Oracle Business Intelligence Applications

Security Guide

Release 7.9.7.2

E18914-03

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Table of Contents

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 for SAP
- Oracle Supply Chain and Order Management Analytics for SAP
- Oracle Procurement and Spend Analytics for SAP

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 Certification Matrix for Oracle Business Intelligence Applications, available at available at.
http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.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/technetwork/middleware/bi-foundation/documentation/bi-apps-098545.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 7.9.7.2 documentation set (available at http://www.oracle.com/technetwork/middleware/bi-foundation/documentation/bi-apps-098545.html):

- *Oracle Business Intelligence Applications Release Notes*
- *Oracle Business Intelligence Applications Installation Guide for Oracle Data Integrator Users*
- *Oracle Business Intelligence Configuration Guide for Oracle Data Integrator Users*
- *Certification matrix (*http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html*)*

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, codein 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 7.9.7.2

This guide includes the following change

- No changes in for this document in 7.9.7.2 release
- Added the topic in release 7.9.7.1 : 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

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 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

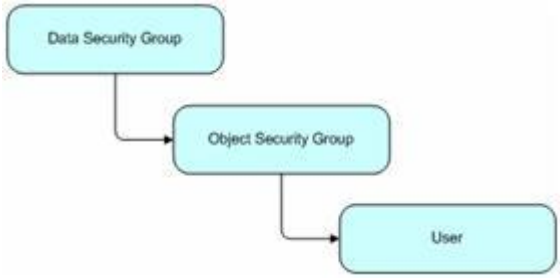
Object-level and data-level security are implemented in Oracle BI Applications using security groups. These security groups are defined using the Security Manager in the Oracle BI Administration Tool. Security groups can be related to data, objects, or both data and objects. For example, the Oracle BI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

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 or roles (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.

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 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	SAP R/3
Operating Unit Org-Based security	Available in 7.9.7.2
Inventory Org-Based Security	Available in 7.9.7.2
Company Org Based Security	Available in 7.9.7.2
Payables Org-Based Security	Available in 7.9.7.2
Receivables Org-Based Security	Available in 7.9.7.2

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
Company Org-Based Security	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
Inventory Org-Based Security	SAP R/3	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
Operating Unit Org-Based Security	SAP R/3	This security group filters data based on the organizations associated to the user that is logged in.	Operating Unit Organizations
Payables Org-Based Security	SAP R/3	This security group filters data based on the payables business units	Payables Organizations

		associated to the user that is logged in. The business unit is the highest level key structure in PeopleSoft. The company code is the Payables Org in SAP R/3.	
Receivables Org-Based Security	SAP R/3	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.	Receivables Organizations

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

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*.

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_POSTION_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 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'
```

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 -- ' Companies'

   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')
```

Inventory Org-Based Security for SAP R/3 Material Management

The sequence for Inventory org-based security for SAP Material Management 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 Plant(s) corresponding to the USER from the staging table that would hold information for User, Role and Plants.

   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
- T001W: Plants

  Field would be “Plant” a.k.a. WERKS for SAP R/3.

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

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

   Initialization block -- ' Inventory Organizations'

   The initialization block 'Inventory Org-based Security' sets value for variable INVENTORY_ORG using the following SQL:

   For SAP R/3:

SELECT DISTINCT 'INV_ORG', ORG_CODE

FROM W_SAP_SEC_USR_TMP

WHERE ORG_CODE <> '*' AND ORG_TYPE='PLANT' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'

UNION

SELECT DISTINCT 'INV_ORG', B.PLANT

FROM W_SAP_SEC_USR_TMP A

INNER JOIN W_SAP_SEC_PLANT_TMP B on 1=1

WHERE A.ORG_CODE = '*' AND A.ORG_TYPE='PLANT' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'

UNION

SELECT DISTINCT 'INV_ORG', ORG_CODE

FROM W_SAP_SEC_USRS_TMP

WHERE ORG_CODE <> '*' AND ORG_TYPE='PLANT' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'

UNION

SELECT DISTINCT 'INV_ORG', B.PLANT

FROM W_SAP_SEC_USRS_TMP A

INNER JOIN W_SAP_SEC_PLANT_TMP B on 1=1

WHERE A.ORG_CODE = '*' AND A.ORG_TYPE='PLANT' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'

Purchasing Org-Based Security for SAP R/3 Material Management

The sequence for Purchasing org-based security for SAP Material Management 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 Purchase Org(s) corresponding to the USER from the staging table that would hold information for User, Role and Purchase Org.

