ORACLE
INSURANCE

# Oracle Health Insurance Datamarts

# Administrator Reference

version 9.32

Part number: E41643-01

June 2013

ORACLE

# CHANGE HISTORY

| Release | Version | Changes |
|---|---|---|
| 10.12.2.0.0 | 9.29 | • Added change history paragraph.<br>• Added description for Oracle Business Intelligence translations external file (WBX_OBIEE_TRANSLATIONS.csv)<br>• Added description for configuring generic structure of claim properties<br>• Added description how to configure DWH_OHI_BI.rpd connection pools. |
| 10.12.3.0.0 | 9.30 | • Added new fact table DWH_ONDERHANDEN_WERK (Work in Progress Claims)<br>• Replaced generic role OZG_ROL_SELECT by secure role OBD_SELECT_ROLE. |
| 10.13.1.0.0 | 9.31 | • Added Exadata Support, Migrating OWB from single instance to multi node RAC, added OBI_SELECT_USER<br>• Added new batch Correct monitorcode (ZRGO005S) |
| 10.13.2.0.0 | 9.32 | • Added description for generic dimension properties (DIM_EIGENSCHAPPEN.dat)<br>• Added new parameter P_MAX_DCE_VERSCHIL; a threshold that is taken into account when performing crosschecks for claims.<br>• Added details for the Members check (three differente reference dates)<br>• Added new description for setting 'Policies to date'<br>• Corrected examples for dynamic claim properties<br>• Added Default Secure Install  and Enable Security Audit |

# CONTENTS

# INTRODUCTION

The purpose of this document is to give an overview of the architecture of Oracle Health Insurance Datamarts, as well as information on installation and management.

This document may be used as both a training material and a reference material. It is assumed that the reader has a basic knowledge of the Oracle tools being used.

With this document the Oracle Health Insurance Datamarts administrator should be in a position to install and maintain the Data Warehouse.

## IMPORTANT DOCUMENTATION

The documentation below is important for the activities of the functional administrator. These describe the design of Oracle Health Insurance Datamarts and contain information that is (potentially) required for installation and management.

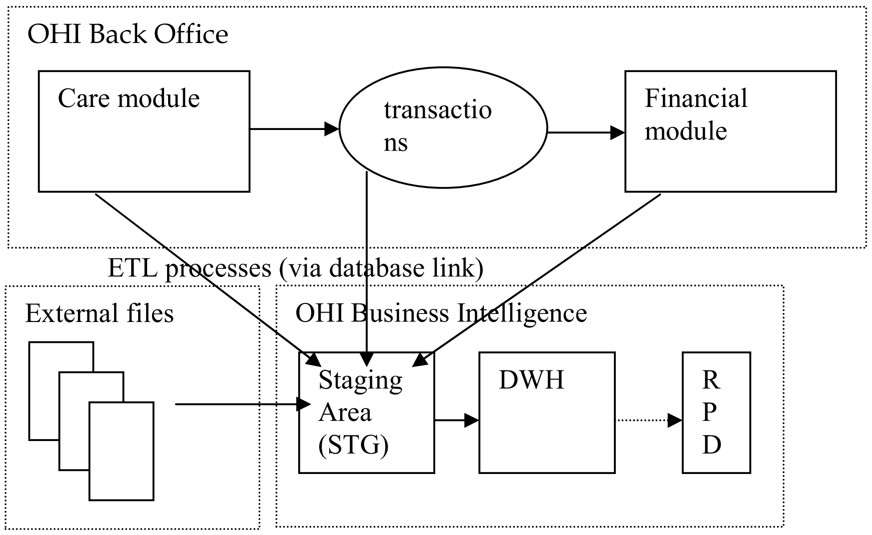
1. Standard product manuals from Oracle DBMS, Oracle Warehouse Builder and Oracle Business Intelligence Enterprise Edition (OBI EE). Documentation on these Oracle products can be found on the Oracle support website (http://support.oracle.com)
2. Certification form 10.13.1.0
3. CTA13508.doc: Oracle Health Insurance Installation, Configuration and DBA Manual
4. CTA13535.doc: Oracle Health Insurance Security Aspects

## SOFTWARE USED

Certifications matrix 10.13.1.0

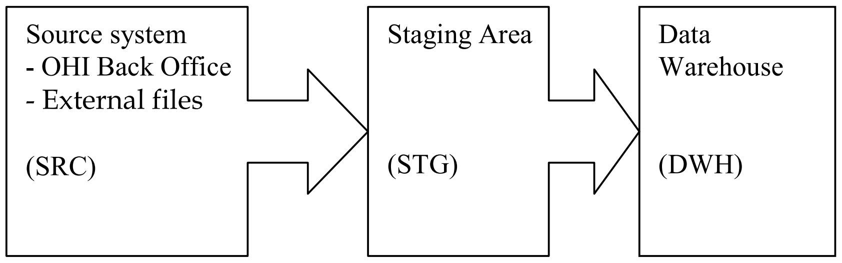
# OVERVIEW

The architecture of Oracle Health Insurance Datamarts broadly looks like this:



Oracle Health Insurance Datamarts makes use of three data areas. These data areas have a logical mutual connection. Their goal is to supply the data from the sources in a verified manner for reading in, checking and publishing.

The figure below shows the mutual relationships between the various data areas:



Data from the source system is always moved through the system in the following way:

1. 'Raw' data is loaded into the staging area.
2. Validated data is moved from the staging area to the Data Warehouse. There, the data is stored optimally for querying purposes and brought into alignment with the previously-loaded data.

# PART I INSTALLATION

## INTRODUCTION

Various components make up Oracle Health Insurance Datamarts. This chapter contains a short description of each.

The set-up and installation of these components is dealt with in the following paragraphs.

## SETTING UP ORACLE HEALTH INSURANCE DATAMARTS ENVIRONMENT

### OS set-up

Oracle Health Insurance Datamarts runs on a database server. The load run of Oracle Health Insurance Datamarts requires that there is an application server on which the OHI Back Office application is installed. Oracle Health Insurance Datamarts makes use of OHI Back Office functionality including, for example, the batch scheduler for starting and scheduling the load run.

For set-up of the application and database server environment, including set-up of the environmental variables and directory structure, see:

Oracle Health Insurance Installation, Configuration and DBA Manual
(chapter 3)

### Software installation

#### Database

Oracle Health Insurance Datamarts must be installed in a dedicated database. It is not permitted to install Oracle Health Insurance Datamarts in the same database as OHI Back Office, as Oracle Health Insurance Datamarts is a Data Warehouse and OHI Back Office is an OLTP system. Different user settings are used in each database.

**Please note** that the JServer option must also be installed in the database (and the *java_pool_size* parameter is filled in in "init.ora"). This is a requirement to be able to install the OWB Runtime environment.

For more details regarding the installation of the database software, see:

Oracle Health Insurance Installation, Configuration and DBA Manual
(chapter 2)

#### Oracle Warehouse Builder

An OWB Runtime environment is required in order to be able to execute the Oracle Health Insurance Datamarts ETL process. As of Oracle 11gR1 the OWB software is installed at the same time as the Oracle 11g software in the Oracle home directory of the database.

Oracle Health Insurance Datamarts releases and patches are installed from an application server. This application server can be used in 2 configurations. As stated before

1. Configuration with database server and application server on different hosts (separate database and application server)

   Because OWB is part of the complete Oracle 11g software installation, it was necessary in release 2010.01.0.0000 through release 2011.01.0000 to perform a full Oracle 11g software installation on both the application server and the database server. As of release 2011.01.0.0000, the installation software of Oracle Health Insurance only uses the OWB software in the 11g Oracle Home on the database server for the installation of Oracle Health Insurance Datamarts. The installation of Oracle Health Insurance Datamarts no longer uses the OWB software on the application server. Therefore, installation of the full Oracle 11g software on the application server is no longer necessary.
2. Configuration with database server and application server on same host (combined database and application server)

   There are no changes for this configuration compared to previous releases.

For the right version of the OWB software, see:

Certifications matrix 10.13.1.0

## Creation of an Oracle Health Insurance Datamarts database

A database needs to be created first before Oracle Health Insurance Datamarts can be installed.

### Parameters

The following database parameters must be set to the specified values:

```
AQ_TM_PROCESSES=1                #For batch scheduler queue time monitoring
DB_BLOCK_SIZE=16384 or DB_BLOCK_SIZE=8192
NLS_LANGUAGE=DUTCH               #DUTCH or AMERICAN
NLS_NUMERIC_CHARACTERS=",."      #Set to the desired combination
NLS_SORT=BINARY                  #For performance reasons
NLS_TERRITORY="THE NETHERLANDS"  #THE NETHERLANDS or AMERICA
OPTIMIZER_MODE=ALL_ROWS          #For performance reasons
OS_AUTHENT_PREFIX=""             #For externally ident. Users
PGA_AGGREGATE_TARGET=<value>     #Automatic SQL Execution Memory Management
REMOTE_OS_AUTHENT=TRUE           #Permit externally ident. Users
STAR_TRANSFORMATION_ENABLED=TRUE #For performance reasons
STATISTICS_LEVEL=TYPICAL         #For self-tuning capabilities
UNDO_MANAGEMENT=AUTO             #Automatic Undo
WORKAREA_SIZE_POLICY=AUTO        #Automatic SQL Execution Memory Management
```

The following database parameters must be set to the specified values at a *minimum*: When a *maximum* value applies, this is specified:

```
DML_LOCKS=500
OPEN_CURSORS=500
SESSION_CACHED_CURSORS=500
PROCESSES=200
JOB_QUEUE_PROCESSES=10
```

As well as the parameters above, there is also a parameter that is extremely important for parallel processing during both the loading process and for queries by end-users:

```
PARALLEL_MAX_SERVERS=number
```

If this setting is left out of the init.ora, it will be set to a very high value by default. It is therefore highly advisable to set a value for this parameter. However, the correct value for this parameter depends on the system (speed of I/O, number and speed of CPUs, among other things). Unfortunately, there is no hard and fast rule. More details on setting these parameters can be found on Oracle Support under note 280939.1 "Checklist for Performance Problems with Parallel Execution". The manual 'Oracle Database Data Warehousing Guide' describes how the execution of parallel statements works. To find out whether parallel statements have actually been executed serially due to a lack of parallel servers, the following statement, for example, can be used:

```
select * from gv$sysstat where name like 'Parallel operation%';
```

☞ **Note 1:** In the production environment it is *not* permitted to have activated the database events, unless explicitly requested by Oracle Health Insurance Development or Oracle Support Services.

☞ **Note 2:** If user settings that have not been recommended or prescribed are used in the Database or Application Server, the customer may be asked to reset these user settings if problems arise that may be connected.

The underlying reason for this is to avoid unnecessary instability risks. Use in custom applications also requires special consideration.

**Tip 1:** Oracle Health Insurance recommends the use of *Oracle Resource Management.*
The script `OZGI002S.sql` can be used to set a standard configuration.
Set the following database parameter:

```
RESOURCE_MANAGER_PLAN="SYSTEM_PLAN"
```

**Tip 2:** Oracle Health Insurance recommends the use of the `WE8ISO8859P15` character set, which supports the use of the Euro symbol €.
This character set can be chosen when creating the database.
To change the character set of an existing database, see

Migration of database character set

## Tablespaces

The tablespaces below must be created for data and indexes:

| Tablespace | Initial | Next |
|---|---|---|
| STG_STAD | 128 Kb | 128 Kb |
| STG_STAI | 128 Kb | 128 Kb |
| STG_DYND | 4 Mb | 4 Mb |
| STG_DYNI | 4 Mb | 4 Mb |
| DWH_STAD | 128 Kb | 128 Kb |
| DWH_STAI | 128 Kb | 128 Kb |
| DWH_DYND | 4 Mb | 4 Mb |
| DWH_DYNI | 4 Mb | 4 Mb |
| OZG_DIM_SYS_TAB | 128 Kb | 128 Kb |
| OZG_DIM_SYS_IND | 128 Kb | 128 Kb |
| OZG_FACT_SYS_TAB | 128 Kb | 128 Kb |
| OZG_FACT_SYS_IND | 128 Kb | 128 Kb |
| OZG_LOG_TAB | 128 Kb | 128 Kb |
| OZG_LOG_IND | 128 Kb | 128 Kb |
| OWBSYS | 2048Mb | 32 Mb |

The created tablespaces must comply with the following requirements:

1. Locally Managed
2. System managed extent allocation
3. Automatic Segment Space Management
4. 8K or 16K blocksize

☞ **Note 3:** Oracle Health Insurance requires the use of a *default temporary* tablespace for temporary segments.

## Database users

The following users should be created in the Oracle Health Insurance Datamarts database:

| User | ID | Description |
|---|---|---|
| Oracle Health Insurance Datamarts repository owner | OWBSYS | This is the owner of the OWB repository on the Oracle Health Insurance Datamarts database server. This user is created during the installation of the OWB 11g software. The data/index tablespace is OWBSYS. |
| Oracle Health Insurance Datamarts Workspace owner | OHI_BI_WS_OWN | This is the owner or the workspace in which the Oracle Health Insurance Datamarts objects have been created and are executed. |
| Oracle Health Insurance Datamarts owner | OBD_OWN | This is the owner of the Oracle Health Insurance Datamarts objects. **This user must be created using the Oracle Health Insurance Datamarts installation software** with USERS as data/index tablespace |
| Batch User | BATCH | This is the user with which the Batch Scheduler scripts that are requested in the OHI Back Office application are executed. This user does not own any objects and therefore does not need its own separate tablespace. The user should be '**externally identified**' so that the loading process can be started remotely from the OHI Back Office application server. **This user is created by the Oracle Health Insurance Datamarts installation software.** (see 'Oracle Health Insurance Security Aspects' document on iProjects Files for security of the BATCH account) |

The following user should be created in the OHI Back Office database:

| User | ID | Description |
|---|---|---|
| Select user for extractions | OBD_SELECT_USER | This is the user that performs the selections on the OHI Back Office database. This user should be assigned the following privileges:<br><br>`CREATE SESSION`<br>`OBD_SELECT_ROLE`<br>`SELECT ON V_$DATABASE` |

## Database link

A database link should be created from the Oracle Health Insurance Datamarts database to the OHI Back Office database with the name SRC_OPENZORG. The link should be created in the following way (under user OBD_OWN) in the Oracle Health Insurance Datamarts database:

```
create database link SRC_OPENZORG
connect to OBD_SELECT_USER
identified by [password]
using '[servicename]'
;
```

## Database directories

For a number of external tables the following database directories need to be created under the OBD_OWN schema:

| Directory | Value | Description |
|---|---|---|
| OBD_INPUT | Value of $OZG_BASE | This is the location of the source files that are used for the external tables |
| OBD_LOG | /tmp | This is the location for the log, discard and bad files of the external tables. |

Example:

if $OZG_BASE = /ozg/app/oracle/product/Zorg/oton

```
create or replace directory OBD_INPUT
as '/ozg/app/oracle/product/Zorg/oton';


XCXXXX
```

## Default Secure Install

To avoid default passwords, and as general good security practice it is recommendable to change your passwords on a regular basis. This section describes how to change your passwords for the various accounts within the OHIBI environment.

The database user OHI_BI_WS_OWN is the Oracle Warehouse Builder user that owns the workspace. To change the password for this user;

1. log on the the OHIBI database as sys or system user:
   sqlplus sys/<PASSWORD>@<DATABASE> as sysdba

   SQL> alter user ohi_bi_ws_own identified by <NEW_PASSWORD>
   type exit to exit sqlplus
   SQL> exit

2. log on to the application server as the batch user:
   ssh batch@<HOSTNAME>

3. Set your environment
   . ozg_init.env <ORACLE_SID>

4. navigate to $OZG_ADMIN/
   cd $OZG_ADMIN

5. open ozg.conf.<ORACLE_SID> in vi
   vi ozg.conf.<ORACLE_SID>

6. Edit the entry rtr_pw
   rtr_pw=<NEW_PASSWORD>

7. Make sure the file ozg.conf.<ORACLE_SID> is not world readable

The database user OWBSYS is the system user for Oracle Warehouse Builder, to reset the password (see also MOS 1305938.1)

1. Connect to SQLplus as dba user and change the OWBSYS password using:

   SQL> alter user OWBSYS identified by <owbsys_new_password>;

2. Connect as OWBSYS and execute the following:

   SQL>@<Oracle_home>/owb/rtp/sql/set_repository_password.sql <owbsys_new_password>

3. Execute :

   SQL>@<Oracle_home>/owb/rtp/sql/service_doctor.sql

4. If the service doctor does not report any issue like "cannot connect user=OWBSYS host=hostname port=portnumber service_name=your_servicename" then you should be able to stop and start the service using start_service.sql/stop_service.sql.

NOTE: It is highly recommendable to restart the service with the start / stop scripts in the step above.

The database user OBD_OWN is the database user that actually owns the datamarts in the OHIBI database. The OBD_OWN password is used on the application server to log on to OHIBI. The password is also used in Oracle Warehouse Builder locations, therefore a change in the OBD_OWN requires multiple actions.

1. Connect to SQLplus as dba user and change the **OBD_OWN** password using:
   SQL> alter user **OBD_OWN** identified by <PASSWORD>;

2. Log on to the application server
   ssh batch@<hostname>

3. Set the environment:
   . ozg_init.env DB11G2

4. Navigate to your wallet
   cd network/admin/special
5. To list all wallet entries:
   mkstore -wrl . -listCredential
6. Modify the credentials:
   mkstore -wrl . -modifyCredential <ORACLE_SID>_INSTALL OBD_OWN <PASSWORD>
7. Check the connection string with:
   sqlplus /@<ORACLE_SID>_INSTALL
8. Follow the Appendix: OWB 11gR2 post-cloning process for OHI Business Intelligence step 8 up to step 11, and update the locations for OBD_OWN.

## Create a Default Password Policy

It is mandatory to have a password policy that conforms to these rules:

- Password length must be at least 8 characters
- Passwords must expire after 180 days
- After 10 failed login attempts, login must be suspended for 5 minutes.

Below is an example password policy function that may be used.

```
--
-- Security Profile
--
CREATE OR REPLACE
 FUNCTION ohibi_pass_validation(
      username     VARCHAR2,
      password     VARCHAR2,
      old_password VARCHAR2)
    RETURN BOOLEAN
  AS
  BEGIN
    IF LENGTH(password) < 8 THEN
      RETURN FALSE;
    ELSE
      RETURN TRUE;
   END IF;
END;
/
create PROFILE ohibi_profile LIMIT
FAILED_LOGIN_ATTEMPTS 3  -- Account locked after 3 failed logins.
PASSWORD_LOCK_TIME (1 / 24 / 60) * 5     -- Number of days account is
locked for. UNLIMITED required explicit unlock by DBA.
PASSWORD_LIFE_TIME 180    -- Password expires after 180 days.
PASSWORD_GRACE_TIME 3    -- Grace period for password expiration.
PASSWORD_REUSE_TIME 120  -- Number of days until a specific password
can be reused. UNLIMITED means never.
PASSWORD_REUSE_MAX 10    -- The number of changes required before a
password can be reused. UNLIMITED means never.
PASSWORD_VERIFY_FUNCTION ohibi_pass_validation;
/
```

All OHIBI account should use this profile.

## Enable Security Audit

To be able to track who tried to logon to the OHIBI database it is mandatory to have auditing enabled for this. Make sure your audit_trail is set to DB and then as DBA run the following stattement:

```
SQL> audit create session whenever not successful;


XXXXX
```

## General instructions

For instructions relating to active management of Oracle Health Insurance Datamarts, see:

Oracle Health Insurance Installation, Configuration and DBA Manual

(chapter 5)

# Installation of Oracle Warehouse Builder (OWB) Repository

## Client installation

Oracle Health Insurance Datamarts was developed with the help of OWB. The metadata relating to the Oracle Health Insurance Datamarts objects is stored in an OWB (Design) repository. This doesn't play a part in the use of Oracle Health Insurance Datamarts. A client installation is therefore not necessary.

## Server installation

### Software

From Oracle Database 11g Release 1, OWB is automatically installed in the same Oracle home as the database. This happens at the same time as the installation of the Oracle database software. As of release 2011.01, the OWB software only has to be present on the database server (see also Software Installation - Oracle Warehouse Builder).

### Repository

An OWB repository has to be present on the Oracle Health Insurance Datamarts database server in order to run the mappings generated by OWB. Information is stored in the repository relating to installation and running of mappings.

Installation of an OWB repository is done as follows:

1) **. ozg_init.env** <*ohi_db_name*>

2) **. ozg_init.env DB11G**

3) Set read (R) rights for OTHERS on the whole of $ORACLE_HOME/owb/bin/unix and explicit execution (X) rights for OTHERS on OMBPlus.sh and setowbenv.sh in the $ORACLE_HOME/owb/bin/unix directory.

4) **Login to the database with SYS (as SYSDBA)**

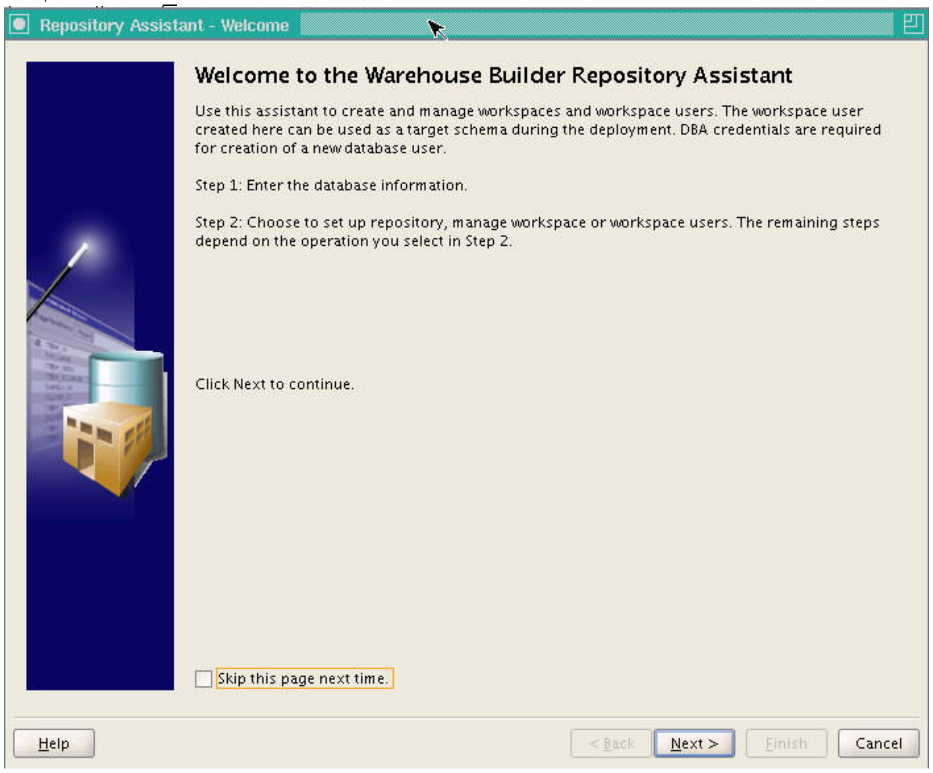
5) **create tablespace OWBSYS logging**
   **datafile '<*datafiledir*>/<*datafilename*>.dbf'**
   **size 2048M**
   **autoextend on**
   **next 32M**
   **extent management local;**

6) Install the new OWB 11g repository with:
   **@$ORACLE_HOME/owb/UnifiedRepos/cat_owb.sql OWBSYS**

7) Unlock the OWBSYS account and set a password:
   **alter user OWBSYS account unlock;**
   **alter user OWBSYS identified by <*password*>;**
   **grant restricted session to OWBSYS;**

   **Exit;**

8) **cd** $ORACLE_HOME/**owb/bin/unix**



Welcome to the Warehouse Builder Repository Assistant

Use this assistant to create and manage workspaces and workspace users. The workspace user created here can be used as a target schema during the deployment. DBA credentials are required for creation of a new database user.

Step 1: Enter the database information.

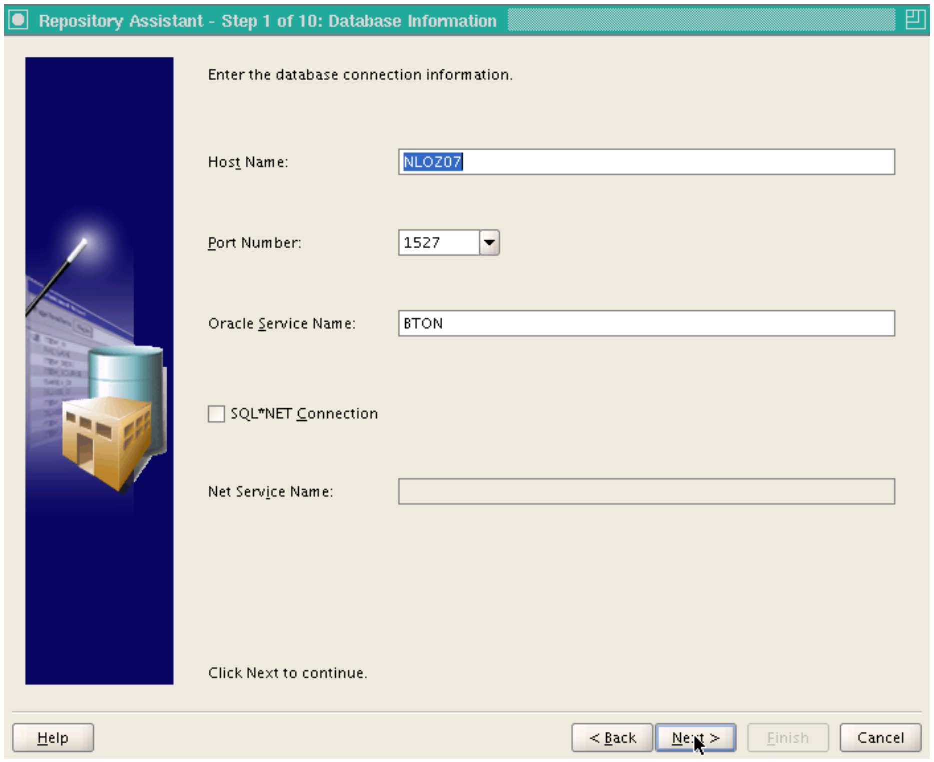
Step 2: Choose to set up repository, manage workspace or workspace users. The remaining steps depend on the operation you select in Step 2.

Click Next to continue.

☐ Skip this page next time.

9) **./reposinst.sh**

Click on Next>



Enter the database connection information.

Host Name: NLOZ07

Port Number: 1527

Oracle Service Name: BTON

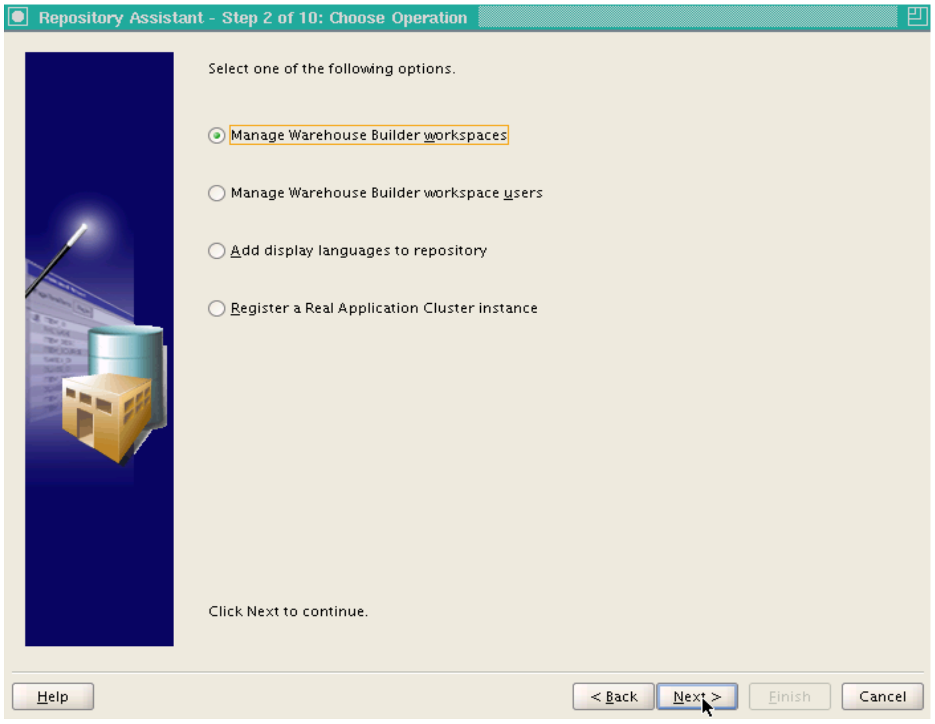
☐ SQL*NET Connection

Net Service Name:

Click Next to continue.

| | |
|---|---|
| Host Name: | Name of the host server |
| Port Number: | Port of the listener for the database |
| Oracle Service Name: | *<ohi_db_name>* |

Then click on Next>


Repository Assistant - Step 2 of 10: Choose Operation

Select one of the following options.

