Oracle® Applications

System Administrator's Guide - Security

Release 11*i*

Part No. B13923-03

December 2005

Oracle Applications System Administrator's Guide - Security, Release 11*i*

Part No. B13923-03

Primary Author: Robert Farrington, Mildred Wang

Contributing Author: George Buzsaki, Anne Carlson, Steve Carter, Siu Chang, Michelle Cheng, Kalyani Venkat Chervela, Kenny Tak Chi Ching, Charles Colt, Killian Evers, Chirag Mehta, Tom Morrow, Srikanth Sallaka, Peter Pil Seo, Jan Smith, Susan Stratton, Sanjeev P. Topiwala, Roger Wigenstam

Contents

5 User and Data Auditing

A Security Configuration and Maintenance

Index

Send Us Your Comments

Oracle Applications System Administrator's Guide - Security, Release 11*i*

Part No. B13923-03

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appsdoc_us@oracle.com
- FAX: 650-506-7200 Attn: Oracle Applications Technology Group Documentation Manager
- Postal service:
  Oracle Applications Technology Group Documentation Manager
  Oracle Corporation
  500 Oracle Parkway
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Intended Audience

Welcome to Release 11*i* of the *Oracle Applications System Administrator's Guide - Security*.

This guide assumes you have a working knowledge of the principles and customary practices of your business area. If you have never used Oracle Applications we suggest you attend one or more of the Oracle Applications System Administration training classes available through Oracle University. (See Other Information Sources for more information about Oracle training.)

This guide also assumes you are familiar with the Oracle Applications graphical user interface. To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See Other Information Sources for more information about Oracle Applications product information.

See Related Documents on page xii for more Oracle Applications product information.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/ .

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

Related Documents

You can choose from many sources of information, including online documentation, training, and support services to increase your knowledge and understanding of Oracle Applications system administration.

If this guide refers you to other Oracle Applications documentation, use only the Release 11i versions of those guides.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **PDF Documentation** - See the Oracle Applications Documentation Library CD for current PDF documentation for your product with each release. The Oracle Applications Documentation Library is also available on Oracle*MetaLink* and is updated frequently.
- **Online Help** - Online help patches (HTML) are available on Oracle*MetaLink*.
- **About Documents** - Refer to the About document for the mini-pack or family pack that you have installed to learn about feature updates, installation information, and new documentation or documentation patches that you can download. About documents are available on Oracle*MetaLink*.

Related Guides

You can read the guides online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Documentation Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at http://oraclestore.oracle.com.

Guides Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release of Oracle Advanced Product

Catalog (and any other Oracle Applications products). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11*i*. It provides a useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user guides and implementation guides.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11*i*. You cannot upgrade to Release 11*i* directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, Auto Patch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle Applications file system and database.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle Forms Developer forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms.

Other Implementation Documentation

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11*i*. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

Multiple Reporting Currencies in Oracle Applications

If you use the Multiple Reporting Currencies feature to record transactions in more than one currency, use this manual before implementing Oracle Applications. This manual details additional steps and setup considerations for implementing Oracle Applications with this feature.

Multiple Organizations in Oracle Applications

This guide describes how to set up and use Oracle Applications' Multiple Organization support feature, so you can define and support different organization structures when running a single installation of Oracle Applications.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how Oracle Applications users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup, and reference information for the Oracle Applications implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This guide also provides information on creating custom reports on flexfields data.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*MetaLink*.

Oracle Applications Message Reference Manual

This manual describes Oracle Applications messages. This manual is available in HTML format on the documentation CD-ROM for Release 11*i*.

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle Applications and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization's structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Applications working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle Database, and your hardware and software environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

1 Introduction

Access Control in Oracle Applications

This release of Oracle Applications provides significant enhancements to the Oracle Applications security system. Core Security now includes a Role Based Access Control model that builds on the existing Function Security and Data Security models. A new set of administrative features that build on Core Security are also introduced in this release.

Oracle User Management

Oracle User Management is a secure and scalable system that enables organizations to define administrative functions and manage users based on specific requirements such as job role or geographic location. With Oracle User Management, instead of exclusively relying on a centralized administrator to manage all its users, an organization can create local administrators and grant them sufficient privileges to manage a specific subset of the organization's users. This provides the organization with a more granular level of security, and the ability to make the most effective use of its administrative capabilities.

Oracle's function and data security models constitute the base layers of this system, and contain the traditional system administrative capabilities. Organizations can optionally add more layers to the system depending on the degree of flexibility they require.

Key features of Oracle User Management include:

- **Role Based Access Control (RBAC)** - Enables organizations to create roles based on specific job functions, and to assign these roles the appropriate permissions. With RBAC, administrative privileges and user access are determined by assigning individuals the appropriate roles.
- **Delegated Administration** - Enables system administrators to delegate some of their administrative privileges to individuals that manage a subset of the organization's users. These individuals are assigned administrative privileges for a limited set of roles that they can assign to the users they manage.
- **Registration Processes** - Enable organizations to provide end-users with a method for requesting various levels of access to the system, based on their eligibility. Registration processes also simplify an administrator's job by providing streamlined flows for account maintenance and role assignment.
- **Self Service Requests and Approvals** - Enable end users to request initial access or additional access to the system.

Oracle User Management is used in both an administrative and a functional capacity. System administrators use Oracle User Management to define the available levels of access control as required, including RBAC, Delegated

Administration, Registration Processes, and Self Service & Approvals. Part of this setup includes defining local administrators primarily by creating administrative roles and assigning them to individuals who serve as an organization's local administrators. Once this is accomplished, local administrators use Oracle User Management to manage a subset of an organization's users.

Oracle Application Object Library Security

Oracle Application Object Library security is primarily comprised of two parts, Function Security and Data Security.

Function Security restricts user access to individual menus of functions, such as forms, HTML pages, or widgets within an application. Function Security by itself restricts access to various functions, but it does not restrict access to the data a user can see or what actions a user can perform on that data.

Data Security restricts the access to the individual data that is shown once a user has selected a menu or menu option. For example, with Data Security you can control the set of users that a particular local security administrator can access within Oracle User Management. In conjunction with Function Security, Data Security provides additional access control on data that a user can see or actions a user can perform on that data.

User and Data Auditing

Oracle Applications allows you to audit users and changes they make on application data.

The Sign-On Audit feature allows you to track your users' activities. You can choose who to audit and what type of user information to track. Sign-On Audit reports give you historical, detailed information on your users' activities within an application. Also, the Monitor Users form allow you to view online, real-time information on user activity.

AuditTrail lets you keep a history of changes to important data: what changed, who changed it, and when. With AuditTrail, you can easily determine how any data row or element obtained its current value. You can track information on most types of fields, including character, number, and date fields.

2 Access Control with Oracle User Management

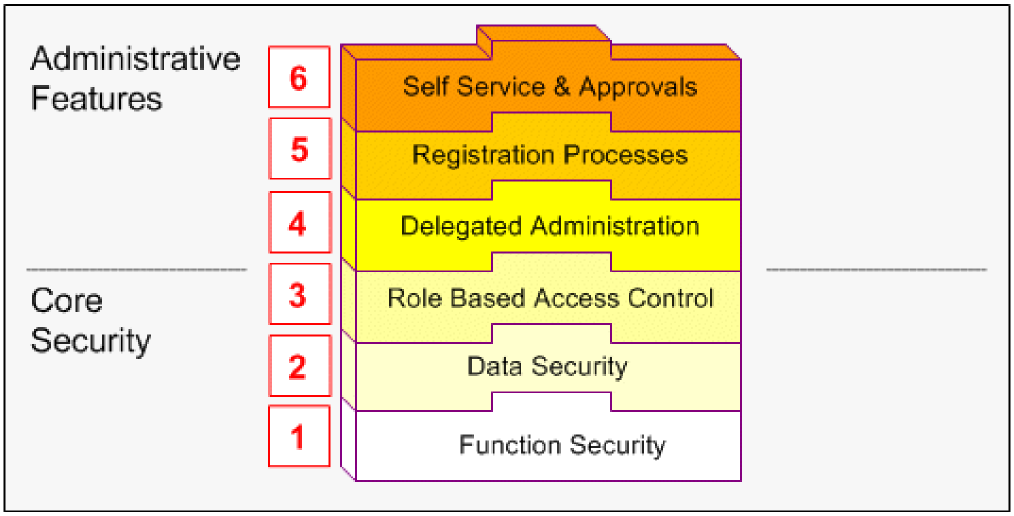
Overview

This chapter introduces the Core Security and Administrative Features of Oracle User Management. Core Security includes Oracle's Function and Data Security models as well as Role Based Access Control. Administrative Features build upon Core Security and include Delegated Administration, Registration Processes, and Self Service and Approvals.

Core Security and Administrative Features are implemented in successive layers and each builds upon the one that precedes it. Organizations can optionally uptake the various layers depending on the degree of automation and scalability that they wish to build upon the existing Function and Data Security models.

In general, Access Control with Oracle User Management begins with basic system administration tasks and then progresses to more distributed, local modes of administration, and ultimately enables users to perform some basic, predefined registration tasks on their own. The following diagram illustrates how the layers build upon each other.

Figure 2-1 Oracle User Management Layers



Oracle User Management provides support for legacy and application-specific security mechanisms through workflow business events. Oracle User Management raises these events once a user's request is approved. Organizations can then intercept these events, determine the appropriate action, and assign any additional privileges that may be required.

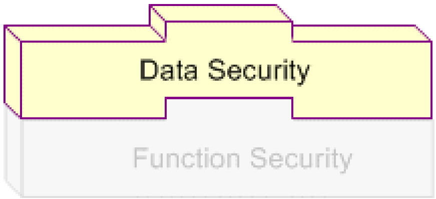
Function Security

Figure 2-2 Function Security Layer

Function Security is the base layer of access control in Oracle Applications. It restricts user access to individual menus and menu options within the system but does not restrict access to the data contained within those menus. For example, an organization could use Function Security to provide its sales representatives with the required menus and menu options for querying customers. It could also control access to specific components of those pages such as a button on a sales forecasting page. For a more comprehensive explanation of function security, please see the Oracle Application Object Library Security chapter, page 4-1.

Data Security

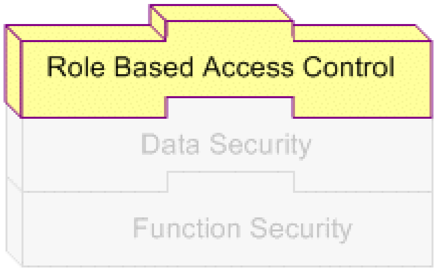
Figure 2-3 Data Security Layer



Data Security is the next layer of access control. Building on Function Security, Data Security provides access control within Oracle Applications on the data a user can access, and the actions a user can perform on that data. Oracle Applications restricts access to individual data that is displayed on the screen once the user has selected a menu or menu option. For example, Data Security restricts the set of users that a local administrator can access within Oracle User Management. Data Security policies can only be defined for applications that have been written to utilize the Data Security Framework. For a more comprehensive explanation of data security, please see the Oracle Application Object Library Security chapter, page 4-1.

Role Based Access Control (RBAC)

Figure 2-4 Role Based Access Control Layer



RBAC is the next layer and builds upon Data Security and Function Security. With RBAC, access control is defined through roles, and user access to Applications is determined by the roles granted to the user. Access control in Oracle Applications closely follows the RBAC ANSI standard (ANSI INCITS 359-2004) originally proposed by the National Institute of Standards & Technology (NIST), which defines a role as "a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role."

A role can be configured to consolidate the responsibilities, permissions, function security and data security polices that users require to perform a specific function. This is accomplished with a one-time setup, in which permissions, responsibilities, and other roles are assigned to the role. Users are not required to be assigned the lower-level permissions directly, since permissions are implicitly inherited on the basis of the roles assigned to the user. This simplifies mass updates of user permissions, since an organization need only change the permissions or role inheritance hierarchy defined for a given role, and the users assigned that role will inherit the new set of permissions automatically.

Organizations can define roles that closely mirror their business situation. For example, an organization can create an "Employee" role and then assign that role to all of its employees. It can also create an "External" role and assign that role to customers and suppliers. Further examples may include specific roles such as "Support Agent", "Sales Rep", "Sales Managers". In these examples, each role contains a specific level of access privileges that restricts its assignees to the scope of their job functions. Some members of the organization will probably be assigned more than one role. A sales representative would be assigned the Employee and Sales Representative roles, and a Sales Manager would be assigned the Employee, Sales Representative, and Sales Manager roles. Roles and role assignments are stored in the workflow directory, which is interpreted by the security system at runtime.

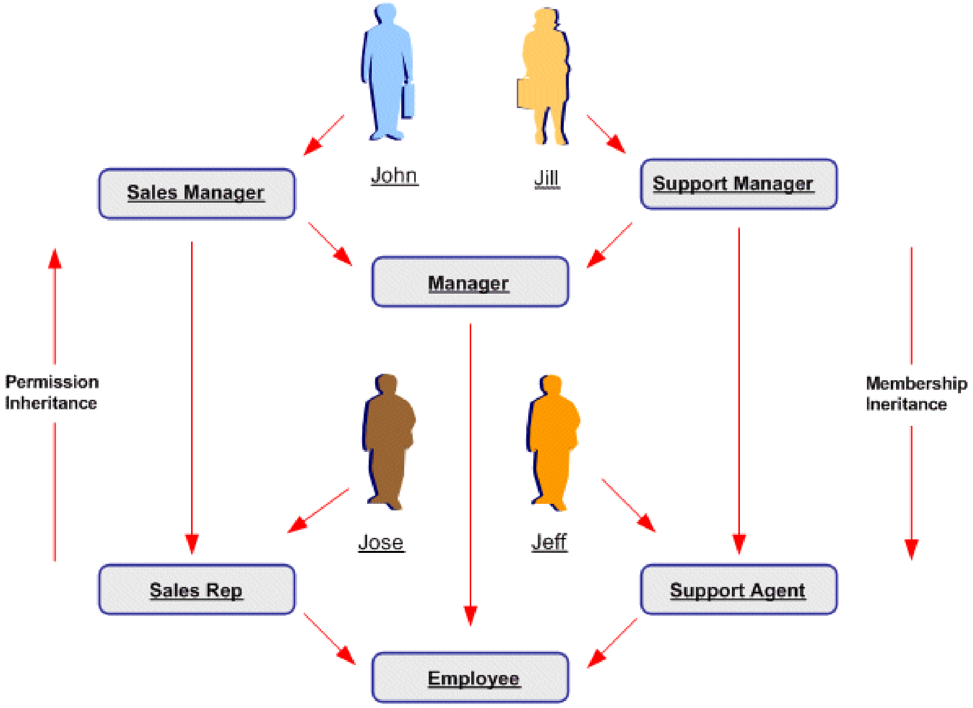
Role Categories

As part of the Oracle Applications RBAC model, Oracle User Management introduces Role Categories. Administrators can create role categories to bundle roles and responsibilities to make the process of searching for roles and responsibilities easier. For example, all sales and marketing related roles could be included in the Sales & Marketing category.

Role Inheritance Hierarchies

Roles can be included in role inheritance hierarchies that can contain multiple sub roles and superior roles. With role inheritance hierarchies a superior role inherits all of the properties of its sub role and any of its sub roles. The following example demonstrates how role inheritance hierarchies can greatly simplify user access control and administration.

Figure 2-5 Role Inheritance Hierarchy

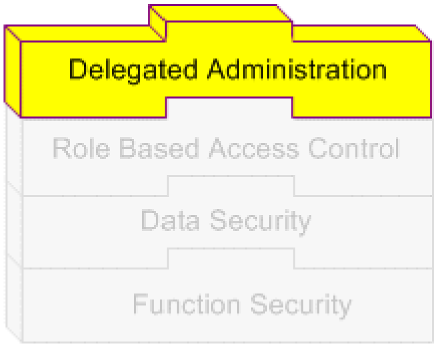


In the above figure, the arrows on each side of the diagram indicate membership inheritance and permission inheritance. Text in the rounded boxes indicates roles. An arrow pointing from an individual to a role indicates that this individual is assigned the role. An arrow pointing from one role to another indicates that the role from which the arrow points is the superior role, and the role to which it points is the subordinate role. Permissions associated with a role are inherited by all of its superior roles and the individuals to which any of these roles are assigned.

In this example, some roles such as "Employee" or "Manager" are assigned general permissions for a given function. For example, the Employee role may provide access to menus generally available to all employees, while the Manager role provides access to menus that should only be viewed by managers. Because the Employee role is a sub-role of the Manager role, anyone assigned the Manager role automatically obtains the permissions associated with the Employee role. Other roles in this example pertain to more specific job functions, such as Sales Manager and Sales Rep, or Support Manager and Support Agent. These roles may provide access to job-specific menus and data such as the Sales Forecasting menu, or the Support application.

Delegated Administration

Figure 2-6 Delegated Administration Layer



Delegated Administration is a privilege model that builds on the RBAC system to provide organizations with the ability to assign the required access rights for managing roles and user accounts. With delegated administration, instead of relying on a central administrator to manage all its users, an organization can create local administrators and grant them sufficient privileges to manage a specific subset of the organization's users and roles. This provides organizations with a tighter, more granular level of security, and the ability to easily scale their administrative capabilities. For example, organizations could internally designate administrators at division or even department levels, and then delegate administration of external users to people within those (external) organizations. Delegation policies are defined as data security policies. The set of data policies that are defined as part of delegated administration are known as Administration Privileges.

Administration Privileges

Administration Privileges determine the users, roles and organization information that delegated administrators (local administrators) can manage. Each privilege is granted separately, yet the three work in conjunction to provide the complete set of abilities for the delegated administrator.

User Administration Privileges. A local administrator must be granted User Administration Privileges to determine the users and people the local administrator can manage. Local administrators can be granted different privileges for different subsets of users. For example, a local administrator can be granted privileges only to query one set of users, and granted full privileges (including update and reset password) for another set. Local administrators cannot query users for which they do not have administration privileges.

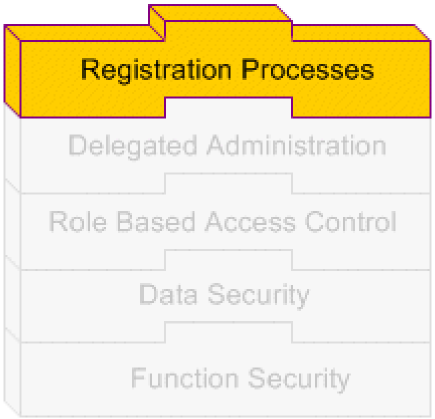
> **Note:** Please note that Oracle Applications continues to support the traditional "System Administrator" level of administration privileges, where a designated group of people manages all users and access privileges. Oracle User Management ships a predefined Security Administrator role, which gives the administrator the privileges to manage all users including system accounts and all roles in the system.

Role Administration Privileges. Role Administration Privileges define the roles that local administrators can directly assign to and revoke from the set of users they manage.

Organization Administration Privileges. Organization Administration Privileges define the external organizations a local administrator can view in Oracle User Management. This privilege enables an administrator to search for people based on their organization, if the local administrator has additionally been granted access to view the people in that organization (User Administration Privileges). Depending on the user administration privileges, an administrator may have the ability to register new people for that organization.

Registration Processes

Figure 2-7 Registration Processes Layer



Registration processes are predefined registration components that enable end users to perform some of their own registration tasks, such as requesting new accounts or additional access to the system. They also provide administrators with a faster and more efficient method of creating new user accounts, as well as assigning roles. Registration processes accomplish this by encapsulating core components of registration, including:

- The role(s) assigned after the user successfully completes the process.
- An optional registration user interface for collecting account or additional information.
- A workflow for approval, confirmation, rejection, and identity verification notifications.
- The Approval Management Transaction Type. A transaction type represents a set of approval routing rules that are interpreted at runtime.
- The set of users that are eligible to sign up for additional access (only applicable for Request for Additional Access registration processes).
- Whether identity verification is required. Identity verification confirms the identity of a requester before the registration request is processed, by sending an email

notification to the requester's email address. If the recipient does not reply within a specified time, the request will be automatically rejected.

- The set of local administrators that should be able to register people and/or create users through the Account Creation by Administrators registration process.

When a user completes registration using a registration process, the system captures the required information from the user, and subsequently assigns that person a new user account, role, or both. Oracle User Management supports three types of registration processes: Self-service Account Requests, Requests for Additional Access, and Account Creation by Administrators.

Self-Service Account Requests

Commonly referred to as Self Service Registration, self-service account requests provide a method for persons to request a new user account. Consider a case where customers may need to register before they can purchase an item from an online store. Once the registration process has been completed, the customer obtains both a user account and the necessary role(s) for accessing some portion of the web site in which they registered.

This release of Oracle User Management provides sample Self Service registration UIs for internal employees, and for new, external individuals. Organizations can copy these sample Self Service registration and extend them based on their own requirements. In addition, organizations that wish to support other types of users, or capture additional information specific to their applications, are able to extend or create their own registration UIs and business logic.

Oracle User Management provides support for displaying different registration links on the login page based on the application tier login page that provides access. The registration link can contain additional parameters that are not known at design time, such as the country code. These additional parameters can be used later during the registration process. Using country code as an example, a registration process could route the approval requests to the most appropriate approver. Therefore, all those who request an account from Norway could be routed to a Norwegian account approver.

> **Note:** "Accounts" and "User Accounts" refer to an individual's login account, stored in the FND_USER table.

Requests for Additional Access

Users can request additional access through the Oracle User Management Access Request Tool (ART,) available in the Global Preferences menu. Requests for Additional Access uses the same Oracle User Management infrastructure and processing logic as Self Service Account Requests.

Additional Access and Self Service Eligibility

Eligibility defines the Roles for which a user can sign up using the Access Request Tool. It determines the groups of users defined in the workflow directory that are entitled to register for a given role. A registration process of type "Additional Access" can be made available to predefined sets of users across all roles or groups. Eligibility is defined as a data security policy, and interrogated at runtime by the Access Request Tool, but is not considered when administrators assign roles.

Because roles are stored in the workflow directory, they can be used both to grant access to applications and to define eligibility. This enables organizations to define an incremental registration process in which new users can sign up for roles if they are first approved for the ones that precede them. For example, once a new user is approved for the A Role, the user can then sign up for the B Role. If, however, the user is not first approved for the A Role, then the user cannot sign up for the B Role.

Oracle User Management can define eligibility policies for any groups and roles stored in the workflow directory.

Delegated Administration and Registration Processes

When an administrator assigns a role to a user, the administrator essentially fulfills a registration request on behalf of the user. When the administrator assigns a role to the user, Oracle User Management invokes the corresponding “additional access” registration process (if defined) and interprets the registration processes metadata. If a registration UI is defined, Oracle User Management launches it and the administrator completes the registration process. In this scenario, Oracle User Management invokes the same processing logic as it does when a user requests additional access. Notification workflows are only invoked when a registration process is defined for the role that is being assigned to the user.

Directly assigning a role to a user bypasses any pre-defined approval routing rules, as defined in Oracle Approval Management. Self-service eligibility, as defined for a registration process, is not considered when administrators assign roles. Administrators can view all roles that are assigned to a user, but cannot assign or revoke roles for which they do not have administrative privileges. An administrator assigning a role to a user is essentially fulfilling a registration request on behalf of the user.

Account Creation By Administrators

Administrators benefit from registration processes having been designed to streamline the process of creating and maintaining user access. Registration processes of this type are geared toward administrators, especially delegated administrators, to ensure consistent application of the organization’s user security policies. Each account creation registration process can be made available to selected administrators.

Registration Process Infrastructure

This section describes components of the common infrastructure that handles all registration requests submitted through Oracle User Management.

User Name Policies

Oracle User Management enables organizations to define their own user name policies for new users. These can include such formats as email address, "firstname.lastname" (or an abbreviated version), employee number, social security number, or some other meaningful information. When the account request is submitted, Oracle User Management reserves the specified user name for the duration of the approval process.

Oracle User Management ships with a default user name policy that identifies users by their *email address*. This is implemented as a configurable infrastructure that organizations can easily customize to suit their specific needs.

Email Verification

Oracle User Management provides a mechanism for verifying the identity of the requester before the registration request is processed. Identity verification is based on the email address provided by the requester. Oracle User Management sends the requester an email notification when the requester has completes the registration flow. If the user does not reply to the email notification within a specified time, the request is automatically rejected. Email verification is only applicable to Self-Service account requests, and is enabled or disabled for each registration process.

Note: Oracle recommends that when building self-service registration UIs with identity verification enabled, an organization should indicate in the UIs and confirmation messages that a response is required to process the user's request. This mechanism is also used by the Forgot Password feature.

Temporary Storage of Registration Data

Oracle User Management provides a mechanism to store registration data in a pending state until a request is approved. This data is available to the workflow notifications used for sending approvals, to Approval Management routing rules, and to the business logic that writes the information in the final destination tables. Oracle User Management accomplishes this by using event objects that are part of the Workflow Business Events infrastructure.

Registration Engine

The Oracle User Management registration engine uses a workflow to define the business logic that drives the registration process once a request has been submitted. The name of the workflow is User Management Registration Workflow (UMXREGWF).

This process:

- Raises business events
- Provides temporary storage of registration data
- Provides identity verification
- Includes the integration point with Oracle Approval Management
- Activates user accounts
- Reserves and releases user names
- Assigns roles
- Maintains registration status in the Oracle User Management schema
- Launches notification workflows

Organizations can customize the components of the registration process (such as notifications, approval routing rules, and user name policies) without having to review and understand all Oracle User Management code.

Routing Approval Requests

Approvers can be configured based on rules that are specific to each type of request. Organizations can define these rules according to their requirements, and can specify types of requests that do not require approval. Oracle User Management is integrated with Oracle Approval Management, an application that provides a flexible and powerful rules engine that can be configured through declarative means to route approval requests. Oracle User Management also provides APIs that enable approval rules to be based on any information captured during the registration process, including any parameters passed from the "Register Here" link on the Login page, which may not have been known at design time.

Workflow Business Events

Oracle User Management raises the following Workflow business events:

Table 2-1 Oracle User Management Workflow Business Events

Event	Description
oracle.apps.fnd.umx.rolerequested	An event that is raised when a role is requested.
oracle.apps.fnd.umx.accountrequested	An event that is raised when an account is requested.
oracle.apps.fnd.umx.requestapproved	An event that is raised when an account or role is approved.
oracle.apps.fnd.umx.requestrejected	An event that is raised when an account or role is rejected.
<custom event>	A custom business event is raised for the owner of the registration process to write the registration. The custom event is raised multiple times. For further details, refer to the UMX Developer's Guide on Oracle*MetaLink*.

> **Note:** Oracle recommends using the UMX events mentioned above only for centralized requirements such as auditing. For any registration-specific processing, use the custom event defined for the registration process.

Depending on the context, the event parameters listed in the following table are set automatically by the Oracle User Management registration engine when business events are raised. Any additional information captured in the registration UI, approval notifications, or programmatically through business logic is also available as event parameters.

Table 2-2 Oracle User Management Workflow Business Event Parameters

Name	Description
REG_SERVICE_CODE	Represents the primary key of the registration process
REG_SERVICE_TYPE	The type of registration process
REQUESTED_BY_USER_ID	Identifies the user submitting the request
REQUESTED_FOR_USER_ID	Identifies the user for whom the request is submitted
REQUESTED_USERNAME	The requested user name
WF_ROLE_NAME*	Represents the primary key value of the requested role or the default role for any account requests
AME_TRANSACTION_TYPE_ID	Represents part of the primary key for the transaction type in Oracle Approval Management
AME_APPLICATION_ID	Represents part of the primary key for the transaction type in Oracle Approval Management

* WF_ROLE_NAME is not required for Self Service Account Creation or Account Creation for Administrators registration processes. In such cases, a null value is passed. Any additional information captured in the registration UI, from approvers, in approval notifications, or set by business logic is also available as parameters when an Oracle User Management business event is raised.

