ORACLE
INSURANCE

Oracle Health Insurance Data Marts

Administrator Reference

version 9.50

Part number: F27899-01

February 2020

U.S. GOVERNMENT RIGHTS

CHANGE HISTORY

Release	Version	Changes
10.12.2.0.0	9.29	• 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.
10.12.3.0.0	9.30	• Added new fact table DWH_ONDERHANDEN_WERK (Work in Progress Claims) • Replaced generic role OZG_ROL_SELECT by secure role OBD_SELECT_ROLE.
10.13.1.0.0	9.31	• Added Exadata Support, Migrating OWB from single instance to multi node RAC, added OBI_SELECT_USER • Added new batch Correct monitorcode (ZRGO005S)
10.13.2.0.0	9.32	• Added description for generic dimension properties (DIM_EIGENSCHAPPEN.dat) • Added new parameter P_MAX_DCE_VERSCHIL; a threshold that is taken into account when performing crosschecks for claims. • Added details for the Members check (three differente reference dates) • Added new description for setting ‘Policies to date’ • Corrected examples for dynamic claim properties • Added Default Secure Install and Enable Security Audit
10.13.2.1.0	9.33	• Added new fact table DWH_OPENSTAANDE_VORDERINGEN (Open Receivables) • Added new batch Load Open Receivables (ZRGO006S)
10.13.3.0.0	9.34	• Updated batch ZRGO005S for new purpose (Update external codings for claims)
10.13.3.3.0	9.34.1	• Added new fact table DWH_DECL_PRESTATIE_DETAILS (Claim Details)
10.14.1.0.0	9.35	• Added new fact tables DWH_OPENSTAANDE_VERPLICHTINGEN (Open Payables), DWH_VERPLICHTINGEN (Payables) and DWH_VORDERINGEN (Receivables)
10.14.1.3.0	9.36	• Added new batch Load Open Policies (ZRGO020S)
10.15.1.0.0	9.37	• Moved documentation of ZRGOE01S, ZRGOS01S, ZRGOD01S, ZRGO003S, ZRGO004S, ZRGO005S, ZRGO006S, ZRGO009S, ZRGO020S, ZRGO021S and ZRGO022S to OLH of OHI Back Office.
10.15.1.1.0	9.38	• Changed paragraph ’SETTING UP ORACLE HEALTH INSURANCE DATA MARTS ENVIRONMENT’ due to migration to RDBMS 12.1.0.2 (theme M-4064).
10.15.3.0.0	9.39	• Added a note in the Oracle Warehouse Builder Software section about OBDXMLPS.sql and root.sh
10.16.1.0.0	9.40	• Added Appendix H: Virtual Private Database
10.16.2.0.0	9.41	• Changed version number, no changes.
10.17.1.0.0	9.41	• No changes.
10.17.2.0.0	9.42	• Updated Appendix D: OWB 11gR2 post-cloning process for OHI Data Marts. Reduced the number of manual steps involved in performing the OWB post-cloning process
10.18.1.0.0	9.43	• Removed obsolete OWB references and removed the following sections: • Installation of Oracle Warehouse Builder (OWB) Repository • Cleaning up OWB Audit data • OWB Runtime service • Appendix B: Use of WBX_LAADRUN_ERRORS_VW and WBX_MAPPING_VW views • Appendix D: OWB 11gR2 post-cloning process for OHI Data Marts • Appendix E: Migrating OWB from single instance to multi node RAC
10.18.1.3.0	9.44	• Updated ‘Configuration of generic structure for claim properties’ to include information on how to make generic claim propertyes available within the OBIEE repository.
10.18.2.0.0	9.45	• Changed part number on title page.
10.18.2.2.0	9.46	• M-5568: Added the WBX_CTR_DRIEHOEKSCONTROLE_VW view
10.19.1.0.0	9.47	• M-5668: Database Vault implementatie voor Data Marts. Added capital to describe the Database Vault installation and setup.
10.19.1.0.0	9.48	• Changed part number on title page.
10.19.2.0.0	9.49	• Changed part number on title page
10.20.1.0.0	9.50	• Changed part number on title page
		•

CONTENTS

INTRODUCTION

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

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

With this document the Oracle Health Insurance Data Marts administrator should be able to install and maintain the Data Warehouse.

IMPORTANT DOCUMENTATION

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

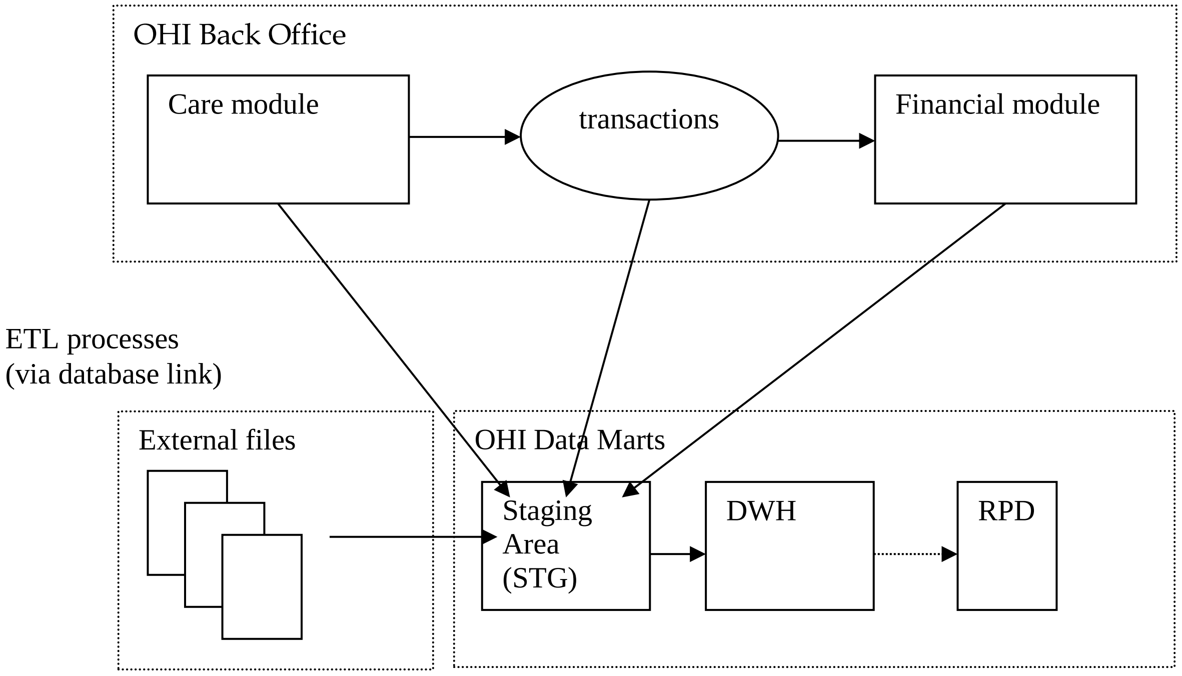
1. Standard product manuals from Oracle DBMS and Oracle Business Intelligence Enterprise Edition (OBI EE). Documentation on these Oracle products can be found on the Oracle support website (http://support.oracle.com)
2. CTA13508.doc: Oracle Health Insurance Installation, Configuration and DBA Manual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

SOFTWARE USED

Oracle Health Insurance Certification on MOS (logon to MOS, choose tab ‘Certifications’, specify product name ‘ Oracle Health Insurance Data Marts’ and the relevant versions)

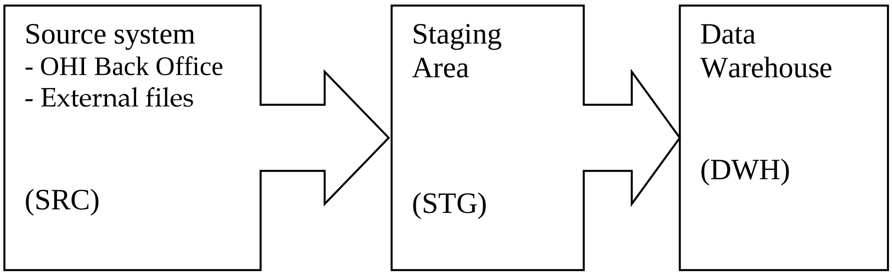
OVERVIEW

The architecture of Oracle Health Insurance Data Marts broadly looks like this:



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

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



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

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

PART I - INSTALLATION

INTRODUCTION

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

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

SETTING UP ORACLE HEALTH INSURANCE DATA MARTS ENVIRONMENT

OS set-up

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

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

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

Software installation

Database

Oracle Health Insurance Data Marts must be installed in a dedicated database. It is not permitted to install Oracle Health Insurance Data Marts in the same database as OHI Back Office, as Oracle Health Insurance Data Marts 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

As of database release Oracle 12cR2 OWB is no longer supported. Accordingly, from Oracle Health Insurance release 10.18.1.0 onwards an OWB Runtime environment is no necessary to execute the Oracle Health Insurance Data Marts ETL process.

Creation of an Oracle Health Insurance Data Marts database

A pluggable database needs to be created first before Oracle Health Insurance Data Marts can be installed.

Parameters

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

```
DB_BLOCK_SIZE                  = 16384 or DB_BLOCK_SIZE=8192
NLS_SORT                       = BINARY          # For performance reasons
OPTIMIZER_MODE                 = ALL_ROWS        # For performance reasons
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 parameters are influenced by the environment variables as used by the process of the connecting session. When they are not set or specified the database instance values apply. Below example values are shown for a Dutch environment.

```
NLS_LANGUAGE                  = DUTCH             # Choose the appropriate language
NLS_NUMERIC_CHARACTERS        = ",." ***          # personal choice
NLS_TERRITORY                 = ”THE NETHERLANDS”

*** = this setting cannot be changed once the application is used
```

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.* See paragraph 8.4.2 of CTA13508.doc: Oracle Health Insurance Installation, Configuration and DBA Manual.

Tablespaces

The tablespaces below must be created for data and indexes:

Tablespace
STG_STAD
STG_STAI
STG_DYND
STG_DYNI

Tablespace
DWH_STAD
DWH_STAI
DWH_DYND
DWH_DYNI
OZG_DIM_SYS_TAB
OZG_DIM_SYS_IND
OZG_FACT_SYS_TAB
OZG_FACT_SYS_IND
OZG_LOG_TAB
OZG_LOG_IND

The created tablespaces must comply with the following requirements:

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

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

Database users

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

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

☞ **Note:** From Oracle database release 12cR2 & Oracle Health Insurance release 10.18.1.0 onwards, the OWBSYS and OHI_BI_WS_OWN users are no longer required.

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

User	ID	Description

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

Database link

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

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

☞ **Note:** After a database has been cloned, create or change the private database link SRC_OPENZORG (as OBD_OWN) to the correct Back Office source. The account OBD_SELECT_USER should be used as the CONNECT TO user:

```
create database link SRC_OPENZORG
connect to OBD_SELECT_USER
using '<OHI Back Office Service Name>';
```

Database directories

For several external tables, the following database directories need to be created under the OBD_OWN schema:

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

Example:

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

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

☞ **Note:** After a database has been cloned, correct the OBD_INPUT directory object in the database (login SYS as SYSDBA).