   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
- T024E: Purchasing Organization

  Field would be “Purchase Org” a.k.a. EKORG for SAP R/3.

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

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

   Initialization block -- ' Operating Unit Organizations'

   The initialization block 'Purchasing Org-based Security' sets value for variable PURCHASING_ORG using the following SQL:

   For SAP R/3:

```
SELECT DISTINCT 'OU_ORG',ORG_CODE
FROM W_SAP_SEC_USR_TMP
WHERE ORG_CODE <> '*' AND ORG_TYPE='PURCH_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'
UNION
SELECT DISTINCT 'OU_ORG', B.PURCH_ORG
FROM W_SAP_SEC_USR_TMP A
INNER JOIN W_SAP_SEC_PURCH_ORG_TMP B on 1=1
WHERE A.ORG_CODE = '*' AND A.ORG_TYPE='PURCH_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'
UNION
SELECT DISTINCT 'OU_ORG', ORG_CODE
FROM W_SAP_SEC_USRS_TMP
WHERE ORG_CODE <> '*' AND ORG_TYPE='PURCH_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'
UNION
SELECT DISTINCT 'OU_ORG', B.PURCH_ORG
FROM W_SAP_SEC_USRS_TMP A
INNER JOIN W_SAP_SEC_PURCH_ORG_TMP B on 1=1
WHERE A.ORG_CODE = '*' AND A.ORG_TYPE='PURCH_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'
UNION
SELECT DISTINCT 'OU_ORG', ORG_CODE
FROM W_SAP_SEC_USR_TMP
WHERE ORG_CODE <> '*' AND ORG_TYPE='SALES_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'
UNION
SELECT DISTINCT 'OU_ORG', B.SALES_ORG
FROM W_SAP_SEC_USR_TMP A
INNER JOIN W_SAP_SEC_SALES_ORG_TMP B on 1=1
WHERE A.ORG_CODE = '*' AND A.ORG_TYPE='SALES_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'
UNION
SELECT DISTINCT 'OU_ORG', ORG_CODE
FROM W_SAP_SEC_USRS_TMP
WHERE ORG_CODE <> '*' AND ORG_TYPE='SALES_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'
```

UNION

SELECT DISTINCT 'OU_ORG', B.SALES_ORG

FROM W_SAP_SEC_USRS_TMP A

INNER JOIN W_SAP_SEC_SALES_ORG_TMP B on 1=1

WHERE A.ORG_CODE = '*' AND A.ORG_TYPE='SALES_ORG' AND USER_NAME = ':USER' AND VALID_USER = 'Y' AND USER_LOCK = 'N'

Sales Org-Based Security for SAP R/3 Sales and Distribution

The sequence for Sales org-based security for SAP Sales & Distribution 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 Purchase Org(s) corresponding to the USER from the staging table that would hold information for User, Role and Sales Org.

   Following SAP tables are used for populating the stagFing table.

- USR02: Users
- AGR_USERS: Roles and Users
- AGR_1252: Roles, Field and Values
- AGR_1251: Roles, field and Values
- TVKO: Sales Organization

  Field would be “Sales Org” a.k.a. VKORG for SAP R/3.

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

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

   Initialization block -- 'Sales Org-based Security'

   The initialization block 'Sales Org-based Security' sets value for variable SALES_ORG using the following SQL:

For SAP R/3:

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.

