

Oracle Insurance Datamarts for Health

Administrator Reference

version 9.29

Part number: E35899-01

June 2012

Copyright © 2011, 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Where an Oracle offering includes third party content or software, we may be required to include related notices. For information on third party notices and the software and related documentation in connection with which they need to be included, please contact the attorney from the Development and Strategic Initiatives Legal Group that supports the development team for the Oracle offering. Contact information can be found on the Attorney Contact Chart.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

CHANGE HISTORY

Release	Version	Changes
10.12.2.0.0.0	9.29	<ul style="list-style-type: none">• Added change history paragraph.• Added description for Oracle Business Intelligence translations external file (WBX_OBIEE_TRANSLATIONS.csv)• Added description for configuring generic structure of claim properties• Added description how to configure DWH_OHI_BI.rpd connection pools.

CONTENTS

Change History	ii
Introduction	5
Important documentation	5
Software used	5
Overview	6
Part I Installation	7
Introduction	7
Setting up Oracle Insurance Datamarts for Health environment	7
OS set-up	7
Software installation	7
Creation of an Oracle Insurance Datamarts for Health database	8
Database users	10
Database link	11
Database directories	11
General instructions	11
Installation of Oracle Warehouse Builder (OWB) Repository	12
Client installation	12
Server installation	12
Software	12
Repository	12
Generation and installation of Oracle Insurance Datamarts for Health objects.	21
Installation	21
Release	21
System parameters	21
External files	23
Set-up of Oracle Business Intelligence Enterprise Edition environment	26
Software installation	26
Installation of Oracle Insurance Datamarts for Health Repository (RPD)	26
Part II Loading	30
Introduction	30
Extracting OHI Back Office (ZRG0E01S)	30
Transforming data (ZRG0S01S)	33

Loading Oracle Insurance Datamarts for Health (ZRGOD01S)	33
Undoing load runs (ZRGO004S)	34
Checks	35
Audit information	36
Log information	36
Resolving errors	37
Checking status request load scripts	37
Verifying the load	38
Part III Application management	39
Introduction	39
Functional management	39
Logging and validation	39
Authorization	42
Configuration of generic structure for claim properties	43
Technical management	49
Cleaning up OWB Audit data	49
New releases of Oracle Insurance Datamarts for Health	49
Reorganization of tables	49
Compression of partitioned fact tables.	50
OWB Runtime service	50
Part IV Appendices	52
Appendix A. Cloning Oracle Insurance Datamarts for Health environments	52
Appendix B. Use of WBX_LAADDRUN_ERRORS_VW and WBX_MAPPINGS_VW views.	52

INTRODUCTION

The purpose of this document is to give an overview of the architecture of Oracle Insurance Datamarts for Health, 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health and contain information that is (potentially) required for installation and management.

1. Standard product manuals from Oracle DBMS, Oracle Warehouse Builder and Oracle Discoverer. Documentation on these Oracle products can be found on the Oracle support website (<http://support.oracle.com>)
2. Certification form 2012.02.0.0000
3. CTA13508.doc: Oracle Health Insurance Installation, Configuration and DBA Manual
4. CTA13535.doc: Oracle Health Insurance Security Aspects

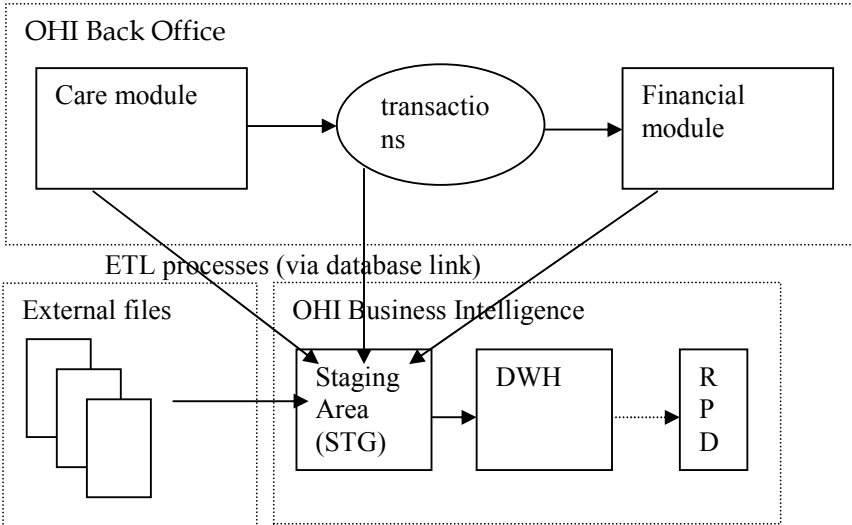
SOFTWARE USED



Certifications matrix 2012.02.0.0000

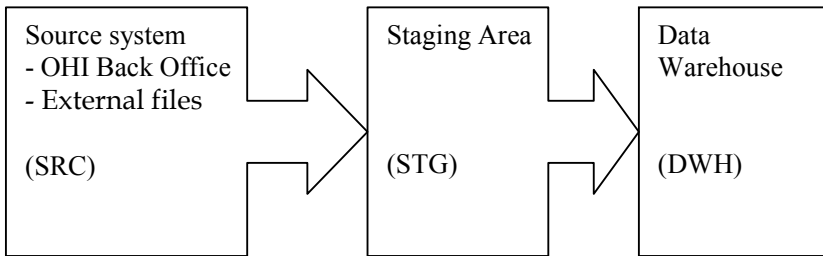
OVERVIEW

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



Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health. 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 INSURANCE DATAMARTS FOR HEALTH ENVIRONMENT

OS set-up

Oracle Insurance Datamarts for Health runs on a database server. The load run of Oracle Insurance Datamarts for Health requires that there is an application server on which the OHI Back Office application is installed. Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health must be installed in a dedicated database. It is not permitted to install Oracle Insurance Datamarts for Health in the same database as OHI Back Office, as Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health. The installation of Oracle Insurance Datamarts for Health 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 2012.03.0.0000

Creation of an Oracle Insurance Datamarts for Health database

A database needs to be created first before Oracle Insurance Datamarts for Health 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	
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 table spaces 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
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

Tablespace	Initial	Next
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 table spaces must comply with the following requirements:

1. Locally Managed
2. System managed extent allocation
3. Automatic Segment Space Management
4. 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 Insurance Datamarts for Health database:

User	ID	Description
Oracle Insurance Datamarts for Health repository owner	OWBSYS	This is the owner of the OWB repository on the Oracle Insurance Datamarts for Health database server. This user is created during the installation of the OWB 11g software. The data/index tablespace is OWBSYS.
Oracle Insurance Datamarts for Health Workspace owner	OHI_BI_WS_OWN	This is the owner or the workspace in which the Oracle Insurance Datamarts for Health objects have been created and are executed.
Oracle Insurance Datamarts for Health owner	OBD_OWN	This is the owner of the Oracle Insurance Datamarts for Health objects. This user must be created using the Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health 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	<p>This is the user under which the selections on OHI Back Office are performed. This user should be assigned the following privileges:</p> <pre> create session alter session ozg_rol_select select, insert, delete on ozg_owner.geb#obd_declaraties execute on ozg_owner.alg_tab_pck execute on ozg_owner.fin_fpm_vars_pck select on v_\$database execute on ozg_owner.geb_odr_pck </pre>