```
SQL> drop directory obd_input;
Directory dropped.
SQL> create directory obd_input as
'/ozg/app/oracle/product/Zorg/oacc';
Directory created.
SQL> grant read, write on directory obd_input to public;
Grant succeeded.
```

Default Secure Install

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

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

1. Connect to SQLplus as dba user and change the **OBD_OWN** password using:
   SQL> alter user **OBD_OWN** identified by <PASSWORD>;
2. Log on to the application server
   ssh batch@<hostname>
3. Set the environment:
   . ozg_init.env DB1202

4. Navigate to your wallet
   cd network/admin/special
5. To list all wallet entries:
   mkstore -wrl . -listCredential
6. Modify the credentials:
   mkstore -wrl . -modifyCredential <ENVIRONMENT_NAME>_INSTALL OBD_OWN <PASSWORD>
7. Check the connection string with:
   sqlplus /@<ENVIRONMENT_NAME>_INSTALL

Create a Default Password Policy

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

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

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

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

All OHI Data Marts accounts should use this profile.

Enable Security Audit

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

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

General instructions

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

Oracle Health Insurance Installation, Configuration and DBA Manual

(chapter 8)

GENERATION AND INSTALLATION OF ORACLE HEALTH INSURANCE DATA MARTS OBJECTS

Installation

Release

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

ORACLE HEALTH INSURANCE Installation of releases

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

System parameters

In OHI Back Office a window is supplied 'Data Marts Parameter Values' (ZRGO032F). All Data Mart parameter values can be adjusted here.

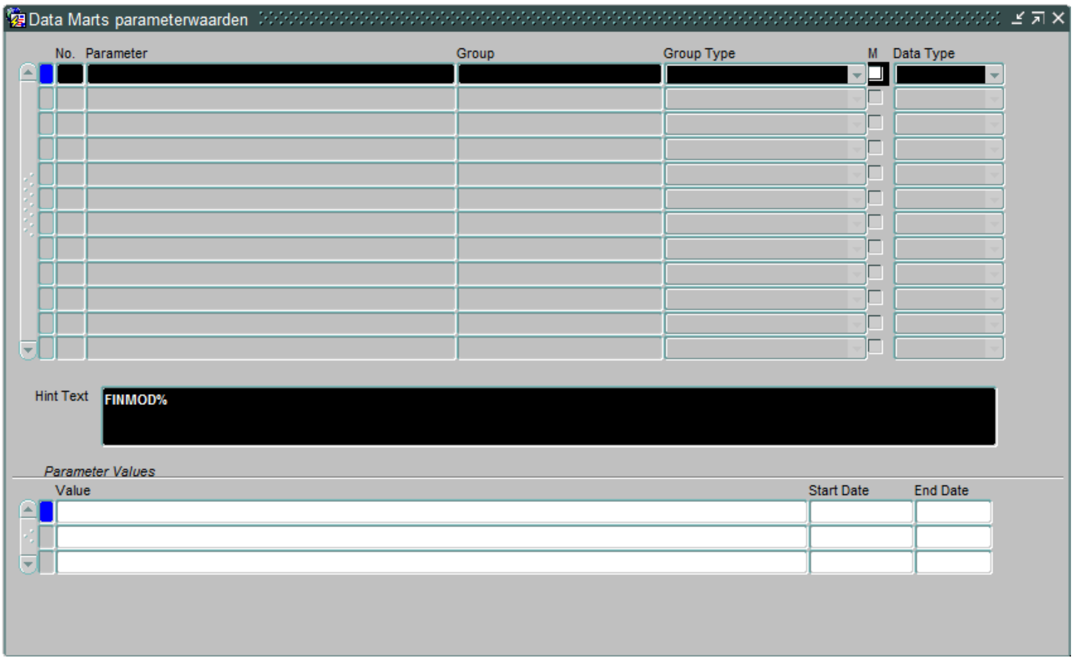
These parameters are customer-specific and the value of a number of these parameters **must** be adjusted here **before** Oracle Health Insurance Data Marts can be loaded. A further number may also be adjusted if desired.

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

Parameter	Description
FINMOD	Indication (J/N [Yes/No]) of whether the financial module of OHI Back Office is used to load financial transactions. The default value is J [meaning yes], which means that Oracle Health Insurance Data Marts assumes that the financial data can be retrieved from OHI Back Office.

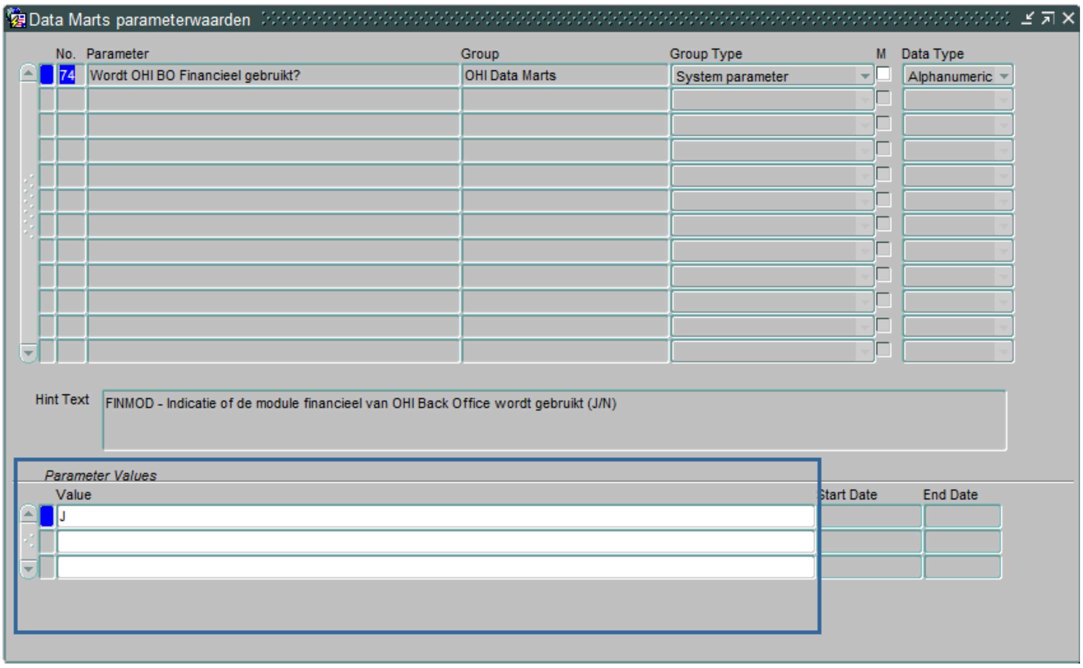
Parameter	Description
EOZNLS	The value that is given for this parameter depends on the NLS language setting. This determines, among other things, how a number is displayed. We distinguish between two different styles of notation. The following values indicate each style for the EOZNLS parameter. 1) Decimals separated by a comma Thousands separated by a period e.g.: 1.000.000,001 This style is used in the DUTCH NLS Language, among others. Value for EOZNLS: nls_numeric_characters = ',.' This is the default value for the parameter. 2) Thousands separated by a comma Decimals separated by a period e.g.: 1,000,000.001 This style is used in the AMERICAN NLS Language, among others. Value for EOZNLS: nls_numeric_characters = '.,'

Query for the parameter (e.q. FINMOD):



Hint Text FINMOD%

And change the value of the parameter:



Data Marts parameterwaarden

No.	Parameter	Group	Group Type	M	Data Type
74	Wordt OHI BO Financieel gebruikt?	OHI Data Marts	System parameter	□	Alphanumeric

Hint Text: FINMOD - Indicatie of de module financieel van OHI Back Office wordt gebruikt (J/N)

Parameter Values

Value	Start Date	End Date
J		

There are a few 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 should 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.

The other parameters concern descriptions of unknown values, for example, that **can** be adjusted if desired. In case of a non-Dutch environment it is advisable to change the value for the parameter ‘LOGLNG’ from the value ‘NL’ to ‘EN’, this will ensure that information displayed in logging files will be in English.

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

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

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

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

External files

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

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

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

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

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

- WBX_OBIEE_TRANSLATIONS.csv
  File with translations for Oracle Data Marts

```
METADATA_OBJECT            VARCHAR2(4000)
SESSION_VARIABLE (PK)      VARCHAR2(4000)
NEDERLANDS                 VARCHAR2(4000)
ENGELS                     VARCHAR2(4000)
SPAANS                     VARCHAR2(4000)
```

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

The file LEEFTIJD_CATEGORIEEN.dat must be filled for correct operation of the Oracle Health Insurance Data Marts load run. The file POSTCODE_VERRIJKINGEN.dat may be left empty.

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

The content of these files can be changed except for 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 Data Marts. 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 Data Marts set the AllowedLanguages parameter in instanceconfig.xml (e.g. <AllowedLanguages>en,nl</AllowedLanguages> as per documentation *Oracle Fusion Middleware System Administrator’s Guide for Oracle Business Intellgence Enterprise Edition, Localizing Oracle Business Intelligence*). When no language has been chosen yet, the language defaults to the browser language. To adjust the language as per login, modify the url (eg http://Server_Name:port_number/analytics/saw.dll?Dashboard&Lang=nl). Once new translations are made available by modifying the WBX_OBIEE_TRANSLATIONS.csv file, the Oracle Business Intelligence server needs to be restarted.

File	Table
LEEFTIJD_CATEGORIEEN.dat	STG_LEEFTIJD_CATEGORIEEN_EXT
POSTCODE_VERRIJKINGEN.dat	STG_POSTCODE_VERRIJKINGEN_EXT

SET-UP OF ORACLE BUSINESS INTELLIGENCE ENTERPRISE EDITION ENVIRONMENT

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

Software installation

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

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

For the right versions of the software, see:

Certifications matrix 10.13.1.0

Oracle Business Intelligence Customizations

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

Creating a NON- PRIVILEGED user: OBI_SELECT_USER

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

This is done in SQLPlus, instructions as per below.

Log on to the OHIBI database and create the OBI_SELECT_USER

