

# Oracle® Business Intelligence Applications

Security Guide

Release 7.9.7

E18914-01

# Table of Contents

Preface

Chapter 1: What's New in This Release

Chapter 2: Integrating Security for Oracle BI Applications

# Preface

Oracle Business Intelligence Applications are comprehensive prebuilt solutions that deliver pervasive intelligence across an organization, empowering users at all levels, from front line operational users to senior management, with the key information they need to maximize effectiveness. Intuitive and role-based, these solutions transform and integrate data from a range of enterprise sources, including Siebel, Oracle, PeopleSoft, JD Edwards, SAP R/3 and corporate data warehouses, into actionable insight that enables more effective actions, decisions, and processes.

Oracle BI Applications are built on Oracle Business Intelligence Suite Enterprise Edition, a comprehensive next-generation BI and analytics platform.

Oracle BI Applications includes the following:

- Oracle Financial Analytics
- Oracle Human Resources Analytics
- Oracle Supply Chain and Order Management Analytics
- Oracle Procurement and Spend Analytics
- Oracle Project Analytics
- Oracle Sales Analytics
- Oracle Service Analytics
- Oracle Contact Center Telephony Analytics
- Oracle Marketing Analytics
- Oracle Loyalty Analytics
- Oracle Price Analytics

For more details on the applications included in this release of Oracle BI Applications, see the *Oracle Business Intelligence Applications Licensing and Packaging Guide*. This guide is included in the Oracle Business Intelligence Media Pack. Also, see the System Requirements and Supported Platforms for Oracle Business Intelligence Applications, available at available at.

[http://www.oracle.com/technology/documentation/bi\\_apps.html](http://www.oracle.com/technology/documentation/bi_apps.html).

*Oracle Business Intelligence Applications Security Guide* contains information about the security features in Oracle BI Applications.

Oracle recommends reading the *Oracle Business Intelligence Applications Release Notes* before installing, using, or upgrading Oracle BI Applications. The most current version of the *Oracle Business Intelligence Applications Release Notes* is available:

- On the Oracle Technology Network at [http://www.oracle.com/technology/documentation/bi\\_apps.html](http://www.oracle.com/technology/documentation/bi_apps.html) (to register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>)

## Audience

This document is intended for BI Managers and implementers of Oracle BI Applications

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. 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.

## Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

## Related Documents

For more information, see the following documents in the Oracle BI Applications release 7.9.7 documentation set (available at [http://www.oracle.com/technology/documentation/bi\\_apps.html](http://www.oracle.com/technology/documentation/bi_apps.html)):

- *Oracle Business Intelligence Applications Release Notes*
- *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*
- *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*
- *Oracle Business Intelligence Configuration Guide for Informatica PowerCenter Users*
- *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*
- *Oracle Business Analytics Warehouse Data Model Reference*

## Conventions

The following text conventions are used in this document:

- **Boldface** type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
- *Italic* type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

- `Monospace` type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

## Chapter 1: What's New in This Release

This topic lists changes described in this version of the documentation to support release 7.9.7 of the software.

### What's New in Oracle Business Intelligence Applications Security Guide, Version 7.9.7

This guide includes the following change

- Added this topic: Integrating Data Security for SAP R/3.

## Chapter 2: Integrating Security for Oracle BI Applications

This topic describes the security features in Oracle Business Intelligence Applications. It contains the following main topics:

- About Security in Oracle BI Applications
- Data-Level Security In Oracle BI Applications
- Object-Level Security in Oracle BI Applications
- User-Level Security in Oracle BI Applications
- Extending Security in Oracle BI Applications
- Integrating Data Security for Oracle EBS
- Integrating Data Security for Oracle's PeopleSoft Enterprise Applications
- Integrating Data Security for Oracle's Siebel CRM Applications
- About Security Integration with Oracle's JDEdwards EnterpriseOne or JDEdwards World
- Integrating Data Security for SAP R/3

### About Security in Oracle BI Applications

This topic contains the following topics:

- Oracle BI Applications Security Types
- Use of Security Groups in Oracle BI Applications
- Checking Oracle BI Applications User Responsibilities
- About Adding a New User Responsibility in Oracle Business Intelligence

### Security Integration Between Oracle Business Intelligence Enterprise Edition and Oracle BI Applications

Oracle BI Applications integrates tightly with Oracle Business Intelligence Enterprise Edition as well as the security model of the operational source system to allow the right content to be shown to the right user.

You should be thoroughly familiar with the security features of Oracle Business Intelligence Enterprise Edition before you begin working with Oracle BI Applications.

Security settings for Oracle Business Intelligence Enterprise Edition are made in the following Oracle Business Intelligence components:

- Oracle BI Administration Tool

You can use the Oracle BI Administration Tool to perform tasks such as setting permissions for business models, tables, columns, and subject areas; specifying filters to limit data accessibility; and setting authentication options. For detailed information, see *Oracle Business Intelligence Server Administration Guide*.

- Oracle BI Presentation Services Administration

You can use Oracle BI Presentation Services Administration to perform tasks such as setting permissions to Presentation Catalog objects, including dashboards and dashboard pages. For detailed information, see *Oracle Business Intelligence Presentation Services Administration Guide*.

## Oracle BI Applications Security Types

Security in Oracle BI Applications can be classified broadly into the following three types:

- **Data-level security.** Data-level security controls the visibility of data (content rendered in subject areas, dashboards, Oracle BI Answers, and so on) based on the user's association to data in the transactional system. For more information, see *Data-Level Security In Oracle BI Applications*.
- **Object-level security.** Object-level security controls the visibility to business logical objects based on a user's role. You can set up object-level security for metadata repository objects, such as subject areas and presentation folders, and for Web objects, such as dashboards and dashboard pages, which are defined in the Presentation Catalog. For more information, see *Object-Level Security in Oracle BI Applications*.
- **User-level security (authentication of users).** User-level security refers to authentication and confirmation of the identity of a user based on the credentials provided. For more information, see *User-Level Security in Oracle BI Applications*.

## Use of Security Groups in Oracle BI Applications

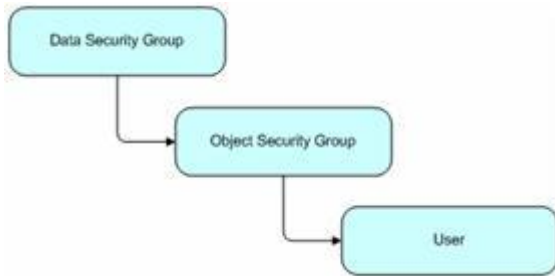
Object-level and data-level security are implemented in Oracle BI Applications using security groups. These security groups are defined using the Security Manager in the Oracle BI Administration Tool. Security groups can be related to data, objects, or both data and objects. For example, the Oracle BI Applications repository (OracleBIAnalyticsApps.rpd) contains the following security groups:

- The Business Group Org-Based security group is a data security group filter used to control access to human resources data.
- The Human Resources Analyst security group is an object security group used to control presentation layer object visibility for the Human Resources Analyst role.
- The Human Resources Recruiting Manager security group is an example of a security group related to both data and objects.

For detailed information about setting up and managing security groups, and for information about using the Oracle BI Administration Tool Security Manager, see *Oracle Business Intelligence Server Administration Guide*.

The standard hierarchical structure of security groups and users in Oracle BI Applications is the following: data security group, then object security group, then user, as shown the following figure.

Figure 1. Security Group Hierarchy in Oracle BI Applications



1. Create security groups in the Oracle BI Repository with the same names as existing responsibilities or groups in the source applications. These security groups are added as members to Oracle BI-specific security groups, and the users will inherit this membership based on their own responsibilities or roles in the OLTP application.
2. Add new Oracle BI-specific responsibilities (Oracle EBS and Siebel CRM Applications) or roles (PeopleSoft Enterprise applications and SAP R/3) in the source applications, making sure their names match the object security groups in Oracle BI Applications, and assign OLTP users to these new groups. The users will then inherit the security group membership in the same way as described in the first method above.

For information about integrating user and object security with JD Edwards, see About Security Integration with Oracle's JD Edwards EnterpriseOne or JD Edwards World.

**Note:** Users should always be created in the operational application databases or directory services, such as LDAP, and never in the Oracle BI Repository. If users are created in the Oracle BI Repository, the security mechanism does not work.

## Checking Oracle BI Applications User Responsibilities

An administrator can check a user's responsibility in the following ways:

- In the Siebel or Oracle EBS operational applications, go to the Responsibilities view.
- In PeopleSoft applications, go to the Roles view to check a user's roles.
- In JD Edwards EnterpriseOne applications, go to the User Profiles application (P0092) to check a user's roles.
- In JD Edwards World, go to User Information Revisions (P0092) to check a user's roles.
- In SAP R/3 application, go to the transaction SU01 and check for roles assigned to the User.
- In the Oracle BI application, click on Settings/My Account link. The Presentation Services group membership for the user is shown near the bottom of the Web page. These are the Presentation Services groups, defined in the Presentation Services Catalog only. These groups are usually used to control the ability to perform actions (privileges). If a Presentation Services group has the same name as an Oracle BI Server security group, and the user is a member of the latter, than he will become automatically a member of the corresponding Presentation Services group.

## About Adding a New User Responsibility in Oracle Business Intelligence

When you add a new responsibility to a user in Oracle BI Presentation Services, the change is not immediately reflected in the Oracle Business Intelligence environment. In order to register the new user responsibility, both the administrator and the user must perform the following tasks:

1. The Oracle BI administrator must reload the Oracle BI Server metadata through Oracle BI Presentation Services. To reload the metadata, in Oracle Business Intelligence Answers, select Settings and then Administration. Next, click Reload Files and Metadata.

For more information on adding a new responsibility, see *Oracle Business Intelligence Server Administration Guide* and *Oracle Business Intelligence Presentation Services Administration Guide*.

2. Then the user must log out from the Oracle BI application (or from Siebel or Oracle EBS operational application if the user is looking at Oracle BI dashboards using an embedded application) and then log in again.

## Data-Level Security In Oracle BI Applications

This topic describes the data-level security features in Oracle BI Applications. It contains the following topics:

- Overview of Data-Level Security in Oracle BI Applications
- Viewing Permissions in Oracle BI Administration Tool
- Implementing Data-Level Security in the Oracle BI Repository
- Initialization Blocks Used for Data-Level Security in Oracle BI Applications

### Overview of Data-Level Security in Oracle BI Applications

Data-level security defines what a user in an OLTP application can access inside a report. The same report, when run by two different users, can bring up different data. This is similar to how the My Opportunities view in an operational application displays different data for different users. However, the structure of the report is the same for all users, unless a user does not have access to a column in a report, in which case the column is not displayed for that user.

Table 1 shows the security groups that are supported in Oracle BI Applications. During installation and configuration, you must make sure the correct security group and initialization blocks are set up for your environment.

For more information about the use of initialization blocks in Oracle Business Intelligence, see *Oracle Business Intelligence Server Administration Guide*.

**Table 1. Summary of Supported Security Groups by Source System**

Security Group	Oracle EBS	PeopleSoft Financials	PeopleSoft HR	Siebel	SAP R/3
Operating Unit Org-Based security	Available since 7.9.3	Available since 7.9.3		Available since 7.9.5	
Inventory Org-Based Security	Available since 7.9				
Company Org Based Security	Available in 7.9.3 and obsolete in 7.9.4	Available in 7.9.3	Available in 7.9.3		Available in 7.9.7
Business Group Org-Based	Available since 7.9.3				

## Security

HR Org-Based  
Security

Available since  
7.9.3 and  
enhanced in  
7.9.6 to  
support  
PeopleSoft  
department  
Security

Payables Org-  
Based Security

Available since  
7.9.3

Available in  
7.9.7

Receivables  
Org-Based  
Security

Available since  
7.9.3

SetID-Based  
Security

Available since  
7.9.3

Available since  
7.9.3

Position-Based  
Security

Available since  
7.9.4 for HRMS

Available since  
7.9.3

Available since  
7.5

Ledger-Based  
Security

Available since  
7.9.4

Available since  
7.9.4

## Implementing Data-Level Security in the Oracle BI Repository

Data-level security in Oracle BI Applications is implemented in three major steps, as described below. For instructions on performing these steps, see *Oracle Business Intelligence Server Administration Guide*.

1. Set up initialization blocks that obtain specific security-related information when a user logs in, for example, the user's hierarchy level in the organization hierarchy, or the user's responsibilities.

For a description of the preconfigured initialization blocks, see Initialization Blocks Used for Data-Level Security in Oracle BI Applications.

2. Set up the joins to the appropriate security tables in the metadata physical and logical layers.

For detailed information about this security feature, see *Oracle Business Intelligence Server Administration Guide*.

3. Set up the filters for each security group on each logical table that needs to be secured.

For detailed information about this security feature, see *Oracle Business Intelligence Server Administration Guide*.

## Viewing Permissions in Oracle BI Administration Tool

You can view (and change) the permissions that define data-level security in the Oracle BI Administration Tool. For detailed information about this security feature, see *Oracle Business Intelligence Server Administration Guide*.

### To view permissions in Oracle BI Administration Tool

1. From the Administration Tool menu bar, select Manage, and then Security.

2. In the tree pane of the Security Manager dialog box, select Groups.
3. In the right pane, double-click the group for which you want to view permissions.
4. In the Group dialog box, click Permissions.
  - The General tab displays objects for which permission has been granted or denied access for the specified group.
  - The Query Limits tab displays, based on the database, the following:
    - Limits placed on the maximum number of rows each query can retrieve.
    - Maximum time a query can run on a database.
    - Time periods during which access to the database is restricted.
    - Status of the Populate privilege.
    - Status of the Execute Direct Database Requests privilege.
  - The Filters tab displays the dimension and fact tables set up as filters for the specified group.

## Initialization Blocks Used for Data-Level Security in Oracle BI Applications

For more information about setting up and managing initialization blocks, see *Oracle Business Intelligence Server Administration Guide*.

In the Oracle BI Repository, the initialization blocks are set up for obtaining a given user's primary position, primary organization, and the owner ID, as described below:

- **Authorization**

This initialization block is used to associate users with all security groups to which they belong. It obtains a user's responsibilities or roles from the source OLTP application, matches them with Oracle BI Applications security groups, and determines the user's applicable object security during the session. This initialization block populates a variable set called GROUP.
- **Business Groups**

This initialization block is used to retrieve the business groups from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called BUSINESS\_GROUP, which is used to drive security permissions for business group org-based security.
- **Companies**

This initialization block is used to retrieve the companies from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called COMPANY, which is used to drive security permissions for company org-based security. COMPANY is mapped to the PeopleSoft business unit and SAP R/3.
- **HR Organizations**

This initialization block is used to retrieve the HR organizations from the OLTP application to which the corresponding login user has access. This initialization block populates a variable set called HR\_ORG, which is used to drive security permissions for HR analysts.
- **Inventory Organizations**

This initialization block is used to retrieve the inventory organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called INV\_ORG, which is used to drive security permissions for inventory org-based security.

- Ledgers

This initialization block is used to retrieve the ledgers from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called LEDGER, which is used to drive security permissions for ledger-based security.

- Operating Unit Organizations

This initialization block is used to retrieve the operating unit organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called OU\_ORG, which is used to drive security permissions for operating unit org-based security.

- Orgs for Org-Based Security

This initialization block is used to retrieve the organizations reporting to the current user's business unit. This initialization block populates a variable set called ORGANIZATION, which is used to drive primary org-based security.