Database link

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

```

General instructions

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



Oracle Health Insurance Installation, Configuration and DBA Manual
(chapter 5)

INSTALLATION OF ORACLE WAREHOUSE BUILDER (OWB) REPOSITORY

CLIENT INSTALLATION

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

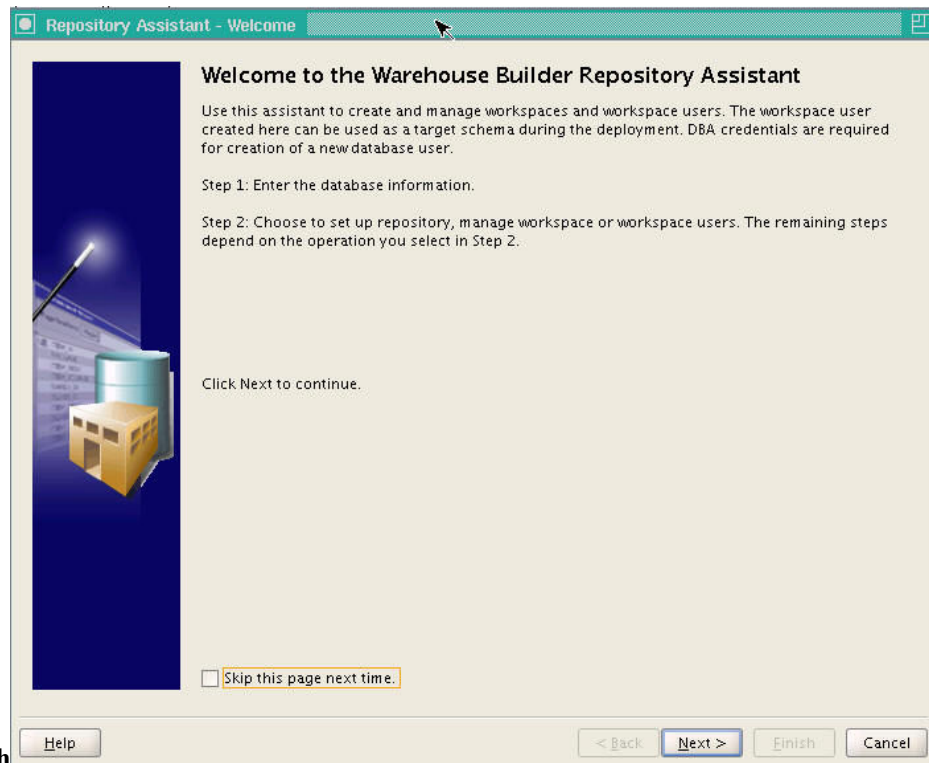
An OWB repository has to be present on the Oracle Insurance Datamarts for Health database server in order to run the mappings generated by OWB. Information is stored in the repository relating to installation and running of mappings.

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;

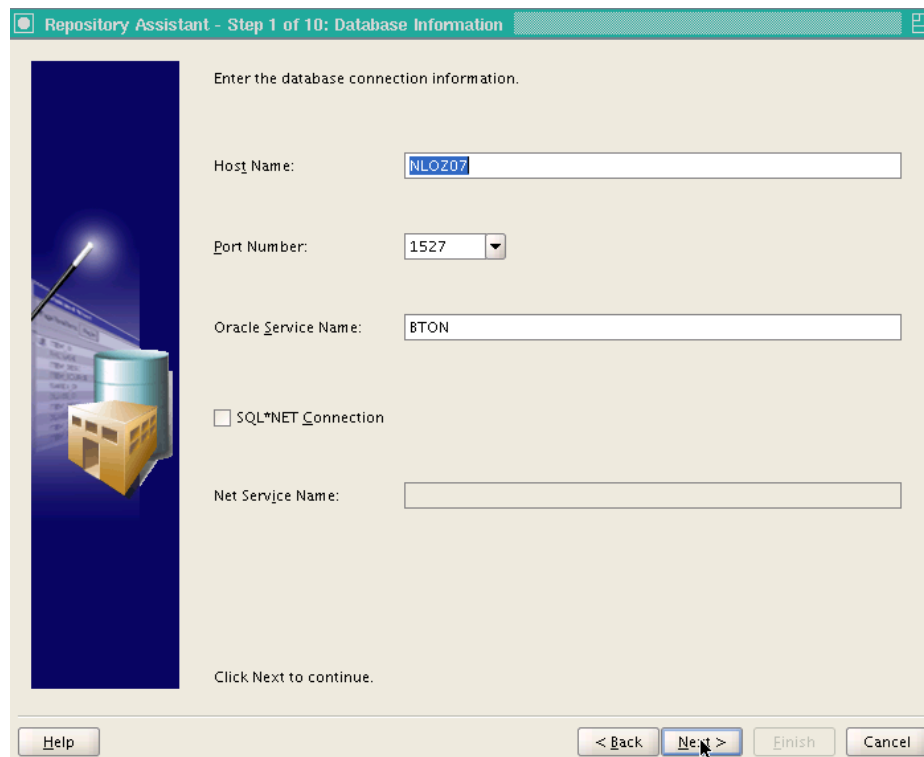
Exit;

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



9) `./reposinst.sh`

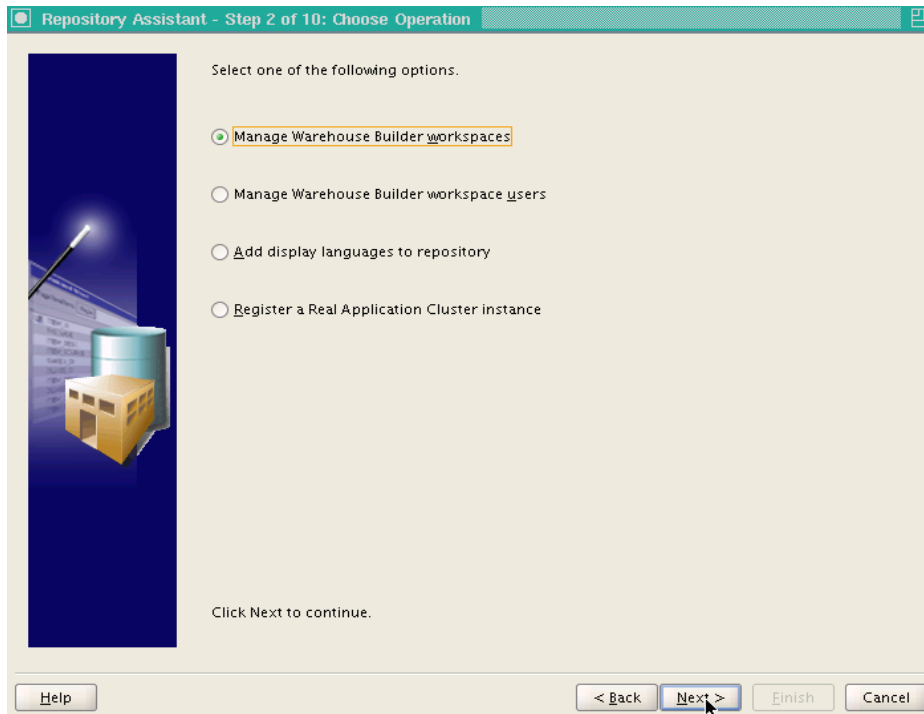
Click on Next>



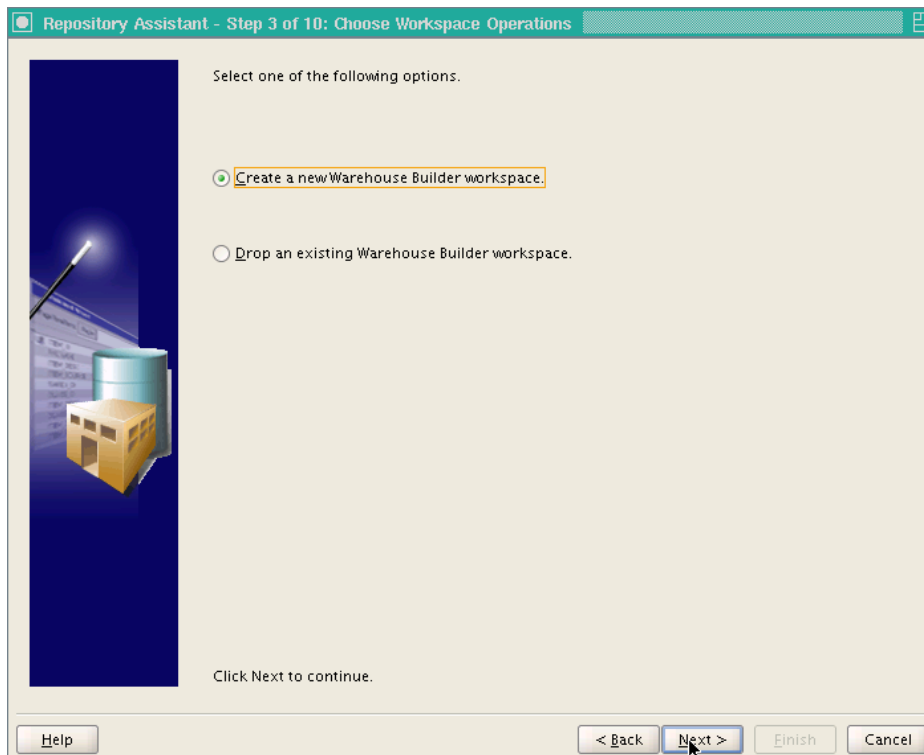
Host Name: Name of the host server

Port Number: Port of the listener for the database

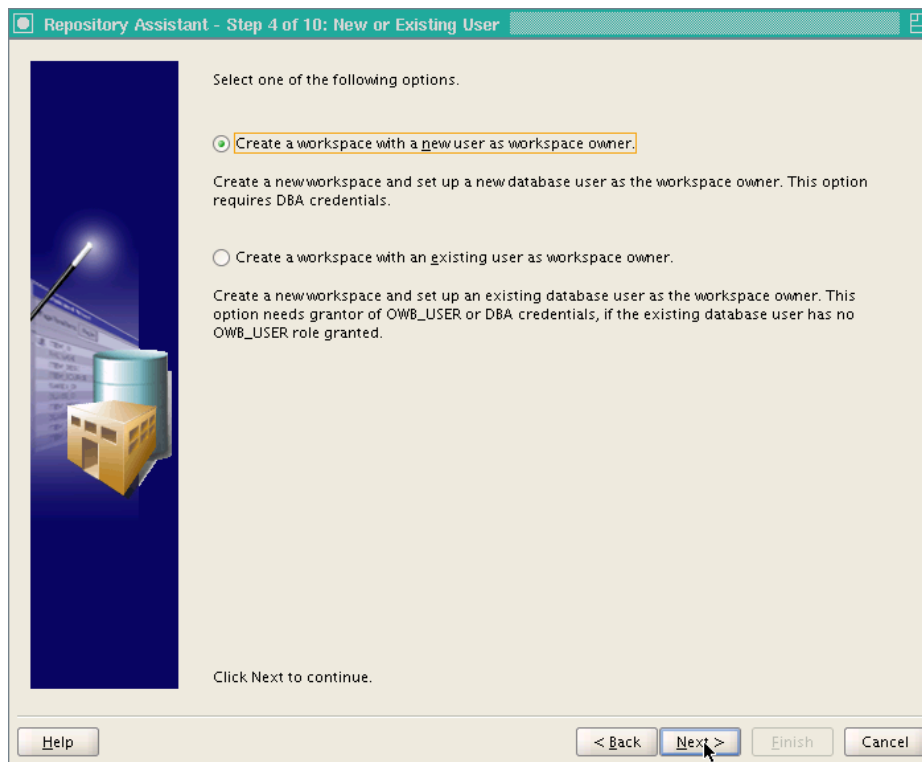
Oracle Service Name: `<ohi_db_name>`
Then click on Next>



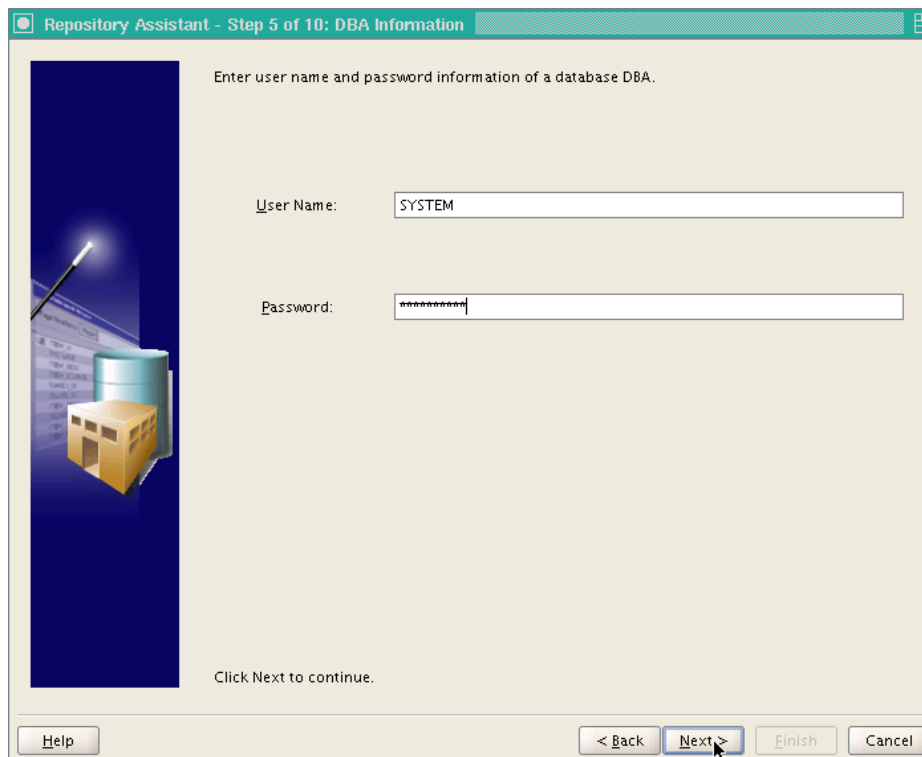
Choose "Manage Warehouse Builder workspaces" and click on Next>



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



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



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.

Help < Back Next > Finish Cancel

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

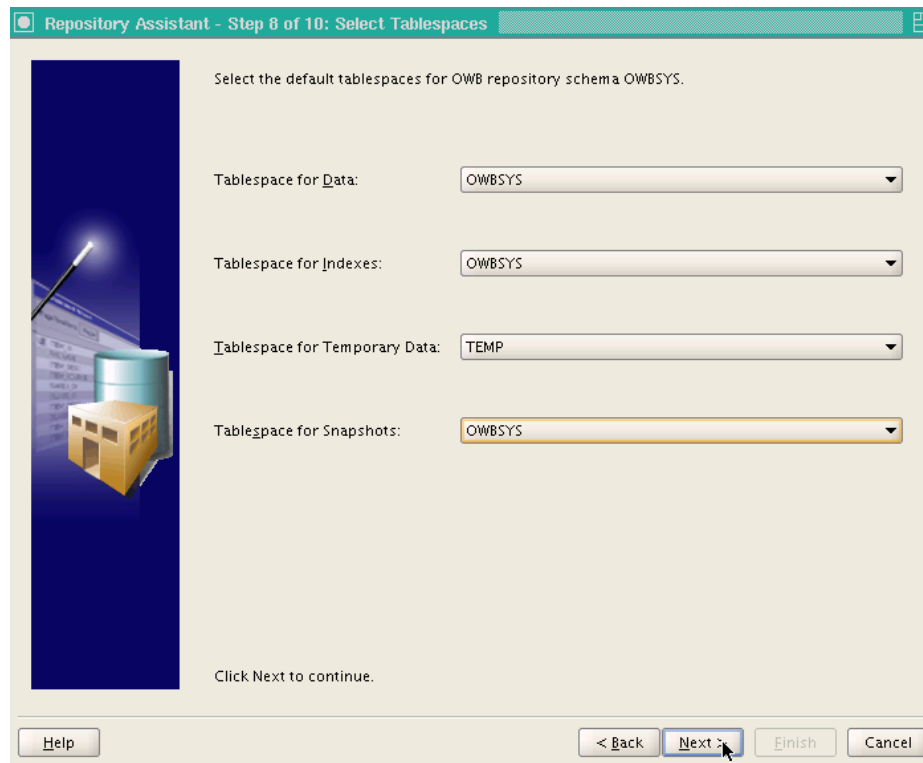
Password: *****

Click Next to continue.

Help < Back Next > Finish Cancel

User Name: OWBSYS
 Password: <password>

