.
17. Drag and drop the Alternate From Date and Alternate To Date item attributes onto the Vacation Rejected message to create the corresponding message attributes. The Source field for both alternate date attributes should be Send.
18. In the property pages for the Vacation Rejected message, choose the Result tab. In the Display Name field for the Result, enter "Resubmit Vacation Proposal?". In the Lookup Type field, select Yes/No.
19. Ensure that the Vacation Rejected message has From Date and To Date message attributes. Set the Source field to Respond for these attributes.
20. Open the process diagram window.
21. Double-click the Vacation Rejected FYI notification activity node. Change the display name for the activity to "Vacation Rejected - Resubmit?". Change the result type for the activity to Yes/No to match the message result.
22. Select the transition that connects the Vacation Proposal notification and the Vacation Rejected - Resubmit? notification and drag the transition to create a vertex point. Select the transition again and right-click it. Choose the Locked option from the menu that appears. In this way, you can avoid drawing a new transition on top of this one.
23. Delete the existing transition between the Vacation Rejected - Resubmit? node and the End node. Draw a new transition from the Vacation Rejected - Resubmit? node to the End node and select No from the transition results menu.
24. Draw a new transition from the Vacation Rejected - Resubmit? node to the Vacation Proposal node and select Yes from the transition results menu.
25. In the Navigator window, click the Verify button to verify your workflow. Because you have defined all the underlying components for your process, the Workflow Builder should not display any warnings. Click OK.
26. From the File menu, choose Save to save your work to your workflow definition file.
27. From the File menu, choose Save As and save your item type to the class database, using the database username, password, and connect string provided by the instructor.



28. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.
  
29. Use the Launch Processes page to launch your workflow process and test your work. You can use the Notifications Worklist to view the notifications sent by the process and use the Workflow Monitor to review the status of the process. As the approver, reject the initial vacation proposal. Then, as the requestor, respond to the Vacation Rejected – Resubmit? notification by submitting new vacation dates.

### Summary

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

- **Check the progress of a workflow using the Workflow Monitor.**
- **Launch a test process.**

ORACLE





# Purging Workflow Data

## Chapter 23

# Purging Workflow Data

ORACLE

### **Objectives**

**After completing this lesson, you should be able to purge obsolete Workflow runtime data.**



ORACLE

### Purging Workflow Data

- **Oracle Workflow accesses several tables that can grow quite large with obsolete workflow information that is stored for all completed workflow processes.**
- **The size of these tables and indexes can adversely affect performance.**
- **You should purge these tables should be purged on a regular basis.**
  - Purge APIs
  - Purge Obsolete Workflow Runtime Data concurrent program (Oracle E-Business Suite only)

ORACLE

### Purging Workflow Data

The persistence type of an item type controls when Oracle Workflow purges runtime status information for work items. The persistence values are:

- Temporary: Item will be deleted in  $n$  days
- Permanent: Item will be deleted only when forced

**Note:** For a work item to be considered eligible for purging, all activities must be complete. This includes FYI notifications, which must be closed.

### Workflow Purge APIs

- **Purge APIs delete obsolete runtime data and activity versions that are no longer in use.**
- **The most commonly used WF\_PURGE APIs include:**
  - **Items:** Purge all runtime data associated with completed items, their processes, and notifications sent by them.
  - **Activities:** Purge obsolete versions of activities that are no longer in use by any item.
  - **Total:** Purge both item data and activity data.
  - **AdHocDirectory:** Purge users and roles in the WF\_LOCAL\_\* tables whose expiration date has elapsed and that are not referenced in any notification.

ORACLE

#### WF\_PURGE APIs

Many of the Purge APIs accept the following parameters:

- **Item Type:** The item type associated with the obsolete runtime data you want to delete. Leave this parameter null to delete obsolete data for all item types.
- **Item Key:** A string generated from the application object's primary key. The string uniquely identifies the item within an item type. Leave this parameter null to purge all items in the specified item type.
- **End Date:** A specified date to delete up to.

**Note:** The WF\_PURGE APIs only purge data associated with Temporary item types whose persistence, in days, has expired. A persistence type PL/SQL variable is set to 'TEMP' (Temporary) by default and should not be changed.

Use the WF\_PURGE.TotalPERM API to delete all eligible obsolete runtime data associated with item types of with a persistence type of Permanent.

### **Purge Obsolete Workflow Runtime Data Concurrent Program**

- **In Oracle E-Business Suite, use the concurrent program Purge Obsolete Workflow Runtime Data (short name FNDWFPR) for purging.**
- **The system administrator should add this concurrent program to the security group for the responsibility from which you want to run the program.**
- **Use the Submit Requests form to run the concurrent program.**
- **Supply the following parameters:**
  - **Item Type**
  - **Item Key**
  - **Age**
  - **Persistence Type**

ORACLE

#### **Concurrent Program Parameters for Purging**

- **Item Type** - Item type associated with the obsolete runtime data you want to delete. Leave null to delete obsolete data for all item types.
- **Item Key** - String generated from the application object's primary key. The string uniquely identifies the item within an item type. If null, purges all items in the specified item type.
- **Age** - Minimum age of data to purge, in days if Persistence Type is set to 'Temporary'. Default is 0.
- **Persistence Type** - Persistence type to be purged, either 'Temporary' or 'Permanent'. Default is 'Temporary'.

## Guided Demonstration - Purging Workflow Data

---

1. In SQL\*Plus, review the contents of a Workflow runtime table such as WF\_ITEM\_ACTIVITY\_STATUSES.
2. In the System Administrator responsibility, navigate to the Submit Requests form and submit the Purge Obsolete Workflow Runtime Data concurrent program with appropriate parameters to purge a particular item type.
3. In SQL\*Plus, run the WF\_PURGE.Total API with appropriate parameters to purge a particular item type.
4. In SQL\*Plus, review the contents of the WF\_ITEM\_ACTIVITY\_STATUSES table again to show that the data has been purged.