[✓ Manage Warehouse Builder workspaces]

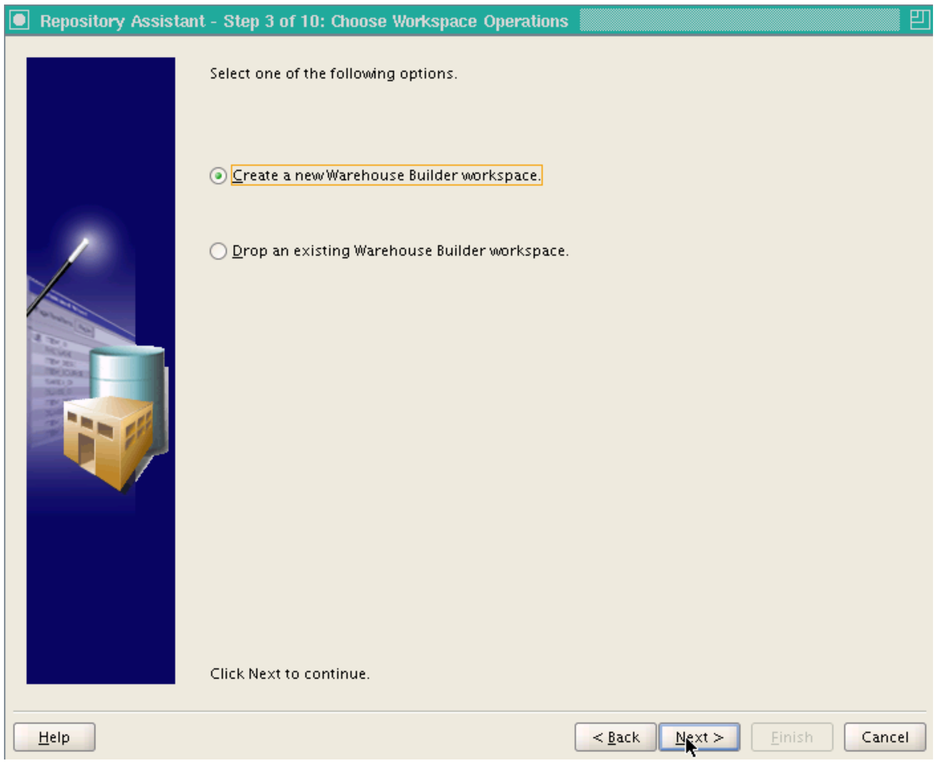
Manage Warehouse Builder workspace users

Add display languages to repository

Register a Real Application Cluster instance

Click Next to continue.

Choose "Manage Warehouse Builder workspaces" and click on Next>


Repository Assistant - Step 3 of 10: Choose Workspace Operations

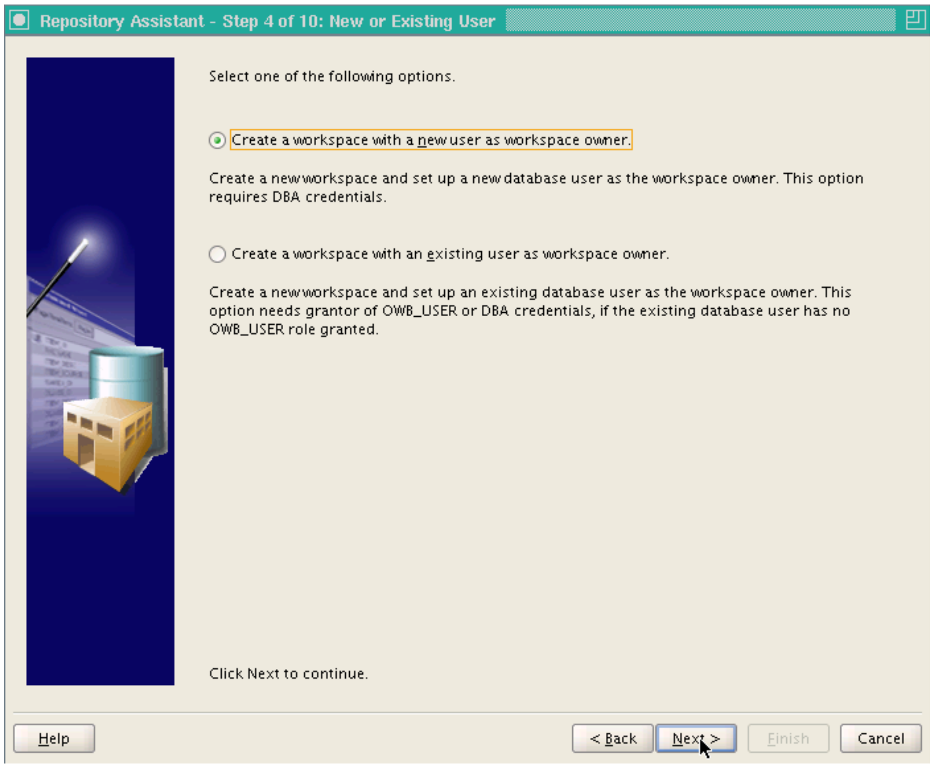
Select one of the following options.

[✓ Create a new Warehouse Builder workspace.]

Drop an existing Warehouse Builder workspace.

Click Next to continue.

Choose "Create a new Warehouse Builder workspace" and click on Next>


Repository Assistant - Step 4 of 10: New or Existing User

Select one of the following options.

(●) Create a workspace with a new user as workspace owner.

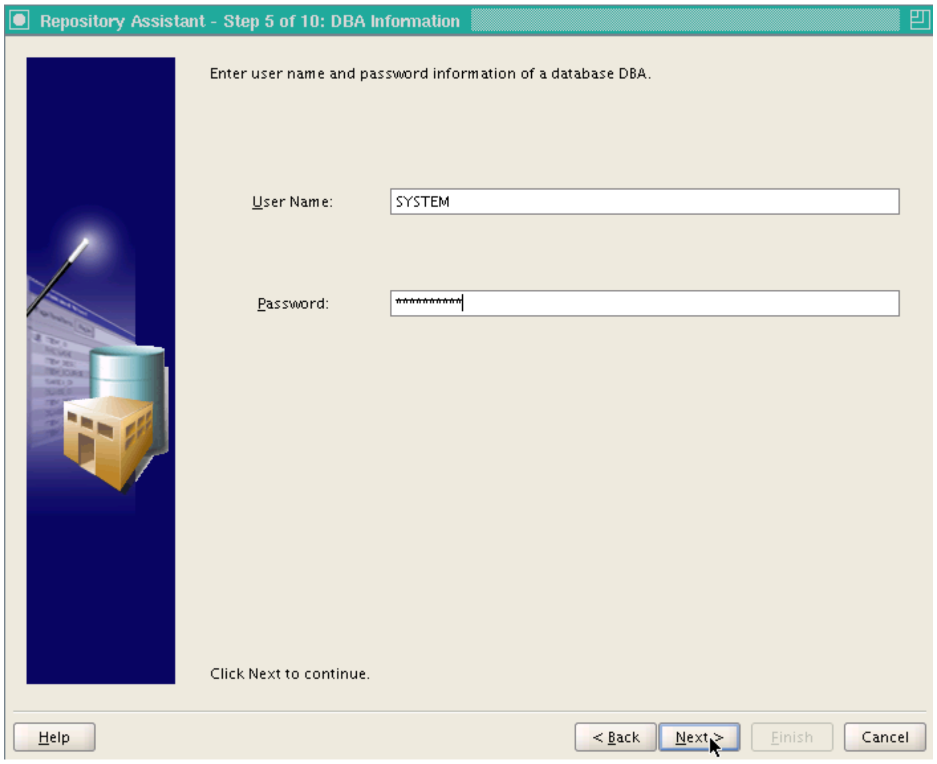
Create a new workspace and set up a new database user as the workspace owner. This option requires DBA credentials.

( ) Create a workspace with an existing user as workspace owner.

Create a new workspace and set up an existing database user as the workspace owner. This option needs grantor of OWB_USER or DBA credentials, if the existing database user has no OWB_USER role granted.

Click Next to continue.

Choose "Create a workspace with a new user as workspace owner" and click on Next>


Repository Assistant - Step 5 of 10: DBA Information

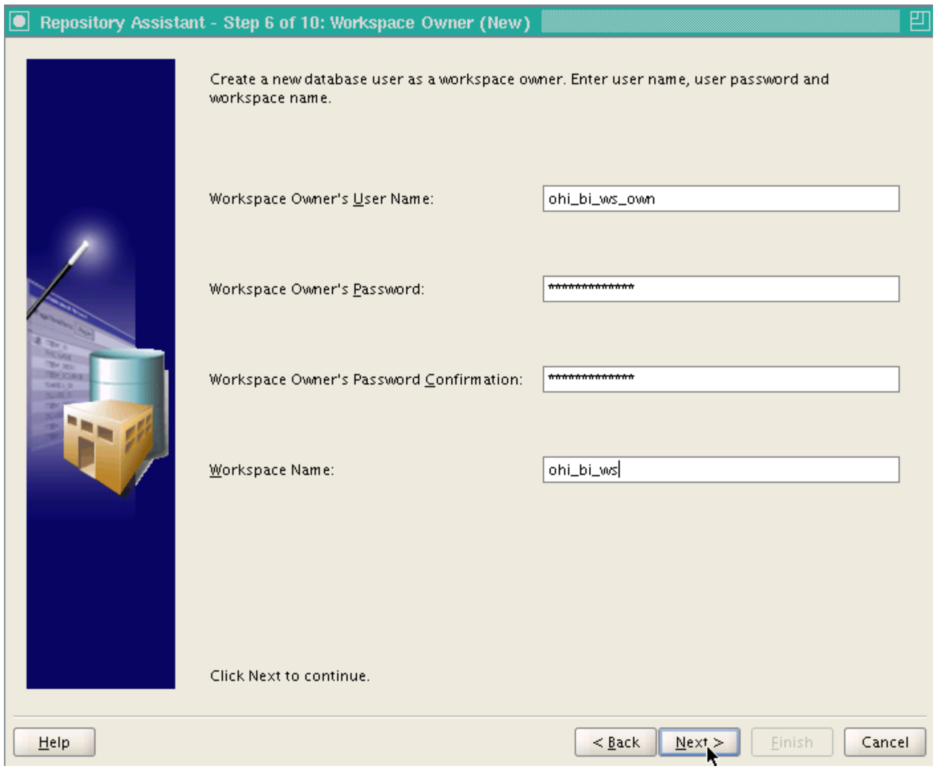
Enter user name and password information of a database DBA.

User Name: SYSTEM

Password: ************

Click Next to continue.

Enter the password for the SYSTEM user and click on Next>


Repository Assistant - Step 6 of 10: Workspace Owner (New)

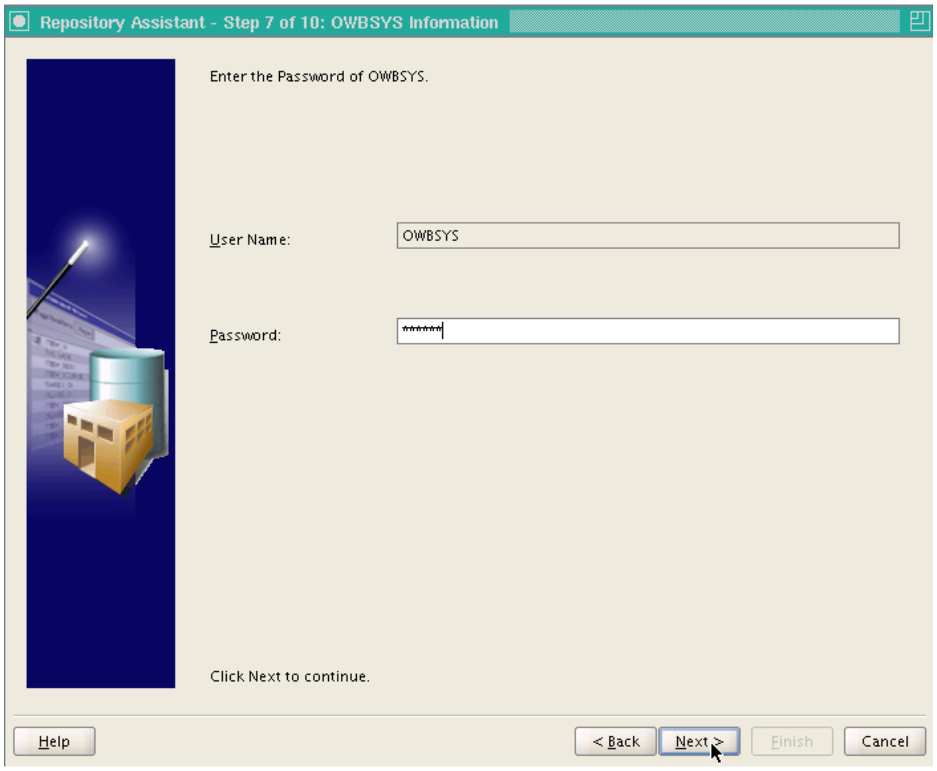
Create a new database user as a workspace owner. Enter user name, user password and workspace name.

| | |
|---|---|
| Workspace Owner's User Name: | ohi_bi_ws_own |
| Workspace Owner's Password: | ************** |
| Workspace Owner's Password Confirmation: | ************** |
| Workspace Name: | ohi_bi_ws |

Click Next to continue.

| | |
|---|---|
| Workspace Owner's User Name: | ohi_bi_ws_own |
| Workspace Owner's Password: | <password> |
| Workspace Owner's Password confirmation: | <password> |
| Workspace Name: | ohi_bi_ws |

Click on Next>


Repository Assistant - Step 7 of 10: OWBSYS Information

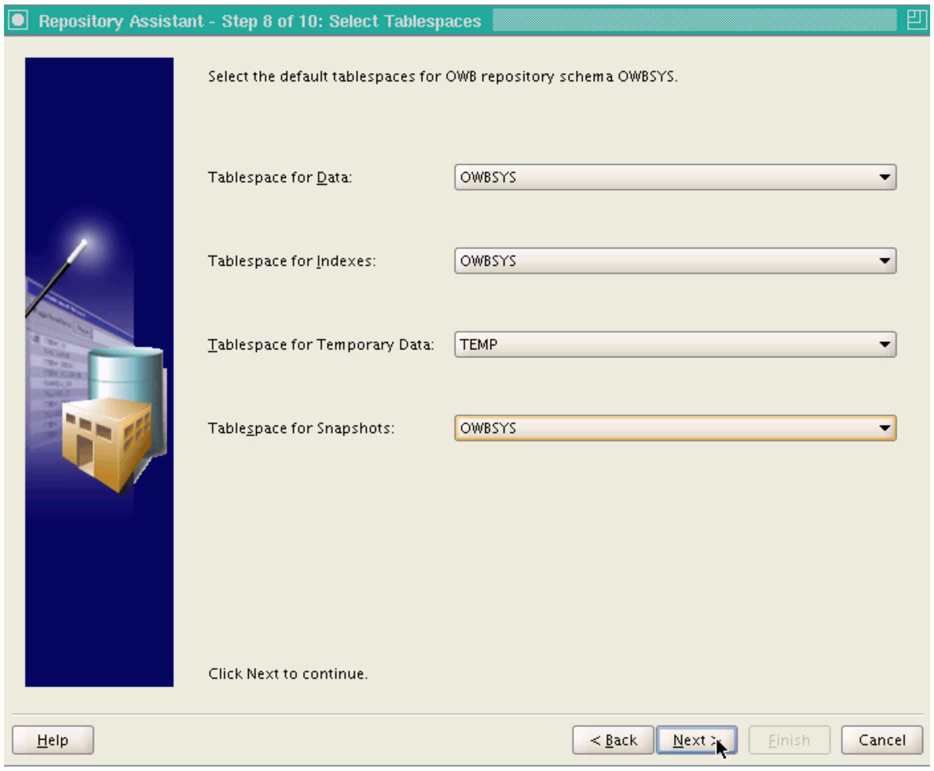
Enter the Password of OWBSYS.

| | |
|---|---|
| User Name: | OWBSYS |
| Password: | ******* |

Click Next to continue.

| | |
|---|---|
| User Name: | OWBSYS |
| Password: | <password> |

Click on Next>



Select the default tablespaces for OWB repository schema OWBSYS.

Tablespace for Data: OWBSYS

Tablespace for Indexes: OWBSYS

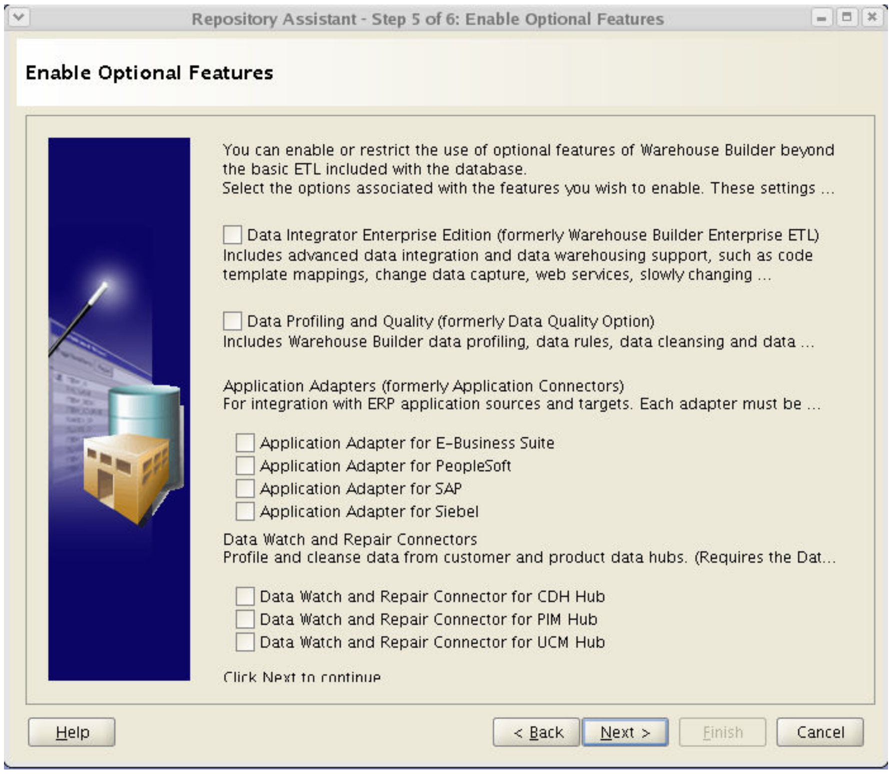
Tablespace for Temporary Data: TEMP

Tablespace for Snapshots: OWBSYS

Click Next to continue.

| | |
|---|---|
| Tablespace for Data: | OWBSYS |
| Tablespace for Indexes: | OWBSYS |
| Tablespace for Temporary Data: | TEMP |
| Tablespace for Snapshots: | OWBSYS |

Click on Next>


Repository Assistant - Step 5 of 6: Enable Optional Features
Help
< Back
Next >
Finish
Cancel

Enable Optional Features

You can enable or restrict the use of optional features of Warehouse Builder beyond the basic ETL included with the database.
Select the options associated with the features you wish to enable. These settings ...

☐ Data Integrator Enterprise Edition (formerly Warehouse Builder Enterprise ETL)
Includes advanced data integration and data warehousing support, such as code template mappings, change data capture, web services, slowly changing ...

☐ Data Profiling and Quality (formerly Data Quality Option)
Includes Warehouse Builder data profiling, data rules, data cleansing and data ...

Application Adapters (formerly Application Connectors)
For integration with ERP application sources and targets. Each adapter must be ...

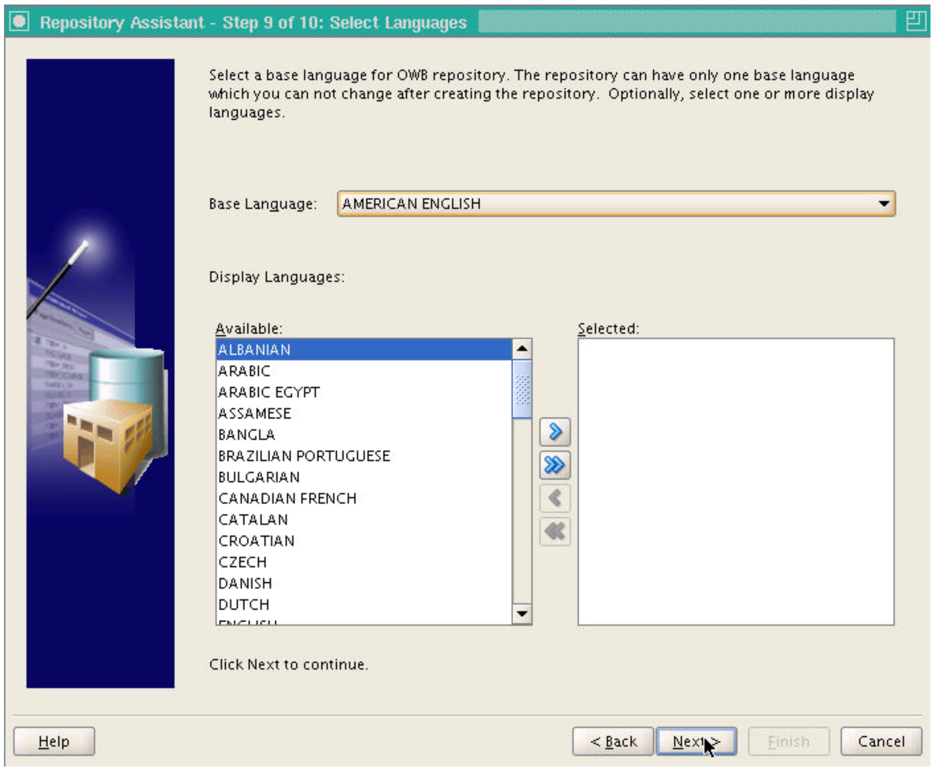
☐ Application Adapter for E-Business Suite
☐ Application Adapter for PeopleSoft
☐ Application Adapter for SAP
☐ Application Adapter for Siebel

Data Watch and Repair Connectors
Profile and cleanse data from customer and product data hubs. (Requires the Dat...

☐ Data Watch and Repair Connector for CDH Hub
☐ Data Watch and Repair Connector for PIM Hub
☐ Data Watch and Repair Connector for UCM Hub

Click Next to continue

Deselect all optional features.


Repository Assistant - Step 9 of 10: Select Languages

Select a base language for OWB repository. The repository can have only one base language which you can not change after creating the repository. Optionally, select one or more display languages.

Base Language: AMERICAN ENGLISH

Display Languages:

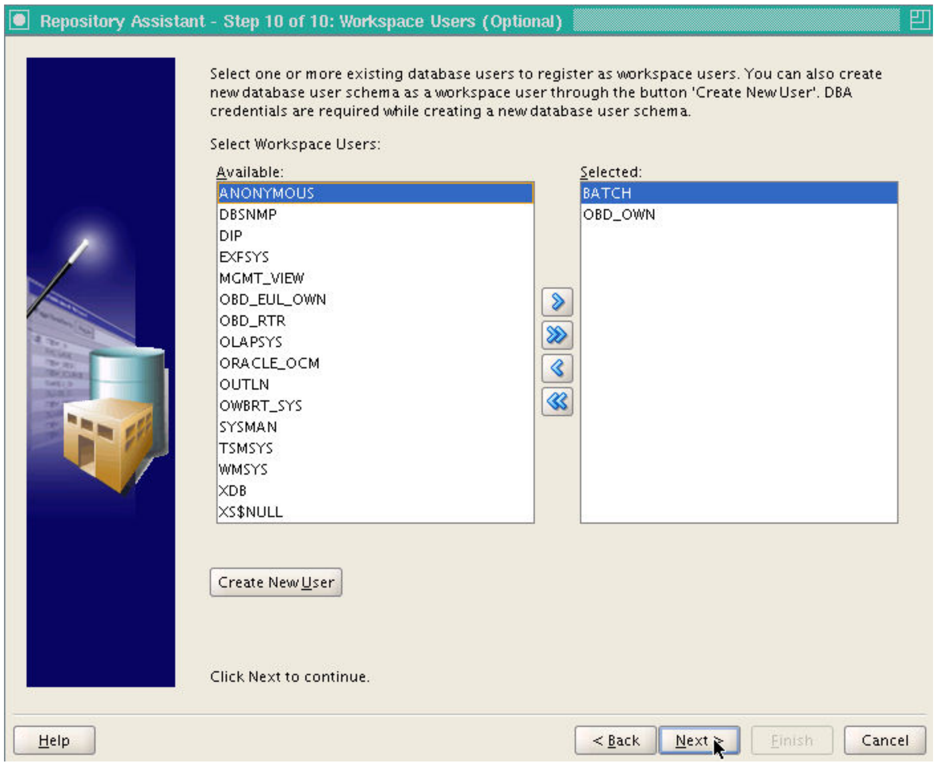
Available: ALBANIAN, ARABIC, ARABIC EGYPT, ASSAMESE, BANGLA, BRAZILIAN PORTUGUESE, BULGARIAN, CANADIAN FRENCH, CATALAN, CROATIAN, CZECH, DANISH, DUTCH

Selected:

Click Next to continue.

Base Language: AMERICAN ENGLISH

Click on Next>


Repository Assistant - Step 10 of 10: Workspace Users (Optional)

Select one or more existing database users to register as workspace users. You can also create new database user schema as a workspace user through the button 'Create New User'. DBA credentials are required while creating a new database user schema.

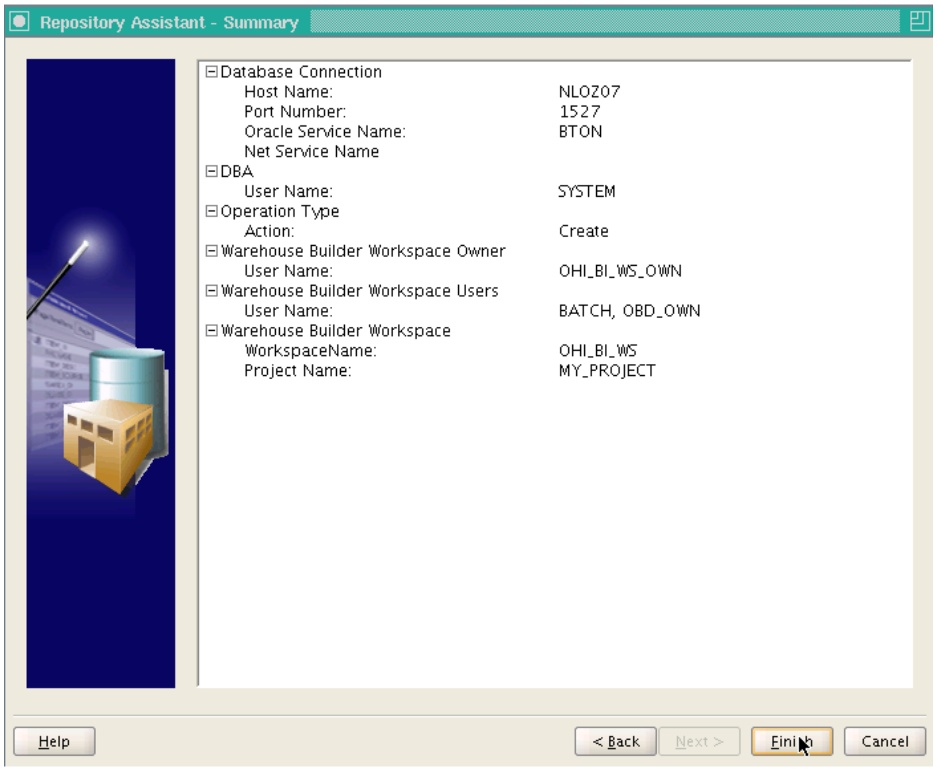
Select Workspace Users:

Available: ANONYMOUS, DBSNMP, DIP, EXFSYS, MGMT_VIEW, OBD_EUL_OWN, OBD_RTR, OLAPSYS, ORACLE_OCM, OUTLN, OWBRT_SYS, SYSMAN, TSMSYS, WMSYS, XDB, XS$NULL

Selected: BATCH, OBD_OWN

Click Next to continue.

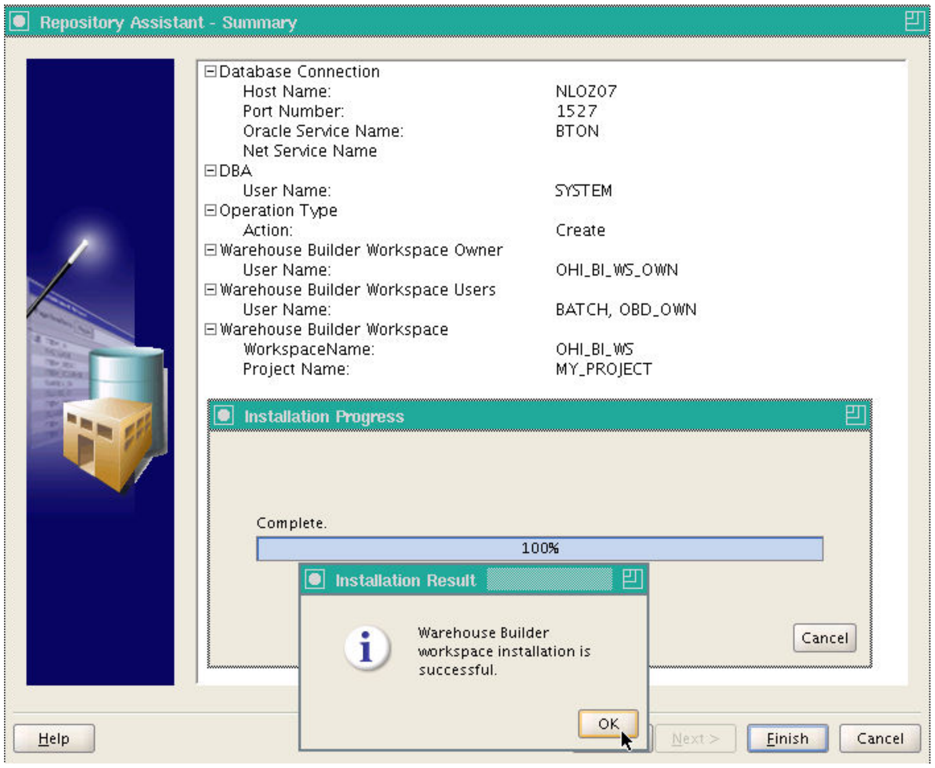
Add the users BATCH and OBD_OWN as Workspace Users if these already exist. If not, these users will have to be added later using /reposinst.sh. Click on Next>



| | |
|---|---|
| ⊟ Database Connection | |
| Host Name: | NLOZ07 |
| Port Number: | 1527 |
| Oracle Service Name: | BTON |
| Net Service Name | |
| ⊟ DBA | |
| User Name: | SYSTEM |
| ⊟ Operation Type | |
| Action: | Create |
| ⊟ Warehouse Builder Workspace Owner | |
| User Name: | OHI_BI_WS_OWN |
| ⊟ Warehouse Builder Workspace Users | |
| User Name: | BATCH, OBD_OWN |
| ⊟ Warehouse Builder Workspace | |
| WorkspaceName: | OHI_BI_WS |
| Project Name: | MY_PROJECT |

Check the displayed 'Summary'. This should also show that no optional features will be installed. If everything is correct, click on Finish.