Click on Next>



Repository Assistant - Step 8 of 10: Select Tablespaces

Select the default tablespaces for OWB repository schema OWBSYS.

Tablespace for Data: OWBSYS

Tablespace for Indexes: OWBSYS

Tablespace for Temporary Data: TEMP

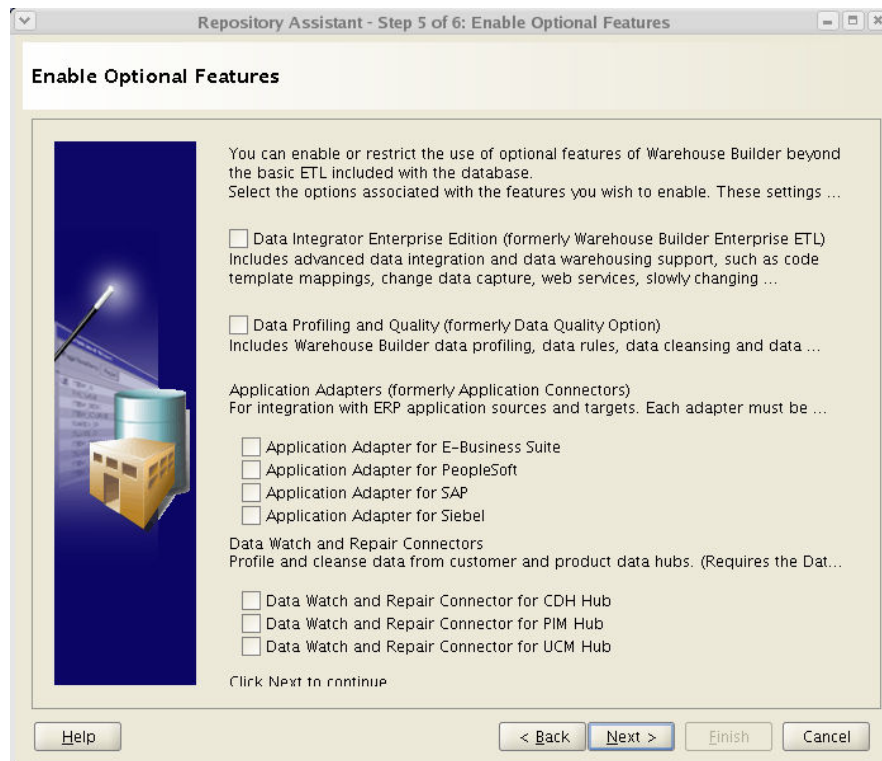
Tablespace for Snapshots: OWBSYS

Click Next to continue.

Help < Back Next > Finish Cancel

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

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

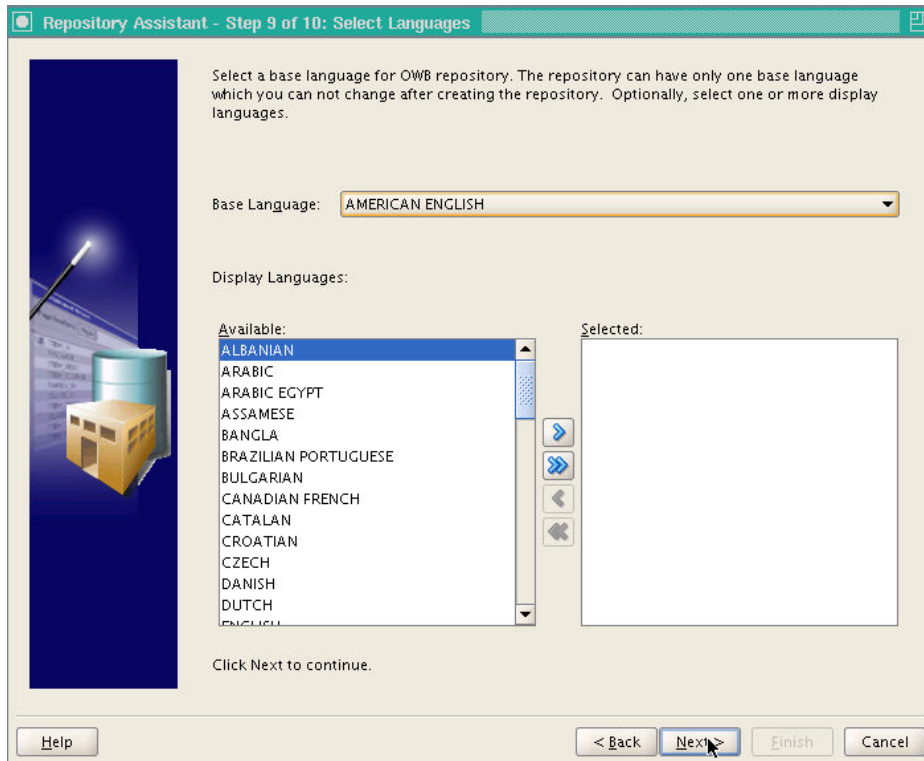
Data Watch and Repair Connectors
Profile and cleanse data from customer and product data hubs. (Requires the Dat...

☐ 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

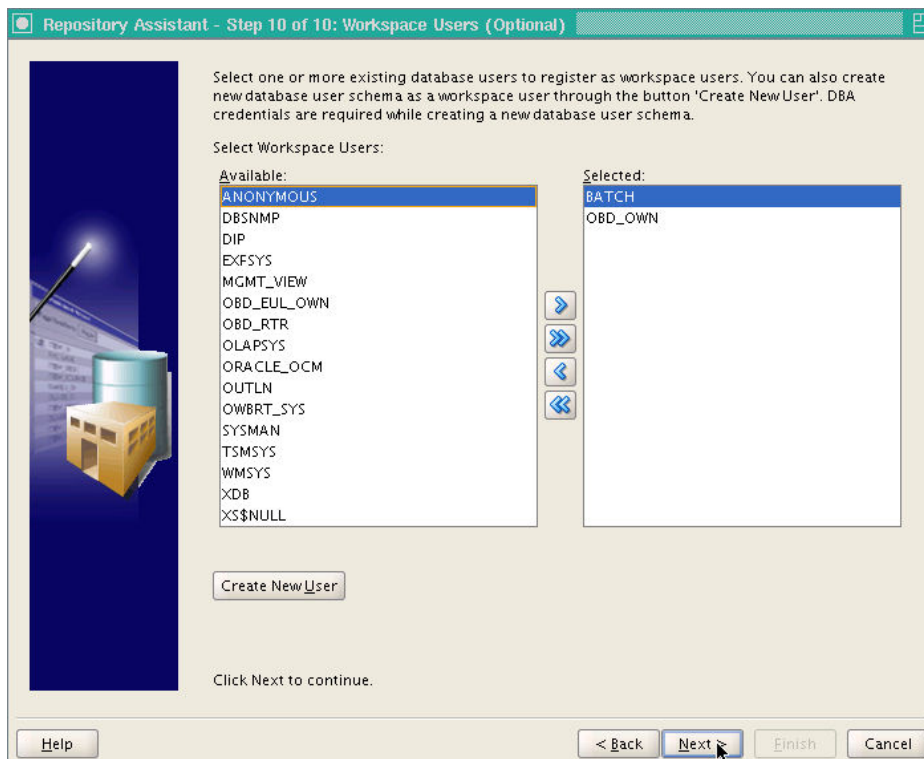
Help < Back Next > Finish Cancel

Deselect all optional features.

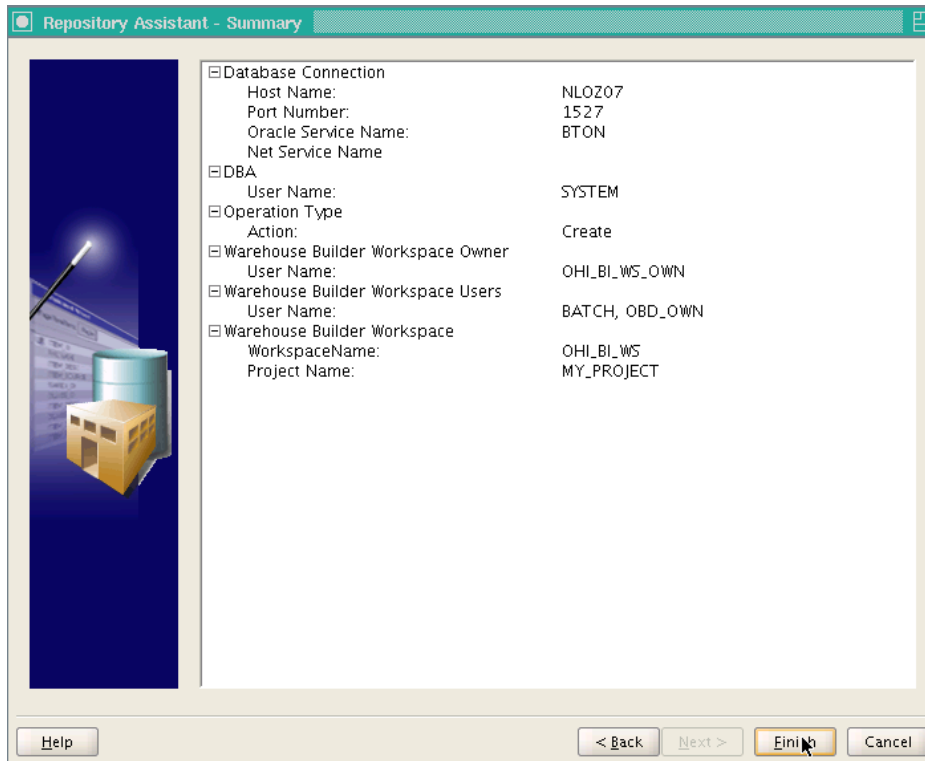


Base Language: AMERICAN ENGLISH

Click on Next>

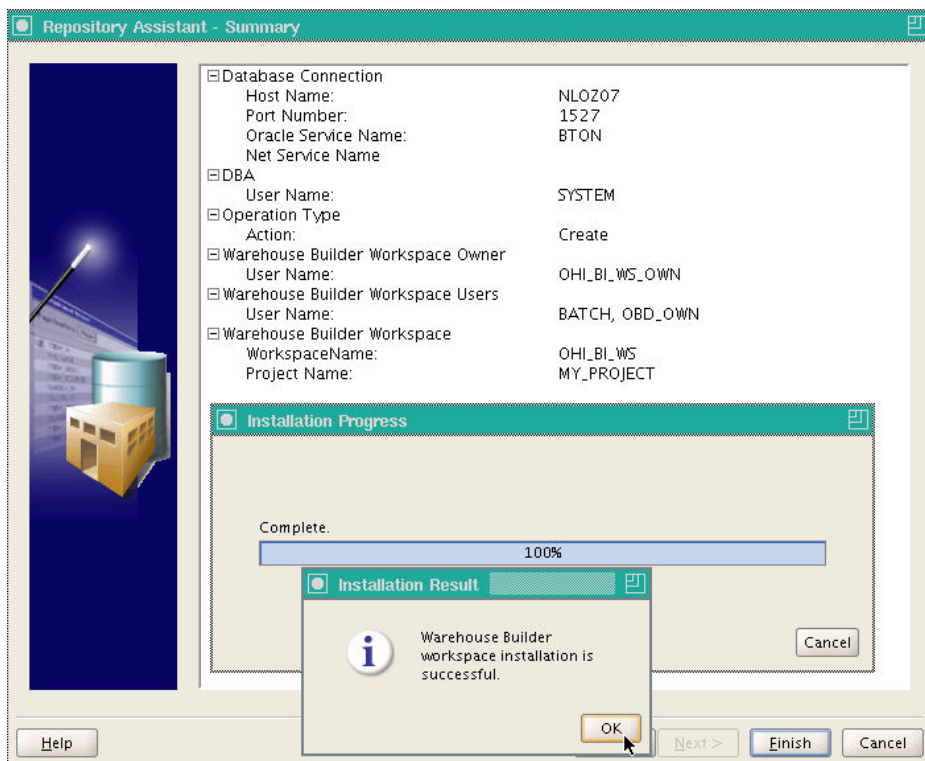


Add the users BATCH and OBD_OWN as Workspace Users if these already exist. If not, these users will have to be added later using /reposinst.sh. Click on Next>



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.



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 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 INSURANCE DATAMARTS FOR HEALTH OBJECTS.

INSTALLATION

Release

Installation of Oracle Insurance Datamarts for Health (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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health environment on the database server. This file contains the control variables for the Oracle Insurance Datamarts for Health loading