Sample Program

```
/**************************************************************
This is a sample subscription to any of the above events.

Function custom_logic (p_subscription_guid in raw,
 p_event in out NOCOPY WF_EVENT_T)
Return varchar2 is
 l_first_name varchar2(30);
Begin
 l_first_name :=  p_event.getvalueforparameter ('FIRST_NAME');
// Manipulate the data
End custom_logic;
*************************************************************/
```

Registration Status

Users can check registration status of requests through the Access Request Tool (ART) and administrators can do so using the Administration screens. For any pending requests, the Show Info icon shows the current approver and confirmation number. The confirmation number represents the number (ITEM_KEY) of the Oracle User Management Registration Workflow (UMXREGWF) workflow process handling the request.

Notification Workflows

Notification workflows enable an organization to define its own email notifications that are specific to each Role or Registration Process. Notifications include:

Table 2-3 Oracle User Management Notification Types

Notification	Recipient
Approver notifications	Each approver.
Approval confirmation notifications	Individual for whom the request was filed.
Rejection notifications	Individual for whom the request was filed.
Identity verification notifications	Individual for whom the request was filed.

For each request that requires approval as determined by the Oracle Approval Management Engine, Oracle User Management invokes the notification workflow to request approval. Notification workflows can be written to allow approvers to review the information submitted in the registration process, make changes, and provide additional information if required.

Any changes or additional information provided can be passed back to the Oracle User Management registration engine for further processing. For example, if Oracle User Management is used to provide self service registration capability for iSP (Internet Supplier Portal), then approvers can provide additional information about site and contact restrictions for the requester. Information entered by previous approvers, including comments, are available to subsequent approvers.

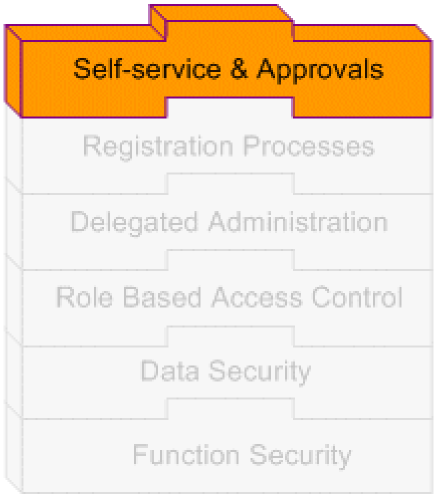
Oracle User Management provides the following sample notification workflows that organizations can use directly or can copy and modify based on their requirements:

Table 2-4 Sample Notification Workflows

Name	Item Type	Description
Oracle User Management Additional Access Request notification workflow	UMXNTWF1	Sends notifications pertaining to all requests for additional access.
Oracle User Management Notification Workflow (Account Request)	UMXNTWF2	Sends notifications pertaining to all account requests.

Self Service and Approvals

Figure 2-8 Self-Service & Approvals Layer



Once registration processes have been configured as required, individuals can subsequently perform self-service registration tasks, such as obtaining new user accounts or requesting additional access to the system. In addition, organizations can use the Oracle Approvals Management engine to create customized approval routing for these requests. For example, an organization may enable users to request a particularly sensitive role: however, before the user is granted the role, the organization can require that two senior members of staff, such as a manager and a vice president, must approve the request.

Oracle User Management also provides self-service features for resetting forgotten passwords, and ships with the following sample self service registration processes:

- Employee Self Service Registration
- Customer Self Service Registration (external individuals)

Organizations can either use these registration processes in their existing form, or as references for developing their own registration processes.

3 Oracle User Management Setup and Administration

Setup Tasks

This section discusses the setup tasks for Oracle User Management. The implementor or system administrator sets up access control and security policies in Oracle Applications by defining roles, role inheritance hierarchies, role categories, and registration processes. These components specify the different levels of access to various application menus and data that are available to administrators.

Defining Role Categories

As part of the Oracle Applications RBAC model, Oracle User Management introduces Role Categories. Administrators can create role categories to bundle roles and responsibilities to make the process of searching for roles and responsibilities easier. In the Oracle User Management Overview section, see Role Based Access Control (RBAC), page 2-3.

Steps

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Role Categories** subtab.
2. Go to the editable table, click the **Update** button and then click the **Create Lookup Code** button.
3. Enter the required information in the Create Lookup Code fields and click the **Apply** button.

Creating and Updating Roles

In Oracle Applications, a role represents a job function that confers the privileges required to perform that job. Roles can be defined to determine what applications (responsibilities) as well as what data and functions within those applications users can access. In the Oracle User Management Overview section, see Role Based Access Control (RBAC), page 2-3.

Steps

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Roles & Role Inheritance** subtab.

2. Click the **Create Role** button.

3. Enter the required information to configure your role and optionally continue to configure it by accessing the following:

   - **Permissions**, page 3-2. Use this tab to assign permissions to your role.

     Delegated Administration

     Information in this section only applies to delegated administration roles in the context of the Oracle User Management application.

     - **User Administration**, page 3-4. Enables you to determine the set of users that can be managed by administrators to whom your role is assigned. The administrator can assign or revoke user accounts and roles for the users you specify here.

     - **Organization Administration**, page 3-6. Enables you to determine the external organizations that can be viewed in Oracle User Management by administrators to whom your role is assigned.

     - **Role Administration**, page 3-5. Enables you to determine which roles the administrator can assign to or revoke from the set of users specified in the User Administration section.

4. Click **Save** or **Apply** to save your changes.

5. Optionally update the role by performing the following:

   1. Locate the role you want to modify by using the Search fields or by expanding the appropriate nodes in the Role Inheritance Hierarchy menu.

   2. Click the **Update** icon and modify the role as required.

Guidelines

The **Save** button saves your changes and continues to display them in the current page. The **Apply** button saves your changes and returns to the previous page. You can optionally organize your roles using role categories during the process of creating and updating roles, otherwise they will be stored under the "Miscellaneous" role category by default. For more information, see role categories, page 3-1. You can also define any required sub roles or superior roles through role inheritance hierarchies, page 3-7.

Assigning Permissions to Roles

You can assign permissions to a role by creating a grant that specifies the navigation menu, permission sets, and/or the data security policies that are available at runtime to the role's assignees. Menus and permission sets in turn include individual functions and permissions. In the Oracle User Management Overview section, see Role Based Access Control (RBAC), page 2-3.

Steps

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Roles & Role Inheritance** subtab.

2. In the Role Inheritance Hierarchy, access the role to which you want to assign a permission and click the **Update** icon.

3. Click the **Permissions** subtab and the click **Create Grant** button.

4. Define the grant by entering the required information and clicking **Next**:

    1. Enter the required information to identify the grant such as Name and Effective From date.

    2. **Security Context**. This optional restricts the availability of the permissions being assigned. If you do not define the security context, then permissions are available to users in all contexts. Security contexts are also referred to as *Activation Contexts*.

        1. **Operating Unit**. In many cases an organization consists of several different operating units. You can limit your grant to only be active in the context of an individual operating unit.

        2. **Responsibility**. Responsibilities determine the applications that can be accessed by users. You can optionally limit your grant to only be available in the context of an individual responsibility or with all responsibilities.

    3. **Data Security**. You must select a business object when you create Data Security policies. For more information, see the Oracle Application Object Library Security chapter, page 4-1.

5. If you have defined a specific object in the preceding step, then choose the object data context for the object, also referred to as the *data scope*. Specifying the object data context provides an additional level of access granularity for the object. Choose one of the following from the Data Context menu:

    - **All Rows**. This option provides access to all rows (instances) for the database object. For example, if the database object is a book, then creating a data security policy for all rows of the object, book provides access to all books catalogued in the database.

    - **Instance**. This option provides access to an instance (single row in the database) of the object. A specific instance generally corresponds to a single row in the database and is typically identified by the primary key value for the object. For example, a data security policy for the book object contains a unique ISBN number, returns only one book from the database.

    - **Instance Set**. This option provides access to a related set of instances of the object. This set is specified as a predicate on the attributes of the object. The predicate is expressed as a SQL WHERE cluase and can optionally be implemented as a VPD policy. For example, a data security policy could include an instance set for all books published in the year 2005.

6. Select the required permission set or navigation menu containing the functions (permissions) that you wish to assign to the role by choosing an option from the LOV.

7. Review your grant information and click **Finish**.

Defining Delegated Administration Privileges for Roles

Delegated Administration Privileges determine the users, roles and organization information that delegated administrators (local administrators) can manage. Each privilege is granted separately, yet the three work in conjunction to provide the complete set of abilities for the delegated administrator. In the Oracle User Management Overview section, see Delegated Administration, page 2-5.

Defining User Administration Privileges for Roles

A local administrator must be granted User Administration Privileges to determine the users and people the local administrator can manage. Local administrators can be granted different privileges for different subsets of users. For example, a local administrator can be granted privileges only to query one set of users, and granted full privileges (including update and reset password) for another set. Local administrators cannot query users for which they do not have administration privileges.

Oracle User Management ships with the following seeded permissions for defining user administration privileges for roles:

Table 3-1 Seeded User Administration Permissions

Function Code	Display name	Description
UMX_OBJ_ACTIVATE_ACCT	Create, Inactivate, Reactivate User Account, Update Username	Permission for creating, inactivating, and reactivating user accounts, and updating username. Must be granted with a data security policy on the User Management Person.
UMX_OBJ_EDIT_PERSON	Edit Person Details	Permission for editing person details. Must be granted with a data security policy on the User Management Person (UMX_PERSON_OBJECT) business object.
UMX_OBJ_PASSWD_MGMT	Reset Password	Permission to reset passwords. Must be granted with a data security policy on the User Management Person (UMX_PERSON_OBJECT) business object.
UMX_OBJ_VIEW_PERSON	Query Person Details	Permission to query person details Must be granted with a data security policy on the User Management Person (UMX_PERSON_OBJECT) business object. **Note:** This is the minimum permission required by any security administrator that wishes to manage people and users in Oracle User Management.
UMX_SYSTEM_ACCT_ADMINSTRATION	Maintain System Accounts (users not linked to a person)	Create, Inactivate, Reactivate, Reset Password for all System Accounts (defined as user accounts not associated with a person). **Note:** Only grant to System Administrators.

Steps

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Roles & Role Inheritance** subtab.
2. In the role hierarchy, access the role to which you want to assign user administration privileges and click the **Update** icon.
3. Click the **User Administration** subtab and then click the **Add More Rows** button.
4. In the Users field, select the set of users that can be managed by Administrators to whom the role is assigned. The drop down list contains various data security policies that pertain to the User Management Person Object (UMX_PERSON_OBJECT). Oracle User Management ships with sample data security policies for users. Organizations can use these policies or create their own. For more information, see Defining Data Security Policies, page 3-7.
5. In the Permissions field, select the permissions that you wish to associate with the delegated administration role. Permissions determine the actions an administrator can perform when managing the set of users defined in the previous step. The Permissions drop down list includes permission sets that contain permissions associated with the User Management Person object. Different combinations of the existing permissions can be grouped into new permission sets, enabling organizations to add permission sets based on their business needs and the level of granularity they prefer for administering users. For more information, see Permission Sets, page 4-46.
6. Click **Save** or **Apply** to save your changes.

Guidelines

Delegated administration can provide different permissions on different subsets of users. Once you define users and permissions for a role, you can optionally view the permissions that belong to the permission set by clicking the **Show** node. You can also remove the user administration privileges for a set of users by clicking the **Remove** icon.

Defining Role Administration Privileges for Roles

Role Administration Privileges define the roles that local administrators can directly assign to and revoke from the set of users they manage.

Oracle User Management ships with the following seeded permission for defining role administration privileges for roles:

Table 3-2 Seeded Role Administration Permission

Function Code	Display Name	Description
UMX_OBJ_ADMIN_ROLE	Assign/Revoke Role	Permission for assigning/ revoking roles in the User Management application. Must be granted with a data security policy on the User Management Role (UMX_ACCESS_ROLE) business object.

Steps

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Roles & Role Inheritance** subtab.
2. In the navigation menu access the role for which you want to define role administration and click the **Update** icon.
3. Click the **Role Administration** link and use the Available Roles fields to search for the role(s) that you want to associate with this role and which administrators can manage once they are assigned this role.
4. Select the desired role(s), move them to the Selected Roles column and click **Save** or **Apply**.

Guidelines

The **Save** button saves your changes and continues to display them in the current page. The **Apply** button saves your changes and returns to the previous page.

Defining Organization Administration Privileges for Roles

Organization Administration Privileges define the external organizations a local administrator can view in Oracle User Management. This privilege enables an administrator to search for people based on their organization, assuming the local administrator has also been granted access to view the people in that organization (User Administration Privileges). Depending on what administration account registration process has been granted, the administrator may have the ability to register new people for that organization.

Oracle User Management ships with the following seeded permission for defining organization administration privileges for roles:

Table 3-3 Seeded Organization Administration Permission

Function Code	Display Name	Description
UMX_OBJ_VIEW_RLTNSHPS	Query/Register Organization Relationship	Permission to query/register organization relationship. Must be granted with a data security policy on the User Management Organization (UMX_ORGANIZATION_ OBJECT) business object.

Steps

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Roles & Role Inheritance** subtab.
2. In the navigation menu access the role to which you want to define organization administration and click the **Update** icon.
3. Click the **Organization Administration** link and then click the **Assign Organization Privileges** button. The drop down list contains various data security policies that pertain to the User Management Person Object (UMX_PERSON_OBJECT). Oracle User Management ships with sample data security policies for organization administration privileges. Organizations can use these policies to create their own.

4. Search for and select the appropriate organization privileges.
5. Click **Save** or **Apply** to save your changes.

Guidelines

The **Save** button saves your changes and continues to display them in the current page. The **Apply** button saves your changes and returns to the previous page.

Defining Data Security Policies

With Oracle Applications, organizations can use Data Security to manage permission assignments that control access to objects. Data Security policies can only be defined for applications that have been written to utilize the Data Security Framework. For more information, see Data Security, page 4-15. Access to the specific object must be formed with a specified Data Security Policy (also referred to as the Data Scope or Access Policy). The Data Security Policy restricts operations so that they only can be performed on a subset of instances of the corresponding database object. For more information, see Object Instance Sets, page 4-36.

Steps

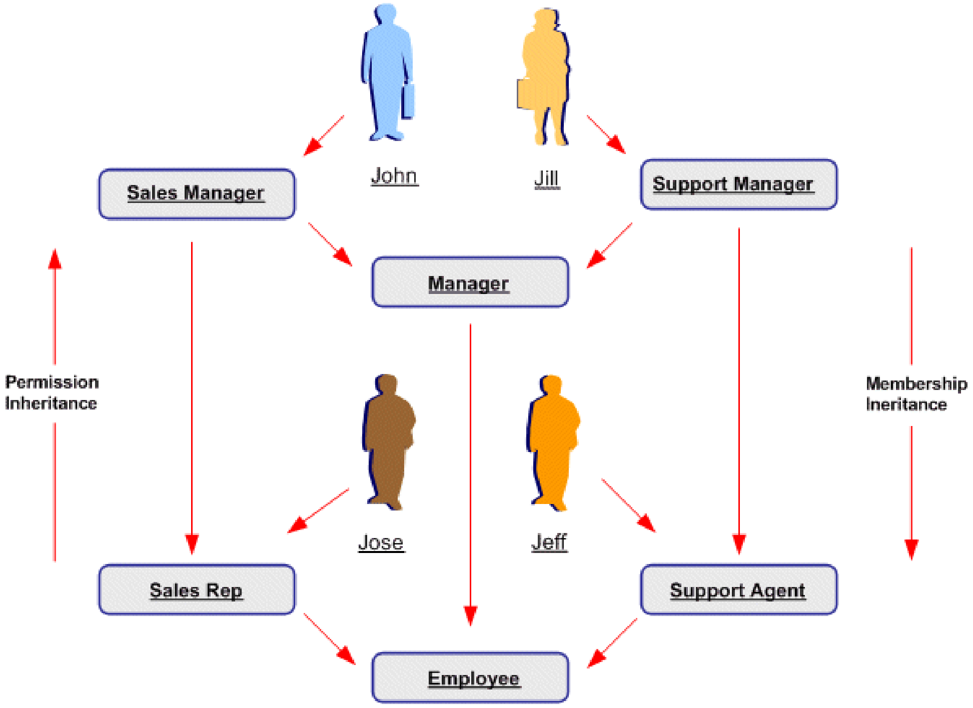
1. Log on as a user with the Functional Developer responsibility, click the **Functional Developer** responsibility in the navigator, navigate to the **Security** tab and then click the **Objects** subtab.
2. Search for and access the object for which you want to create data security policies. For example, to locate the User Management Person business object (UMX_PERSON_OBJECT), enter "UMX%" in the Code field, click the **Go** button, and then click User Management Person object (UMX_PERSON_OBJECT) in the search results list. For any object for which you are creating a policy, ensure that the SQL statement returns the primary key value for that object. In this example, this is a list of person party IDs.
3. Click the Object Instance Sets subtab. Click the **Create Instance Set** button to create a new object instance set or click the **Update** icon to modify an existing one.
4. Enter the required information and then click the **Apply** button.

   Caution: For performance reasons, ensure that SQL predicates are tuned properly. For security reasons, ensure that they are tested and that they return the correct result. Oracle is not responsible for the performance or correctness of data security policies defined by organizations.

Defining Role Inheritance Hierarchies

With role inheritance hierarchies, a role can contain sub roles. When a user is assigned a role, the user inherits the privileges defined for that role and for all of its sub roles. For example, the Sales Manager role can contain the Manager and Sales Rep roles, both of which in turn contain the Employee role. Any individual who is granted the Sales Manager role automatically inherits the Manager, Sales Rep and Employee roles.

Figure 3-1 Role Inheritance Hierarchies



With Role Inheritance Hierarchies, roles inherit the permissions assigned to their sub roles.

Steps

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Roles & Role Inheritance** subtab.

2. Locate the role for which you want to create a role inheritance hierarchy by using the Search fields or by expanding the appropriate nodes in the Role Inheritance Hierarchy menu. If you are building a role inheritance hierarchy that contains several roles, start with highest level role to which you want to add inherited sub roles.

3. Click the **Add Node** icon next to this role.

4. In the resulting menu, search for the role either by using the Search fields or by locating it in the Role Inheritance Hierarchy menu.

5. Select the role and then click the **Select** button or the **Quick Select** icon.

6. Repeat this process until you have added all of the required sub roles to their corresponding super roles. You can optionally verify the results by expanding the nodes for all super roles within your role inheritance hierarchy. You can also remove any sub roles by clicking the **Remove Node** icon.

Deployment Options

Organizations can use different deployment options for role inheritance hierarchies depending on their requirements.

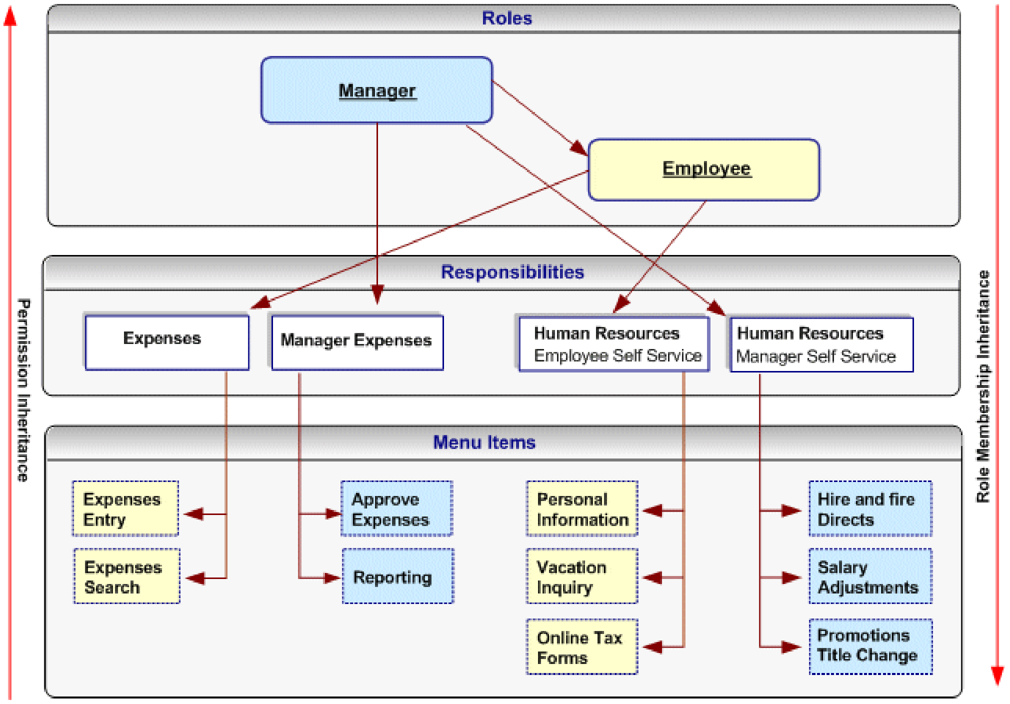
Assigning Existing Responsibilities to Roles Using Role Inheritance

Organizations that have already defined their responsibilities can utilize RBAC by creating roles and assigning their existing responsibilities to those roles. For example an organization could create an Employee role and a Manager role to which it adds the Expenses and Human Resources responsibilities that it wishes to make available to employees and managers as required. Then, instead of manually assigning or revoking each of these responsibilities to or from its employees, the organization can simply assign or revoke the Employee and Manager roles as required. Since the Manager role inherits the employee role, managers that are assigned the Manager role also inherit all of the responsibilities and privileges associated with the Employee role.

In the following example, a Human Resource Manager inherits the Human Resources Manager Self Service responsibility through the Manager role as well as the Human Resources Employee Self Service responsibility, which the Manager role inherits from the Employee role.

> **Note:** In this section, references to the Expenses and Human Resources responsibilities are used as examples only. Some applications may require organizations to create multiple responsibilities to operate with their existing security models. For more information, please consult the application-specific documentation.

Figure 3-2 Assigning Existing Responsibilities to Roles Using Role Inheritance



Steps

1. Create roles representing the required job functions such as Manager and Employee.
2. Define a role inheritance hierarchy. For more information, see Defining Role Inheritance Hierarchies, page 3-7.
3. Ensure the responsibilities are inherited by their corresponding roles.

4. Assign the roles to users as required.

Fully Utilizing RBAC and Role Inheritance to Determine Access to an Application

In previous releases of Oracle Applications, access to individual functions within an application could only be defined through responsibilities, menu hierarchies, and menu exclusions. Responsibilities had the dual role of defining application navigation menus and granting permissions to the application. New responsibilities with one of the following had to be defined for each set of users with different job functions that required access to a set of pages within an application:

- A completely new menu hierarchy for each responsibility, or
- A common menu covering the superset of all functions within the application, and menu exclusion rules defined for each responsibility.

The Human Resources application, for example, typically required a minimum of two responsibilities, one for employees and one for managers.

Separating Navigation Menus and Access Control

Oracle User Management provides new alternatives for defining access to an application with RBAC and Role Inheritance, allowing organizations to separate navigation menus from access control. Responsibilities can now be defined to represent an application itself and as a result, only one responsibility may be required for each application. A menu can be tailored for each application with specific consideration to usability and end user navigation experience. Access to parts of the application (responsibility) and its corresponding menu hierarchy are instead controlled by different roles, each representing a specific job function or set of people.

Benefits

Using this mechanism for determining access control provides several benefits.

- Administration and changes are accomplished with minimal effort:
  - A new page only has to be added to a single menu.
  - The permission to access a new page, only has to be granted once to the lowest level (sub role) in the role inheritance hierarchy.
  - An entirely new application (responsibility) can automatically be assigned to a set of people by simply defining it as the sub role of an existing role.
  - Permissions to access the various pages/functions within a new application must only be assigned at the lowest level in the role inheritance hierarchy. The permissions are then automatically inherited by all superior roles in the hierarchy.
  - Revoking access to a page, or an entire application, can be accomplished as easily as adding access.
- Improved end user experience. End users will see a short list of applications to which they have access in the applications navigator. Access to the various functions within each application is determined by the roles assigned to the end user.

Steps

> **Note:** In this section, references to the Expenses and Human Resources responsibilities are used as examples only. Some applications may require organizations to create multiple responsibilities to operate with their existing security models. For more information, please consult the application-specific documentation.

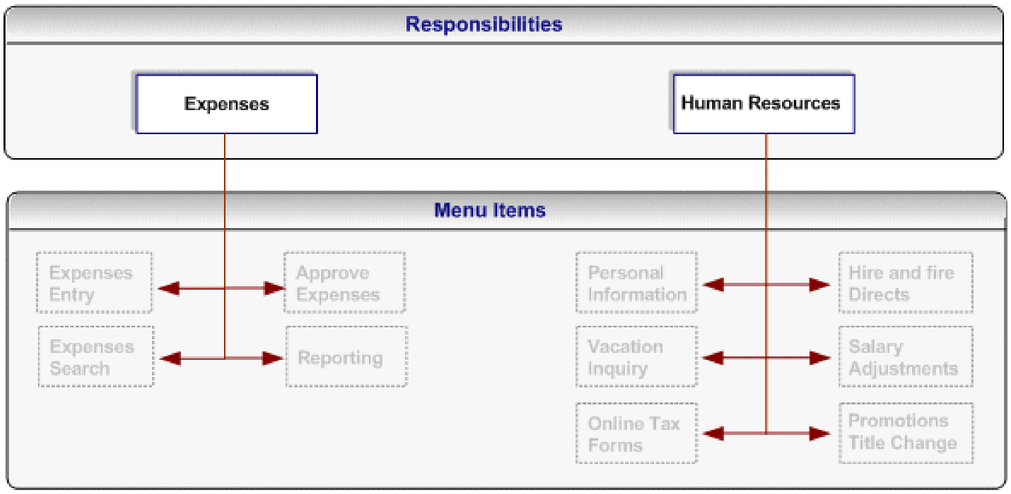
1. Define a new responsibility that will be used to represent a specific application such as Expenses or Human Resources. For more information, see Defining a Responsibility, page 4-3.

2. Design a complete menu that includes all the menu functions within an application as well as any required sub menus, and attach this menu to the new responsibility. For example, both the Expenses and Human Resources responsibilities would include all employee and manager menus. For more information, see Defining a New Menu Structure, page 4-29.

3. Following the "principle of least privilege", all the menu options within the application (each menu item corresponds to a function/permission) should be disabled by default. To accomplish this, remove the selection from the "grant" checkbox for each menu item:

   The following figure illustrates application responsibilities (in this case, Expenses and Human Resources) with all their menus disabled:

   Figure 3-3 Responsibilities Representing an Entire Application with Disabled Menus

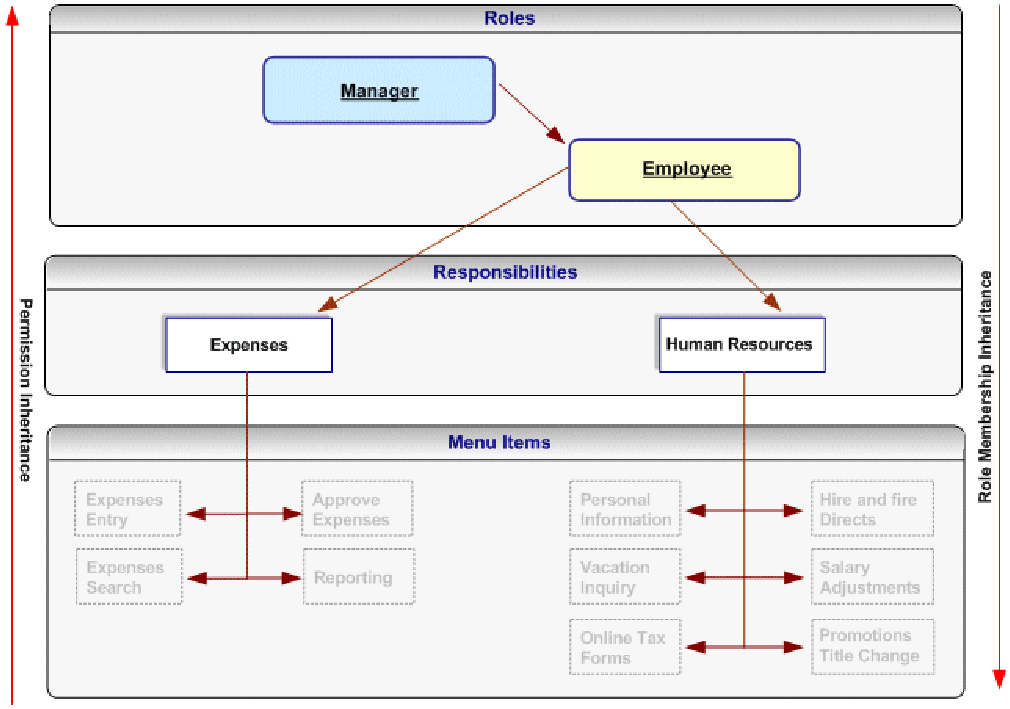


> **Note:** A user cannot access any of the menu items (functions) within the application if you assign the responsibility to the user at this stage.