### Summary

**In this lesson, you should have learned how to purge obsolete Workflow runtime data.**

ORACLE



# Overview of Oracle Alert

## Chapter 24

# Overview of Oracle Alert

ORACLE

### **Course Objectives**

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

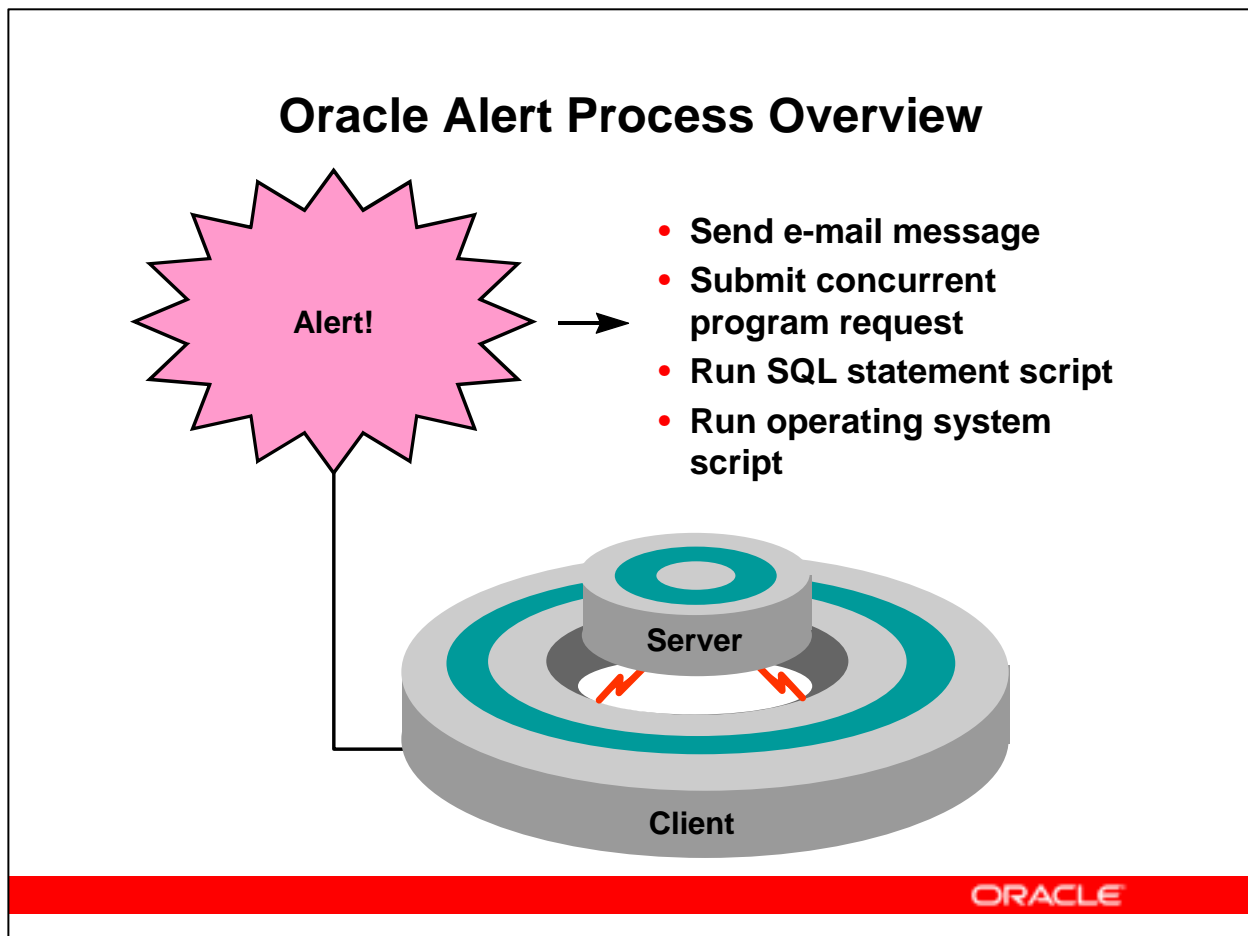
- **Create, run, and test alerts**
- **Plan for implementation of Oracle Alert**

ORACLE

### **Introduction**

This course provides instructions for creating exception control solutions for Oracle Applications. Participants will participate in discussions on how they will use Oracle Alert in the workplace.

## Oracle Alert Process Overview



### Oracle Alert Process Overview

How do you find out about important or unusual activity in your database? How do you stay aware of regular, yet critical database events without sorting through lengthy reports?

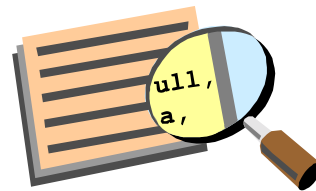
Alerts monitor your database information and notify you when the condition you have specified is found. You can define Oracle alerts in any Oracle application or custom Oracle application.

You can define one of two types of alerts: event and periodic.

- Event alert - Notifies you of activity in your database as soon as it occurs
- Periodic alert - Checks the database for information according to a schedule that you define

### What Is an Exception?

- An exception is a specified condition found during an alert check.
- For example, an alert checking for users who did not change their passwords within the designated time finds five users that meet the criteria. Each user found is an **exception**.