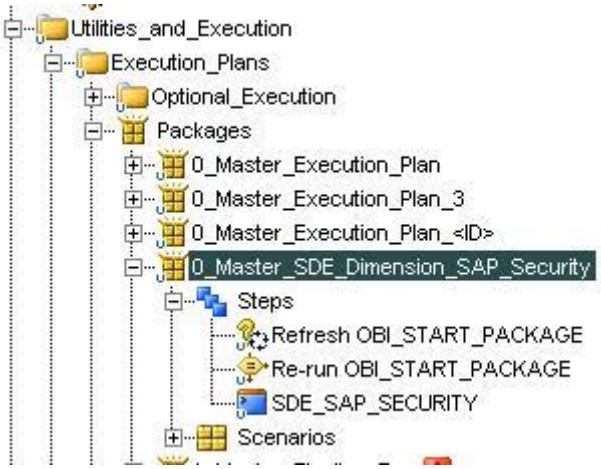
Configuring Data Extract From SAP R/3 using Oracle Data Integrator

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

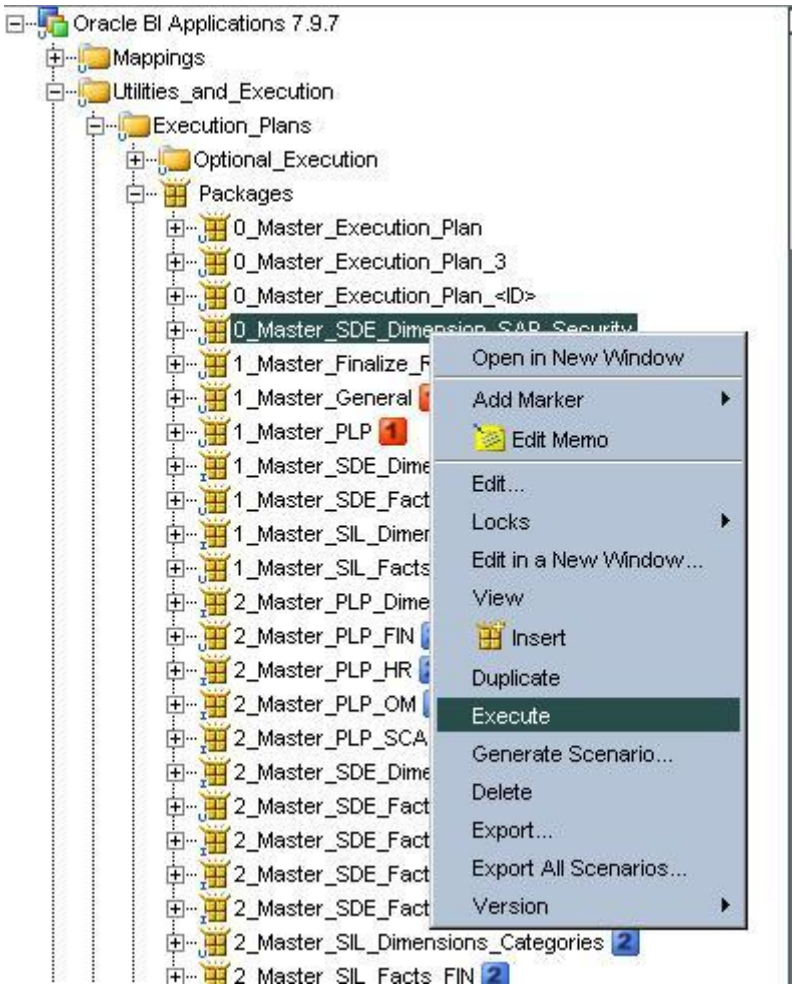
Utilities_and_Execution -> Execution_Plans -> Packages -> 0_Master_SDE_Dimension_SAP_Security.