4. Create roles representing the people with various job functions that require access to the application, for example, a Manager role and an Employee role. For more information, see Creating and Updating Roles, page 3-1

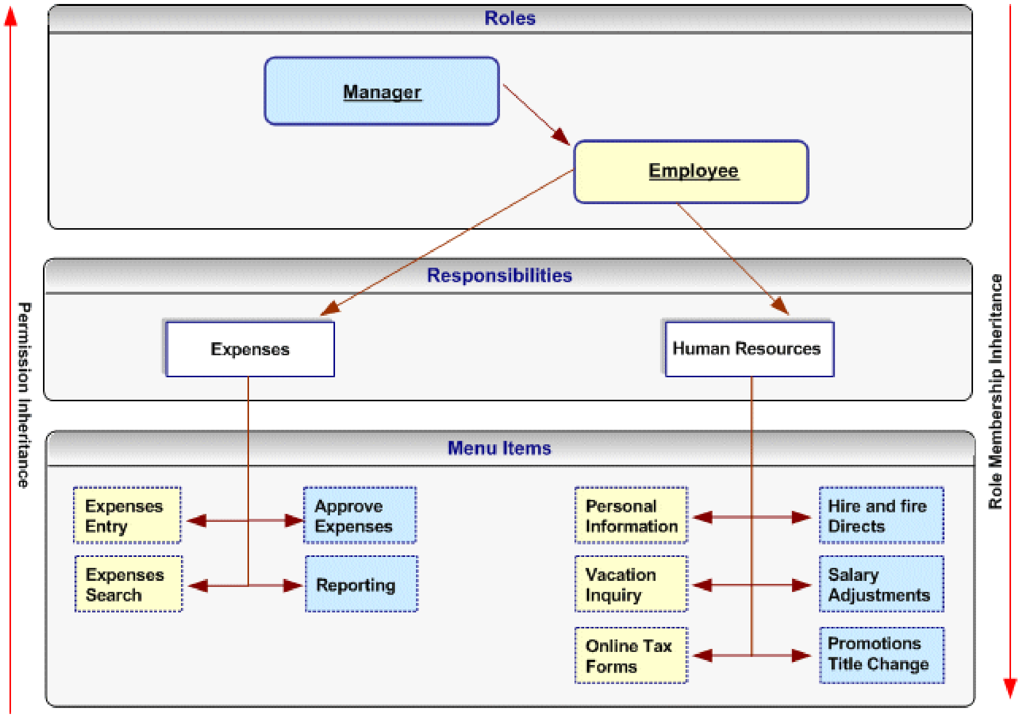
5. Define role inheritance relationships. For more information, see Defining Role Inheritance Hierarchies, page 3-7For example, the Manager role should inherit the Employee role, and the Employee role should inherit the Expenses and Human Resources responsibilities. The following figure illustrates a role inheritance relationship in which a role inherits the responsibilities that are inherited by its sub role:

Figure 3-4 Role Inheritance Relationship in Which a Role Inherits the Responsibilities Inherited by its Sub Role



6. Assign permissions to each role. For more information, see Assign permissions to each role, page 3-2. Each permission maps to a menu item (function) within the application (responsibility) that should be available to the users to whom the role is assigned. For example, an organization will grant the employee-related permissions from the Expenses and Human Resources responsibilities to the Employee role and will grant the manager-related permissions for these responsibilities to the Manager role. Consequently, the manager role will have access to all the menu items within these responsibilities but the Employee role will only have access to the Employee-related functions.

Figure 3-5 Permissions, Roles and Inheritance



Permissions assigned to a sub role in the role inheritance hierarchy are automatically inherited by the superior roles. For example, if you grant the permission for accessing the Online Tax Forms page to the Employee role, anyone with the Manager role will automatically have access to this page through role inheritance. Because the Hire and Fire Directs page is only granted to the Manager role, it is not available to users that are only assigned the Employee role.

Permissions are always assigned through permission sets, which represent named sets of functions (permissions). When determining what permissions (functions/menu items) should be granted to each role, you may have to create new permission sets, page 4-46. Menus and permission sets are stored in the same tables in the database; which means that they are interchangeable (both can be used) to assign permissions.

7. Optionally assign any additional permissions and data security policies to roles as required by each application.

Guidelines

Oracle User Management ships with the following Customer Administrator and Security Administrator roles. These roles illustrate how to setup Roles and Role Inheritance to determine user access within an application (responsibility). Both roles inherit the User Management responsibility but each role is granted different permissions and data security policies. The User Management responsibility has the grant flag removed for all functions (permissions) in the menu hierarchy. Instead, these permissions are granted to the role depending on each role's requirements:

Table 3-4 Role Attributes and Roles

Role Attributes	Customer Administrator	Security Administrator
Permission Sets	• User Maintenance UIs.	• User Maintenance UIs. • Setup screens. • Maintain system accounts.
User Administration	• Data security policies to manage people and user accounts for the customer administrator's own organization. • Typically, the Customer Administrator can only assign or revoke a subset of roles.	• Data security policies to manage all people and user accounts. • The Security Administrator can assign or revoke all roles.

Creating and Updating Registration Processes

Registration processes are predefined registration components that enable end users to perform some of their own registration tasks, such as requesting new accounts or requesting additional access to the system. They also provide administrators with a faster and more efficient method of creating new user accounts.

Oracle User Management provides three types of registration processes:

- Self Service Account Requests
- Requests for Additional Access
- Account Creation by Administrators

In the Oracle User Management Overview section, see Registration Processes, page 2-6.

Steps

Registration processes all use the same infrastructure and processing logic. Steps for defining a registration process will vary depending on the type of registration process you are creating.

1. Log on as a user that is assigned the Security Administrator role (typically as sysadmin), select the User Management responsibility in the navigator and then click the **Registration Processes** subtab.
2. Click the **Create Registration Process** button.
3. Enter the required information for the Registration Process Description and click the **Next** button. This information specifies:
   - **Role**. The role with which you optionally associate the registration process and that is assigned to the user at the end of the registration process once the request has been processed.
   - **Type**. The type of registration process you wish to create.
   - **Registration Process Code**. The unique identifier for the registration process.
   - **Display Name**. The display name for the registration process.

- **Description**. A description of the registration process.
- **Application**. The application with which the registration process is classified. This can be used to help query the registration process.
- **Active From**. The date from which the registration process is first active.
- **Active To**. The date you can optionally specify to terminate the registration process.

4. Enter the runtime execution information for the registration process and click the **Next** button. This information specifies:

   - **Registration Start Page**. The first page (which is represented as a function) in the registration process that captures any additional user registration information. This is optional unless you are creating a Self Service Account Request registration process.
   - **Notification Event**. The workflow business event that invokes a workflow. The notification workflow subscribes to the event and subsequently sends notifications to the approver or to the user.
   - **Approval Transaction Type**. The set of approval routing rules that is interpreted at runtime by the Oracle Approval Management rules engine. The rules determine whether approval is required and by what set of users based on user transaction types you have defined specifically for use with Oracle User Management.

5. Enter the eligibility information for the registration process by selecting the appropriate roles or groups from the Available Groups column and clicking the **Submit** button. For Requests for Additional Access, eligibility defines the users who are able to register for the role associated with the registration process. For Account Creation by Administrators, eligibility determines what administrators can register new users through the registration process. Oracle User Management ships with the following seeded permissions for defining eligibility policies:

Table 3-5 Seeded Permissions for Self Service Additional Access and Account Creation by Administrators Eligibility

Function Code	Display Name	Description
UMX_OBJ_ADMIN_CRTN_FLOW	Administrator Assisted Account Creation	Permission representing "Administrator Assisted Account Creation" registration processes. This must be granted as a data security policy on the Registration Process (UMX_REG_SRVC) business object.
UMX_OBJ_ROLE_ELGBLTY	Self Service Eligibility	Permission representing registration processes for additional access. Determines the set of end users that should be eligible to register for a given role/registration process. This must be granted as a data security policy on the Registration Process (UMX_REG_SRVC) business object.

6. Register subscriptions to the appropriate business events raised by Oracle User Management, and ensure that your subscription logic writes the registration data into the appropriate destination schemas.
7. Optionally update the registration process by searching for it and clicking the **Update** button in the search results page.
8. Optionally set the following profile options for registration processes of type Self Service Account Request:
   - **Registration Links**. Oracle User Management provides support for displaying different registration links on the login page based upon the mid-tier through which the login page is accessed. Organizations can set the server level profile option, “UMX: Register Here Link: Default Registration Process” (UMX_REGISTER_HERE_REG_SRV) to specify different destinations for the registration link.
   - **Registration Parameters**. The registration link can also contain additional parameters that are not known at design time. These parameters are available at all stages of the registration process; for example, for routing approval requests. You can set the server level profile option “UMX: Register Here Link: Default Registration Parameters” (UMX_REGISTER_HERE_REGPARAMS) for this purpose. The format for setting this profile option is: "ParamName1=ParamValue1&ParamName2=ParamValue2":
   - **UI-specific Parameters**. Organizations can additionally specify parameters used to control the rendering of the registration user interface, such as the menu displayed in the registration UI. The server level profile option, “UMX: Register Here Link: Default Html Parameters” (UMX_REGISTER_HERE_HTML PARAMS) can be set for this purpose. The format for setting this profile option is: "ParamName1=ParamValue1&ParamName2=ParamValue2":

9. Optionally set the UI attributes for the Login page using the profile option, Local Login Mask: FND_SSO_LOCAL_LOGIN_MASK. For the Login page to display one or more of these optional attributes, add the numeric values of all desired attributes, and set the value of the profile option to the corresponding value:

   - USERNAME_HINT = 01
   - PASSWORD_HINT = 02
   - CANCEL_BUTTON = 04
   - FORGOT_PASSWORD_URL = 08
   - REGISTER_URL = 16
   - LANGUAGE_IMAGES = 32
   - SARBANES_OXLEY_TEXT = 64

   For example to show PASSWORD_HINT and FORGOT_PASSWORD_URL only, set the profile option to 10 (02+08). To show just the LANGUAGE_IMAGES set the value to 32, which is the default.

   > **Note:** The Apache server may need to be restarted for the changes to take effect.

Setting Up The Forgot Password Feature

Oracle User Management includes a "Forgot Password" feature that can be used by local users (users whose passwords are not managed in the Oracle Internet Directory LDAP server) to request a password reset. The Forgot Password feature requires Identity Verification; the owner of the user account must confirm via email that the password should be changed. You can configure the reset password link by setting the required values for the Local Login Mask profile option, page 3-17. The Forgot Password feature uses the UMX Password (UMXUPWD) workflow.

Configuring the User Name Policy

The Oracle User Management registration infrastructure supports a *configurable user name policy*. This policy is used to generate a suggested user name in the sample user creation flows shipped with the application, as well as for validating the chosen user name format.

> **Note:** Oracle User Management is supplied with a default policy that identifies users by their email address.

Seeded User Name Policies

The following table lists the seeded user name policies that are shipped with Oracle Applications.

Table 3-6 Seeded User Name Policies

Code	Description
UMX_USERNAME_POLICY:EMAIL_ ADDRESS	User name policy with email address format defined as the policy.
UMX_USERNAME_POLICY:NONE	User name policy with no restriction on user name format.

Administrators can configure either of these seeded policies. In addition to these, custom policies can also be implemented if desired.

> **Note:** Refer to the UMX Developer's Guide on Oracle*MetaLink* for details of how to create a custom policy.

Configuration of user name policy is a three-stage process.

Stage 1 - Suggested User Name Generation Subscription Setup

1. Log on as a user that is assigned the Workflow Administrator Web Applications responsibility (typically sysadmin).
2. Go to Workflow Administrator Web Applications > Business Events
3. From the Business Events page, search for the Business Event with the name *oracle.apps.fnd.umx.username.generate*.
4. Click on the Subscription icon to go to the Subscriptions page.
5. For the subscription corresponding to the policy, change the status to "Enabled".

Stage 2 - Validation Event Subscription Setup

1. Log on as a user that is assigned the Workflow Administrator Web Applications responsibility (typically sysadmin).
2. Go to Workflow Administrator Web Applications > Business Events
3. From the Business Events page, search for the Business Event with the name *oracle.apps.fnd.user.name.validate*.
4. Click on the Subscription icon to go to the Subscriptions page.
5. For the subscription corresponding to the policy, change the status to "Enabled".

Stage 3 - Profile Option Setup

1. Log on as a user that is assigned the Functional Administrator responsibility (typically sysadmin).
2. Go to Functional Administrator > Core Services > Profiles
3. Search with the Profile Name of UMX: User Name Policy in the Maintain Profile Options page.
4. Click on the Update icon to go to the Update Profile Option page.
5. Choose a value corresponding to the policy and click on the Apply button.

Additional Requirements

- In all the three of the stages above, the values set must correspond to the same user name policy.
- The Listener and JVMs must be restarted after the user name policy is changed.

Delegated Administration Tasks

The Delegated Administration layer of Access Control in Oracle Applications enables local administrators to perform a variety of specifically defined administrative tasks. Once they are assigned the appropriate roles, local administrators manage the subset of users and people to which they have access by creating, updating, or disabling accounts, granting or revoking a limited subset of their organization's roles, and changing passwords.

Maintaining People and Users

Oracle User Management enables local administrators to manage people and users in the system. People are individuals in the system who may or may not possess a user account, whereas users are individuals in the system who possess user accounts. In addition, system administrators can also manage system accounts, such as a Guest account, that are not linked to people.

Typically, people and users are managed by local administrators, who can perform the following tasks:

- Register new people (optional: requires access to have been granted to the "Account Creation by Administrators" registration process)
- Create, update, or disable user accounts
- Reset passwords
- Grant users access to different parts of the system by assigning or revoking roles

Common Prerequisites

The following are prerequisites for performing any delegated administration task listed in the preceding section. Each task may have additional prerequisites:

- A role that is granted the *User Maintenance UIs* (UMX_USER_ADMIN_UI_PERMS) permission set. The role must also inherit the User Management responsibility.
- Appropriate privileges for User Administration, Role Administration, and Organization Administration.
- The Query Person Details (UMX_PERSON_OBJECT) permission for the set of people and administrator can manage.
- Optionally, the Edit Person Details (UMX_OBJECT_EDIT_PERSON) permission for the set of people that the administrator can manage.
- For system administrators, the Maintain System Accounts (UMX_SYSTEM_ACCOUNT_ADMINISTRATION) permission.

Steps

1. Navigate to the **User Management** responsibility and then click the **Users** subtab.
2. Use the search fields to locate the required people or users.
3. Manage the generated list of people or users by clicking the required icon and performing the necessary steps in the resulting window. Options for managing people and users vary depending on the permissions assigned to the administrator. Oracle User Management ships with the following basic and advanced options for maintaining people and users:
   - Query users
   - Edit personal information
   - Reset password
   - Maintain account information (create, inactivate, reactivate accounts)

- Maintain system accounts
- Assign or revoke roles

Creating, Inactivating, and Reactivating User Accounts

Administrators can create a user account for any person in the system who does not already possess one.

Prerequisites

To create, inactivate, and reactivate user accounts, an administrator must be assigned the following:

- Common prerequisites, as detailed in the Maintain People and Users section, Common Prerequisites, page 3-19.
- The Create, Inactivate, Reactivate User Account (UMX_OBJ_ACTIVATE_ACCT) permission for the set of people that the administrator can manage.

By default, user names are derived from the person's email address.

Steps

1. Log in as a user with a role granting you access to the User Management responsibility, select the User Management responsibility in the navigator and click the **Users** subtab.
2. Search for the person for whom you wish to create an account and then click the **Create Account** icon next to the person's name if the account does not already exist. Your search will only generate results for the subset of users that you are eligible to manage.
3. Enter or modify the required information and click the **Submit** button.

Guidelines

Oracle recommends that you base user names on the person's email address.

Resetting User Passwords

Oracle User Management enables administrators to reset passwords for the set of users in the system that they manage. When the password is reset, an email message is sent to the user using the UMX Password (UMXUPWD) workflow.

Prerequisites

To reset user passwords, an administrator must be assigned the following:

- In the Maintain People and Users section, see the Common Prerequisites, page 3-19.
- The Reset Password (UMX_OBJ_PASSWD_MGMT) permission for the users that the administrator can manage

Steps

1. Log in as a user with a role granting you access to the User Management responsibility, select the User Management responsibility in the navigator and click the **Users** subtab.
2. Use the Search field to locate the user whose password you wish to change and then click the **Reset Password** icon next to the user.

3. Select one of the following options, provide any required information and click the **Submit** button.

   - **Generate Automatically**. No additional information is required and the system automatically generates the new password.
   - **Enter Manually**. The system prompts you to enter the password and a confirmation of the password.

The person for whom you reset the password receives and email notification stating that the password has expired and must be reset the next time the user logs in. This notification is sent by the UMX Password (UMXUPWD) workflow.

Assigning Roles to or Revoking Roles from Users

Oracle User Management enables administrators to assign roles to or revoke roles from the subset of users that they manage. Administrators can only assign or revoke the roles for which they have been granted administrative privileges.

Prerequisites

To assign roles to or revoke roles from users, an administrator must be assigned the following:

- Common prerequisites from the Maintain People and Users section, Common Prerequisites, page 3-19.
- The appropriate administrative privileges for the role the administrator assigns or revokes. For more information, see Defining Role Administration Privileges for Roles, page 3-5.

Steps

1. Log in as a user with a role granting you access to the User Management responsibility, select the User Management responsibility in the navigator, and click on the **Users** subtab.
2. Search for the person for whom you wish to assign or revoke roles, click the **Update** icon next to the person's name, and click on the **Roles** subtab.
3. To assign a role to the user, click the **Assign Roles** button and select the desired role.
4. To remove a role, you must end-date the role. If the role is an inherited role, you can only remove it by removing the role from which it originates in the role inheritance hierarchy. You can view a role's inheritance hierarchy by clicking on the **Show** hyperlink next to the role.

Guidelines

The administrator can only grant or revoke roles for which he has the appropriate privileges. When granting or revoking roles, the administrator bypasses any approval processes. If a registration process exists for the role, it will be invoked and the request will be handled by the Oracle User Management registration engine. If not, then the role is assigned directly. If the role is associated with a registration process for existing users and the registration process has a reference for capturing additional information, then the "Additional Information Required" link is rendered. The administrator must click on this link and provide any required additional information before the request is processed.

Registering External Organization Contacts

Oracle User Management provides a sample registration process that enables administrators to register new people for their organizations. Organizations can use the sample registration process directly or reference it as an example of how to define their own administration registration processes.

Prerequisites

To register new people, an administrator must be assigned the following:

- The common prerequisites detailed in the the Maintain People and Users section, Common Prerequisites, page 3-19.
- The necessary privileges to invoke the specific administrative account creation registration processes; these are defined as part of the registration process definition.
- Organization Administration privileges for all organizations for which an administrator needs to be able to register new people.

Steps

1. Log in as a user with a role granting you access to the User Management responsibility, select the User Management responsibility in the navigator and click the **Users** subtab.
2. In the Register dropdown list, select administrative account registration process you wish to invoke, and click the **Go** button.
3. Enter the information required by the registration process as defined by the registration UI for the registration process, click the **Submit** button and then click the **OK** button in the resulting page.

Self Service Features

Implementors and administrators can verify the successful configuration of end user functions by performing the tasks described in this section.

Self Service Registration

Oracle User Management enables users to register for access to applications without requiring assistance from administrators. To register for application access, users must provide information in the required fields and click the **Submit** button.

Oracle User Management ships with the following sample self service registration processes:

- Employee Self Service Registration
- Customer Self Service Registration (external individuals)

Organizations can use these registration processes in their existing form or can use them as references for developing their own registration processes.

Requesting Additional Application Access

Oracle User Management enables you to request additional access to the specific applications for which you are eligible. Application access is based on roles and to access an application you must be granted the appropriate role. Perform the following to view the roles you have been assigned and to request additional ones.

Steps

1. After logging into the system, click the **Preferences** link in the upper right corner, and click the **Access Requests** link in the sidebar menu. The Access Requests page displays the roles you have been assigned. Click the **Request Access** button to request one or more additional roles.
2. Most roles are organized according to role categories: roled that are not categorized appear under the Miscellaneous node. Select the role category that contains the role you want to request. If you do not see the required role, then either you are not eligible for the role or it has not been set up to for additional access requests.
3. Select the role or roles you require for additional access to the system, and click on the **Add to List** button. You can optionally remove roles from your list by clicking on the **Remove Roles** button.
4. When you have selected all your required roles, click on the **Next** button.
5. Enter a justification for your request and click on the **Next** button. You can remove any pending roles or check their status in the page that appears next.

Guidelines

Some roles may require you to provide additional information. In such cases, the system will prompt you for additional information before you can complete the process for requesting a role.

Reset Forgotten Password

If you have forgotten your password, Oracle User Management enables you to reset it from the login page by clicking the forgot password link. Enter your user name in the User Name field and click the **Submit** button. After you submit your user name to the system, you will receive a verification email message. You must respond to this message to receive your new password.

4

Oracle Application Object Library Security

Overview of Oracle Applications Security

As System Administrator, you define Oracle Applications users, and assign one or more responsibilities to each user.

Defining Application Users

You allow a new user to sign-on to Oracle Applications by defining an *application user*. An application user has a username and a password. You define an initial password, then the first time the application user signs on, they must enter a new (secret) password.

When you define an application user, you assign to the user one or more responsibilities. If you assign only one responsibility, the user, after signing on, immediately enters an application.

If you assign two or more responsibilities, the user, after signing on, sees a window listing available responsibilities.

Case Sensitivity in Oracle Applications User Passwords

In previous releases of Oracle Applications, user passwords were treated as case insensitive. Now, Oracle Applications user passwords can optionally be treated as case sensitive, depending on the mode you choose.

Case-sensitivity in passwords is controlled by the site-level profile option *Password Case Option*. This profile has three possible settings:

- Insensitive (or unset) - Passwords are treated as case insensitive. In Insensitive mode, passwords are stored and compared in uppercase. This option is the default behavior, similar to that in earlier releases. The entered password and the decrypted password are converted to uppercase prior to comparison. New passwords are converted to uppercase prior to encryption.
- Sensitive - (New behavior) Passwords are stored and compared as they are, with the password case preserved. During comparison, if the entered password does not match the decrypted version, then an error message is displayed. New passwords are unaltered prior to encryption.
- Mixed - (New behavior) This mode extends Sensitive mode by adding insensitive behavior for the benefit of those users who have not yet changed their passwords since case-sensitive passwords were enabled. If the initial password comparison fails, the entered password is converted to uppercase and compared again. New passwords are unaltered prior to encryption.

To enable case sensitivity, you must expicitly set the Password Case Option to either 'Sensitive' or 'Mixed'. If you want to preserve case insensitivity in passwords, i.e. retain the behavior from previous releases, ensure that Password Case Option is either set to 'Insensitive', or not set at all.

There are no upgrade or data migration issues with this new feature. If users are migrated from an environment with case-insensitive passwords to an environment in the Sensitive mode, then users need to log in with their old passwords in uppercase. If the mode for an environment is changed from Insensitive or Mixed to Sensitive, users will need to log in with their old passwords in uppercase. Once logged in, users can update their passwords if they wish.

Responsibilities Define Application Privileges

A *responsibility* is a level of authority in Oracle Applications that lets users access only those Oracle Applications functions and data appropriate to their roles in an organization.

Each responsibility allows access to:

- A specific *application* (or applications) such as Oracle General Ledger or Oracle Planning.
- A *set of books*, such as U.S. Operations or German Sales, or an *organization*, such as New York Manufacturing or New York Distribution.
- A restricted list of windows that a user can navigate to; for example, a responsibility may allow certain Oracle Planning users to enter forecast items, but not enter master demand schedule items.
- A restricted list of functions a user can perform. For example, two responsibilities may have access to the same window, but one responsibility's window may have additional function buttons that the other responsibility's window does not have.
- Reports in a specific application; as system administrator, you can assign groups of reports to one or more responsibilities, so the responsibility a user choose determines the reports that can be submitted.

Each user has at least one or more responsibilities, and multiple users can share the same responsibility. A system administrator can assign users any of the standard responsibilities provided with Oracle Applications, or create new custom responsibilities if required.

HRMS Security

The Human Resources Management Systems (HRMS) products have an additional feature using Security Groups. For more information, see: *Customizing, Reporting, and System Administration in Oracle HRMS*.

Related Topics

Defining a Responsibility, page 4-3

Defining a Request Security Group, page 4-4

Overview of Function Security, page 4-7

Form Functions, page 4-25

Responsibilities, page 4-18

Users Window, page 4-22

Defining a Responsibility

When you define a responsibility, you assign to it some or all of the components described below.

Data Group (Required)

A data group defines the mapping between Oracle Applications products and ORACLE database IDs. A data group determines which Oracle database accounts a responsibility's forms, concurrent programs, and reports connect to. See: Defining Data Groups, *Oracle Applications System Administrator's Guide - Configuration*.

Request Security Group (Optional)

A request security group defines the concurrent programs, including requests and request sets, that may be run by an application user under a particular responsibility. See: Defining a Request Security Group, page 4-4. See: Organizing Programs into Request Groups, *Oracle Applications System Administrator's Guide - Configuration*.

Menu (Required)

A menu is a hierarchical arrangement of application functions (forms) that displays in the Navigate window. Menus can also point to non-form functions (subfunctions) that do not display in the Navigate window, but that define the range of application functionality available for a responsibility. Each responsibility is associated with a menu. See: Overview of Function Security, page 4-7.

Function and Menu Exclusions (Optional)

A responsibility may optionally have function and menu exclusion rules associated with it to restrict the application functionality enabled for that responsibility. See: Overview of Function Security, page 4-7.

Additional Notes About Responsibilities

Predefined Responsibilities

All Oracle Applications products are installed with predefined responsibilities. Consult the reference guide for your Oracle Applications product for the names of those predefined responsibilities.

Additionally, instances of the major components that help define a responsibility (data groups, request security groups, menus, and functions) are predefined for Oracle Applications.

Responsibilities and Request Security Groups

When a request group is assigned to a responsibility, it becomes a *request security group*.

From a standard submission form, such as the Submit Requests form, users can run only the reports, concurrent programs, and request sets that are in their responsibility's request security group.

- If you do not include the Submit Requests form on the menu for a responsibility, then you do not need to assign a request security group to the responsibility.
- If a request security group is not assigned to a responsibility, then users working under that responsibility cannot run any reports, request sets, or other concurrent programs from a standard submission form.