ORACLE

### Examples of Event Alerts

- **Requisition receipt: Informs requester that Purchasing received/entered the request**
- **Purchase approval: Informs manager that a purchase order needs to be approved**
- **Database monitoring: Informs DBA when database tables need more space and allocates space if response indicates to do so**

ORACLE

### What Event Alerts Specify

By creating event alerts, you can have an immediate view of the activity in your database, thus keeping up with important or unusual events as they happen.

When you create an event alert, you specify the following:

- A database event that you want to monitor—that is, an insert or an update to a specific database table
- A SQL SELECT statement that retrieves specific database information as a result of the database event
- Actions that you want Oracle Alert to perform as a result of the database event

### Examples of Periodic Alerts

- **Personnel:** Shows all employees that have terminated in the last six months (monthly)
- **Payroll:** Shows current balance and vacation reported by month (monthly)
- **Purchasing:** Detects creation or edit of vendor with nonstandard payment terms (weekly)
- **Inventory:** Shows when quantity on hand minus quantity ordered is less than reorder quantity (weekly)

ORACLE

### What Periodic Alerts Specify

By creating periodic alerts, you can have current measurements of staff and organization performance, so that you can focus on potential trouble spots. You can automate routine transactions.

When you create a periodic alert, you specify the following:

- A SQL SELECT statement that retrieves specific database information
- The frequency with which you want the periodic alert to run the SQL statement
- Actions that you want Oracle Alert to perform once it runs the SQL statement

### **Action Types**

**An action is an event that occurs once a monitored database event occurs or once a periodic checking of the database has been performed. Oracle Alert can perform these actions:**

- **Send an e-mail message**
- **Submit a concurrent program request**
- **Run a SQL statement script**
- **Run an operating system script**

ORACLE

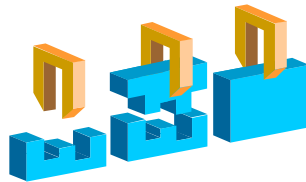
#### **Four Action Types**

- Messages
  - UNIX Sendmail
  - VMS mail
  - Any MAPI-compliant Windows NT mail application
- Concurrent program request: supply arguments
- SQL statement script
  - Note: The only tables that you can write to directly are custom application tables and open interface tables.
- Operating system script



### Action Levels

- **Detail:** Perform the action for each occurrence of the condition.
- **Summary:** Perform the action for a group of occurrences of the condition.
- **No Exception:** Perform the action when nothing in the database meets the search criteria.



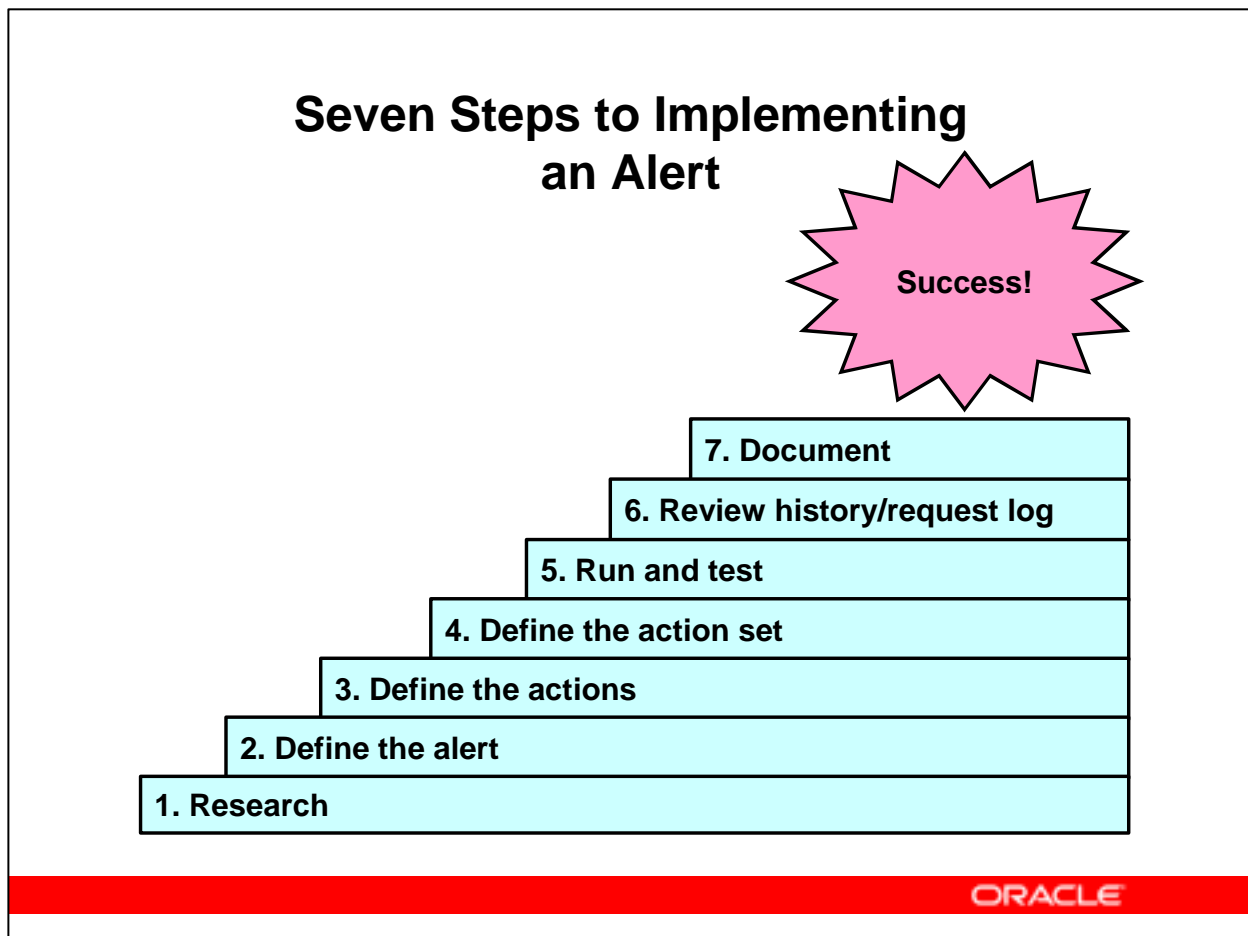
ORACLE

### Example

Inform the account manager of invoices on hold

- **Detail Action:** Send a separate e-mail message for each invoice that meets the search criteria of invoices on hold.
- **Summary Action:** Send a single e-mail message listing all invoices that meet the search criteria, or send one summary for each vendor.
- **No Exception Action:** Send an e-mail message stating that nothing in the database is on hold.