```
SQLPlus> create user obi_select_user identified by <password>;
SQLPlus> grant obd_rol_select to obi_select_user;
SQLPlus> grant create session to obi_select_user;
SQLPlus> grant read on directory obd_input to obi_select_user;
```

When VPD policies are enabled privacy, sensitive data will be masked (see `VPD Policies` for which tables/columns data will be maked).

Installation of Oracle Health Insurance Data Marts Repository (RPD)

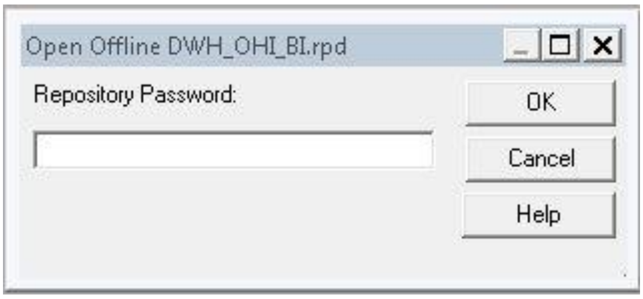
Oracle supplies an OBI EE repository as part of OHI Data Marts. This repository makes the OHI Data Marts database accessible. The repository can be installed on the Oracle BI Server.

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

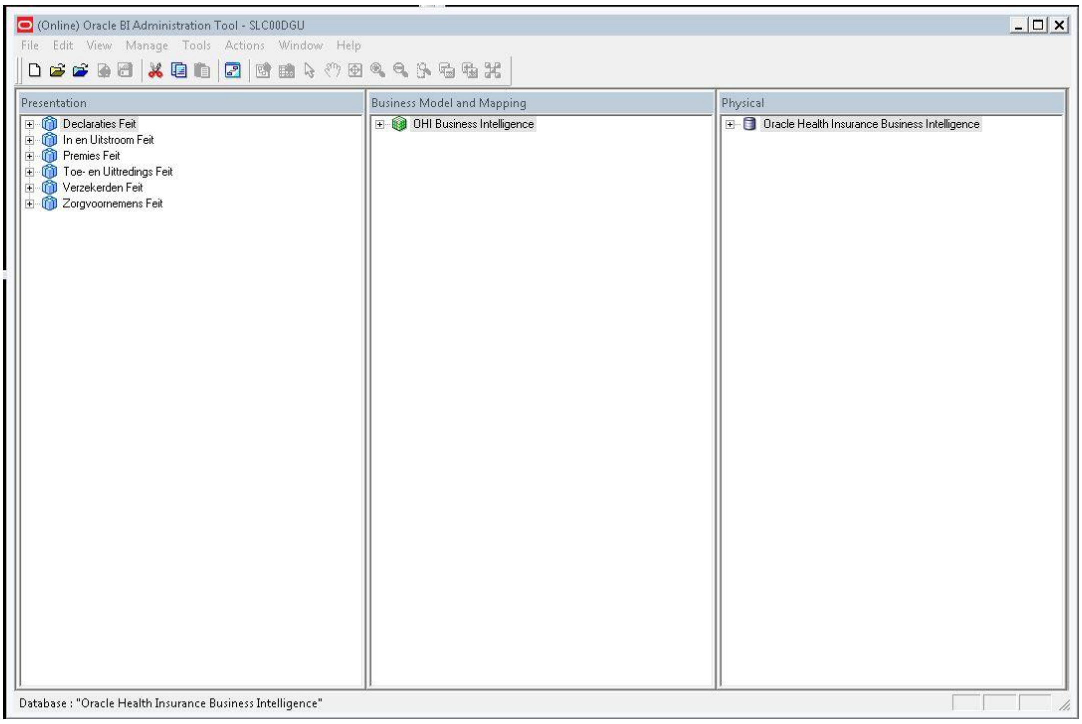
Before being able to use the DWH_OHI_BI.rpd file in your OBIEE environment, you will have to configure 3 connection pools using the Oracle BI Administration Tool.

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

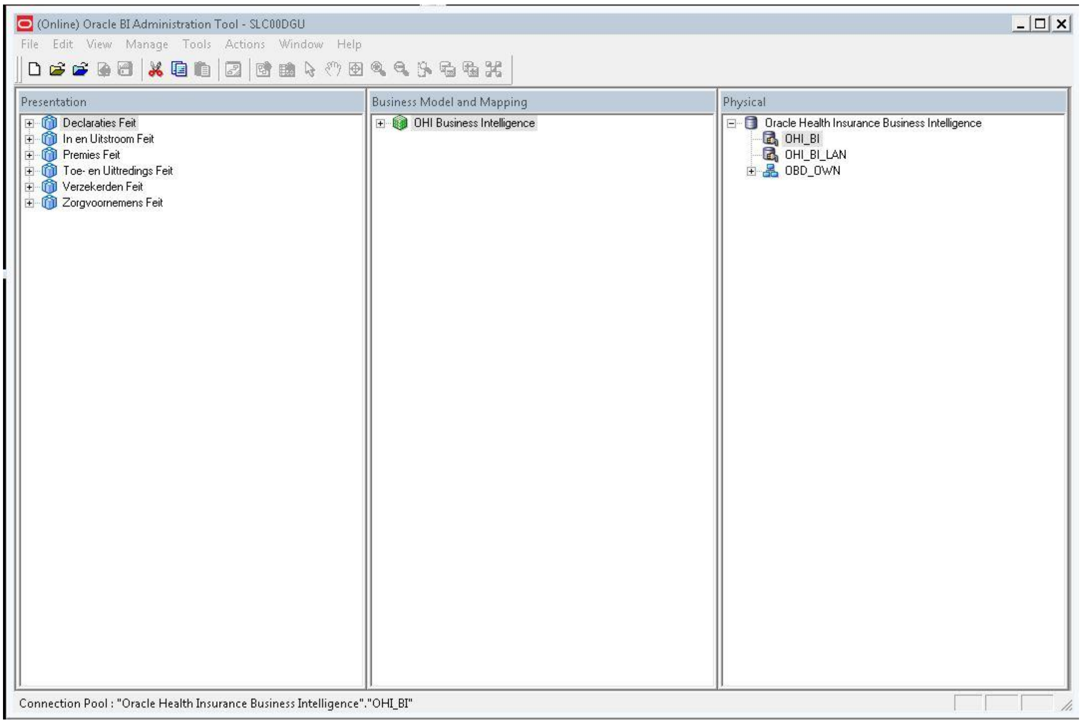
Open the DWH_OHI_BI.rpd offline using the Oracle BI Adminsitration Tool:



Enter adm1n1strator as the initial password.

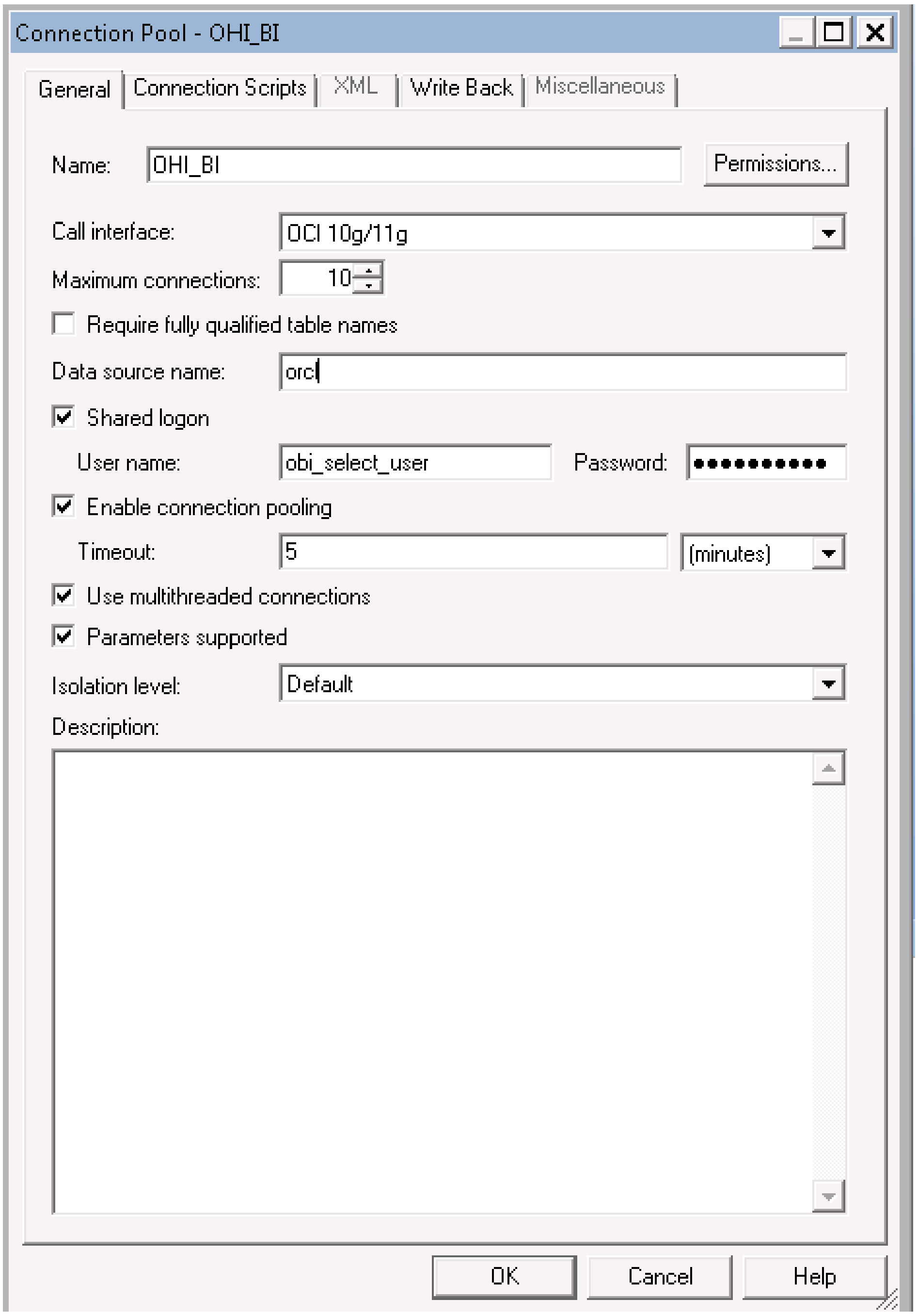

(Online) Oracle BI Administration Tool - SLC00DGU
File Edit View Manage Tools Actions Window Help
Presentation
Declaraties Feit
In en Uitstroom Feit
Premies Feit
Toe- en Uittredings Feit
Verzekerden Feit
Zorgvoornemens Feit
Business Model and Mapping
OHI Business Intelligence
Physical
Oracle Health Insurance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

Open the Oracle Health Insurance Data Marts node in the Physical layer.


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

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

Open the OHI_BI connection pool by double clicking OHI_BI.



Name: OHI_BI

Call interface: OCI 10g/11g

Maximum connections: 10

□ Require fully qualified table names

Data source name: orcl

☑ Shared logon

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

☑ Enable connection pooling

Timeout: 5 (minutes)

☑ Use multithreaded connections

☑ Parameters supported

Isolation level: Default

Description:

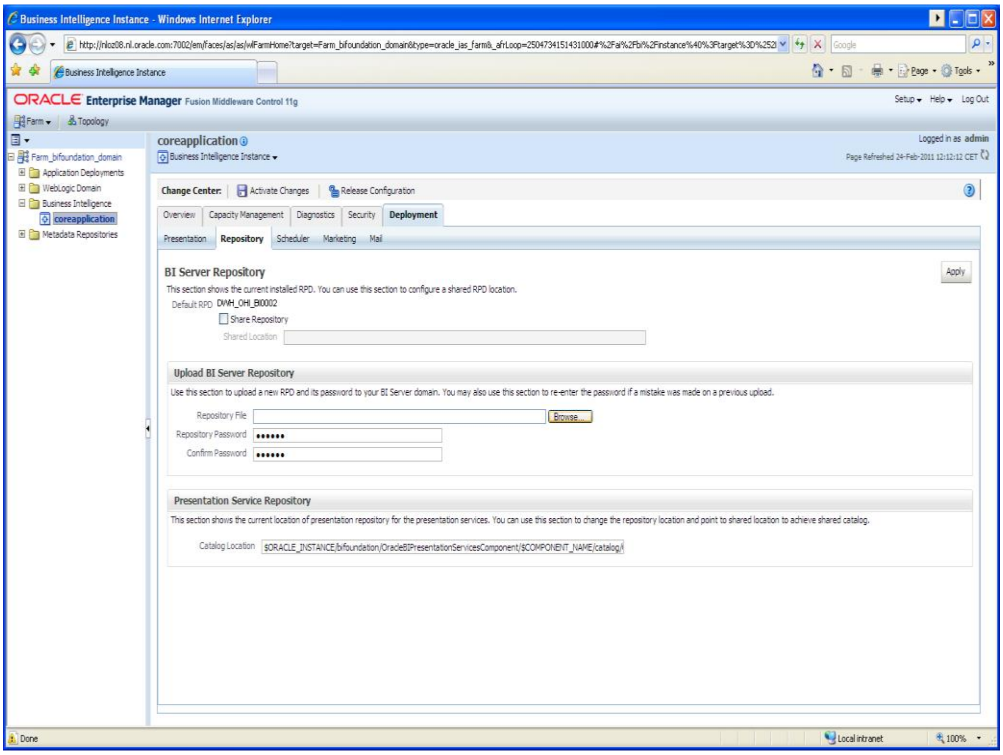
Change the data source name to your OHI Data Marts database.

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

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

Save the DWH_OHI_BI.rpd.

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


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

BI Server Repository

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

Default RPD DWH_OHI_BI0002

□ Share Repository

Shared Location

Upload BI Server Repository

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

Repository File

Repository Password ••••••

Confirm Password ••••••

Presentation Service Repository

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

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

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

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

Configure Usage Tracking

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

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

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

Upgrading Oracle Business Intelligence repository

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

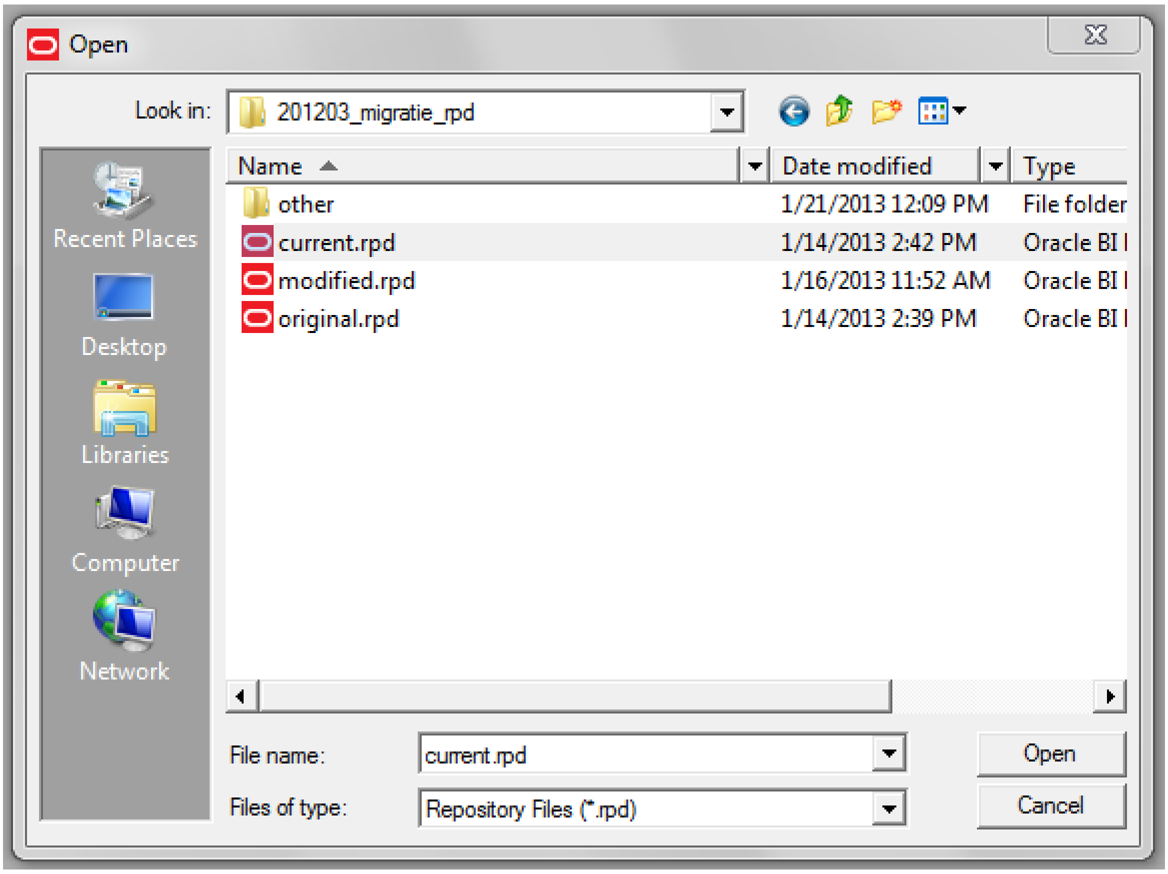
Definitions used in this topology:

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

In short, the above-mentioned procedure needs to be followed.

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

1. Open the current.rpd.

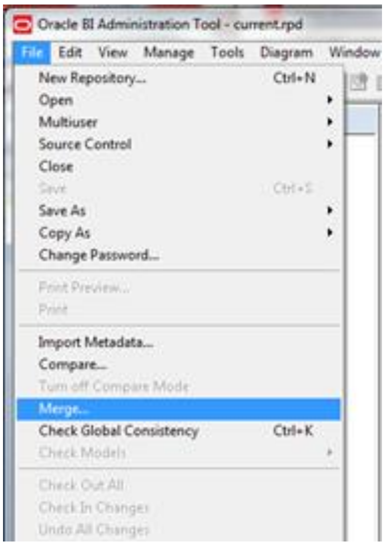


Name	Date modified	Type
other	1/21/2013 12:09 PM	File folder
current.rpd	1/14/2013 2:42 PM	Oracle BI I
modified.rpd	1/16/2013 11:52 AM	Oracle BI I
original.rpd	1/14/2013 2:39 PM	Oracle BI I

File name: current.rpd

Files of type: Repository Files (*.rpd)

2. File Merge

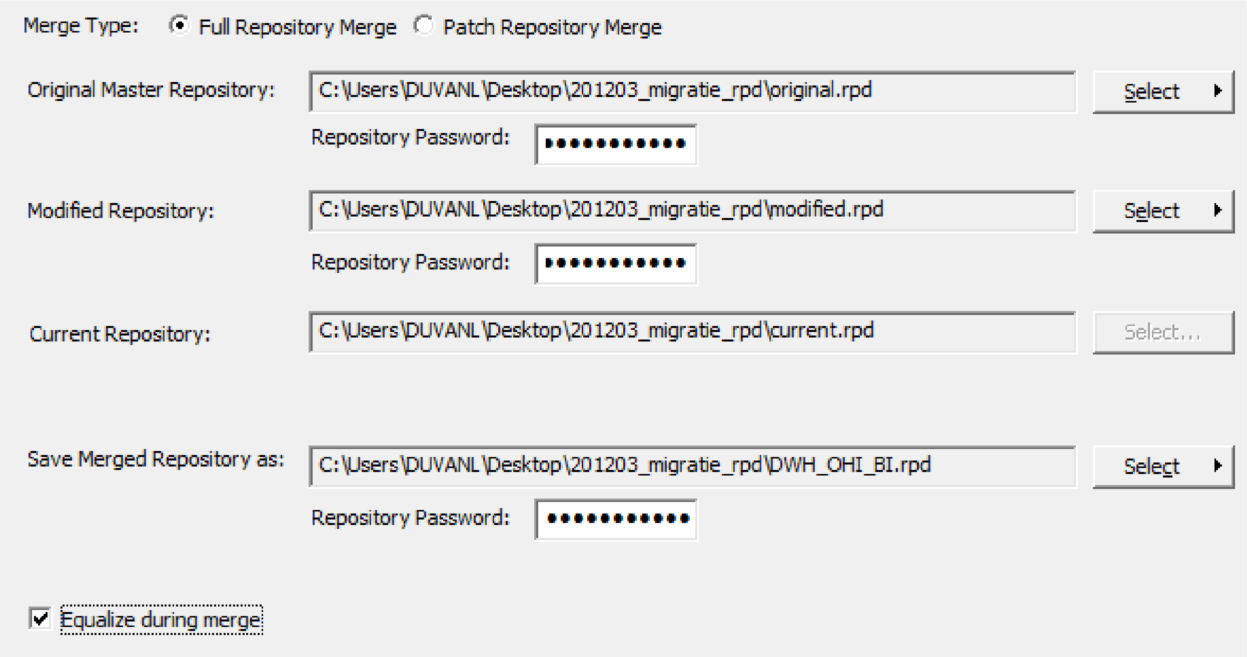


3. Tick equalize during merge

4. Open original and merge rpds

5. Type the passwords for original and modified repositories.

6. Save merged repository as: DWH_OHI_BI.rpd



Merge Type: ● Full Repository Merge ○ Patch Repository Merge

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

Repository Password: ●●●●●●●●●●●

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

Repository Password: ●●●●●●●●●●●

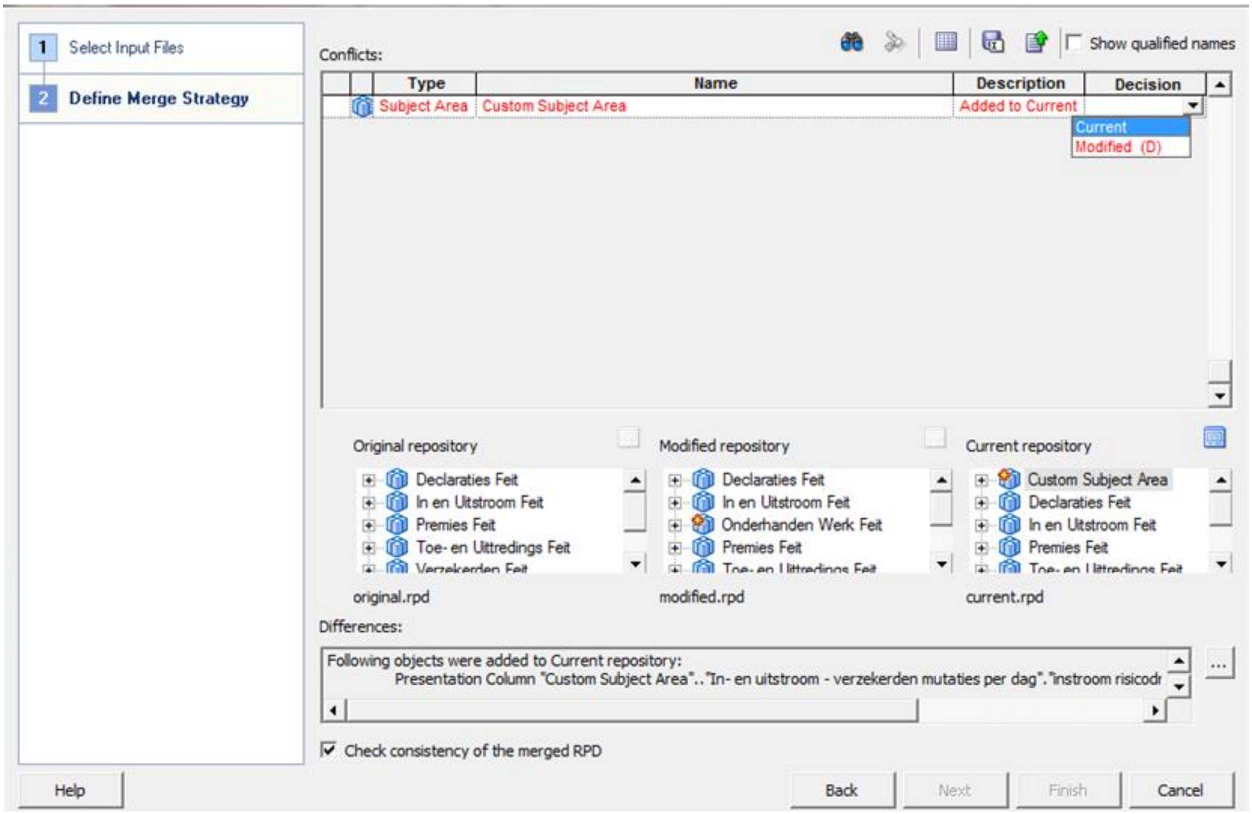
Current Repository: C:\Users\DUVANL\Desktop\201203_migratie_rpd\current.rpd

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

Repository Password: ●●●●●●●●●●●

☑ Equalize during merge

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



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

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

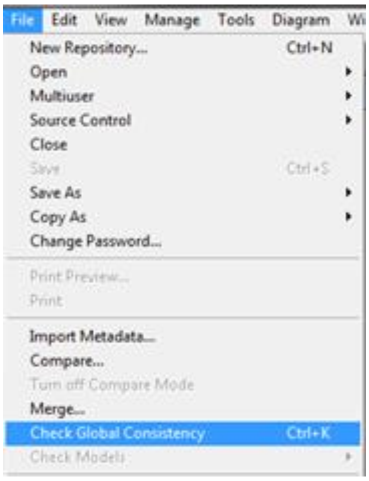
☑ Check consistency of the merged RPD

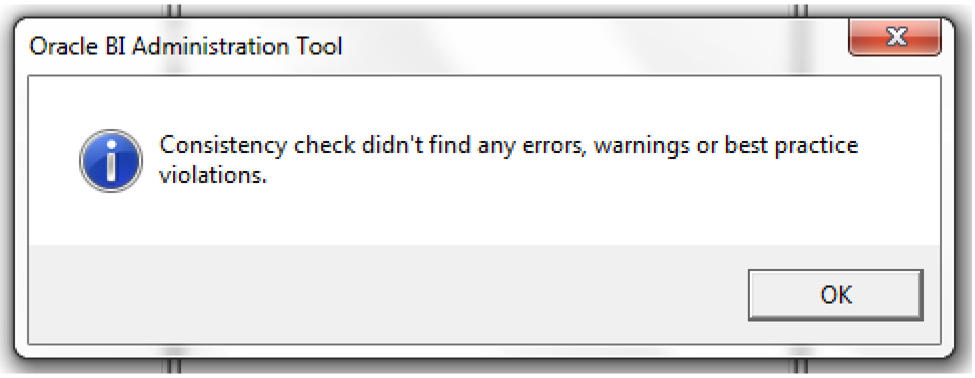
-- choose current to keep the changes in Current repository (Health Insurer RPD)

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

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