Responsibilities and Function Security

Oracle Applications GUI-based architecture aggregates several related business functions into a single form. Parts of an application's functionality may be identified as individual Oracle Applications functions, which can then be secured (i.e. included or excluded from a responsibility).

See: Overview of Function Security, page 4-7

Defining a Request Security Group

Request security groups are used to organize requests and request sets for user access control. Beyond this short introduction, request groups and request security groups are discussed in greater detail, as part of a broader range of topics not necessarily limited to application security, in *Oracle Applications System Administrator's Guide - Configuration*.

Using Request Security

You use request security to specify the reports, request sets, and concurrent programs that your users can run from a standard submission form, such as the Submit Requests form.

To set up request security, you define a request group using the Request Groups form. Using the Responsibilities form, you assign the request group to a responsibility. The request group is then referred to as a *request security group*. See: Request Security Groups, *Oracle Applications System Administrator's Guide - Configuration*.

You can define a request group to contain single requests, request sets, or all the requests and request sets in an application.

If you choose to include all the requests and requests sets in an application, the user has automatic access to any new requests and request sets (without owners) in the future.

A request security group can contain requests and request sets from different applications. If you want to define request security groups that own requests from different applications, refer to the discussion on Data Groups. See: Defining Data Groups, *Oracle Applications System Administrator's Guide - Configuration*.

> **Note:** A *request security group* or *request group* is not the same as a *security group*.

Individual Requests and Request Sets

Reports or concurrent programs which are not included in a request security group on an individual basis, but do belong to a request set included in a request security group, have the following privileges:

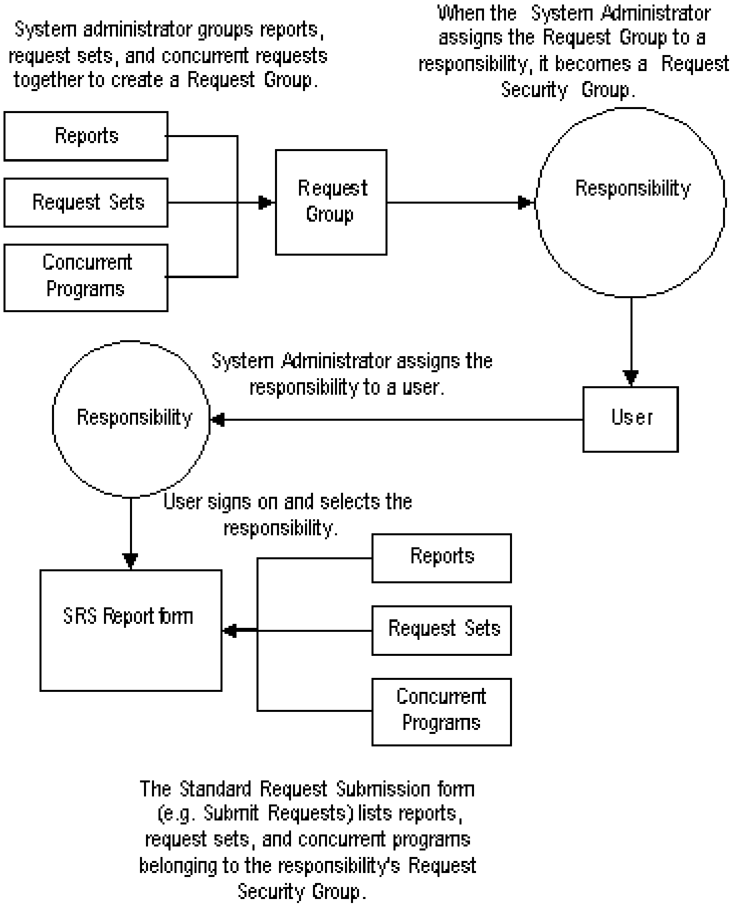
- Users cannot use the Submit Requests form to run single requests and request sets that are not in their responsibility's request security group.
- Users can, however, run request sets that contain requests that are not in their request security group, if the request set is in their request security group.

If you assign a request set, but not the requests in the set, to a request security group, the user:

- Can edit the request set by deleting requests from it or adding other requests to it, only if the user is the assigned owner of the request set.
- Cannot edit request information in the request set definition.
- Cannot stop specific requests in the set from running.

The Request Security Groups figure below illustrates the relationship between a request security group, application user, and a responsibility.

Figure 4-1 Responsibilities, Request Groups, and Request Security Groups


System administrator groups reports, request sets, and concurrent requests together to create a Request Group.
When the System Administrator assigns the Request Group to a responsibility, it becomes a Request Security Group.
Reports
Request Sets
Concurrent Programs
Request Group
Responsibility
System Administrator assigns the responsibility to a user.
Responsibility
User
User signs on and selects the responsibility.
SRS Report form
Reports
Request Sets
Concurrent Programs
The Standard Request Submission form (e.g. Submit Requests) lists reports, request sets, and concurrent programs belonging to the responsibility's Request Security Group.

Related Topics

Request Sets and Owners, *Oracle Applications System Administrator's Guide - Configuration*

Overview of Oracle Applications Security, page 4-1

Defining a Responsibility, page 4-3

Form Functions, page 4-25

Menus, page 4-29

Responsibilities, page 4-18

Users, page 4-22

User Session Limits

Using the following profile options you can specify limits on user sessions.

ICX:Session Timeout

Use this profile option to enforce an inactivity time-out. If a user performs no Oracle Applications operation for a time period longer than the time-out value (specified in minutes), the user's session is disabled. The user is provided an opportunity to re-authenticate and re-enable a timed-out session. If re-authentication is successful, the session is re-enabled and no work is lost. Otherwise, Oracle Applications exits without saving pending work.

If this profile option to 0 or NULL, then user sessions will never time out due to inactivity.

ICX: Limit time

Use this profile option to specify the absolute maximum length of time (in hours) of any user session, active or inactive.

ICX: Limit connect

Use this profile option to specify the absolute maximum number of connection requests a user can make in a single session.

Overview of Security Groups in Oracle HRMS

Security groups, used exclusively by Oracle HRMS, allow data to be partitioned in a single installation. A single installation can use a particular set of configuration data, but store data for multiple clients, where the data is partitioned by security groups. A user with an assignment of one security group can only access data within that security group.

A security group represents a distinct client or business entity. Data that must be distinct for each client in an installation is partitioned by security group. All other data is shared across all security groups.

Security is maintained at the level of responsibility/security group pairs. That is, users are assigned specific responsibilities within each security group. A user may be assigned a global responsibility that is valid in all security groups. When signing on to Oracle Applications, a user, if assigned more than one responsibility, will be asked to choose a responsibility and security group pair. Partitioned data accessed through security group sensitive views will show only data assigned to the current security group.

Defining Security Groups

Every installation will have a single "Standard" security group seeded in. If no other security groups are created, this single group will be hidden from users when they sign on.

In the Users form, you assign a security group when you assign a responsibility.

For more information, see: *Configuring, Reporting and System Administration in Oracle HRMS*.

Overview of Function Security

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

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

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

Terms

Function

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

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

Form (Form Function)

A form function (*form*) invokes an Oracle Forms form. Form functions have the unique property that you may navigate to them using the Navigate window.

Subfunction (Non-Form Function)

A non-form function (*subfunction*) is a securable subset of a form's functionality: in other words, a function executed from within a form.

A developer can write a form to test the availability of a particular subfunction, and then take some action based on whether the subfunction is available in the current responsibility.

Subfunctions are frequently associated with buttons or other graphical elements on forms. For example, when a subfunction is enabled, the corresponding button is enabled.

However, a subfunction may be tested and executed at any time during a form's operation, and it need not have an explicit user interface impact. For example, if a subfunction corresponds to a form procedure not associated with a graphical element, its availability is not obvious to the form's user.

Menu

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

Menu Entry

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

Responsibility

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

- The functions that the user may access. Functions are determined by the menu assigned to the responsibility.
- The concurrent programs, such as reports, that the user may run.
- The application database accounts that forms, concurrent programs, and reports connect to.

Related Topics

How Function Security Works, page 4-9

Form Functions, page 4-25

Forms and Subfunctions , page 4-8

Functions, Menus, and the Navigate Window, page 4-9

Overview of Oracle Applications Security, page 4-1

Implementing Function Security, page 4-10

Forms and Subfunctions

A form is a special class of function that differs from a subfunction in two ways:

- Forms appear in the Navigate window and can be navigated to. Subfunctions do not appear in the Navigate window and cannot be navigated to.
- Forms can exist on their own. Subfunctions can only be called by logic embodied within a form; they cannot exist on their own.

A form as a whole, including all of its program logic, is always designated as a function. Subsets of a form's program logic can optionally be designated as subfunctions if there is a need to secure those subsets.

For example, suppose that a form contains three windows. The entire form is designated as a function that can be secured (included or excluded from a responsibility.) Each of the form's three windows can be also be designated as functions (subfunctions), which means they can be individually secured. Thus, while different responsibilities may include this form, certain of the form's windows may not be accessible from each of those responsibilities, depending on how function security rules are applied.

Related Topics

Overview of Function Security, page 4-7

Functions, Menus, and the Navigate Window, page 4-9

How Function Security Works, page 4-9

Functions, Menus, and the Navigate Window

Form functions or *forms* are selected using the Navigate window. The arrangement of form names in the Navigate window is defined by the menu structure assigned to the current responsibility.

The following types of menu entries are not displayed by the Navigate window:

- Subfunctions
- Menus without Entries
- Menu Entries without a Prompt

If none of the entries on a menu are displayed by the Navigate window, the menu itself is not displayed.

Menu Entries with a Submenu and Functions

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

A function is considered executable if it can be executed directly from the current running user interface. For example, an Oracle Applications form using Oracle Forms is an executable function from within Oracle Forms, but not within the Self Service applications.

How Function Security Works

Registering Functions

- Developers can require parts of their Oracle Forms code to look up a unique *function name*, and then take some action based on whether the function is available in the current responsibility.
- Developers can register functions. They can also register parameters that pass values to a function. For example, a form may support data entry only when a function parameter is passed to it.

> **Warning:** In general, system administrators should not modify parameters passed to predefined functions for Oracle Applications products. The few exceptions are documented in the relevant manuals or product notes.

- Typically, developers define a menu including all the functions available in an application (i.e. all the forms and their securable subfunctions). For some applications, developers may define additional menus that restrict the application's functionality by omitting specific forms and subfunctions.
- When developers define menus of functions, they typically group the subfunctions of a form on a subfunction menu they associate with the form.

Excluding Functions

- Each Oracle Applications product is delivered with one or more predefined menu hierarchies. System Administrators can assign a predefined menu hierarchy to a responsibility. To tailor a responsibility, System Administrators exclude functions or menus of functions from that responsibility using exclusion rules.
- If System Administrators cannot create the desired menu by applying exclusion rules to a predefined menu, they can define a new menu hierarchy. In this case, we recommend that they construct their menu hierarchy using forms and their associated menus of subfunctions. In other words, System Administrators should leave the developer-defined associations between forms and their menus intact.

Available Functions Depend on the Current Responsibility

- When a user first selects or changes their responsibility, a list of functions obtained from the responsibility's menu structure is cached in memory.
- Functions a System Administrator has excluded from the current responsibility are marked as unavailable.
- Form functions in the function hierarchy (i.e. menu hierarchy) are displayed in the Navigate window. Available subfunctions are accessed by working with the application's forms.

Visibility of Excluded Functions

Some subfunctions are associated with a graphical element, for example, a button, and their exclusion may result in:

- Dimming of the button
- Absence of the button

Other subfunctions may not correspond to a graphical element, and their exclusion may not be obvious to an end user.

Related Topics

Overview of Function Security, page 4-7

Forms and Subfunctions , page 4-8

Overview of Oracle Applications Security, page 4-1

Form Functions, page 4-25

Implementing Function Security

A "full access" responsibility with a menu that includes all the functions in an application is predefined for each Oracle Applications product. Some applications may provide additional predefined responsibilities that include a smaller set of functions (i.e. fewer forms and subfunctions).

As a System Administrator, you can restrict the functionality a responsibility provides by defining rules to exclude specific functions or menus of functions. In fact, we recommend that you use exclusion rules to customize a responsibility in preference to constructing a new menu hierarchy for that responsibility.

For example, suppose you want to customize a responsibility to restrict the functionality of a form included in that responsibility. First, you examine the predefined menus that group the subfunctions associated with that form. Then, using exclusion rules, you can restrict the form's functionality by excluding certain of the form's subfunctions from the responsibility.

If you cannot create the responsibility you need by applying exclusion rules, you may build a custom menu for that responsibility using predefined forms (i.e. form functions) and their associated menus of subfunctions. However, we recommend that you do not disassociate a form from its developer-defined menus of subfunctions.

Securing Functions Using Predefined Menus

Use the Responsibilities form to:

- Limit a predefined responsibility's functionality by excluding menus and functions from it.
- Define a new responsibility and assign a predefined menu to it. Customize the new responsibility's functionality by excluding menus and functions.
- By assigning the same menu hierarchy to different responsibilities and excluding different functions and menus, you can easily customize an application's functionality.

Securing Functions Using New Menus

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

- Use forms and their associated menus of subfunctions to define new menus.

Assign the menu structure to a new responsibility using the Responsibilities form.

- For that responsibility, tailor a form's functionality by excluding particular subfunctions.
- By excluding a subfunction executed from within a form, the functionality of that form can be varied from one responsibility to another.
- By applying exclusion rules to the predefined menus of subfunctions associated with a form, you can easily customize a form's functionality.

Excluding Functions from a Responsibility

A system administrator may exclude functions or menus from the menu structure assigned to a responsibility.

- When a menu is excluded, all of its menu entries, that is, all the functions and menus of functions that it selects, are excluded.

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

  > **Note:** If your product (e.g. Oracle HRMS) uses task flows, excluding a function from a responsibility using Menu Exclusions does *not* exclude the function from any task flow for that responsibility. If you do not want the function to be accessible from the task flow, you must update the task flow definition accordingly.

Defining a New Menu Structure

When defining a new menu structure:

- Create a logical, hierarchical listing of functions. This allows for easy exclusion of functions when customizing the menu structure for different responsibilities.
- Create a logical, hierarchical menu that guides users to their application forms.

Tasks for Defining a Custom Menu Structure

- Determine the application functionality required for different job responsibilities.
- Identify predefined menus, forms, and form subfunctions to use as entries when defining a new menu. Understand predefined menus by printing Menu Reports using the Submit Requests window.

  > **Tip:** To simplify your work, use predefined menus for your menu entries. You can exclude individual functions after a menu structure is assigned to a responsibility.

- Plan your menu structure. Sketch out your menu designs.
- Define the lowest-level menus first. A menu must be defined before it can be selected as an entry on another menu.
- Assign menus and functions to higher-level menus.
- Assign menus and functions to a top-level menu (root menu).
- Document your menu structure by printing a Menu Report.

  > **Warning:** Always start with a blank Menus form (blank screen). See Notes About Defining Menus, below.

Notes About Defining Menus

Build Menus From Scratch

- Menus cannot be copied. Menu definitions cannot be saved under a different name (i.e. there is no “Save As" capability).
- When a menu name displays in the Menus form, be sure you are in Query mode before overwriting the menu's name.

Define Menus for Fast and Easy Keyboard Use

- Design menu prompts with unique first letters, so typing the first letter automatically selects the form or menu.

- Design the sequence of menu prompts with the most frequently used functions first (i.e. lower sequence numbers).
- Entries cannot be copied from one menu definition to another.

Note when Changing Menu Names or Modifying Entries

- When you change a menu's name, the menu entries are not affected. The menu's definition exists under the new name.
- Other menus calling the menu by its old menu name automatically call the same menu by its new (revised) name.
- When defining menus or selecting a "root" menu to assign to a responsibility, the old menu name is not in a list of values.
- When modifying a predefined menu, all other menus that call that menu display the menu's modifications.
- For example, if you modify GL_TOP by adding another prompt that calls a form function, all menus that call GL_TOP will display the additional prompt when GL_TOP displays.

Menu Compilation

The Compile Security (FNDSCMPI) concurrent program is used to compile menus so that the system can more quickly check if a particular function is available to a particular responsibility/menu.

You should compile your menus after you make changes to your menu data. A request for this concurrent program is automatically submitted after you make changes using the Menus form.

After you apply a patch that includes menu changes, you should also run this concurrent program. You can do this through the AD Administration utility. For more information, see: *Maintaining Oracle Applications*.

Related Topics

Menus Window, page 4-29

Compile Security Concurrent Program, page 4-48

Preserving Custom Menus Across Upgrades

Preserve custom menus during upgrades of Oracle Applications by using unique names for your custom menus. For example, you can start the menu's name with the application short name of a custom application. Define a custom application named *Custom General Ledger*, whose application short name is XXCGL. Define your custom menu names to start with XXCGL, for example, XXCGL_MY_MENU.

Remember that the Oracle Applications standard menus may be overwritten with upgrade versions. Therefore, if you attached your custom menu as a submenu to one of the preseeded Oracle Applications menus, recreate the attachment to it following an upgrade. An alternative is to attach a standard Oracle Applications menu as a submenu to your custom menu; the link from your custom menu to the standard menu should survive the upgrade.

Related Topics

Overview of Oracle Applications Security, page 4-1

Overview of Function Security, page 4-7

Implementing Function Security, page 4-10

Special Function for Oracle HRMS, Oracle Sales and Marketing, page 4-14

Summary of Function Security, page 4-14

Form Functions, page 4-25

Function Security Reports, page 4-48

Special Function for Oracle HRMS, Oracle Sales and Marketing

In most Oracle Applications products, you can open multiple forms from the Navigator window without closing the form you already have open. However, when you define a new responsibility whose custom menu accesses Oracle Sales and Marketing forms, or Oracle HRMS task flows, you must include the function *Disable Multiform, Multisession* as an entry on the responsibility's top-level menu.

> **Tip:** You can identify an Oracle Sales and Marketing form by the OSM prefix contained in the form's function name.

In Oracle HRMS, a task flow is a method of linking windows so that you carry information from one window to the next, in sequence, to complete a task. You can identify an Oracle HRMS form that may be part of a task flow by the PER or PAY prefix in the form's function name. For details on administering Oracle HRMS task flows, and on determining whether a form is part of a task flow, see the Oracle HRMS documentation.

> **Important:** You should not include the *Disable Multiform, Multisession* function on menus that do not include either Oracle Sales and Marketing or Oracle HRMS forms.

To include the *Disable Multiform, Multisession* function on a menu:

- Add a Function menu entry to the top-level menu (i.e. the menu referenced by your new responsibility).
- Select the function whose User Function Name and Function Name are:
  - Disable Multiform, Multisession
  - FND_FNDSCSGN_DISABLE_MULTIFORM
- Save your changes.

Related Topics

Overview of Function Security, page 4-7

Implementing Function Security, page 4-10

Summary of Function Security

Functions:

- A function is a set of code in Oracle Applications that is executed only if the name of the function is present in a list maintained on a responsibility-by-responsibility basis.
- Functions can be excluded from a responsibility by a System Administrator.
- There are two types of function: a form function or *form*, and a non-form function or *subfunction*. A subfunction represents a securable subset of a form's functionality.

Form Functions:

- A function that invokes a form.
- Form functions appear in the Navigate window and can be navigated to.

Subfunctions:

- A function that is executed from within a form. Subfunctions can only be called by logic embodied within a Form Function.
- Subfunctions do not appear in the Navigate window and cannot be navigated to.

Menus:

- Menus contain menu entries which point to a function, another menu, or a function *and* another menu.
- Menus appear in the Navigate window.
- Menus can be excluded from a responsibility by a System Administrator.

Related Topics

Overview of Function Security, page 4-7

Implementing Function Security, page 4-10

Defining a New Menu Structure , page 4-12

Special Function for Oracle HRMS, Oracle Sales and Marketing, page 4-14

Form Functions, page 4-25

Function Security Reports, page 4-48

Overview of Oracle Applications Security, page 4-1

Overview of Data Security

Data Security allows administrators to control user access to specific data, as well as what functions users can apply to that data.

Concepts and Definitions

Objects

Data Security uses the concept of an Object to define the data records that are secured.

Object

Data security permissions are managed on objects. Business entities such as Projects and Users are examples of objects. Only a securable business-level concept should be registered as an object.

An object definition includes the business name of the object and identifies the main table and primary key columns used to access the object.

Object Instance

An object instance is a specific example of an object, such as Project Number 123 or User JDOE. An object instance generally corresponds to a row in the database. An instance is identified by a set of one or more primary key values as defined by the object.

In addition, "All Rows" for an object indicates all data rows of the object.

Object Instance Set

An object instance set is a group of related object instances within an object. A set is specified as a predicate on the keys or attributes of an object, expressed as a SQL "WHERE clause". All instances that satisfy the predicate are considered members of the object instance set. For example:

```
STATUS = ’ACTIVE’
```

could determine a set of object instances with the "Active" status.

The specific instances in the set can vary over time as object instance attributes change, or as new object instances are created.

An example is:

```
OWNER = FND_GLOBAL.USER_ID
```

The predicate can also be parameterized, so that the logic can define instance sets as a function of one or more input parameters. An example is:

```
COLOR = :PARAM1
```

Object instance sets are also called "data instance sets".

Users and Groups

Privileges given to users and groups determine their access to secured objects.

The data security system allows you to assign privileges to groups of users instead of assigning privileges to each user individually.

Users

Users are individuals who have access to software applications at a particular enterprise.

A user must have a unique name and should map one-to-one with an individual human or system. "Group" accounts are not correct uses of the user entity.

Groups

Users can belong to Groups. The grouping can come from position or organization relationships modeled in applications such as Oracle Human Resources. Alternatively, ad-hoc groups can be created explicitly for security purposes. A group is sometimes referred to as a role.

Functions

A function is the smallest unit of securable product functionality. You can register function definitions with the security system to represent actions that can be performed on an object or on the system in general. Granting a function to a set of users gives them permission to perform that function, and so a function may also be referred to as a permission.

There are two broad categories of functions: executable functions and abstract functions:

- An *executable function* can be invoked from a generic navigator user interface. An executable function definition must contain all information necessary to launch the function; often this includes the form name or URL plus parameters.
- An *abstract function* does not refer to a specific piece of code, but represents permission to perform a higher-level business action. The code that implements an abstract function calls the function security system to test whether the abstract function is granted. The system only allows the action if the abstract function is granted.

Permissions

A permission is the smallest unit of securable actions that can be performed on the system. These can either be abstract permissions or executable functions (menu). They can either be a system level permission or be sensitive to a data context. Example: A particular JSP page (executable) or View Person (abstract).

Navigation Menus and Permission Sets

Functions are grouped into related sets so that administration of these functions can be performed in higher-level business terms.

Although there are different types of function groupings, the same data structure is used to store them: menus and menu entries.

A navigation menu is simply a named container for a set of menu entries. Each menu entry points to a function and/or a sub-menu. The same sub-menu can be included on many parent menus. The resulting structure for these relationships is referred to as a hierarchy. This data structure is used to organize functions for two distinct purposes: navigation menus and permission sets.

Permission sets are sets of functions that could be granted to a user in order to allow them to perform a specific business operation, role, or responsibility. These are only used as a definition for the set of distinct functions they contain.

Grants

A *grant* authorizes a particular user to perform a specified action (function) on a specified object instance (or object instance set). Granting any function to a user on an object instance also gives the user the ability to query that object instance.

Note that where you are creating a data security policy for an object by creating a grant, you need to include that object in your grant definition. Other than in this specific type of case, you do not need to specify an object in your definition.

Object Function

An object function is a function that is performed on an object. An object function is associated with an object. For example, "Accept Purchase Order - PO_ACCEPT", "Decline Purchase Order - PO_DECLINE", and "Cancel Purchase Order - PO_CANCEL" are object functions associated with the Purchase Order object.

Object functions are also referred to as simply "functions".

Function sets are sometimes called "object roles".

Security Context

Security context refers to the context of the data in which the user is working. For example, data context could be the organization or responsibility with which the user is logged in.

Security Group Context (for Oracle HRMS only)

For Oracle HRMS, data can be partitioned into separate security groups, Each security group can contain unique configuration data, and multiple security groups can exist in the same installation.

For more information on security groups, see: *Configuring, Reporting and System Administration in Oracle HRMS.*

Implementation of Data Security

Implementing data security can involve two distinct tasks:

- Creating a data security policy, in which you secure access to an object
- Granting access to a set of functions (either a navigation menu or a permission set) to a user or group of users

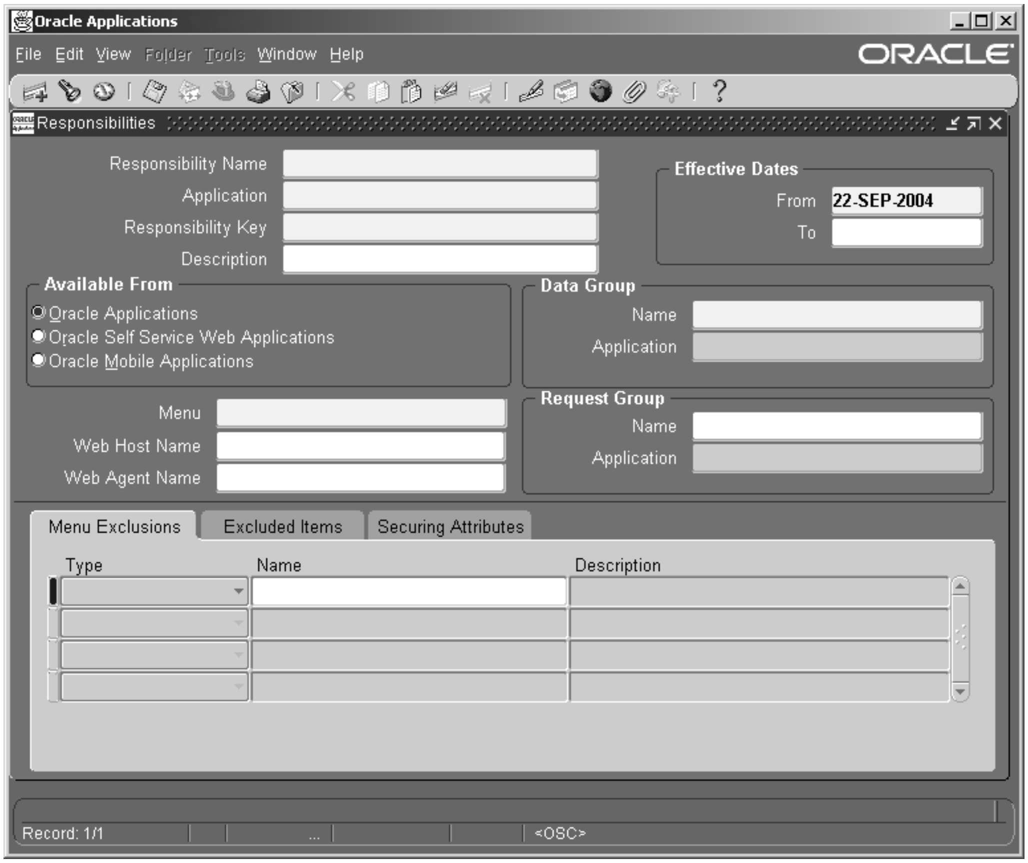
Data security policies can reflect access to:

- A specific instance (row) identified by a primary key value
- All instances (rows) of an object
- An instance set defined by a SQL predicate (WHERE clause)

Responsibilities Window

Use this window to define a responsibility. Each application user is assigned at least one responsibility.

Figure 4-2 Responsibilities Window



From 22-SEP-2004

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

Responsibilities cannot be deleted. To prevent a responsibility from being used, set the Effective Date's To field to a past date and restart Oracle Applications.

See: Overview of Function Security, page 4-7

Prerequisites

❒ Use the Data Groups window to list the ORACLE username your responsibility's concurrent programs reference on an application-by-application basis.

❒ Use the Request Groups window to define the Request Group you wish to make available with this responsibility.