process. These parameters are customer-specific and the value of a number of these parameters **must** be adjusted in the file **before** Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health assumes that the financial data can be retrieved from OHI Back Office.
EOZNLS	<p>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.</p> <p>1) Decimals separated by a comma Thousands separated by a period</p> <p>e.g.: 1.000.000,001 This style is used in the DUTCH NLS Language, among others.</p> <p>Value for EOZNLS: nls_numeric_characters = ','</p> <p>This is the default value for the parameter.</p> <p>2) Thousands separated by a comma Decimals separated by a period</p> <p>e.g.: 1,000,000.001 This style is used in the AMERICAN NLS Language, among others.</p> <p>Value for EOZNLS: nls_numeric_characters = '.,'</p>

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 self-provided flex fields, namely:

Parameter	Description
CCTAE1	The description of the first alphanumeric field of the group contract.
CCTAE2	The description of the second alphanumeric field of the group contract.
CCTAE3	The description of the third alphanumeric field of the group contract.
CCTAE4	The description of the fourth alphanumeric field of the group contract.
CCTAE5	The description of the fifth alphanumeric field of the group contract.
CCTAE6	The description of the sixth alphanumeric field of the group contract.
CCTAE7	The description of the seventh alphanumeric field of the group contract.
CCTAE8	The description of the eighth alphanumeric field of the group contract.
CCTAE9	The description of the ninth alphanumeric field of the group contract.