The repository with the OHI_BI_WS_OWN workspace is now installed.



| | |
|---|---|
| ⊟ Database Connection | |
| Host Name: | NLOZ07 |
| Port Number: | 1527 |
| Oracle Service Name: | BTON |
| Net Service Name | |
| ⊟ DBA | |
| User Name: | SYSTEM |
| ⊟ Operation Type | |
| Action: | Create |
| ⊟ Warehouse Builder Workspace Owner | |
| User Name: | OHI_BI_WS_OWN |
| ⊟ Warehouse Builder Workspace Users | |
| User Name: | BATCH, OBD_OWN |
| ⊟ Warehouse Builder Workspace | |
| WorkspaceName: | OHI_BI_WS |
| Project Name: | MY_PROJECT |

Warehouse Builder workspace installation is successful.

Click on OK.

10) **sqlplus / as sysdba**

11) Assign grant from SYS to OHI_BI_WS_OWN.

    **grant restricted session to ohi_bi_ws_own;**

12) Assign grants from OWBSYS.
    **connect OWBSYS/<ww>**

    The following grants should be assigned to OHI_BI_WS_OWN.

    **grant select on wb_rt_service_nodes to ohi_bi_ws_own;**

    If the OBD_OWN user exists already, the following grants can be assigned. If the OBD_OWN user does not exist yet, this must be done later.

    **grant select on wb_rt_audit to obd_own;**
    **grant select on wb_rt_errors to obd_own;**
    **grant select on WB$_IV_CONTROL_CENTERS to obd_own;**
    **grant select on wb_rt_warehouse_objects to obd_own;**
    **grant execute on wb_workspace_management to obd_own;**

    **exit;**

13) Extract the files DWH_M2667_02.tcl and DWH_M2667_02_LOC.mdl from 2012.01.0.0000.zip (xml directory) and place these in the $OZG_BASE/xml directory.

14) Navigate to the $ORACLE_HOME/owb/bin/unix directory

15) ./OMBPlus.sh $OZG_BASE/xml/DWH_M2667_02.tcl

The installation of the OWB repository is now complete.

# GENERATION AND INSTALLATION OF ORACLE HEALTH INSURANCE DATAMARTS OBJECTS.

## INSTALLATION

### Release

Installation of Oracle Health Insurance Datamarts (patch) releases is described in the OHI Back Office Release Installation Manual and is performed on the application server.

ORACLE HEALTH INSURANCE Installation of releases

Because Oracle Health Insurance Datamarts is dependent on OHI Back Office as source environment, the patch level of both must be the same (this can differ at interim patch level if the patches between are only Oracle Health Insurance Datamarts patches or only OHI Back Office patches).
In addition, when installing patches the OHI Back Office environment must always be patched first, and then the Oracle Health Insurance Datamarts environment, otherwise packages may be invalidated during installation.

### System parameters

One of the supplied files is the SYS_PARAMETERS.dat file, which must be placed in the $OZG_BASE directory of the relevant Oracle Health Insurance Datamarts environment on the database server. This file contains the control variables for the Oracle Health Insurance Datamarts loading process. These parameters are customer-specific and the value of a number of these parameters **must** be adjusted in the file **before** Oracle Health Insurance Datamarts can be loaded. A further number may also be adjusted if desired.

The parameters that **must** be adjusted to ensure correct operation of the loading process:

| Parameter | Description |
|---|---|
| FINMOD | Indication (J/N [Yes/No]) of whether the financial module of OHI Back Office is used to load financial transactions. The default value is J [meaning yes], which means that Oracle Health Insurance Datamarts assumes that the financial data can be retrieved from OHI Back Office. |
| EOZNLS | The value that is given for this parameter depends on the NLS language setting. This determines, among other things, how a number is displayed. We distinguish between two different styles of notation. The following values indicate each style for the EOZNLS parameter.<br><br>1) Decimals separated by a comma<br>Thousands separated by a period<br><br>e.g.: 1.000.000,001<br>This style is used in the DUTCH NLS Language, among others.<br><br>Value for EOZNLS: nls_numeric_characters = ',.'<br><br>This is the default value for the parameter.<br><br>2) Thousands separated by a comma<br>Decimals separated by a period<br><br>e.g.: 1,000,000.001<br>This style is used in the AMERICAN NLS Language, among others.<br><br>Value for EOZNLS: nls_numeric_characters = '.,' |

There are a number of parameters that are not mandatory for the correct operation of the load run, but are required for functional population, namely:

| Parameter | Description |
|---|---|
| DCTYP1 | First type of third-party code |
| DCTYP2 | Second type of third-party code |
| DCTYP3 | Third type of third-party code |
| DCTYP4 | Fourth type of third-party code |
| DCTYP5 | Fifth type of third-party code |

This shows the third-party codes that should be loaded in the members dimension as alternative identifying codes. The values that have to be entered here are the codes of the 'code type' as shown in the 'Code' window in the relation management subsystem of OHI Back Office. The default value is empty.

A further 16 parameters can be included in SYS_PARAMETERS.dat, to also enable population of a number of procedure claim group flex fields for the authorization element, namely:

| Parameter | Description |
|---|---|
| VEWAE1 | The description of the first alphanumeric field of the care authorization element. |
| VEWAE2 | The description of the second alphanumeric field of the care authorization element. |
| VEWAE3 | The description of the third alphanumeric field of the care authorization element. |
| VEWAE4 | The description of the fourth alphanumeric field of the care authorization element. |
| VEWAE5 | The description of the fifth alphanumeric field of the care authorization element. |
| VEWAE6 | The description of the sixth alphanumeric field of the care authorization element. |
| VEWAE7 | The description of the seventh alphanumeric field of the care authorization element. |
| VEWAE8 | The description of the eighth alphanumeric field of the care authorization element. |
| VEWAE9 | The description of the ninth alphanumeric field of the care authorization element. |
| VEWAE0 | The description of the tenth alphanumeric field of the care authorization element. |
| VEWDE1 | The description of the first date field of the care authorization element. |
| VEWDE2 | The description of the second date field of the care authorization element. |
| VEWDE3 | The description of the third date field of the care authorization element. |
| VEWNE1 | The description of the first numeric field of the care authorization element. |
| VEWNE2 | The description of the second numeric field of the care authorization element. |
| VEWNE3 | The description of the third numeric field of the care authorization element. |

This shows the procedure claim group flex fields that should be loaded in the care authorization fact. The values that have to be entered here are the descriptions of the procedure claim group flex fields as shown in the 'Procedure claim group flex field' window in the Back Office subsystem of OHI Back Office. Here, choose either 'Message' or 'Authorization' as Usage Type. The default value is empty. Message/authorization

The other parameters concern descriptions of unknown values, for example, that **can** be adjusted if desired.

In addition a number of date system parameters have been added to the table DWH_SYS_PARAMETERS (which are not in the file) that are used to determine from which date a number of fact tables should be loaded.

| Parameter | Description |
|---|---|
| DCEVDM | Date from for load run of Claim fact. |
| PREVDM | Date from for load run of Premiums fact. |
| VZEVDM | Date from for load run of Members fact. |
| ZVNVDM | Date from for load run of Care authorizations fact. |
| OHWVDM | Date from for load run of Work in Progress Claim fact. |

These dates have an initial value of '01-01-1980', which means that this is the date from for the listing of data when loading for the first time. If there is older data that also needs to be loaded, then the relevant date has to be adjusted once in the table.

Finally, there is the parameter INDDCA that specifies whether the aggregation of claims in table DWH_DECLARATIES_AGG should be performed. This parameter should be set manually to 'J' [meaning yes] or 'N' [meaning no].

## External files

Three external data files are defined within Oracle Health Insurance Datamarts. This relates to the following files:

- SYS_PARAMETERS.dat
  File with control variables for the Oracle Health Insurance Datamarts load run (see previous paragraph).

| | |
|---|---|
| CODE (PK) | VARCHAR2(6) |
| OMSCHRIJVING | VARCHAR2(100) |
| OMSCHRIJVING_EN | VARCHAR2(100) |
| WAARDE_CHAR | VARCHAR2(80) |
| WAARDE_NUM | NUMBER |
| WAARDE_DATE | DATE "DDMMYYYY" |

- LEEFTIJD_CATEGORIEEN.dat:
  File with a number of age categories, e.g. youth or senior citizen.

| | |
|---|---|
| LEEFTIJD (PK) | NUMBER |
| CAT_VIJF | VARCHAR2(30) |
| CAT_TIEN | VARCHAR2(30) |
| CAT_VIJFTIEN | VARCHAR2(30) |
| CAT_JEUGD | VARCHAR2(30) |
| CAT_PENSIOEN | VARCHAR2(30) |
| CAT_SPECIAAL | VARCHAR2(30) |
| CAT_LOGO | VARCHAR2(30) |

- POSTCODE_VERRIJKINGEN.dat:
  File with geographic data enhancements (e.g. disadvantaged neighborhood or DHV area code).

| | |
|---|---|
| POSTCODE_NR **(PK)** | NUMBER |
| ACHTERSTANDSWIJK | VARCHAR2(1) |
| ADVISEUR_BUITENDIENST | VARCHAR2(50) |
| VESTIGINGSMANAGER | VARCHAR2(50) |
| RAYONCODE | VARCHAR2(50) |
| CONSUMENTEN_MARKTREGIO | VARCHAR2(100) |
| DHV_REGIOCODE | VARCHAR2(10) |
| DHV_REGIO_OMSCHRIJVING | VARCHAR2(100) |
| WGR_REGIOCODE | VARCHAR2(10) |
| WGR_REGIO_OMSCHRIJVING | VARCHAR2(100) |
| OAD_CODE | VARCHAR2(10) |
| ZIP_CODE | VARCHAR2(10) |
| WZV_REGIOCODE | VARCHAR2(10) |
| WZV_REGIO_OMSCHRIJVING | VARCHAR2(100) |

- WBX_OBIEE_TRANSLATIONS.csv
  File with translations for Oracle Business Intelligence

| | |
|---|---|
| LANG_ID (PK) | VARCHAR2(2) |
| METADATA_OBJECT | VARCHAR2(4000) |
| MSG_NUM (PK) | VARCHAR2(4000) |
| MSG_TXT | VARCHAR2(4000) |

- DIM_EIGENSCHAPPEN.dat
  File with mapping definition of dimension properties into OHI Business Intelligence. More information about the configuration of this file can be found in the application management chapter.

```
DIMENSIE_TABEL (PK)           VARCHAR2(30)
DIMENSIE_KOLOM (PK)           VARCHAR2(30)
BRON_TABEL                    VARCHAR2(30)
EIGENSCHAP                    VARCHAR2(60)
```

- GEN_EIGENSCHAPPEN.dat
  File with mapping definition of claim properties into OHI Business Intelligence. More information about the configuration of this file can be found in the application management chapter.

```
GEBRUIKGROEP (PK)             CHAR(60)
EIGENSCHAP (PK)               VARCHAR2(60)
TABEL_SPECIFIEK               VARCHAR2(30)
KOLOM_SPECIFIEK               VARCHAR2(30)
TABEL_GENERIEK                VARCHAR2(30)
KOLOM_GENERIEK                VARCHAR2(30)
```

Templates of these files can be retrieved from iProjects Files (NL Oracle Health Insurance Public - OHI Releases (all products) - Release 2011.01 - Templates - Business Intelligence).

The files SYS_PARAMETERS.dat and LEEFTIJD_CATEGORIEEN.dat must be filled for correct operation of the Oracle Health Insurance Datamarts load run. The file POSTCODE_VERRIJKINGEN.dat may be left empty.

The files should be placed in the $OZG_BASE directory of the relevant Oracle Health Insurance Datamarts environment on the database server. This is a manual process.

The content of these files can be changed with the exception of the key fields **(PK)**.

When adjusting the files, spaces should be used and not tabs. To check that any changes have been made correctly and that the files have been placed in the correct location, select queries can be performed on the external tables that are populated by the files.

WBX_OBIEE_TRANSLATIONS.csv is used for Oracle Business Intelligence. The translation for all metadata is described here. The default language is set to English (LANG_ID = ‘en’) which means that in WBX_OBIEE_TRANSLATIONS.csv there should always be a translation for the English language. To limit the languages to be used in Oracle Business Intelligence set the AllowedLanguages parameter in instanceconfig.xml (e.g. <AllowedLanguages>en,nl</AllowedLanguages> as per documentation *Oracle Fusion Middleware System Administrator’s Guide for Oracle Business Intellgence Enterprise Edition, Localizing Oracle Business Intelligence*). When no language has been chosen yet, the language defaults to the browser language. To adjust the language as per login, modify the url (eg http://Server_Name:port_number/analytics/saw.dll?Dashboard&Lang=nl). Once new translations are made available by modifying the WBX_OBIEE_TRANSLATIONS.csv file, the Oracle Business Intelligence server needs to be restarted.

| **File** | **Table** |
|---|---|
| LEEFTIJD_CATEGORIEEN.dat | STG_LEEFTIJD_CATEGORIEEN_EXT |
| POSTCODE_VERRIJKINGEN.dat | STG_POSTCODE_VERRIJKINGEN_EXT |
| SYS_PARAMETERS.dat | STG_SYS_PARAMETERS_EXT |
| DIM_EIGENSCHAPPEN.dat | STG_DIM_EIGENSCHAPPEN_EXT |
| GEN_EIGENSCHAPPEN.dat | STG_GEN_EIGENSCHAPPEN_EXT |

# SET-UP OF ORACLE BUSINESS INTELLIGENCE ENTERPRISE EDITION ENVIRONMENT

Oracle Business Intelligence Enterprise Edition (OBI EE) is the best reporting tool to use.

## SOFTWARE INSTALLATION

For installation of OBI EE, reference is made to the installation documentation of this product.

For reports, end users can make use of the presentation layer of OBI EE. Reports can be made with Oracle BI Answers and these can then be shared using the Oracle BI Interactive Dashboard component.

For the right versions of the software, see:

Certifications matrix 10.13.1.0

## ORACLE BUSINESS INTELLIGENCE CUSTOMIZATIONS

Customization of OBI is supported, to support your company style. This includes but is not limited to logo, background color and font color, basically all HTML customization may be made. For detailed instructions on Customizing Oracle Business Intelligence see the White Paper. (HTTP://WWW.ORACLE.COM/TECHNETWORK/MIDDLEWARE/BI/CUSTOMIZING-ORACLE-BIEE-11G-176387.PDF)

## CREATING A NON PRIVILEGED USER: OBI_SELECT_USER

Based on the principle of "the least privileged" a user should be created with minimal privileges. The process of creating such a user consists of creating a database user and granting privileges to that user. The priviliges needed are obd_rol_select, which contains only select privileges on the warehouse tables. The create session privilege is required to be able to logon to the database. Access to the directory obd_input is required for accessing the translations stored in an external table.

This is done in SQLPlus, instructions as per below.

Log on to the OHIBI database and create the OBI_SELECT_USER

```
SQLPlus> create user obi_select_user identified by <password>;

SQLPlus> grant obd_rol_select to obi_select_user;

SQLPlus> grant create session to obi_select_user;

SQLPlus> grant read on directory obd_input to obi_select_user;
```

## INSTALLATION OF ORACLE HEALTH INSURANCE DATAMARTS REPOSITORY (RPD)

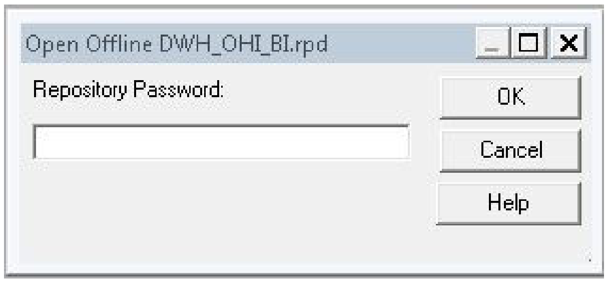
Oracle supplies an OBI EE repository as part of OHI Business Intelligence. This repository makes the OHI Business Intelligence database accessible. The repository can be installed on the Oracle BI Server.

This repository includes a subject area for each star schema as well as an over-arching subject area in which the entire data model has been made accessible.

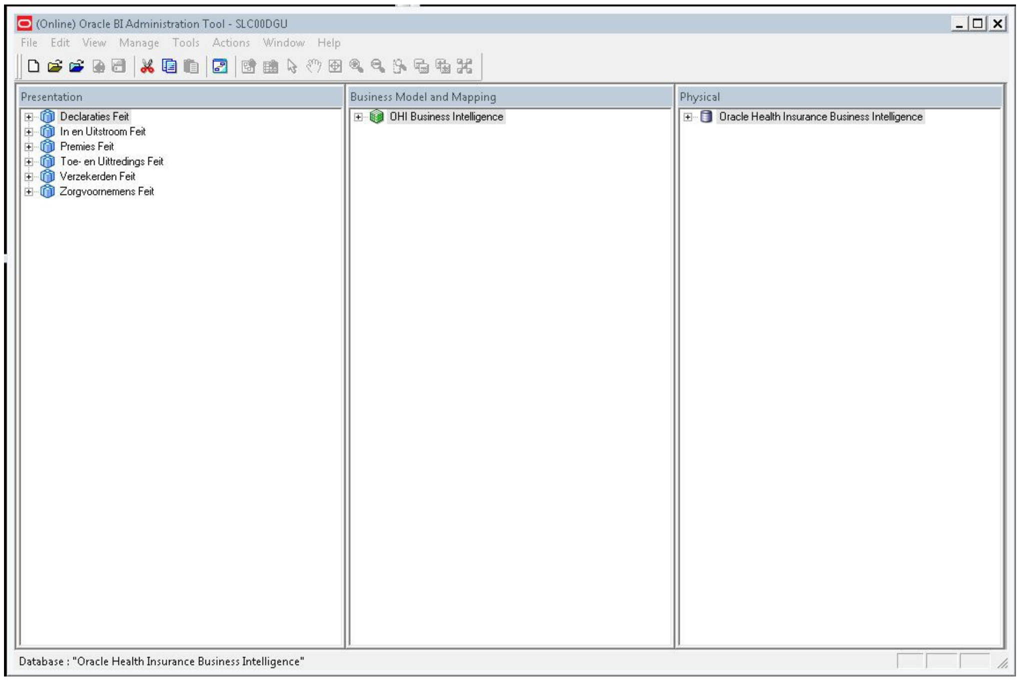
Before being able to use the DWH_OHI_BI.rpd file in your OBIEE environment, you will have to configure 3 connection pools using the Oracle BI Administration Tool.

Please follow the steps below to configure the 3 connection pools.

Open the DWH_OHI_BI.rpd offline using the Oracle BI Adminsitration Tool:

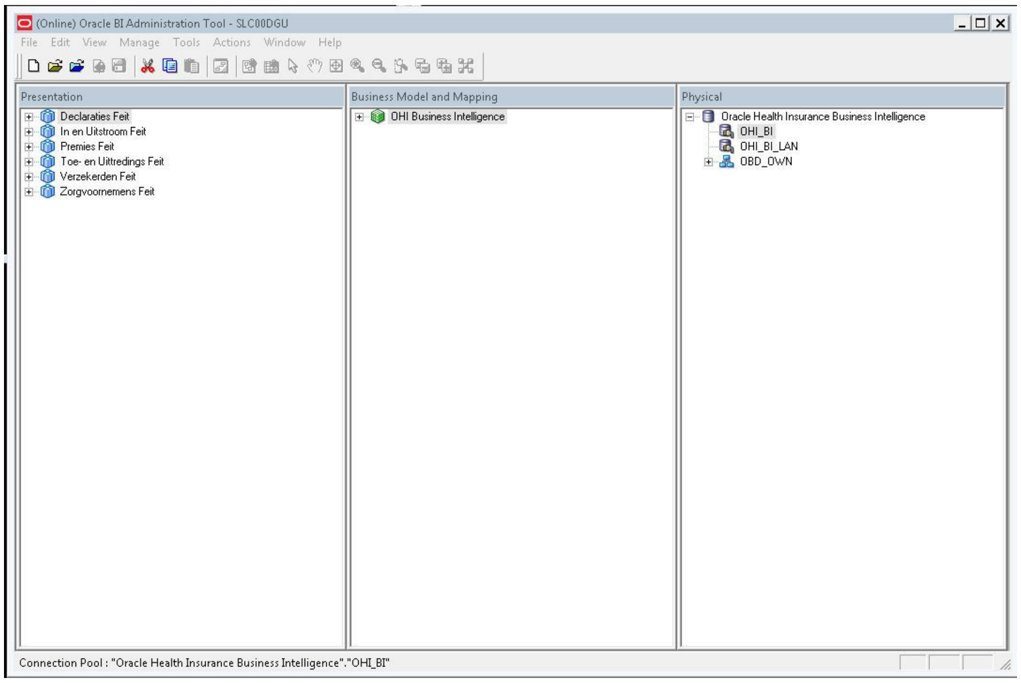


Enter adm1n1strator as the initial password.


(Online) Oracle BI Administration Tool - SLC00DGU
File Edit View Manage Tools Actions Window Help
Presentation
Declaraties Feit
In en Uitstroom Feit
Premies Feit
Toe- en Uittredings Feit
Verzekerden Feit
Zorgvoornemens Feit
Business Model and Mapping
OHI Business Intelligence
Physical
Oracle Health Insurance Business Intelligence

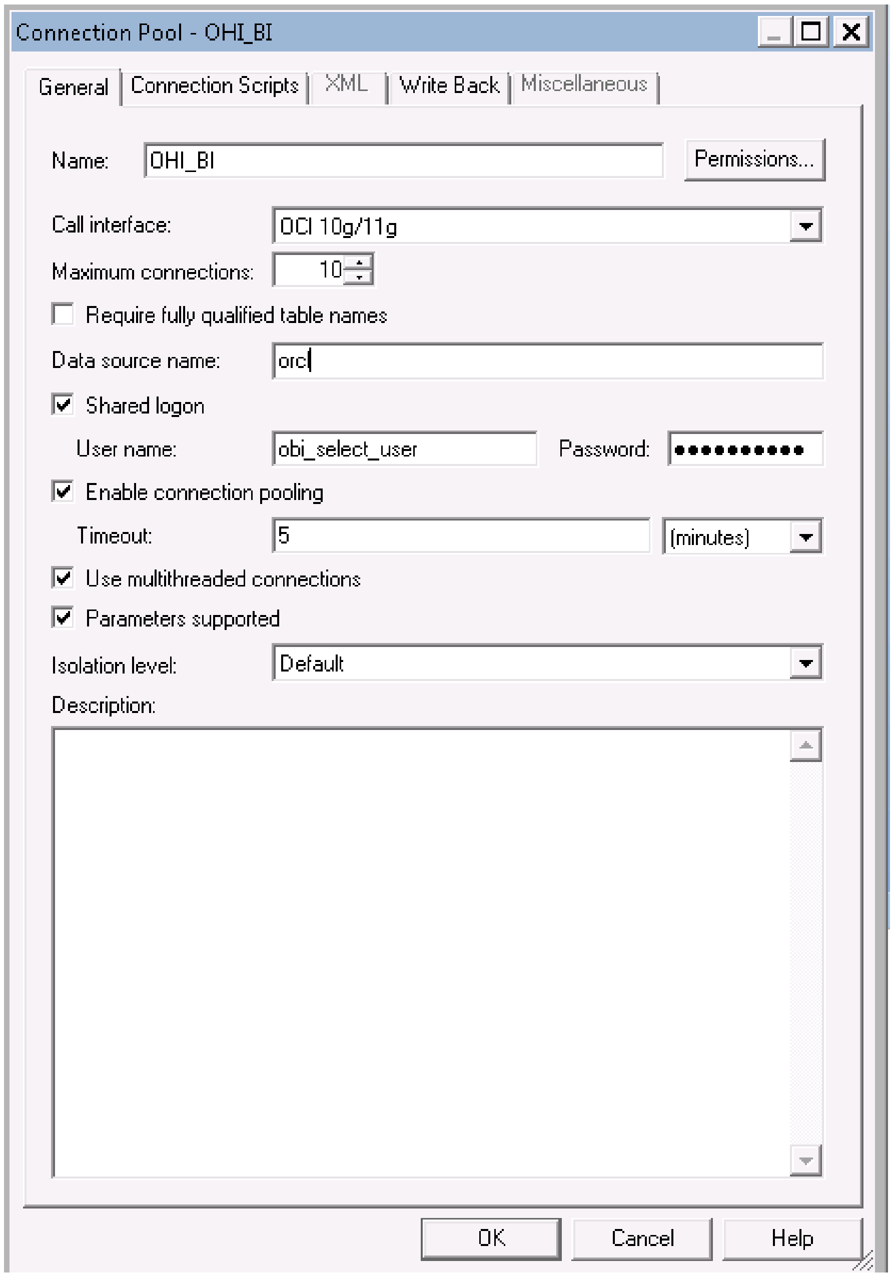
Database : "Oracle Health Insurance Business Intelligence"

Open the Oracle Health Insurance Business Intelligence node in the Physical layer.


(Online) Oracle BI Administration Tool - SLC00DGU
File Edit View Manage Tools Actions Window Help
Presentation
Declaraties Feit
In en Uitstroom Feit
Premies Feit
Toe- en Uittredings Feit
Verzekerden Feit
Zorgvoornemens Feit
Business Model and Mapping
OHI Business Intelligence
Physical
Oracle Health Insurance Business Intelligence
OHI_BI
OHI_BI_LAN
OBD_OWN

Connection Pool : "Oracle Health Insurance Business Intelligence"."OHI_BI"

Open the OHI_BI connection pool by double clicking OHI_BI.


Connection Pool - OHI_BI

General | Connection Scripts | XML | Write Back | Miscellaneous

Name: OHI_BI

Permissions...

Call interface: OCI 10g/11g

Maximum connections: 10

☐ Require fully qualified table names

Data source name: orcl

☑ Shared logon

User name: obi_select_user Password: ●●●●●●●●●●●

☑ Enable connection pooling

Timeout: 5 (minutes)

☑ Use multithreaded connections

☑ Parameters supported

Isolation level: Default

Description:

OK Cancel Help

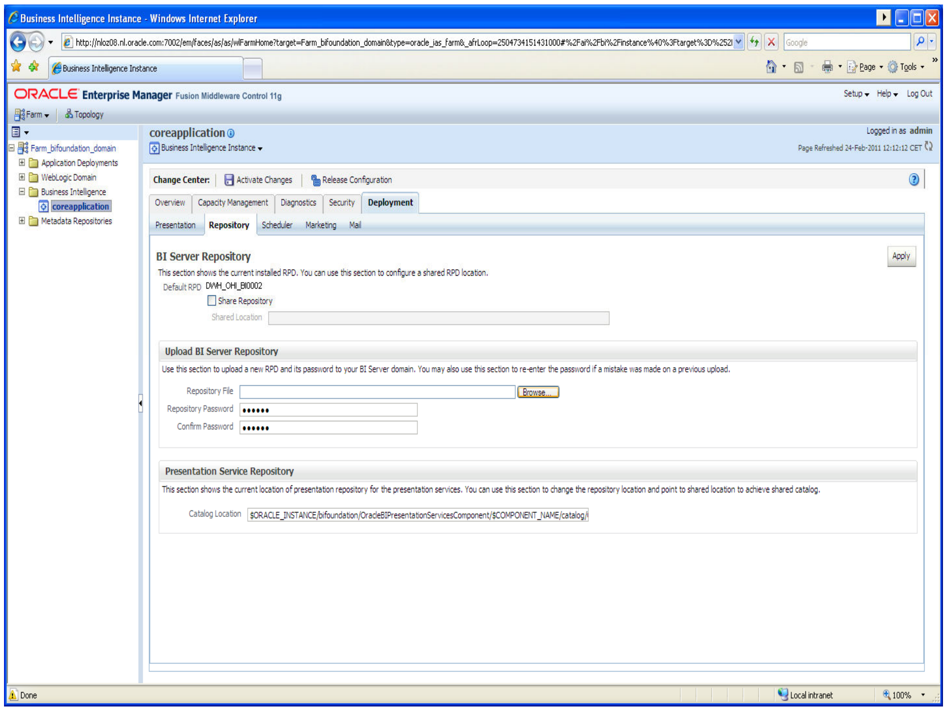
Change the data source name to your OHI Business Intelligence database.

The Connection pool OHI_BI_LAN is the connection pool used by translations of the repository to enable usage of OBI in your native language. This is not the same connection pool as OHI_BI due to the fact that the translation table may be stored on a different server as opposed to where the datamarts reside. To configure the connection pool for OHI_BI_LAN, execute the above steps for the OHI_BI_LAN connection pool.

The connection pool for Usage Tracking is by default the repository used by OBI and should be a different Oracle schema as the datamarts. With security in mind this would also typically be a different database as the database where the datamarts reside.  To configure the connection pool for Usage Tracking please execute the above steps for the OHI_BI_USAGE_TRACKING connection pool, and use the credentials of the OBI repository.

Save the DWH_OHI_BI.rpd.

After this you can import the DWH_OHI_BI.rpd in your OBI EE server environment using the Oracle Enterprise Manager (Fusion Middleware Control 11g) shown below.


Business Intelligence Instance - Windows Internet Explorer
http://nloz08.nl.oracle.com:7002/em/faces/as/as/wlFarmHome?target=Farm_bifoundation_domain&type=oracle_ias_farm&_afrLoop=2504734151431000#%2Fai%2Fbi%2Finstance%40%3Ftarget%3D%252l
Business Intelligence Instance
Page
Tools
ORACLE Enterprise Manager Fusion Middleware Control 11g
Setup
Help
Log Out
Farm
Topology
Farm_bifoundation_domain
Application Deployments
WebLogic Domain
Business Intelligence
coreapplication
Metadata Repositories
coreapplication
Business Intelligence Instance
Logged in as admin
Page Refreshed 24-Feb-2011 12:12:12 CET
Change Center:
Activate Changes
Release Configuration
Overview
Capacity Management
Diagnostics
Security
Deployment
Presentation
Repository
Scheduler
Marketing
Mail
Apply
Browse...
Done
Local intranet
100%

BI Server Repository

This section shows the current installed RPD. You can use this section to configure a shared RPD location.

Default RPD DWH_OHI_BI0002

Share Repository

Shared Location

Upload BI Server Repository

Use this section to upload a new RPD and its password to your BI Server domain. You may also use this section to re-enter the password if a mistake was made on a previous upload.