❒ Use the Menus window to view the predefined Menu you could choose to assign to this responsibility.

Responsibilities Block

An application name and a responsibility name uniquely identify a responsibility.

Responsibility Name

If you have multiple responsibilities, a pop-up window includes this name after you sign on.

Application

This application name does not prevent the user of this responsibility from accessing other applications' forms and functions if you define the menu to access other applications.

Responsibility Key

This is a unique name for a responsibility that is used by loader programs. Loaders are concurrent programs used to "load" such information as messages, user profiles and user profile values into your Oracle Applications tables. To help ensure that your responsibility key is unique throughout your system, begin each Responsibility Key name with the application short name associated with this responsibility.

Effective Dates

From/To

Enter the start/end dates on which the responsibility becomes active/inactive. The default value for the start date is the current date, and if you do not enter an end date, the responsibility is valid indefinitely.

You cannot delete a responsibility because its information helps to provide an audit trail. You can deactivate a responsibility at any time by setting the end date to the current date. If you wish to reactivate the responsibility, change the end date to a date after the current date, or clear the end date.

Available From

A responsibility may be associated with only one applications system.

Data Group

Name/Application

The data group defines the pairing of application and ORACLE username.

Select the application whose ORACLE username forms connect to when you choose this responsibility. The ORACLE username determines the database tables and table privileges accessible by your responsibility. Transaction managers can only process requests from responsibilities assigned the same data group as the transaction manager.

Menu

The menu whose name you enter must already be defined with Oracle Applications. See: Menus, page 4-29.

Web Host Name

If your Web Server resides on a different machine from your database, you must designate the host name (URL) here. Otherwise, the Web Host Name defaults to the current database host server.

Web Agent Name

Enter the PL/SQL Agent Name for the database used by this responsibility. If you do not specify an Agent Name, the responsibility defaults to the agent name current at log-on.

Request Group

Name/Application

If you do not assign a request security group to this responsibility, a user with this responsibility cannot run requests, request sets, or concurrent programs from the Submit Requests window, except for request sets owned by the user. The user can access requests from a Submit Requests window you customize with a request group code through menu parameters.

See:

Overview of Oracle Applications Security, page 4-1

Customizing the Submit Requests Window Using Codes, *Oracle Applications System Administrator's Guide - Configuration*

Request Groups, *Oracle Applications System Administrator's Guide - Configuration*

Menu Exclusions Block

Define function and menu exclusion rules to restrict the application functionality accessible to a responsibility.

> **Note:** If your product uses task flows (for example, Oracle HRMS), excluding a function from a responsibility using Menu Exclusions does not exclude the function from any task flow for that responsibility. If you do not want the function to be accessible from the task flow, you must update the task flow definition accordingly.

Type

Select either Function or Menu as the type of exclusion rule to apply against this responsibility.

- When you exclude a function from a responsibility, all occurrences of that function throughout the responsibility's menu structure are excluded.
- When you exclude a menu, all of its menu entries, that is, all the functions and menus of functions that it selects, are excluded.

Name

Select the name of the function or menu you wish to exclude from this responsibility. The function or menu you specify must already be defined in Oracle Applications.

HTML-Based Applications Security

Oracle HTML-based applications use columns, rows and values in database tables to define what information users can access. Table columns represent "attributes" that can be assigned to a responsibility as Securing Attributes or Excluded Attributes. These attributes are defined in the Web Application Dictionary.

For more information, see the *Oracle Self-Service Web Applications Implementation Manual.*

Excluded Items

Use the List of Values to select valid attributes. You can assign any number of Excluded Attributes to a responsibility.

Securing Attributes

Use the List of Values to select valid attributes. You may assign any number of securing attributes to the responsibility.

Security Groups Window

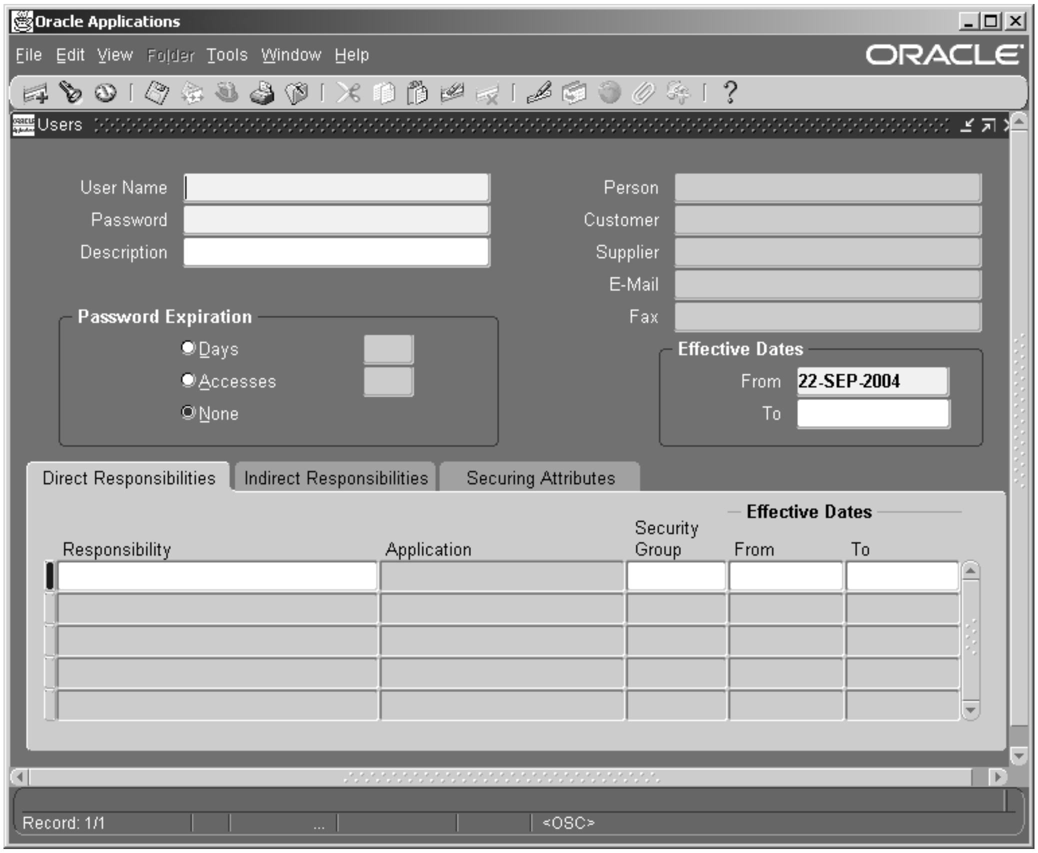
This form is for HRMS security only.

For more information on setting up system administration for the HRMS products, see: *Customizing, Reporting, and System Administration in Oracle HRMS.*

Users Window

Use this window to define an application user. An application user is an authorized user of Oracle Applications or Oracle Self-Service Applications, and is uniquely identified by an application username.

Figure 4-3 Users Window



Password Expiration: ● None

Effective Dates From: 22-SEP-2004

Record: 1/1

Once defined, a new application user can sign on to Oracle Applications and access data through Oracle Applications windows. See: Overview of Oracle Applications Security, page 4-1.

> **Note:** If you have upgraded from a previous release of Oracle Applications, ensure that you have run the Party Merge concurrent

program to update your user data. If you have not run this program, you may receive errors in querying your user data.

Users Block

User Name

An application user enters this username to sign on to Oracle Applications.

The username must:

- Be a single word, i.e. contain no spaces.
- Include only alphanumeric characters ('A' through to 'Z', and '0' through to '9').
- Only employ characters supported by the operating system's character set.

> **Tip:** We recommend that you define meaningful usernames, such as the employee's first initial followed by their last name. Or, for a group account, you can define the application username so as to indicate the purpose or nature of the group account.

Password

Enter the initial password of an application user. An application user enters this password along with his or her username to sign on to Oracle Applications.

- A password must be at least five (5) characters and can be up to thirty (30) characters.
- All characters are allowed except control characters, which are non-printable. Oracle encourages the use of non-alphanumeric characters because they add complexity, making passwords harder to guess.

This window does not display the password you enter. After you enter a password, you must re-enter it to ensure you did not make a typing error.

If the application user already exists and the two entries do not match, the original password is not changed, and you navigate automatically to the next field.

If you are defining a new application user and the two entries do not match, you are required to enter the password again. For a new user, you cannot navigate to the next field until the two entries match.

The first time an application user signs on, he must change his password. If a user forgets his password, you can reassign a new password in this field.

As System Administrator, you can set an initial password or change an existing password, but you cannot access the user's chosen password.

You can set the minimum length of Oracle Applications user passwords using the profile option Signon Password Length. If this profile option is left unset, the minimum length defaults to 5.

You can set the minimum number of days that a user must wait before being allowed to reuse a password with the Signon Password No Reuse profile option.

You can use the profile option Signon Password Hard to Guess to set rules for choosing passwords to ensure that they will be "hard to guess." A password is considered hard-to-guess if it follows these rules:

- The password contains at least one letter and at least one number.
- The password does not contain the username.

- The password does not contain repeating characters.

For information on case sensitivity in passwords, see: Case Sensitivity in Oracle Applications User Passwords, page 4-1.

Person, Customer, and Supplier

Use these fields to enter the name of an employee (person), customer, or supplier contact. Enter the last name and first name, separated by a comma, of the employee, customer, or supplier who is using this application username and password. Use the List of Values to select a valid name.

Email/Fax

Enter the email address and/or fax number for this user.

Password Expiration

Days

Enter the maximum number of days between password changes. A pop-up window prompts an application user to change his or her password after the maximum number of days you specify has elapsed.

Accesses

Enter the maximum allowed number of sign-ons to Oracle Applications allowed between password changes. A pop-up window prompts an application user to change her or his password after the maximum number of accesses you specify has elapsed.

> **Tip:** We recommend that you require application users to make regular password changes. This reduces the likelihood of unauthorized access to Oracle Applications.

Effective Dates

From/To

The user cannot sign onto Oracle Applications before the start date and after the end date. The default for the start date is the current date. If you do not enter an end date, the username is valid indefinitely.

You cannot delete an application user from Oracle Applications because this information helps to provide an audit trail. You can deactivate an Oracle Applications user at any time by setting the End Date to the current date.

If you wish to reactivate a user, change the End Date to a date after the current date, or clear the End Date field.

Direct Responsibilities

Direct responsibilities are responsibilities assigned to the user directly.

Responsibility

Select the name of a responsibility you wish to assign to this application user. A responsibility is uniquely identified by application name and responsibility name.

Security Group

This field is for HRMS security only. See: *Customizing, Reporting, and System Administration in Oracle HRMS*.

From/To

You cannot delete a responsibility because this information helps to provide an audit trail. You can deactivate a user's responsibility at any time by setting the End Date to the current date.

If you wish to reactivate the responsibility for the user, change the End Date to a date after the current date, or clear the End Date.

Indirect Responsibilities

Indirect responsibilities are used with Oracle User Management only. A user may "inherit" an indirect responsibility through membership in a group to which the responsibility has been assigned.

Securing Attributes

Securing attributes are used by Oracle HTML-based applications to allow rows (records) of data to be visible to specified users or responsibilities based on the specific data (attribute values) contained in the row.

You may assign one or more values for any of the securing attributes assigned to the user. If a securing attribute is assigned to both a responsibility and to a user, but the user does not have a value for that securing attribute, no information is returned for that attribute.

For example, to allow a user in the ADMIN responsibility to see rows containing a CUSTOMER_ID value of 1000, assign the securing attribute of CUSTOMER_ID to the ADMIN responsibility. Then give the user a security attribute CUSTOMER_ID value of 1000.

When the user logs into the Admin responsibility, the only customer data they have access to has a CUSTOMER_ID value of 1000.

Attribute

Select an attribute you want used to determine which records this user can access. You can select from any of the attributes assigned to the user's responsibility.

Value

Enter the value for the attribute you want used to determine which records this user can access.

Related Topics

Overview of Oracle Applications Security, page 4-1

Defining a Responsibility, page 4-3

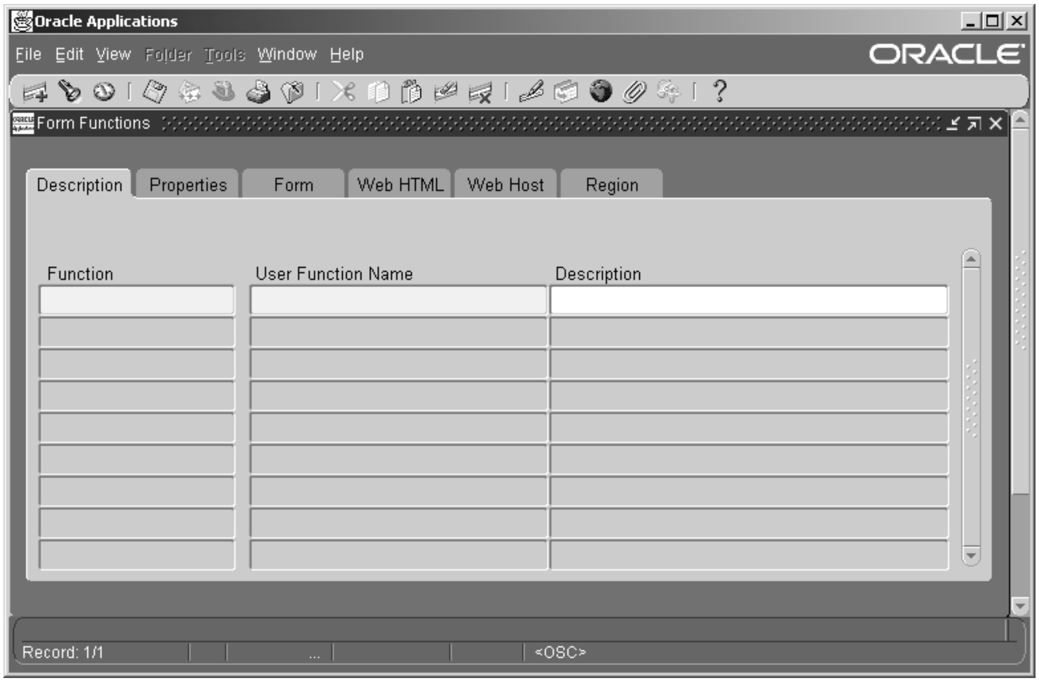
Overview of Function Security, page 4-7

Responsibilities, page 4-18

Form Functions Window

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

Figure 4-4 Form Functions Window



There are two types of functions: form functions, and non-form functions.

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

Form Functions Block

Description

Function

Users do not see this unique function name. However, you may use this name when calling your function programmatically. You should follow the naming conventions for functions.

User Function Name

Enter a unique name that describes your function. You see this name when assigning functions to menus. This name appears in the Top Ten List of the Navigator window.

Properties

Type

Type is a free-form description of the function's use (function type will be validated in a future version of this form). A function's type is passed back when a developer tests the availability of a function. The developer can write code that takes an action based on the function's type.

Standard function types include the following:

FORM

Oracle Applications form functions are registered with a type of FORM. Even if you do not register a form function with a type of FORM, Oracle Applications treats it as a form if you specify a valid Form Name/Application.

SUBFUNCTION

Subfunctions are added to menus (without prompts) to provide security functionality for forms or other functions.

JSP

Functions used for some products in the Oracle Self-Service Web Applications. These are typically JSP functions.

WWW

Functions used for some products in the Oracle Self-Service Web Applications. These are typically PL/SQL functions.

WWK

Functions used for some products in the Oracle Self-Service Web Applications. These are typically PL/SQL functions that open a new window.

WWR or WWL

Functions used for some products in the Oracle Self-Service Web Applications.

WWJ

OA Framework JSP portlet.

SERVLET

Servlet functions used for some products in the Oracle Self-Service Web Applications.

DBPORTLET

Database provider portlet.

WEBPORTLET

Web provider portlet.

Maintenance Mode Support

This field is reserved for future use only.

Context Dependence

Some functions are controlled by profile options that affect what the user can perform within the current context. Types of context dependence are:

- **Responsibility** - The function is controlled by the user's responsibility (RESP_ID/RESP_APPL_ID (includes ORG_ID)).
- **Organization** - The function is controlled by the user's organization (ORG_ID).
- **Security Group** - The function is controlled by the user's security group (service bureau mode).
- **None** - There is no dependence on the user's session context.

Form

Form /Application

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

Parameters

Enter the parameters you wish to pass to your function. Separate parameters with a space.

For a form function, if you specify the parameter QUERY_ONLY=YES, the form opens in query-only mode. Oracle Application Object Library removes this parameter from the list of form parameters before opening the form in query-only mode.

You can also specify a different form name to use when searching for help for a form in the appropriate help file. The syntax to use is:

HELP_TARGET = "*alternative_form_name*"

Your form name overrides the name of the form. See: Help Targets in Oracle Applications, *Oracle Applications System Administrator's Guide - Configuration*.

TITLE="*appl_short_name*:*message_name*"

where *appl_shortname*:*message_name* is the name of a Message Dictionary message. See: Customizing the Submit Requests Window using Codes, *Oracle Applications System Administrator's Guide - Configuration*.

> **Warning:** In general, system administrators should not modify parameters passed to predefined functions for Oracle Applications products. The few exceptions are documented in the relevant manuals or product notes.

Web HTML

The fields in the Web HTML and Web Host are only required if your function will be accessed from Oracle Applications Framework. You do not need to enter any of these fields for functions based on Oracle Forms Developer forms.

HTML Call

The last section of your function URL is the HTML Call. The HTML Call is used to activate your function. The function may be either a static web page or a procedure.

For functions used with Mobile Application Server, enter the full name of your java class file, including <package name>.<class name>. The class name and package name are case sensitive. Mobile Application Server will try to load this class from the classpath as it is. For example, 'oracle.apps.mwa.demo.hello.HelloWorld'.

Web Host

The fields in the Web HTML and Web Host are only required if your function will be accessed from Oracle Applications Framework. You do not need to enter any of these fields for functions based on Oracle Forms Developer forms.

Host Name

The URL (universal resource locator) or address required for your function consists of three sections: the Host Name, Agent Name, and the HTML Call. The Host name is the IP address or alias of the machine where the Web server is running.

Agent Name

The second section of your function URL is the Oracle Web Agent. The Oracle Web Agent determines which database is used when running your function. Defaults to the last agent used.

Icon

Enter the name of the icon used for this function.

Secured

Secured is only required when your function is accessed by Oracle Workflow. Checking Secured enables recipients of a workflow email notification to respond using email.

Encrypt Parameters

Checking Encrypt Parameters adds a layer of security to your function to ensure that a user cannot access your function by altering the URL in their browser window. You must define Encryption Parameters when you define your function to take advantage of this feature.

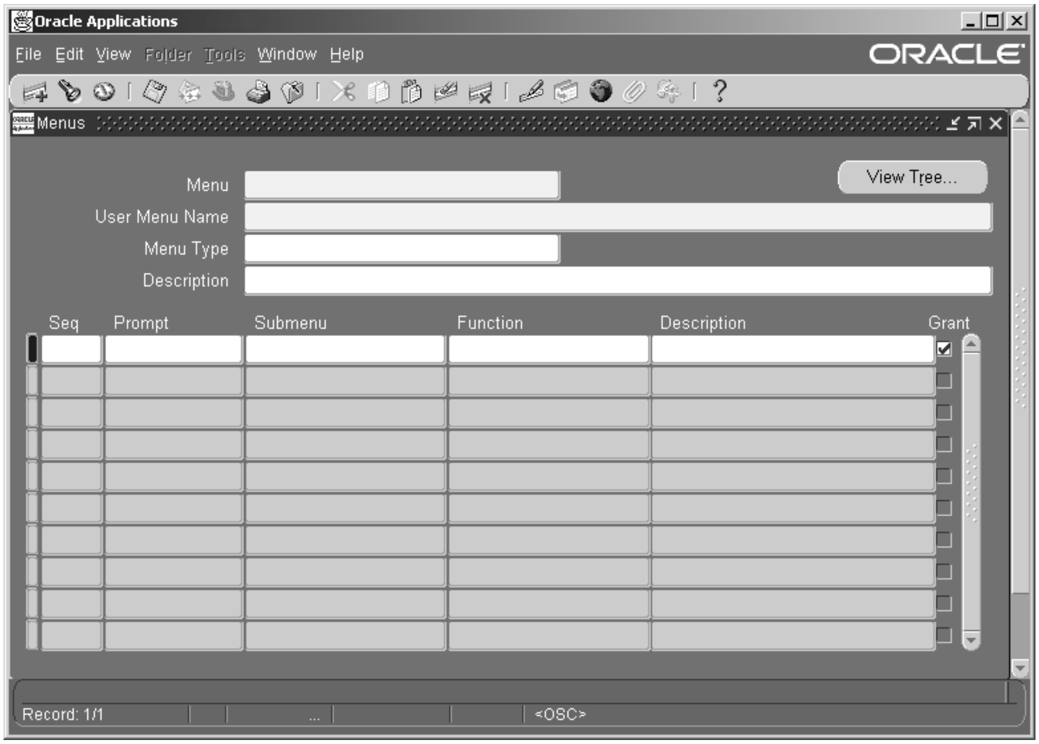
Region

The fields on this page are for future use.

Menus Window

Used to define a new menu or modify an existing menu.

Figure 4-5 Menus Window



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

A "full access" responsibility with a menu that includes all the functions in an application is predefined for each Oracle Applications product. As a System Administrator, you can restrict the functionality a responsibility provides by defining rules to exclude specific functions or menus of functions. In fact, we recommend that you use exclusion rules to customize a responsibility in preference to constructing a new menu hierarchy for that responsibility.

If you cannot create the responsibility you need by applying exclusion rules, you may build a custom menu for that responsibility using predefined forms (i.e. form functions) and their associated menus of subfunctions. However, we recommend that you do not disassociate a form from its developer-defined menus of subfunctions.

After you save your changes in this form, a request is submitted to compile the menu data.

See:

Overview of Function Security, page 4-7

Implementing Function Security, page 4-10

Before you define your menu, perform the following:

- Register your application with Oracle Application Object Library using the Applications window.
- Register any forms you wish to access from your menu with Oracle Application Object Library using the Forms window.
- Define any menus that you intend to call from your menu. Define the lowest-level submenus first. A submenu must be defined before it can be called by another menu.

> **Tip:** By calling submenus from your menu, you can group related windows together under a single heading on your menu. You can reuse your menu on other menus.

Menus Block

Menu entries detail the options available from your menu.

Menu

Choose a name that describes the purpose of the menu. Users do not see this menu name.

> **Note:** Once the menu is saved, this menu name cannot be updated.

View Tree...

Once you have defined a menu, you can see its hierarchical structure using the "View Tree..." button. See: Menu Viewer, page 4-31.

User Menu Name

You use the user menu name when a responsibility calls a menu or when one menu calls another.

Menu Type

Optionally specify a menu type to describe the purpose of your menu.

- Standard - for menus that would be used in the Navigator form
- Tab - for menus used in self service applications tabs
- Security - for menus that are used to aggregate functions for data security or specific function security purposes, but would not be used in the Navigator form

Menu Entries Block

Sequence

Enter a sequence number to specify where a menu entry appears relative to other menu entries in a menu. The default value for this field is the next whole sequence number.

> **Important:** You can only use integers as sequence numbers.

A menu entry with a lower sequence number appears before a menu entry with a higher sequence number.

> **Important:** If you change sequence numbers, or frequently insert and delete menu entries, carefully check the default value. This value may be a duplicate sequence number or an out of sequence number.

You cannot replace a menu entry sequence number with another sequence number that already exists. If you want to add menu entries to a menu entry sequence, carefully renumber your menu entries to a sequence range well outside the sequence range you want, ensuring that you do not use existing sequence numbers.Once you save this work, you can go back and renumber each entry to have the final sequence number you want.

Navigator Prompt

Enter a user-friendly, intuitive prompt your menu displays for this menu entry. You see this menu prompt in the hierarchy list of the Navigator window.

> **Tip:** Enter menu prompts that have unique first letters so that power users can type the first letter of the menu prompt to choose a menu entry.

User Exit

Invoke a user exit. A user exit is a subroutine. Examples of user exits are SQL*Forms user exits and custom user exits.

Submenu

Call another menu and allow your user to select menu entries from that menu.

Function

Call a function you wish to include in the menu. A form function (form) appears in the Navigate window and allows access to that form. Other non-form functions (subfunctions) allow access to a particular subset of form functionality from this menu.

Description

Descriptions appear in a field at the top of the Navigate window when a menu entry is highlighted.

Grant

The Grant check box should usually be checked. Checking this box indicates that this menu entry is automatically enabled for the user. If this is not checked then the menu entry must be enabled using additional data security rules.

Menu Viewer

The Menu Viewer is a read-only window that provides a hierarchical view of the submenus and functions of a menu, and also lists properties of the menus and functions.

You can launch the viewer from the Menus form by clicking on the "View Tree..." button. The viewer will appear for the menu specified in the Menus form.

> **Note:** When you are creating or editing a new menu, your changes must be committed to the database before you will be able to see them in the Menu Viewer.

Functionality

Menu Tree

To view the menu tree, click on the plus (+) sign next to the menu. You will see a hierarchical tree with a number of nodes. Each node represents a function or submenu of your main menu.

> **Note:** The menu tree displays the user menu name for the main menu, and displays the prompts from the Menus form for submenus and functions. If no prompt has been specified, then no label will appear for the node.

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

Node Properties

To view properties of a particular menu or function, highlight the node in the menu tree. The node properties will appear in the Properties pane. You can create a separate Properties page for a node by clicking the "push pin" button at the top of the Properties pane.

The entry's sequence number, prompt, and description are shown.

View Options

The View menu provides options on how the viewer displays your menu.

You can specify whether the Node Properties pane, the toolbar, or the status bar are displayed. You can also choose the display style in which you view your menu tree.

Display Styles

There are three styles for viewing your menu tree. You can select one from the View menu or from the buttons on the toolbar.

Vertical
Menu entries are displayed vertically, similar to how they appear in the Navigator window when you log on to Oracle Applications.

Interleaved
Menu entries are displayed horizontally and vertically.

Org-Chart
Menu entries are displayed horizontally as in an organizational chart.

Edit Menu

From the Edit menu you can bring up a Properties window for the node you have highlighted in the menu tree.

> **Note:** You can view the properties for your menu or function here, but you cannot edit them.

You can view and edit your Preferences for the Menu Viewer. You can choose colors for your menu tree pane as well as the text font and size.

Objects

Use these pages to find, create, and edit data objects. You define objects to be secured in the Data Security system.

In these pages, objects are described with the following

- The **Name** is the name that appears in the Object Instance Set and Grants pages. This name should be user-friendly.
- The **Code** is the internal name of the object.
- The **Application Name** is the owning application.
- The **Database Object Name** is the name of the underlying database object, usually a table.

Related Topics

Overview of Data Security, page 4-15

Find Objects

Use this page to find an existing object.

Simple Search

Name

The display name of the object.

Code

The object name.

Application Name

The object's owning application.

Database Object Name

The database object name.

Advanced Search

Use the Advanced Search screen to find data that meet a set of criteria. With the Advanced Search screen, you can enter in special conditions based on the given fields, and the search results will consist of all data that match the conditions.

For example, for a specified application, you can search for all objects whose name begins with a letter before "P". (Note: all uppercase letters precede all lowercase letters for this type of search).

Search Results

The search results are shown in a table with the following columns:

- Name - click on the object name to view details on the object.
- Code
- Application Name
- Database Object
- Description
- Last Update

To update an object, click on the icon under the Update column.

Update Object

Use this page to update the fields listed below for an object. You cannot change the internal Object Name of an existing object.

Display Name

Enter a user-friendly name for the object.

Application Name

The owning application for the object. This application owns the database table on which the object is based.

Database Object Name

Typically this is a table in the database.

Description

Enter a description for the object.