## Seven Steps to Implementing an Alert



### How to Create an Alert

- 1 Research all the information needed to create your alert.
  - E-mail addresses
  - Required parameters for concurrent programs
  - Table names
  - Column names
  - Application name that owns the table
- 2 Define the alert condition.
  - Name your alert.
  - Select the type.
  - Define event or periodic details.
  - Write your SELECT statement.
  - Verify the SELECT statement.
  - Run the SELECT statement (if a periodic alert).

- 3 Define the actions.
  - Select action level.
  - Select action type.
- 4 Define the action set.
  - Create the sequence of actions.
  - Define what to do when an actions fails.
- 5 Run and test your alert by triggering your alert.
  - Event alerts: perform the event
  - Periodic alerts: use Request—>Check
- 6 Review the alert history.
  - Find checks.
  - Find exceptions.
  - Find actions.
  - View request log.
7. Document the alert.
- 8.

**Refer to Periodic Alert Demo [DEM00006]**



# **Identifying Implementation Issues and Responsibilities for Oracle Alert**

**Chapter 25**

**Identifying Implementation Issues and  
Responsibilities for Oracle Alert**

ORACLE

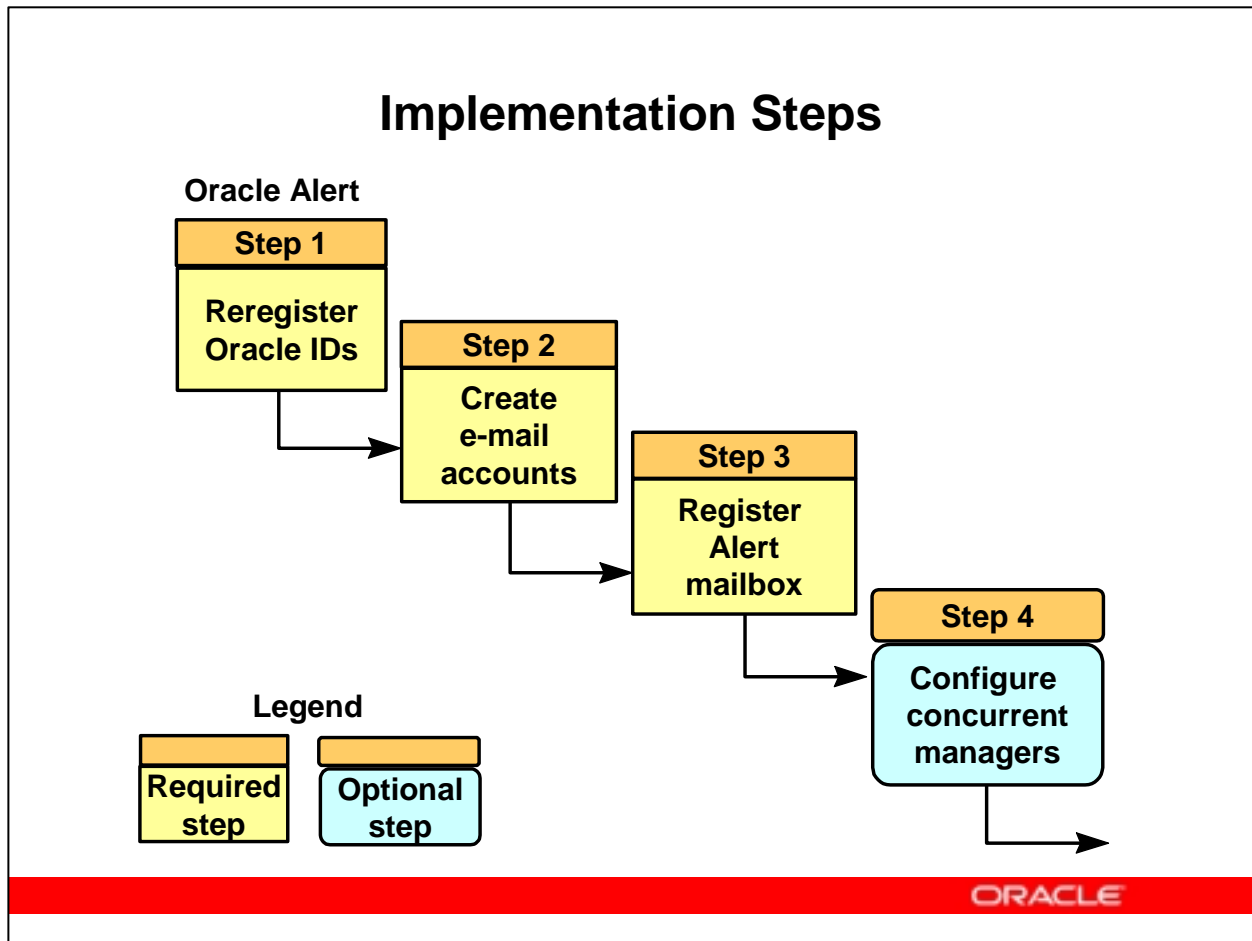
### Objectives

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

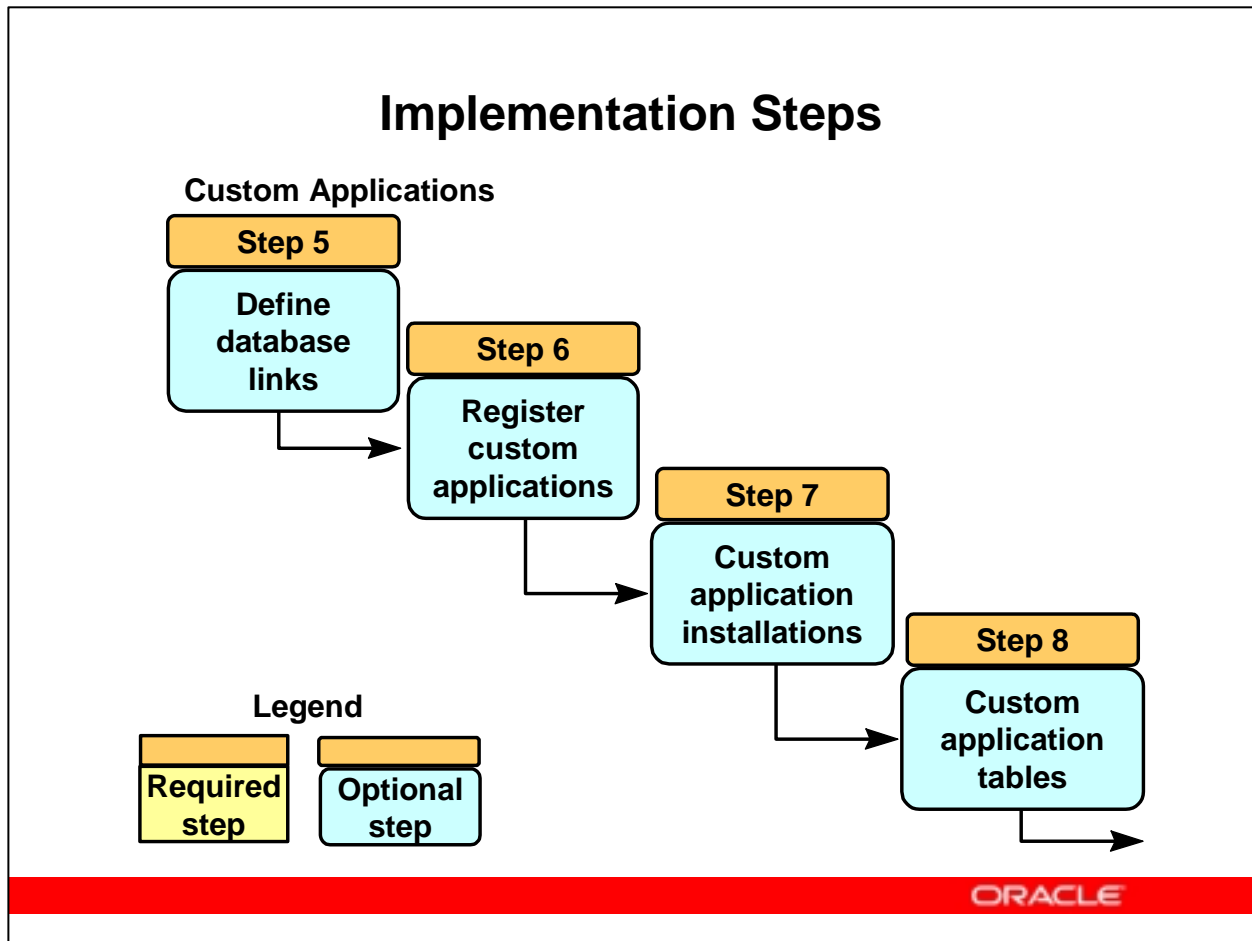
- **Plan a successful implementation of Oracle Alert**
- **Configure Oracle Alert for your business**
- **Identify your business needs and map them to responsibilities**
- **Determine whether you can use the predefined Oracle Alert Manager responsibility or create a new application Alert Manager responsibility**