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




Oracle BI Administration Tool

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

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

PART II - APPLICATION MANAGEMENT

INTRODUCTION

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

FUNCTIONAL MANAGEMENT

Logging and validation

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

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

Logging of loading

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

WBX_LAADRUNS_VW

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

Column	Description
LAADRUN_ID	Unique generated key
RELEASE_NR	Oracle Health Insurance Data Marts release number
SAV_ID_EXT	ID of script request ZRGOE01S.
SAV_ID_STG	ID of script request ZRGOS01S.
SAV_ID_DWH	ID of script request ZRGOD01S.
DECLARATIES_GELADEN	Indication (J/N [meaning Y/N]) of whether the claims fact is loaded in this run.
VERZEKERDEN_GELADEN	Indication (J/N [meaning Y/N]) of whether the policies fact is loaded in this run.
PREMIES_GELADEN	Indication (J/N [meaning Y/N]) of whether the premiums fact is loaded in this run.
TOE_EN_UITTREDINGEN_GELADEN	Indication (J/N [meaning Y/N]) of whether the In- and Out-fluxs fact is loaded in this run.
ZORGVOORNEMENS_GELADEN	Indication (J/N [meaning Y/N]) of whether the care authorizations fact is loaded in this run.
ABONNEMENTSHONORARIUM_GELADEN	Indication (J/N [meaning Y/N]) of whether the per capita agreement fact is loaded in this run.
VERBINTENISSEN_GELADEN	Indication (J/N [meaning Y/N]) of whether the provider relationships fact is loaded in this run.
ONDERHANDEN_WERK_GELADEN	Indication (J/N [meaning Y/N]) of whether the work in progress fact is loaded in this run.
ZORG_INKOOP_GELADEN	Indication (J/N [meaning Y/N]) of whether the procurement agreement fact is loaded in this run.
FINANCIEEL_GELADEN	Indication (J/N [meaning Y/N]) of whether the financial 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.
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.

Column	Description
LAADPERIODE_ONDERHANDEN_WERK	Date used for this load as date to for selection of the source data for work in progress.
LAADPERIODE_FINANCIEEL	Date used for this load as date to for selection of the source data for financials.
CONTROLE_UITGEVOERD	Is the loaded data checked?
DOORLOOPTIJD_EXTRACTIE	Run time of the extraction phase
DOORLOOPTIJD_TRANFORMATIE	Run time of the transformation phase
DOORLOOPTIJD_LADEN	Run time of the load phase
DOORLOOPTIJD_TOTAAL	Total run time of the extraction + transformation + load phase
LAADRUN_ID	Unique generated key
RELEASE_NR	Oracle Health Insurance Data Marts release number

WBX_MAPPING_VW

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

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

WBX_LAADRUN_ERRORS_VW

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

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

WBX logging

Additional logging is recorded in the 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

Column	Comments
NUM_SELECTED	Number of selected rows
NUM_INSERTED	Number of inserted rows
NUM_UPDATED	Number of updated rows
NUM_DELETED	Number of deleted rows
NUM_ERRORS	Number of errors occurred
NUM_DISCARDED	Number of ignored rows
NUM_MERGED	Number of merged rows
MAP_SQLERR	SQL-error code (ORA-XXX)
MAP_SQLERRD	Time on which the error occurred
MAP_SQLERRM	SQL-error message

Results of the compatibility checks

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

WBX_CTR_DECL_DETAILS_VW

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

Column	Description
SAV_ID	The ID of the script request from the OHI Back Office batch scheduler
CTR_ID	Identification of the check result
CODE	Code of the check
OMSCHRIJVING	Description of the check
DCR_NR	Claim number
VOLGNR	Sequence number of the claim line
DATUM_ACCOORD	The date on which the claim line was approved
DER_ID	The OHI Back Office ID of this version of the claim line
FFL_ID	Reference to the financial transaction detail from which this general ledger entry originated
SFL_CODE	Financial transaction detail type
SOORT_BEDRAG	The amount type of the columns below

WBX_CTR_DECL_BOEKINGEN_VW

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

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
PAKKET	The product offered by the coverage
VEL_ID	Identification of the journal entry
FFL_ID	Reference to the financial transaction detail from which this general ledger entry originated
GRG_NR	General ledger account number
GBF_NR	General ledger unit number
RBK_NR	Category number
DATUM_IMPORT	Date imported of the liability
DATUM_ONTSTAAN	Creation date of the liability
SOORT_BEDRAG	The amount type of the columns below
BEDRAG_BO	Total amount of the claim in Oracle Back Office
BEDRAG_DM	Total amount of the claim in Oracle Health Insurance Data Marts

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

Column	Description
OMSCHRIJVING	Description of the check
DCR_NR	Claim number
VOLGNR	Sequence number of the claim line
DATUM_ACCOORD	The date on which the claim line was approved
PAKKET	The product offered by the coverage