Create Object

Use this page to create a new object. Enter the following information:

Name

Enter a user-friendly name for the object.

Code

Enter a code that will be used as an internal name for the object. This name cannot include spaces and can include underscores and hyphens. You cannot update the object name after the object is created and saved.

Application Name

The owning application for the object. This application owns the database table on which the object is based.

Database Object

Typically this is a table in the database.

Description

Enter a description for the object.

Object Column Details

Enter in information on the primary key for the object (*n* below indicates an integer between 1 and 5). The primary key is used to identify rows (object instances) for inclusion in object instance sets.

PK*n* Column Name

The primary key column name.

PK*n* Column Type

The datatype for the column.

Object Detail

This page provides the following information for an object:

- Object Name
- Display Name
- Application
- Database Object Name
- Description

Columns

You can also view details on columns that comprise the primary key (*n* below indicates an integer between 1 and 5):

- PK*n* Column Name
- PK*n* Column Type

Instances of an object can be grouped together into an object instance set. For example, you may want to create a group of projects or a group of items. To create and manage objects instance sets, click on the "Manage Object Instance Sets" button.

Click on the "Return to Object Search" link to go back to the main Objects page.

Delete Object

Confirm the deletion of an object from this page. Review the information shown, and click the "Delete" button.

Related Topics

Object Details, page 4-35

Object Instance Sets

After you create an object you can create a set of instances of the object. For example, you could define the object "User" corresponding to the User table. Each row in the User table becomes an instance of the User object. Users in the sales organization could then be grouped into an Object Instance Set named "Sales Organization".

Object Instance Sets are described by the following:

- The **Object Instance Set Name** is its internal name. This name must not contain any spaces and can include underscores.
- The **Display Name** is a user-friendly name for the object that appears in the Grants pages.
- The **Predicate** is the WHERE clause used to define the object instances in the set.

Manage Object Instance Set

Use this page to manage existing object instance sets or create new ones.

The following object information is displayed:

- Object Name
- Display Name
- Application
- Database Object Name
- Description

Existing Object Instance Sets

- Instance Set Name - click on the Instance Set Name to view details
- Display Name
- Description

To update an object, click on the icon under the Details column to open up the Update Object page.

To delete a row, click on the icon under the Delete icon, or select the object and click the Delete button.

To return to the main Objects page, click on the "Return to Object Search" link.

Related Topics

Objects, page 4-33

Create Object Instance Set

The containing object's Name, Display Name, Application ID, Database Object Name, and Description are shown.

Enter the following for the Object Instance Set:

Code

Enter a name that will be used internally for the object instance set. This name cannot include spaces and can include underscores and hyphens. The Object Instance Set Name cannot be updated once the object instance set has been created and saved.

Name

Enter a user-friendly, descriptive name to appear in the Grants pages.

Description

Enter a description for the object instance set.

Predicate

This predicate determines which object instances are included in the set. Do not include "WHERE" in your entry, but only the body of the WHERE clause.

Update Object Instance Set

The containing object's Name, Display Name, Application ID, Database Object Name, and Description are shown.

> **Note:** The Object Instance Set Name cannot be updated after the object instance set has been created and saved.

Display Name

Enter a user-friendly, descriptive name to appear in the Grants pages.

Description

Enter a description for the object instance set.

Predicate

This predicate determines which object instances are included in the set. Do not include "WHERE" in your entry, but only the body of the WHERE clause.

Delete Object Instance Set

Confirm the deletion of an object from this page. Review the information shown, and click the "Delete" button.

Related Topics

Object Instance Set Details, page 4-37

Object Instance Set Details

Details of an object instance set are shown on this page.

The containing object's Name, Display Name, Application ID, Database Object Name, and Description are shown.

The following is shown for the object instance set:

- Code

- Name
- Description
- Predicate

Use the "Return to Manage Object Instance Sets" to return to the main page.

Related Topics

Object Instance Sets, page 4-36

Grants

Search Grants

Use this page to search for grants.

You can search using the following criteria:

- Name
- Grantee Type - Select from one of the following:
  - All Users - The grant applies to all users.
  - Group of Users - The grant applies to a group of users.
  - Specific User - The grant applies to a single user.

    If you select Group of Users or Specific User, you will be prompted to specify the group or the user.
- Set - The Navigation Menu or Permission Set included in the grant.
- Object Type - A grant can apply to either all objects or only a specific object. Under Object Type, specify if your search should include only grants that apply to all objects ("All Objects"), only grants that apply to a specific object ("Specific Object"), or both ("Any").

  If you select Specific Object, you will be prompted to specify the object.
- Effective Dates

Create Grant

Use these pages to create a grant. Grants are used to manage user access to product functionality. In these pages you give access to functions to specified users.

Related Topics

Overview of Data Security, page 4-15

Define Grant

In this page you specify basic information for the grant.

To define a grant:

1. Enter a name and description for your grant.
2. Enter effective dates for your grant.

3. Enter the security context information.

   The security context defines who will receive the grant.

   For Grantee, select one of the following:

4. All Users

5. Group of Users - The grant will apply to a group of users you specify.

6. Single User - The grant will apply to a user you specify.

   If you choose Group of Users or Single user, you are prompted for Grantee.

   For Operating Unit, specify an operating unit if you want your grant to apply to a specific one.

   For Responsibility, specify a responsibility if you want your grant to apply to a specific one.

7. Enter the Data Security information if you are creating a data security policy for an object. The grant applies to the object you specify.

   If you are not creating a data security policy, you will skip the next step.

   > **Note:** You cannot change a data security policy once it has been saved. You can delete it or provide an end date to a data security policy.

Select Object Data Context

If you specified that your grant applies to a single object, you add context for that object in this page.

Choose one of the following:

- All Rows
- Instance - A specific instance (row) of the object
- Instance Set - A set of instances (rows) of the object

Define Object Parameters and Select Set

If you selected either an object instance or an instance set earlier, you can further customize the resulting set by additional information for the data context.

Additionally, you can select either a permission set or a navigation menu that can additionally specify how the grant will be applied in the security context.

For an instance set:

1. In the Predicate region, the predicate that defines the instance set is shown. In the Instance Set Details region, specify the values for the parameters to be used in the predicate above.
2. Select the permission set or navigation menu set that defines the grantee's access.

For an instance:

1. In the Instance Details region, specify information identifying the instance.
2. Select the permission set or navigation menu set that defines the grantee's access.

Review and Finish

Use this page to review the definition of your grant. Click **Finish** to save your work.

Update Grant

Use this page to update the definition of your grant.

View Grant

Use this page to view details for a grant, including:

- Security Context
- Object information, if applicable
- Set information

You can update or delete a grant from this page.

Functions

Use these pages to define new functions. A function is a part of an application's functionality that is registered under a unique name for the purpose of assigning it to, or excluding it from, a responsibility.

You can search for functions from the main page.

Function Types

When you define a function, you assign it one of the following types:

- Form - an Oracle Forms form function
- Mobile Application - a function used in an Oracle mobile application
- Database Provider Portlet
- JSP Interoperable with OA
- Generic Plug
- Plug
- Process
- SSWA JSP function
- SSWA PL/SQL function
- SSWA PL/SQL function that opens a new window (kiosk mode)
- SSWA servlet function
- Web Provider portlet

Related Topics

Form Functions Window, page 4-25

Search

Using Simple Search, You can search for functions using the following criteria:

- Name
- Code
- Type

Advanced Search

Using Advanced Search, you can be more flexible with your criteria, as well as search on the description field.

Create Function

Use these pages to create a function.

1. Specify a name for the function.
2. Specify a code for the function. The code is the internal name for the function. Once the function has been saved, the code cannot be updated.
3. Specify a type for the function.
4. For context dependence, specify ’None’ or Responsibility.
5. If you are defining a form function, select the name and application of your form. If the function applies to a specific object, select the object name and specify parameters.

   Note: Maintenance Mode Support is reserved for future use only.

Update Function

Use this page to update an existing function. Note that you cannot update the code for an existing function.

To update a function:

1. Specify a name for the function.
2. If this function applies to a specific object, specify the object.
3. Specify a type for the function.
4. For context dependence, specify ’None’ or Responsibility.

To update function details:

1. If this is a form function, select the name and application of your form.
2. If the function applies to a specific object, you can update the object name and specify parameters.

In updating menus,

- You can remove the function from menus containing it using the Menus subtab.
- You can also update menu prompts and descriptions for the function here.

  Note: Maintenance Mode Support is reserved for future use only.

Duplicate Function

Use this page to duplicate an existing function.

Note that you must enter a unique code for the new function you are creating.

To duplicate a function:

1. Specify a name for the function.
2. Specify a code for the function. The code is the internal name for the function. Once the function has been saved, the code cannot be updated.
3. Specify a type for the function.
4. Specify the level of maintenance mode support for the function.
5. For context dependence, specify ’None’ or Responsibility.
6. If you are defining a form function, select the name and application of your form. If the function applies to a specific object, select the object name and specify parameters.

View Function

Use this page to view details on an existing function.

You can update and duplicate a function from this page. If the function is not on a menu, you can also delete the function.

Delete Function

Use this page to delete a function.

Navigation Menus

Define a new menu or modify an existing menu.

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

A "full access" responsibility with a menu that includes all the functions in an application is predefined for each Oracle Applications product. As a System Administrator, you can restrict the functionality a responsibility provides by defining rules to exclude specific functions or menus of functions. In fact, we recommend that you use exclusion rules to customize a responsibility in preference to constructing a new menu hierarchy for that responsibility.

If you cannot create the responsibility you need by applying exclusion rules, you may build a custom menu for that responsibility using predefined forms (i.e., form functions). However, we recommend that you do not disassociate a form from its developer-defined menus.

Before creating a menu, perform the following:

- Register your application with Oracle Application Object Library using the Forms-based Applications window.
- Define any menus that you intend to call from your menu. Define the lowest-level submenus first. A submenu must be defined before it can be called by another menu.

  > **Tip:** By calling submenus from your menu, you can group related windows together under a single heading on your menu. You can reuse your menu on other menus.

Terms

Terms used in defining menus include:

- Name - The display name for the menu
- Code - The internal name for the menu
- Type - The purpose of the menu
  - Permission Set - For menus that are used to aggregate functions for data security or specific function security purposes, but would not be used in the Navigator form.
  - Standard - For menus used in the Navigator form
  - App Pref Menu Container - For preferences
  - Global Menu - For providing access to tasks and content that are applicable to the entire application
  - HTML Side Navigator Menu
  - HTML SideBar
  - HTML SideList
  - HTML Sub Tab - A tab-like control for switching content or action views in the page's content area. Sub tabs can be used with a horizontal navigation element, with a tab and horizontal navigation elements, or with a side navigation
  - HTML Tab
  - Homepage

If you are creating a menu to be used with Oracle Applications Framework, additional information can be found in the Oracle Applications Framework documentation on Oracle*MetaLink*. See: "Oracle Applications Framework Release 11i Documentation Road Map", Oracle*Metalink* Note 275880.1.

Search for Menus

Enter any of the following criteria for the menu:

- Name
- Code
- Type

Create Navigation Menu

Use this page to create a navigation menu.

1. Choose a user-friendly name that describes the purpose of the menu.
2. Enter a code for the menu. Choose an internal name that indicates the purpose of the menu. Users do not see this menu code.
3. Optionally specify a menu type and description to describe the purpose of your menu.

Add your information for your menu entries using the Menu Builder.

1. Enter a prompt for your menu entry.

Enter a user-friendly, intuitive prompt your menu displays for this menu entry. You see this menu prompt in the hierarchy list of the Forms Navigator window.

> **Tip:** Enter menu prompts that have unique first letters so that power users can type the first letter of the menu prompt to choose a menu entry.

2. If this menu entry is a menu itself (a submenu), enter in the menu name.

   You can call another menu and allow your user to select menu entries from that menu.

3. If this menu entry is a function, enter in the function name.

   Call a function you wish to include in the menu.

4. Specify the function type.

5. Apply your changes.

If you want to reorder the menu entries, click the **Reorder** button.

Menu Manager

Once you have your menu defined, you can update its list of entries in the Menu Manager tab.

Hierarchy of Children

The Hierarchy of Children subtab provides information on the child nodes within the menu structure. Child nodes are either functions or menus (submenus). Child nodes are displayed in a hierarchy with the following information, as applicable: display name, internal menu name, function name, type, and description.

Direct Parents

The Direct Parents subtab allows the user to see the direct parent(s), if any, of the navigation menu. A direct parent is a menu that contains this menu directly as a submenu. This feature is useful in identifying the direct impact of any changes that may be made to this menu.

For each parent, the prompt and internal menu name is shown.

Grants

The Grants subtab displays the associated grants that secure the navigation menu.

For each associated grant the following is shown: name, grantee type, grantee, valid dates, data context type, object, and instance set.

Update Menu

Use this page to update an existing navigation menu.

All fields can be updated except for the menu code.

The direct parents of a menu can be deleted in the Direct Parents tab.

You cannot update a parent menu from this tab. You must navigate to the parent menu record itself to update it.

> **Note:** You cannot replace an existing parent menu with another menu, as the parent menu is used as the primary key of the hierarchy mapping. Instead, you have to delete this existing (child) menu and add a new menu. Also, the sequence number cannot be updated since it is the primary key. You can update the prompt and description.

Duplicate Menu

Use this page to duplicate a menu and copy its hierarchy of children. You must give the duplicate menu and new code (internal name).

View Menu

Use this page to view details of a menu.

Delete Menu

Use this page to delete a menu.

Note that you cannot delete a referenced menu. A menu can be referenced by any of the following:

- Children (menu or function)
- Menu parents
- Grants

Permissions

A permission is the smallest unit of securable action that can be performed on the system. A permission can either be abstract permissions or executable functions (menu). It can either be a system level permission or be sensitive to a data context. For example, a particular JSP page may be an executable permission and "View Person" may be an abstract permission.

You can search for permissions from the main page. You can update, duplicate, or remove a permission found in your search results. You can also create a new permission from this page.

Search for permissions using the following criteria:

- Name
- Code
- Object Name

Create Permission

Use these pages to create a permission.

1. Specify a name for the permission.
2. Specify a code for the permission. The code is the internal name for the permission. Once the permission has been saved, the code cannot be updated.
3. If this permission applies to a specific object, specify the object.

4. If you want to add this permission to a permission set now, select a permission set.

Update Permission

Use this page to update an existing permission.

Note that you cannot update the code (internal name) for the permission.

1. You can specify a new name for the permission.
2. You can specify a new object if the permission applies to a specific object.

You can update the permission set information as well:

1. To add this permission to a permission set, select a permission set from the list of values for "Add this to a Permission Set".
2. To delete this permission from a permission set, select the permission set in the table and click the **Remove** button.

Select the **Apply** button to save your changes.

Duplicate Permission

Use this page to duplicate an existing permission.

Note that you must enter a unique code for the new permission you are creating.

1. Specify a name for the permission.
2. Specify a code for the permission. The code is the internal name for the permission. Once the permission has been saved, the code cannot be updated.
3. If this permission applies to a specific object, specify the object.
4. If you want to add this permission to a permission set now, select a permission set.

View Permission

Use this page to view details on an existing permission.

You can update or duplicate a permission from this page. You can delete a permission from this page if it does not belong to a permission set.

Delete Permission

Use this page to delete a permission.

Permission Sets

Permission sets provide a way to group related permissions together. You can create a new permission set from this page.

You can search for permission sets using the following criteria:

- Name
- Code

You can update, duplicate, or delete permission sets found in your search.

Create Permission Set

Use this page to create a permission set.

1. Specify a name for the permission set.
2. Specify a code for the permission set. The code is the internal name for the permission set. Once the permission set has been saved, the code cannot be updated.

Use the **Permission Set Builder** to add permissions to your new permission set. You can also add existing permission sets to the new permission set.

Update Permission Set

Use this page to update an existing permission set.

You can specify a new name for the permission set. Note that you cannot update the code (internal name) for the permission set.

If you want to update which permissions and permission sets belong to this permission set, use the **Permission Set Builder** to do so.

Permission Set Manager

Once you have your permission set defined, you can update the contents of the permission set in the Permission Set Manager tab.

Hierarchy of Children

The Hierarchy of Children subtab provides information on the child nodes in the permission set structure. A child node is either a permission or permission set. Child nodes are displayed in a hierarchy with the following information: display name, permission set name (if applicable), permission name (if applicable), and description.

Direct Parents

The Direct Parents subtab allows you to see the permission sets, if any, that include the current permission set. This feature is useful in identifying the direct impact of any changes that may be made to this permission set.

Grants

The Grants subtab displays the associated grants that secure the navigation menu.

For each associated grant, the name, grantee type, grantee, valid dates, data context type, object name, and instance set name is displayed.

Duplicate Permission Set

Use this page to duplicate an existing permission set.

Note that you must enter a unique code for the new permission set you are creating.

1. Specify a name for the permission set.
2. Specify a code for the permission set. The code is the internal name for the permission set. Once the permission set has been saved, the code cannot be updated.

If you want to update which permissions and permission sets belong to this permission set, use the **Permission Set Builder** to do so.

View Permission Set

Use this page to view details on an existing permission set.

Click **Update** to update the permission set.

Delete Permission Set

Use this page to delete a permission set. If a permission set is a child of another permission set, it cannot be deleted without first being removed from its parent permission set.

Compile Security Concurrent Program

Use this concurrent program to compile your menu data. Compiling your menu data allows for the system to determine more quickly whether a function is available to a particular responsibility/menu.

A request to run this program is automatically submitted when you make changes using the Menus form.

Parameter

Everything

This parameter takes the value Yes or No. "No" is used to recompile only those entities that are marked as needing recompilation. "Yes" is used to recompile all entities, and can take a long time. "No" is the default value.

Function Security Reports

Use the function security reports to document the structure of your 11i menus. You can use these reports as hardcopy to document your customized menu structures before upgrading your Oracle Applications software.

The function security reports consist of the Function Security Functions Report, the Function Security Menu Report, and the Function Security Navigator Report.

These reports are available through the Function Security Menu Reports request set. For each report, specify the responsibility whose function security you want to review.

Function Security Function Report

Specify a responsibility when submitting the report. The report output lists the functions accessible by the specified responsibility.

The report does not include items excluded by function security rules.

Function Security Menu Report

Specify a responsibility when submitting the report. The report output lists the complete menu of the responsibility, including all submenus and functions.

The report indicates any excluded menu items with the rule that excluded it.

Function Security Navigator Report

Specify a responsibility when submitting the report. The report output lists the menu as it appears in the navigator for the responsibility specified.

This report does not include items excluded by function security rules, or non-form functions that do not appear in the navigator.

Users of a Responsibility Report

This report documents who is using a given responsibility. Use this report when defining or editing application users.

Report Parameters

Application Name

Choose the name of the application to which the responsibility you want in your report belongs.

Responsibility Name

Choose the name of the responsibility you want in your report.

Report Heading

The report heading indicates the application name and responsibility for which you requested a report.

Column Headings

User Name

The name of the user who is assigned to the responsibility.

Start Date

The date the responsibility became active for the user.

End Date

The date the responsibility either becomes inactive or became inactive for the user. If no end date appears for a user, then this responsibility is always enabled for the user.

Description

The description of the user who is assigned to the responsibility.

Related Topics

Overview of Oracle Applications Security, page 4-1

Defining a Responsibility, page 4-3

Overview of Function Security, page 4-7

Responsibilities field help, page 4-18

Users field help, page 4-22

Active Responsibilities Report

This report shows all the responsibilities that are currently active, the users who can currently access each responsibility, and the start and end dates when they can access the responsibility.

Report Parameters

None.

Report Heading

This displays the name of the report, the date and time the report was run, and the page number.

Column Headings

Application Name

The name of the application associated with the responsibility.

Responsibility Name

The name of the currently active responsibility.

User Name

The name of the user who can currently access the responsibility.

Start Date

The date when the user can begin accessing the responsibility.

End Date

The date when the user can no longer access the responsibility. See: Overview of Oracle Applications Security, page 4-1.

Related Topics

Overview of Oracle Applications Security, page 4-1

Defining a Responsibility, page 4-3

Responsibilities field help, page 4-18

Users field help, page 4-22

Active Users Report

This report shows all the usernames that are both currently active and have at least one active responsibility. It also displays all the responsibilities that users can access, and the start and end dates when they can access each responsibility.

Report Parameters

None.

Report Heading

The report heading displays the name of the report, the date that the report was run, and the page number.

Column Headings

User Name

The Oracle Applications name of the currently active user. The start and end dates that you specify in the Users window determine whether a username is currently active.

Application Name

The name of the application associated with the responsibility.

Responsibility Name

The name of the currently active responsibility.

Start Date

The date when the user can begin accessing the responsibility. You can specify a start date when you assign the responsibility to the user in the Responsibilities block of the Users window.

End Date

The date when the user can no longer access the responsibility. You specify an end date when you assign the responsibility to the user in Responsibilities block of the Users window.

Reports and Sets by Responsibility Report

This report identifies which reports (and other concurrent programs) and report sets are included in the request security groups available to any given responsibility. Use this report when defining or editing responsibilities.

Report Parameters

If you enter no parameters, the report documents all reports and report sets accessible from each responsibility.

Application Short Name

Choose the application name associated with the responsibility whose available reports and report sets you wish to report on.

If you do not choose an application name, the report documents all reports and report sets accessible from each responsibility.

Responsibility Name

Choose the name of a responsibility whose available reports and report sets you wish to report on. You must enter a value for Application Short Name before entering a value for Responsibility Name.

Report Headings

The report headings list the report parameters you specify, and provide you with general information about the contents of the report.

Related Topics

Overview of Oracle Applications Security, page 4-1

Defining a Request Security Group, page 4-4

Responsibilities field help, page 4-18

5

User and Data Auditing

Overview of User and Data Auditing

There are two types of auditing in Oracle Applications: auditing users, and auditing database row changes.

Auditing User Activity

Auditing users is supported by:

- Sign-On:Audit Level profile option setting
- Audit Reports

Based on the audit level you choose, Sign-On audit records usernames, dates, and times of users accessing the system, as well as what responsibilities, forms, and terminals users are using.

Auditing Database Row Changes

Auditing database row changes is supported by:

- From the **Help** menu, **About This Record** ...
- AuditTrail:Activate profile option setting
- Audit forms - see below.

Related Topics

Auditing User Activity, page 5-2

Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Monitor Users, page 5-19

Reporting on AuditTrail Data, page 5-6

Setting Up AuditTrail, page 5-7

AuditTrail Tables, Triggers and Views, page 5-7

Reporting on Audit Information, page 5-13

Disabling AuditTrail and Archiving Audit Data, page 5-14

Audit Installations, page 5-20

Audit Groups, page 5-22

Audit Tables, page 5-24

Auditing User Activity

Oracle Applications provides a Sign-On Audit feature that allows you to:

- Track what your users are doing and when they do it.
- Choose who to audit and what type of information to audit.
- View quickly online what your users are doing.
- Check the security of your application.

With Sign-On Audit, you can record usernames, terminals, and the dates and times your users access Oracle Applications. Sign-On Audit can also track the responsibilities and forms your users use, as well as the concurrent processes they run.

Major Features

Selective Auditing

Sign-On Audit lets you choose who to audit and what type of user information to track. You can selectively determine what audit information you need, to match your organization's needs.

Monitor Application Users

The Monitor Users form gives you online, real-time information about who is using Oracle Applications and what they are doing.

You can see what users are signed on (application username and operating system login name), what responsibilities, forms, and terminals they are using, how long they have been working on forms, and what Oracle database processes they are using.

Sign-On Audit Reports

Sign-On Audit Reports give you historical, detailed information on what your users do in your application.

You can give search criteria to narrow your search for information. You can also sort your Sign-On Audit information to create easy-to-read reports.

Setting Up Sign-On Audit

You use the Sign-On:Audit Level user profile option to control who Sign-On Audit tracks and the level at which they are audited.

Use the *Monitor Users* form to view online what your users are doing.

Use the *Submit Reports* form to submit Sign-On Audit Reports that give you detailed audit information.

Enabling Sign-On Audit

Use the System Profile Values form to enable Sign-On Audit. Choose the scope of your audit and who to audit by setting the user profile level at the user, responsibility, application, or site profile levels.

> **Note:** Users cannot see or change this profile option.

After you set or change audit levels, the new audit levels for a user take effect the next time the user signs onto Oracle Applications from the operating system.

Selecting Audit Levels

The Sign-On:Audit Level profile option allows you to select a level at which to audit users who sign on to Oracle Applications.

Four audit levels provide increasing levels of monitoring: None, User, Responsibility, and Form.

Auditing level None is the default, and tracks:

- No activities by any users who sign on to Oracle Applications

Auditing at the User level tracks:

- Who signs on to your system
- The times users log on and off
- The terminals in use

Auditing at the Responsibility level performs the User level audit functions and also tracks:

- The responsibilities users choose
- How much time users spend using each responsibility

Auditing at the Form level performs the Responsibility and User level audit functions, and also tracks:

- The forms users choose
- How long users spend using each form

Auditing Levels and System Overhead

In planning your organization's Sign-On Audit implementation, you should consider the additional system overhead required to monitor and audit your users as they access Oracle Applications. The more users you audit, and the higher the level of auditing, the greater the system overhead such as processing costs and disk space. You should balance your organization's auditing needs with the resources available, obtaining addititional resources if the existing ones are insufficient to support the required auditing activities as well as the actual workload.

Example - Audit Users, Responsibilities, & Forms

An example implementation of Sign-On Audit would be to audit all of your users' sign-ons, the responsibilities they select, and the forms they access.

To accomplish this, you would set Sign-On:Audit Level to:

- Form audit

- At the Site profile level

Example - Audit a specific responsibility, excepting one user

Another example of using Sign-On Audit is for an organization to audit all users of the Personnel Manager responsibility, except for MJONES.

In this example, you do not need to audit the forms the users access or the responsibilities they select.

To set up this implementation, set Sign-On:Audit Level to:

- User audit
- At the responsibility profile level for the Personnel Manager responsibility

You also set Sign-On:Audit Level to:

- None
- At the user profile level for the application user MJONES

Using the Application Monitor

Use the Monitor Users form to monitor who is using Oracle Applications and what they are doing. You can monitor your users at any time.

The Application Monitor lets you see what users are signed on, what responsibilities, forms, and terminals they are using, how long they have been working on forms, and what Oracle database processes they are using.

> **Important:** You can only monitor those users that are being audited by Sign-On Audit. The Application Monitor also reflects the level of auditing you define for your users.

About This Record Window

You can display Sign-On Audit data by choosing from the **Help** menu, **About This Record**...

Sign-On Audit can automatically tie in "About This Record" information for records that are inserted or updated by audited users. This additional information appears in the "About This Record" window when you set the Who:Display Type profile option to Extended.

Extended information shows the Oracle Applications session number, the operating system login name, and the terminal that a user you are tracking with Sign-On Audit used to insert or update a row.

As System Administrator, you can use the System Profile Values form to set “Who:Display Type" to let any user, responsibility, application, or site view Extended "About This Record" information.

Who: Display Type Profile Option

The Who: Display Type profile option allows you to choose between two different displays in the About This Record window:

"Normal" displays the:

- Name of the user who created the row

- Date the user created the row
- Name of the table containing the row
- Name of the user who last updated the row

"Extended" displays Normal information, plus the:

- User's operating system logon
- User's terminal identification

> **Note:** Users cannot see or change this profile option.

This profile option is visible and updatable at all four levels.

Notifying of Unsuccessful Logins

Sign-On Audit can track user logins and provide users with a warning message if anyone has made an unsuccessful attempt to sign on with their application username since their last sign-on. This warning message appears after a user signs on.

You or your users can activate this feature using the Personal Profile Values form by setting the "Sign-On:Notification" user profile option to Yes.

You do not have to audit the user with Sign-On Audit to use this notification feature.