ORACLE

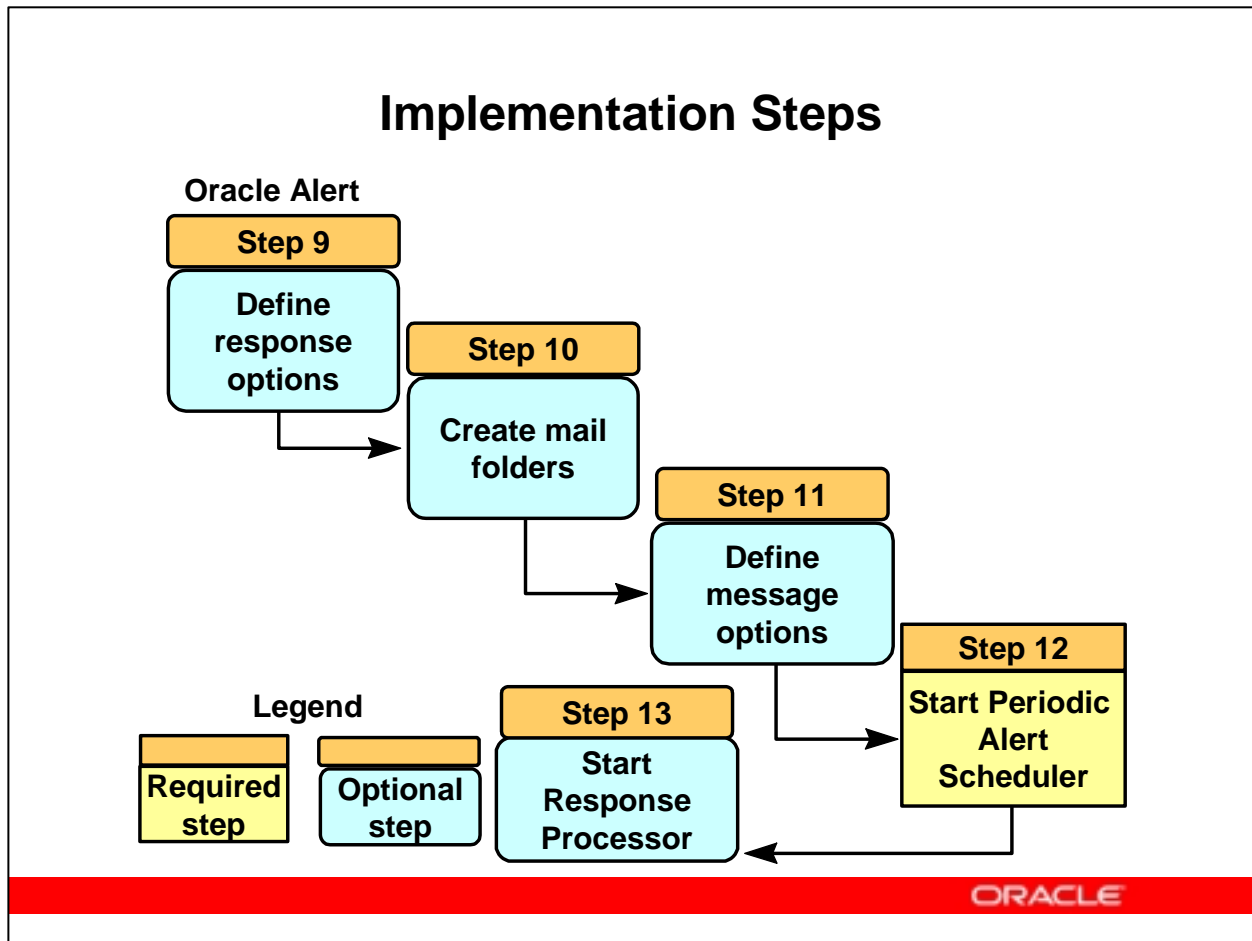
# Implementation Steps







# Implementation Steps



### Implementation Terminology

Term	Definition
<b>Custom application</b>	<b>Any Oracle Application residing in the same database as Oracle Alert that was not installed with Rapid Install</b>  <b>or</b> <b>Any custom Oracle application residing in the same database as Oracle Alert</b>  <b>or</b> <b>Any application residing in a different database from Oracle Alert</b>

ORACLE

### Implementation Terminology

Term	Definition
Database link	An object stored in the local database that identifies a remote database, a communication path to that database, and optionally, a username and password
Application installation	A unique combination of an application and an Oracle ID

ORACLE

### **Step 1: Reregister Oracle IDs**

**After installing Oracle Alert, reregister any Oracle IDs that were not registered during installation and for which you want to create alerts.**

- **Include any custom Oracle IDs for which you want to create alerts.**
- **You do not need to reregister the APPLSYS Oracle ID.**
- **The Concurrent Manager must recognize these Oracle IDs.**
- **Oracle Alert must recognize these Oracle IDs in order to check alerts based on tables that reside outside its own Oracle ID.**



ORACLE

### Step 1: Reregister Oracle IDs

Use the Oracle Users window to enter:

- Database User Name
- Password
- Privilege
- Install Group
- Description

The Concurrent Manager executes the Grant and Synonym scripts for the Oracle IDs.

**System Administrator: (N) Security—>ORACLE—>Register**

ORACLE

### **Step 1: Reregister Oracle IDs**

**Key implementation questions:**

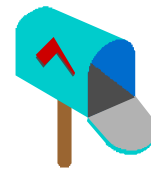
- **Are all Oracle IDs for which you want to create alerts registered?**
- **When you create new Oracle IDs, have you registered them?**
- **Have you registered all Oracle IDs for custom applications that will be using alerts?**

ORACLE

### **Step 2: Create Electronic Mail Accounts**

**Create an electronic mail account from which Oracle Alert can send mail.**

- **Define at least one electronic mail account as the Oracle Alert mailbox. You may want to name this account ALERT.**
- **This mail account is the sender of messages for alert message actions.**



ORACLE



### **Step 2: Create Electronic Mail Accounts**

**You can define the account to send messages in any of the following mail systems:**

- **UNIX Sendmail**
- **VMS mail**
- **MAPI-compliant Windows NT mail application**
- **Any mail system supported by UNIX gateways**



ORACLE

### **Step 2: Create Electronic Mail Accounts**

**Define accounts to send and receive responses if you want to use response processing.**

- **Oracle Alert supports both UNIX Sendmail and Windows NT MAPI-compliant mail applications for response processing.**
- **You can use either the same account or different accounts to send messages and to process responses.**
- **Set up either one central response account, or a separate response account for each application.**



ORACLE

### **Step 2: Create Electronic Mail Accounts**

**Key implementation questions:**

- **Is your e-mail application properly installed?**
- **Are the necessary gateways in place?**
- **Will you use response processing?**
- **Will you use response processing with more than one application?**
- **Do you need to enable decentralized administration of the response mail?**
- **Is response mail security an issue?**