- Payable Organizations

This initialization block is used to retrieve the payable organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called PAYABLE\_ORG, which is used to drive security permissions for payable org-based security.

- Primary Owner ID

This initialization block obtains the owner ID for the given user. It obtains this information from the Siebel OLTP and populates the PR\_OWNER\_ID variable.

- Payables Organizations

This initialization block is used to retrieve the payables organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called PAYABLE\_ORG, which is used to drive security permissions for payables org-based security.

- SetID

This initialization block is used to retrieve the set IDs from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called SET\_ID, which is used to drive security permissions for Set ID-based security.

- User Hierarchy Level

This initialization block obtains the fixed hierarchy level of the given user, based on the user's login, from W\_POSITION\_DH. It populates the variable HIER\_LEVEL. The SQL used by the block is run against the data warehouse. Therefore, it reflects the hierarchy level at the time of the last ETL run that populated this table (W\_POSITION\_DH).

- User HR Organizations

This initialization block is used to retrieve the current HR organization from OLTP application to which the current user belongs. This initialization block populates a variable called USER\_HR\_ORG.

## Data Security Groups in Oracle BI Applications

The table below describes the security groups used in Oracle BI Applications and the application to which they apply. Some selected security groups share the same name as responsibilities for Siebel CRM and Oracle EBS applications and roles for PeopleSoft applications. A user who has any of these responsibilities or roles in the source application will be a member of the corresponding data security group automatically when he logs in to the Oracle BI application. Other security groups based on similar objects in the source application can be added to the Oracle BI Repository and added to these data-level security groups, if you need the corresponding data filters to apply to any additional group of users. The following table shows the security groups that are supported in Oracle BI Applications.

**Table 2. Data Security Groups in Oracle BI Applications**

Security Group Name	Supported Source Application	Description	Associated Initialization Block Name
Business Group Org-Based Security	Oracle EBS, PeopleSoft HR	A business group is the highest level in the organization structure and is usually used to represent the entire enterprise or a major division. A business group can have several set of books.	Business Groups
Company Org-Based Security	PeopleSoft HR and Financials, SAP R/3	This security group filters data based on the GL or HR business units associated to the user that is logged in. The business unit is the highest level key structure in PeopleSoft. The company code is the central organizational unit of external accounting within the SAP R/3 System.	Companies
HR Org-Based Security	PeopleSoft HR	This security group filters data based on the HR organizations that the user is authorized to see. This security works in conjunction with HR Organization hierarchy to restrict users to access organizations that they are authorized to view and the	HR Organizations

		organizations that report to them.	
Human Resource Personnel Data Security	Oracle EBS, PeopleSoft HR	This security group gives HR staff access to all organizations that they are authorized to see (except for their own organization) and to the supervisors they are authorized to see based on the Supervisor (W_POSITION_DH) security.	HR Organizations User HR Organizations
Inventory Org-Based Security	Oracle EBS	An inventory organization tracks inventory transactions and balances, and/or manufactures or distributes products or components. This security group filters data based on the inventory orgs associated to the user that is logged in.	Inventory Organizations
Ledger-Based Security	Oracle EBS, PeopleSoft Financials	A ledger is essentially a reporting organization that uses a common chart of accounts, functional currency, fiscal calendar, and accounting method. This security group filters data based on the ledgers associated to the user that is logged in.	Ledgers
Operating Unit Org-Based Security	Oracle EBS, PeopleSoft Financials, Siebel CRM	This security group filters data based on the organizations associated to the user that is logged in.	Operating Unit Organizations
Payables Org-Based Security	PeopleSoft Financials, SAP R/3	This security group filters data based on the payables business units associated to the user that is logged in. The	Payables Organizations

		business unit is the highest level key structure in PeopleSoft. The company code is the Payables Org in SAP R/3.	
Primary Employee/Position Hierarchy-Based Security	Oracle EBS, PeopleSoft HR, Siebel CRM	<p>This security group allows managers to view employee data in their supervisory hierarchy, including their direct reports and those reporting up the chain of command.</p> <p>Note: Primary Employee/Position Hierarchy-Based security is not available for Siebel Service Analytics. The security available for Siebel Service Analytics is visibility granted to the primary owner organization.</p>	User Hierarchy Level
Primary Owner-Based Security	Siebel CRM	This security group filters data based on the user that is logged.	Primary Owner ID
Receivables Org-Based Security	PeopleSoft Financials, Siebel CRM,SAP R/3	<p>This security group filters data based on the receivables business units associated to the user that is logged in. The business unit is the highest level key structure in PeopleSoft. . The company code is the Receivables Org in SAP R/3.</p>	Receivables Organizations
SET ID-Based Security	PeopleSoft Financials, Oracle EBS	This security group filters data based on the Set IDs associated to the user that is logged in.	Set-ID

## Object-Level Security in Oracle BI Applications

This topic describes the object-level security features in Oracle BI Applications. It contains the following topics:

- Metadata Object-Level Security (Repository Groups)
- Metadata Object-Level Security (Presentation Services)

### Metadata Object-Level Security (Repository Groups)

Repository groups control access to metadata objects, such as subject areas, tables and columns. For example, users in a particular department can view only the subject areas that belong to their department.

Metadata object security is configured in the Oracle BI Repository, using the Oracle BI Administration Tool. The Everyone user group is denied access to each of the subject areas. Each subject area is configured to give explicit read access to selected related responsibilities. This access can be extended to tables and columns.

Note: By default in Oracle BI Applications, only permissions at the subject area level have been configured.

Note: The Siebel Communications and Financial Analytics industry applications have tables and columns that are industry-specific, and, therefore, hidden from other groups.

Oracle Business Intelligence supports hierarchies within the groups in the Oracle BI Repository. In the repository, there are certain groups that are parent groups, which define the behavior of all the child groups. Inheritance is used to let permissions ripple through to child groups. The parent groups and their purpose are shown in the following table.

**Table 3 Repository Parent Groups**

Parent Group	Permissions Inherited By
Finance	All Financial applications groups
Insurance	All Insurance applications groups
CM Genral	All Communications applications
Consumer Sector	Consumer Sector groups
Pharma	Life Sciences/Pharmaceuticals applications groups
Channel Managers	All Channel applications groups
Partner Managers	All Partner application groups

For more information about setting up and managing initialization blocks, see *Oracle Business Intelligence Server Administration Guide*.

### Metadata Object-Level Security (Presentation Services)

Oracle BI Presentation Services objects are controlled using Presentation Services groups. Access to these objects, such as dashboards and pages, reports, and Web folders, is controlled using the Presentation Services groups. Presentation Services groups are customized in the Oracle BI Presentation Services interface.

For detailed information about Presentation Services groups, see *Oracle Business Intelligence Presentation Services Administration Guide*.

## User-Level Security in Oracle BI Applications

User security concerns the authentication and confirmation of the identity of the user based on the credentials provided, such as username and password. User-level security is set up in Oracle Business Intelligence Enterprise Edition. For more information, see *Oracle Business Intelligence Server Administration Guide*.

## Extending Security in Oracle BI Applications

You can extend the preconfigured Oracle BI Applications security model to match your operational source system. The general process for extending data-level security for repository objects is described below.

1. Extend the physical table by adding the attribute by which the dimension or fact needs to be secured. (This step results in a change to the data model.)
2. Populate the relevant attribute value for each row in the fact or dimension table. (This step results in a change to the ETL mapping.)
3. Use the Oracle BI Administration Tool to create an initialization block to fetch the attribute values and populate them into a session variable when each user logs into Oracle BI Applications. You can create a target session variable for the initialization block if the initialization block is not a row-wise initialization block. (This step results in a change to the Oracle BI Repository.) For instructions, see *Oracle Business Intelligence Server Administration Guide*.
4. Use the Oracle BI Administration Tool to create a security group and filters for each of the fact and dimension tables that need to be secured by the attribute you added in Step 1. For instructions, see *Oracle Business Intelligence Server Administration Guide*.
5. Use the Oracle BI Administration Tool to add security groups that provide content or object access to users under the security group you created in Step 4. (This step results in a change to the Oracle BI Repository.) For instructions, see *Oracle Business Intelligence Server Administration Guide*.

## Integrating Data Security for Oracle EBS

This topic explains how security in Oracle BI Applications is deployed with Oracle EBS. Read this topic if you want to understand how the default security settings are configured so that you can change the way security is implemented if required. This topic contains the following topics:

- Oracle BI Applications Authorization for Oracle EBS
- Operating Unit-Based Security for Oracle EBS
- Inventory Org-Based Security for Oracle EBS
- Ledger-Based Security for Oracle EBS
- Business Group Org-Based Security for Oracle EBS
- HR Org-Based Security for Oracle EBS
- Human Resource Personnel Data Analyst Security for Oracle EBS
- Employee-Based Security for Oracle EBS

Oracle BI Applications Authorization for Oracle EBS

The authorization process of Oracle BI Applications fetches a user's responsibilities from source Oracle EBS applications, matches them with all Oracle BI Applications security groups, and determines the user's applicable object security during a user's session. The initialization block Authorization is used to fetch roles and assign the result set to a special session variable called GROUP. The initialization block SQL is the following:

```
SELECT DISTINCT 'GROUP', RESPONSIBILITY_NAME FROM
FND_USER, FND_USER_RESP_GROUPS, FND_RESPONSIBILITY_VL WHERE
FND_USER.user_id=FND_USER_RESP_GROUPS.user_id
AND FND_USER_RESP_GROUPS.RESPONSIBILITY_ID = FND_RESPONSIBILITY_
VL.RESPONSIBILITY_ID
AND FND_USER_RESP_GROUPS.RESPONSIBILITY_APPLICATION_ID = FND_
RESPONSIBILITY_VL.APPLICATION_ID AND
FND_USER_RESP_GROUPS.START_DATE < SYSDATE AND
(CASE WHEN FND_USER_RESP_GROUPS.END_DATE IS NULL THEN SYSDATE ELSE
TO_DATE(FND_USER_RESP_GROUPS.end_date) END) >= SYSDATE
AND FND_USER.user_id = (SELECT USER_ID FROM FND_USER WHERE UPPER(USER_NAME =
UPPER(':USER'))
```

## Operating Unit-Based Security for Oracle EBS

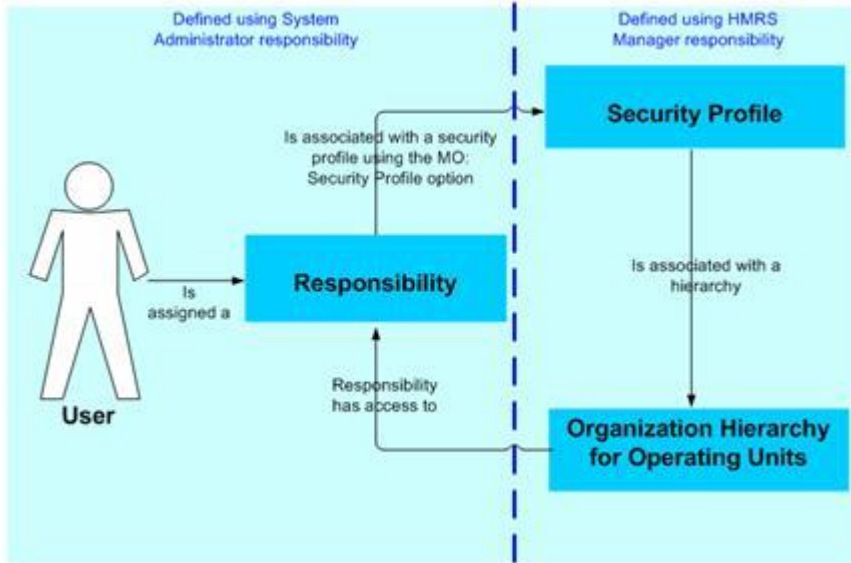
This topic contains the following topics:

- About Operating Unit-Based Security for Oracle EBS
- Implementation Steps for Operating Unit-Based Security for Oracle EBS

### About Operating Unit-Based Security for Oracle EBS

Operating units are secured by attaching a security profile to a user ID or a responsibility. In turn, a security profile is associated with an organization hierarchy, which also has access to the user ID or responsibility (see the following figure). The user ID or responsibility is defined using the System Administrator responsibility. The security profile and organization hierarchy are defined using the HRMS Manager responsibility.

*Figure 2. Operating Unit-Based Security for Oracle EBS*



Operating Unit assignment is decided by looking at the profiles set at the following levels, with the order of precedence indicated:

1. User
2. Responsibility
3. Application
4. Site

In other words, if a value is set in the profile at the user level and at the site level, the value set at the user level takes precedence.

## Implementation Steps for Operating Unit-Based Security for Oracle EBS

The sequence for operating unit-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

```
EBS_SSO_INTEGRATION_MODE
```

This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether or not Oracle BI Applications is integrated with EBS SSO.

3. The 'EBS Security Context' initialization block then populates these session variables:

```
OLTP_EBS_RESP_ID
```

The session variable is initialized with the responsibility of the user's session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise, it is defaulted to a random value, which will be ignored.

```
OLTP_EBS_RESP_APPL_ID
```

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. The Oracle BI Server will get the operating unit corresponding to the USER from FND\_USER\_RESP\_GROUPS. The following session variable is set automatically:

OU\_ORG (Row-wise variable)

The initialization block 'Operating Unit Organizations', which sets the value for this variable, is shown below.

#### **Initialization block -- 'Operating Unit Organizations'**

The initialization block 'Operating Unit Organizations' sets the value for variable OU\_ORG using the following SQL:

```
SELECT DISTINCT 'OU_ORG', TO_CHAR(PER_ORGANIZATION_LIST.ORGANIZATION_ID)
FROM PER_ORGANIZATION_LIST,
(SELECT FND_PROFILE.VALUE_SPECIFIC('XLA_MO_SECURITY_PROFILE_LEVEL',
USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID) PROFILE_ID
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID
FROM FND_USER_RESP_GROUPS
WHERE START_DATE < SYSDATE
AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_DATE(END_DATE) END)
>= SYSDATE
AND USER_ID = (SELECT USER_ID FROM FND_USER WHERE UPPER(USER_NAME) =
UPPER(':USER')))
AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)))
WHERE PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_ID UNION
SELECT DISTINCT 'OU_ORG', FND_PROFILE.VALUE_SPECIFIC('ORG_ID', USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID) ORGANIZATION_ID
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID
FROM FND_USER_RESP_GROUPS WHERE START_DATE < SYSDATE
AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_DATE(END_DATE) END)
>= SYSDATE
AND USER_ID = (SELECT USER_ID FROM FND_USER WHERE UPPER(USER_NAME) =
UPPER(':USER')))
AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
```

```
VALUEOF (NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE  
RESPONSIBILITY_APPLICATION_ID END))
```

## Inventory Org-Based Security for Oracle EBS

This topic contains the following topics:

- About Inventory Org-Based Security for Oracle EBS
- Implementation Steps for Inventory Org-Based Security for Oracle EBS

### About Inventory Org-Based Security for Oracle EBS

With inventory org-based security, the organization that a user belongs to determines which rows of data they can access. Inventory org-based security is applied based on the current logged-in responsibility rather than the current user. With Oracle EBS sources, an inventory organization can be associated with multiple responsibilities.