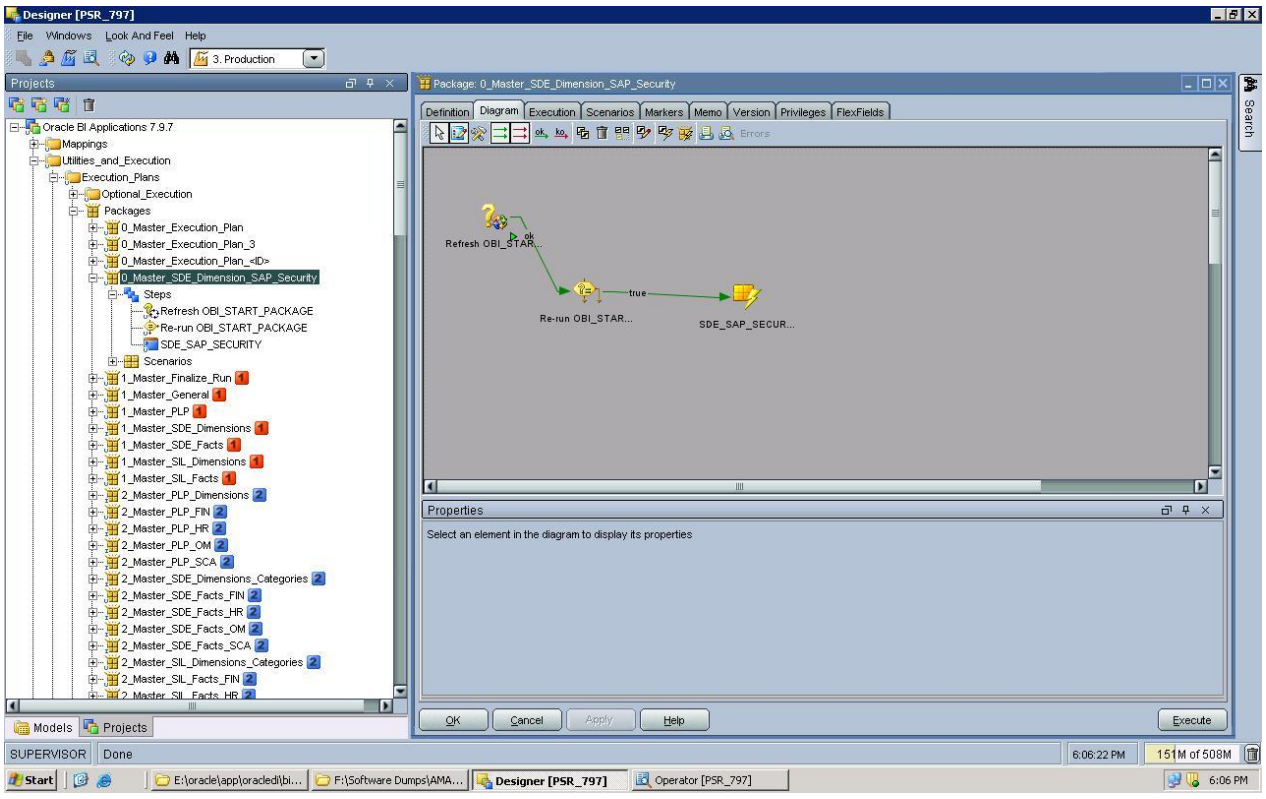
b. Right Click on the Master Package 0_Master_SDE_Dimension_SAP_Security and press Execute.

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.