Sign-On Audit Reports

Use the Submit Requests form to print standard audit reports.

You can generate reports detailing what users are signing on, what responsibilities they are accessing, what forms they are using, what concurrent requests they are submitting, and who is attempting to log on to other users' accounts.

Oracle Applications provide the following Sign-On Audit reports:

Signon Audit Concurrent Requests, page 5-27 (shows who submitted what requests)

Signon Audit Forms, page 5-28 (shows who accessed what forms)

Signon Audit Responsibilities, page 5-30 (shows who accessed what responsibilities)

Signon Audit Unsuccessful Logins, page 5-32 (shows who unsuccessfully attempted to sign on as another user)

Signon Audit Users, page 5-33 (shows who signed on to Oracle Applications)

For each report, you can also specify search criteria that makes your report as brief as you need.

Related Topics

Overview of User and Data Auditing, page 5-1

Auditing User Activity, page 5-2

Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Monitor Users, page 5-19

Reporting On AuditTrail Data

AuditTrail lets you keep a history of changes to your important data: what changed, who changed it, and when. With AuditTrail, you can easily determine how any data row or element obtained its current value. You can track information on most types of fields, including character, number and date fields.

When you enter or update data in your forms, you change the database tables underlying those forms. AuditTrail tracks which rows in the database were updated at what time, and which user was logged in using the associated form(s).

AuditTrail

Oracle Applications Releases 10.4 and above provide a mechanism based on Oracle database triggers. AuditTrail stores change information in a "shadow table" of the audited table. This mechanism saves audit data in an uncompressed but "sparse" format, and you enable auditing for particular tables and groups of tables ("audit groups").

Audit Trail Update Tables Report

This program creates database triggers on the tables in your audit groups for your installations. It also creates shadow tables, one for each audited table, to contain the audit information. If you have changed your audit definitions or disabled auditing for an audit group, the program drops or modifies the auditing triggers and shadow tables appropriately.

The program also builds special views you can use to retrieve your audit data for reporting.

Changing Your Audit Tables

You may add additional columns to audit after auditing has begun on a table. However, the shadow table does not track the column changes that occurred before the column(s) were added. If you add columns you must rerun the AuditTrail Update Tables Report to:

- Add the necessary column(s) to the shadow table
- Regenerate the audit triggers and procedures for the table so that they now audit the additional column(s)

Related Topics

Overview of User and Data Auditing, page 5-1

Reporting on AuditTrail Data, page 5-6

Setting Up AuditTrail, page 5-7

Reporting on Audit Information, page 5-13

Disabling AuditTrail and Archiving Audit Data, page 5-14

Audit Installations, page 5-20

Audit Groups, page 5-22

Audit Tables, page 5-24

Setting Up AuditTrail

You can choose to store and retrieve a history of all changes users make on a given table. Auditing is accomplished using *audit groups*, which functionally group tables to be audited. For a table to be audited, it must be included in an enabled audit group.

The steps for setting up AuditTrail include:

Verify Select Privileges on SYS.DBA_TABLES

Have your database administrator grant SELECT privileges on SYS.DBA_TABLES to the APPLSYS account. Normally, this step would already have been done as part of your installation or upgrade.

Define Audit Groups

These are groups of tables and columns, where you do not necessarily need to include all the columns in a given table. You enable auditing for audit groups rather than for individual tables. You would typically group together those tables that belong to the same business process (for example, purchase order tables).

A given table can belong to more than one audit group. If so, the table is audited according to the highest "state" of enabling for any of its groups, where Enabled is the highest, followed by Disable Dump Data, Disable No Growth, and Disable Purge Table, in that order.

You can enable auditing for a maximum of 240 columns for a given table, and you can enable auditing for all types of table columns except LONG, RAW, or LONG RAW. Your audit group must include all columns that make up the primary key for a table; these columns are added to your audit group automatically. Once you have added a column to an audit group, you cannot remove it. See: Audit Groups, page 5-22.

Define Audit Installations

You choose the registered Oracle IDs at your site that you want to audit. This allows you to audit across multiple application installations. When a table is added to an audit group, auditing will automatically be enabled for all installations of the table for which audit is enabled. See: Audit Installations, page 5-20.

Run the Audit Trail Update Tables Report to Enable Auditing

Your AuditTrail definitions (and auditing) do not take effect until you run the Audit Trail Update Tables Report. If you change any of your definitions later, you must rerun this program. You run the Audit Trail Update Tables Report from the standard submission (Submit Reports) form.

> **Important:** AuditTrail requires two database connections. If your operating platform does not automatically support two database connections (e.g. VMS or MPE/XL), then add to your environment file the environment variable FDATDB=<database connect string>.

AuditTrail Tables, Triggers and Views

When auditing is enabled for the first time, a shadow table to the audited table is automatically created in the same Oracle ID as the audited table. The shadow table contains only the columns to be audited, and all columns in the shadow table are unconstrained, regardless of their status in the table to be audited.

For example, NULLs are always permitted in the shadow table. All columns in the shadow table have the same data types and sizes as their counterparts in the audited table.

The name of the shadow table is the first 24 characters of the original table name plus the suffix "_A" (Audit).

Shadow Table Columns

All AuditTrail shadow tables contain certain special auditing columns. These columns include:

- AUDIT_USER_NAME (the Application User ID, except when changes are applied using SQL*Plus, in which case it is the Oracle ID).
- AUDIT_TIMESTAMP (the date/time when the insertion occurred).
- AUDIT_TRANSACTION_TYPE (I for Insert, U for Update, D for Delete, L for Last, and C for Current).
- AUDIT_TRUE_NULLS (VARCHAR2(250) column containing a delimited list of column names that have changed from NULL).
- The primary key for the table. This is not a special column, but rather all the columns comprising the primary key of the audited table. Note that, by convention, all audited columns are stored when a row is deleted. Likewise, an insert results in a row of NULL values in the shadow table. Changes to the primary key are marked as deletes, but new primary key values are inserted also.

For example, suppose you have the following table:

```
 SQL> DESCRIBE AUDIT_DEMO

 NAME            NULL?    TYPE
 --------------- -------- ----
 PRIMARY_KEY              NUMBER(5)
 VALUE_ONE                VARCHAR2(5)
 VALUE_TWO                VARCHAR2(5)
 VALUE_THRE               VARCHAR2(5)
```

Its shadow table is as the following (assuming you audit all your table columns):

```
 SQL> DESCRIBE AUDIT_DEMO_A

NAME                    NULL?     TYPE
----------------------  --------  ----
AUDIT_TIMESTAMP         NOT NULL  DATE
AUDIT_TRANSACTION_TYPE  NOT NULL  VARCHAR2(1)
AUDIT_USER_NAME         NOT NULL  VARCHAR2(100)
AUDIT_TRUE_NULLS                   VARCHAR2(250)
AUDIT_SESSION_ID        NOT NULL    NUMBER
AUDIT_SEQUENCE_ID    NOT NULL    NUMBER
AUDIT_COMMIT_ID        NOT NULL   NUMBER
PRIMARY_KEY                        NUMBER
VALUE_ONE                          VARCHAR2(5)
VALUE_TWO                         VARCHAR2(5)
VALUE_THREE                       VARCHAR2(5)
```

Auditing Triggers and Procedures

When auditing is enabled, the automatically-generated database trigger in the "After" event on the audited table performs the auditing.

This trigger calls a stored procedure to compare each column being audited to see if its value is changing. If so, the procedure saves the previous (old) value to the shadow table.

Auditing creates one row in the shadow table for each audited transaction against the table; thus, a single row in the shadow table represents all old values for all changed columns on that transaction.

The data is not compressed, since a table uses only one byte for a NULL, and AuditTrail represents all unchanged values as NULLs in the shadow table ("sparse" format).

The audit trigger names contain the first 24 characters of the audited table name plus "_AI", "_AU" or "_AD", where one of I, U or D indicates Insert, Update or Delete, respectively. Likewise, the audit procedure names use the first 24 characters of the table name plus "_AIP", "_AUP" or "_ADP". Your table names must be unique within the first 24 characters.

Views

After a shadow table is created, views onto the shadow table are created to allow easier access to the data in the "sparse" rows. These views simplify tasks such as querying a row/column's value on a given date and tracking changes to a row/column over time.

The view name contains the first 24 characters of the audited table name plus "_AC#" or "_AV#" where C or V indicates the type of view and # indicates a number. Due to limitations in creation size, the shadow table columns may need to be broken into multiple views, which are numbered sequentially.

Each view allows slightly different access to the data. One allows the user to reconstruct the value for a row at a given time (_AC), while the other provides simple access to when a value was changed (_AV).

For our example table, the _AV1 and _AC1 views are created as follows:

```
 SQL> DESCRIBE AUDIT_DEMO_AV1

NAME                    NULL? TYPE
-------------------------- ----- ----
PRIMARY_KEY                   NUMBER
AUDIT_TIMESTAMP               DATE
AUDIT_SEQUENCE_ID            NUMBER
AUDIT_SESSION_ID              NUMBER
AUDIT_TRANSACTION_TYPE         VARCHAR2(1)
AUDIT_USER_NAME                VARCHAR2(100)
VALUE_ONE                    VARCHAR2(5)
VALUE_TWO                    VARCHAR2(5)
VALUE_THREE                  VARCHAR2(5)
```

```
 SQL> DESCRIBE AUDIT_DEMO_AC1

NAME                    NULL? TYPE
-----------------------   ----- ----
PRIMARY_KEY                     NUMBER
AUDIT_TIMESTAMP                  DATE
AUDIT_SEQUENCE_ID                NUMBER
AUDIT_SESSION_ID                 NUMBER
AUDIT_TRANSACTION_TYPE          VARCHAR2(1)
AUDIT_USER_NAME                  VARCHAR2(100)
AUDIT_COMMIT_ID                  NUMBER
VALUE_ONE                        VARCHAR2(5)
VALUE_TWO                        VARCHAR2(5)
VALUE_THREE                   VARCHAR2(5)
```

How Data Appears in Tables and Views

Here is an example of how data appears in your original table, your shadow table, and your audit views after a series of changes (starting with an empty AUDIT_DEMO table).

```
SQL> INSERT INTO AUDIT_DEMO VALUES (1,’A’,’A’,’A’);
SQL> INSERT INTO AUDIT_DEMO VALUES (2,’X’,’X’,’X’);
SQL> SELECT PRIMARY_KEY KEY, VALUE_ONE VAL_1,
     VALUE_TWO VAL_2, VALUE_THREE VAL_3 FROM AUDIT_DEMO;

 KEY VAL_1 VAL_2 VAL_3
---- ----- ----- -----
   1 A     A     A
   2 X     X     X


 SQL> UPDATE AUDIT_DEMO SET VALUE_ONE =’B’
     WHERE PRIMARY_KEY = 1;

 KEY VAL_1 VAL_2 VAL_3
---- ----- ----- -----
   1 B     A     A
   2 X     X     X


 SQL> UPDATE AUDIT_DEMO SET VALUE_TWO =’B’
          WHERE PRIMARY_KEY = 1;

 KEY VAL_1 VAL_2 VAL_3
---- ----- ----- -----
   1 B     B     A
   2 X     X     X

SQL> UPDATE AUDIT_DEMO SET VALUE_THREE =’B’
     WHERE PRIMARY_KEY = 1;
SQL> UPDATE AUDIT_DEMO SET VALUE_ONE =’Y’
     WHERE PRIMARY_KEY = 2;
SQL> UPDATE AUDIT_DEMO SET VALUE_ONE = NULL
     WHERE PRIMARY_KEY = 1;
SQL> UPDATE AUDIT_DEMO SET VALUE_ONE =’C’
     WHERE PRIMARY_KEY = 1;
```

After our two inserts and six updates, the final values in the audited table are:

```
 KEY VAL_1 VAL_2 VAL_3
---- ----- ----- -----
   1 C     B     B
   2 Y     X     X
```

The final values in the corresponding shadow table are as follows. A row in the shadow table represents the state of the audited row *before* the audited row was changed. Note that if a value in a row doesn't change during the transaction, the shadow table records a null for that value in that transaction.

In our example, the first two rows in the shadow table represent the state where there was no data for our two audited rows before they were inserted. The "prior values" are null values for the two insert transaction (type I) rows. Similarly, when we update the first value of row 1 to be the value B instead of A, the shadow table records the value A in its third row:

```
 SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
     AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME,
     PRIMARY_KEY KEY, VALUE_ONE VAL_1, VALUE_TWO VAL_2,
     VALUE_THREE VAL_3, AUDIT_TRUE_NULLS FROM AUDIT_DEMO_A;

TIME     TYPE NAME    KEY VAL_1 VAL_2 VAL_3 AUDIT_TRUE_NULLS
-------- ---- ------ ---- ----- ----- ----- ----------------
11:08:16 I    FND60     1
11:08:40 I    FND60     2
11:18:40 U    FND60     1 A
11:20:12 U    FND60     1       A
11:21:54 U    FND60     1             A
11:22:15 U    FND60     2 X
14:20:50 U    FND60     1 B
14:21:15 U    FND60     1                   NYNN

8 rows selected.
```

Given the current values of the row in the audited table, you can trace the changes made to the row by backing up through the corresponding rows in the shadow table.

In our example table, we made two insert and six update transactions, so we see those eight transactions in our shadow table. In the last row, the NYNN indicates that the value in the second table column (VALUE_ONE) has changed from an actual null value (the Y) rather than being an unchanged value (represented by null in the shadow table).

The following two views provide further ways of examining your audited data.

The rows with a transaction type of C in the view indicate the current value of the row when the data was selected (the view is a join between the shadow table and the audited table, so the current value row reflects the current state of the audited table).

The _AC view provides a "filled-in" version of the data, where unchanged values appear instead of being represented by null values. You can order this view by the primary key (rather than by timestamp), so all rows in the shadow table that correspond to a single audited row appear together, with a secondary ordering by timestamp.

```
 SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, ’HH24:MI:SS’) TIME,
     AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME,
     PRIMARY_KEY KEY, VALUE_ONE VAL_1, VALUE_TWO VAL_2,
     VALUE_THREE VAL_3 FROM AUDIT_DEMO_AC1
     ORDER BY PRIMARY_KEY, AUDIT_TIMESTAMP;

TIME     TYPE NAME        KEY VAL_1 VAL_2 VAL_3
-------- ---- ---------- ---- ----- ----- -----
11:08:16 I    FND60         1 A     A     A
11:18:40 U    FND60         1 B     A     A
11:20:12 U    FND60         1 B     B     A
11:21:54 U    FND60         1 B     B     B
14:20:50 U    FND60         1       B     B
14:21:15 U    FND60         1 C     B     B
17:53:34 C                  1 C     B     B
11:08:40 I    FND60         2 X     X     X
11:22:15 U    FND60         2 Y     X     X
17:53:34 C                  2 Y     X     X

10 rows selected.
```

> **Important:** If the changes to your audited table occur faster than one change per second (that is, more frequently than the one-second granularity provided by SYSDATE), you may see "blurring" of records (i.e. more than one record per transaction) in the _AC view, because of joins used in this view. However, the shadow table itself remains correct, and you can resolve the relevant transactions by referring to the shadow table directly.

The _AV1 view provides a more sparse view of the audit data, ordered by timestamp:

```
 SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, ’HH24:MI:SS’) TIME,
     AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME,
     PRIMARY_KEY KEY, VALUE_ONE VAL_1, VALUE_TWO VAL_2,
     VALUE_THREE VAL_3, AUDIT_TRUE_NULLS
     FROM AUDIT_DEMO_AV1;

TIME     TYPE NAME    KEY VAL_1 VAL_2 VAL_3 AUDIT_TRUE_NULLS
-------- ---- ------ ---- ----- ----- ----- ----------------
11:08:16 I    FND60     1
11:08:40 I    FND60     2
11:18:40 U    FND60     1 A
11:20:12 U    FND60     1       A
11:21:54 U    FND60     1             A
11:22:15 U    FND60     2 X
14:20:50 U    FND60     1 B
14:21:15 U    FND60     1                   NYNN
17:58:31 C              1 C     B     B
17:58:31 C              2 Y     X     X

10 rows selected.
```

Here is an example of how you might use a view to determine who changed a particular value and when:

```
 SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
     AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME
     FROM AUDIT_DEMO_AV1
     WHERE PRIMARY_KEY = 1
     AND VALUE_ONE = 'B';

TIME     TYPE NAME
-------- ---- ------
14:20:50 U    FND60
```

Similarly, you might want to determine who changed a value to null and when:

```
 SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
     AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME
     FROM AUDIT_DEMO_AV1
     WHERE PRIMARY_KEY = 1
     AND VALUE_ONE  IS NULL
     AND SUBSTR(AUDIT_TRUE_NULLS,2,1) = 'Y';

TIME     TYPE NAME
-------- ---- ------
14:21:15 U    FND60
```

Reporting on Audit Information

Report on Your Audit Data

You should write audit reports as needed. AuditTrail provides the views of your shadow tables to make audit reporting easier; you can write your reports to use these views.

You may want to create one or more indexes to your shadow table to speed up your reporting. However, such indexes decrease performance during actual auditing of transactions, so you should drop your indexes from the shadow table when you have finished reporting.

> **Important:** Because the structure of the audited table may change between product versions, AuditTrail does not support upgrading existing shadow tables or audited data. Before an upgrade, you should archive the shadow tables and perform all necessary reporting on the audited data.

Related Topics

Overview of User and Data Auditing, page 5-1

Reporting on AuditTrail Data, page 5-6

Setting Up Release AuditTrail, page 5-7

AuditTrail Tables, Triggers and Views, page 5-7

Disabling AuditTrail and Archiving Audit Data, page 5-14

Audit Installations, page 5-20

Audit Groups, page 5-22

Audit Tables, page 5-24

Disabling AuditTrail and Archiving Audit Data

You may report on your audits or disable auditing at any time. When you disable auditing, you should do the following procedure:

Stop Auditing New Transactions

Disable auditing using *either* "Disable - Prepare for Archive" or "Disable - Interrupt Audit" and running the Audit Trail Update Tables report.

Disable - Prepare for Archive
Copies the current values of all rows in the audited table into the shadow table, and then disables the auditing triggers. There is no longer any recording of any changes. You should archive the shadow table before you purge it.

Disable - Interrupt Audit
Modifies the triggers to store one “final" row in the shadow table for each row that is modified in the audit table (remember that a given row in the shadow table represents the data in the audited row *before* an update). If a row in the table being audited is changed again (a second time), that change is not recorded. The shadow table grows slowly, until it contains one row for each row in the table being audited. Then there is no longer any recording of any changes.

Archive Your Audit Data

You should archive the information in the shadow tables according to your business needs.

Clean Out the Shadow Table

Before you restart auditing, you should clean out the shadow table. If there were transactions during the time auditing was disabled, and you did not clean out the shadow table, the data in the shadow table would be invalid because it would have a gap where transactions were not recorded. You purge the shadow table(s) by setting the audit group to Disable - Purge Table and running the Audit Trail Update Tables report.

Disable - Purge Table
Drops the auditing triggers and views and deletes all data from the shadow table.

Restart Auditing (If Desired)

You restart auditing by setting the audit group to Enable Requested and running the Audit Trail Update Tables report again.

> **Important:** If you disable using Disable Purge Table and then reenable auditing for a table, AuditTrail flushes the contents of the shadow table when auditing is reenabled. You should archive any shadow table data that you want to keep before you reenable auditing.

Related Topics

Overview of User and Data Auditing, page 5-1

Reporting on AuditTrail Data, page 5-6

Setting Up AuditTrail, page 5-7

AuditTrail Tables, Triggers and Views, page 5-7

Reporting on Audit Information, page 5-13

Audit Installations, page 5-20

Audit Groups, page 5-22

Audit Tables, page 5-24

Additional Audit Trail Reporting

This section describes how to set up and manage Audit Trail Reporting functions that are used within OPM.

The following topics are covered:

- Audit Industry Template
- Audit Hierarchy Navigator
- Audit Query Navigator
- Running the Audit Report

Audit Industry Template

This window defines the Industry Audit templates. These templates facilitate binding of the required Audit groups together for easy querying and inquiries.

Before using this window, perform the following:

- Define Audit Tables and Audit columns using Oracle Application Audit under the System Administrator responsibility
- Define Audit Groups using Oracle Application Audit under the System Administrator responsibility

Audit Industry Template Procedure

Use this procedure in completing the Industry Template.

1. Navigate to the **Industry Template** window.
2. Complete the fields as described.
3. Save your changes.

Audit Industry Template Fields

These are the fields in the Audit Industry templates.

Template Name

Enter the name of the desired Audit Template.

Functional Areas

- Functional Group - Enter the functional group associated with this template. This is the same as the Audit Group field on the Audit Group window in System Administration.

Audit Hierarchy Editor

Auditing Navigation

In addition to the standard menu and toolbar, a navigator tree provides a hierarchical display of the objects in a treelike framework.

Nodes and Leaves

The higher level nodes in the navigator tree include windows and database objects. All other nodes, and the objects they contain, are indented to indicate that they belong to these higher level nodes. The terminal node is a leaf.

On the Hierarchy Navigator, the highest level is the Audit Template. The next level is the Audit Group (Functional Group), then the audit table, and finally the columns being audited.

On the Query Navigator, the highest level is the Audit Group (Functional Group). The next level is the audit table, and below the audit table are the actual data being audited.

Using the Audit Hierarchy Editor

You can navigate to find what has been set up for auditing. This functionality is accomplished by a tree navigator that starts with the Industry template and drill down to groups, tables, and columns. The navigator lets you see a drill-down view of what columns are being audited. A search facility on the tree is provided to search a table or column.

The navigator fetches the data from the audit table to construct the tree, and relies on the Oracle Applications Object Library table, column registration and uses USER_TABLE_NAME and USER_COLUMN_NAME fields from the FND_TABLES and FND_COLUMNS, respectively.

Before using this window, perform the following:

- Define Audit Tables and Audit columns using the Oracle Application Audit under the System Administrator responsibility
- Define Audit Groups using Oracle Application Audit under the System Administrator responsibility
- Define Industry Audit Templates under the OPM System Administrator responsibility
- Enable Audit Trail, a concurrent process under the System Administrator responsibility

Audit Hierarchy Navigation Procedures

Navigate to the Audit Hierarchy window.

To view table information:

1. Use the tree navigator to view the table names.

2. Select the table name and right-click to display the pop-up menu.
3. Select Display Columns. The Define Query Navigator Display for the Table window displays.

To use the Find Audit Hierarchy function:

1. Use the tree navigator to view the column names.
2. Select the column name and right-click to display the pop-up menu.
3. Select Find. The Find Audit Hierarchy window displays.
4. Select criteria and click Find. A list of templates displays. You can save these as a new audit.

Audit Query Navigator

This interactive query window lets you investigate the changes to any functional group interactively, using a visual approach that is similar to Windows Explorer. When a Particular Node in the left frame is selected, audit trail details are displayed in the right frame. The right frame shows all columns set for auditing. This information is retrieved from the FND_AUDIT_COLUMNS table. The left tree is linked to the right frame with the primary key combination of the table.

Auditing Navigation

In addition to the standard menu and toolbar, a navigator tree provides a hierarchical display of the objects in a treelike framework.

Nodes and Leaves

The higher level nodes in the navigator tree include windows and database objects. All other nodes, and the objects they contain, are indented to indicate that they belong to these higher level nodes. The terminal node is a leaf.

On the Hierarchy Navigator, the highest level is the Audit Template. The next level is the Audit Group (Functional Group), then the audit table, and finally the columns being audited.

On the Query Navigator, the highest level is the Audit Group (Functional Group). The next level is the audit table, and below the audit table are the actual data being audited.

Before using this window, perform the following:

- Define Audit Tables and Audit columns using the Oracle Application Audit under the System Administrator responsibility.
- Define Audit Groups using Oracle Application Audit under the System Administrator responsibility.
- Define Industry Audit Templates under the OPM System Administrator responsibility.
- Define the display look up using the Audit Hierarchy Navigator (Admin Mode). This setup step is not mandatory.
- Enable Audit Trail, a concurrent process under the System Administrator responsibility.

Audit Query Navigation Procedures

Navigate to the Audit Query window.

To use the Find Functional Groups function:

1. Use the tree navigator to view the table names.
2. Select the table name and right-click to display the pop-up menu.
3. Select Find. The Find Functional Groups window displays.
4. Select criteria and click Find. A list of templates displays. You can save these as a new audit.

To view the Audit Results window:

1. Use the tree navigator to view the column names.
2. Select a column name. The Audit Results window automatically displays.
3. Use the Horizontal View and Vertical View buttons to toggle between the two views.

   In the horizontal view, you see the first ten auditing columns. In the vertical view, the column number is unlimited, and can be viewed using the scroll bar.

Audit Report

In situations where comprehensive documentation is needed, (e.g. to support legal or regulatory requirements), a single report request resulting in a single comprehensive report is desirable. This report can then be printed, emailed, or archived.

Since this report could involve a considerable amount of data, a detailed parameter screen is available, allowing you to select only the items of interest.

Submitting the Report

1. Navigate to the Audit Report window. The Enter Report Parameters window is displayed.
2. Select the functional group, or a functional group and audit table name.
3. Complete the optional fields as necessary.
4. Click Select Columns. The Select Reporting Columns window is displayed.
5. Enter at least one column to run the report. The columns displayed are based on the functional group, or a functional group and audit table name criteria selected on the Enter Report Parameters window.
6. Select Print Options. The Select Printing Options window is displayed.
7. Enter the necessary print information.
8. Select OK.
9. Run the report by selecting Run Report.

Enter Report Parameters Field Reference

Functional Group

Specify the name of the functional group for the report. This is the same as the Audit Group field on the Audit Group window in System Administration.

Audit Table Name (Optional)

Specify the table name from the functional group for the report.

Transacted By (Optional)

Specify the user who is requesting the report.

Transaction Type (Optional)

Specify the type of transaction.

From Date (Optional)

Specify the beginning date for the date range the report will run.

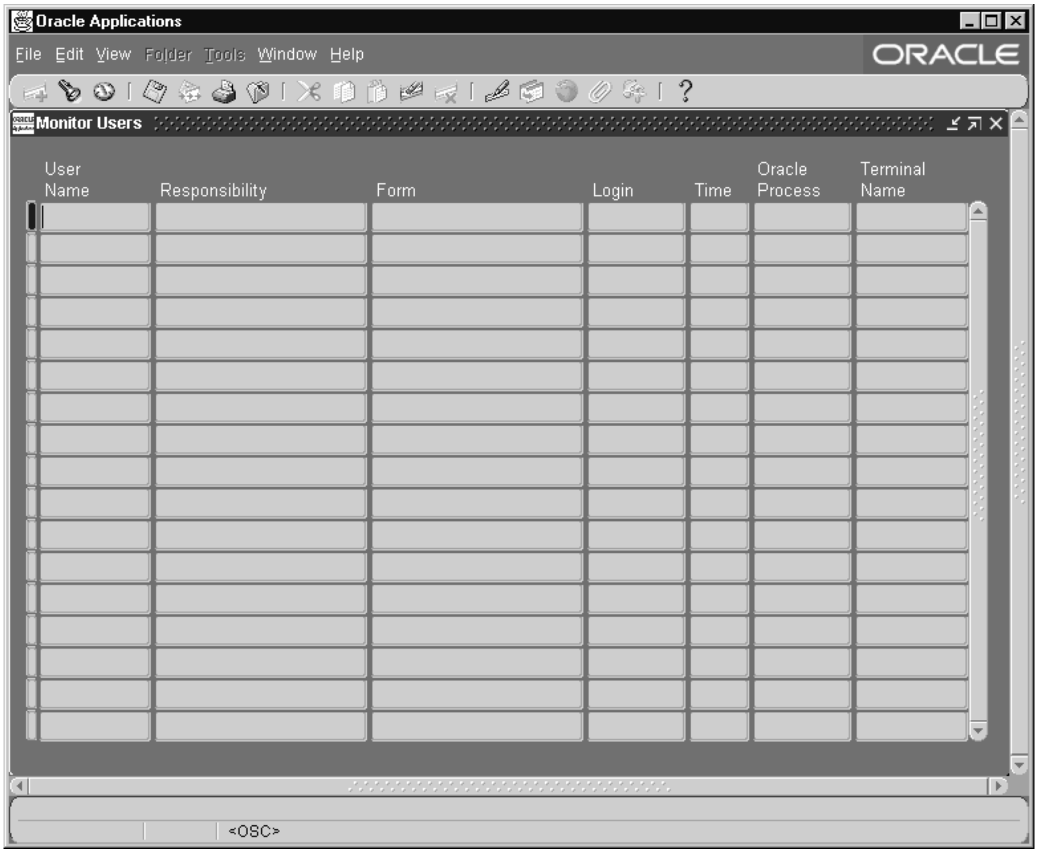
To Date (Optional)

Specify the end date for the date range the report will run.