ORACLE

### **Step 3: Register Oracle Alert Mailboxes**

**Register the electronic mail accounts you want to use with Oracle Alert.**

- **Identify the appropriate command to invoke the electronic mail system.**
- **Determine the number of mail accounts you want to use.**
- **Oracle Alert automatically displays two default mail accounts, the Send Mail Account and the Response Mail Account.**
- **You can define as many additional application-specific response accounts as you need.**
- **Identify the username and password of each mail account.**

**ORACLE**

### Step 3: Register Oracle Alert Mailboxes

Use the Mail Systems region of the Oracle Alert Options window to enter:

- Name
- Command
- Parameters
- In Use

(N) System—>Options

ORACLE

## Step 3: Register Oracle Alert Mailboxes

---

### Step 3: Register Oracle Alert Mailboxes

Use the Mail Server Options region of the Oracle Alert Options window to enter:

- Mail Database/Server Name
- Accounts
  - Application
  - Installation
  - User Name
  - Password

(N) System—>Options

ORACLE

### **Step 4: Configure Concurrent Managers**

**Your system administrator should configure concurrent managers to optimize event alert request handling.**

- **Define one concurrent manager to run only the Check Event Alert program (ALECTC).**
- **Define all other concurrent managers to run all other programs (except the Check Event Alert program).**

ORACLE

### **Step 4: Configure Concurrent Managers**

#### **Key implementation questions:**

- **Will you be defining many event alerts?**
- **Do you need event alerts to be processed immediately once an event is triggered?**
- **Can you meet your business needs using a periodic alert and :date\_last\_checked?**

ORACLE



### **Step 5: Define Database Links**

**Create database links for applications where necessary.**

- **Define a database link if you want to define periodic alerts for an application that resides in a database outside the database where Oracle Alert is installed. (To define event alerts for an application, the application must reside in the same database where Oracle Alert is installed.)**
- **Define a database link for an Oracle Application if that application is installed in a different database than Oracle Alert.**

ORACLE

### **Step 5: Define Database Links**

- **Define a database link for an Oracle Application if that application is installed in the same database as Oracle Alert, but in a different Oracle Application Object Library context than Oracle Alert.**

ORACLE

### **Step 5: Define Database Links**

**Key implementation questions:**

- **Do you want to use Oracle Alert with an application on a machine where Oracle Alert is not installed?**
- **Do you want to use Oracle Alert with an application on a database where Oracle Alert is not installed?**

ORACLE

### Step 5: Define Database Links

- **Syntax**

```
CREATE DATABASE LINK linkname  
CONNECT TO username IDENTIFIED BY password  
USING 'connect string';
```

- **Example**

```
CREATE DATABASE LINK BUGDB  
CONNECT TO SCOTT IDENTIFIED BY TIGER  
USING 'T:WRVMS:V60BUG';
```

ORACLE

### **Step 6: Register Custom Applications**

**Register any custom applications you want to use with Oracle Alert.**

- **You must register your custom application to tell Oracle Alert that your application exists.**
- **Custom applications include:**
  - **Oracle applications not installed using Rapid Install**
  - **Custom Oracle applications**
  - **Any applications residing in a different database from Oracle Alert**

**ORACLE**

### Step 6: Register Custom Applications

Use the Applications window to enter:

- **Application**
- **Short Name**
- **Basepath: The environment variable that represents the basepath of your custom application so that Oracle Alert can locate your SQL statement script and operating system script external files**
- **Description**

**(N) System—>Applications**

ORACLE

### **Step 7: Define Custom Application Installations**

**If you registered a custom application in the previous step, you must also register the custom application's installation so that Oracle Alert knows which Oracle ID to connect to.**

- **Register applications that reside in the same database as Oracle Alert and that are not installed with Rapid Install.**
- **Register installations of applications that reside in databases other than Oracle Alert's database.**
- **When you register a custom application that resides in another database, use Oracle Alert's Oracle ID.**

The Oracle logo is displayed in white capital letters on a red horizontal bar.

### **Step 7: Define Custom Application Installations**

**Use the Define Application Installations window to enter:**

- **Application**
- **Oracle Username**
- **Status**

**(N) System—>Installations**

**ORACLE**



## Step 8: Register Custom Application Tables

---

### Step 8: Register Custom Application Tables

If you have a custom application with which you want to define event alerts, and you registered the application and application installation in the previous steps, you need to register the tables and columns in your custom application by calling the `AD_DD.register_table( )` and `AD_DD.register_column( )` APIs.



ORACLE

### Step 8: Register Custom Application Tables

See Table Registration API, *Oracle Applications Developer's Guide*.

### **Step 8: Register Custom Application Tables**

- **This step is required for defining event alerts for non-Oracle applications that reside in the same database as Oracle Alert.**
- **This step informs Oracle Alert of the names of your custom application tables so that they appear in the of values for the Table field in the Alerts window for event alerts.**



ORACLE

### **Step 9: Define Response Options**

**Control how Oracle Alert processes responses by defining response options. Note that the response accounts and response mailboxes must already be defined as indicated in Steps 2 and 3.**

- **The Response Processor runs every 24 hours at midnight by default if these options are left blank:**
  - Response Processor Start Time
  - Response Processor End Time
  - Response Processor Interval
- **You can change the Response Processor Interval to process mail more frequently.**
- **If the Response Processor gets off schedule, it will always run at the next scheduled interval.**

**ORACLE**

### **Step 9: Define Response Options**

**Use the Response Processing region of the Oracle Alert Options window to enter:**

- **Enable Response Processing**
- **Check for Open Responses Before Reading Response Accounts**
- **Response Processor Start Time**
- **Response Processor End Time**
- **Response Processor Interval**
- **Oracle Alert Installation Number**

**(N) System—>Options**

**ORACLE**

### **Step 9: Define Response Options**

**Key implementation questions:**

- **Do you want to use response processing?**
- **How often do you need responses to be processed?**

ORACLE

### **Step 10: Create Mail Folders for Response Accounts**

Create two new folders, **Reviewed** and **Reviewed\_OK**, in your electronic mail response account.