Repository File

Repository Password ••••••

Confirm Password ••••••

Presentation Service Repository

This section shows the current location of presentation repository for the presentation services. You can use this section to change the repository location and point to shared location to achieve shared catalog.

Catalog Location $ORACLE_INSTANCE/bifoundation/OracleBIPresentationServicesComponent/$COMPONENT_NAME/catalog/

In this window, under the heading "Upload BI Server Repository", choose the new DWH_OHI_BI.rpd file and enter the password. Enter the new password again for verification. The new repository will be imported and receives a new sequence number.

Note: this will cause any changes made to the repository to be removed.

## CONFIGURE USAGE TRACKING

To configure usage tracking see the documentation in "*Oracle® Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*", chapter 9 "*Managing Usage Tracking*".

As a reference below values are an example of Direct Insertion in $ORACLE_INSTANCE//config/OracleBIServerComponent/coreapplication_obis1/NQSConfig.INI

```
DIRECT_INSERT = YES;
PHYSICAL_TABLE_NAME = "Oracle Health Insurance Business
Intelligence"."OBI_BIPLATFORM"."S_NQ_ACCT";
CONNECTION_POOL = "Oracle Health Insurance Business
Intelligence"."OHI_BI_USAGE_TRACKING";
SUMMARY_STATISTICS_LOGGING = YES;
SUMMARY_ADVISOR_TABLE_NAME = "Oracle Health Insurance Business
Intelligence"."OBI_BIPLATFORM"."S_NQ_SUMMARY_ADVISOR";
```

## UPGRADING ORACLE BUSINESS INTELLIGENCE REPOSITORY

To upgrade the OBIEE repository, the procedure used, is refered to as a "Full Merge With a Common Parent with Binary Repositories". The documentation may be found in "Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition".

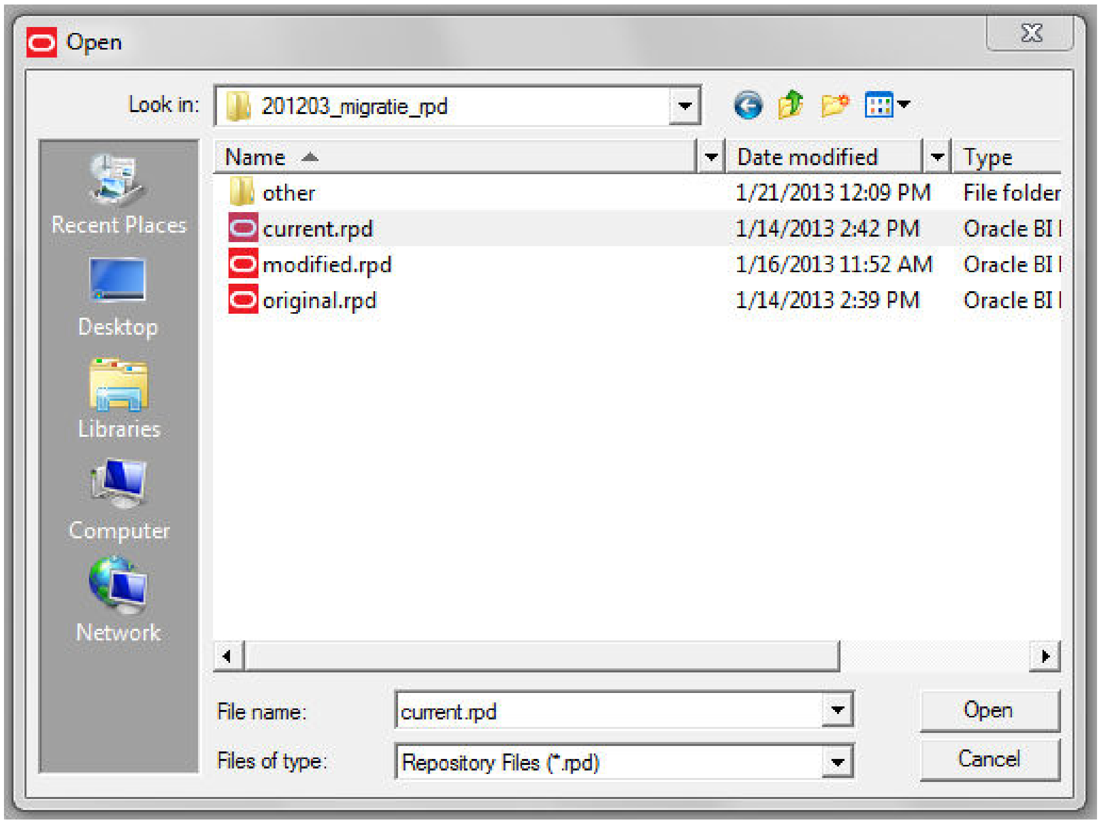
Definitions used in this topology:

- Orignal RPD: The rpd of the release which is migrated from.
  (This is the version provided in the previous release of OHI Business Intelligence and is unmodified by the Health Insurer, for instance of version 2012.03.01, in the example original.rpd)
- Modified RPD: The rpd of the release which is migrated to.
  (This is the version provided in the new release of OHI Business Intelligence and is unmodified by the Health Insurer, for instance of version 10.13.1.0.0, in the example modified.rpd)
- Current RPD: The rpd of the release of the customer where is migrated to
  (This is the client version modified i.e. the version of the Health Insurer of the current release for instance 2012.03.01, in the example current.rpd ).

In short the above mentioned procedure needs to be followed.

First Download and Install the Oracle Business Intelligence Developer Client Tools Installer. To download the correct version, please consult the certification matrix
(From OHI BI release 10.13.1.0.0 and further the required version for the Client tool is 11.1.1.6.5).

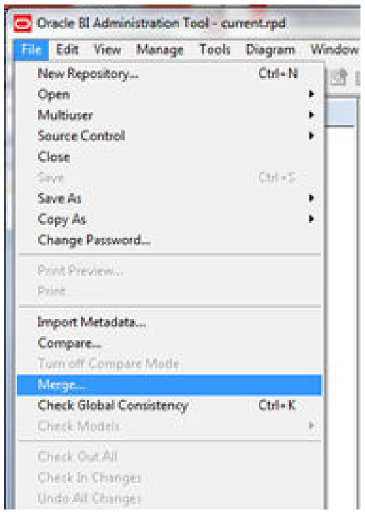
1. Open the current.rpd.



| Name | Date modified | Type |
|---|---|---|
| other | 1/21/2013 12:09 PM | File folder |
| current.rpd | 1/14/2013 2:42 PM | Oracle BI I |
| modified.rpd | 1/16/2013 11:52 AM | Oracle BI I |
| original.rpd | 1/14/2013 2:39 PM | Oracle BI I |

File name: current.rpd
Files of type: Repository Files (*.rpd)

2. File Merge

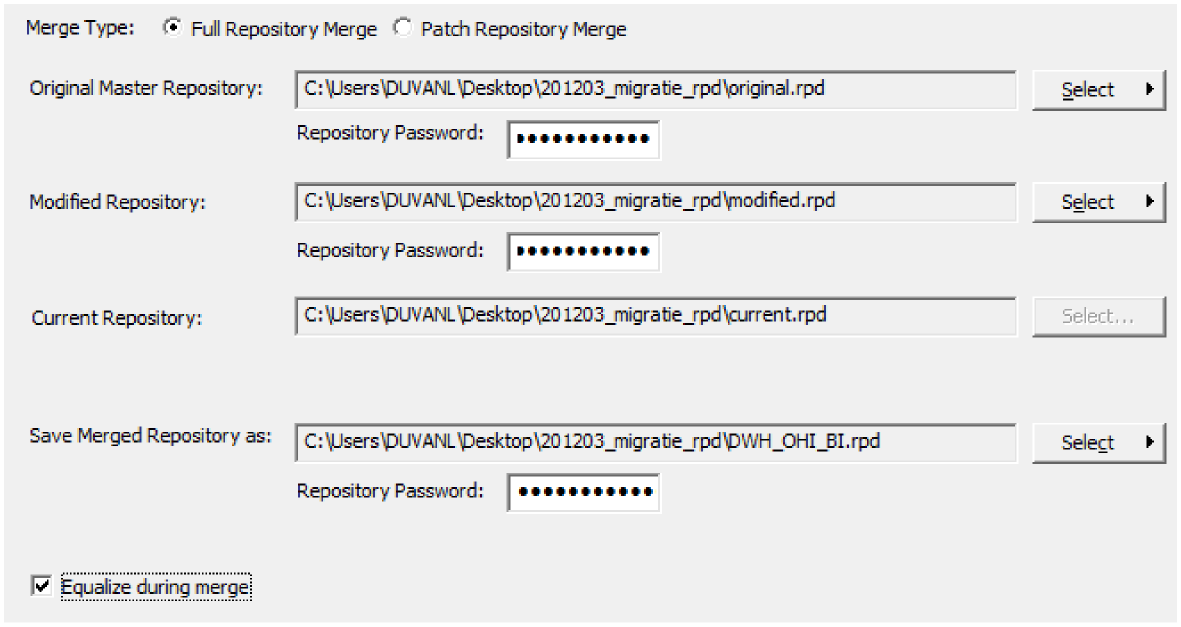


3. Tick equalize during merge

4. Open original and merge rpds

5. Type the passwords for original and modified repositories.

6. Save merged repository as: DWH_OHI_BI.rpd



Merge Type: [✓ Full Repository Merge] Patch Repository Merge

Original Master Repository: C:\Users\DUVANL\Desktop\201203_migratie_rpd\original.rpd
Repository Password: ●●●●●●●●●●●●

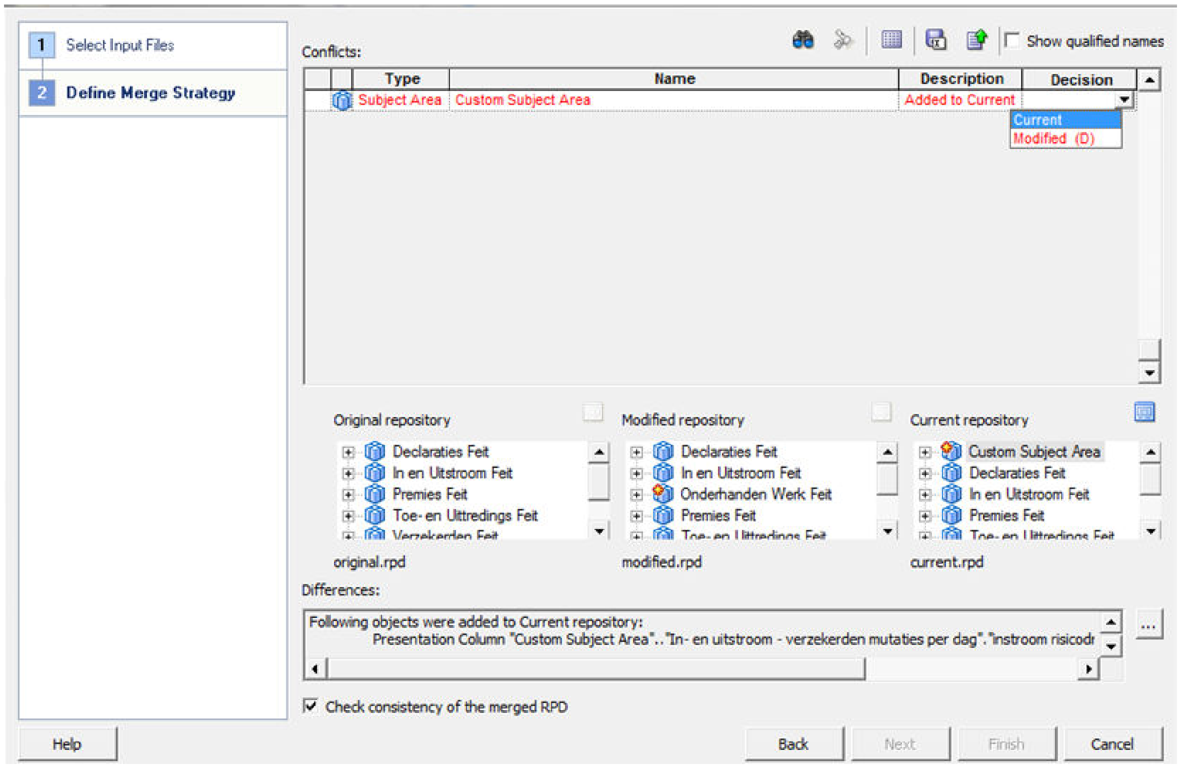
Modified Repository: C:\Users\DUVANL\Desktop\201203_migratie_rpd\modified.rpd
Repository Password: ●●●●●●●●●●●●

Current Repository: C:\Users\DUVANL\Desktop\201203_migratie_rpd\current.rpd

Save Merged Repository as: C:\Users\DUVANL\Desktop\201203_migratie_rpd\DWH_OHI_BI.rpd
Repository Password: ●●●●●●●●●●●●

[✓] Equalize during merge

7. Define Merge strategy:
   Tick Check Consistency of the merged RPD.



Conflicts:

| | Type | Name | Description | Decision |
|---|---|---|---|---|
| | Subject Area | Custom Subject Area | Added to Current | Current / Modified (D) |

| Original repository | Modified repository | Current repository |
|---|---|---|
| Declaraties Feit | Declaraties Feit | Custom Subject Area |
| In en Uitstroom Feit | In en Uitstroom Feit | Declaraties Feit |
| Premies Feit | Onderhanden Werk Feit | In en Uitstroom Feit |
| Toe- en Uitredings Feit | Premies Feit | Premies Feit |
| Verzekerden Feit | Toe- en Uittredings Feit | Toe- en Uittredings Feit |
| original.rpd | modified.rpd | current.rpd |

Differences:

```
Following objects were added to Current repository:
          Presentation Column "Custom Subject Area".."In- en uitstroom - verzekerden mutaties per dag"."instroom risicodr
```

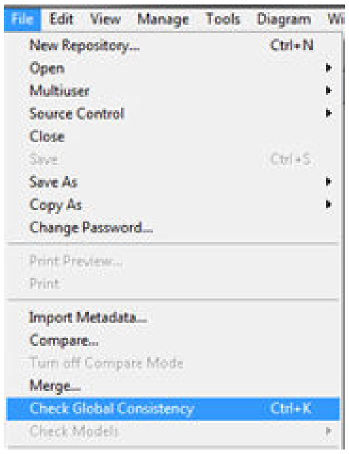
☑ Check consistency of the merged RPD

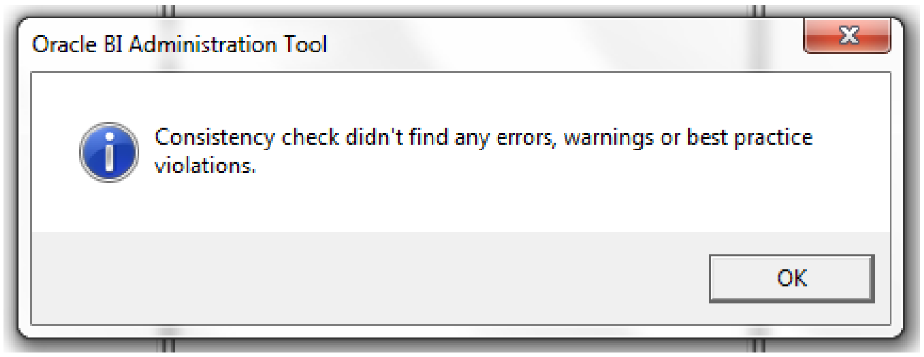
-- choose current to keep the changes in Current repository (Health Insurer RPD)

-- choose modified to keep the change In Modified repository (Oracle RPD) the (D) stands for delete, because this when choosing this it will actually remove the Custom Subject Area.

In the example current is chosen, to keep the Custom Subject Area in the newly to be generated PRD.

8. Review the results of the Consistency Check, and make sure there are no consistency errors.





Oracle BI Administration Tool

Consistency check didn't find any errors, warnings or best practice violations.

9. Check the connection details of the connection pools; while merging the two repositories the connection details will be overwritten.
10. Upload the newly created rpd DWH_OHI_BI.rpd to the OBIEE repository see "Upload BI Server Repository".

# Part II Loading

## Introduction

The full load run of Oracle Health Insurance Datamarts consists of three steps.

1. Population of the Oracle Health Insurance Datamarts Staging Area (STG) tables from the sources (OHI Back Office and external files) by script ZRGOE01S.
2. Transformation of the data in the Oracle Health Insurance Datamarts Staging Area (STG) tables via script ZRGOS01S.
3. Population of the Oracle Health Insurance Datamarts Data Warehouse Area (DWH) tables from the Staging Area using script ZRGOD01S.

The scripts above may **not** be run at the same time because the Data Warehouse section is dependent on the data in the Staging Area. There may well be dependencies specified in the OHI Back Office application so that all scripts can be requested but the next script can only be started once the previous script has terminated (successfully).

In this chapter the load run is described in detail.

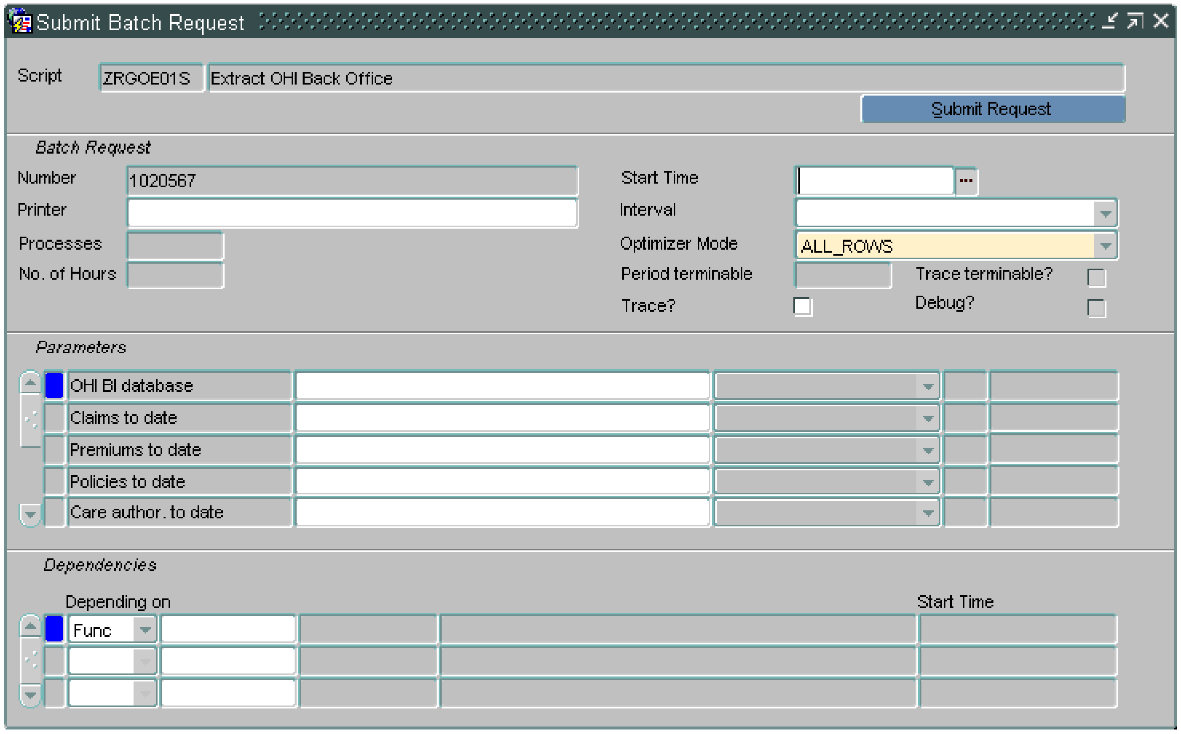
## Extract OHI Back Office (ZRGOE01S)

The first step is loading of the source data in the Staging Area (STG). The Staging Area is a data area to temporarily store and potentially process data from the OHI Back Office source system that is to be processed.

This data area therefore serves as a temporary buffer for storage of data and is not intended for queries by the end user. There are **no** relations made between the various tables that normally aim to be able to immediately perform a number of trivial integrity checks. These checks already take place in the source system and are therefore not necessary in Oracle Health Insurance Datamarts.

Starting up this load run happens by calling the script **ZRGOE01S** via the batch scheduler in the OHI Back Office application. The script runs a number of PL/SQL packages (generated by OWB) to populate a number of tables.

This script can be started simply from the OHI Back Office menu by selecting the menu option Oracle Health Insurance Datamarts → Extract OHI Back Office.



| | |
|---|---|
| Script | ZRGOE01S Extract OHI Back Office |
| Number | 1020567 |
| Printer | |
| Processes | |
| No. of Hours | |
| Start Time | |
| Interval | |
| Optimizer Mode | ALL_ROWS |
| Period terminable | |
| Trace terminable? | ☐ |
| Trace? | ☐ |
| Debug? | ☐ |

| | |
|---|---|
| OHI BI database | |
| Claims to date | |
| Premiums to date | |
| Policies to date | |
| Care author. to date | |

| Depending on | | | | Start Time |
|---|---|---|---|---|
| Func | | | | |

The parameters should be filled as follows.

- Fill in 'OHI BI database' with the alias of the Oracle Health Insurance Datamarts database (e.g. OACC).

  **Note: The script request is executed under the batch user, it is important that the database name is specified as defined in the secure external password store (wallet). This means the db_connect_string that is specified for the batch user on the Oracle Health Insurance Datamarts environment (e.g. oacc_batch). This db_connect_string should be specified as 'OHI BI database' parameter when submitting the ZRGOE01S.**

- Fill in 'Claim to date' with the date up to which the claims should be loaded on the basis of the approval date.
  If this field is left empty then the system date -1 day is used (default).

- Fill in 'Premium to date' with the date up to which the premiums should be loaded on the basis of the date and time of the status change.
  If this field is left empty then the system date -1 day is used (default).

- Fill in 'Policies to date' with the date up to which the Policies should be loaded on the basis of the modification date of the policy modification.
  If this field is left empty then the policies are loaded until the last modification timestamp.

  The ‘Policies to date’ is also used for the member mutations star schema. If the ‘Policies to date’ is empty then member mutations is loaded until the highest date -1 day that is loaded into the Policies starschema. If the ‘Policies to date’ is set and before the system date, member mutations are loaded until the set ‘Policies to date’.

- Fill in Care auth. to date' with the date up to which the Care authorizations should be loaded on the basis of the approval date or date terminated of the care authorization.
  If this field is left empty then the system date -1 day is used (default).

- Fill in ‘WIP to date' with the date up to which the Work in Progress Claims should be loaded on the basis of the registration date of the Work in Progress Claim.
  If this field is left empty then the system date -1 day is used (default).

- Fill in 'Data from time dim.' with the date from when the time dimension should be filled. The default value is '01-01-1980'.
- Fill in 'Nr of years of time dim.' with the number of years that should be filled in the time dimension. The default value is 40.
- For 'Claims?', 'Premiums?', 'In- and Out-flux?', 'Policies?', 'Care authorizations?' and ‘Work in Progress Claims?’ enter whether the relevant star schema should be loaded (J) or not (N=default).
- For 'Hospitalization?' enter whether the field ind_eerste_opname in dwh_declaraties should be updated after a load run. The default value is 'N'.
- For 'Perform checks?' enter whether data checks should be carried out when loading the data warehouse.
- For 'Check source?' enter whether the OHI Back Office source tables should be checked for integrity before extraction.
- For ‘Max. claim diff.’ enter the maximum amount that the claims crosschecks between OHI Back Office en OHI Business Intelligence may result in.
- For Suppress errors?' specify whether potential overridable errors from the previous load run (e.g. compatibility checks) can be generated.

Once the parameters have been filled in the script can be submitted to the Batch Scheduler with 'Submit request'.

A number of checks are first carried out before the extraction of data from OHI Back Office is started:

1. Did the last DWH load run (ZRGOD01S) terminate successfully? If this is not the case the extraction is not performed unless the value 'J' [meaning yes] has been entered for the parameter 'Override faults?'.
2. In the event that the claim star schema is loaded, it is checked whether all financial transactions have been duplicated in the OHI Back Office source environment. If this is not the case, the loading of claims is stopped.
3. In the event that the policies star schema is loaded, it is checked whether all policy events in the OHI Back Office source environment have been processed into modifications. If this is not the case, the load run is still started but it may occur that in the compatibility checks the number of members in Oracle Health Insurance Datamarts deviates from the number in OHI Back Office.
4. If the star schemas are loaded for which a date to can be specified then a check is performed on the specified date to value:
    - If the specified value is empty, the value of the date to parameter is set to the highest date -1 of the data being loaded. The date -1 is necessary because the data of the highest date may be an incomplete day.

      For example: if claims are loaded on May 27 and the highest approval date of the claims in OHI Back Office is May 25 then the date to is filled with May 24.
      If a value has been given for the date to parameter, the above check is also performed; in that case the final value is determined by the given or the found value, whichever is lower. However, a day is not subtracted unless the specified date is the same as the system date.

      For example: if claims are loaded on May 27, the date specified is May 26 and the highest approval date of the claims in OHI Back Office is May 25, then the date to is set to May 25.

      For example: if claims are loaded on May 27, the date specified is May 27 and the highest approval date of the claims in OHI Back Office is May 27, then the date to is set to May 26.

    - If the value is before the date to of the previous load run, then the load run is not performed for the fact concerned. This is reported in the extraction process log.

1. The current release of Oracle Health Insurance Datamarts is checked to see if it matches the current release of OHI Back Office. This check takes place at the patchset level; 2011.03.5, for example. If the releases do not match at the patchset level, it may not be loaded.
2. If the 'Check source?' value is set to 'J' [meaning yes], it is checked that statistics are present for all tables in OHI Back Offices that serve as a source for the load run and that all integrity rules have been activated and successfully validated. This concerns both the declarative constraints (PK, UK, FK and CHECK constraints, via USER_CONSTRAINTS) and procedural constraints (CDM Ruleframe via ALG#BUSINESS_RULES).

   If there are tables for which no integrity rules have been activated and successfully validated, then these are excluded. It is recommended to first correct the data in the source system and validate the integrity rules. It is possible to opt to continue loading; this is done by specifying a parameter.

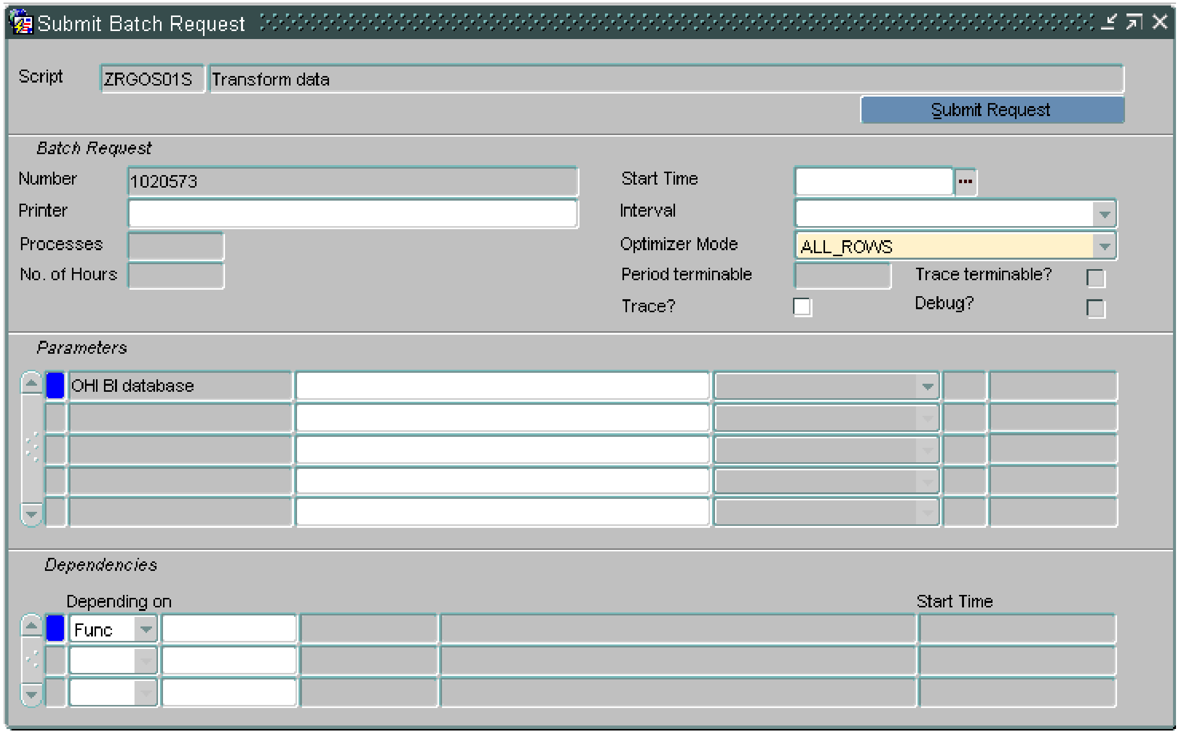
   It is checked whether the source tables in OHI Back Office actually have statistics. If no statistics are present, this may lead to inefficient execution scheduling during the extraction due to the fact that in that case 'dynamic sampling' is used. To avoid this, an error report is issued if there are tables without statistics.

## Transform data (ZRGOS01S)

The second step is the execution of a number of transformations in the Staging Area (STG). The Staging Area is a data area for temporarily transforming data that may be loaded in the extraction phase to the correct format and potentially processing it.

Starting this step happens by calling the script **ZRGOS01S** via the batch scheduler in the OHI Back Office application. The script runs a number of PL/SQL packages (generated by OWB) to enhance a number of tables with 'calculated' data and transfer the data structure to star schema.

This script can be started simply from the OHI Back Office menu by selecting the menu option Oracle Health Insurance Datamarts → Transform data.



Script: ZRGOS01S Transform data

Number: 1020573
Printer:
Processes:
No. of Hours:
Start Time:
Interval:
Optimizer Mode: ALL_ROWS
Period terminable:
Trace terminable? ☐
Trace? ☐
Debug? ☐

Parameters: OHI BI database

Depending on: Func
Start Time:

The name of the OHI BI database in which the transformations should be performed has to be specified as a parameter. Make sure that the OHI BI database name given here is the same as that specified for the batch user in the 'wallet'.