Monitor Users Window

Use this window to monitor what your application users are currently doing.

Figure 5-1 Monitor Users Window



As well as seeing which users are signed on, you can see:

- Which responsibilities, forms (windows), and terminals they are using
- How long they have been logged in
- What Oracle database processes they are using

In addition, you can monitor all users at a site, all users accessing a specific application or a specific responsibility, or individual users.

> **Note:** You can only monitor those users for whom you have activated Sign-On Audit. See: Overview of User and Data Auditing, page 5-1

Prerequisites

- Select a value for the Sign-On:Audit Level profile option, using the Update System Profile Options window.

Monitor Users Block

Responsibility

The user's responsibility only appears if you have enabled Sign-On Audit at either the Responsibility or Form audit level.

Form

The user's form only appears if you have enabled Sign-On Audit at the Form audit level.

Login

The user's login name.

Time

The length of time the user has been logged on to this application.

ORACLE Process

The ORACLE process of the user.

Terminal Name

The name of the terminal that the user is working on.

Related Topics

Overview of User and Data Auditing, page 5-1

Auditing User Activity, page 5-2

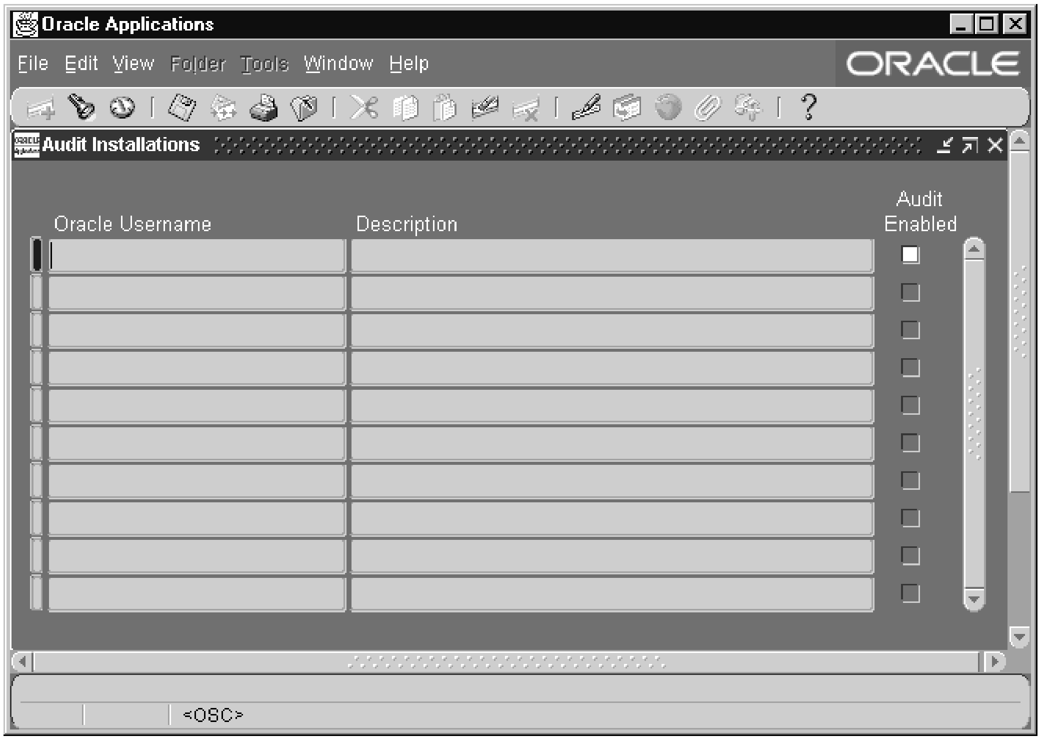
Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Audit Installations Window

Use this window to enable AuditTrail for an Oracle database username at your installation. Such a username grants access privileges to an application's tables and database objects.

Figure 5-2 Audit Installations Window



For auditing to take effect, you must also define one or more audit groups and run the Audit Trail Update Tables report. See: Reporting on AuditTrail Data, page 5-6.

Prerequisites

❒ Register your Oracle username. See: ORACLE Users, *Oracle Applications System Administrator's Guide - Configuration*.

Audit Installations Block

Oracle Username

Select the Oracle username that owns the tables you wish to audit.

Audit Enabled

Check the Audit Enabled check box to enable AuditTrail for an Oracle username. Before auditing takes effect you must define one or more audit groups and run the Audit Trail Update Tables report.

Related Topics

Overview of User and Data Auditing, page 5-1

Reporting on AuditTrail Data, page 5-6

Setting Up AuditTrail, page 5-7

AuditTrail Tables, Triggers and Views, page 5-7

Reporting on Audit Information, page 5-13

Disabling AuditTrail and Archiving Audit Data, page 5-14

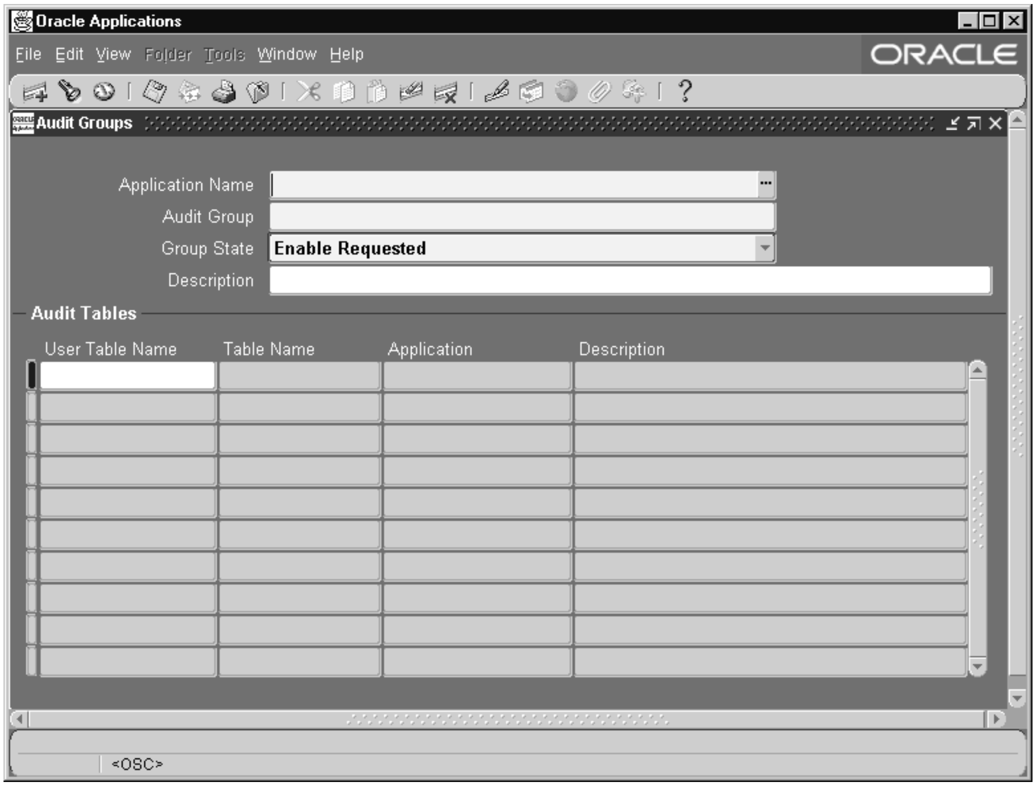
Audit Groups, page 5-22

Audit Tables, page 5-24

Audit Groups Window

Use this window to select the tables that you wish to audit. You audit a table by defining an audit group, which can consist of one or more tables.

Figure 5-3 Audit Groups Window



Group State: Enable Requested

First, identify the tables you want to audit, then, using the Audit Tables window, select which columns in each table you wish to audit. Or, select which columns in a particular table you wish to audit (using the Audit Tables window), then define your audit group (using this window).

To enable or disable auditing for the tables in your audit group, run the Audit Trail Update Tables program using the Submit Requests window. If you change the definition or audit state of your group later, you must rerun this program.

Ensure you have done the following before defining your audit groups:

- Define an audit installation using the Audit Installations window.

> **Important:** Your tables and their primary key information must already be registered and defined for successful auditing. If the table you want to audit is a custom table (not shipped as part of Oracle Applications), you should also perform the following two steps:

- Register your table *and* its primary key columns using Oracle Application Object Library's Tables window (Application Developer Responsibility).
- Run the Register Tables concurrent program from the Submit Requests window.

Audit Groups Block

Identify your audit group and enable or disable auditing for this group.

Application Name

Select the name of an application to associate with your audit group. The combination of application name and group name uniquely identifies your audit group. An audit group may be used to audit tables in additional applications.

Audit Group

Enter the name of the audit group.

Group State

Choose Enable Requested if you are defining a new audit group. When you run the Audit Trail Update Tables report, the concurrent program creates database triggers for the tables in your audit group. Once you have run the program, this field displays Enabled for audit groups where AuditTrail is active.

> **Important:** All primary key columns in each table in an audit group are automatically selected for auditing, whether or not you use the Audit Tables window to select which columns you wish to audit.

To disable auditing for a group, choose one of the following options and then run the Audit Trail Update Tables report to have your changes take effect.

Disable - Prepare for Archive
Copies the current values of all rows in the audited table into the shadow table, and then disables the auditing triggers. This option requires the most space, since there is at least one row in the shadow table for every row in the audited table (and another row in the shadow table for each transaction on the original row in the audited table). You should then archive the table before you empty the shadow table.

Disable - Interrupt Audit
Modifies the triggers to store one final row in the shadow table as the audited row is modified in the audit table (remember that a given row in the shadow table represents the data in the audited row *before* an update). Inserts or further changes are no longer audited. The shadow table then grows slowly, and the data may be accessed by the existing audit views.

Disable - Purge Table
Drops the auditing triggers and views and deletes all data from the shadow table.

Audit Tables Block

Identify the application tables you want to audit in your audit group.

User Table

Select the end user table name (frequently the same name as the table name) for your database table. Once you choose a table, you see its table name and associated application.

Table Name

This field displays the actual name for the table you have selected to include in your audit group.

Application

This field displays the application name for the table you have selected to include in your audit group.

Description

This field displays the description for the table you have selected to include in your audit group.

Related Topics

Overview of User and Data Auditing, page 5-1

Reporting on AuditTrail Data, page 5-6

Setting Up AuditTrail, page 5-7

AuditTrail Tables, Triggers and Views, page 5-7

Reporting on Audit Information, page 5-13

Disabling AuditTrail and Archiving Audit Data, page 5-14

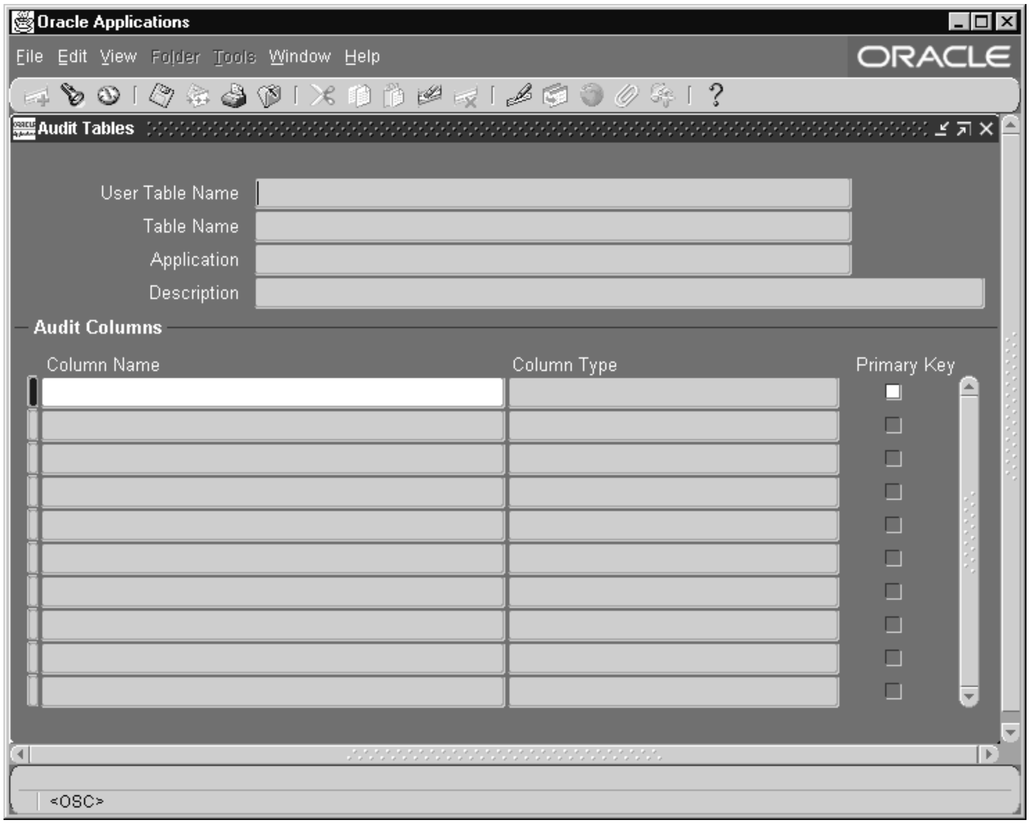
Audit Installations, page 5-20

Audit Tables, page 5-24

Audit Tables Window

Use this window to select which columns in a table you wish to audit.

Figure 5-4 Audit Tables Window



First, identify the columns in a table you want to audit. Then, using the Audit Groups window, include the table as part of an audit group. Or, you may define your audit group first (using the Audit Groups window), and then select which columns in the table you want to audit (using this window).

To enable or disable auditing for the tables in your audit group (i.e., the columns you have selected here), you must run the Audit Trail Update Tables program using the Submit Requests window. If you select additional columns to audit, or change the definition or audit state of your group later, you must rerun this program.

Ensure the following is done before defining your audit tables:

- Define an audit installation using the Audit Installations window.

> **Important:** Your tables and their primary key information must already be registered and defined for successful auditing. If the table you want to audit is a custom table (not shipped as part of Oracle Applications), you should also perform the following two steps:

- Register your table *and* its primary key columns using Oracle Application Object Library’s Tables window (Application Developer Responsibility).
- Run the Register Tables concurrent program from the Submit Requests window.

Define Audit Tables Block

Identify the application table you want to audit. Successively selecting *Go - Next Record* from the menu or toolbar displays, in alphabetical order, the name of each application table registered at your installation site.

User Table Name

Select the end user table name (frequently the same name as the table name) for your database table. Once you choose a table, you see its table name and associated application.

Table Name

This field displays the actual name for the table you have selected to include in your audit group.

Application

This field displays the application name for the table you have selected to include in your audit group.

Audit Columns Block

Select the columns you want to audit. Successively selecting *Go - Next Record* from the menu or toolbar displays, in alphabetical order, the name of each application table registered at your installation site.

- You cannot delete a column from auditing once it has been selected.
- You may add additional columns to be audited.
- Each time you select a column to be audited, that change affects every audit group that includes the table which owns the column.

Column Name

Enter the name of the database column you want to audit. You should not explicitly enter the names of your table's primary key columns, since they are entered automatically, and you will get an error message if you try to save a duplicate column name. You can query to see which columns appear automatically.

Note that once you have chosen a column, you cannot delete it from the audit set, though you may add other columns to the set later.

Once you choose a column, you see its column type and whether it is part of the primary key for this table.

Column Type

This field describes the type of data the column stores, for example, varchar2.

Primary Key

This field displays Yes or No indicating whether the column you are auditing is a primary key column.

Any primary key columns you do not select to audit are automatically included when you save your column selections. For example, if the table you are auditing has two primary key columns, and you choose to audit one of them, the second primary key column is automatically selected when you save your column selections.

Related Topics

Overview of User and Data Auditing, page 5-1

Reporting on AuditTrail Data, page 5-6

Setting Up AuditTrail, page 5-7

AuditTrail Tables, Triggers, and Views, page 5-7

Reporting on Audit Information, page 5-13

Disabling AuditTrail and Archiving Audit Data, page 5-14

Audit Installations, page 5-20

Audit Groups, page 5-22

Signon Audit Concurrent Requests Report

Use this report to view information about who is requesting what concurrent requests and from which responsibilities and forms.

> **Important:** You can only generate Signon Audit Concurrent Requests Reports for those users you are auditing.

Report Parameters

Sort By

Sort the information in your report by operating system login name, the requested start date, and/or application username.

Login Name

Search for a specific login name that meets your other search criteria. If you leave this parameter blank, your report contains all login names that meet your other search criteria.

User Name

Search for a specific application username that meets your other search criteria. If you leave this parameter blank, your report contains all application usernames that meet your other search criteria.

From Request Start Time/To Request Start Time

Search for concurrent requests that meet your other search criteria and have requested start times in a specific time period. Use these parameters to specify the start and end of your time period. If you leave these parameters blank, your report contains concurrent requests from any date that also meet your other search criteria to the current date for this parameter.

Report Heading

The report heading displays the search criteria you entered as parameter values.

Column Headings

Login Name

The operating system login name of the user who submitted the concurrent request.

Request ID

The concurrent request ID of the submitted concurrent request. Use the Concurrent Requests form to view completion information for a concurrent request ID.

Concurrent Program Name

The name of the concurrent program the user submitted. Use the Concurrent Programs form to view detail information about a concurrent program.

User Name

The Oracle Applications username of the user who submitted the concurrent request. Use the Users form to view detail information about an application user. See: Users, page 4-22.

Responsibility Name

The name of the responsibility from which the user submitted the concurrent request. The responsibility displays only if you audited the user at the responsibility or form Sign-on Audit level. Use the Responsibilities form to view detailed information about a responsibility. See: Responsibilities, page 4-18.

Form Name

The name of the form from which the user submitted the concurrent request. The form name displays only if you audited the user at the form Sign-On Audit level.

Requested Start Time

The date and time the concurrent request started running.

Related Topics

Overview of User and Data Auditing, page 5-1

Auditing User Activity, page 5-2

Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Monitor Users field help, page 5-19

Signon Audit Forms Report

Use this report to view who is navigating to what form and when they do it.

> **Important:** You can only generate a Signon Audit Forms Report for those users you are auditing.

Report Parameters

Sort By

Sort the information in your report by the time users entered or left a form, the name of the form that users access, the operating system login name of the user, the responsibility users access, the terminal that users are on, and/or the application username.

Login Name

Search for information about a specific login name that meets your other search criteria. If you leave this parameter blank, your report contains all login names that meet your other search criteria.

User Name

Search for information about a specific application username that meets your other search criteria. If you leave this parameter blank, your report contains all application usernames that meet your other search criteria.

Terminal Name

Search for information about a specific terminal that meets your other search criteria. If you leave this parameter blank, your report contains all terminal names that meet your other search criteria.

Responsibility Name

Search for information about a specific responsibility that meets your other search criteria. If you leave this parameter blank, your report contains all responsibilities that meet your other search criteria.

Form Name

Search for information about a specific form that meets your other search criteria. If you leave this parameter blank, your report contains all forms that also meet your other search criteria.

From Active Date/To Active Date

Search for information about forms accessed by users within a specific time period and that meet your other search criteria. Use these parameters to specify the start and end of your time period. If you leave these parameters blank, your report contains forms accessed from any date that also meet your other search criteria to the current date for this parameter.

Report Heading

The report heading displays the search criteria you entered as parameter values.

Column Headings

Username

The Oracle Applications username of the user who accessed the form. Use the Users form to view detailed information about an application user. See: Users, page 4-22.

Login Name

The operating system login name of the user who accessed the form.

Terminal Name

The operating system ID of the terminal from which the user accessed the form.

Responsibility Name

The name of the responsibility from which the user accessed the form. The responsibility displays only if you audited the user at the responsibility or form Sign-on Audit level. Use the Responsibilities form to view detailed information about a responsibility. See: Responsibilities, page 4-18.

Start Active Time/End Active Time

The dates and times when the user accessed/exited the form. The start active time and end active time display only if you audited the user at the form Sign-on Audit level.

Form Name

The name of the form that the user accessed. The form name displays only if you audited the user at the form Sign-on Audit level.

Related Topics

Overview of User and Data Auditing, page 5-1

Auditing User Activity, page 5-2

Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Monitor Users field help, page 5-19

Signon Audit Responsibilities Report

Use this report to view who is selecting what responsibility and when they do it.

> **Important:** You can only generate Signon Audit Responsibilities Reports for those users you are auditing.

Report Parameters

Sort By

Sort the information in your report by the time users entered or left a responsibility, the operating system login name of the user, the responsibility name, the terminal that users are on, and/or the application username.

Login Name

Search for information about a specific login name that meets your other search criteria. If you leave this parameter blank, your report contains all login names that meet your other search criteria.

User Name

Search for information about a specific application username that meets your other search criteria. If you leave this parameter blank, your report contains all application usernames that meet your other search criteria.

Terminal Name

Search for information about a specific terminal that meets your other search criteria. If you leave this parameter blank, your report contains all terminal names that meet your other search criteria.

Responsibility Name

Search for information about a specific responsibility that meets your other search criteria. If you leave this parameter blank, your report contains all responsibilities that meet your other search criteria.

From Active Date/To Active Date

Search for information about responsibilities accessed by users within a specific time period and that meet your other search criteria. Use these parameters to specify the start and end of your time period. If you leave these parameters blank, your report contains responsibilities accessed from any date that also meet your other search criteria to the current date for this parameter.

Report Heading

The report heading displays the search criteria you entered as parameter values.

Column Headings

Username

The Oracle Applications username of the user who selected the form. Use the Users form to view detail information about an application user. See: Users, page 4-22.

Login Name

The operating system login name of the user who selected the responsibility.

Terminal Name

The operating system ID of the terminal from which the user selected the responsibility.

Responsibility Name

The name of the responsibility the user used. The responsibility displays only if you audited the user at the responsibility or form Sign-on Audit level. Use the Responsibilities form to view detailed information about a responsibility. See: Responsibilities, page 4-18.

Start Active Time/End Active Time

The dates and times when the user selected/exited the responsibility. The start active time and end active time display only if you audited the user at the responsibility or form Sign-On Audit level.

Related Topics

Overview of User and Data Auditing, page 5-1

Auditing User Activity, page 5-2

Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Monitor Users field help, page 5-19

Signon Audit Unsuccessful Logins Report

Use this report to view who unsuccessfully attempted to sign on to Oracle Applications as another user. An unsuccessful login occurs when a user enters a correct username but an incorrect password.

You can generate Signon Audit Unsuccessful Logins Reports for any users, regardless of whom you are auditing.

Report Parameters

Sort By

Sort the information in your report by the time users attempt to login, operating system login name of the user, the terminal that users are on, and/or the application username.

Login Name

Search for information about a specific login name that meets your other search criteria. If you leave this parameter blank, your report contains all login names that meet your other search criteria.

User Name

Search for information about a specific application username that meets your other search criteria. If you leave this parameter blank, your report contains all application usernames that meet your other search criteria.

Terminal Name

Search for information about a specific terminal that meets your other search criteria to make your report as brief as you need. If you leave this parameter blank, your report contains all terminal names that meet your other search criteria.

From Attempt Date/To Attempt Date

Search for information about unsuccessful logins within a specific time period and that meet your other search criteria. Use these parameters to specify the start and end of your time period. If you leave these parameters blank, your report contains unsuccessful logins from any date that also meet your other search criteria to the current date for this parameter.

Report Heading

The report heading displays the search criteria you entered as parameter values.

Column Headings

Username

The Oracle Applications username of the user who unsuccessfully tried to sign on. Use the Users form to view detail information about an application user. See: Users, page 4-22.

Login Name

The operating system login name of the user who unsuccessfully tried to sign on.

Terminal

The operating system ID of the terminal from which the user unsuccessfully tried to sign on.

Attempt Time

The date and time when the user unsuccessfully tried to sign on. See: Monitor Users, page 5-19.

Related Topics

Overview of User and Data Auditing, page 5-1

Auditing User Activity, page 5-2

Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Signon Audit Users Report

Use this report to view who signs on and for how long.

> **Important:** You can only generate Signon Audit Users Reports for those users you are auditing.

Report Parameters

Sort By

Sort the information in your report by the time users start or finish using an application username, the operating system login name of the user, the terminal that users are on, and/or the application username.

Login Name

Search for information about a specific login name that meets your other search criteria to make your report as brief as you need. If you leave this parameter blank, your report contains all login names that meet your other search criteria.

User Name

Search for information about a specific application username that meets your other search criteria to make your report as brief as you need. If you leave this parameter blank, your report contains all application usernames that meet your other search criteria.

Terminal Name

Search for information about a specific terminal that meets your other search criteria to make your report as brief as you need. If you leave this parameter blank, your report contains all terminal names that meet your other search criteria.

From Active Date/To Active Date

You can search for information about users logged into Oracle Applications within a specific time period and that meet your other search criteria. Use these parameters to specify the start and end of your time period. If you leave these parameters blank, your report contains user information from the first date that also meets your other search criteria to the current date.

Report Heading

The report heading displays the search criteria you entered as parameter values.

Column Headings

Session Number

The Oracle Applications session number that ‘uniquely identifies each application user sign-on.

User Name

The Oracle Applications username of the user who signed on. Use the Users form to view detailed information about an application user. See: Users, page 4-22.

Login Name

The operating system login name of the user who signed on.

Terminal Name

The operating system ID of the terminal from which the user signed on.

Start Active Time/End Active Time

The dates and times when the user signed on and off from Oracle Applications. The start active time and end active time display only if you audited the user at the user Sign-On Audit level.

Oracle Process

The Oracle database process ID used during the user's sign-on. Consult your Database Administrator for more information concerning Oracle processes.

System Process

The operating system process ID used during the user's sign-on. Consult your operating system administrator for more information concerning your operating system process ID.

Related Topics

Overview of User and Data Auditing, page 5-1

Auditing User Activity, page 5-2

Setting Up Sign-On Audit, page 5-2

Sign-On Audit Reports, page 5-5

Monitor Users field help, page 5-19

Purge Signon Audit Data Program

Use this program to purge Sign-On Audit information created before a specified date.

The following data is deleted:

- Data for who signs on and for how long
- Data for who is selecting what responsibility and when they do it
- Data for who uses which forms in an application and when

Parameters

Audit Date

The Sign-On Audit information creation date. This program will delete all Sign-On Audit information created before this date.

A

Security Configuration and Maintenance

Security Configuration and Maintenance

Oracle Applications offers additional features that help you secure your system and monitor access to the system.

The following is described in *Oracle Applications System Administrator's Guide - Configuration:*

- Administering server security
- Restricting access to responsibilities based on a user's web server
- Integrating with Oracle9*i* Application Server, including implementing Single Sign-On functionality via Oracle Portal, Oracle Login Server, and Oracle Internet Directory
- Managing SQL*Net access from middle-tier hosts

The following is described in *Oracle Applications System Administrator's Guide - Maintenance:*

- Monitoring Security using Oracle Applications Manager

Index