### Implementation Steps for Inventory Org-Based Security for Oracle EBS

The sequence for inventory org-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the following session variable is set automatically.

```
USER (System variable)
```

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

```
EBS_SSO_INTEGRATION_MODE
```

This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

3. The 'EBS Security Context' initialization block then populates these session variables:

```
OLTP_EBS_RESP_ID
```

The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

```
OLTP_EBS_RESP_APPL_ID
```

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. The Oracle BI Server will get the inventory org corresponding to the USER from FND\_USER\_RESP\_GROUPS. The following session variable is set automatically:

```
INV_ORG (Row-wise variable)
```

The initialization block 'Inventory Organizations', which sets the value for this variable, is shown below.

#### Initialization block -- 'Inventory Organizations'

The initialization block 'Inventory Organizations' sets the value for variable INV\_ORG using the following SQL:

```

SELECT DISTINCT 'INV_ORG', BIS_ORGANIZATIONS_V.ID FROM
FND_USER_RESP_GROUPS, BIS_ORGANIZATIONS_V

WHERE FND_USER_RESP_GROUPS.RESPONSIBILITY_ID = BIS_
ORGANIZATIONS_V.RESPONSIBILITY_ID

AND FND_USER_RESP_GROUPS.START_DATE < SYSDATE

AND (CASE WHEN FND_USER_RESP_GROUPS.END_DATE IS NULL THEN SYSDATE ELSE

AND FND_USER_RESP_GROUPS.USER_ID = (SELECT USER_ID FROM FND_USER WHERE
UPPER(USER_NAME) = UPPER(':USER'))

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN

VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) =

'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)

```

## Ledger-Based Security for Oracle EBS

Ledger-based security for Oracle EBS was introduced in Oracle BI Applications release 7.9.4. It replaces the company-based security to support the Oracle EBS GL set of books.

This topic contains the following topics:

- About Ledger-Based Security for Oracle EBS
- Implementation Steps for Ledger-Based Security for Oracle EBS

### About Ledger-Based Security for Oracle EBS

In Oracle EBS Release 11i, a set of books is essentially a reporting entity that defines the reporting context including a chart of accounts, a functional currency, and an accounting calendar. A set of books can be assigned to a user, a responsibility, or to the site as the default for all responsibilities. Each user is associated with a single set of books when they log in to the application under a given responsibility in Oracle Applications. Ledger-based security filters data based on the set of books associated with the user that is logged in.

In Oracle EBS Release 12, the set of books is replaced by the ledger. A ledger determines the currency, chart of accounts, accounting calendar, ledger processing options and subledger accounting method. The data access set assigned to the user's responsibility controls what ledgers the user can access. A user may be able to access multiple ledgers from a responsibility. Ledger-based security filters data based on the ledgers associated with the user that is logged in.

### Implementation Steps for Ledger-Based Security for Oracle EBS

The sequence for ledger-based security for Oracle EBS is described below:

1. When a user logs in to Oracle Business Intelligence Enterprise Edition, the session variable below is set automatically.

```
USER (System variable)
```

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

```
EBS_SSO_INTEGRATION_MODE
```

This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

3. The 'EBS Security Context' initialization block then populates these session variables:

```
OLTP_EBS_RESP_ID
```

The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

```
OLTP_EBS_RESP_APPL_ID
```

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. Then this session variable would be initialized in another init block, "Ledgers", which gets the ledgers (which is essentially the set of books in EBS) corresponding to the USER and OLTP\_EBS\_RESP\_ID and OLTP\_EBS\_RESP\_APPL\_ID, via table FND\_USER\_RESP\_GROUPS and procedure FND\_PROFILE.

Row-wise variable:

```
LEDGER (Row-wise variable)
```

5. The Oracle BI server gets the set of books or ledgers corresponding to the USER and OLTP\_EBS\_RESP\_ID from the OLTP. The 'Ledgers' initialization block then populates these session variables.

The Ledgers initialization block should be set according to the Oracle EBS release, as follows:

- If you are using EBS release 12 or after, the following SQL applies as the data source in the initialization block:

```
SELECT DISTINCT 'LEDGER', TO_CHAR(GAL.LEDGER_ID)
FROM GL_ACCESS_SET_LEDGERS GAL, (SELECT FND_PROFILE.VALUE_
SPECIFIC('GL_ACCESS_SET_ID',USER_ID, RESPONSIBILITY_ID,
RESPONSIBILITY_APPLICATION_ID) PROFILE_VALUE
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_ APPLICATION_ID
FROM FND_USER_RESP_GROUPS
WHERE START_DATE < SYSDATE AND (CASE WHEN END_DATE IS NULL THEN SYSDATE
ELSE
TO_DATE(END_DATE) END) >= SYSDATE AND USER_ID = (CASE WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_
ID FROM FND_USER WHERE
USER_NAME = 'OPERATIONS') END) AND RESPONSIBILITY_ID = (CASE WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'
```

```

THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)))
WHERE GAL.ACCESS_SET_ID = PROFILE_VALUE

```

- If you are using Oracle EBS 11i, the following SQL applies as the data source in the Ledgers initialization block:

```

SELECT DISTINCT 'LEDGER', FND_PROFILE.VALUE_SPECIFIC('GL_SET_OF_BKS_ID',
USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID)
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID
FROM
FND_USER_RESP_GROUPS
WHERE START_DATE < SYSDATE
AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_DATE(END_DATE) END)
>= SYSDATE
AND USER_ID IN (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) =
'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_ID FROM
FND_USER WHERE UPPER(USER_NAME) = UPPER(':USER')) END)
AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN
VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))

```

## Business Group Org-Based Security for Oracle EBS

This topic contains the following topics:

- About Business Group Org-Based Security for Oracle EBS
- Implementation Steps for Business Group Org-Based Security for Oracle EBS

### About Business Group Org-Based Security for Oracle EBS

A business group is the highest level in the organization structure. It is usually used to represent the entire enterprise or a major division. A business group can have several sets of books.

### Implementation Steps for Business Group Org-Based Security for Oracle EBS

The sequence for business group org-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

USER (System variable)

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

EBS\_SSO\_INTEGRATION\_MODE

This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

3. The 'EBS Security Context' initialization block then populates these session variables:

OLTP\_EBS\_RESP\_ID

The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

OLTP\_EBS\_RESP\_APPL\_ID

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. The Oracle BI Server will get the business group corresponding to the USER and OLTP\_EBS\_RESP\_ID from FND\_USER\_RESP\_GROUPS. The following session variable is set automatically:

BUSINESS\_GROUP (Row-wise variable)

The initialization block 'Business Groups', which sets the value for this variable, is shown below.

#### **Initialization block -- 'Business Groups'**

The initialization block 'Business Groups' sets value for variable INV\_ORG using the following SQL:

```
SELECT DISTINCT 'BUSINESS_GROUP',
TO_CHAR(FND_PROFILE.VALUE_SPECIFIC('PER_BUSINESS_GROUP_ID',USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID))
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_ APPLICATION_ID
FROM FND_USER_RESP_GROUPS WHERE START_DATE < SYSDATE AND (CASE WHEN
END_DATE IS NULL THEN SYSDATE ELSE TO_ DATE(END_DATE) END) >= SYSDATE AND
USER_ID = (CASE WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_ ID FROM FND_USER
WHERE UPPER(USER_NAME) = UPPER(':USER')) END)
AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_
SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))
```

**Note:** The 'Business Group Org-Based Security' security group contains all the data access permission filters.

## HR Org-Based Security for Oracle EBS

This topic contains the following topics:

- About HR Org-Based Security for Oracle EBS
- Implementation Steps for HR Org-Based Security for Oracle EBS

### About HR Org-Based Security for Oracle EBS

HR org-based security for Oracle EBS supports the standard HRMS organization security defined in Oracle EBS Human Resources. Oracle EBS Human Resources restricts access by organization, position, and payroll based on the security policies defined in the security profile.

### Implementation Steps for HR Org-Based Security for Oracle EBS

The sequence for HR org-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

`USER` (System variable)

2. The Oracle BI Server gets the HR organizations corresponding to the `USER` from the following tables:

- `FND_USER_RESP_GROUPS`
- `FND_USER`
- `PER_SECURITY_PROFILES`
- `PER_SEC_PROFILE_ASSIGNMENTS`
- `PER_PERSON_LIST`

**Note:** Before the `PER_PERSON_LIST` table can be used, you must ensure that you have run the Oracle EBS HRMS Security List Maintenance process.

- `PER_ALL_ASSIGNMENTS_F`

3. The following session variable is set automatically:

`HR_ORG` (Row-wise variable)

The initialization block 'HR Organizations', which sets the value for this variable, is shown below.

#### Initialization block -- 'HR Organizations'

The initialization block 'HR Organizations' sets value for variable `HR_ORG` using the following SQL. The actual SQL query differs depending on whether Multiple Security Group (MSG) is set up or not.

The following SQL should be used when MSG is not in place:

```
SELECT
DISTINCT 'HR_ORG'
,TO_CHAR (SEC_DET.ORGANIZATION_ID) FROM
(
SELECT
'HR_ORG', ASG.ORGANIZATION_ID
FROM
```

```

FND_USER_RESP_GROUPS URP
, FND_USER_USR
, PER_SECURITY_PROFILES PSEC
, PER_PERSON_LIST PER
, PER_ALL_ASSIGNMENTS_F ASG WHERE
URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE
TO_DATE(URP.END_
DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = UPPER(':USER') AND USR.USER_ID = URP.USER_ID
AND TRUNC(SYSDATE)
BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME) AND
PSEC.SECURITY_PROFILE_ID = FND_PROFILE.VALUE_SPECIFIC('PER_SECURITY_
PROFILE_ID', URP.USER_ID, URP.RESPONSIBILITY_ID, URP.RESPONSIBILITY_
APPLICATION_ID)
AND PER.SECURITY_PROFILE_ID = PSEC.SECURITY_PROFILE_ID AND PER.PERSON_ID =
ASG.PERSON_ID
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND
ASG.EFFECTIVE_END_DATE AND URP.RESPONSIBILITY_ID =
DECODE(FND_GLOBAL.RESP_ID,
-1, URP.RESPONSIBILITY_ID,
NULL, URP.RESPONSIBILITY_ID, FND_GLOBAL.RESP_ID)
UNION
SELECT DISTINCT 'HR_ORG', ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
FND_USER_USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID AND USR.USER_NAME = UPPER(':USER')
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND
ASG.EFFECTIVE_END_DATE AND ASG.PRIMARY_FLAG = 'Y'
) SEC_DET

```

The following SQL should be used when MSG is in place:

```

SELECT
DISTINCT 'HR_ORG', TO_CHAR(PER_ORGANIZATION_LIST.ORGANIZATION_ID)
FROM PER_ORGANIZATION_LIST,
(SELECT FND_PROFILE.VALUE_SPECIFIC('PER_BUSINESS_GROUP_ID', USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID) PROFILE_ID
FROM

```

```

(SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID FROM
FND_USER_RESP_GROUPS

WHERE START_DATE < SYSDATE

AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_DATE(END_DATE) END) >=
SYSDATE

AND USER_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' =
'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_ID FROM
FND_USER WHERE USER_NAME = UPPER(':USER')) END)

AND RESPONSIBILITY_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_
MODE)' = 'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END) AND
RESPONSIBILITY_APPLICATION_ID = (CASE WHEN

'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)

) WHERE

PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_ID

```

**Note:** The 'HR Org-Based Security' security group contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

## Human Resource Personnel Data Analyst Security for Oracle EBS

HR personnel need to see all data for the internal organizations for which they are responsible and the data for their subordinates in their own organization. The 'Human Resource Personnel Data Security' security group supports this requirement. The security mechanism for this group uses the following metadata elements:

- **HR\_ORG** variable. This variable is defined by the row-wise initialization block HR Organizations. This data set stores all the organizations the user is responsible for, plus the user's own organization, which is the same as the organization selected in USER\_HR\_ORG. The query for populating this data set is:

**Note:** The actual SQL query differs depending on whether Multiple Security Group (MSG) is set up or not.

The following SQL is used when MSG is not in place:

```

SELECT

DISTINCT 'HR_ORG', TO_CHAR(PER_ORGANIZATION_LIST.ORGANIZATION_ID) FROM
PER_ORGANIZATION_LIST,

(SELECT FND_PROFILE.VALUE_SPECIFIC('PER_BUSINESS_GROUP_ID', USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID)

PROFILE_ID FROM

```

```

(SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID FROM FND_
USER_RESP_GROUPS WHERE START_DATE < SYSDATE AND (CASE WHEN END_DATE IS NULL
THEN SYSDATE ELSE TO_DATE(END_DATE) END) >= SYSDATE
AND USER_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' =
'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_ID
FROM FND_USER WHERE USER_NAME = UPPER(':USER')) END)
AND RESPONSIBILITY_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_
MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_
INTEGRATION_MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)
) WHERE
PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_ID

```

**The following SQL is used when MSG is in place:**

```

SELECT
DISTINCT 'HR_ORG'
,TO_CHAR(SEC_DET.ORGANIZATION_ID) FROM
(
SELECT 'HR_ORG', ASG.ORGANIZATION_ID
FROM FND_USER_RESP_GROUPS URP, FND_USER USR, PER_SEC_PROFILE_ASSIGNMENTS
SASG, PER_SECURITY_PROFILES PSEC, PER_PERSON_LIST PER,
PER_ALL_ASSIGNMENTS_F ASG
WHERE URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE
TO_DATE(URP.END_DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = UPPER(':USER')
AND URP.SECURITY_GROUP_ID = SASG.SECURITY_GROUP_ID AND URP.USER_ID =
USR.USER_ID
AND TRUNC(SYSDATE)
BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME) AND
URP.USER_ID = SASG.USER_ID
AND URP.RESPONSIBILITY_ID = SASG.RESPONSIBILITY_ID
AND URP.RESPONSIBILITY_APPLICATION_ID = SASG.RESPONSIBILITY_APPLICATION_ID
AND PSEC.SECURITY_PROFILE_ID = SASG.SECURITY_PROFILE_ID
AND PSEC.SECURITY_PROFILE_ID = PER.SECURITY_PROFILE_ID AND PER.PERSON_ID =
ASG.PERSON_ID

```

```

AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND
ASG.EFFECTIVE_END_DATE AND TRUNC(SYSDATE) BETWEEN SASG.START_DATE AND
NVL(SASG.END_DATE, HR_GENERAL.END_OF_TIME)

AND URP.RESPONSIBILITY_ID = DECODE(FND_GLOBAL.RESP_ID,
-1, URP.RESPONSIBILITY_ID, NULL, URP.RESPONSIBILITY_ID, FND_GLOBAL.RESP_ID)

UNION

SELECT DISTINCT 'HR_ORG', ORGANIZATION_ID FROM PER_ALL_ASSIGNMENTS_F ASG,
FND_USER USR

WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID AND USR.USER_NAME = UPPER(':USER')

AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND
ASG.EFFECTIVE_END_DATE

AND ASG.PRIMARY_FLAG= 'Y'

) SEC_DET

```

- **USER\_HR\_ORG variable.** This variable is defined using the initialization block User HR Organizations. This variable stores the user's own organization. The query for populating this variable is:

```

SELECT DISTINCT 'USER_HR_ORG', ORGANIZATION_ID FROM PER_ALL_ASSIGNMENTS_F
ASG,

FND_USER USR

WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID AND USR.USER_NAME = UPPER(':USER')

AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND
ASG.EFFECTIVE_END_DATE

AND ASG.PRIMARY_FLAG= 'Y'

```

- **Human Resources Analyst security group.** The data filter defined for this group is the following:

```

Core."Dim - Employee Organization"."Employee Organization Number" =
VALUEOF(NQ_SESSION."HR_ORG") AND (Core."Dim - Employee
Organization"."Employee Organization Number" <>
VALUEOF(NQ_SESSION."USER_HR_ORG") OR Core."Dim - Position
Security"."Hierarchy
Based Column" = VALUEOF(NQ_SESSION."USER"))

```

This filter joins the fact table used in the report to the Employee Organization dimension to get the organization number for the employee owner of the fact record. If this organization is among the HR orgs, then it will be compared next to the user's own organization. If they are different, then there is no further check, and the record is selected. If they are the same, then an additional filter is applied based on the employee hierarchy, to make sure the employee owner of this fact record is one of the user's subordinates.