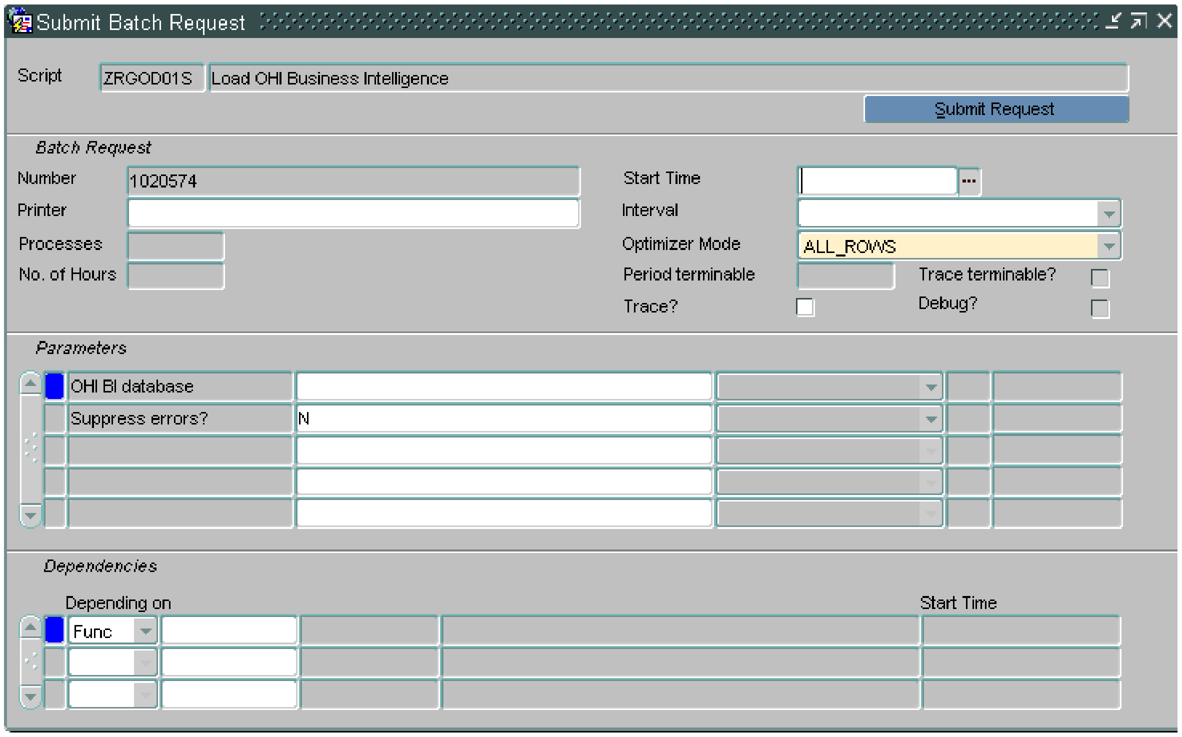
After the transformation step has run, a compatibility check between OHI Back Office and Oracle Health Insurance Datamarts will be performed. The results of this compatibility check are shown in the log of the ZRGOS01S script. More information on available checks can be found in the chapter **checks**.

## LOAD OHI BUSINESS INTELLIGENCE (ZRGOD01S)

The final step in the process is loading the data from the staging area into the Data Warehouse Tables. The Data Warehouse tables are intended for query and analysis purposes by the end user. The data model is set up as a dimensional model (star schema) and can contain both atomic data and aggregates. This model fits the way in which users approach the data (dimensions and facts).

Starting up the load run happens by calling the script **ZRGOD01S** in the OHI Back Office application. The script runs, among other things, a number of PL/SQL packages (generated by OWB) to populate the tables.

This script can be started simply from the OHI Back Office menu by selecting the menu option Oracle Health Insurance Datamarts → Load Oracle Health Insurance Datamarts.



Script: ZRGOD01S — Load OHI Business Intelligence

| | | | |
|---|---|---|---|
| Number | 1020574 | Start Time | |
| Printer | | Interval | |
| Processes | | Optimizer Mode | ALL_ROWS |
| No. of Hours | | Period terminable | |
| | | Trace? | ☐ |
| | | Trace terminable? | ☐ |
| | | Debug? | ☐ |

| | |
|---|---|
| OHI BI database | |
| Suppress errors? | N |

Depending on: Func — Start Time

As well as the name of the OHI BI database it should be specified as a parameter whether potential non-fatal errors arising during the transformation process (namely compatibility checks) may be overridden.

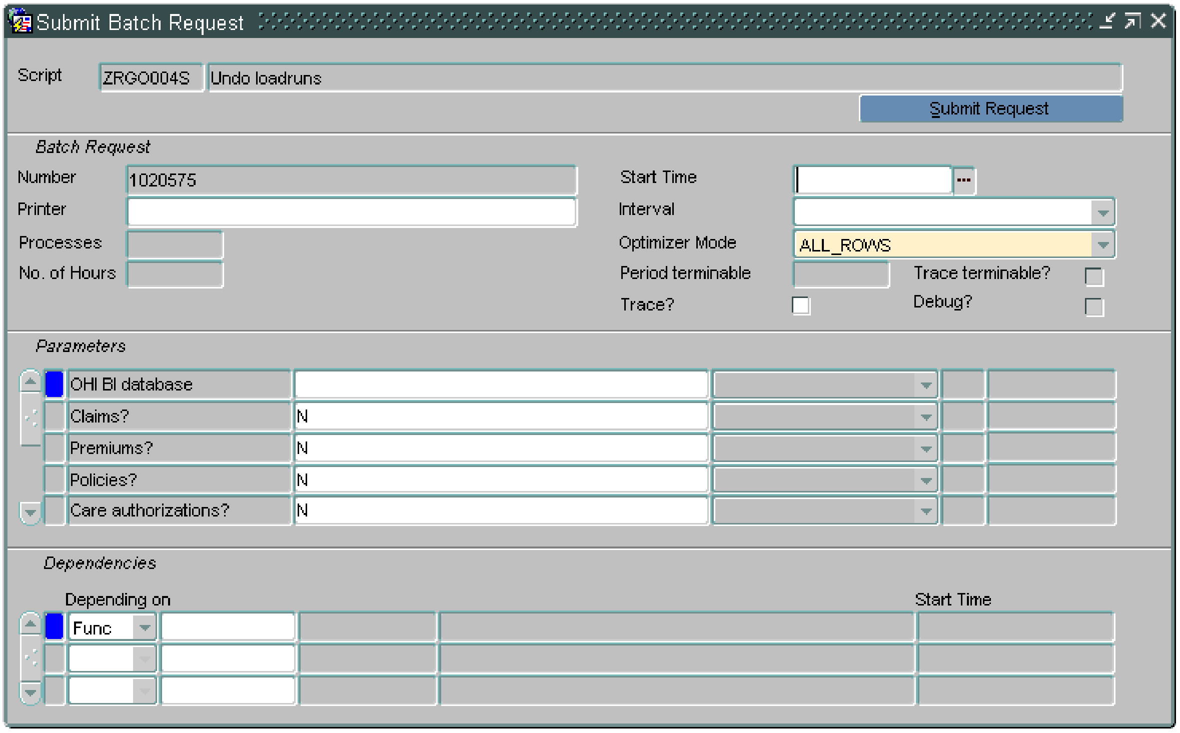
## UNDO LOADRUNS (ZRGO004S)

If, after refreshing Oracle Health Insurance Datamarts, it is discovered that errors have arisen and it is not possible to restore from a backup, the script 'Undo load runs' can be used for a specific star schema to remove one or more load runs from Oracle Health Insurance Datamarts at the same time. The registration of the date to on which the relevant star schema was refreshed is also restored. After running ZRGO004S the data can be reloaded using the regular load run (ZRGOE01S, ZRGOS01S and ZRGOD01S).

In the event of a software error or conversion, data usually only needs to be removed for one star schema. If, for example, first claims are loaded, then policies, and then claims again and the claims star schema appears to contain errors in the data, it is not desirable that the policies also have to be removed. However it is also possible to remove data for multiple star schemas at once.

With the script ZRGO004S load runs can be removed for the following star schemas:

1. Claims
2. Policies
3. Premiums
4. Care authorizations
5. Work in Progress Claims



| | |
|---|---|
| Script | ZRGO004S Undo loadruns |
| Number | 1020575 |
| Printer | |
| Processes | |
| No. of Hours | |
| Start Time | |
| Interval | |
| Optimizer Mode | ALL_ROWS |
| Period terminable | |
| Trace terminable? | ☐ |
| Trace? | ☐ |
| Debug? | ☐ |

| | |
|---|---|
| OHI BI database | |
| Claims? | N |
| Premiums? | N |
| Policies? | N |
| Care authorizations? | N |

Depending on: Func

Start Time

If the script 'Undo loadruns' is executed while the current loadrun has not yet terminated, the current loadrun is cancelled first.

☞ **Note:** If the script 'Undo load runs' is executed while the current load run has not yet terminated, the load run of ALL star schemas loaded during the current load run is canceled. Regardless of whether the star schemas were loaded correctly and regardless of the supplied script parameters.

## Script parameters

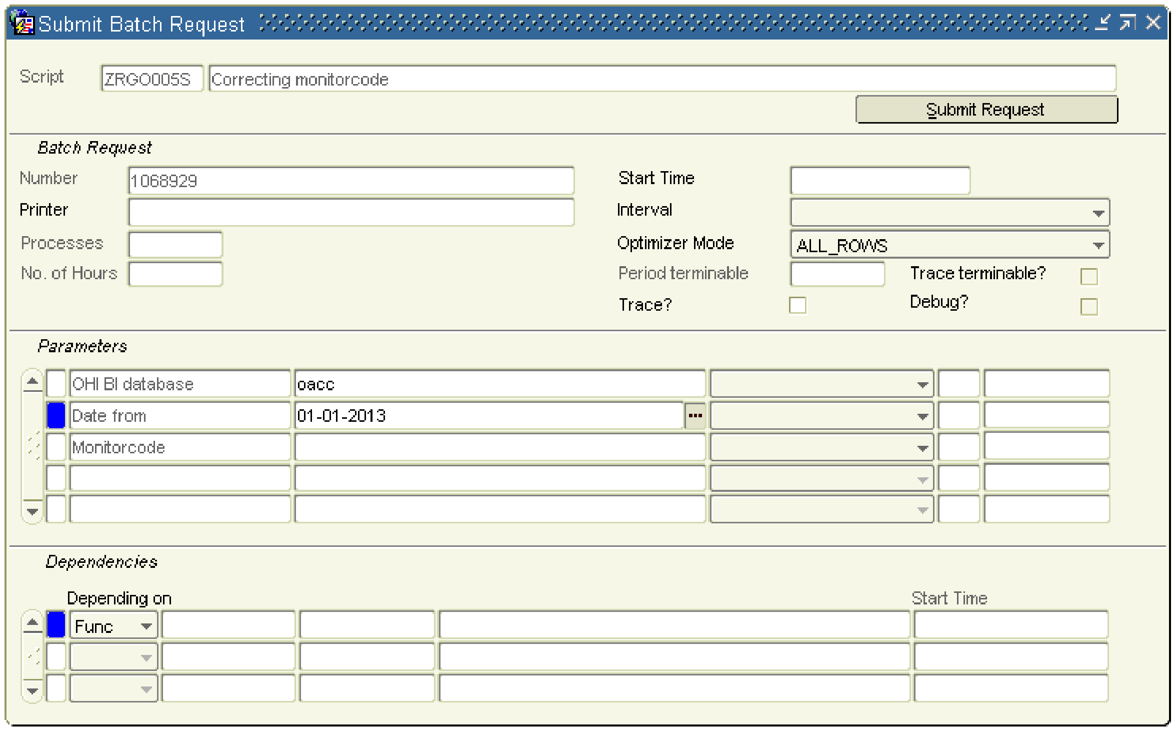
| Parameter | Values | Hint | Mandatory? |
|---|---|---|---|
| OHI BI Database | | Name of the Oracle Health Insurance Datamarts database | Y |
| Claims? | 'J', 'N' | Undo claims star schema load run(s)? | Y |
| Premiums? | 'J', 'N' | Undo premiums star schema load run(s)? | Y |
| Policies? | 'J', 'N' | Undo policies star schema load run(s)? | Y |
| Care authorizations? | 'J', 'N' | Undo care authorizations star schema load run(s)? | Y |
| WIP Claims? | 'J', 'N' | Undo Work in Progress star schema load run(s)? | Y |

| Parameter | Values | Hint | Mandatory? |
|---|---|---|---|
| Date | Default empty | Load runs on and after this date are removed. Empty = only remove the most recent load run. | Y |

## CORRECT MONITORCODE (ZRGO005S)

If the mapping of procedures on monitorcode in the source system OHI Back Office has been changed it might be necessary to correct the references to the Monitorcode dimension in the claims datamart in OHI Business Intelligence. Script ZRGO005S can be used for this purpose.

This script first updates the monitorcode dimensions and then corrects the incorrect references in the claims datamart for all claims with a DATUM_ACCOORD (finalized date) equal to or greater than the date parameter specified. If parameter Monitorcode is specified as well only claims that reference to that particular monitorcode will be corrected (if necessary).



Script: ZRGO005S Correcting monitorcode

Number: 1068929
Printer:
Processes:
No. of Hours:
Start Time:
Interval:
Optimizer Mode: ALL_ROWS
Period terminable:
Trace terminable?
Trace?
Debug?

OHI BI database: oacc
Date from: 01-01-2013
Monitorcode:

Func

### Script parameters

| Parameter | Values | Hint | Mandatory? |
|---|---|---|---|
| OHI BI Database | | Name of the Oracle Health Insurance Datamarts database | Y |
| Date from | Default empty | The monitorcode of all claims with DATUM_ACCOORD equal to or greater than this parameter will be corrected if necessary. | Y |
| Monitorcode | Default empty | If this parameter is filled only claims with this monitorcode will be corrected | N |

## CHECKS

At the end of each step any errors that may have arisen are printed in the log of the script request concerned.

If the script request parameter 'Perform check?' is set to 'J' [meaning yes] for the extraction process, then data checks are performed during the warehouse load run and after completion. These check whether particular data in Oracle Health Insurance Datamartss matches OHI Back Office. These checks are available for the claims (including aggregated claims), premiums, care authorization and policies star schemas. The totals and number of differences in OHI Back Office and Oracle Health Insurance Datamarts are printed in the log.

It is also possible to perform the star schema checks separately from the load run using script ZRGO003S ("Perform compatibility checks"). Here, the name of the Oracle Health Insurance Datamarts database needs to be specified as well as for which star schema(s) the check should be run. A period can also be specified for which it must be checked. If this period is not specified then the period read in during the last load run is checked.

The following checks are available:

| **Star schema** | **Check** | **Description** |
|---|---|---|
| Claims | DCACTL1 | Check aggregated claims |
| Claims | DCECTL1 | Check booked amount per date imported, general ledger account and claim number |
| Claims | DCECTL2 | Check booked amount per general ledger unit/account/category/product, date imported and accounting period |
| Claims | DCECTL3 | Check cost price per approval date and claim number |
| Claims | DCECTL4 | Check coverage amount per risk-bearing insurance company |
| Claims | DCECTL5 | Check deductible amounts per claim line |
| Premiums | PRECTL1 | Check gross premium amount per renewal date and member |
| Premiums | PRECTL2 | Check net premium amount per renewal date and member |
| Premiums | PRECTL3 | Check net premium amount per risk-bearing insurance company |
| Policies | VZECTL1 | Check number of members per branded product combination and member |
| Care authorizations | ZVNCTL1 | Check number per procedures per authorization period |
| Care authorizations | ZVNCTL2 | Check amount per procedure per authorization period |
| WIP Claims | OHWCTL1 | Check amount per Work in Progress claims line |
| WIP Claims | OHWCTL2 | Check number per Work in Progress claims line |

Individual checks can be turned on and off using procedures WBX_LAADPROCES_CONTROLES.ZET_CONTROLE_AAN and ZET_CONTROLE_UIT respectively. If all checks need to be turned on/off, this can be done using the procedures ZET_ALLE_CONTROLES_AAN and ZET_ALLE_CONTROLES_UIT respectively. These procedures must be performed under the OBD_OWN schema on the Oracle Health Insurance Datamarts database.

For checks concerning Claims (DCECTL1 through DCECTL5) a threshold that can be specified when starting ZRGOE01S is taken into account; if the sum of all differences (per check) is less than this threshold no ERROR is raised but only a WARNING is given. This threshold can be used to prevent the loading process for stopping only based on a small difference (e.g. less than 100 euros).

The check concerning Policies (VZECTL1) will be executed a maximum of three times. Once for the specified ‘Policies to date’ (or when left empty the system determined date), and based on the data loaded for policy modifications, once for the minimum and once for the maximum effect date. Only one check will be performed for each unique reference date.

## AUDIT INFORMATION

In Oracle Health Insurance Datamarts, during the load run audit information is established for the newly read-in records. This means that each loaded record receives a LAADRUN_ID that refers to a specific record in the audit table (STG_SYS_AUDIT). In this audit record, information is recorded about the time of loading and extraction, the code of the source system and the load dates of the various facts of the load run.

This audit data is also transferred to the Data Warehouse in table DWH_SYS_AUDIT.

## LOG INFORMATION

Oracle Health Insurance Datamarts has an internal audit mechanism that keeps log information up to date with mappings generated by OWB (among others: number of records selected, number inserted, updated, number of errors arising and why).

This information can be requested using a few views. These are the views WBX_LAADRUNS_VW, WBX_MAPPING_VW and WBX_LAADRUN_ERRORS_VW. These views are included in the schema of OBD_OWN.  The view WBX_LAADRUNS_VW gives an overview of all load runs that have been performed. For each load run it will be shown which star schemas were loaded, what the run times were, which period was loaded and whether a compatibility check was performed. The view WBX_LAADRUN_ERRORS_VW gives an overview for each script request ID of all mappings that produced an error report during a load run. The view shows which mapping has produced an error message for each script request, at what moment this occurred and what the error message was. The view WBX_LAADRUNS_VW gives an overview of all data warehouse mappings that have been run.

As well as this, an additional logging mechanism is implemented, because not all packages run during the load run are generated by OWB and the standard log is therefore not complete.

The tables WBX_LOG_EVENTS and WBX_LOG_MESSAGES have been added so that log information can also be stored for non-OWB packages.

Furthermore, the output of the Unix load scripts is written to disk in log and out files in $OZG_BASE/log or out/<user>. In these it is possible to find out whether everything ran successfully.

## RESOLVING ERRORS

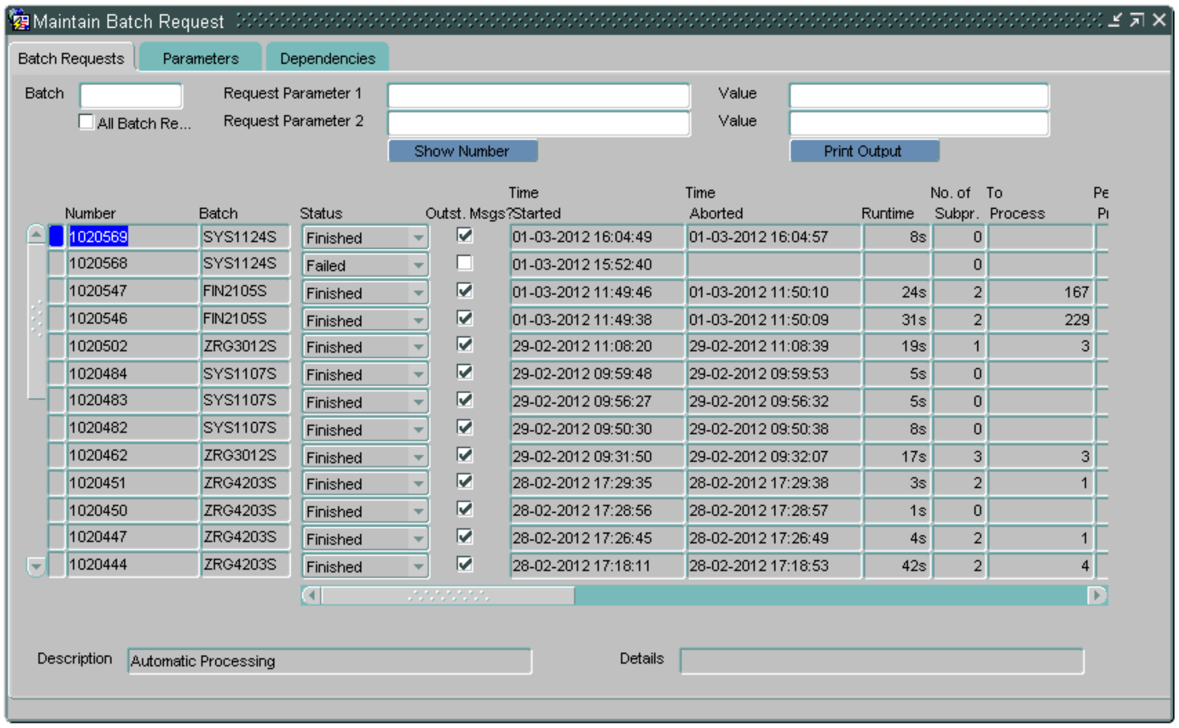
If errors have arisen then in most cases the run can be restarted. Loading of the Staging Area can be done as many times as desired, as the Staging tables are made empty for the read in.

[see also Part III Application management]

## CHECKING STATUS REQUEST LOAD SCRIPTS

When the scripts are run the status of the load run can be followed to see if the load run has started, is still running, has finished, or failed.

Log in to the OHI Back Office application. Navigate to 'System' → 'Batch request' to 'Maintain'. For the script name, type the name of the script that is requested and click on 'Perform search request'. The window looks likes this:



Batch

All Batch Re...

Request Parameter 1 | Value

Request Parameter 2 | Value

| Number | Batch | Status | Outst. Msgs? | Time Started | Time Aborted | Runtime | No. of Subpr. | To Process | Pe Pı |
|---|---|---|---|---|---|---|---|---|---|
| 1020569 | SYS1124S | Finished | ✓ | 01-03-2012 16:04:49 | 01-03-2012 16:04:57 | 8s | 0 | | |
| 1020568 | SYS1124S | Failed | | 01-03-2012 15:52:40 | | | 0 | | |
| 1020547 | FIN2105S | Finished | ✓ | 01-03-2012 11:49:46 | 01-03-2012 11:50:10 | 24s | 2 | 167 | |
| 1020546 | FIN2105S | Finished | ✓ | 01-03-2012 11:49:38 | 01-03-2012 11:50:09 | 31s | 2 | 229 | |
| 1020502 | ZRG3012S | Finished | ✓ | 29-02-2012 11:08:20 | 29-02-2012 11:08:39 | 19s | 1 | 3 | |
| 1020484 | SYS1107S | Finished | ✓ | 29-02-2012 09:59:48 | 29-02-2012 09:59:53 | 5s | 0 | | |
| 1020483 | SYS1107S | Finished | ✓ | 29-02-2012 09:56:27 | 29-02-2012 09:56:32 | 5s | 0 | | |
| 1020482 | SYS1107S | Finished | ✓ | 29-02-2012 09:50:30 | 29-02-2012 09:50:38 | 8s | 0 | | |
| 1020462 | ZRG3012S | Finished | ✓ | 29-02-2012 09:31:50 | 29-02-2012 09:32:07 | 17s | 3 | 3 | |
| 1020451 | ZRG4203S | Finished | ✓ | 28-02-2012 17:29:35 | 28-02-2012 17:29:38 | 3s | 2 | 1 | |
| 1020450 | ZRG4203S | Finished | ✓ | 28-02-2012 17:28:56 | 28-02-2012 17:28:57 | 1s | 0 | | |
| 1020447 | ZRG4203S | Finished | ✓ | 28-02-2012 17:26:45 | 28-02-2012 17:26:49 | 4s | 2 | 1 | |
| 1020444 | ZRG4203S | Finished | ✓ | 28-02-2012 17:18:11 | 28-02-2012 17:18:53 | 42s | 2 | 4 | |

Description: Automatic Processing

Details

## VERIFYING THE LOAD

An important task of the Oracle Health Insurance Datamarts Administrator is the checking of the load run. That doesn't just mean checking that a process has completed, but also checking that the load run was successful. For example, a process can terminate correctly even though no data is loaded. It is therefore not sufficient just to check that a process has terminated.

**NOTE!!!: If the status of the script in the window above is 'Complete', this does not automatically mean that the script ran successfully. The administrator must perform additional checks to determine if the process ran successfully!**

# PART III APPLICATION MANAGEMENT

## INTRODUCTION

This chapter describes aspects of both the technical and functional application management of Oracle Health Insurance Datamarts.

## FUNCTIONAL MANAGEMENT

### Logging and validation

Validation of the Oracle Health Insurance Datamarts load runs is an important part of the functional/technical management.

Currently, the log information concerning the run load runs can be found in multiple locations. **It is therefore not sufficient just to check the status of the script request in the batch scheduler.** It may be that the status of the script after running the load run is 'Complete', but that errors have in fact occurred. Therefore, the administrator should check the sources of log information described below.

#### Logging of loading

The following views are present in the obd_own schema to request information on the results of the load run:

WBX_LAADRUNS_VW

*This view gives a complete overview of all load runs that have taken place. The following is shown for each load run:*

| Column | Description |
|---|---|
| LAADRUN_ID | Unique generated key |
| RELEASE_NR | Oracle Health Insurance Datamarts release number |
| SAV_ID_EXT | ID of script request ZRGOE01S. |
| SAV_ID_STG | ID of script request ZRGOS01S. |
| SAV_ID_DWH | ID of script request ZRGOD01S. |
| DECLARATIES_GELADEN | Indication (J/N [meaning Y/N]) of whether the claims fact is loaded in this run. |
| VERZEKERDEN_GELADEN | Indication (J/N [meaning Y/N]) of whether the policies fact is loaded in this run. |
| PREMIES_GELADEN | Indication (J/N [meaning Y/N]) of whether the premiums fact is loaded in this run. |
| TOE_EN_UITTREDINGEN_GELADEN | Indication (J/N [meaning Y/N]) of whether the In- and Out-fluxs fact is loaded in this run. |
| ZORGVOORNEMENS_GELADEN | Indication (J/N [meaning Y/N]) of whether the care authorizations fact is loaded in this run. |
| ABONNEMENTSHONORARIUM_GELADEN | Indication (J/N [meaning Y/N]) of whether the per capita agreement fact is loaded in this run. |
| VERBINTENISSEN_GELADEN | Indication (J/N [meaning Y/N]) of whether the provider relationships fact is loaded in this run. |
| DATUM_EXTRACTIE | Date of extraction of the source data |
| LAADPERIODE_DECLARATIES | Date used for this load as date to for selection of the source data for claims |
| LAADPERIODE_VERZEKERDEN | Date used for this load as date to for selection of the source data for policies. |
| LAADPERIODE_PREMIES | Date used for this load as date to for selection of the source data for premiums. |

| Column | Description |
|---|---|
| LAADPERIODE_ZORGVOORNEMENS | Date used for this load as date to for selection of the source data for care authorizations. |
| LAADPERIODE_VERBINTENISSEN | Date used for this load as date from for selection of the source data for provider relationships. |
| CONTROLE_UITGEVOERD | Is the loaded data checked? |
| DOORLOOPTIJD_EXTRACTIE | Run time of the extraction phase |
| DOORLOOPTIJD_TRANFORMATIE | Run time of the transformation phase |
| DOORLOOPTIJD_LADEN | Run time of the load phase |
| DOORLOOPTIJD_TOTAAL | Total run time of the extraction + transformation + load phase |
| LAADRUN_ID | Unique generated key |
| RELEASE_NR | Oracle Health Insurance Datamarts release number |

WBX_MAPPING_VW

In this view all mappings are shown that are run during a load run. The following information is available:

| Column | Description |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| FASE | Phase in which the load run is found |
| MAPPING_NAAM | Name of the mapping |
| MAPPING_GESTART | Time when the mapping was started |
| AANTAL_VERWERKTE_RIJEN | Number of processed rows |
| AANTAL_FOUTEN | Number of errors arisen |
| AANTAL_MINUTEN | Number of minutes the mapping took |

WBX_LAADRUN_ERRORS_VW

In this view all errors are shown that have occurred during a load run. This concerns the technical error message that may arise during a mapping. The following information is available:

| Column | Description |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| FASE | Phase in which the load run is found |
| MAPPING_NAAM | Name of the mapping |
| TIJDSTIP_FOUT | Time when the error occurred |
| FOUT_MELDING | Which error occurred |

### WBX logging

Since there are also a number of non-OWB packages that run during the load run for which no logging exists in the OWB audit logging described above, two logging tables have been added in which logging is also written for the non-OWB packages.

This concerns the tables WBX_LOG_EVENTS and WBX_LOG_MESSAGES (master – detail).

In WBX_LOG_EVENTS the following log data is saved:

| Column | Comments |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| AUDIT_ID | Audit ID of the load run, refers to the ID of the table stg_sys_audit. |
| SCRIPTNAAM | The code of the script request from the OHI Back Office batch scheduler |
| STARTTIJD | Start time of the script request |
| EINDTIJD | End time of the script request |
| GELADEN_SCHEMAS | The star schemas that have been loaded with this script request |
| EINDSTATUS | Final status of the script request (Start, Error, Complete) |

In WBX_LOG_MESSAGES the following log data is saved:

| Column | Comments |
|---|---|
| WB_RT_AUDIT_ID | The OWB runtime audit ID (only for mappings generated by OWB) |
| SAV_ID | The ID of the script request, the FK column to WBX_LOG_EVENTS. |
| OBJECT_NAAM | The name of the object (package, procedure, ...) which is being logged. |
| OPMERKINGEN | Potential remarks (step numbers in the case of partitioning) |
| STARTTIJD | Start time of the object |
| EINDTIJD | End time of the object |

## Results of the compatibility checks

The views below provide detailed information on the compatibility checks performed between OHI Back Office and Oracle Health Insurance Datamarts.