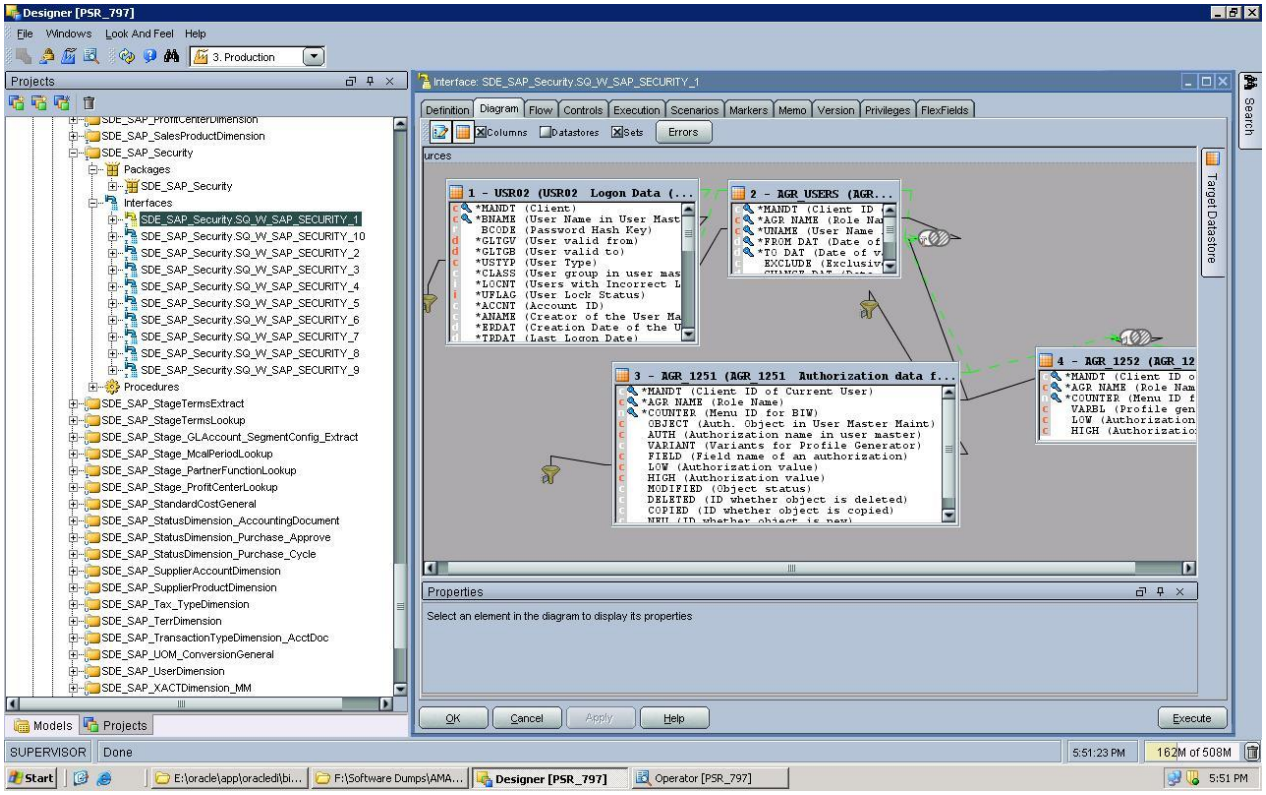
The SAP Tables used for extracting Users, Roles & Company Codes are given below :