VEL_ID	Identification of the journal entry
GRG_NR	General ledger account number
GBF_NR	General ledger unit number
RUK_NR	Category number
DATUM_IMPORT	Date imported of the liability
RISICODRAGER	The relationship number of the insurance company bearing the risk
BETAALMAAND	The month in which the payment took place
SOORT_BEDRAG	The amount type of the columns below
BEDRAG_BO	Total amount of the claim in Oracle Back Office
BEDRAG_BI	Total amount of the claim in Oracle Health Insurance Data Marts

WBX_CTR_DECL_PRESTATIE_DTLS_VW

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

Column	Description
SAV_ID	The ID of the script request from the OHI Back Office batch scheduler
CTR_ID	Identification of the check result
CODE	Code of the check
OMSCHRIJVING	Description of the check
DCR_NR	Claim number
VOLGNR	Sequence number of the claim line
DATUM_ACCOORD	The date on which the claim line was approved
SOORT	The number type of the columns below
AANTAL_BO	Total number of performed activities/time spent in Oracle Back Office
AANTAL_DM	Total number of performed activities/time spent in Oracle Health Insurance Data Marts

WBX_CTR_PREMIES_VW

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

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

WBX_CTR_VERZEKERDEN_VW

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

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

WBX_CTR_ONDERHANDEN_WERK_VW

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

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

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_VA	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 Health Insurance Data Marts

WBX_CTR_DRIEHOEKSCONTROLE_VW

This view shows the results of the ‘driehoekscontrole’ on Claims, Financial Transactions and Grand Ledger Journal Entries.

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
FFL_ID	Unique ID of the Financial Transaction record. Referencing to the FVS_FINANCIEEL_FEIT_REGELS table
DATUM_FFL_ONTSTAAN	Creation date of the Financial Transaction record
BEDRAG_FINANCIEEL_FEIT	The amount of the Financial Transaction
DDL_ID	Unique ID of the Claims record. Referencing to the DWH_DECLARATIE_DETAILS table
DATUM_DDL_GELADEN	Date on which the Claims record was loaded to the DWH_DECLARATIE_DETAILS table
BEDRAG_DECLARATIE_DETAIL	The amount associated with the Claims record
GBS_ID	Unique ID of the Grand Ledger Journal Entry record. Referencing to the FVS#GROOTBOEK_REGEL_DETAILS table
DATUM_GBS_GELADEN	Date on which the Grand Ledger Journal Entry record was loaded to the FVS#GROOTBOEK_REGEL_DETAILS table
BEDRAG_GROOTBOEKREGEL	The amount associated with the Grand Ledger Journal Entry record
AANTAL_VERSCHILLEN	Amount of differences
FFL_VS_DDL_OK	Indicator for differences between Financial Transactions and Claims
FFL_VS_GBS_OK	Indicator for differences between Financial Transactions and Grand Ledger Journal Entries
DDL_VS_GBS_OK	Indicator for differences between Claims and Grand Ledger Journal Entries

WBX_CTR_FINANCIEEL_VW

This view shows the results of the compatibility checks on the financial 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
FACTUURGATEGORIE	Financial category
DATUM_AFGEHANDELD	Date completed
SOORT_BEDRAG	Description of the checked amount
BEDRAG_BO	Amount of the payable/receivable in OHI Back Office
BEDRAG_BI	Amount of the payable/receivable in Oracle Health Insurance Data Marts

Logging load scripts

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

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

Authorization

User access

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

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

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

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

External tables

The Oracle Health Insurance Data Marts load run makes use of external tables. These are files on the server that are treated as tables by the database. These external tables reside on the Oracle Health Insurance Data Marts 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 must have a reference to these directories, these directory objects are created in the Oracle Health Insurance Data Marts 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 Data Marts it can be added by means of window 'Claim Flex Fields' (ZRGO030F).

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).

Examples

Example 1

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

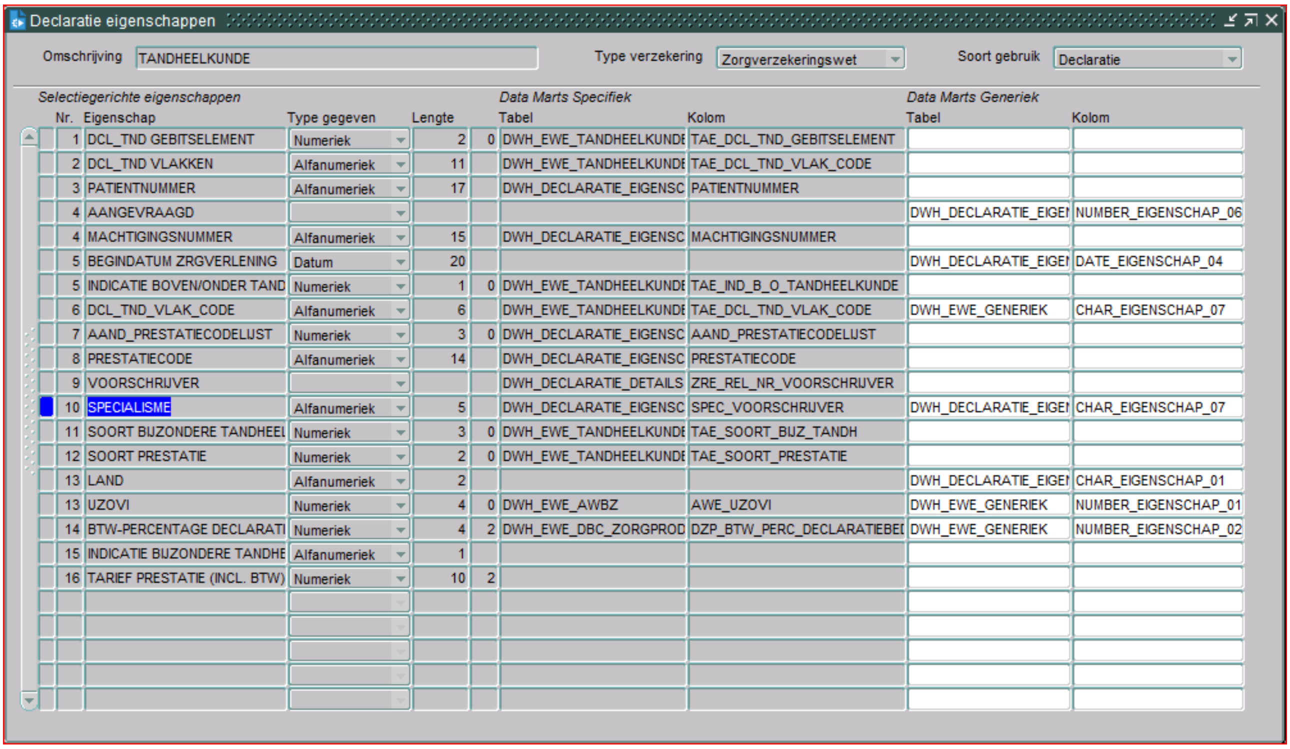
The following claim properties of the procedure claim group 'Tandheelkunde' are available as fixed columns in OHI Data Marts:

Claim Property	BI table	BI column
Aand. prestatiecodelijst	DWH_DECLARATIE_EIGENSCHAPPEN	AAND_PRESTATIECODELIJST
Gebitselement	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_GEBITSELEMENT
Vlakcode	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE
Machtigingsnummer	DWH_DECLARATIE_EIGENSCHAPPEN	MACHTIGINGSNUMMER
Patientnummer	DWH_DECLARATIE_EIGENSCHAPPEN	PATIENTNUMMER
Prestatiecode	DWH_DECLARATIE_EIGENSCHAPPEN	PRESTATIECODE