WBX_CTR_DECLARATIES_VW

This view shows the results of compatibility checks performed on the claim fact.

| Column | Description |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| CTR_ID | Identification of the check result |
| CODE | Code of the check |
| OMSCHRIJVING | Description of the check |
| DCR_NR | Claim number |
| VOLGNR | Sequence number of the claim line |
| DATUM_ACCOORD | The date on which the claim line was approved |
| PAKKET | The product offered by the coverage |
| VEL_ID | Identification of the journal entry |
| GRG_NR | General ledger account number |
| GBF_NR | General ledger unit number |
| RUK_NR | Category number |
| DATUM_IMPORT | Date imported of the liability |
| RISICODRAGER | The relationship number of the insurance company bearing the risk |
| BETAALMAAND | The month in which the payment took place |
| SOORT_BEDRAG | The amount type of the columns below |
| BEDRAG_BO | Total amount of the claim in Oracle Back Office |
| BEDRAG_BI | Total amount of the claim in Oracle Health Insurance Datamarts |

WBX_CTR_PREMIES_VW

This view shows the results of the compatibility check on the premium fact.

| Column | Description |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| CTR_ID | Identification of the check result |
| CODE | Code of the check |
| OMSCHRIJVING | Description of the check |
| DATUM_VA | Start date of the check period |
| DATUM_TM | End date of the check period |
| PTL_ID | The unique ID of the premium time line in OHI Back Office |
| REL_NR | The party number of the member |
| DATUM_PROLONGATIE | The month for which the renewal was performed |
| RISICODRAGER | The number of risk bearer |
| PAKKET | The code of the product |
| DEKKINGSMAAND | The month for which the coverage applies |
| SOORT_BEDRAG | Description of the check amount |
| BEDRAG_BO | The monthly amount of the premium including potential discounts and surcharges in OHI Back Office |
| BEDRAG_BI | The monthly amount of the premium including potential discounts and surcharges in Oracle Health Insurance Datamarts |

WBX_CTR_VERZEKERDEN_VW

This view shows the results of the compatibility check on the policies fact. For all branded product combinations of a member it is checked that this is present in both OHI Back Office and Oracle Health Insurance Datamarts.

| Column | Description |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| CTR_ID | Identification of the check result |
| CODE | Code of the check |
| OMSCHRIJVING | Description of the check |
| PEILDATUM | End date of the check period |
| CLI_REL_NR | The member which is a member of the policy |
| MERK_CODE | A unique identifying code for the brand |
| PAKKET_CODE | The product that is offered |
| PREMIE_CONSTRUCTIE_CODE | The premium structure that is offered in the product |
| DEKKING_CONSTRUCTIE_CODE | The unique code of the coverage structure unit |
| EIGEN_RISICO_CONSTRUCTIE_CODE | The unique code of the yearly deductible structure unit |
| EIGEN_RISICO_HOOGTE_CODE | The code by which the yearly deductible level is identified |
| ZORGPLICHT_CODE | The code of the contracted care |
| AANTAL_BI | Number of memberships in Oracle Health Insurance Datamarts |
| AANTAL_BO | Number of memberships in OHI Back Office |

WBX_CTR_ONDERHANDEN_WERK_VW

This view shows the results of the compatibility checks on the Work in Progress Claims fact.

| Column | Description |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| CTR_ID | Identification of the check result |
| CODE | Code of the check |
| OMSCHRIJVING | Description of the check |
| DATUM_VA | Start date of the check period |
| DATUM_TM | End date of the check period |
| DCR_NR | Claim number |
| VOLGNR | Sequence number of the claim line |
| SOORT_BEDRAG | Description of the check amount |
| BEDRAG_BO | Number or amount of the Work in Progress Claim in OHI Back Office |
| BEDRAG_BI | Number or amount of the Work in Progress Claim in Oracle Health Insurance Datamarts |

WBX_CTR_ZORG_VOORNEMENS_VW

This view shows the results of the compatibility checks on the care authorizations fact.

| Column | Description |
|---|---|
| SAV_ID | The ID of the script request from the OHI Back Office batch scheduler |
| CTR_ID | Identification of the check result |
| CODE | Code of the check |
| OMSCHRIJVING | Description of the check |
| DATUM_VANAF | Start date of the check period |
| DATUM_TM | End date of the check period |
| ZVN_NR | Care authorization identification number |
| VOLGNR | Sequence number of the period within the care authorization |
| SOORT_BEDRAG | Description of the check amount |
| BEDRAG_BO | Number or amount of the care authorization in OHI Back Office |

| Column | Description |
|---|---|
| BEDRAG_BI | Number or amount of the care authorization in Oracle Health Insurance Datamarts |

### Logging load scripts

The output of the scripts started through the OHI Back Office application is saved in .out files. These files show how the load run ran, including run times and potential errors. These scripts can be found on the OHI Back Office application server under $OZG_BASE/out/<user>.

<user>: user used to log in to the batch scheduler to start the load run.

## Authorization

### User access

It is recommended to create a separate account for each user of Oracle Health Insurance Datamarts. This is particularly convenient from a security standpoint.

This account must be created in the Oracle Health Insurance Datamarts database. CREATE SESSION rights must be assigned to the account at database level, as well as the database role OBD_ROL_SELECT. The database role OBD_ROL_SELECT has select rights on all relevant DWH tables and views.

Creating an account in the database and assigning the correct rights/roles can be done in many ways, for example with the following statement in SQL*Plus:

```
create user username identified by password;
grant create session to username;
grant obd_rol_select to username;
```

### External tables

The Oracle Health Insurance Datamarts load run makes use of external tables. These are files on the server that are treated as tables by the database. These external tables reside on the Oracle Health Insurance Datamarts database server in the directory referred to by the Unix variable $OZG_ADMIN. The input and output (log and bad files) end up in the directory referred to by the Unix variable $TMP.

Because the database for the external tables has to have a reference to these directories, these directory objects are created in the Oracle Health Insurance Datamarts database.

These files contain data that can be adjusted by the functional administrator, after which these adjustments are made to the Data Warehouse in the next load run.

## Configuration of generic structure for claim properties

### Introduction

It is possible to register a set of claim properties in OHI Back Office, which can be used for a specific claim type (depending on the type of procedure). When a claim property is required for use in OHI Business Intelligence it can be added by means of configuring the external file GEN_EIGENSCHAPPEN.dat. This file has the following structure:

- GEN_EIGENSCHAPPEN.dat
  File with containing the mapping definition of claim properties into OHI Business Intelligence.

```
GEBRUIKGROEP       VARCHAR2(60)
EIGENSCHAP         VARCHAR2(60)
TABEL_SPECIFIEK    VARCHAR2(30)
KOLOM_SPECIFIEK    VARCHAR2(30)
TABEL_GENERIEK     VARCHAR2(30)
KOLOM_GENERIEK     VARCHAR2(30)
```

GEBRUIKGROEP contains the Procedure Claim Group of the property that is mapped.

EIGENSCHAP contains the exact name of the property as it is defined in OHI Back Office.

Before functionality was used to add the claim property into a generic structure, claim properties were fixed attributes in the data warehouse (fixed properties which were not configurable). TABEL_SPECIFIEK contains the specific table where a claim property is a part of. Refer to the examples below for additional details related to this topic. KOLOM_SPECIFIEK contains the specific attribute name for a claim property that is part of the old fixed structure. Refer to the examples below for additional details related to this topic. Note that the columns are only relevant for properties that were already available in the old structure (release 2012.01 and beyond).

TABEL_GENERIEK is used to define to which table the claim property should be mapped to, this can be a dimension (DWH_EWE_GENERIEK) or the fact table (DWH_DECLARATIES). Chapter 'When to place a claim property in dimension table' outlines when DWH_DECLARATIES should be used and when DWH_EWE_GENERIEK should be used.

KOLOM_GENERIEK contains the generic attribute that will contain the value of the claim property in OHI Business Intelligence. Refer to the examples in the next paragraph for additional information related to this topic.

☞ **Note:** Columns TABEL_SPECIFIEK and KOLOM_SPECIFIEK may not be changed to another value than specified in the template, these columns are used to enclose the old fixed version. Only TABEL_GENERIEK and KOLOM_GENERIEK can be used for customizing the (new) claim properties. This note is only applicable for claim properties that were already available in the old fixed structure. When a new claim property is introduced, the columns TABEL_SPECIFIEK and KOLOM_SPECIFIEK must be empty (refer to example 2.

☞ **Note**: Once a claim property is allocated to a generic column, we advise to not reallocate this and allocate another claim property to this same generic column within the same claim procedure group.

Example:
Until 1-6-2012 Property A for claim procedure group 'GRPA' is allocated to generic attribute CHAR_EIGENSCHAP_01 in DWH_EWE_GENERIEK. After 1-6-2012 the mapping definition is changed: Property B for group 'GRPA' is allocated to generic attribute CHAR_EIGENSCHAP_01. This will result in a situation where claims before 1-6-2012 contain Property A in CHAR_EIGENSCHAP_01. After 1-6-2012 claims contain PropertyB in CHAR_EIGENSCHAP_01.

This situation must be avoided.

## Examples

Example 1
*Move claim properties that are already present in the old fixed structure to the generic structure.*

The following claim properties of the procedure claim group 'Tandheelkunde' are available as fixed columns in OHI Business Intelligence:

| **Claim Property** | **BI table** | **BI column** |
|---|---|---|
| Aand. prestatiecodelijst | DWH_DECLARATIES | AAND_PRESTATIECODELIJST |
| Gebitselement | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_GEBITSELEMENT |
| Vlakcode | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_VLAK_CODE |
| Machtigingsnummer | DWH_DECLARATIES | MACHTIGINGSNUMMER |
| Patientnummer | DWH_DECLARATIES | PATIENTNUMMER |
| Prestatiecode | DWH_DECLARATIES | PRESTATIECODE |
| Soort prestatie | DWH_EWE_TANDHEELKUNDE | TAE_SOORT_PRESTATIE |
| Specialisme | DWH_DECLARATIES | SPEC_VOORSCHRIJVER |
| Voorschrijver | DWH_DECLARATIES | ZRE_REL_NR_VOORSCHRIJVER |

In this example these attributes are moved into generic claim property columns.

In order to load these claim properties into the generic structure, the mapping to the generic structure must be added into the file GEN_EIGENSCHAPPEN.dat. The claim properties can be loaded into the following generic attributes:

| **Claim Property** | **Table** | **Generic Attribute** |
|---|---|---|
| Aand. prestatiecodelijst | DWH_DECLARATIES | NUMBER_EIGENSCHAP_01 |
| Gebitselement | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_01 |
| Vlakcode | DWH_EWE_GENERIEK | CHAR_EIGENSCHAP_01 |
| Machtigingsnummer | DWH_DECLARATIES | CHAR_EIGENSCHAP_01 |
| Patientnummer | DWH_DECLARATIES | CHAR_EIGENSCHAP_02 |
| Prestatiecode | DWH_DECLARATIES | CHAR_EIGENSCHAP_03 |
| Soort prestatie | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_02 |
| Specialisme | DWH_DECLARATIES | CHAR_EIGENSCHAP_04 |
| Voorschrijver | DWH_DECLARATIES | NUMBER_EIGENSCHAP_02 |

In this case the GEN_EIGENSCHAPPEN.dat should contain the following rows:

```
GEBRUIKGROEP  EIGENSCHAP            TABEL_SPECIFIEK      KOLOM_SPECIFIEK         TABEL_GENERIEK   KOLOM_GENERIEK
```

| | | | | | |
|---|---|---|---|---|---|
| TANDHEELKUNDE | MACHTIGINGSNUMMER | DWH_DECLARATIES | MACHTIGINGSNUMMER | DWH_DECLARATIES | CHAR_EIGENSCHAP_01 |
| TANDHEELKUNDE | PATIENTNUMMER | DWH_DECLARATIES | PATIENTNUMMER | DWH_DECLARATIES | CHAR_EIGENSCHAP_02 |
| TANDHEELKUNDE | PRESTATIECODE | DWH_DECLARATIES | PRESTATIECODE | DWH_DECLARATIES | CHAR_EIGENSCHAP_03 |
| TANDHEELKUNDE | AAND_PRESTATIECODELIJST | DWH_DECLARATIES | AAND_PRESTATIECODELIJST | DWH_DECLARATIES | NUMBER_EIGENSCHAP_01 |
| TANDHEELKUNDE | DCL_TND_VLAK_CODE | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_VLAK_CODE | DWH_EWE_GENERIEK | CHAR_EIGENSCHAP_01 |
| TANDHEELKUNDE | DCL_TND_GEBITSELEMENT | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_GEBITSELEMENT | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_01 |
| TANDHEELKUNDE | SOORT_PRESTATIE | DWH_EWE_TANDHEELKUNDE | TAE_SOORT_PRESTATIE | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_02 |
| TANDHEELKUNDE | SPECIALISME | DWH_DECLARATIES | SPEC_VOORSCHRIJVER | DWH_DECLARATIES | CHAR_EIGENSCHAP_04 |
| TANDHEELKUNDE | VOORSCHRIJVER | DWH_DECLARATIES | ZRE_REL_NR_VOORSCHRIJVER | DWH_DECLARATIES | NUMBER_EIGENSCHAP_02 |

Column GEBRUIKGROEP contains the procedure claim group as it is defined in OHI Back Office; in this case this is ‘TANDHEELKUNDE’.

The column TABEL_SPECIFIEK contains the location of the old fixed structure. In this example a few properties are part of DWH_DECLARATIES and several others are part of DWH_EWE_TANDHEELKUNDE.

The column KOLOM_SPECIFIEK contains the column name in the old structure. TABEL_SPECIFIEK and KOLOM_SPECIFIEK are already pre-defined in the template that is provided by Oracle.

The next two columns provide the mapping to the new generic structure. TABEL_GENERIEK contains the table were the claim property should be loaded and the column KOLOM_GENERIEK contains the attribute where the claim property is loaded.

The value of the column GEBRUIKGROEP is loaded into DWH_DECLARATIES (column dwh_declaraties.gebruikgroep). The combination of the generic column and the column dwh_declaraties.gebruikgroep determines the functional meaning of the column. In the abovementioned example, the column dwh_declaraties.char_eigenschap_01 for DWH_DECLARATIES with dwh_declaraties.gebruikgroep is equal to ‘TANDHEELKUNDE’, and contains the claim property MACHTIGINGSNUMMER.

Example 2:
*A new claim property ‘IND_TAND’ is introduced in OHI Back Office for procedure claim group ‘TANDHEELKUNDE’, this claim property should be added to OHI Business Intelligence. The claim property can have two values in OHI Back Office: ‘Y’ or ‘N’.*

In order to add this claim property, the file GEN_EIGENSCHAPPEN.dat should be edited. The new property must be added and a column should be chosen where the claim property ‘IND_TAND’ is loaded into OHI Business Intelligence.

The first thing to decide is whether this property should be loaded into the dimension table (DWH_EWE_GENERIEK), or into the fact table (DWH_DECLARATIES). When a claim property does not have many different values, it is recommended to save it in the dimension table for storage optimization. In this case there are only two possible values (‘Y’ or ‘N’); therefore this property should be added to the dimension table DWH_EWE_GENERIEK.

The property is a character, in example 1 we already added some properties for the procedure claim group ‘TANDHEELKUNDE; One character claim property is already used for this Procedure Claim Group, namely ‘DCL_TND_VLAK_CODE’. This one is mapped to column CHAR_EIGENSCHAP_01 in table DWH_EWE_GENERIEK. Therefore the new claim property can be added to CHAR_EIGENSCHAP_02, this attribute is not yet used for this procedure claim group. The following line should be added to GEN_EIGENSCHAPPEN.dat:

| **GEBRUIKGROEP** | **EIGENSCHAP** | **TABEL_SPECIFIEK** | **KOLOM_SPECIFIEK** | **TABEL_GENERIEK** | **KOLOM_GENERIEK** |
|---|---|---|---|---|---|
| TANDHEELKUNDE | IND_TAND | | | DWH_EWE_GENERIEK | CHAR_EIGENSCHAP_02 |

The complete file will now look as follows:

| GEBRUIKGROEP | EIGENSCHAP | TABEL_SPECIFIEK | KOLOM_SPECIFIEK | TABEL_GENERIEK | KOLOM_GENERIEK |
|---|---|---|---|---|---|
| TANDHEELKUNDE | MACHTIGINGSNUMMER | DWH_DECLARATIES | MACHTIGINGSNUMMER | DWH_DECLARATIES | CHAR_EIGENSCHAP_01 |
| TANDHEELKUNDE | PATIENTNUMMER | DWH_DECLARATIES | PATIENTNUMMER | DWH_DECLARATIES | CHAR_EIGENSCHAP_02 |
| TANDHEELKUNDE | PRESTATIECODE | DWH_DECLARATIES | PRESTATIECODE | DWH_DECLARATIES | CHAR_EIGENSCHAP_03 |
| TANDHEELKUNDE | AAND_PRESTATIECODELIJST | DWH_DECLARATIES | AAND_PRESTATIECODELIJST | DWH_DECLARATIES | NUMBER_EIGENSCHAP_01 |
| TANDHEELKUNDE | DCL_TND_VLAK_CODE | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_VLAK_CODE | DWH_EWE_GENERIEK | CHAR_EIGENSCHAP_01 |
| TANDHEELKUNDE | DCL_TND_GEBITSELEMENT | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_GEBITSELEMENT | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_01 |
| TANDHEELKUNDE | SOORT_PRESTATIE | DWH_EWE_TANDHEELKUNDE | TAE_SOORT_PRESTATIE | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_04 |
| TANDHEELKUNDE | IND_TAND | | | DWH_EWE_GENERIEK | CHAR_EIGENSCHAP_02 |

Note that the columns TABEL_SPECIFIEK and KOLOM_SPECIFIEK are not set up for IND_TAND, this is because this claim property does not exists in the old fixed structure (claim property is added after release 2012.01).

In the situation where both fixed and generic structures are set up, the new records are only loaded in the generic structure; this is due to storage optimization. An exception to this are claim properties that are used for dimension keys, these claim properties are also loaded in the old structure if a definition is available for the generic structure.

<u>Example 3:</u>
*Claim property 'MACHTIGINGSNUMMER' for Procedure Claim Group 'TANDHEELKUNDE' should not be loaded in the generic structure but only in the old fixed structure (table: DWH_DECLARATIES, attribute: MACHTIGINGSNUMMER).*

In this example, the old fixed attribute for 'MACHTIGINGSNUMMER' should be used. The claim property must not be mapped to a generic attribute. This can be easily done by removing the mapping to the generic table and attribute in GEN_EIGENSCHAPPEN.dat. The file will look as follows after implementing this:

| GEBRUIKGROEP | EIGENSCHAP | TABEL_SPECIFIEK | KOLOM_SPECIFIEK | TABEL_GENERIEK | KOLOM_GENERIEK |
|---|---|---|---|---|---|
| TANDHEELKUNDE | MACHTIGINGSNUMMER | DWH_DECLARATIES | MACHTIGINGSNUMMER | | |
| TANDHEELKUNDE | PATIENTNUMMER | DWH_DECLARATIES | PATIENTNUMMER | DWH_DECLARATIES | CHAR_EIGENSCHAP_02 |
| TANDHEELKUNDE | PRESTATIECODE | DWH_DECLARATIES | PRESTATIECODE | DWH_DECLARATIES | CHAR_EIGENSCHAP_03 |
| TANDHEELKUNDE | AAND_PRESTATIECODELIJST | DWH_DECLARATIES | AAND_PRESTATIECODELIJST | DWH_DECLARATIES | NUMBER_EIGENSCHAP_01 |
| TANDHEELKUNDE | DCL_TND_VLAK_CODE | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_VLAK_CODE | DWH_EWE_GENERIEK | CHAR_EIGENSCHAP_01 |
| TANDHEELKUNDE | DCL_TND_GEBITSELEMENT | DWH_EWE_TANDHEELKUNDE | TAE_DCL_TND_GEBITSELEMENT | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_01 |
| TANDHEELKUNDE | SOORT_PRESTATIE | DWH_EWE_TANDHEELKUNDE | TAE_SOORT_PRESTATIE | DWH_EWE_GENERIEK | NUMBER_EIGENSCHAP_04 |
| TANDHEELKUNDE | IND_TAND | | | DWH_DECLARATIES | CHAR_EIGENSCHAP_02 |

Note that only TABEL_SPECIFIEK and KOLOM_SPECIFIEK are set up for MACHTIGINGSNUMMER.

Also note that this is only relevant for claim properties that were already available in the old fixed structure in OHI Business Intelligence.

### When to place a claim property in a dimension table

There are two places where a claim property can be added in OHI Business Intelligence: fact table DWH_DECLARATIES and dimension table DWH_EWE_GENERIEK.

It is important to make a good decision where to place a claim property in the generic structure. A claim property should be placed in DWH_DECLARATIES if there are many different potential values for the claim property. For instance the claim property PATIENTNUMMER contains the patient number, this is different for every patient and therefore there are many potential values in OHI Back Office for this claim property. As a result of this it should be placed in DWH_DECLARATIES. If this is placed in DWH_EWE_GENERIEK, the dimension will be very large. A new dimension record will be created for almost every fact record, which results in very insufficient storage optimization.

It can be generally stated that when a claim property has more than a few dozen different values, that it should be placed in the fact table.

### Generate views to create a functional overlay over the generic structure

It is possible to generate a view per Procedure Claim Group that contains a fixed set of DWH_DECLARATIES attributes and a dynamic created set of claim properties. In DWH_DECLARATIES and DWH_EWE_GENERIEK there is a generic naming convention (e.g. CHAR_EIGENSCHAP_01) for claim properties. With the view generator it is possible to create a view that contains all claim properties of a Procedure Claim Group, the attributes will obtain the functional name of the claim property (as defined in GEN_EIGENSCHAPPEN.dat) instead of the generic name. The view can be used to represent the data with column names that have a functional meaning.

The view can be generated with a procedure that is available in the package WBX_VIEW_GENERATOR. The procedure is called WBX_VIEW_GENERATOR.GENERATE_DECLARATIE_VIEW. This procedure must be executed under the OBD_OWN schema on the OHI Business Intelligence database. It is mandatory to give a value for the parameter ‘P_GEBRUIKGROEP’. This parameter should contain the exact name of the Procedure Claim Group. For example, the view for the Procedure Claim Group ‘TANDHEELKUNDE’ can be generated with the following statement:

```
exec wbx_view_generator.generate_declaratie_view(‘TANDHEELKUNDE’);
```

This will generate a view named DWH_DCE_TANDHEELKUNDE_VW that can subsequently be used to integrate information into the reporting environment.

### Add claim properties from generic structure to OBI EE repository

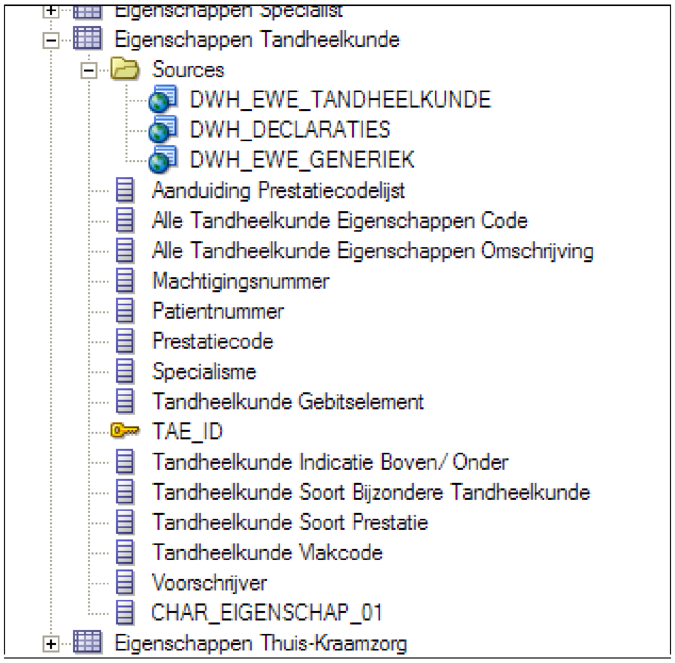
This paragraph describes how to add generically mapped claim properties to the OBI EE repository business model.

All generic claim property attributes are already added to the physical layer of the OBI EE repository.

How a claim property attribute can be added to the business model layer and presentation layer in OBI EE is described in the following example:

In Procedure Claim Group ‘Tandheelkunde’ the claim property ‘EXAMPLE123’ is added. This column is mapped to ‘CHAR_EIGENSCHAP_01’ in DWH_EWE_GENERIEK. The claim property ‘EXAMPLE123’ should be added to the logical table ‘Eigenschappen Tandheelkunde’ and should be named ‘Example 123’.

1. <u>Add the generic column to the business model</u>
   Drag the attribute ‘CHAR_EIGENSCHAP_01’ from the Physical Table Source ‘DWH_EWE_GENERIEK’ to the Logical Table ‘Eigenschappen Tandheelkunde’. Once this has been done, DWH_EWE_GENERIEK is automatically added as a Logical Table Source and CHAR_EIGENSCHAP_01 is added to the logical table, represented as follows:



Eigenschappen Specialist
Eigenschappen Tandheelkunde
Sources
DWH_EWE_TANDHEELKUNDE
DWH_DECLARATIES
DWH_EWE_GENERIEK
Aanduiding Prestatiecodelijst
Alle Tandheelkunde Eigenschappen Code
Alle Tandheelkunde Eigenschappen Omschrijving
Machtigingsnummer
Patientnummer
Prestatiecode
Specialisme
Tandheelkunde Gebitselement
TAE_ID
Tandheelkunde Indicatie Boven/ Onder
Tandheelkunde Soort Bijzondere Tandheelkunde
Tandheelkunde Soort Prestatie
Tandheelkunde Vlakcode
Voorschrijver
CHAR_EIGENSCHAP_01
Eigenschappen Thuis-Kraamzorg

2. <u>Rename the column to the functional meaning</u>
   Change the name of CHAR_EIGENSCHAP_01 to ‘Example 123’ in the logical table

3. <u>Add the associated Claim Procedure Group</u>
   Due to the fact that generic claim property attributes can have a differenct functional meaning per Claim Procedure Group, it is needed to add information that CHAR_EIGENSCHAP_01 only means ‘Example 123’ as the Claim Procedure Group is equal to ‘TANDHEELKUNDE’. Open the Logical Table Source ‘DWH_EWE_GENERIEK’.

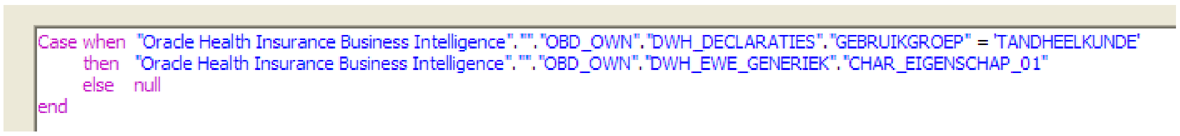
   After opening the Logical Table Source ‘DWH_EWE_GENERIEK’, open the ‘General’ tab. In pane ‘Map to these tables:’ the physical table source ‘DWH_EWE_GENERIEK’ is already there. The attribute ‘GEBRUIKGROEP’ from ‘DWH_DECLARATIES’ is also needed, therefore add

   ‘DWH_DECLARATIES’ as physical table by means of selecting the ‘Add’ button: ✚

   Add ‘DWH_DECLARATIES’ as a source
   Now open the ‘Column Mapping’ tab. ‘Example 123’ is mapped to CHAR_EIGENSCHAP_01. This is correct, but it only should be mapped as the Claim Property Group is ‘TANDHEELKUNDE’. Therefore we need to change the expression. Open the expression editor for

   ‘CHAR_EIGENSCHAP_01’ by means of using this button:

   The editor window is shown. CHAR_EIGENSCHAP_01 should be used only if GEBRUIKGROEP is equal to ‘TANDHEELKUNDE’. This can be achieved by means of using this expression:



```
Case when  "Oracle Health Insurance Business Intelligence"."". "OBD_OWN". "DWH_DECLARATIES". "GEBRUIKGROEP" = 'TANDHEELKUNDE'
     then  "Oracle Health Insurance Business Intelligence"."". "OBD_OWN". "DWH_EWE_GENERIEK". "CHAR_EIGENSCHAP_01"
     else  null
end
```

Add the new attribute to the presentation layer
Add ‘Example 123’ to the Presentation table ‘- Eigenschappen Tandheelkunde’

Check in the changes, save the repository and the claim property will now be available for end-users.

## Configuration of generic structure for dimension properties

### Introduction

It is possible to register a set of dimension properties in OHI Back Office. When a dimension property is required for use in OHI Business Intelligence it can be added by means of configuring the external file DIM_EIGENSCHAPPEN.dat. This file has the following structure:

- DIM _EIGENSCHAPPEN.dat
  File with containing the mapping definition of dimension properties into OHI Business Intelligence.

```
DIMENSIE_TABEL     VARCHAR2(60)
DIMENSIE_KOLOM     VARCHAR2(60)
BRON_TABEL         VARCHAR2(30)
EIGENSCHAP         VARCHAR2(30)
```

DIMENSIE_TABEL contains the extact table name within OHI Business Intelligence

DIMENSIE_KOLOM contains the exact column name of the table mentioned in DIMENSIE_KOLOM

BRON_TABEL contains the name of the table for which the property (mentioned in EIGENSCHAP) is defined in OHI Back Office.

EIGENSCHAP contains the name of the property as it is defined in OHI Back Office.

☞ **Note:** Columns DIMENSIE_TABEL and DIMENSIE_KOLOM may not be changed to another value than specified in the template, these columns are used to indicate target within OHI Business Intelligence and is thus fixed. Only BRON_TABEL and EIGENSCHAP can be used for customizing the (new) dimension properties.

### Examples

Example
*A new dimension property 'GROEPCODE' is introduced in OHI Back Office for group contracts, this dimension property should be added to OHI Business Intelligence.*

In order to add this dimension property, the file DIM_EIGENSCHAPPEN.dat should be edited. The new property must be added and a column should be defined where the dimension property 'GROEPCODE' is loaded into OHI Business Intelligence.