## Employee-Based Security for Oracle EBS

Employee-based security restricts data visibility of the records to the owner of that record, and all employees he or she reports to in the company's employee hierarchy. This security mechanism uses data from the data warehouse database, and shares the metadata components with other supported applications (Siebel CRM and PeopleSoft). By default, this type of security supports only HR Analytics

facts. For more information on how this security mechanism works, see About Primary Position-Based Security for Siebel CRM Industry Applications.

## Integrating Data Security for Oracle's PeopleSoft Enterprise Applications

This topic explains how data security is implemented for Oracle's PeopleSoft Enterprise Applications in Oracle BI Applications. Read this topic if you want to understand how the default security settings are configured so that you can change the way security is implemented if required. This topic contains the following topics:

- Oracle BI Applications Authorization for PeopleSoft
- Operating Unit-Based Security for PeopleSoft Financials
- Company Org-Based Security for PeopleSoft Financials and PeopleSoft HR
- Ledger-Based Security for PeopleSoft Financials
- HR Org-Based Security for PeopleSoft HR
- Payables Org-Based Security for PeopleSoft Financials
- Receivables Org-Based Security for PeopleSoft Financials
- SetID-Based Security for PeopleSoft HR and PeopleSoft Financials
- Human Resource Personnel Data Analyst Security for PeopleSoft HR

### Oracle BI Applications Authorization for PeopleSoft

The authorization process of Oracle BI Applications fetches a user's role from the source PeopleSoft application, matches the role with all Oracle BI Applications security groups, and determines the user's applicable object security during a user's session. The initialization block 'Authorization' is used to fetch roles and assign the result set to a special session variable called 'GROUP', which the Oracle BI Server then uses for matching. The initialization block SQL is the following:

```
SELECT DISTINCT
'GROUP', ROLENAME FROM
PSROLEUSER WHERE
ROLEUSER = '' :USER''
```

### Operating Unit-Based Security for PeopleSoft Financials

The sequence for operating unit-based security for PeopleSoft Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The Oracle BI Server then gets the operating units (or the general ledger business units in PeopleSoft Financials) corresponding to the USER from the following tables:

- PS\_SEC\_BU\_OPR
- PS\_BUS\_UNIT\_TBL\_GL
- PS\_INSTALLATION\_FS
- PS\_SEC\_BU\_CLS

The following session variable is set automatically:

OU\_ORG (Row-wise variable)

The initialization block 'Operating Unit Organizations', which sets the value for this variable, is shown below.

#### **Initialization block -- 'Operating Unit Organizations'**

The initialization block 'Operating Unit Organizations' sets value for variable OU\_ORG using the following SQL:

```
SELECT DISTINCT 'OU_ORG', S1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR S1, PS_BUS_UNIT_TBL_GL A, PS_INSTALLATION_FS I
WHERE S1.OPRID = '':USER''
AND S1.BUSINESS_UNIT = A.BUSINESS_UNIT AND I.SECURITY_TYPE = 'O'
AND I.BU_SECURITY = 'Y' UNION
SELECT DISTINCT 'OU_ORG', S2.BUSINESS_UNIT FROM PS_SEC_BU_CLS S2,
PS_BUS_UNIT_TBL_GL A, PS_INSTALLATION_FS I2, PSOPRDEFN P
WHERE P.OPRID = '':USER''
AND S2.BUSINESS_UNIT = A.BUSINESS_UNIT AND P.OPRCLASS = S2.OPRCLASS
AND I2.SECURITY_TYPE = 'C' AND I2.BU_SECURITY = 'Y'
```

**Note:** The 'Operating Unit Org-Based Security' security group contains all the data access permission filters.

## **Company Org-Based Security for PeopleSoft Financials and PeopleSoft HR**

The sequence for company org-based security for PeopleSoft Financials and PeopleSoft HR is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.  
USER (System variable)
2. The Oracle BI Server then gets the companies or business units corresponding to the USER from the following tables:
  - PS\_SEC\_BU\_OPR
  - PS\_BUS\_UNIT\_TBL\_GL
  - PS\_SCRTY\_TBL\_DEPT
  - PS\_BU\_DEPT\_VW
  - PS\_BUS\_UNIT\_TBL\_GL
  - PSOPRDEFN for PeopleSoft HR
  - PS\_INSTALLATION\_FS for PeopleSoft Financials
  - PSOPRDEFN for PeopleSoft Financials
  - PS\_SEC\_BU\_CLS for PeopleSoft Financials

The following session variable is set automatically:

COMPANY (Row-wise variable)

The initialization block 'Companies', which sets the value for this variable, is shown below.

### Initialization block -- 'Companies'

The initialization block 'Companies' sets value for variable COMPANY using the following SQL:

#### For PeopleSoft Financials:

```
SELECT DISTINCT 'COMPANY', S1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR S1, PS_BUS_UNIT_TBL_GL A, PS_INSTALLATION_FS I
WHERE S1.OPRID = ':USER'
AND S1.BUSINESS_UNIT = A.BUSINESS_UNIT AND I.SECURITY_TYPE = 'O'
UNION
SELECT DISTINCT 'COMPANY', S2.BUSINESS_UNIT FROM PS_SEC_BU_CLS S2,
PS_BUS_UNIT_TBL_GL A, PS_INSTALLATION_FS I2, PSOPRDEFN P
WHERE P.OPRID = ':USER'
AND S2.BUSINESS_UNIT = A.BUSINESS_UNIT AND P.OPRCLASS = S2.OPRCLASS
AND I2.SECURITY_TYPE = 'C' AND I2.BU_SECURITY = 'Y'
```

#### For PeopleSoft HR:

```
SELECT DISTINCT 'COMPANY', C.BUSINESS_UNIT
FROM PSOPRDEFN A, PS_SCRTY_TBL_DEPT B, PS_BU_DEPT_VW C, PS_BUS_UNIT_TBL_GL
D
WHERE
A.ROWSECCLASS = B.ROWSECCLASS AND B.ACCESS_CD = 'Y' AND
B.DEPTID = C.DEPTID AND C.BUSINESS_UNIT = D.BUSINESS_UNIT AND A.OPRID =
':USER'
```

**Note:** The 'Company Org-Based Security' security group contains all the data access permission filters.

## Ledger-Based Security for PeopleSoft Financials

Ledger data in PeopleSoft is reference data that is secured by and shared by business units. The Ledger table includes the SetID field and uses the TableSet feature in PeopleTool. In addition, ledger data access is controlled by row-level security, which enables you to implement security to restrict individual users or permission lists from specific rows of data that are controlled by the ledger. Ledger-based security filters data based on the ledgers associated with the user that is logged in.

When you set up ledger-based security for a PeopleSoft application, you should also set up the company org-based security for PeopleSoft. Ledger-based security does not automatically restrict the data by the GL business unit.

The sequence for ledger-based security for PeopleSoft Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

USER (System variable)

2. The Oracle BI Server gets the ledgers corresponding to the USER from the following tables:

- PS\_LED\_DEFN\_TBL
- PS\_INSTALLATION\_FS
- PS\_SEC\_LEDGER\_CLS
- PS\_LED\_GRP\_TBL
- PSOPRDEFN
- PSROLEUSER
- PSROLECLASS

The following session variable is set automatically:

LEDGER (Row-wise variable)

The initialization block 'Ledgers', which sets the value for this variable, is set as follows.

```
SELECT DISTINCT 'LEDGER', LG.SETID || SO.LEDGER
FROM PS_SEC_LEDGER_OPR SO, PS_LED_DEFN_TBL LG, PS_INSTALLATION_FS IFS
WHERE SO.LEDGER = LG.LEDGER AND IFS.SECURITY_TYPE = 'O' AND
IFS.LEDGER_SECURITY = 'Y' AND SO.OPRID = ':USER' UNION
SELECT distinct 'LEDGER', LG.SETID || SC.LEDGER
FROM PS_SEC_LEDGER_CLS SC, PS_LED_GRP_TBL LG, PSOPRDEFN OP, PSROLEUSER ORL,
PSROLECLASS RCL, PS_INSTALLATION_FS IFS
WHERE SC.LEDGER_GROUP = LG.LEDGER_GROUP AND SC.OPRCLASS = RCL.CLASSID AND
OP.OPRID = ORL.ROLEUSER
AND ORL.ROLENAM = RCL.ROLENAM and IFS.SECURITY_TYPE = 'C' AND
IFS.LEDGER_SECURITY = 'Y' AND OP.OPRID = ':USER'
```

**Note:** The 'Ledger-Based Security' security group contains all the data access permission filters.

## HR Org-Based Security for PeopleSoft HR

HR org-based security for PeopleSoft HR supports the PeopleSoft department security by tree.

The sequence for HR org-based security with PeopleSoft HR is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

USER (System variable)

2. The Oracle BI Server gets the HR business units corresponding to the USER from the following tables:

- PSOPRDEFN
- PS\_SCRTY\_TBL\_DEPT

The following session variable is set automatically:

HR\_ORG (Row-wise variable)

The initialization block 'HR Organizations', which sets the value for this variable, is shown below.

### Initialization block -- 'HR Organizations'

The initialization block 'HR Organizations' sets value for variable HR\_ORG using the following SQL:

```
SELECT 'DEPT_ID', DEPT.DEPTID FROM
```

```

PS_DEPT_TBL DEPT, PSOPRDEFN OPR WHERE DEPT.EFFDT = (
SELECT MAX(DEPT1.EFFDT)
FROM PS_DEPT_TBL DEPT1
WHERE DEPT1.SETID = DEPT.SETID AND DEPT1.DEPTID = DEPT.DEPTID) AND (EXISTS (
SELECT 'X'
FROM PS_SJT_DEPT SEC, PS_SJT_CLASS_ALL CLS, PS_SJT_OPR_CLS SOC
WHERE SEC.SETID = DEPT.SETID
AND SEC.DEPTID = DEPT.DEPTID
AND CLS.SCRTY_SET_CD = 'PPLJOB' AND CLS.SCRTY_TYPE_CD = '001' AND CLS.TREE
= 'Y'
AND CLS.SCRTY_KEY1 = SEC.SCRTY_KEY1
AND CLS.SCRTY_KEY2 = SEC.SCRTY_KEY2
AND CLS.SCRTY_KEY3 = SEC.SCRTY_KEY3
AND SOC.OPRID = OPR.OPRID
AND SOC.CLASSID = CLS.CLASSID
AND SOC.CLASSID = OPR.ROWSECCLASS AND SOC.SEC_RSC_FLG <> '2' )
OR EXISTS ( SELECT 'X'
FROM PS_SJT_DEPT SEC,
PS_SJT_CLASS_ALL CLS, PS_SJT_OPR_CLS SOC
WHERE SEC.SETID = DEPT.SETID
AND SEC.DEPTID = DEPT.DEPTID AND CLS.SCRTY_SET_CD = 'DEPT'
AND CLS.SCRTY_TYPE_CD = SEC.SCRTY_TYPE_CD
AND CLS.SCRTY_KEY1 = SEC.SCRTY_KEY1
AND CLS.SCRTY_KEY2 = SEC.SCRTY_KEY2
AND CLS.SCRTY_KEY3 = SEC.SCRTY_KEY3
AND CLS.TREE = 'Y'
AND SOC.OPRID = OPR.OPRID
AND SOC.CLASSID = CLS.CLASSID
AND SOC.CLASSID = OPR.ROWSECCLASS AND SOC.SEC_RSC_FLG <> '2' )
OR EXISTS (
SELECT 'X'
FROM PS_SJT_DEPT SEC, PS_SJT_CLASS_ALL CLS, PS_SJT_OPR_CLS SOC
WHERE SEC.SETID = DEPT.SETID AND SEC.DEPTID = DEPT.DEPTID AND
CLS.SCRTY_SET_CD = 'DEPT'
AND CLS.SCRTY_TYPE_CD = SEC.SCRTY_TYPE_CD AND CLS.SCRTY_KEY1 =
SEC.SCRTY_KEY1

```

```

AND CLS.SCRTY_KEY2 = SEC.SCRTY_KEY2
AND CLS.SCRTY_KEY3 = SEC.SCRTY_KEY3
AND CLS.TREE = 'N'
AND SOC.OPRID = OPR.OPRID
AND SOC.CLASSID = CLS.CLASSID)) AND OPR.OPRID = ':USER'

```

**Note:** The 'HR Org-Based Security' security group contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

## Payables Org-Based Security for PeopleSoft Financials

The sequence for payables org-based security for PeopleSoft Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The Oracle BI Server gets the Payables business units corresponding to the USER from the following tables:

- PSOPRDEFN
- PS\_SEC\_BU\_OPR
- PS\_SEC\_BU\_CLS
- PS\_INSTALLATION\_FS
- PS\_BUS\_UNIT\_TBL\_AP

The following session variable is set automatically:

```
PAYABLES_ORG (Row-wise variable)
```

The initialization block 'Payables Organizations', which sets the value for this variable, is shown below.

### Initialization block -- 'Payables Organizations'

The initialization block 'Payables Organizations' sets value for variable PAYABLES\_ORG using the following SQL:

```

SELECT DISTINCT 'PAYABLES_ORG', s1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR s1, PS_BUS_UNIT_TBL_AP a, PS_INSTALLATION_FS i
WHERE s1.OPRID = ':USER'
AND s1.BUSINESS_UNIT = a.BUSINESS_UNIT AND i.SECURITY_TYPE = 'O'
AND i.BU_SECURITY = 'Y' UNION
SELECT DISTINCT 'PAYABLES_ORG', s2.BUSINESS_UNIT
FROM PS_SEC_BU_CLS s2, PS_BUS_UNIT_TBL_AP a, PS_INSTALLATION_FS i2,
PSOPRDEFN p
WHERE p.OPRID = ':USER'
AND s2.BUSINESS_UNIT = a.BUSINESS_UNIT AND p.OPRCLASS = s2.OPRCLASS

```

```
AND i2.SECURITY_TYPE = 'C' AND i2.BU_SECURITY = 'Y'
```

**Note:** The 'Payables Org-Based Security' security group contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

## Receivables Org-Based Security for PeopleSoft Financials

The sequence for receivables org-based security for PeopleSoft Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The Oracle BI Server gets the Receivables business units corresponding to the USER from the following tables:

- PS\_SEC\_BU\_OPR
- PS\_SEC\_BU\_CLS
- PS\_INSTALLATION\_FS
- PS\_BUS\_UNIT\_TBL\_AR

The following session variable is set automatically:

```
RECEIVABLES_ORG (Row-wise variable)
```

The initialization block 'Receivables Organizations', which sets the value for this variable, is shown below.