USR02

AGR_USERS

AGR_1251

AGR_1252

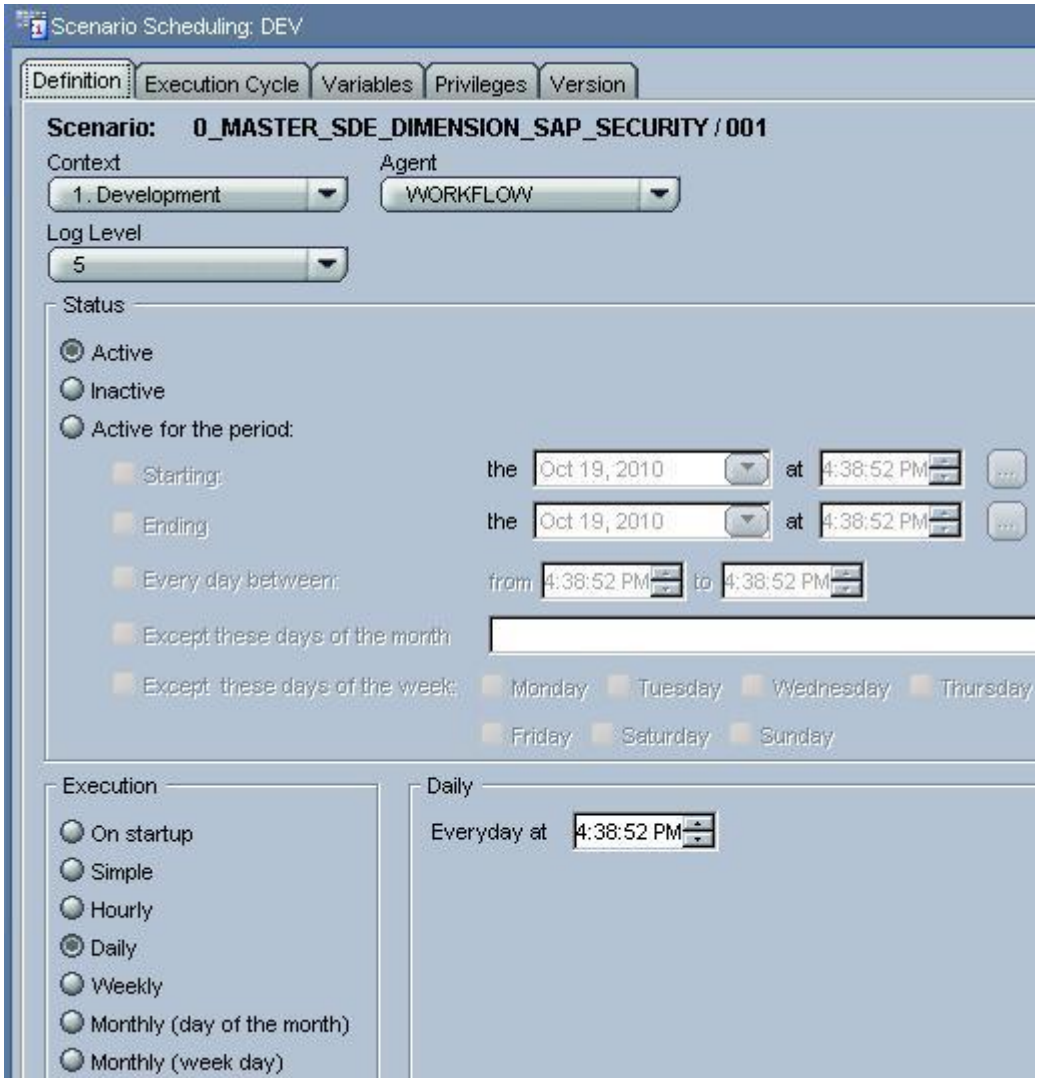


Scheduling the Security Package

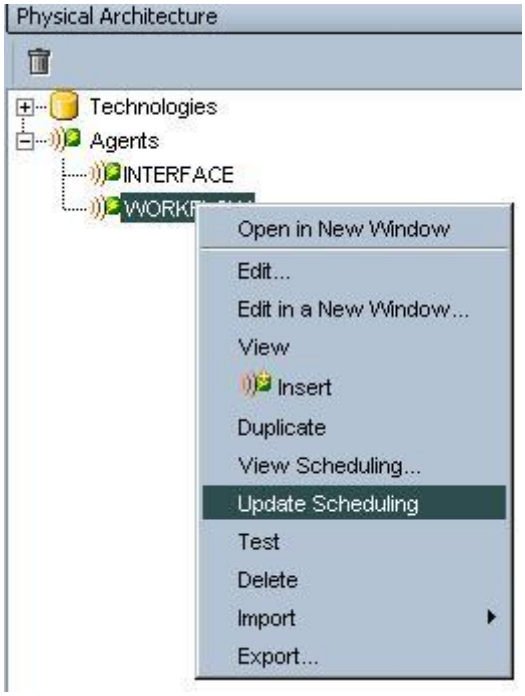
Please follow the steps to schedule the scenario of the 0_MASTER_SDE_DIMENSION_SAP_SECURITY package for daily run

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.



Restarting the Security Package in case of Failure

In case of failure in between the run due to some issue(environment related or other), Security package can be restarted by following the below steps manually after correcting the cause of error.

1. Delete entries from c_load_dates table from the data schema using below query:

delete from c_load_dates

where package_name like '%SDE_SAP_SECURITY%'

2. Execute package 0_Master_SDE_Dimension_SAP_Security.