Soort prestatie	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE
Specialisme	DWH_DECLARATIE_EIGENSCHAPPEN	SPEC_VOORSCHRIJVER

In this example these attributes are moved into generic claim property columns.
To load these claim properties into the generic structure, the mapping to the generic structure must be added. The claim properties can be loaded into the following generic attributes:

Claim Property	Table	Generic Attribute
Aand. prestatiecodelijst	DWH_DECLARATIE_EIGENSCHAPPEN	NUMBER_EIGENSCHAP_01
Gebitselement	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_01
Vlakcode	DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_01
Machtigingsnummer	DWH_DECLARATIE_EIGENSCHAPPEN	CHAR_EIGENSCHAP_01
Patientnummer	DWH_DECLARATIE_EIGENSCHAPPEN	CHAR_EIGENSCHAP_02
Prestatiecode	DWH_DECLARATIE_EIGENSCHAPPEN	CHAR_EIGENSCHAP_03
Soort prestatie	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_02
Specialisme	DWH_DECLARATIE_EIGENSCHAPPEN	CHAR_EIGENSCHAP_04

In this case the window 'Claims Flex Fields' (ZRGO030F) should look like this:



Declaratie eigenschappen

Omschrijving: TANDHEELKUNDE

Type verzekering: Zorgverzekeringswet

Soort gebruik: Declaratie

Selectiegerichte eigenschappen					Data Marts Specifiek		Data Marts Generiek	
Nr.	Eigenschap	Type gegeven	Lengte		Tabel	Kolom	Tabel	Kolom
1	DCL_TND GEBITSELEMENT	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_GEBITSELEMENT		
2	DCL_TND VLAKKEN	Alfanumeriek	11		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE		
3	PATIENTNUMMER	Alfanumeriek	17		DWH_DECLARATIE_EIGENSC	PATIENTNUMMER		
4	AANGEVRAAGD						DWH_DECLARATIE_EIGEI	NUMBER_EIGENSCHAP_06
4	MACHTIGINGSNUMMER	Alfanumeriek	15		DWH_DECLARATIE_EIGENSC	MACHTIGINGSNUMMER		
5	BEGINDATUM ZRGVERLENING	Datum	20				DWH_DECLARATIE_EIGEI	DATE_EIGENSCHAP_04
5	INDICATIE BOVEN/ONDER TAND	Numeriek	1	0	DWH_EWE_TANDHEELKUNDE	TAE_IND_B_O_TANDHEELKUNDE		
6	DCL_TND_VLAK_CODE	Alfanumeriek	6		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE	DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_07
7	AAND_PRESTATIECODELIJST	Numeriek	3	0	DWH_DECLARATIE_EIGENSC	AAND_PRESTATIECODELIJST		
8	PRESTATIECODE	Alfanumeriek	14		DWH_DECLARATIE_EIGENSC	PRESTATIECODE		
9	VOORSCHRIJVER				DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER		
10	SPECIALISME	Alfanumeriek	5		DWH_DECLARATIE_EIGENSC	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGEI	CHAR_EIGENSCHAP_07
11	SOORT BIJZONDERE TANDHEEL	Numeriek	3	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BIJZ_TANDH		
12	SOORT PRESTATIE	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE		
13	LAND	Alfanumeriek	2				DWH_DECLARATIE_EIGEI	CHAR_EIGENSCHAP_01
13	UZOVI	Numeriek	4	0	DWH_EWE_AWBZ	AWE_UZOVI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATI	Numeriek	4	2	DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_02
15	INDICATIE BIJZONDERE TANDHE	Alfanumeriek	1					
16	TARIEF PRESTATIE (INCL. BTW)	Numeriek	10	2				

In this window:

- 'Omschrijving' contains the procedure claim group as it is defined in OHI Back Office; in this case this is ‘TANDHEELKUNDE’.
- 'Data Marts Specifiek, Tabel' contains the location of the old fixed structure. In this example a few properties are part of DWH_DECLARATIE_EIGENSCHAPPEN and several others are part of DWH_DECLARATIE_DETAILS and DWH_EWE_TANDHEELKUNDE.
- 'Data Marts Specifiek, Column' contains the column name in the old structure.

The next two columns provide the mapping to the new generic structure.

- 'Data Marts Generiek, Tabel' contains the table where the claim property should be loaded.
- The column 'Data Marts Generiek, Column' contains the attribute where the claim property is loaded.

The value of 'Omschrijving' is loaded into DWH_DECLARATIE_DETAILS, DWH_DECLARATIE_EIGENSCHAPPEN (column dwh_declaratie_details.gebruikgroep / dwh_declaratie_eigenschappen.gebruikgroep). The combination of the generic column and the column dwh_declaratie_details.gebruikgroep, dwh_declaratie_eigenschappen determines the functional meaning of the column.

In the above example, the column dwh_declaratie_eigenschappen.char_eigenschap_07 for DWH_DECLARATIE_DETAILS.SPEC_VOORSCHRIJVER with gebruikgroep is equal to ‘TANDHEELKUNDE’. And it contains the claim property SPECIALISME.

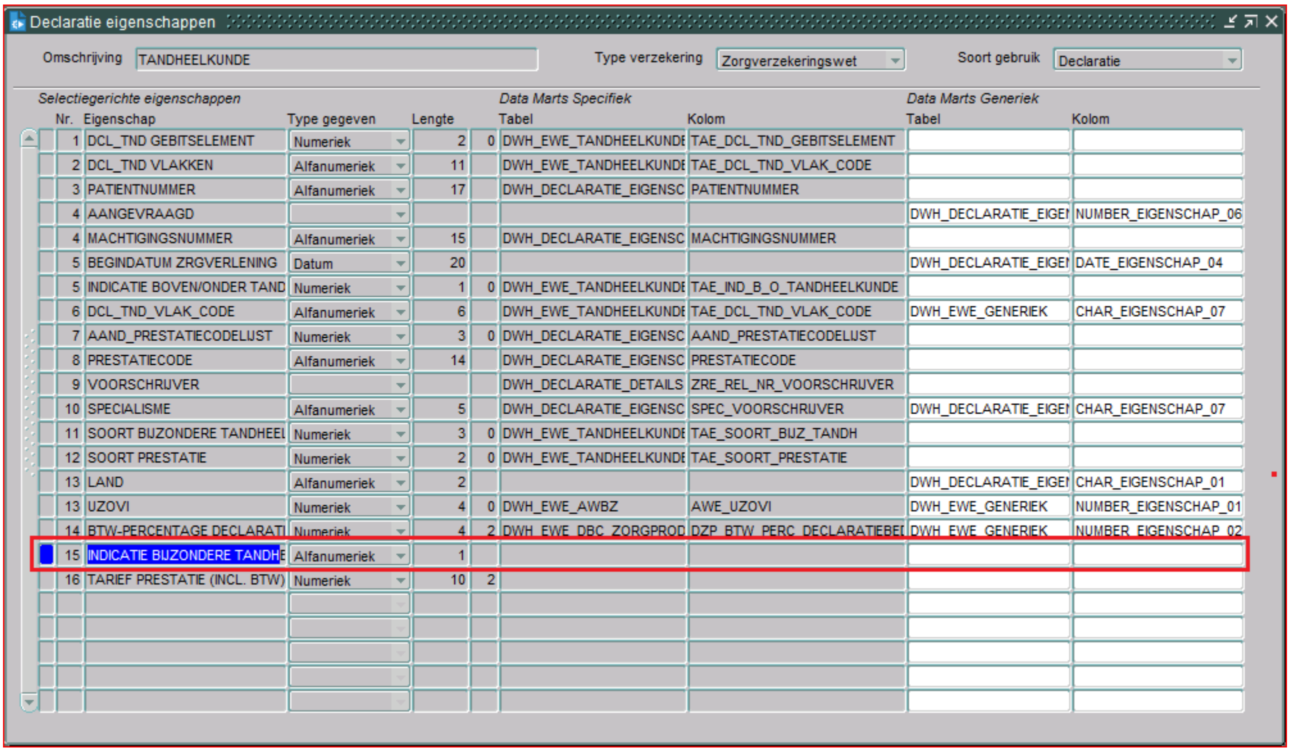
Example 2:

A new claim property ‘INDICATIE_BIJZONDERE_TANDHEELKUNDE’ is introduced in OHI Back Office for procedure claim group ‘TANDHEELKUNDE’, this claim property should be added to OHI Data Marts. The claim property can have two values in OHI Back Office: ‘Y’ or ‘N’.

To add this claim property, in the window 'Claim Flex Fields' (ZRGO030F) the new property must be altered and a column should be chosen where the claim property 'INDICATIE_BIJZONDERE_TANDHEELKUNDE' is loaded into OHI Data Marts.

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

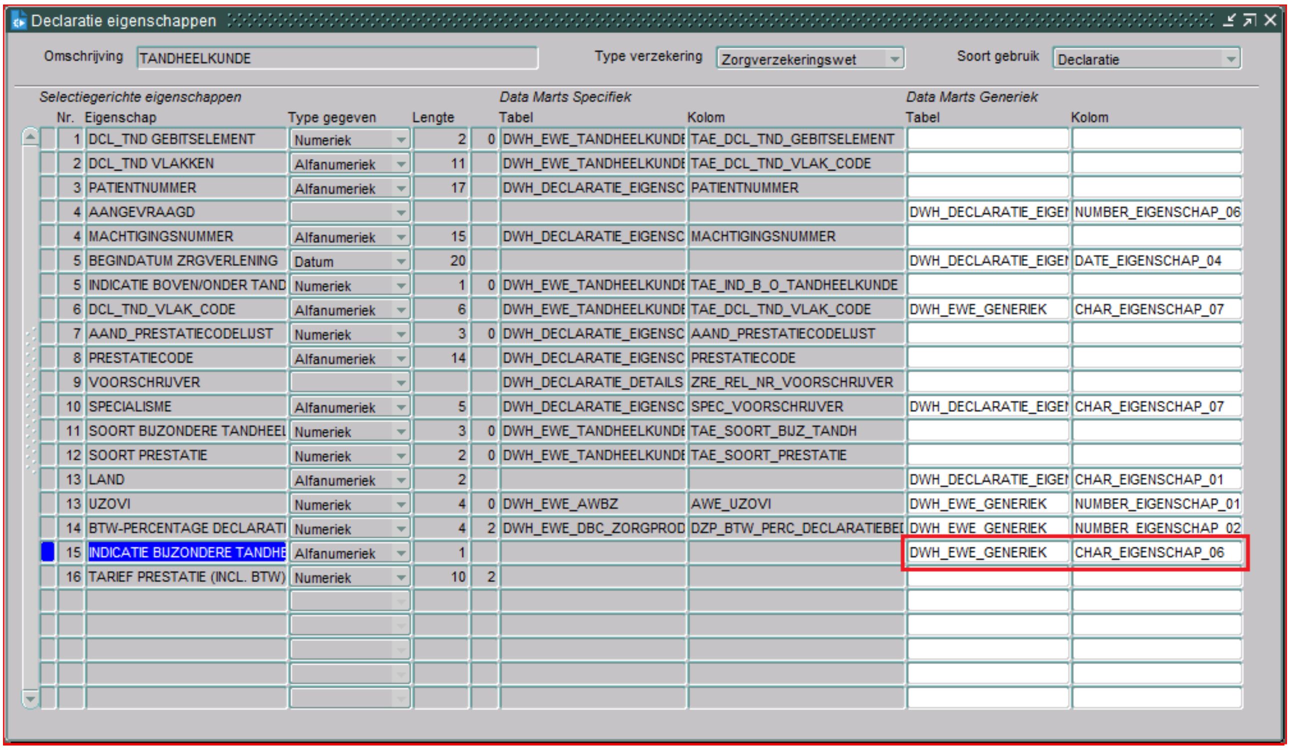
The property is a character so we should map it to a character-typed attribute, which is not yet in use. For example, DWH_EWE_GENERIEK.CHAR_EIGENSCHAP_06. This should be changed in window 'Claim Flex Fields' (ZRGO030F):



Declaratie eigenschappen

Omschrijving: TANDHEELKUNDE