### Initialization block -- 'Receivables Organizations'

The initialization block 'Receivables Organizations' sets value for variable RECEIVABLES\_ORG using the following SQL:

```
SELECT DISTINCT 'RECEIVABLES_ORG', s1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR s1, PS_BUS_UNIT_TBL_AR a, PS_INSTALLATION_FS i
WHERE s1.OPRID = ':USER'
AND s1.BUSINESS_UNIT = a.BUSINESS_UNIT AND i.SECURITY_TYPE =
'0'
AND i.BU_SECURITY = 'Y' UNION
SELECT DISTINCT 'RECEIVABLES_ORG', s2.BUSINESS_UNIT
FROM PS_SEC_BU_CLS s2, PS_BUS_UNIT_TBL_AR a, PS_INSTALLATION_FS i2,
PSOPRDEFN p
WHERE p.OPRID = ':USER'
AND s2.BUSINESS_UNIT = a.BUSINESS_UNIT AND p.OPRCLASS =
s2.OPRCLASS AND i2.SECURITY_TYPE = 'C' AND i2.BU_SECURITY = 'Y'
```

**Note:** The 'Receivables Org-Based Security' security group contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

## SetID-Based Security for PeopleSoft HR and PeopleSoft Financials

The sequence for SetID-based security for PeopleSoft Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The Oracle BI Server gets the SetIDs corresponding to the USER from the following tables:

- PS\_SEC\_SETID\_OPR
- PS\_SEC\_SETID\_CLS
- PS\_INSTALLATION\_FS
- PSOPRDEFN

The following session variable is set automatically:

```
SET_ID (Row-wise variable)
```

The initialization block 'Set ID' sets value for variable SET\_ID using the following SQL:

### For PeopleSoft Financials:

```
SELECT DISTINCT 'SET_ID', s1.SETID
FROM PS_SEC_SETID_OPR s1, PS_INSTALLATION_FS i
WHERE s1.OPRID = ':USER' AND i.SECURITY_TYPE = 'O' AND i.SETID_SECURITY =
'Y' UNION
SELECT DISTINCT 'SET_ID', s2.SETID
FROM PS_SEC_SETID_CLS s2, PS_INSTALLATION_FS i2, PSOPRDEFN p
WHERE p.OPRID = ':USER'
AND p.OPRCLASS = s2.OPRCLASS AND i2.SECURITY_TYPE = 'C' AND
i2.SETID_SECURITY = 'Y'
```

**Note:** The 'Set ID-Based Security' security group contains all the data access permission filters.

## Human Resource Personnel Data Analyst Security for PeopleSoft HR

HR personnel need to see all data for the internal organizations for which they are responsible for and the data for their subordinates in their own organization. The 'Human Resource Personnel Data Security' security group supports this requirement. The security mechanism for this group uses the following metadata elements:

- HR\_ORG variable. This variable is defined by the row-wise initialization block HR Organizations. This data set stores all the organizations the user is responsible for, plus the user's own organization, which is the same organization as the one selected in USER\_HR\_ORG. The query for populating this data set is the following:

```
SELECT DISTINCT
'HR_ORG', C.BUSINESS_UNIT FROM
PSOPRDEFN A, PS_SCRTY_TBL_DEPT B, PS_BU_DEPT_VW C, PS_BUS_UNIT_TBL_HR D
WHERE
A.ROWSECCLASS = B.ROWSECCLASS AND
```

```

B.ACCESS_CD = 'Y' AND B.DEPTID = C.DEPTID AND C.BUSINESS_UNIT =
D.BUSINESS_UNIT AND A.OPRID = ':USER'

UNION

SELECT DISTINCT 'HR_ORG', FINAL_JOB.BUSINESS_UNIT FROM (
SELECT X.EMPLID, MAX(X.BUSINESS_UNIT) BUSINESS_UNIT FROM (
SELECT A.EMPLID, A.EMPL_RCD, A.EFFDT, EFFSEQ, A.JOB_
INDICATOR, A.EMPL_STATUS, A.BUSINESS_UNIT
FROM PS_JOB A,
    (SELECT EMPLID, MAX(EFFDT) MAX_EFFDT FROM PS_JOB
WHERE
JOB_INDICATOR = 'P' AND EMPL_STATUS IN ('A', 'L', 'P', 'W') GROUP BY
EMPLID) B
WHERE
A.EMPLID = B.EMPLID
AND A.EFFDT = B.MAX_EFFDT
AND A.JOB_INDICATOR = 'P' AND A.EMPL_STATUS IN ('A', 'L',
'P', 'W')
AND A.EFFSEQ = (SELECT MAX (C.EFFSEQ) FROM PS_JOB C
WHERE
C.EMPLID = A.EMPLID AND C.EMPL_RCD = A.EMPL_RCD AND C.EFFDT = A.EFFDT AND
C.JOB_INDICATOR = 'P' AND C.EMPL_STATUS IN ('A', 'L', 'P',
'W'))
) X
GROUP BY X.EMPLID
) FINAL_JOB, PSOPRDEFN WHERE
FINAL_JOB.EMPLID = PSOPRDEFN.EMPLID AND PSOPRDEFN.OPRID = ':USER'

```

- **USER\_HR\_ORG** variable. This variable is defined using the initialization block User HR Organizations. This variable stores the user's own organization. The query for populating this variable is the following:

```

SELECT DISTINCT FINAL_JOB.BUSINESS_UNIT FROM (
SELECT X.EMPLID, MAX(X.BUSINESS_UNIT) BUSINESS_UNIT FROM (
SELECT A.EMPLID, A.EMPL_RCD, A.EFFDT, EFFSEQ, A.JOB_INDICATOR,
A.EMPL_STATUS, A.BUSINESS_UNIT
FROM PS_JOB A,
    (SELECT EMPLID, MAX(EFFDT) MAX_EFFDT FROM PS_JOB
WHERE

```

```

JOB_INDICATOR = 'P' AND EMPL_STATUS IN ('A', 'L', 'P', 'W') GROUP BY
EMPLID) B

WHERE

A.EMPLID = B.EMPLID

AND A.EFFDT = B.MAX_EFFDT

AND A.JOB_INDICATOR = 'P' AND A.EMPL_STATUS IN ('A', 'L',
'P', 'W')

AND A.EFFSEQ = (SELECT MAX (C.EFFSEQ) FROM PS_JOB C

WHERE

C.EMPLID = A.EMPLID AND C.EMPL_RCD = A.EMPL_RCD AND C.EFFDT = A.EFFDT AND
C.JOB_INDICATOR = 'P' AND C.EMPL_STATUS IN ('A', 'L', 'P',
'W'))

) X

GROUP BY X.EMPLID

) FINAL_JOB, PSOPRDEFN WHERE

FINAL_JOB.EMPLID = PSOPRDEFN.EMPLID AND PSOPRDEFN.OPRID = ':USER'

```

- **Human Resources Analyst security group.** The data filter defined for this group is the following:

```

Core."Dim - Employee Organization"."Employee Organization Number" =
VALUEOF(NQ_SESSION."HR_ORG") AND (Core."Dim - Employee
Organization"."Employee Organization Number" <>
VALUEOF(NQ_SESSION."USER_HR_ORG") OR Core."Dim - Position
Security"."Hierarchy Based Column" = VALUEOF(NQ_SESSION."USER"))

```

This filter joins the fact table used in the report to the Employee Organization dimension to get the organization number for the employee owner of the fact record. If this organization is among the HR orgs, then it will be compared next to the user's own organization. If they are different, then there is no further check, and the record is selected. If they are the same, then an additional filter is applied based on the employee hierarchy, to make sure the employee owner of this fact record is one of the user's subordinates.

## Employee-Based Security for PeopleSoft

Employee-based security restricts data visibility of the records to the owner of that record, and all employees he or she reports to in the company's employee hierarchy. This security mechanism uses data from the Oracle Business Analytics Warehouse database, and shares the metadata components with other supported applications (for example, Oracle EBS, Siebel CRM, or PeopleSoft). By default, this type of security supports only HR Analytics facts. For more information about how this security mechanism works, see About Primary Position-Based Security for Siebel CRM Industry Applications.

## Integrating Data Security for Oracle's Siebel CRM Applications

This topic explains how data security in Oracle BI Applications is deployed with Siebel CRM. Read this topic if you want to understand how the default security settings are configured so that you can change the way security is implemented if required.

**Note:** Primary Employee/Position Hierarchy-Based security is not available for Siebel Service Analytics. The security available for Siebel Service Analytics is visibility granted to the primary owner organization.

This topic contains the following topics:

- About Primary Position-Based Security
- About Primary Position-Based Security for Siebel CRM Industry Applications
- About Partner Analytics Security Settings
- About Usage Accelerator Analytics Security Settings
- About Primary Owner-Based Security
- About Business Unit-Based Security

## About Primary Position-Based Security

This topic covers primary position-based security. It contains the following topics:

- Introduction
- Primary Employee/Position Hierarchy-Based Security Group
- Configuring Oracle BI Repository Table Joins for Primary Employee/Position Hierarchy-Based Security

### Introduction

Primary position-based security restricts data visibility for a fact or dimension record to the primary owner of this record and those above him in the hierarchy. The primary owner of a record could be a position or an employee. Primary position-based security uses a flattened hierarchy table called `W_POSITION_DH`, which is based on `W_POSITION_D` and is treated as a slowly changing dimension.

For Siebel CRM-based data, `W_POSITION_D` is populated from the Position table in Siebel CRM. A new record is created for the same position every time a new employee is associated with this position as the primary employee.

Consequently, every record in the source tables can be represented by more than one record in `W_POSITION_DH`, but only one record can have the value of `CURRENT_FLG` as 'Y' at any time. The `W_POSITION_DH` table also contains one set of columns prefixed with `CURRENT`, and another set of columns not prefixed with `CURRENT`. The columns that are prefixed with `CURRENT` reflect the current hierarchy structure for the position or employee record at any time. The columns that are not prefixed with `CURRENT` reflect the hierarchy structure for the same position or employee record during the period between `EFFECTIVE_START_DT` and `EFFECTIVE_END_DT`. This latter set of columns is used to enable fact records to be visible to the owner of a record and his upper level managers at the time the record was created, even after he changes position or managers in the company hierarchy.

Facts join to this dimension by the record owner; for example, `W_REVN_F` is joined using `PR_POSITION_DH_WID`, where `PR_POSITION_DH_WID` is the primary position on the revenue line in the source application.

### Primary Employee/Position Hierarchy-Based Security Group

This security group uses the following metadata elements in the repository:

- `HIER_LEVEL` session variable. This variable is populated by the initialization block 'User Hierarchy Level' using the SQL below. For a description of the User Hierarchy Level initialization block, see Initialization Blocks Used for Data-Level Security in Oracle BI Applications.

```
Select round(FIXED_HIER_LEVEL) FROM VALUEOF(OLAPTBO).W_POSITION_DH WHERE
BASE_LOGIN= ':USER' AND CURRENT_FLG='Y'
```

The HIER\_LEVEL value can be a number between 0 and 17. It designates the current Fixed Hierarchy level of the user in the company hierarchy. The Company hierarchy is based on the Employee hierarchy tree for Oracle EBS and PeopleSoft applications and on the Position hierarchy tree for Siebel Applications. For example, the CEO of the company is the only employee whose HIER\_LEVEL takes the value 17, if the employee hierarchy is a full tree.

- Dim - Position Security logical dimension. This logical dimension is joined to the supported fact tables. It is defined on the physical table W\_POSITION\_DH.
- Hierarchy-Based Column logical column. This column is a logical column in the Dim - Position Security logical dimension. It is defined as follows:

```
"INDEXCOL(VALUEOF(NQ_SESSION."HIER_LEVEL"), "Core"."Dim - Position
Security"."Current Base Level Login", "Core"."Dim - Position
Security"."Current Level 1 Login", "Core"."Dim - Position
Security"."Current Level 2 Login", "Core"."Dim - Position
Security"."Current Level 3 Login", "Core"."Dim - Position
Security"."Current Level 4 Login", "Core"."Dim - Position
Security"."Current Level 5 Login", "Core"."Dim - Position
Security"."Current Level 6 Login", "Core"."Dim -
Position Security"."Current Level 7 Login", "Core"."Dim - Position
Security"."Current Level 8 Login", "Core"."Dim - Position
Security"."Current Level 9 Login", "Core"."Dim - Position
Security"."Current Level 10 Login", "Core"."Dim - Position
Security"."Current Level 11 Login", "Core"."Dim - Position
Security"."Current Level 12 Login", "Core"."Dim - Position
Security"."Current Level 13 Login", "Core"."Dim - Position
Security"."Current Level 14 Login", "Core"."Dim - Position
Security"."Current Level 15 Login", "Core"."Dim - Position
Security"."Current Level 16 Login", "Core"."Dim - Position
Security"."Current Top Level Login")".
```

- The IndexCol function in this definition makes the Hierarchy-Based Column default to one of the logical columns in the list based on the value of HIER\_LEVEL. So, if the value of HIER\_LEVEL is 0, the new column will default to the first column in the list, and so on.
- A filter in the security group 'Primary Employee/Position Hierarchy-Based Security' defined as follows: ("Core"."Dim - Position Security"."Hierarchy Based Column" = VALUEOF(NQ\_SESSION."USER")).

A user needs to be a member of the security group 'Primary Employee/Position Hierarchy-Based Security', through one of his responsibilities (for Siebel and Oracle EBS applications) and Roles (for PeopleSoft applications), for the data security filters to apply. Users are assigned to this security group based on their responsibilities, using the Authorization initialization block, as described in the topic: Initialization Blocks Used for Data-Level Security in Oracle BI Applications. By default, this initialization block is populated using the following SQL:

```
Select 'GROUP', R.NAME
from VALUEOF(TBO).S_RESP R, VALUEOF(TBO).S_PER_RESP P, VALUEOF(TBO).S_USER
U
```

```

where U.LOGIN=Upper(':USER') and U.ROW_ID=P.PER_ID and
P.RESP_ID=R.ROW_ID UNION

select 'GROUP', CASE VALUEOF(NQ_SESSION.HIER_LEVEL) WHEN 0 THEN 'Hierarchy
Level (Base)'

when 1 then 'Hierarchy Level 1' when 2 then 'Hierarchy Level 2' when 3 then
'Hierarchy Level 3' when 4 then 'Hierarchy Level 4' when 5 then 'Hierarchy
Level 5' when 8 then 'Hierarchy Level 8' when 6 then 'Hierarchy Level 6'
when 7 then 'Hierarchy Level 7' when 8 then 'Hierarchy Level 8' when 9 then
'Hierarchy Level 9'

when 10 then 'Hierarchy Level 10'

when 11 then 'Hierarchy Level 11' when 12 then 'Hierarchy Level 12' when 13
then 'Hierarchy Level 13' when 14 then 'Hierarchy Level 14' when 15 then
'Hierarchy Level 15' when 16 then 'Hierarchy Level 16' When 17 then
'Hierarchy Level (Top)'

ELSE'NOGROUP'END from VALUEOF(TBO).S_DUAL

```

The first part of this SQL selects the user's responsibilities from the Siebel CRM application. The user will be assigned automatically to the security groups with the same name in the Oracle BI Repository.