The property is a character and will be mapped to column CHAR_EIGENSCHAP_01 in table DWH_COLLECTIEVE_CONTRACTEN. The following line should be edited within DIM_EIGENSCHAPPEN.dat:

| DIMENSIE_TABEL | DIMENSIE_KOLOM | BRON_TABEL | EIGENSCHAP |
|---|---|---|---|
| DWH_COLLECTIEVE_CONTRACTEN | CHAR_EIGENSCHAP_01 | VER_COLLECTIEVE_CONTRACTEN | GROEPCODE |

### Generate views to create a functional overlay over the generic structure

It is possible to generate a view per dimension that contains all attributes of the dimension. For the generic dimension properties there is a generic naming convention (e.g. CHAR_EIGENSCHAP_01). With the view generator it is possible to create a view that contains all dimension properties, the attributes will obtain the functional name of the dimension property (as defined in DIM_EIGENSCHAPPEN.dat) instead of the generic name. The view can be used to represent the data with column names that have a functional meaning.

The view can be generated with a procedure that is available in the package WBX_VIEW_GENERATOR. The procedure is called WBX_VIEW_GENERATOR.GENERATE_DIMENSIE_VIEW. This procedure must be executed under the OBD_OWN schema on the OHI Business Intelligence database. It is mandatory to give a value for the parameter 'P_DIMENSIE_TABEL'. This parameter should contain the exact name of the dimension table. For example, the view for the group contract dimension 'DWH_COLLECTIEVE_CONTRACTEN' can be generated with the following statement:

```
exec wbx_view_generator.generate_dimensie_view('DWH_COLLECTIEVE_CONTRACTEN');
```

This view can subsequently be used to integrate information into the reporting environment.

### Add dimension properties from generic structure to OBI EE repository

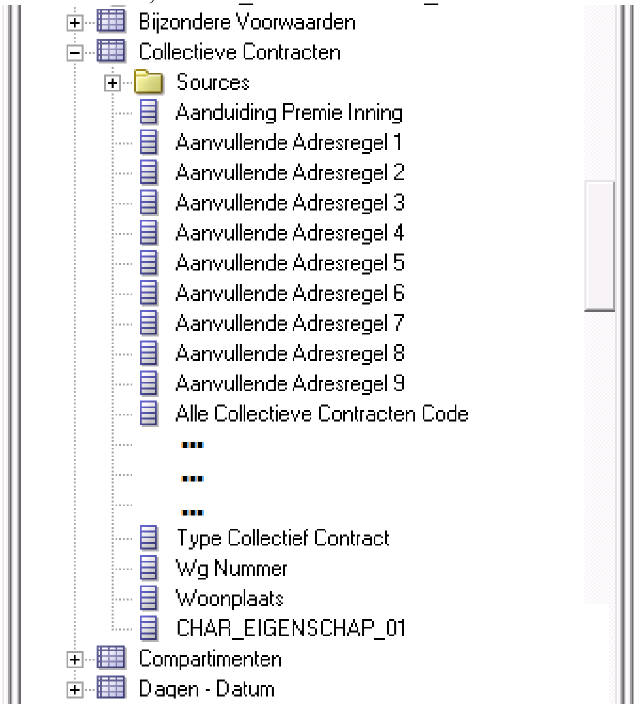
This paragraph describes how to add generically mapped dimension properties to the OBI EE repository business model.

All generic dimension property attributes are already added to the physical layer of the OBI EE repository.

How a dimension property attribute can be added to the business model layer and presentation layer in OBI EE is described in the following example:

In the Group Contract dimension the property 'GROEPCODE' is added. This column is mapped to 'CHAR_EIGENSCHAP_01' in DWH_COLLECTIEVE_CONTRACTEN. The dimension property 'GROEPCODE' should be added to the logical table 'Collectieve Contracten' and should be named 'Groep Code'.

4. <u>Add the generic column to the business model</u>
   Drag the attribute 'CHAR_EIGENSCHAP_01' from the Physical Table Source 'DWH_COLLECTIEVE_CONTRACTEN' to the Logical Table 'Collectieve Contracten'. Once this has been done, CHAR_EIGENSCHAP_01 is added to the logical table, represented as follows:



```
Bijzondere Voorwaarden
Collectieve Contracten
    Sources
    Aanduiding Premie Inning
    Aanvullende Adresregel 1
    Aanvullende Adresregel 2
    Aanvullende Adresregel 3
    Aanvullende Adresregel 4
    Aanvullende Adresregel 5
    Aanvullende Adresregel 6
    Aanvullende Adresregel 7
    Aanvullende Adresregel 8
    Aanvullende Adresregel 9
    Alle Collectieve Contracten Code
    ...
    ...
    ...
    Type Collectief Contract
    Wg Nummer
    Woonplaats
    CHAR_EIGENSCHAP_01
Compartimenten
Dagen - Datum
```

5. Name the column to the functional meaning
   Change the name of CHAR_EIGENSCHAP_01 to ‘Groep Code’ in the logical table

6. Add the new attribute to the presentation layer
   Add ‘Groep Code’ to the Presentation table ‘Collectieve Contracten’

   Check in the changes, save the repository and the claim property will now be available for end-users.

# TECHNICAL MANAGEMENT

## Cleaning up OWB Audit data

As described earlier, OWB generates audit information while running the OWB mappings. This information is stored in the run time tables of the Oracle Health Insurance Datamarts repository schema OWBSYS (WB_RT_% tables). These tables are located in the OWBSYS tablespace. Every time that it's loaded this tablespace grows.

To avoid this table space becoming too large it is possible to delete (part of) the audit data. A package is available for this in the OWBSYS schema. This is the package ***WB_RT_API_PURGE*** that contains a number of 'purge' procedures that can remove data in a variety of ways:

| | |
|---|---|
| `purge_execution` | `All audit data is cleaned` |
| `purge_execution (exec_id)` | `Audit data with rte_id = exec_id is cleaned` |
| `purge_execution (start_date, end_date)` | `All data from the period between the start and end date is cleaned` |

This audit data forms an important source of information for how a load run has run and for resolving problems. It is therefore important to consider carefully whether the information is genuinely no longer needed before running this procedure.

This data can also be removed using the script purge_audit_template.sql in the directory $ORACLE_HOME/owb/rtp/sql/.

## New releases of Oracle Health Insurance Datamarts

When new releases of Oracle Health Insurance Datamarts are brought out new versions of this documentation will also be supplied via iProjects files.

The Oracle Health Insurance installation menu OZGPATCH must be used for the installation of new Oracle Health Insurance Datamarts releases or patches. For operation of the installation menu: see document ‘PCM02101.pdf’ (Oracle Health Insurance Installation of Releases).

## Reorganization of tables

It is recommended to regularly (depending on the load frequency, e.g. once per quarter/half year) reorganize the Data Warehouse facts tables and indexes on facts tables.

For the partitioned facts tables:

- DWH_AFGEWEZEN_DECLARATIES
- DWH_DECLARATIES
- DWH_DECLARATIES_AGG
- DWH_DEELNAME_MUTATIES
- DWH_ONDERHANDEN_WERK
- DWH_PREMIES
- DWH_VERBINTENISSEN
- DWH_VERZEKERDEN

- DWH_ZORG_VOORNEMENS

this can be done using:

- ALTER TABLE [table name] MOVE PARTITION [partition name]
- ALTER INDEX [index name] REBUILD PARTITION [partition name]

For the non-partitioned facts table:

- DWH_TOE_UITTREDINGEN

this can be done using:

- ALTER TABLE [table name] MOVE
- ALTER INDEX [index name] REBUILD

## Compression of partitioned fact tables.

As of Oracle Health Insurance Datamarts version 2011.03 it is possible to compress the partitioned tables. By compressing large fact tables, a large amount of disk space can be saved.

Step '850 - Partition/compress tables' in OZGPATCH.pl is available for this. This choice is the same for OHI Back Office and Oracle Health Insurance Datamarts, however the following submenu choices are only shown for Oracle Health Insurance Datamarts.

```
INFO : ============================================
INFO : = Redefine a table:                        =
INFO : = C - Compress tables and indexes          =
INFO : = U - Uncompress  (revert compression)     =
INFO : ============================================
INFO : Which table redefinition would you like to execute (C, U)?
```

If 'C'ompress tables and indexes is chosen, then you see a list of tables that can be compressed. By typing in the table name the choice is confirmed and the (sub)partitions of the table are compressed, including local partitioned indexes. For large fact tables this may take a long time. After compression, all local partitioned indexes are no longer usable and have to be rebuilt. Option '870 - Rebuild unusable indexes' in OZGPATCH.pl is available for this. Rebuilding these 'unusable' indexes can also take a long time if the indexes concern large facts tables.

If 'U'ncompress (revert compression) is chosen, then you see a list of tables that can be uncompressed. The rest of the procedure is the same as for compressing tables. After uncompressing tables, the local partitioned tables must also be rebuilt.

**Note:** Ensure that there is enough disk space available for compression and uncompression.

## OWB Runtime service

As of Oracle Health Insurance Datamarts version 2010.01, use is made of OWB11g. As a consequence, as well as the Oracle Health Insurance Datamarts OBD_OWN schema, an OWB Runtime repository schema OWBSYS and a workspace owner OHI_BI_WS_OWN are also available for the administration of the installation and execution of OWB mappings (see Server installation).

To enable performance of this administration, the repository makes use of the OWB Runtime Service. This is a process on the server that is started by the OS user oracle when starting the Oracle Health Insurance Datamarts database.

There are a number of scripts available relating to the management of this runtime service (in $ORACLE_HOME/owb/rtp/swl) and these can be run in SQLPlus under the OWBSYS account:

| | |
|---|---|
| show_service.sql | This script shows the current status of the service. |
| start_service.sql | This script starts the service. |
| stop_service.sql | This script stops the service. |
| service_doctor.sql | This script can be used to perform a diagnosis of the service if it will not start. |
| reset_repository.sql | Sets the password of the repository owner (OWBSYS) and initializes a number of repository-specific values. |

See the header of these scripts for a more detailed explanation of their use.

# PART IV APPENDICES

## APPENDIX A: CLONING ORACLE HEALTH INSURANCE DATAMARTS ENVIRONMENTS

From a management standpoint, it is sometimes necessary to make a copy of an Oracle Health Insurance Datamarts environment and place it in another environment, for example to make a production environment available on a test environment.

An Oracle Health Insurance Datamarts environment consists of the repository schema OWBSYS, the OWB workspace owner OHI_BI_WS_OWN and the Oracle Health Insurance Datamarts schema OBD_OWN.

Seeing as environment-specific information is stored in the repository, a number of things have to be done after the transfer to adjust this connection information for the new environment to ensure that everything continues to work correctly.

This environment-specific information includes:

- Connection information for the repository
- Connection information relating to the registered locations

For a description of the cloning of an Oracle Health Insurance Datamarts environment there is a page available on My Oracle Support. See:

How To Update Warehouse Builder After A Database Cloning [ID 434272.1]

This also refers to a Java based tool for correcting the locations used by Oracle Health Insurance Datamarts:

How To Use The OWB 11.2.0.3 Java Tool oracle.wh.util.locationhelper.LocationTool for Failover [ID 1362745.1]

## APPENDIX B: USE OF WBX_LAADRUN_ERRORS_VW AND WBX_MAPPINGS_VW VIEWS.

A number of extra steps are required to use views WBX_LAADRUN_ERRORS_VW and WBX_MAPPINGS_VW. This is because, in contrast to older OWB versions, from OWB 11gR2 makes use of workspaces. The WBX_LAADRUN_ERRORS_VW and WBX_MAPPINGS_VW views calculate OWB public views in the OWBSYS schema that is subdivided into workspaces. Before the view is calculated it must be specified for which workspace the view should be used. This is done as follows:

Extra grants must be assigned from OWBSYS to OBD_OWN:

*grant select on wb_rt_audit  to obd_own with grant option;*
*grant select on wb_rt_errors to obd_own with grant option;*

Users who want to calculate the view should be set up as workspace users (of workspace OHI_BI_WS) using the OWB Repository assistant (reposinst.sh).

Therefore before the view is used, the following command has to be run:

Execute owbsys.wb_workspace_management.set_workspace('OHI_BI_WS','OHI_BI_WS_OWN');

This can potentially be solved using a LOGON trigger:

```
create or replace trigger gebruiker_logon_trigger
   after logon on gebruiker.schema
begin
owbsys.wb_workspace_management.set_workspace('OHI_BI_WS','OHI_BI_WS_OWN');
end;
```

For this, an explicit grant has to first be assigned to 'user' from OWBSYS:

```
grant execute on wb_workspace_management to user;
```

In the example above replace 'user' with the user that wishes to use the named views.

# APPENDIX C: EXADATA USAGE

When migrating to Exadata bear in mind the following migration path.

1. Please refer to chapter C: "*Converting to Oracle RAC and Oracle Rac One Node from Single Instance Oracle Database.", "Oracle Real Application Clusters Installation Guide 11g Release 2".*

2. Please refer to chapter 8: "*Performing Oracle ASM Data Migration with RMAN*", "*Oracle Automatic Storage Management Administrator's Guide*."

After migration steps

Check to see if owbsys or obd_own accounts are locked

Sqlplus obd_own/<PASSWORD>

*select username,account_status,lock_date,expiry_date from dba_users where username in ('OWBSYS','OBD_OWN','OWBRT_SYS','OHI_BI_WS_OWN');*

If the owbsys repository need to be reset run: $ORACLE_HOME/owb/rtp/sql/reset_repository.sql)

```
For example:
-- reset home
-- all homes on exadata should be located here:
update wb_rt_service_nodes
set server_side_home            = '/u01/app/oracle/product/11.2.0.3/dbhome_1';
--
commit
$ORACLE_HOME/owb/UnifiedRepos/reset_owbcc_home.sql
</u01/app/oracle/product/11.2.0.3/dbhome_1>

$ORACLE_HOME/owb/rtp/sql/reset_repository.sql
<Password for OWBSYS>
```

Check to see if locations are registered

```
BEGIN
  sys_mapping_util_pck.set_workspace;
END;
/

SELECT object_name
FROM user_objects
WHERE object_type          = 'PACKAGE BODY'
AND SUBSTR(object_name,1,3) IN ('DWH','STG')            -- Destination
AND SUBSTR(object_name,9,3) IN ('DWH','STG','SRC','EXT') -- Source
MINUS
SELECT wo.object_name FROM owbsys.wb_rt_warehouse_objects wo ORDER BY 1;
/
```

(should not show any rows, if any redo migration, do not unregister locations)

Check to see if OWB is running

*select sid,serial#,username,status,server,machine,module from gv$session where module = 'OWB_CCS';*

If there is a module OWB_CCS then shutdown owb by running:

Sqlplus OWBSYS/<PASSWORD>

@$ORACLE_HOME/ owb/rtp/sql/stop_service.sql

In release 101201 for Open Beleid the proper execute rights should be granted to
Exadata DataMachine machine (755)
$ORACLE_HOME/owb/bin/unix/OMBPlus.sh
$ORACLE_HOME/owb/bin/unix/setowbenv.sh
$ORACLE_HOME/owb/bin/unix/ombplus
If this is not set correctly patching will fail with a permission denied error

3. Make sure the DB link SRC_OPENZORG.WORLD point to the correct Open Zorg environment.

4. Grant permissions as per administrator reference on the Open Zorg Database.

```
Sqlplus system/password@BO
grant create session to obd_select_user;
Grant alter session to obd_select_user;
Grant ozg_rol_select to obd_select_user;
Grant select, insert, delete on ozg_owner.geb#obd_declaraties to obd_select_user;
Grant execute on ozg_owner.alg_tab_pck  to obd_select_user;
Grant execute on ozg_owner.fin_fpm_vars_pck to obd_select_user;
Grant select on v_$database to obd_select_user;
Grant execute on ozg_owner.geb_odr_pck to obd_select_user;
```

5. Make sure the following patchsets are installed on the Exadata Database Machine: 13257502, 13575868.
Note 1: that these patchsets are installed in 2012.01, however if the physical machine changes, the software of new and old machines need to be identical.
Note 2: please note that no SQL statements of the patches need to be applied: they should be already applied in 2012.01 (ie only run Opatch apply)

6. Make sure java and xmldb are installed.

7. For multi node RAC: Apply the steps described in “Migrating OWB from Single Instance to **multi node RAC** “
For single instance: Apply the steps described in “OWB 11gR2 post-cloning process for OHI Business Intelligence”

8. Gather schema statistics for OBD_OWN.

9. Make sure the tnsnames.ora entries on application server are pointing to the correct environment for Open Beleid as well as Open Zorg.

10. Recompile schema OBD_OWN, after which make sure no invalid objects are found in schema OBD_OWN

```
begin
dbms_utility.compile_schema( schema  => user , compile_all   =>TRUE,  reuse_settings =>TRUE);
end;

select count(*) from user_objects where status != 'VALID';
```

11. Create wallet entries on Application server for 3 users, and display mkstore is similar to below on the application server.
mkstore –wrl . -listCredential
<db> batch
<db>_batch batch
<db>_install obd_own
Also make sure there is a tnsnames entry for
<db>
<db>_batch
<db>_install
**To check for multi instance rac node (setup 1, non preferred setup):**
mkstore –wrl . –listCredential
oton1 batch
oton1_batch batch
oton1_install obd_own

where oton1 is the service_name which is used for extraction

sqlplus /@oton1
sqlplus /@oton1_batch
sqlplus /@oton1_install
should all be able to connect

**To check for multi instance rac node (setup 2, preferred setup ) or single instance:**

mkstore –wrl . –listCredential
oton  batch
oton_batch batch
oton_install obd_own

where oton is the service_name which is used for extraction

sqlplus /@oton
sqlplus /@oton_batch
sqlplus /@oton_install
should all be able to connect

12. Make sure the input files of the External tables, are a the OBD_INPUT location, if not place them in OBD_INPUT directory and make sure oracle has read, write permissions on that directory.

13. Make sure there is a batch scheduler running on the application server.

Note For **Multi node Rac**: To transfer the service to another node:
srvctl modify service -d db_unique_name -s service_name -i old_instance_name  -t new_instance_name [-f]
also please note that you might need a batch scheduler on the other node, and tnsnames entry on the second node of the first service to be able to connect to the relocated service.

Note for Exadate: the use of a IORM plan is highly recommendable, so is setting the limit clause for a consistent performance experience see MOS "Configuring Exadata I/O Resource Manager for Common Scenarios [ID 1363188.1]"

The database parameter parallel_degree_policy = auto takes care of maximum utilization of resources. Resource Management is about maximum consumption of resources and acquiring the maximum resources. If there are for instance 2 databases A and B, and the total available resources is 100. Then maximum utilization of resources for A means utilizing 100 of that resources; but for B then there are no resources left.

The manual distribution of resource, for instance A can consume 80% of the resources and B 20% of the resources, is an example of IOResource Management, and should make sure B can also aquire their maximum of resources of 20%.

In ideal situation IOResources Management is assigned and utilized to a maximum.

Therefore an IOResource Management Plan is recommended to prevent a case where patching an OHIBI installation takes significcant amount of time, since for example all resources are taken by A, acquiring resources failed.

When statements run in parallel but they could in potential have a higher degree of parallism, maximum utilization of resources failed.

The first situation if more concerning than the latter, therefore an IOResourcemanagement plan is highly recommended. Setting the parallel_degree_policy = auto is optional, but only supported for the ETL user OBD_OWN and not for the OBD_SELECT_USER, and should be implemented with an after logon trigger as shown below. Reason for this is that statement queing will occur.

```
create or replace trigger trg_parallel_etl after logon on database
begin
if user = 'OBD_OWN' then
execute immediate 'alter session set parallel_degree_policy=AUTO';
execute immediate 'alter session enable parallel dml';
end if;
end;
/
```

For DOP to work properly IO Calibration is needed. This can be achieved by executing the one time only process `DBMS_RESOURCE_MANAGER.CALIBRATE_IO, failure to do so would lead to the message "` *automatic DOP: skipped because of IO calibrate statistics are missing" in the explain plan. The example below is derived from the documentation*:
http://docs.oracle.com/cd/E11882_01/appdev.112/e10577/d_resmgr.htm#CJGHGFEA

## Example of using I/O Calibration procedure

```
SET SERVEROUTPUT ON
DECLARE
   lat  INTEGER;
  iops INTEGER;
  mbps INTEGER;
 BEGIN
-- DBMS_RESOURCE_MANAGER.CALIBRATE_IO (<DISKS>, <MAX_LATENCY>, iops, mbps, lat);
DBMS_RESOURCE_MANAGER.CALIBRATE_IO (2, 10, iops, mbps, lat);
    DBMS_OUTPUT.PUT_LINE ('max_iops = ' || iops);
    DBMS_OUTPUT.PUT_LINE ('latency  = ' || lat);
    DBMS_OUTPUT.PUT_LINE ('max_mbps = ' || mbps);
end;
/
While running
```

## Appendix D: OWB 11gR2 post-cloning process for OHI Business Intelligence

The post-cloning process means the steps to be executed after the files of a shutdown of an Oracle Health Insurance Data Marts database have been copied and started up again as another database.

This document describes the post-cloning process of an Oracle Health Insurance Data Marts (a.k.a. OHI Business Intelligence) environment running on an Oracle 11.2.0.3 database and Oracle Warehouse Builder version. It is presumed that the database and owb software for the database to be cloned is in the same Oracle Home as the database that is used for cloning. If this is not the case than the following steps need to be executed first:

– connect as sysdba

– execute reset_owbcc_home.sql in $ORACLE_HOME/owb/UnifiedRepos
when asked for the control center home, enter the full path to the $ORACLE_HOME
for instance /ozg/app/oracle/product/11.2.0.3/db_1
note: in some instances when running @start_service.sql later on to (re)start the CC service this may result in OWB not being able to find the rtrepos.properties file. In that case please run reset_owbcc_home.sql again but now add the owb directory to the path (i.e. /ozg/app/oracle/product/11.2.0.3/db_1/owb). After this start_service will be able to start normally. When on a rac node, make sure to copy rtrepos.propereties to all other nodes.

The post-clone steps:

1) . ozg_init.env <sid of the ohi business intelligence database>

2) . ozg_init.env DB11G2

On Exadata or in absence of ozg_init.env make sure ORACLE_HOME ORACLE_SID en PATH environmental variables are set.

3) cd $ORACLE_HOME/owb/rtp/sql

4) sqlplus OWBSYS/<password OWBSYS user>

5) @reset_repository.sql <password OWBSYS user)

6) @stop_service.sql

7) exit

8) start the OWB repository listener ($OWB_HOME/bin/win32/startOwbbInst.bat in windows or find the linux equivalent and start it there)

9) start owb repository browser or openRAB.sh on linux (do not pay attention to certificate warning) and enter user OHI_BI_WS_OWN plus the rest of all the runtime repository database credentials. Click 'locations report' next.

10) Select LOC_DWH
Click on the 'Validation' link next to the location.

Change the connection details HOST, PORT and SERVICE_NAME to the appropriate values, they will initially show the values of the database the clone is taken from.

Press 'Update Details'

Enter password for the location (OBD_OWN password) and press 'Get Status'. What follows is shown next:

| | |
|---|---|
| Info | Connection successful |
| Info | Registered password is same as entered password |
| Info | This location has been deployed to. |
| Info | Installed components are at the correct version level (11.2.0.3.0) |
| Info | Registered connection properties (NLOZ08:1527:OACC) are the same as the connection properties for the Control Center (nloz08:1527:oacc) |

| | |
|---|---|
| Info | User OBD_OWN is registered as an OWB User. |
| Info | This Schema contains 2 invalid objects. |
| Info | All objects are compatible with the current Workspace. |
| Info | Actual version of this location (11.2) is the same as the version registered in the Control Center (11.2) |
| Info | There were 0 Warnings and 0 Errors. |

*table class = pdf-full-page-width*

On top of the screen 'Connection Tested OK.' is shown.

Execute the same steps for all locations except for the 'PlatformSchema'. This will be changed manually in a later stage of the process. Make sure that you enter the correct service name. Have a look at the example given for LOC_SRC_OPENZORG below.

Host: NLOZ11
Port: 1527
Service Name: ACCT.WORLD

– press 'update details'
– enter password (ozg_owner)
– press 'upgrade'
– press 'get status'

For LOC_SRC_EXT_TABLES and LOC_SRC_OPENZORG the following is shown.

| | |
|---|---|
| Info | Connection successful |
| Info | Registered password is same as entered password |
| Info | Location is not a deployment candidate so does not have an installed version |
| Info | Actual version of this location (11.2) is the same as the version registered in the Control Center (11.2) |
| Info | There were 0 Warnings and 0 Errors. |

*table class = pdf-full-page-width*

**Rule of thumb in changing locations is: First change connection details, press 'update details', enter password, press Get Status, enter password press upgrade (button next to password), enter password press Get Status. It may take a short while before the password has been registered.**

11) Log out, close browser and stop browser listener ($OWB_HOME/owb/bin/win32/stopOwbbInst.bat)

12) Go back to linux if not already there.
 . ozg_init.env <db_name>
and
. ozg_init.env DB11G2
have probably been executed in this session (see step 1 and 2). If not, execute the commands.

13) cd $ORACLE_HOME/owb/rtp/sql

14) sqlplus OWBSYS/<password OWBSYS user>

15) @UpdateLocation.sql

```
This sql script sets the properties for Host:Port:Service location, SQL*Net Connection location, and FTP or
General file system location in the OWB Client Repository.  It must be run from user OWBSYS.

The connection type must be Host:Port:Service, or SQL*Net Connection for Oracle Location, and FTP or General
for File System Location. The connection type of a location can not be changed by this script.

The location may be registered or unregistered.

Restart OWB client in order to see the new version in the UI.

Enter Workspace Name: OHI_BI_WS
Enter Workspace User Name: OHI_BI_WS_OWN
Enter Location Name: LOC_DWH
New Host: nloz08
New Port: 1527
New Service Name: oacc
New Net Service Name:
New Version: 11.2
New File Path:
New FTP Host:
New FTP Path:
Select Workspace Id for workspace OHI_BI_WS and user OHI_BI_WS_OWN
Workspace id = 2
Update location properties for LOC_DWH
Location Type =  Oracle Database
Location LOC_DWH Found
Connection Type = Default HOST:PORT:SERVICE Updating...
Updating CMPLocation_Host = nloz08
Updating CMPLocation_Port = 1527
Updating CMPLocation_ServiceName = oacc
Updating CMPLocation_Version = 11.2
PL/SQL procedure successfully completed.
```

Execute the same steps for the locations LOC_STG, LOC_WBX, LOC_SRC_EXT_TABLES and LOC_SRC_OPENZORG. Be careful to enter the correct LOC_SRC_OPENZORG credentials, because they differ from the other locations. In all cases enter the exact same credentials as previously in repository browser screens. In some cases the LOC_SRC_OPENZORG is the same as the other locations although this may seem odd, it is not relevant since ETL is executed over a database link in LOC_DWH and no objects are deployed to LOC_SRC_OPENZORG.

16) Now the Control Center must be edited. Update the connect_spec column in the wb_rt_service_nodes table. Connect_spec should by updated with host:port:service_name of the cloned database.

```
update wb_rt_service_nodes
set connect_spec = 'nloz08:1527:oacc'
;

commit
;

@UpdateControlCenter.sql
```

```
This sql script updates the host:port:service, net service name parameters for a control center in the OWB Client
Repository.  It must be run from user OWBSYS.

The connection type must be host:port:service or SQL*Net Connection. The connection type cannot be changed
using this script.

The default control center (DEFAULT_CONTROL_CENTER) cannot be changed using this script.

Enter Workspace Name: OHI_BI_WS
Enter Workspace User Name: OHI_BI_WS_OWN
Enter Control Center Name: DEFAULT_CONTROL_CENTER
Host: nloz08
Port: 1527
Service Name: oacc
New Net Service Name:
Select Workspace Id for workspace OHI_BI_WS and user OHI_BI_WS_OWN
Workspace id = 2
Update location properties for DEFAULT_CONTROL_CENTER
Location Type =
Control Center DEFAULT_CONTROL_CENTER Found
```

```
    Connection Type = Default HOST:PORT:SERVICE Updating...
    Updating CMPLocation_Host = nloz08
    CMPLocation_Host Not Found Inserting
    Updating CMPLocation_Port = 1527
    CMPLocation_Port Not Found Inserting
    Updating CMPLocation_ServiceName = oacc
    PL/SQL procedure successfully completed.
```

Although the script states that the default control center can not be updated, the procedure does what needs to be done.

17) @start_service.sql
This may fail the first time. Please execute again after a minute and the service will start.

```
Role set.
Available
Database managed service using nloz08:1527:oacc and home /ozg/app/oracle/product/11.2.0.3/db_1

PL/SQL procedure successfully completed.
```

Make absolutely sure that the service is DATABASE MANAGED (as opposed to CLIENT MANAGED). If this is not the case straight away, stop and start the service again.

18) @service_doctor.sql

```
Role set.

>>>>>> There are errors in one or more PL/SQL packages and functions
Platform properties have been loaded correctly
Platform location has been seeded correctly
NLS messages have been loaded correctly
The platform service is available
Service script is accessible to the database server
Connection information stored within the repository is correct
PL/SQL procedure successfully completed.
```

19) Correct the OBD_INPUT directory object in the database (login SYS as SYSDBA).

```
SQL> drop directory obd_input;

Directory dropped.

SQL> create directory obd_input as '/ozg/app/oracle/product/Zorg/oacc';

Directory created.

SQL> grant read, write on directory obd_input to public;

Grant succeeded.
```

20) Change SRC_OPENZORG database link to the correct Back Office source.