- Create these two folders in every electronic mail response account that you define in the Oracle Alert Options window.
- Oracle Alert uses these folders to manage the responses it receives for different alert messages.
- Consult your electronic mail application documentation to learn how to create the folders.



ORACLE

### **Step 11: Define Message Options**

**Create generic message elements that Oracle Alert appends to the alert messages it sends.**

- **Define the Message Action Header and Message Action Footer to save time when defining message actions.**
- **Define the Response Text to clearly communicate reply instructions to message recipients.**
- **Define the Returned Message Header to let respondents know that their reply to an alert message was not understood and that they must reply again.**
- **Oracle Alert provides a generic version of the Response Text and Returned Message Header that you can use or edit.**

ORACLE

### **Step 11: Define Message Options**

**Oracle Alert inserts the generic message elements in the following positions:**

- **The Message Action Header is displayed at the beginning of the message text of every message.**
- **The Message Action Footer is displayed at the end of the message text—but before the response text—of every message.**
- **The Response Text is displayed at the end of each alert message that solicits a response.**
- **The Returned Message Header is displayed at the beginning of every Return Original Message invalid response message action.**

**ORACLE**



### **Step 11: Define Message Options**

Use the **Message Elements** region of the **Oracle Alert Options** window to enter:

- **Message Action Header**
- **Message Action Footer**
- **Response Text**
- **Returned Message Header**

**(N) System—>Options**

ORACLE

### **Step 12: Start the Periodic Alert Scheduler**

**After you have successfully installed Oracle Alert, you must verify that the Periodic Alert Scheduler is active to enable automatic periodic alert submission.**

- **Select Requests from the View menu to find requests and display the Requests window.**
- **You should see a concurrent request for the program name Periodic Alert Scheduler. Its phase should be Pending.**
- **If the phase of the Periodic Alert Scheduler is not Pending, start the program from the Schedule Alert Programs window.**

ORACLE

### **Step 12: Start the Periodic Alert Scheduler**

**Use the Schedule Alert Programs window to:**

- **Select the Periodic Alert Scheduler program.**
- **Click the Activate button.**

**Never deactivate the Periodic Alert Scheduler.**

**(N) Request—>Schedule**

**ORACLE**

### **Step 12: Start the Periodic Alert Scheduler**

**The Periodic Alert Scheduler runs automatically every day at midnight and performs the following actions:**

- **Purges expired history**
- **Submits periodic alerts that are scheduled to be checked that day**
- **Resubmits itself to run for the next day**

ORACLE

### **Step 13: Start the Response Processor**

**After you have successfully installed Oracle Alert, start the Response Processor so that Oracle Alert automatically reads and processes the mail in the response account.**

- **The Response Processor also submits itself once for each response account that you define in the Oracle Alert Options window.**
- **The Response Processor runs automatically every day at midnight, or according to the schedule you define in your response options.**

ORACLE

### **Step 13: Start the Response Processor**

**Use the Schedule Alert Programs window to:**

- **Select the Response Processor program**
- **Click the Activate button**

**(N) Request—>Schedule**

**ORACLE**

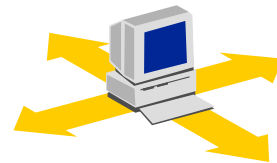
### **Step 13: Start the Response Processor**

- **Each time the Response Processor runs, it moves mail from the Inbox folder to the Reviewed folder of the response account.**
- **The Response Processor then reads the messages and does the following:**
  - **Acts on valid responses, then moves those messages to the Reviewed\_OK folder**
  - **Acts on invalid responses, and leaves those messages in the Reviewed folder**
  - **Acts on a lack of response if the follow-up period has expired**

ORACLE

### Planning Responsibilities for Oracle Alert

- **Identify your business needs:**
  - Who will be defining alerts?
  - Will the user writing alert **SELECT** statements be different from the user modifying alert message actions?
- **Will you use response processing?**
  - If so, do you want to decentralize administration of the mail accounts?
  - Will you divide the Alert responsibility by application?



ORACLE



### Oracle Alert Manager Responsibility

The standard Oracle Alert Manager responsibility includes:

- All alert windows
- All action windows
- Response set windows
- All history windows
- All definition windows
- Access to data and history for all applications
- Permission to submit periodic alerts on demand for any application



ORACLE

### **Defining an Application Alert Manager Responsibility**

**Use the Responsibilities window to define new Alert Manager responsibilities if you want to divide your alert responsibilities by application.**

**System Administrator: (N) Security—>Responsibility—>Define**

ORACLE

### **Application Alert Manager Responsibility**

**An Application Alert Manager responsibility includes:**

- **All alert windows**
- **All action windows**
- **Response set windows**
- **All history windows**
- **The Oracle Alert Options window with access to the response account for this application**
- **Access to data and history for only this application**
- **Permission to submit periodic alerts on demand for this application**



**ORACLE**

### Summary

- **After Alert is installed, you need to reregister any Oracle IDs that were not registered during the Alert installation.**
- **Create at least one electronic mail account to originate alert messages.**
- **If you are using response processing, you must set up the accounts through which Oracle Alert can process responses to alert messages.**
- **To optimize alert request handling, define one concurrent manager to run only the Check Event Alert program and other concurrent managers to run all other programs.**
- **Create an implementation plan that considers whether any of the optional implementation steps are needed for your installation.**

ORACLE