The second part of this SQL assigns the user to one of the Oracle BI-specific security groups, such as Hierarchy Level (Base), Hierarchy Level 1 through 16, and Hierarchy Level (Top), based on the variable HIER\_LEVEL. These security groups are not used for data security purposes; they are used for Presentation column purposes, in conjunction with the Web Choose function defined in some reports. The purpose of this function is to allow a multi-user report to show different position columns to the user, based on his hierarchy level. This is very similar to the IndexCol function described in the topic: Primary Employee/Position Hierarchy-Based Security Group.

## **Configuring Oracle BI Repository Table Joins for Primary Employee/Position Hierarchy-Based Security**

The procedures below provide instructions for adding primary position-based security to a new dimension or fact table. The following procedures use the W\_AGREE\_D (Agreement) dimension as an example.

### **To add primary position-based security to a dimension table**

1. In the Physical layer of the Oracle BI Server Administration Tool, create an alias on W\_POSITION\_DH specifically to join to the underlying physical table.
2. Configure the join in the physical layer.
3. In the Business Model layer of the Server Administration Tool, add the W\_POSITION\_DH alias to the dimension's logical table source.
4. Add new logical columns CURRENT\_BASE\_LOGIN, CURRENT\_LVL1ANC\_LOGI, and so on, to the logical table, and map them to the corresponding physical columns.
5. Add the Hierarchy column 'Hierarchy Based Column.'
6. In the Server Administration Tool, open the Security Manager by selecting Tools and then Security from the menu bar.

- a. Right-click the group 'Primary Employee/Position Hierarchy-Based Security,' and choose Properties.
- b. In the Properties dialog box, click Permissions, and select the Filter tab.
- c. To add a new filter, click Add.
- d. In the new dialog box, select the Business Model tab, and find the logical table Dim - Agreement. A new record will be added to the list of filters automatically.
- e. Click on the ellipsis box, and add the filter condition "Core"."Dim - Customer"."Hierarchy Based Login" = VALUEOF(NQ\_SESSION."USER") in the Security Filter Expression Builder.
- f. Click OK.

**To add primary position-based security support to a fact table**

1. In the Physical layer of the Oracle BI Server Administration Tool, join the underlying physical table to Dim\_W\_POSITION\_DH\_Position\_Hierarchy. This assumes you already created the appropriate foreign key in the fact table and populated it correctly.
2. Join the logical table to the Dim - Position Security.
3. In the Server Administration Tool, open the Security Manager by selecting Tools and then Security from the menu bar.
  - a. Right-click the group 'Primary Employee/Position Hierarchy-based Security', and choose Properties.
  - b. In the Properties dialog, click Permissions, and select the Filter tab.
  - c. To add a new filter, click Add.
  - d. In the new dialog box, select the Business Model tab, and find the logical table: Dim - Agreement. A new record will be added to the list of filters automatically.
  - e. Click on the ellipsis box, and add the condition "Core"."Dim - Position Security"."Hierarchy Based Column" = VALUEOF(NQ\_SESSION."USER") in the Security Filter Expression Builder and click OK.

**About Primary Position-Based Security for Siebel CRM Industry Applications**

This topic covers primary position-based security for CRM Industry Applications. It contains the following topics:

- Consumer Sector Analytics Security Settings
- Communications, Media, and Energy (CME) Analytics Security Settings
- Financial Services Analytics Security Settings
- Pharma Sales Analytics and Pharma Marketing Analytics Security Settings

**Consumer Sector Analytics Security Settings**

The following tables describes the consumer sector responsibilities associated with each CS Dashboard.

**Table 4 Consumer Sector Responsibilities Associated with Each CS Dashboard**

Responsibility	Dashboard	Pages
VP Sales	VP Sales	Business Overview, Product

		Overview
Key Account Manger	Sales Performance	Sales Volume Planning, Hierarchy, Trends, Growth
	Promotion	Plan Year to Date, Corporate
	Key Account Manager	Business, Category
	Promotion	Plan year to date, Key account
	Funds	Account
	Retail Audit	Last Audit, Trends
	Sales Performance	Sales Volume Planning, Hierarchy, Trends, Growth

## Communications, Media, and Energy (CME) Analytics Security Settings

Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics, Oracle Communications, Media and Energy Service Analytics, Oracle Communications, Media and Energy Marketing Analytics) use the Siebel operational applications security model; that is, it uses Siebel operational applications responsibilities (and corresponding repository and Presentation Services groups) for controlling access to Siebel operational applications objects (both metadata and Presentation Services objects). This security model is described in the topic: About Security in Oracle BI Applications.

In addition to responsibilities provided by the operational applications, Oracle Communications, Media, and Energy (CME) provides additional responsibilities, and responsibility-specific security, as indicated in the following table.

**Table 5 CME Responsibilities Associated with Each CME Dashboard**

CME Responsibility	CME Dashboard	Dashboard Pages
CM Marketing Analytics User	Loyalty Management	<ul style="list-style-type: none"> <li>Customer Lifetime Value</li> <li>Churn Propensity</li> </ul>
CM Marketing Analytics Administrator		<ul style="list-style-type: none"> <li>Selling Propensity</li> <li>Financial Risk</li> <li>Actual Churn</li> </ul>
CM Sales Analytics User	Revenue Management	<ul style="list-style-type: none"> <li>Sales Portal</li> <li>Service Activations</li> </ul>
CM Sales Analytics Administrator	Account Management	<ul style="list-style-type: none"> <li>Service Modifications</li> <li>Service Disconnections</li> <li>Sales Portal</li> <li>Service Activations</li> <li>Service Modifications</li> <li>Service Disconnections</li> </ul>

CM Service Analytics User

Account Management

- Trouble Tickets
- Customer Satisfaction

CM Service Analytics Administrator

## Financial Services Analytics Security Settings

The following applications use the Siebel operational applications security model:

- The Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics).
- The Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics).

In addition to responsibilities provided by the Siebel operational applications, these applications provide additional responsibilities, and responsibility-specific security, as indicated in Table 6.

For the Financial Services products, the Siebel operational applications security model has been extended in the following ways:

- **Financial Analytics user**

A finance-specific responsibility (and corresponding repository and Presentation Services group) that must be used in conjunction with Siebel operational applications responsibilities and groups to control access to Finance-specific objects in Financial Analytics.

- A user in the Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics)

An insurance-specific responsibility (and corresponding repository and Presentation Services group) that must be used to control access to the Insurance and Healthcare-specific objects in Insurance and the Healthcare Analytics family of products (Healthcare Sales Analytics, Healthcare Service Analytics, Healthcare Marketing Analytics, Healthcare Partner Manager Analytics).

For example, when you give a salesperson all horizontal Sales responsibilities and also include the finance responsibility Financial Analytics User, this user is able to see, in addition to all horizontal sales objects (dashboards, subject areas, folders in the Presentation layer, and so on), all finance-specific Sales objects. Similarly, in order to see Insurance and Healthcare-specific objects, you need to add one of the Insurance Analytics families of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics) user responsibilities to this user.

**Parent and Child Group Behavior** Oracle BI Applications supports hierarchies in the repository groups, and certain groups within the Oracle BI Repository are parent groups that define the behavior of all the child groups. For Financial Services Analytics, the parent groups are the following:

- Finance

Parent group for all Financial applications groups. Financial Analytics User is a child group of Finance group.

- Insurance

Parent group for all Insurance applications groups. Insurance Analytics User is a child group of Insurance group.

Inheritance is used to let permissions ripple through to child groups. The parent groups for Financial Services and their purpose are shown in the following table.

**Note:** A Financial Services Analytics user is provided as a child to both Finance and Insurance. Therefore, this user has permissions available to both Finance and Insurance. If you have purchased both Financial Analytics and one of the Oracle Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics), you should use the Financial Services Analytics user responsibilities to view all relevant dashboards.

Table 6 shows the additional responsibilities, and responsibility-specific security in Oracle's Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics), the Oracle Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics), and the Oracle Healthcare Analytics family of products (Healthcare Sales Analytics, Healthcare Service Analytics, Healthcare Marketing Analytics, Healthcare Partner Manager Analytics).

If you are also deploying Usage Accelerator, Financial Services-specific Usage Accelerator responsibilities are shown in the following table.

**Table 6. Financial Services Responsibility Required to View FS Dashboards**

FS Responsibilities	Dashboards
Financial Analytics User	Credit
	Credit Card
	Private Banking
	Consumer Banking
	Corporate and Commercial Banking
	Investment Holdings
	Separate Account Management
	Wealth Management
	Institutional Sales
	Investment Banking
	Finance Marketing
	Finance Executive
	User in one of the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics)
Policy Service	
Insurance Marketing	
Insurance Executive	
Insurance Claims	
	Helath-Plan Sales

Health Plan Service  
Health Plan Marketing  
Health Plan Executive  
Insurance Agents / Partners

## Pharma Sales Analytics and Pharma Marketing Analytics Security Settings

Data-level security in Pharma Sales Analytics and Pharma Marketing Analytics is based on the Siebel position ID for all Pharma Analytics responsibilities except PH Executive Analytics. The Siebel position ID is always resolved through the fact table.

Data visibility is unconstrained for administrative roles. For other roles, data visibility is controlled by the position ID. The Oracle Business Analytics Warehouse uses the table W\_POSITION\_DH for user position-based security control. A user sees only the data that is available to that user's positions. This security model is enforced for all queries, with the exception of queries that deal exclusively with dimension data, such as:

- Time period
- Product
- Invitee status

The following table shows Pharma Analytics responsibilities and functions.

**Table 7. Pharma Analytics Responsibilities and Functions**

Responsibility	Use
LS Administrator	Administrator privileges to all options on Pharma Analytics.
PH Call Activity Analytics Admin	Administrator privileges to Call Activity Analytics option.
PH EMEA Call Activity Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.  Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Executive Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.  Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment

pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.

PH EMEA Marketing Analytics User

Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.

Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.

PH EMEA Sales Analytics User

Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.

Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.

PH Executive Analytics Admin

Unrestricted access to all Pharma Analytics options with ZIP territories.

PH Marketing Analytics Administrator

Administrator privileges to Pharma ROI, Call Activity Profit & Loss Report, Pharma Promotional Effectiveness Subject Area, and Medical Education Effectiveness Subject Area.

PH Medical Education Analytics Admin

Administrator privileges to Medical Education Analytics option.

PH Medical Education Analytics User

Enables access to Medical Education Analytics option.

PH Disconnected Analytics Admin

Administrator privileges to the PH Disconnected Manager Analytics User and Sales Rep Analytics Dashboards.

PH Disconnected Analytics User

Enables the Pharma Disconnected Analytics Home Page. Allows access to Sales Rep Dashboard as part of the Sales Rep Analytics option.

PH Disconnected Manager Analytics Admin

Administrator privilege to the PH Disconnected Manager Analytics User and District Manager Analytics Dashboards.

PH Disconnected Manager Analytics User	Enables the Pharma Disconnected Analytics Home Page. Allows access to the District Manager Dashboard as part of the Sales Rep Analytics option.
PH Sales Analytics Administrator	Administrator privileges to Rx Sales Analytics option.
PH US Call Activity Analytics User	Enables access to Call Activity Analytics Option for ZIP territory alignments.
PH US Executive Analytics User	Unrestricted access to all Pharma Disconnected Analytics options with ZIP-based territories.

## About Partner Analytics Security Settings

Oracle Partner Analytics incorporates the concept of role-based analytics. Role-based analytics provides brand owners the ability to display dashboards and pages to users based on their specific roles. For example, a sales manager would have the ability to view dashboards related to pipeline and sales effectiveness, whereas the marketing manager would have the ability to view dashboards related to campaigns. Oracle Partner Analytics also includes flexible security mechanisms to control access to subject areas and to data.

Oracle Partner Analytics roles map to Siebel responsibilities in the Siebel operational application. This topic describes the roles and associated dashboards and pages for both Partner Manager and Partner Portal applications. It also includes subject area and data-level security settings for responsibilities.

## PRM Partner Portal Role-Based Interactive Dashboards Mapping

The dashboard and page tab mapping for specific responsibilities in the PRM Partner Portal application are shown in the following table.

**Table 8. Responsibilities for PRM Partner Portal Analytics**

Responsibility	Dashboard	Page Tab Name
Partner Executive Analytics User	Partner Executive	Pipeline
	Partner Executive	Products
	Partner Executive	Sales Effectiveness
	Partner Executive	Service
Partner Operations Analytics User	Partner Commerce	Overview
	Partner Commerce	Products
	Partner Marketing	Overview
	Partner Marketing	ROI
	Partner Sales	Pipeline
	Partner Sales	Revenue
	Partner Service	Customer Sat
Partner Service	Overview	

	Partner Service	Service Requests
	Partner Training	Training
Partner Sales Manager Analytics User	Partner Commerce	Orders
	Partner Commerce	Overview
	Partner Commerce	Quotes
	Partner Sales	Pipeline
	Partner Sales	Revenue
	Partner Sales	Subordinates
	Partner Training	Subordinates
Partner Sales Rep Analytics User	Partner Commerce	Orders
	Partner Commerce	Overview
	Partner Commerce	Quotes
	Partner-Sales	Pipeline
	Partner-Sales	Revenue
	Partner-Sales	Subordinates
	Partner Training	Subordinates
Partner Service Manager Analytics User	Partner Service	Customer Sat
	Partner Service	Overview
	Partner Service	Service Requests
	Partner Service	Subordinates
	Partner Training	Subordinates
Partner Service Rep Analytics User	Partner Service	Overview
	Partner Service	Service Requests
	Partner Service	Subordinates
	Partner Training	Subordinates

## Partner Manager Role-Based Interactive Dashboards Mapping

The following table provides the dashboard and page tab mapping for specific responsibilities in the Siebel PRM Partner Manager application.

**Table 9. Siebel Responsibilities for PRM Analytics**

Responsibility	Dashboard	Page Tab Name
Channel Account Manager Analytics User	Channel Customers	Overview
	Channel Customers	Sales
	Channel Sales	Products
	Channel Sales	Sales

	Channel Service	Products
	Channel Service	Service
	Channel Training	Training Profile
Channel Executive Analytics User	Channel Customers	Customer Profile
	Channel Executive	Customer Satisfaction
	Channel Executive	Pipeline
	Channel Executive	Product
	Channel Executive	Program
	Channel Executive	Revenue
	Channel Executive	Service
	Channel Segmentation	Channel Mix
	Channel Segmentation	Partner Territory
	Channel Segmentation	Partner Tier
	Channel Segmentation	Partner type
Channel Marketing Manager Analytics User	Channel Customers	Overview
	Channel Customers	Sales
	Channel Marketing	Effectiveness
	Channel Marketing	Response
	Channel Marketing	ROI
Channel Operations Analytics User	Channel Commerce	Orders
	Channel Commerce	Overview
	Channel Commerce	Quotes
	Channel Commerce	Products
	Channel Customers	Overview
	Channel Customers	Sales
	Channel Customers	Service
	Channel Marketing	Effectiveness
	Channel Marketing	Overview
	Channel Sales	Margins
	Channel Sales	Pipeline
	Channel Sales	Revenue
	Channel Sales	Sales Cycle
	Channel Sales	Wins
	Channel Segmentation	Partner Territory

Channel Segmentation	Partner Tier
Channel Segmentation	Partner Type
Channel Service	Customer Satisfaction
Channel Service	Overview
Channel Service	Products
Channel Service	Resolution Time
Channel Service	Service Requests
Channel Training	Overview
Channel Training	Performance

## PRM Analytics Subject Area Mappings

Ad hoc queries in Siebel PRM Analytics are built by the user, depending on user responsibilities and based on columns in subject areas in the Oracle BI application. By restricting visibility to subject areas based on responsibilities, PRM Analytics provides brand owners a flexible way to deploy role-based analytics.