Type verzekering: Zorgverzekeringswet
Soort gebruik: Declaratie

Selectiegerichte eigenschappen					Data Marts Specifiek		Data Marts Generiek	
Nr.	Eigenschap	Type gegeven	Lengte		Tabel	Kolom	Tabel	Kolom
1	DCL_TND GEBITSELEMENT	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_GEBITSELEMENT		
2	DCL_TND VLAKKEN	Alfanumeriek	11		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE		
3	PATIENTNUMMER	Alfanumeriek	17		DWH_DECLARATIE_EIGENSC	PATIENTNUMMER		
4	AANGEVRAAGD						DWH_DECLARATIE_EIGEN	NUMBER_EIGENSCHAP_06
4	MACHTIGINGSNUMMER	Alfanumeriek	15		DWH_DECLARATIE_EIGENSC	MACHTIGINGSNUMMER		
5	BEGINDATUM ZRGVERLENING	Datum	20				DWH_DECLARATIE_EIGEN	DATE_EIGENSCHAP_04
5	INDICATIE BOVEN/ONDER TAND	Numeriek	1	0	DWH_EWE_TANDHEELKUNDE	TAE_IND_B_O_TANDHEELKUNDE		
6	DCL_TND_VLAK_CODE	Alfanumeriek	6		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE	DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_07
7	AAND_PRESTATIECODELIJST	Numeriek	3	0	DWH_DECLARATIE_EIGENSC	AAND_PRESTATIECODELIJST		
8	PRESTATIECODE	Alfanumeriek	14		DWH_DECLARATIE_EIGENSC	PRESTATIECODE		
9	VOORSCHRIJVER				DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER		
10	SPECIALISME	Alfanumeriek	5		DWH_DECLARATIE_EIGENSC	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGEN	CHAR_EIGENSCHAP_07
11	SOORT BIJZONDERE TANDHEEL	Numeriek	3	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BIJZ_TANDH		
12	SOORT PRESTATIE	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE		
13	LAND	Alfanumeriek	2				DWH_DECLARATIE_EIGEN	CHAR_EIGENSCHAP_01
13	UZOVI	Numeriek	4	0	DWH_EWE_AWBZ	AWE_UZOVI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATI	Numeriek	4	2	DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_02
15	INDICATIE BIJZONDERE TANDHE	Alfanumeriek	1					
16	TARIEF PRESTATIE (INCL. BTW)	Numeriek	10	2				



Declaratie eigenschappen

Omschrijving: TANDHEELKUNDE
Type verzekering: Zorgverzekeringswet
Soort gebruik: Declaratie

Selectiegerichte eigenschappen					Data Marts Specifiek		Data Marts Generiek	
Nr.	Eigenschap	Type gegeven	Lengte		Tabel	Kolom	Tabel	Kolom
1	DCL_TND GEBITSELEMENT	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_GEBITSELEMENT		
2	DCL_TND VLAKKEN	Alfanumeriek	11		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE		
3	PATIENTNUMMER	Alfanumeriek	17		DWH_DECLARATIE_EIGENSC	PATIENTNUMMER		
4	AANGEVRAAGD						DWH_DECLARATIE_EIGEN	NUMBER_EIGENSCHAP_06
4	MACHTIGINGSNUMMER	Alfanumeriek	15		DWH_DECLARATIE_EIGENSC	MACHTIGINGSNUMMER		
5	BEGINDATUM ZRGVERLENING	Datum	20				DWH_DECLARATIE_EIGEN	DATE_EIGENSCHAP_04
5	INDICATIE BOVEN/ONDER TAND	Numeriek	1	0	DWH_EWE_TANDHEELKUNDE	TAE_IND_B_O_TANDHEELKUNDE		
6	DCL_TND_VLAK_CODE	Alfanumeriek	6		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE	DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_07
7	AAND_PRESTATIECODELIJST	Numeriek	3	0	DWH_DECLARATIE_EIGENSC	AAND_PRESTATIECODELIJST		
8	PRESTATIECODE	Alfanumeriek	14		DWH_DECLARATIE_EIGENSC	PRESTATIECODE		
9	VOORSCHRIJVER				DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER		
10	SPECIALISME	Alfanumeriek	5		DWH_DECLARATIE_EIGENSC	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGEN	CHAR_EIGENSCHAP_07
11	SOORT BIJZONDERE TANDHEEL	Numeriek	3	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BIJZ_TANDH		
12	SOORT PRESTATIE	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE		
13	LAND	Alfanumeriek	2				DWH_DECLARATIE_EIGEN	CHAR_EIGENSCHAP_01
13	UZOVI	Numeriek	4	0	DWH_EWE_AWBZ	AWE_UZOVI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATI	Numeriek	4	2	DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_02
15	INDICATIE BIJZONDERE TANDHE	Alfanumeriek	1				DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_06
16	TARIEF PRESTATIE (INCL. BTW)	Numeriek	10	2				

The complete window will now look as follows:

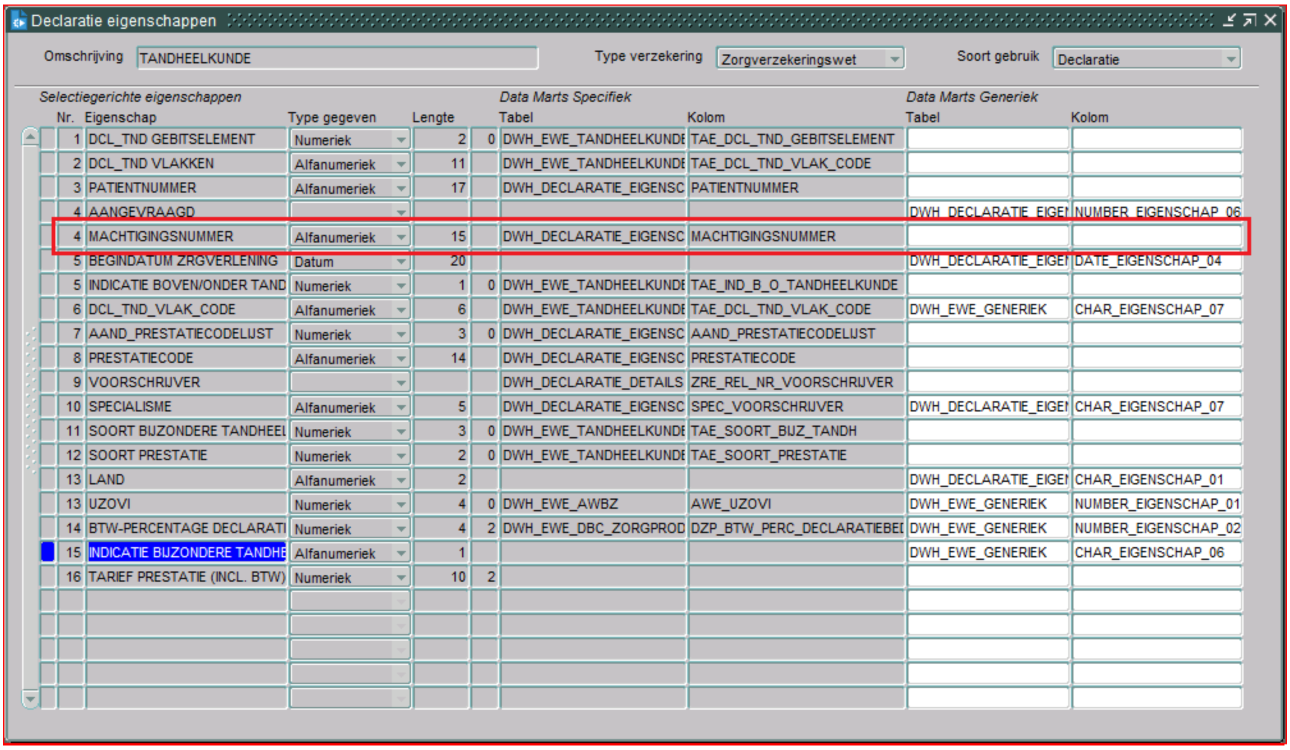
Note that the columns 'Tabel' and 'Kolom' of 'Data Marts Specifiek' are not set up for INDICATIE_BIJZONDERE_TANDHEELKUNDE, this is because this claim property does not exist in the old fixed structure (claim property is added after release 2012.01).

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_DECLARATIE_EIGENSCHAPPEN, 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. Once a property is mapped to a generic attribute, this can not be undone!



Declaratie eigenschappen

Omschrijving TANDHEELKUNDE

Type verzekering Zorgverzekeringswet

Soort gebruik Declaratie

Selectiegerichte eigenschappen					Data Marts Specifiek		Data Marts Generiek	
Nr.	Eigenschap	Type gegeven	Lengte		Tabel	Kolom	Tabel	Kolom
1	DCL_TND GEBITSELEMENT	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_GEBITSELEMENT		
2	DCL_TND VLAKKEN	Alfanumeriek	11		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE		
3	PATIENTNUMMER	Alfanumeriek	17		DWH_DECLARATIE_EIGENSC	PATIENTNUMMER		
4	AANGEVRAAGD						DWH_DECLARATIE_EIGEI	NUMBER_EIGENSCHAP_06
4	MACHTIGINGSNUMMER	Alfanumeriek	15		DWH_DECLARATIE_EIGENSC	MACHTIGINGSNUMMER		
5	BEGINDATUM ZRGVERLENING	Datum	20				DWH_DECLARATIE_EIGEI	DATE_EIGENSCHAP_04
5	INDICATIE BOVEN/ONDER TAND	Numeriek	1	0	DWH_EWE_TANDHEELKUNDE	TAE_IND_B_O_TANDHEELKUNDE		