CCTAE10	The description of the tenth alphanumeric field of the group contract.
CCTDE1	The description of the first date field of the group contract.
CCTDE2	The description of the second date field of the group contract.
CCTDE3	The description of the third date field of the group contract.
CCTNE1	The description of the first numeric field of the group contract.
CCTNE2	The description of the second numeric field of the group contract.
CCTNE3	The description of the third numeric field of the group contract.

This shows the flex fields that should be loaded in the collective contracts dimension. The values that have to be entered here are the descriptions of the flex fields as shown in the 'Flex field' window in the Back Office subsystem of OHI Back Office. The default value is empty.

The group contract flex fields have not yet been added to the Discover End User Layer or the OBIEE RPD. If these fields are desired they have to be added manually to the repository concerned.

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.
VEWAE10	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.

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 Insurance Datamarts for Health. This relates to the following files:

- **SYS_PARAMETERS.dat:**
File with control variables for the Oracle Insurance Datamarts for Health 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_REGIODE   VARCHAR2 (10)
DHV_REGIO_OMSCHRIJVING VARCHAR2 (100)
WGR_REGIODE   VARCHAR2 (10)
WGR_REGIO_OMSCHRIJVING VARCHAR2 (100)
OAD_CODE     VARCHAR2 (10)
ZIP_CODE     VARCHAR2 (10)
WZV_REGIODE   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)
```