The subject area visibility for responsibilities in Partner Manager are shown in the following table, where an X indicates that the subject area is visible for the user holding that responsibility.

**Table 10. Responsibilities for PRM Partner Manager Analytics**

Subject Area	Channel Executive Analytics User	Channel Operations Analytics User	Channel Account Manager Analytics User	Channel Marketing Manager analytics User
Activities	X	X	X	X
Assets	X	X	X	
Campaigns	X	X	X	X
Consumers	X	X	X	X
Customer Satisfaction	X	X	X	
Customers	X	X	X	X
Orders	X	X	X	X
Partner Training	X	X	X	
Partners	X	X	X	X
Pipeline	X	X	X	X
Pricing	X	X	X	X
Products	X	X	X	X
Real-Time Activity				



## PRM Analytics Data-Level Visibility

PRM Analytics also provides brand owners the ability to restrict security based on the user's organization or position. This security mechanism makes sure that one user does not have access to another user's data. It also makes sure that one partner does not have access to another partner's data. Data-level security is administered for responsibilities. Details regarding setting up data -level visibility are provided in the topic: Implementing Data-Level Security in the Oracle BI Repository.

The following table shows the data-level security settings included for the responsibilities in Partner Manager and Partner Portal.

**Table 12. Oracle PRM Data-Level Security Settings**

Responsibility	Data-Level Security	Type	Comments
Channel Executive Analytics User	No	N/A	N/A
Channel Operations Analytics User	No	N/A	N/A
Channel Account Manager Analytics User	No	N/A	N/A
Channel Marketing Manager Analytics User	No	N/A	N/A
Partner Executive Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Sales Manager Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Sales Rep Analytics User	Yes	Position	Displayed records should match position of the user.
Partner Service Manager Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Service Rep Analytics User	Yes	Position	Displayed records should match position of the user.

## About Usage Accelerator Analytics Security Settings

The following table describes the additional security configurations that may be necessary and the particular responsibilities associated with the Oracle Usage Accelerator dashboards.

**Table 13. Usage Accelerator Responsibilities and Dashboards**

User Responsibility	Data-Level Security	Dashboard Name(View)	Dashboard Page
---------------------	---------------------	----------------------	----------------

Usage Accelerator: Sales Rep	Primary Position Data-Level Security	Score Card	Individual ScoreCard
Usage Accelerator: Financial Services Sales Rep		Action Plan	Account Coverage Contact Coverage Opportunity Coverage Financial Account Coverage: Financial Services only Account Completeness Contact Completeness Opportunity Updates
Usage Accelerator: Sales Manager	No Position-Based Security	Score Card	Team ScoreCard Individual ScoreCard
		Action Plan	Account Coverage (Team) Contact Coverage (Team) Opportunity Coverage (Team) Financial Account Coverage (Team): Financial Services only Account Completeness (Team) Contact Completeness (Team) Opportunity Updates (Team)
		Master Data Management Customer Hub	Master Record Completeness Master Records Completeness Detail Accuracy
Usage Accelerator: Financial Services Sales Manager		Coverage	Account Coverage Account Coverage (Team) Contact Coverage

			Opportunity Coverage
			Financial Account Coverage:
		Completeness	Financial Services only
			Account Completeness
		Opportunity Updates	Contact Completeness
		User Adoption	Opportunity Updates
			Active Users
			Application Usage: excluded for Financial Services
			Application Usage: Financial Services only*
Usage Accelerator: Sales Executive	No Position Based Security	ScoreCard	Organization Scorecard
			Individual Scorecard
		Master Data Management Customer Hub	Master Record Completeness
			Master Records Completeness Detail
			Accuracy
Usage Accelerator: Financial Services Sales Executive		Action Plan	Account Coverage (Org)
			Contact Coverage (Org)
			Opportunity Coverage (Org)
			Financial Account Coverage(Org): Financial Services only
			Account Completeness (Org)
			Contact Completeness (Org)
			Opportunity Updates (Org)

## About Primary Owner-Based Security

Primary owner-based security is supported through the "Primary Owner-Based Security" security group. This type of security mechanism allows records to be visible only to their primary owner. By default, this type of security supports a few dimensions in the Core business model, but other tables can be added if they have a primary owner's source Integration ID column.

The security filter in this security group is defined as:

```
"Core"."Dim - Activity"."VIS_PR_OWN_ID" = VALUEOF(NQ_SESSION."PR_
OWNER_ID")
```

The session variable PR\_OWNER\_ID is a single value variable, populated by the Primary Owner ID initialization block. This initialization block runs the following SQL, for the Siebel OLTP data source, to populate the variable:

```
select PAR_ROW_ID
from VALUEOF(TBO).S_USER
where LOGIN = ':USER'
```

## About Business Unit-Based Security

Business unit-based security is supported through the "Primary Org-Based Security" security group. By default, only a few dimensions in the Core, Workforce Analytics and Forecasting business models support this data security type. Other fact and dimension tables can be added to this security group if they have the column VIS\_PR\_BU\_ID column populated.

The security filter in this security group is defined as:

```
"Core"."Dim - Order"."VIS_PR_BU_ID" = VALUEOF(NQ_SESSION."ORGANIZATION")
```

The session variable ORGANIZATION is a Row-wise variable, initialized using the Initialization block: Orgs for Org-Based Security. This Init Block runs the following SQL for the Siebel OLTP data source, to populate the ORGANIZATION variable:

```
select distinct 'ORGANIZATION', PRR.SUB_PARTY_ID
from VALUEOF(TBO).S_POSTN P, VALUEOF(TBO).S_USER U,
VALUEOF(TBO).S_PARTY_PER PP,VALUEOF(TBO).S_PARTY_RPT_REL PRR
where U.ROW_ID=PP.PERSON_ID and P.ROW_ID=PP.PARTY_ID and PRR.PARTY_ID =
P.BU_ID and PRR.PARTY_TYPE_CD = 'Organization' and U.LOGIN = ':USER'
```

## About Security Integration with Oracle's JDEdwards EnterpriseOne or JDEdwards World

All information in this topic pertaining to JD Edwards EnterpriseOne also applies to JD Edwards World.

This topic covers an approach to security integration between Oracle Business Intelligence Enterprise Edition (Oracle BI EE) and JD Edwards EnterpriseOne using the Lightweight Directory Access Protocol (LDAP). It contains the following topics:

- How Oracle BI EE and JD Edwards Enterprise One Use LDAP
- Integration of User and Object Security
- Implementing LDAP Integration for User and Object Security

## How Oracle BI EE and JD Edwards Enterprise One Use LDAP

LDAP can serve as a central repository of security information for both JD Edwards EnterpriseOne and Oracle BI EE, allowing administrators to configure security once for both systems. An LDAP server stores credentials required for authentication as well as user profile information such as JD Edwards EnterpriseOne roles and Oracle BI EE security groups.

At login, the Oracle BI EE server passes a user's credentials to the LDAP server for authentication. Upon successful authentication, an Oracle BI EE initialization block retrieves the Oracle BI EE security group

names from the user's LDAP record. These group names are stored in the Oracle BI EE session variable GROUP and used throughout the user's session to determine which applications, dashboards, and other objects the user has permission to access.

Similarly, at login the JD Edwards EnterpriseOne security kernel passes a user's credentials to the LDAP server. Upon successful authentication, the JD Edwards EnterpriseOne security kernel retrieves the user-role relationship information which is used for both object and data security.

## **Integration of User and Object Security**

LDAP can provide an integration for Oracle Business Intelligence Enterprise Edition and JD Edwards EnterpriseOne for user and object security only. LDAP cannot provide an integrated data security solution. Therefore, to implement data security, you must configure security separately on each server. This requires user authentication to be set up on both the Oracle Business Intelligence Enterprise Edition server and the JD Edwards EnterpriseOne server. If data security is a requirement, LDAP integration of user and object security provides no value.

## **Implementing LDAP Integration for User and Object Security**

This topic contains the following topics:

- About Configuring Oracle Business Intelligence Enterprise Edition to Use LDAP
- About Configuring JD Edwards EnterpriseOne to Use LDAP

### **About Configuring Oracle Business Intelligence Enterprise Edition to Use LDAP**

For instructions on how to configure Oracle Business Intelligence Enterprise Edition to allow authentication of users through LDAP, see the topic about setting up LDAP authentication in the Oracle Business Intelligence Server Administration Guide.

When Oracle Business Intelligence Enterprise Edition is configured with LDAP, the Oracle BI EE GROUP initialization block needs to be created to retrieve the user's group information from the LDAP record. For information about the process of creating initialization blocks, see the *Oracle Business Intelligence Server Administration Guide*.

### **About Configuring JDEdwards EnterpriseOne to Use LDAP**

For instructions on how to configure JD Edwards EnterpriseOne to allow authentication of users through LDAP, see the *JD Edwards EnterpriseOne Tools Security Administration Guide*.

### **About Configuring JDEdwards World to Use LDAP**

For instructions on how to configure JD Edwards World to allow authentication of users through LDAP, see the *JD Edwards World Technical Foundation Guide*.

## **Integrating Data Security for SAP R/3**

This topic explains how security in Oracle BI Applications is deployed with SAP R/3. Read this topic if you want to understand how the default security settings are configured so that you can change the way security is implemented if required. This topic contains the following topics:

- Company Org-Based Security for SAP R/3 Financials
- Payables Org-Based Security for SAP R/3 Financials

- Receivables Org-Based Security for SAP R/3 Financials

## Company Org-Based Security for SAP R/3 Financials

The sequence for Company org-based security for SAP R/3 is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

USER (System variable)

2. The Oracle BI Server then gets the company code(s) corresponding to the USER from the staging table that would hold information for User, Role and Company Codes.

Following SAP tables are used for populating the staging table.

- USR02: Users
- AGR\_USERS: Roles and Users
- AGR\_1252: Roles, Field and Values
- AGR\_1251: Roles, field and Values
- T001: Company Codes

Field would be "Company Code" a.k.a. BUKRS for SAP R/3.

3. The following session variable is set automatically: COMPANY (Row-wise variable)

The initialization block 'Companies', which sets the value for this variable, is shown below.

### Initialization block -- 'Companies'

The initialization block 'Companies' sets value for variable COMPANY using the following SQL:

#### For SAP R/3:

```
(SELECT DISTINCT 'COMPANY', COMPANY_CODE
FROM W_SAP_SEC_USR_TMP
WHERE COMPANY_CODE <> '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N'
UNION
SELECT DISTINCT 'COMPANY', B.COMPANY_CODE
FROM W_SAP_SEC_USR_TMP A
INNER JOIN W_SAP_SEC_COMPCODE_TMP B on 1=1
WHERE A.COMPANY_CODE = '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N')
UNION
(SELECT DISTINCT 'COMPANY', COMPANY_CODE
FROM W_SAP_SEC_USRS_TMP
WHERE COMPANY_CODE <> '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N'
UNION
SELECT DISTINCT 'COMPANY', B.COMPANY_CODE
```

```

FROM W_SAP_SEC_USRS_TMP A
INNER JOIN W_SAP_SEC_COMPCODES_TMP B on 1=1
WHERE A.COMPANY_CODE = '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N')

```

## Payables Org-Based Security for SAP R/3 Financials

The sequence for payables org-based security for SAP Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The Oracle BI Server then gets the company code(s) corresponding to the USER from the staging table that would hold information for User, Role and Company Codes.

Following SAP tables are used for populating the staging table.

- USR02: Users
- AGR\_USERS: Roles and Users
- AGR\_1252: Roles, Field and Values
- AGR\_1251: Roles, field and Values
- T001: Company Codes

Field would be "Company Code" a.k.a. BUKRS for SAP R/3.

3. The following session variable is set automatically: PAYABLES\_ORG (Row-wise variable)

The initialization block 'Payables Org-based Security', which sets the value for this variable, is shown below.

### Initialization block -- 'Payables Org-based Security'

The initialization block 'Payables Org-based Security' sets value for variable PAYABLES\_ORG using the following SQL:

#### For SAP R/3:

```

(SELECT DISTINCT 'PAYABLES_ORG', COMPANY_CODE
FROM W_SAP_SEC_USR_TMP
WHERE COMPANY_CODE <> '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N'
UNION
SELECT DISTINCT 'PAYABLES_ORG', B.COMPANY_CODE
FROM W_SAP_SEC_USR_TMP A
INNER JOIN W_SAP_SEC_COMPCODE_TMP B on 1=1
WHERE A.COMPANY_CODE = '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N')
UNION
(SELECT DISTINCT 'PAYABLES_ORG', COMPANY_CODE

```

```

FROM W_SAP_SEC_USRS_TMP
WHERE COMPANY_CODE <> '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N'

UNION

SELECT DISTINCT 'PAYABLES_ORG', B.COMPANY_CODE
FROM W_SAP_SEC_USRS_TMP A
INNER JOIN W_SAP_SEC_COMPCODES_TMP B on 1=1
WHERE A.COMPANY_CODE = '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N')

```

## Receivables Org-Based Security for SAP R/3 Financials

The sequence for receivables org-based security for SAP Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The Oracle BI Server then gets the company code(s) corresponding to the USER from the staging table that would hold information for User, Role and Company Codes.

Following SAP tables are used for populating the staging table.

- USR02: Users
- AGR\_USERS: Roles and Users
- AGR\_1252: Roles, Field and Values
- AGR\_1251: Roles, field and Values
- T001: Company Codes

Field would be "Company Code" a.k.a. BUKRS for SAP R/3.

3. The following session variable is set automatically: RECEIVABLES\_ORG (Row-wise variable)

The initialization block 'Receivables Org-based Security', which sets the value for this variable, is shown below.

### Initialization block -- 'Receivables Org-based Security'

The initialization block 'Receivables Org-based Security' sets value for variable RECEIVABLES\_ORG using the following SQL:

#### For SAP R/3:

```

(SELECT DISTINCT 'RECEIVABLES_ORG', COMPANY_CODE
FROM W_SAP_SEC_USR_TMP
WHERE COMPANY_CODE <> '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N'

UNION

SELECT DISTINCT 'RECEIVABLES_ORG', B.COMPANY_CODE
FROM W_SAP_SEC_USR_TMP A

```

```

INNER JOIN W_SAP_SEC_COMPCODE_TMP B on 1=1
WHERE A.COMPANY_CODE = '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N')
UNION
(SELECT DISTINCT 'RECEIVABLES_ORG', COMPANY_CODE
FROM W_SAP_SEC_USRS_TMP
WHERE COMPANY_CODE <> '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N')
UNION
SELECT DISTINCT 'RECEIVABLES_ORG', B.COMPANY_CODE
FROM W_SAP_SEC_USRS_TMP A
INNER JOIN W_SAP_SEC_COMPCODES_TMP B on 1=1
WHERE A.COMPANY_CODE = '*' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND
USER_LOCK = 'N')

```

USR02 stores the logon data.

AGR\_USERS stores assignment of Roles to Users. A role describes the activities of a SAP user.

The Company Code in a role can be defined at individual level (Manually) as well as at Organization level.

Note that the values for org level fields maintained through the org level option in PFCG (PFCG is the transaction that is used to create and maintain security roles in SAP) are stored in AGR\_1252 rather than AGR\_1251. In AGR\_1251, for authorizations containing org level fields, the field value record will appear as \$BUKRS (for Company Code). With BUKRS being the actual field the actual value maintained for this authorization will then be stored in AGR\_1252.