# APPENDIX E: MIGRATING OWB FROM SINGLE INSTANCE TO MULTI NODE RAC

Steps for OWB for RAC Environment

1. Create Unique Service Names

   ```
   srvctl add service -d rovac -s rovac1 -r rovac1
   srvctl add service -d rovac -s rovac2 -r rovac2
   ```

2. Start the service

   ```
   srvctl start service -d rovac -s rovac1
   srvctl start service -d rovac -s rovac2
   ```

3. Check the Status of added service's

   ```
   srvctl status service -d rovac
   ```

   ```
   Service rovac1 is running on instance(s) rovac1
   Service rovac2 is running on instance(s) rovac2
   ```

4. Complete post Clone steps as mentioned in the previous chapter. Please check the instructions mentioned below.

   - Use SCAN Address as the Host name for the Locations (SLC03FNR-S-SCAN.US.ORACLE.COM)

   - While running @UpdateControlCenter.sql step make sure to use the following parameters.

     | | |
     |---|---|
     | Enter Workspace Name: | OHI_BI_WS |
     | Enter Workspace User Name: | OHI_BI_WS_OWN |
     | Enter Control Center Name: | DEFAULT_CONTROL_CENTER |
     | Host: | SLC03FNR.US.ORACLE.COM **(Make sure to use host name)** |
     | Port: | 1521 |
     | Service Name: | ROVAC1 **(Make sure you use ROVAC1 and not ROVAC for Unique Service Name for Each Node)** |
     | New Net Service Name: | |

   - When running update control center use ROVAC1:

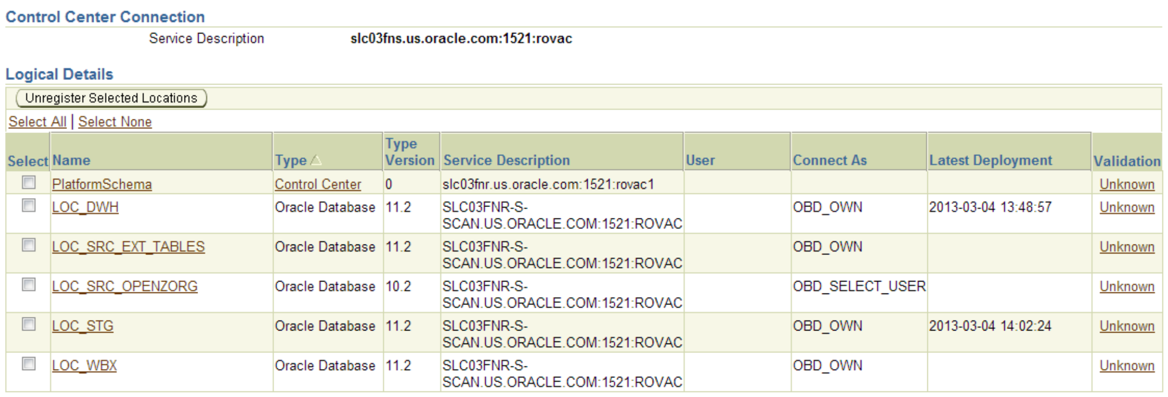
     ```
     update wb_rt_service_nodes
     set connect_spec = 'slc03fnr.oracle.com:1521:rovac1';

     commit;
     ```

   - Since service nodes need to be unique the service name for the node will be different than the service of the location, therefore the following error may be ignored.

     Error: Registered connection properties (SLC03FNR.ORACLE.COM:1521:ROVAC) are NOT the same as the connection properties for the Control Center (slc03fnr.us.oracle.com:1521:rovac1)

   - Location Report Should look as following.



**Control Center Connection**

Service Description slc03fns.us.oracle.com:1521:rovac

**Logical Details**

Unregister Selected Locations

Select All | Select None

| Select | Name | Type | Type Version | Service Description | User | Connect As | Latest Deployment | Validation |
|---|---|---|---|---|---|---|---|---|
| ☐ | PlatformSchema | Control Center | 0 | slc03fnr.us.oracle.com:1521:rovac1 | | | | Unknown |
| ☐ | LOC_DWH | Oracle Database | 11.2 | SLC03FNR-S-SCAN.US.ORACLE.COM:1521:ROVAC | | OBD_OWN | 2013-03-04 13:48:57 | Unknown |
| ☐ | LOC_SRC_EXT_TABLES | Oracle Database | 11.2 | SLC03FNR-S-SCAN.US.ORACLE.COM:1521:ROVAC | | OBD_OWN | | Unknown |
| ☐ | LOC_SRC_OPENZORG | Oracle Database | 10.2 | SLC03FNR-S-SCAN.US.ORACLE.COM:1521:ROVAC | | OBD_SELECT_USER | | Unknown |
| ☐ | LOC_STG | Oracle Database | 11.2 | SLC03FNR-S-SCAN.US.ORACLE.COM:1521:ROVAC | | OBD_OWN | 2013-03-04 14:02:24 | Unknown |
| ☐ | LOC_WBX | Oracle Database | 11.2 | SLC03FNR-S-SCAN.US.ORACLE.COM:1521:ROVAC | | OBD_OWN | | Unknown |

5. How to Register the Node's for RAC

No need of registering the first node. It will automatically get Registered as 1st Node.

We only need to register the 2nd Node.

**scp /u01/app/oracle/product/11.2.0/dbhome_1/owb/bin/admin/rtrepos.properties to 2nd Node ---> Before Registering this file should be copied to 2nd Node.**

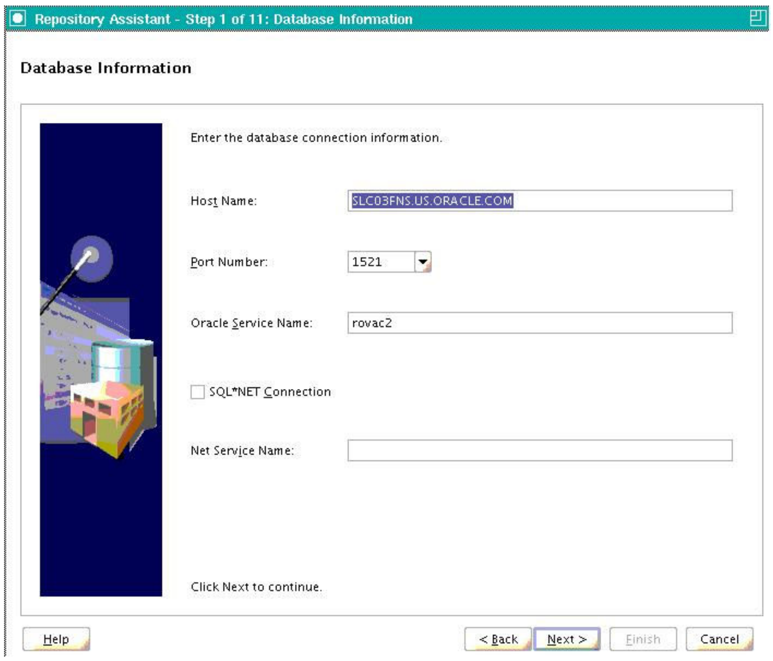
Register Node 2 for RAC Services.
Login to Node 2 and perform the following steps.

. ozg_init.env rovac2
. ozg_init.env DB11G2
set the Appropriate DISPLAY according to the VNC Session (eg: export DISPLAY=:10.0) if vnc running on port 10

On Exadata above steps would be:
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_1
export ORACLE_SID=<SID>
export PATH=$ORACLE_HOME/bin:$PATH
export DISPLAY=:<displaynumber>

cd $ORACLE_HOME/owb/bin/unix
./reposinst.sh

Enter the Hostname as Node2 Host name (SLC03FNS.US.ORACLE.COM)
Port 1521
Oracle Service Name = rovac2 (2nd Node Unique Service Name)



Repository Assistant - Step 1 of 11: Database Information

Database Information

Enter the database connection information.

Host Name: SLC03FNS.US.ORACLE.COM

Port Number: 1521

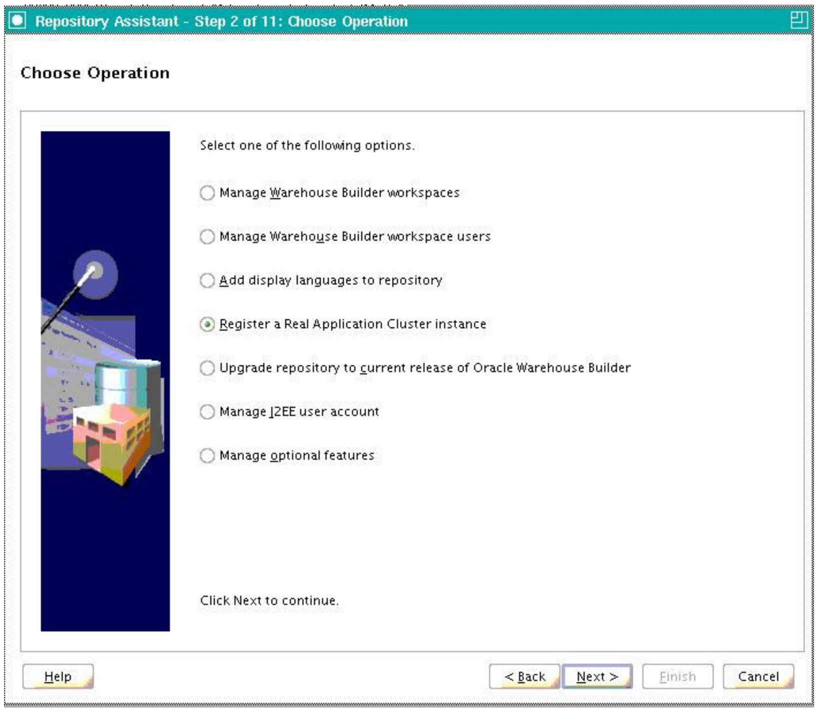
Oracle Service Name: rovac2

☐ SQL*NET Connection

Net Service Name:

Click Next to continue.

Select Register a Real Application Cluster Instance Option and click on Next



Repository Assistant - Step 2 of 11: Choose Operation

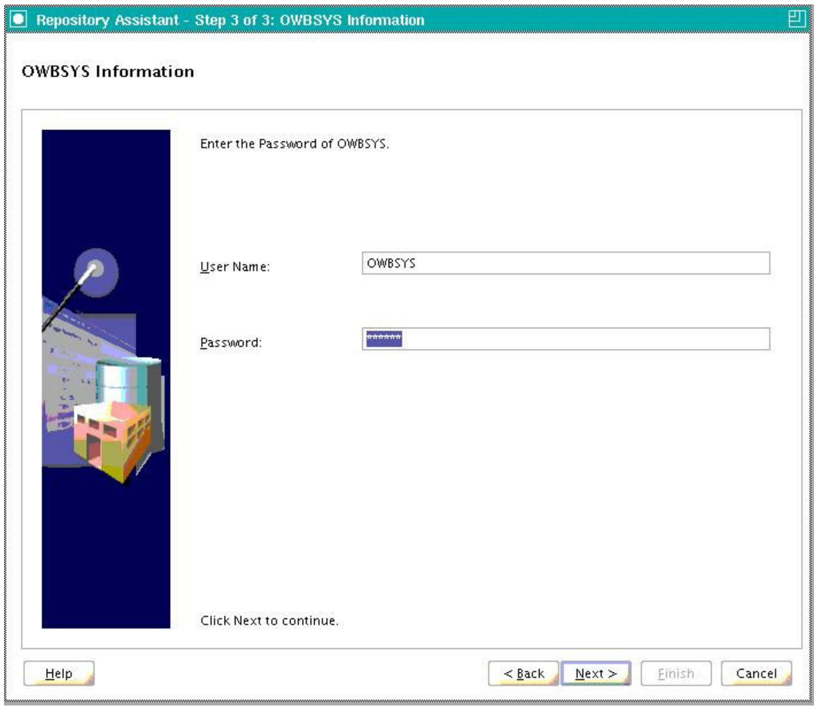
Choose Operation

Select one of the following options.

- ○ Manage Warehouse Builder workspaces
- ○ Manage Warehouse Builder workspace users
- ○ Add display languages to repository
- [✓ Register a Real Application Cluster instance]
- ○ Upgrade repository to current release of Oracle Warehouse Builder
- ○ Manage J2EE user account
- ○ Manage optional features

Click Next to continue.

Enter the owbsys password and click on Next



Repository Assistant - Step 3 of 3: OWBSYS Information

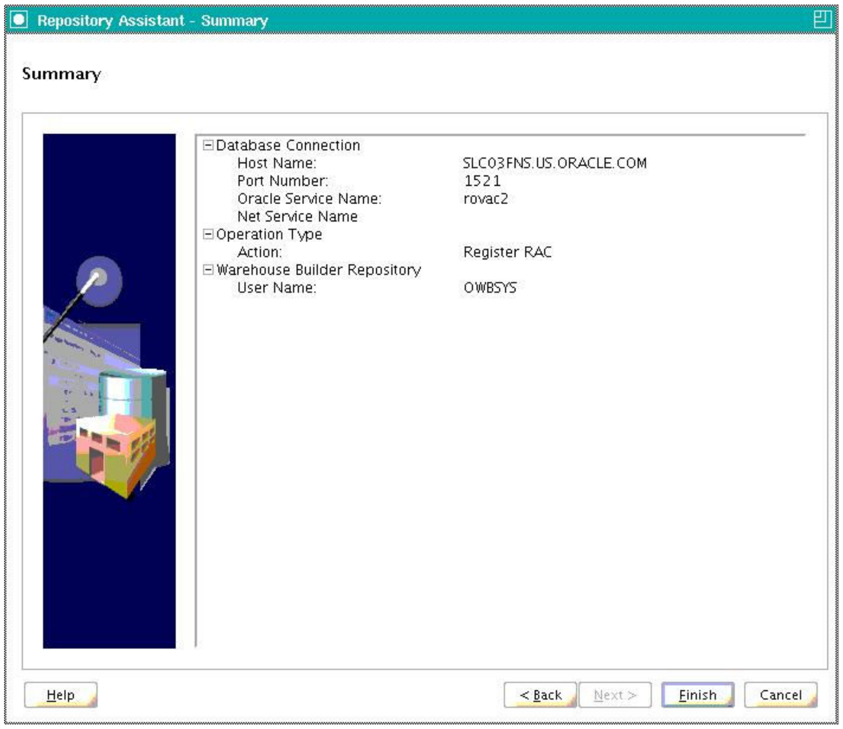
OWBSYS Information

Enter the Password of OWBSYS.

User Name: OWBSYS

Password: ******

Click Next to continue.



Summary

```
⊟Database Connection
    Host Name:              SLC03FNS.US.ORACLE.COM
    Port Number:            1521
    Oracle Service Name:    rovac2
    Net Service Name
⊟Operation Type
    Action:                 Register RAC
⊟Warehouse Builder Repository
    User Name:              OWBSYS
```

Click on Finish.

A Message will appear that it has successfully registered the Node.

6. Finally the obd_own user needs permssion on the WB$_IV_CONTROL_CENTERS view for OBDPATCH.pl to succeed.

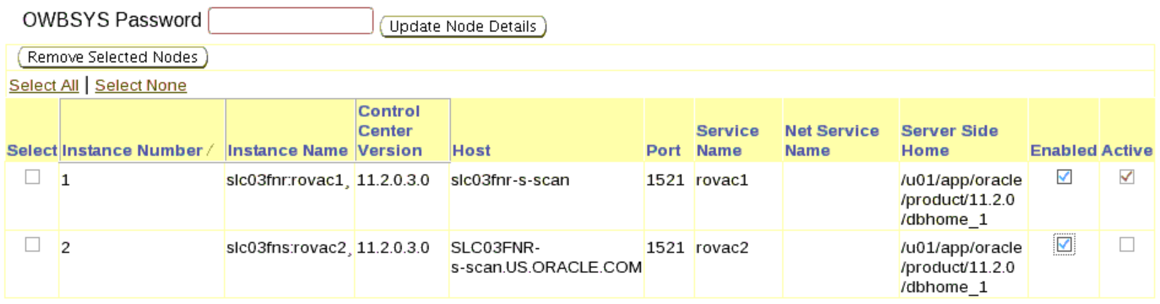
SQL> grant select on WB$_IV_CONTROL_CENTERS to OBD_OWN with grant option;

To Verify the Node is Registered do the following.
Start the Repository Browser Listener:

./startOwbbInst.sh

Login to Repository Browser Listener
https://slc03fnr.us.oracle.com:8999/owbb/RABLogin.uix

Click on Service Node Report.. Following should be visible.



| Select | Instance Number | Instance Name | Control Center Version | Host | Port | Service Name | Net Service Name | Server Side Home | Enabled | Active |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | slc03fnr:rovac1 | 11.2.0.3.0 | slc03fnr-s-scan | 1521 | rovac1 | | /u01/app/oracle/product/11.2.0/dbhome_1 | ☑ | ☑ |
| ☐ | 2 | slc03fns:rovac2 | 11.2.0.3.0 | SLC03FNR-s-scan.US.ORACLE.COM | 1521 | rovac2 | | /u01/app/oracle/product/11.2.0/dbhome_1 | ☑ | ☐ |

The first time you login the 2nd node will not be enabled. Whether to enable node 2 depends on the setup you choose. Please refer to the next paragraph for the details; the default situation is 2 nodes enabled.

This screen shows that Unique Service names are used for each instance, ROVAC1 and ROVAC2.

**How To Verify if OWB is Installed Correctly on RAC and Exadata (Doc ID 455999.1)**

1. UNIQUE Service Names

srvctl status service -d rovac

Service rovac1 is running on instance(s) rovac1
Service rovac2 is running on instance(s) rovac2

2. LISTENER Configuration
   We are using SCAN listener, so tnsnames.ora file has been updated as follows.

```
ROVAC1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = slc03fnr-vip.us.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rovac1)
    )
  )

ROVAC2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = slc03fns-vip.us.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rovac2)
    )
  )
```

3. To get the instance and status per node, execute the following command on one of the nodes:

```
select inst_id
,      instance_number
,      instance_name
,      database_status
,      status
,      host_name
from   gv$instance
;

   INST_ID INSTANCE_NUMBER INSTANCE_NAME    DATABASE_STATUS   STATUS       HOST_NAME
---------- --------------- ---------------- ----------------- ------------ -------------
         1 1               rovac1           ACTIVE            OPEN         slc03fnr
         2 2               rovac2           ACTIVE            OPEN         slc03fns
```

4. To get the service_name for all nodes, execute the following command on one of the nodes:

```
select inst_id
,      name
from   gv$services
;

INST_ID    NAME
---------- ----------------------------------------------------------------
         1 rovac1
         2 rovac2
```

5. If the service_names have been assigned to the instances, use following query to verify this.

```
select s.inst_id
,      instance_number
,      instance_name
,      name service_name
,      host_name
from   gv$services s
,      gv$instance i
where  s.inst_id = i.inst_id
;

 INST_ID   INSTANCE_NUMBER INSTANCE_NAME    SERVICE_NAME    HOST_NAME
---------- --------------- ---------------- --------------- ---------------
         1 1               rovac1           rovac1          slc03fnr
         2 2               rovac2           rovac2          slc03fns
```

6. How the OWB configuration should look like

   For OWB 11.2 queries are as follows:

```
select * from owbsys.owbrtps;

KEY VALUE
----------------------------------- ----------------------------------------
11.2.0.3.0[2] /u01/app/oracle/product/11.2.0/dbhome_1
11.2.0.3.0[1] /u01/app/oracle/product/11.2.0/dbhome_1
```

7. Check Service Node Report under OWB Browser Listener
   https://slc03fnr.us.oracle.com:8999/owbb/RABLogin.uix

   1 slc03fnr:rovac1 11.2.0.3.0 SLC03FNR.US.ORACLE.COM 1521 ROVAC1
   /u01/app/oracle/product/11.2.0/dbhome_1
   2 slc03fns:rovac2 11.2.0.3.0 SLC03FNS.US.ORACLE.COM 1521 ROVAC2
   /u01/app/oracle/product/11.2.0/dbhome_1

8. Enable all RAC nodes

```
select schema_user, what,
to_char(last_date, 'DD-MON-YYYY HH24:MI:SS') last_date,
to_char(next_date, 'DD-MON-YYYY HH24:MI:SS') next_date
from user_jobs
where what like '%check_service%'
```

```
SCHEMA_USER WHAT                                  LAST_DATE            NEXT_DATE
----------- ------------------------------------- -------------------- --------------------
OWBSYS      wb_rti_service_job.check_service(1);  09-OKT-2012 09:27:54 09-OKT-2012 09:33:54
OWBSYS      wb_rti_service_job.check_service(26); 09-OKT-2012 09:27:54 09-OKT-2012 09:33:54
```

```
begin
  wb_rti_service_job.check_service(1);
  wb_rti_service_job.check_service(26);
end;
/
```

9. Locations

   Locations that point to the RAC should be using Net Service Names. If host:port:service Locations are used, this would cause mappings to connect to a specific RAC node during mapping execution. A crash of such a node would stop the mapping. This can be prevented by using Net Service Names Locations.

**Reference MOS Note ID's used to configure.**

1. How to Install a Warehouse Builder Repository on a RAC (Doc ID 459961.1)
2. How To Verify if OWB is Installed Correctly on RAC and Exadata (Doc ID 455999.1)
3. How to Regenerate the rtrepos.properties File (Doc ID 473151.1)
4. Option To Register Node is Missing when Configuring OWB 11g On RAC (Doc ID 791362.1)

## OWB RAC Rationale

With regard to RAC there are two configuration modes or rather setups possible. In Setup 1 there is one RAC node exclusively for ETL, and one exclusively for "Query". Setup 2 has two nodes, both for query and for ETL.
In setup 1 ETL was performed on one node, and node 2 was queried constantly by two "Query" sessions over a database link. The databaselink connected to node 2 only.

In setup 2 ETL was performed on both nodes and both nodes were queried constantly by two "Query" sessions over a database link. The database link connected to the service.

Multiple tests were performed to define optimum: Setup 1 or Setup 2. AWR snapshots were taken during the tests. Below is a snippet from the AWR report from Setup 1

**Global Cache Load Profile**

| | Per Second | Per Transaction |
|---|---|---|
| Global Cache blocks received: | 1.14 | 17.69 |
| Global Cache blocks served: | 1.58 | 24.40 |
| GCS/GES messages received: | 130.18 | 2,013.60 |
| GCS/GES messages sent: | 72.16 | 1,116.16 |
| DBWR Fusion writes: | 0.05 | 0.72 |
| Estd Interconnect traffic (KB) | 83.06 | |

Below is a snippet from the AWR report from Setup 2

**Global Cache Load Profile**

| | Per Second | Per Transaction |
|---|---|---|
| Global Cache blocks received: | 0.39 | 4.87 |
| Global Cache blocks served: | 1.07 | 13.45 |
| GCS/GES messages received: | 75.68 | 950.75 |
| GCS/GES messages sent: | 42.95 | 539.55 |
| DBWR Fusion writes: | 0.03 | 0.39 |
| Estd Interconnect traffic (KB) | 46.50 | |

From this can be deducted that there is far more communication between the nodes in Setup 1 than in Setup 2. Alhough this might look strange since Partition Exchange Loading is used (PEL), this is actually not strange since PEL only is used in the fact tables and not in dimension tables.

Extraction Transformation and Loading Times:
Setup 1 approx: 5hr:20 min
Setup 2 approx: 4hrs:53 min

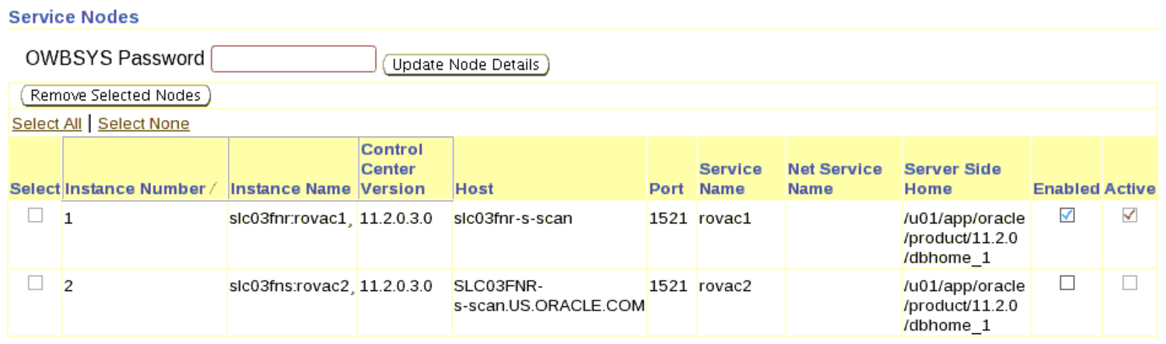
From the tests is also concluded that Setup 2 performed ETL quicker than Setup 1, although the load times did not differ too much. Therefore preferred setup is setup 2, ETL and Query on both nodes.

## OWB RAC SETUP 1

This setup describes pinning the ETL to node 1, reason for this is that other queries can be directed to node 2.

This Oracle Warehouse Builder setup for OHI Business Intelligence consists of the following;
OHI Back Office - on multi node rac node 1 and node 2
OHI Business Intelligence - on multi node rac, with OWB configured to run ETL on node1, node 2 is for custom ETL and query (EUL).

Setup 1 relies on an application server with wallet and tnsnames entry rovac1; sqlplus connects to the first node. In the Runtime Audit Browser only one node is enabled for ETL.



| Select | Instance Number | Instance Name | Control Center Version | Host | Port | Service Name | Net Service Name | Server Side Home | Enabled | Active |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | slc03fnr:rovac1, | 11.2.0.3.0 | slc03fnr-s-scan | 1521 | rovac1 | | /u01/app/oracle /product/11.2.0 /dbhome_1 | ✓ | ✓ |
| ☐ | 2 | slc03fns:rovac2, | 11.2.0.3.0 | SLC03FNR-s-scan.US.ORACLE.COM | 1521 | rovac2 | | /u01/app/oracle /product/11.2.0 /dbhome_1 | ☐ | ☐ |

Queries are bound to the instance by an after logon trigger or database parameter:
parallel_force_local = true

```
CREATE OR REPLACE TRIGGER trg_parallel_force_local AFTER logon ON database
  DECLARE
    l_osuser             VARCHAR2(1000);
    l_application_server VARCHAR2(1000);
  BEGIN
    SELECT lower(sys_context('USERENV', 'OS_USER')) l_osuser,
      lower(sys_context('USERENV', 'HOST')) l_application_server
    INTO l_osuser,
      l_application_server
    FROM DUAL;
    IF USER = 'BATCH' AND l_osuser = 'batch'
       AND l_application_server = 'slc03fnr'
    THEN --- change l_application_server to the name of your application server
      execute immediate 'ALTER session SET parallel_force_local = true';
    END IF;
  END;
  /
```

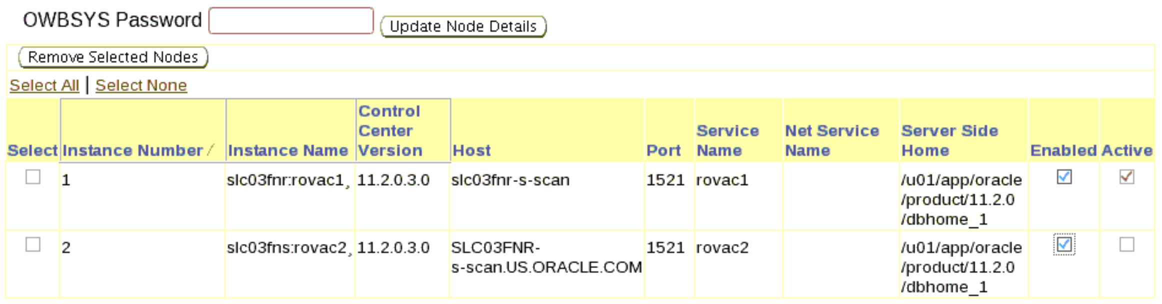
The wallet entries on the application server are directed to the first instance (node 1), and “Scriptaanvragen” are supplied with the ORACLE_SID of the first node.

## OWB RAC SETUP 2

This setup describes ETL that can run on both nodes, this is the preferred setup.

OHI Back Office - on multi node rac
OHI Business Intelligence - on multi node rac with OWB to configured ETL to run on both nodes, and custom ETL / query (EUL) on two nodes.

In the Runtime Audit Browser two nodes are enabled for ETL.



| Select | Instance Number | Instance Name | Control Center Version | Host | Port | Service Name | Net Service Name | Server Side Home | Enabled | Active |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | slc03fnr:rovac1, | 11.2.0.3.0 | slc03fnr-s-scan | 1521 | rovac1 | | /u01/app/oracle /product/11.2.0 /dbhome_1 | ✓ | ✓ |
| ☐ | 2 | slc03fns:rovac2, | 11.2.0.3.0 | SLC03FNR- s-scan.US.ORACLE.COM | 1521 | rovac2 | | /u01/app/oracle /product/11.2.0 /dbhome_1 | ✓ | ☐ |

Queries are not bound to the instance by either a logon trigger or database parameter:
parallel_force_local = false

```
CREATE OR REPLACE TRIGGER trg_parallel_force_local AFTER logon ON database
  DECLARE
    l_osuser             VARCHAR2(1000);
    l_application_server VARCHAR2(1000);
  BEGIN
    SELECT lower(sys_context('USERENV', 'OS_USER')) l_osuser,
      lower(sys_context('USERENV', 'HOST')) l_application_server
    INTO l_osuser,
      l_application_server
    FROM DUAL;
    IF USER = 'BATCH' AND l_osuser = 'batch' AND l_application_server = 'slc03fnr'
    THEN --- change l_application_server to the name of your application server
      execute immediate 'ALTER session SET parallel_force_local = false;
    END IF;
  END;
  /
```

The wallet entries on the application server are directed to the service, not to a specific node.

## RESTRICTED MODE

While patching an OHIBI version, it is recommended to have the database in restricted mode. For details about restriced mode see the Oracle Database Administrator Guide, for example part E25494-03, Altering Database Availability, Restricting Access to an Open Database.
To restrict access to an instance use the following command:

SQL> alter system enable restricted session;

Please note that if on a RAC cluster all the instances should be in restricted modes.
If however the database uses a dynamic listener, then the OHIBI database cannot be in restricted mode. In such a case it is the responsiblity of the DBA to make sure there are no users logged in at the time of patching a release. To make sure there are no users logged in at the moment of patching and during the patch process make sure there are no active sessions from existing users by quering gv$session. Also make sure no user accounts can login to the database, by either informing your users, or locking accounts.
As an example two examples SQL are provided here:

SQL> select sid,serial#,username,osuser,machine from gv$session where osuser != 'oracle';
SQL> alter user <username> account lock;

To allow access to the instance use the following command;

SQL> alter system disable restricted session;

## CHAINED ROWS

```
When installing or patching a release, chained rows will be detected
if present in user_tables. To removed the chained rows do the
following in SQLPlus.

SQL> alter table <TABLE_NAME> move;
SQL> alter table <TABLE_NAME> compute statistics;

After which the indexes are unusable and need to be rebuild with step
870 of the OZGPATCH.pl script.
```