- **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     CHAR (60) ,
EIGENSCHAP        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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 Intelligence 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
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 2012.02.0.0000

INSTALLATION OF ORACLE INSURANCE DATAMARTS FOR HEALTH REPOSITORY (RPD)

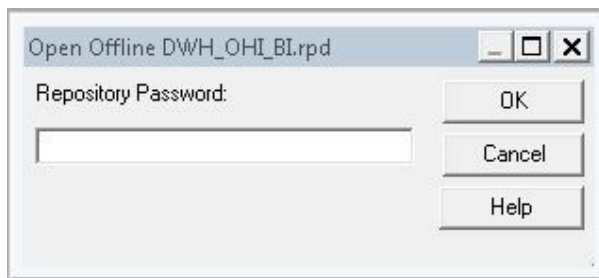
Oracle supplies an OBI EE repository as part of OHI Business Intelligence. This repository makes the OHI Business Intelligence database accessible. The repository can be installed on the Oracle BI Server.

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.

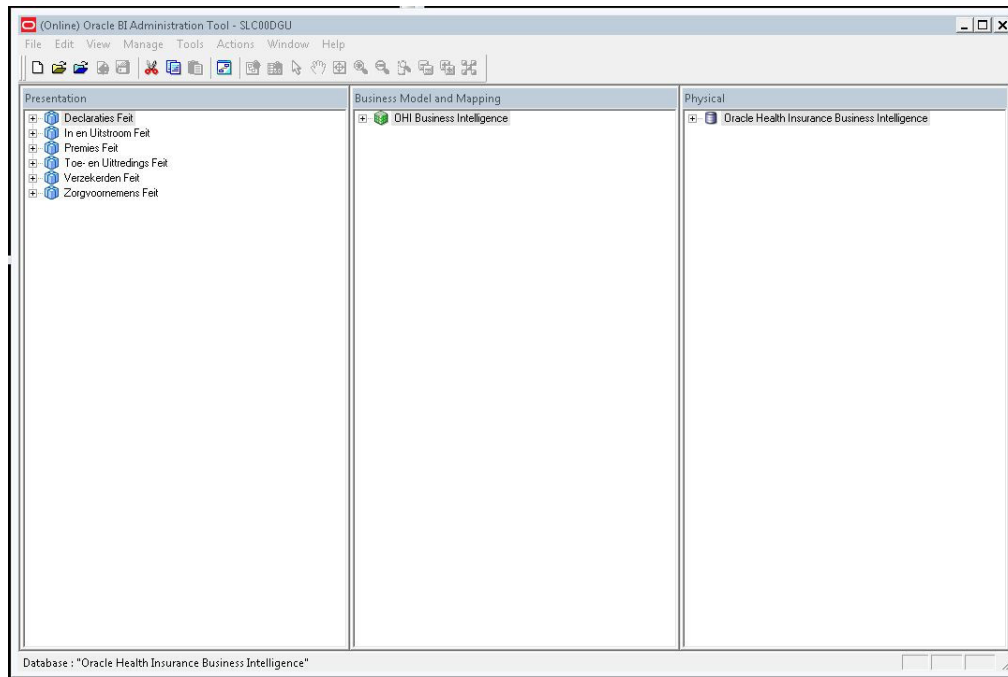
Before being able to use the DWH_OHI_BI.rpd file in your OBIEE environment, you will have to configure 2 connection pools using the Oracle BI Administration Tool.

Please follow the steps below to configure the 2 connection pools.

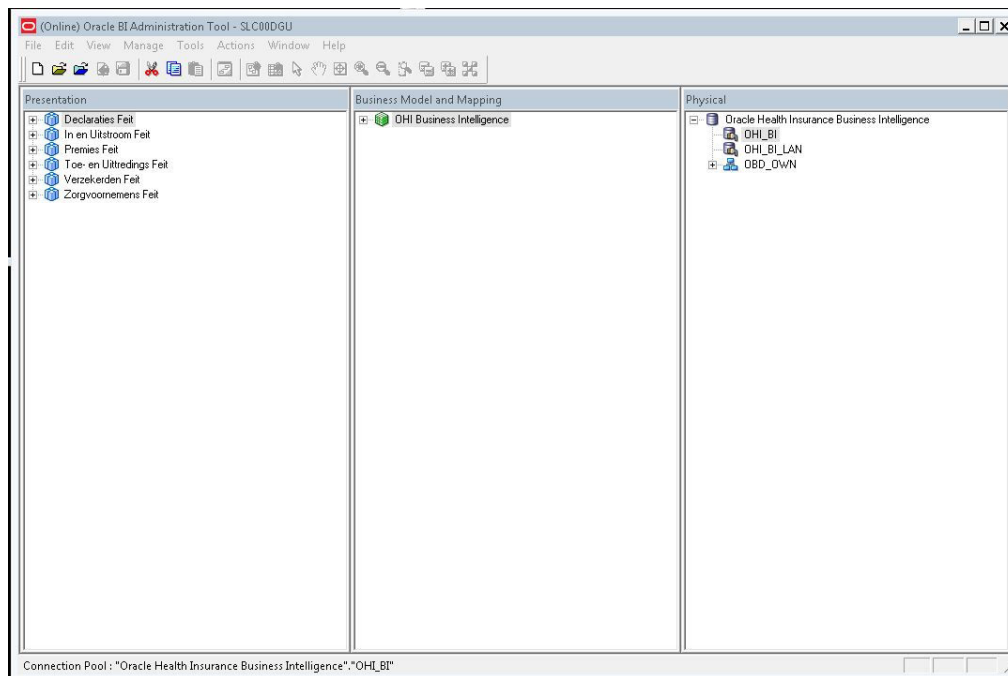
Open the DWH_OHI_BI.rpd offline using the Oracle BI Administration Tool:



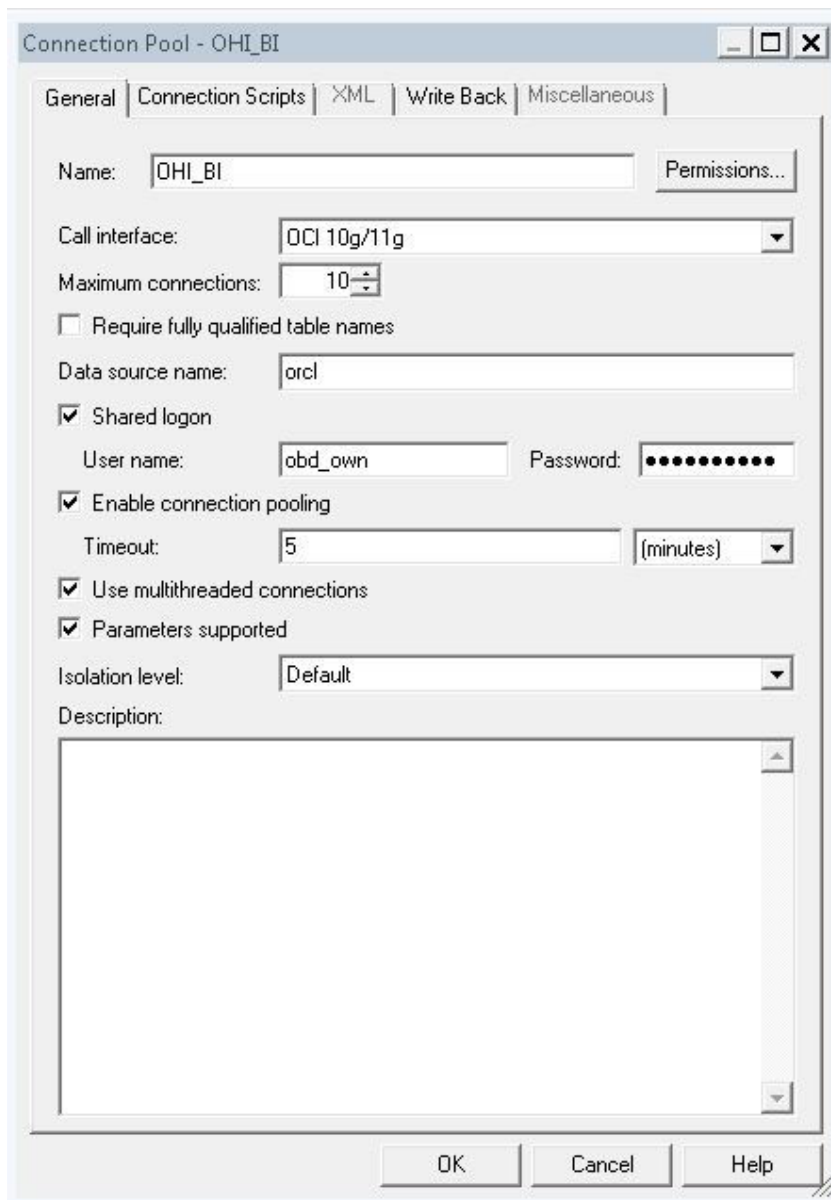
Enter adm1n1strator as the initial password.



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



Open the OHI_BI connection pool by double clicking OHI_BI.

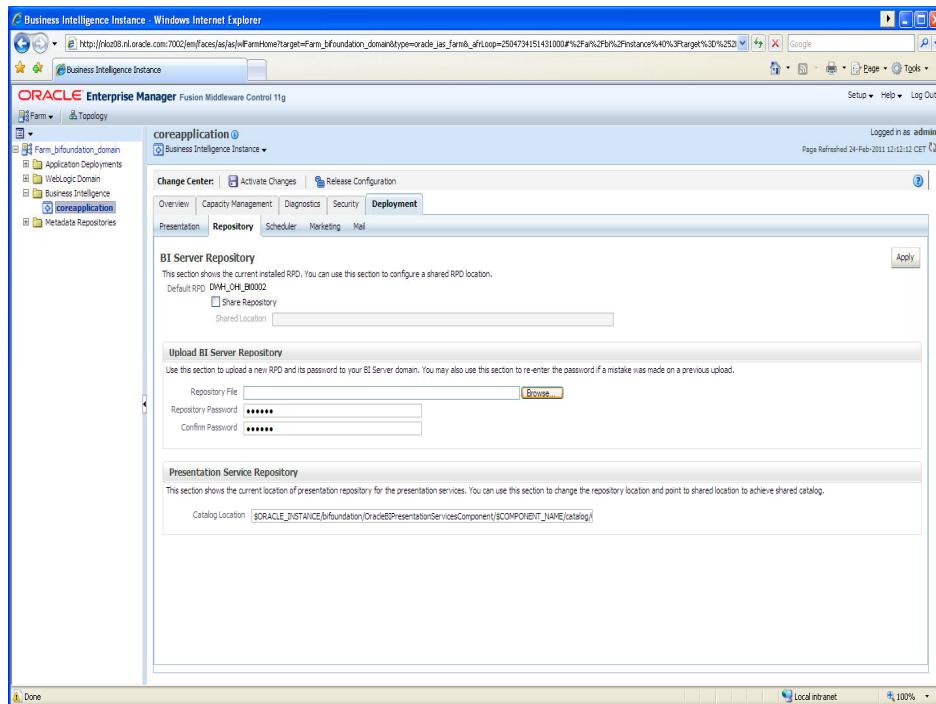


Change the data source name to your OHI Business Intelligence database.

Execute the above steps for the OHI_BI_LAN connection pool.

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.



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.

PART II LOADING

INTRODUCTION

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

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

EXTRACTING 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 Insurance Datamarts for Health.

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 Insurance Datamarts for Health → Extract OHI Back Office.

Submit Batch Request

Script:

Batch Request

Number: Start Time:

Printer: Interval:

Processes: Optimizer Mode:

No. of Hours: Period terminable: ☐ Trace terminable?: ☐

Trace?: ☐ Debug?: ☐

Parameters

<input checked="" type="checkbox"/>	OHI BI database	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Claims to date	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Premiums to date	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Policies to date	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Care author. to date	<input type="text"/>	<input type="text"/>	<input type="text"/>

Dependencies

	Depending on	Start Time
<input checked="" type="checkbox"/>	Func <input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

The parameters should be filled as follows.

- Fill in 'OHI BI database' with the alias of the Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health environment (e.g. oacc_batch). This db_connect_string should be specified as 'OHI BI database' parameter when submitting the ZRG0E01S.

- 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 system date -1 day is used (default).
- 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 '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?' and 'Care authorizations?' 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 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 'Y' [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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 'Y' [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.

TRANSFORMING 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 Insurance Datamarts for Health → Transform data.