The Company Code field in an authorization Object can have following values :

- \* (meaning all company codes) : The user is having access to all company codes.
- Range : Access to all Company codes in the given range including the 'From' and 'To' value.
- Value like 1000, 2000 etc.

Note - There is one authorization field found in most Authorization objects i.e. Activity which defines the possible actions which could be performed over a particular application object has not been captured.

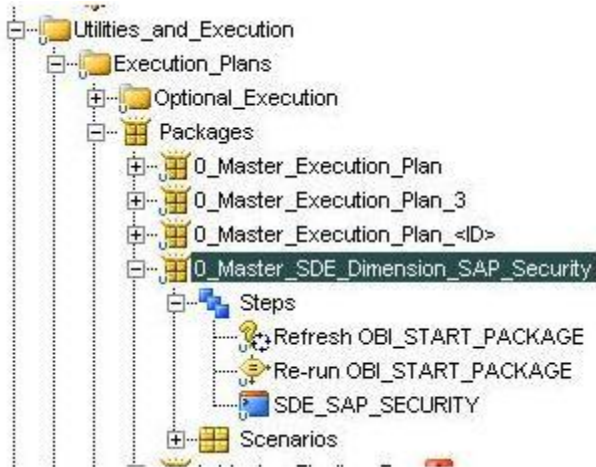
### **Configuring Data Extract From SAP R/3 using Oracle Data Integrator**

A Master Package has been created to extract Security information from SAP source tables.

This Package will be executed manually. The steps involved in scheduling this package is shown below :

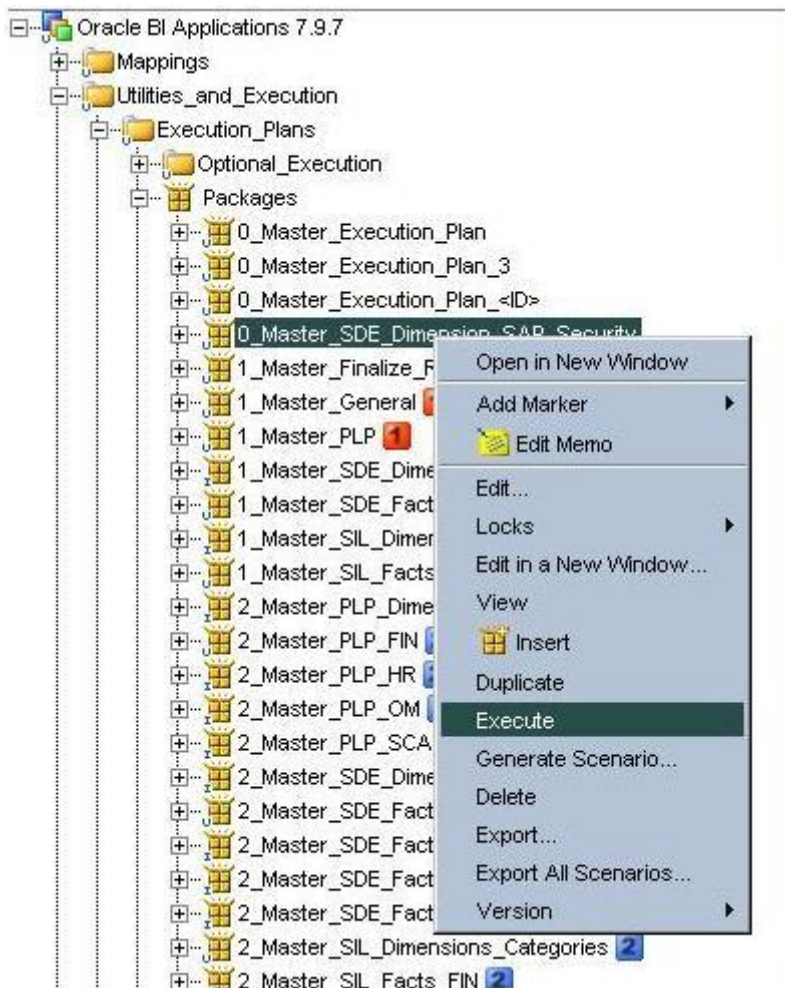
- a. Expand the folders

Utilities\_and\_Execution -> Execution\_Plans -> Packages ->  
 0\_Master\_SDE\_Dimension\_SAP\_Security.

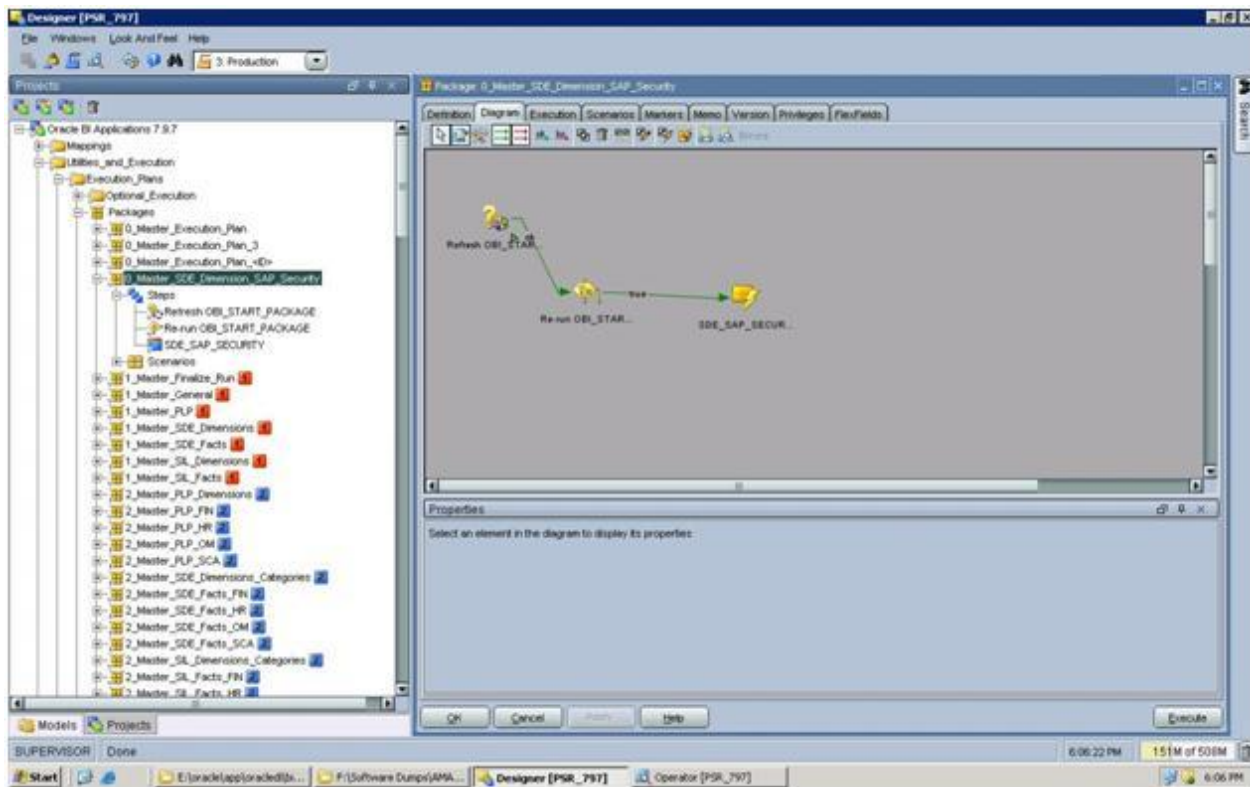


- b. Right Click on the Master Package 0\_Master\_SDE\_Dimension\_SAP\_Security and press Execute.
- c. Use the appropriate Context.

Below is the Screenshot for the same.



Double click on the Master package , the Designer would look as shown in below screenshot.



The table holding the information for Roles, Users and Company Codes would lie in Staging Area as SAP R/3 is an application and SAP recommends extracting data through ABAP. The table would be always Truncate and Load. Master package execution will always start from Oracle data integrator designer.

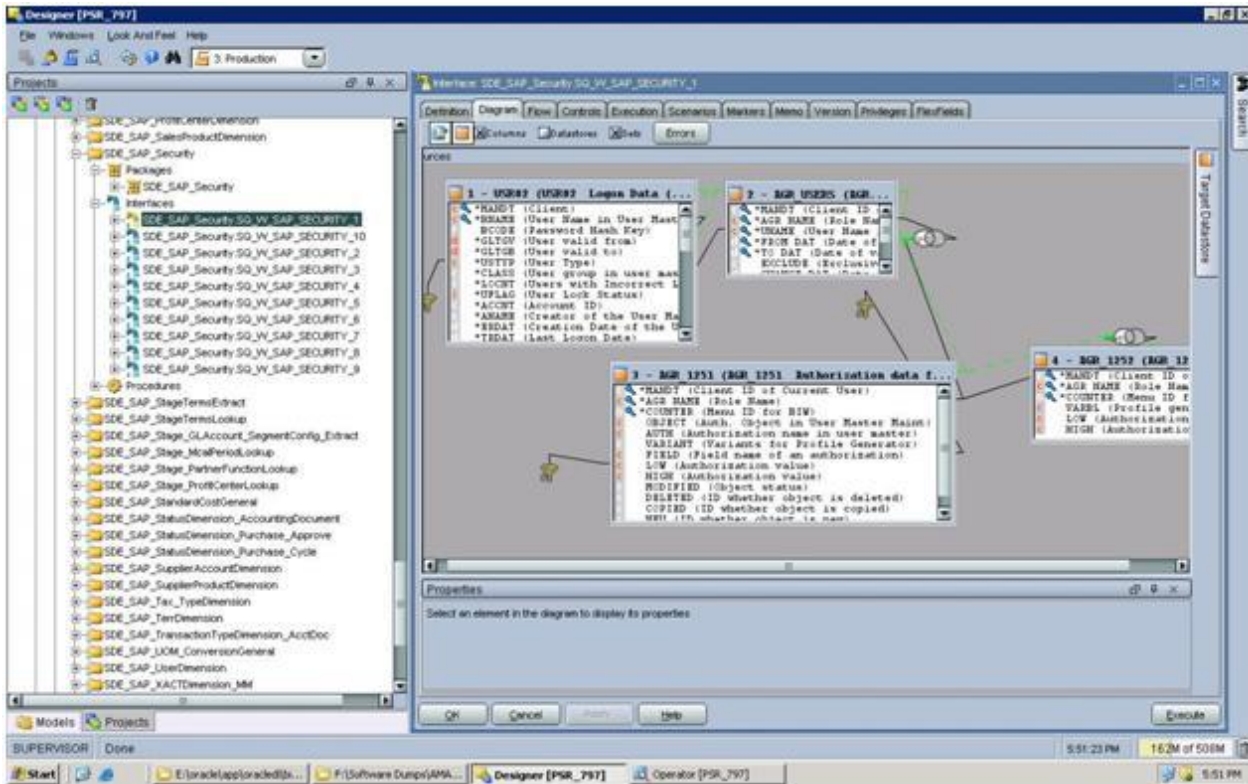
The SAP Tables used for extracting Users, Roles & Company Codes are given below :

USR02

AGR\_USERS

AGR\_1251

AGR\_1252



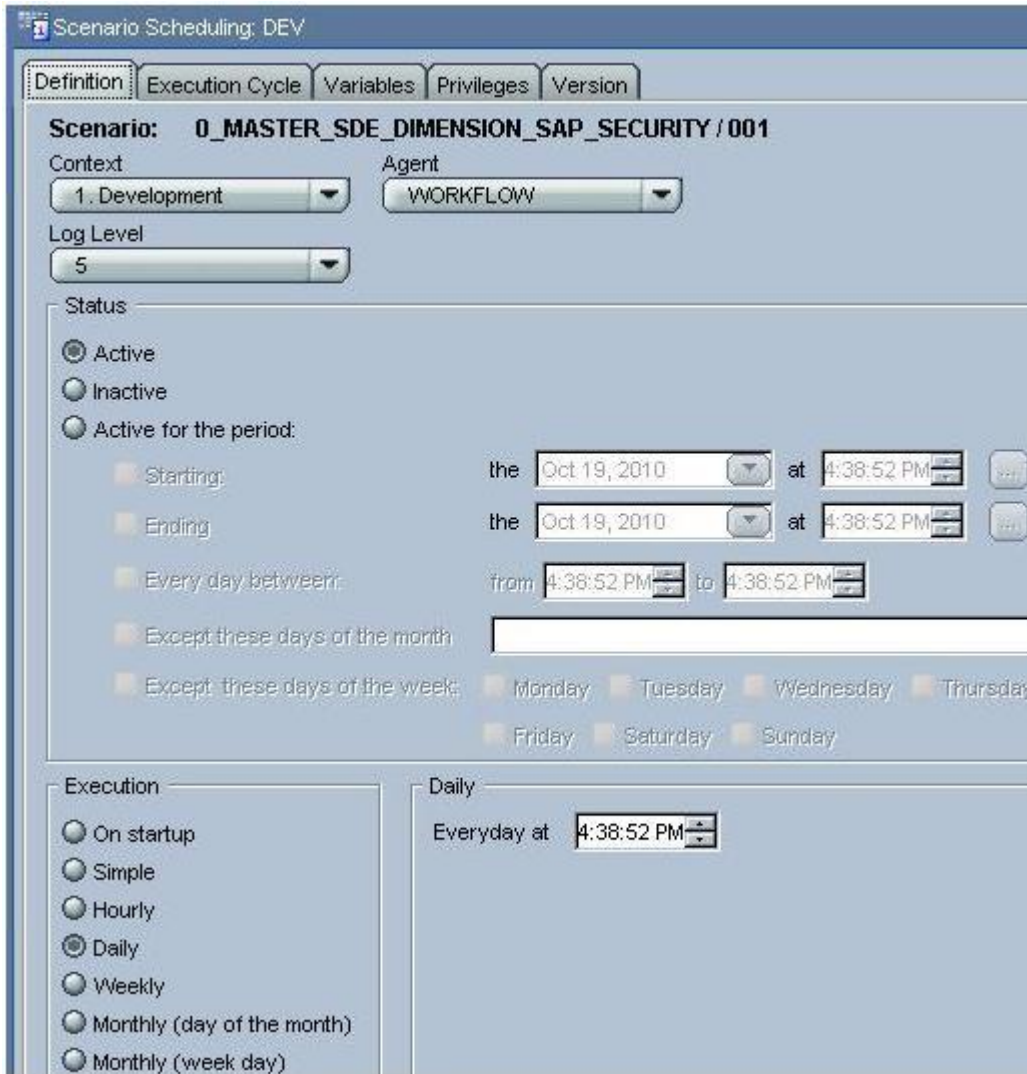
## Scheduling the Security Package

Please follow the steps to schedule the scenario of the 0\_MASTER\_SDE\_DIMENSION\_SAP\_SECURITY package for daily run

1. Expand the 0\_Master\_SDE\_Dimension\_SAP\_Security, and click Scenarios.
2. Expand 0\_MASTER\_SDE\_DIMENSION\_SAP\_SECURITY Version 001 and right click on Scheduling and choose Insert Scheduling.



3. Choose the options as per the requirement to schedule the job. Select the appropriate agent and the log level and Apply the changes.



4. As the last step, select the appropriate AGENT from Topology manager 'Physical Architecture' tab, and right click to select update scheduling.