The screenshot shows the 'Submit Batch Request' window. The 'Script' section has 'ZRGOS01S' and 'Transform data'. The 'Batch Request' section includes fields for Number (1020573), Printer, Processes, No. of Hours, Start Time, Interval, Optimizer Mode (set to ALL_ROWS), Period terminable, Trace?, Trace terminable?, and Debug?. The 'Parameters' section is a table with columns for database, name, and value. The 'Dependencies' section is a table with columns for function, name, and 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 Insurance Datamarts for Health 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**.

LOADING ORACLE INSURANCE DATAMARTS FOR HEALTH (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 Insurance Datamarts for Health → Load Oracle Insurance Datamarts for Health.

The screenshot shows the 'Submit Batch Request' dialog box. The 'Script' field is set to 'ZRGOD01S Load OHI Business Intelligence'. The 'Batch Request' section includes fields for 'Number' (1020574), 'Start Time', 'Interval', 'Optimizer Mode' (set to 'ALL_ROWS'), 'Period terminable', 'Trace?', 'Trace terminable?', and 'Debug?'. The 'Parameters' section has a table with columns for 'OHI BI database', 'Suppress errors?', and a value of 'N'. The 'Dependencies' section has a table with columns for 'Depending on', 'Start Time', and a value of 'Func'.

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.

UNDOING LOAD RUNS (ZRGO004S)

If, after refreshing Oracle Insurance Datamarts for Health, 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 Insurance Datamarts for Health 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

Submit Batch Request

Script:

Batch Request

Number: Start Time:

Printer:

Processes: Interval:

No. of Hours: Optimizer Mode:

Period terminable: ☐ Trace terminable?: ☐

Trace?: ☐ Debug?: ☐

Parameters

<input checked="" type="checkbox"/>	OHI BI database	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Claims?	N	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Premiums?	N	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Policies?	N	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Care authorizations?	N	<input type="text"/>	<input type="text"/>	<input type="text"/>

Dependencies

Depending on	Start Time
<input checked="" type="checkbox"/> Func	<input type="text"/>
<input type="checkbox"/>	<input type="text"/>
<input type="checkbox"/>	<input type="text"/>

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?
Database		Name of the Oracle Insurance Datamarts for Health database	Y
Owner	Default OBD_OW	OBD schema owner	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
Date	Default empty	Load runs on and after this date are removed. Empty = only remove the most recent load run.	Y

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 Insurance Datamarts for Healths 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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

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 Insurance Datamarts for Health database.

AUDIT INFORMATION

In Oracle Insurance Datamarts for Health, 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 Insurance Datamarts for Health 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 like this:

Maintain Batch Request

Batch Requests Parameters Dependencies

Batch Request Parameter 1 Value
☐ All Batch Re... Request Parameter 2 Value

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

Description Details

VERIFYING THE LOAD

An important task of the Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health.

FUNCTIONAL MANAGEMENT

Logging and validation

Validation of the Oracle Insurance Datamarts for Health 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
laadrn_id	Unique generated key
release_nr	Oracle Insurance Datamarts for Health release number
sav_id_ext	ID of script request ZRG0E01S.
sav_id_stg	ID of script request ZRG0S01S.
sav_id_dwh	ID of script request ZRG0D01S.
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 Insurance Datamarts for Health 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

These views are also available in the OBI EE repository.

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 Insurance Datamarts for Health.

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_accord	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 Insurance Datamarts for Health

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 Insurance Datamarts for Health

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 Insurance Datamarts for Health.

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 Insurance Datamarts for Health
aantal_bo	Number of memberships in OHI Back Office

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
bedrag_bi	Number or amount of the care authorization in Oracle Insurance Datamarts for Health

These views are also available in the Discoverer End User Layer.

Logging load scripts

The output of the scripts started through the OHI Back Office application are 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 Insurance Datamarts for Health. This is particularly convenient from a security standpoint.

This account must be created in the Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health 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
Voorschnijver	DWH_DECLARATIES	ZRE_REL_NR_VOORSCHRIJVER

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

In order to load these claim properties into the generic structure, they mapping to the generic structure must be added into the file GEN_EIGENSCHAPPEN.dat. The claim properties should 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

In this case the GEN_EIGENSCHAP.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_04

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 where 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 defined 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 (DWH_EWE_GENERIEK), or into the fact (DWH_DECLARATIES). When a claim property does not have multiple values, it is recommended to save it in a 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:

GEBRUIKSGROEP	EIGENSCHAP	TABEL_SPECIFIEK	KOLOM_SPECIFIEK	TABEL_GENERIEK	KOLOM_GENERIEK
TANDHEELKUNDE	IND_TAND			DWH_DECLARATIES	CHAR_EIGENSCHAP_02

The complete file will now look as follows:

GEBRUIKSGROEP	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_DECLARATIES	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 exist 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.

Example 3:

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:

GEBRUIKSGROEP	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 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_VIEW. This procedure must be executed under the OBD_OW 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_view('TANDHEELKUNDE');
```

This view 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.

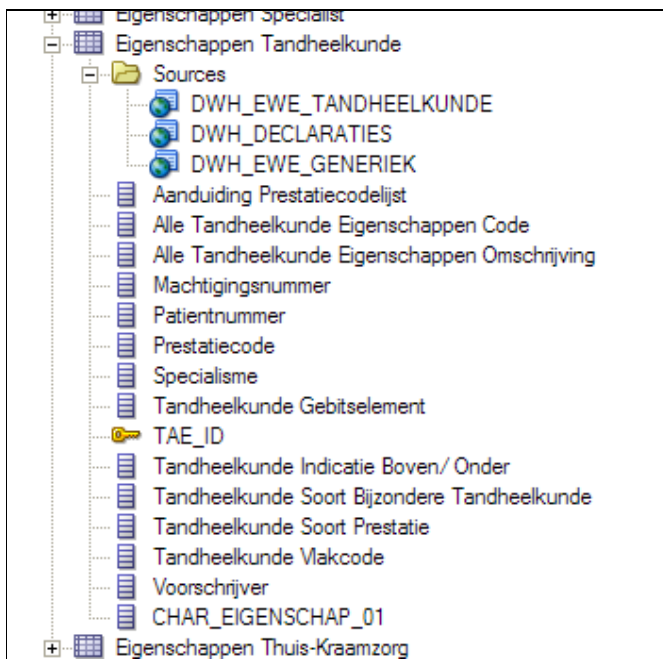
All generic claim property attributes are already added to the physical layer of the OBI EE repository.

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. Add the generic column to the business model


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:



2. Rename the column to the functional meaning
Change the name of CHAR_EIGENSCHAP_01 to 'Example 123' in the logical table


3. Add the associated Claim Procedure Group
Due to the fact that generic claim property attributes can have a different 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

```

A

```

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

```

A

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 Insurance Datamarts for Health 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 Insurance Datamarts for Health

When new releases of Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health 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_AFGWEZEN_DECLARATIES
- DWH_DECLARATIES
- DWH_DECLARATIES_AGG
- DWH_DEELNAME_MUTATIES
- 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 Insurance Datamarts for Health 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 Insurance Datamarts for Health, however the following submenu choices are only shown for Oracle Insurance Datamarts for Health.

```
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 Insurance Datamarts for Health version 2010.01, use is made of OWB11g. As a consequence, as well as the Oracle Insurance Datamarts for Health 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 Insurance Datamarts for Health 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 INSURANCE DATAMARTS FOR HEALTH ENVIRONMENTS

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

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

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.