6	DCL_TND_VLAK_CODE	Alfanumeriek	6		DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE	DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_07
7	AAND_PRESTATIECODELIJST	Numeriek	3	0	DWH_DECLARATIE_EIGENSC	AAND_PRESTATIECODELIJST		
8	PRESTATIECODE	Alfanumeriek	14		DWH_DECLARATIE_EIGENSC	PRESTATIECODE		
9	VOORSCHRIJVER				DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER		
10	SPECIALISME	Alfanumeriek	5		DWH_DECLARATIE_EIGENSC	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGEI	CHAR_EIGENSCHAP_07
11	SOORT BIJZONDERE TANDHEEL	Numeriek	3	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BIJZ_TANDH		
12	SOORT PRESTATIE	Numeriek	2	0	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE		
13	LAND	Alfanumeriek	2				DWH_DECLARATIE_EIGEI	CHAR_EIGENSCHAP_01
13	UZOVI	Numeriek	4	0	DWH_EWE_AWBZ	AWE_UZOVI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATI	Numeriek	4	2	DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_02
15	INDICATIE BIJZONDERE TANDHE	Alfanumeriek	1				DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_06
16	TARIEF PRESTATIE (INCL. BTW)	Numeriek	10	2				

Note that only 'Tabel' and 'Kolom' of 'Data Marts Specifiek' are set up for MACHTIGINGSNUMMER. Also note that this is only relevant for claim properties that were already available in the old fixed structure in OHI Data Marts.

When to place a claim property in a dimension table

There are two places where a claim property can be added in OHI Data Marts: fact table DWH_DECLARATIE_EIGENSCHAPPEN 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_DECLARATIE_EIGENSCHAPPEN 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. Because of this it should be placed in DWH_DECLARATIE_EIGENSCHAPPEN. 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_DECLARATIE_EIGENSCHAPPEN attributes and a dynamic created set of claim properties. In DWH_DECLARATIE_EIGENSCHAPPEN 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 instead of the generic name. The view can be used to represent the data with column names that have a functional meaning.

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

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

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

Add claim properties from generic structure to OBI EE repository

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

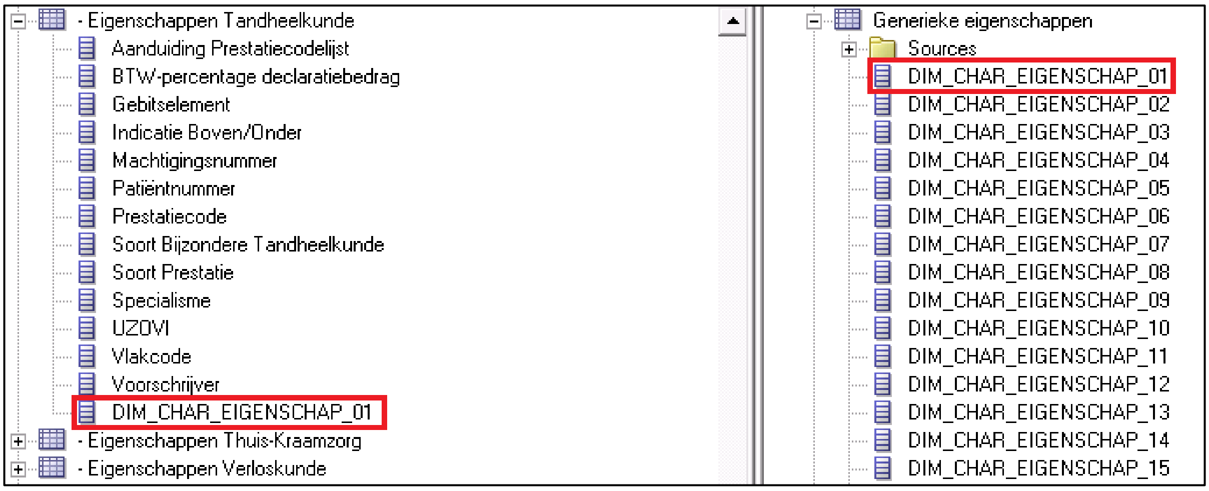
All generic claim property attributes are already added to the physical and business model layer of the OBI EE repository.

How a claim property attribute can be added to the 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 presentation table ‘Eigenschappen Tandheelkunde’ and should be named ‘Example 123’.

☞ **Note 1:** Attributes sourced from DWH_EWE_GENERIEK are prefixed with ‘DIM_’ within the Logical Table ‘Generieke eigenschappen’, while attributes sourced from DWH_DECLARATIE_EIGENSCHAPPEN and DWH_DECLARATIES are prefixed with ‘FACT_’.

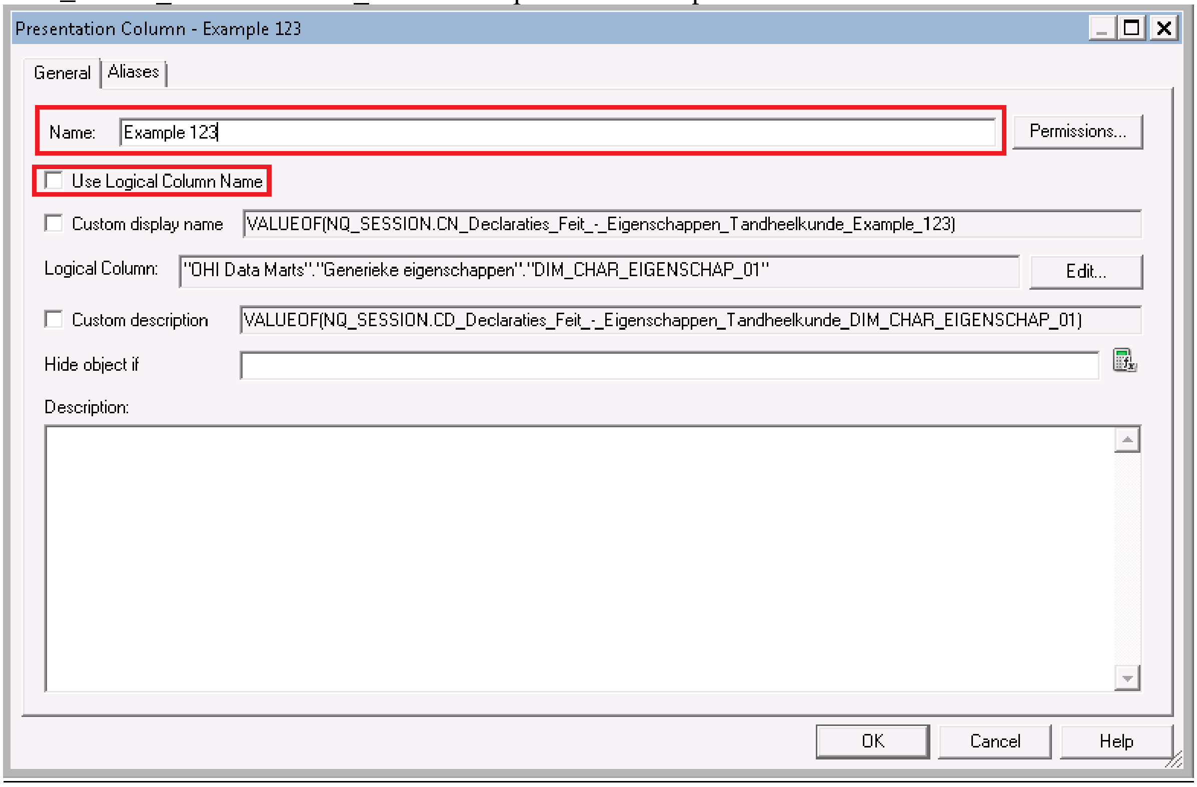
1. Add the generic column to the presentation model
   Drag the attribute ‘DIM_CHAR_EIGENSCHAP_01’ from the Logical Table ‘Generieke eigenschappen’ to the Presentation Table ‘- Eigenschappen Tandheelkunde’. The should result in the following situation



- Eigenschappen Tandheelkunde
  - Aanduiding Prestatiecodelijst
  - BTW-percentage declaratiebedrag
  - Gebitselement
  - Indicatie Boven/Onder
  - Machtigingsnummer
  - Patiëntnummer
  - Prestatiecode
  - Soort Bijzondere Tandheelkunde
  - Soort Prestatie
  - Specialisme
  - UZOVI
  - Vlakcode
  - Voorschrijver
  - DIM_CHAR_EIGENSCHAP_01
- Eigenschappen Thuis-Kraamzorg
- Eigenschappen Verloskunde

Generieke eigenschappen
- Sources
- DIM_CHAR_EIGENSCHAP_01
- DIM_CHAR_EIGENSCHAP_02
- DIM_CHAR_EIGENSCHAP_03
- DIM_CHAR_EIGENSCHAP_04
- DIM_CHAR_EIGENSCHAP_05
- DIM_CHAR_EIGENSCHAP_06
- DIM_CHAR_EIGENSCHAP_07
- DIM_CHAR_EIGENSCHAP_08
- DIM_CHAR_EIGENSCHAP_09
- DIM_CHAR_EIGENSCHAP_10
- DIM_CHAR_EIGENSCHAP_11
- DIM_CHAR_EIGENSCHAP_12
- DIM_CHAR_EIGENSCHAP_13
- DIM_CHAR_EIGENSCHAP_14
- DIM_CHAR_EIGENSCHAP_15

2. Provide a functional description for the Presentation attribute
   Uncheck the checkbox 'Use Logical Column Name' and change the property 'Name' of DIM_CHAR_EIGENSCHAP_01 to 'Example 123' in the presentation table



Presentation Column - Example 123

Name: Example 123

□ Use Logical Column Name

□ Custom display name VALUEOF(NQ_SESSION.CN_Declaraties_Feit_-_Eigenschappen_Tandheelkunde_Example_123)

Logical Column: "OHI Data Marts"."Generieke eigenschappen"."DIM_CHAR_EIGENSCHAP_01"

□ Custom description VALUEOF(NQ_SESSION.CD_Declaraties_Feit_-_Eigenschappen_Tandheelkunde_DIM_CHAR_EIGENSCHAP_01)

Hide object if

Description:

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

Configuration of generic structure for dimension properties

Introduction

It is possible to register a set of dimension properties in OHI Back Office. When a dimension property is required for use in OHI Data Marts it can be added by window 'Other Flex Fields' (ZRGO031F).

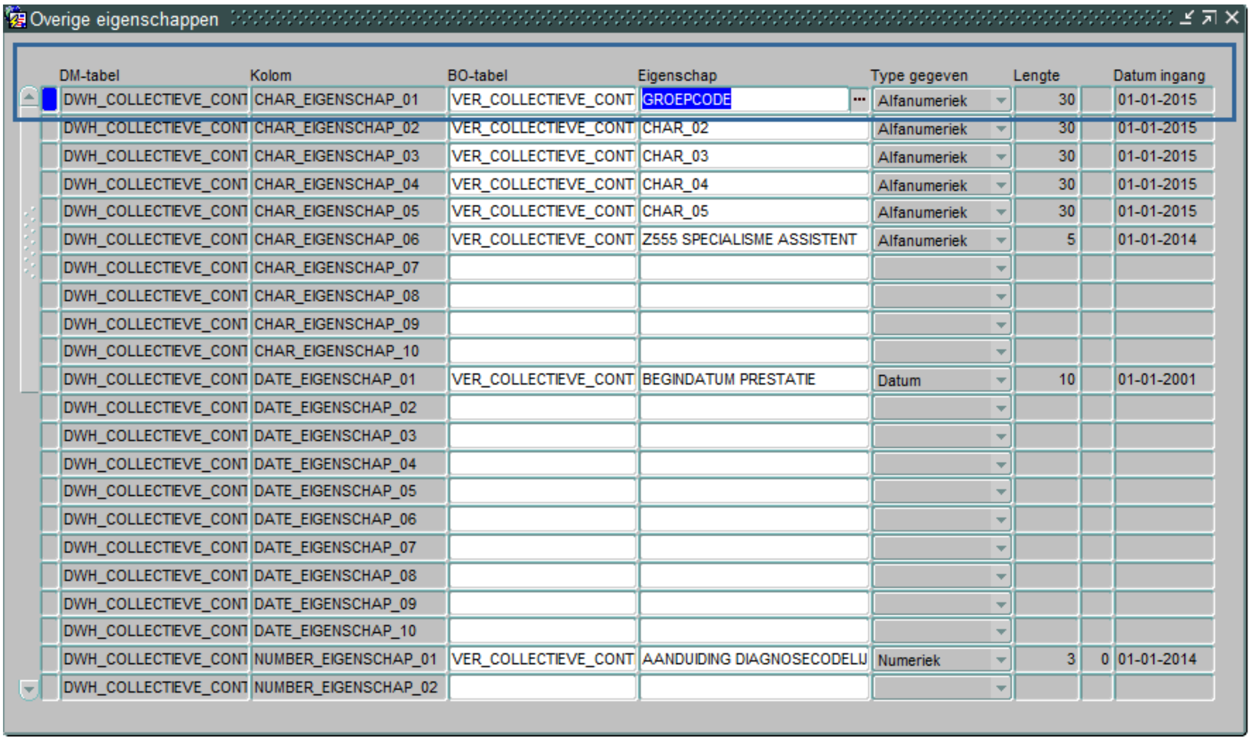
Examples

<u>Example</u>

A new dimension property ‘GROEPCODE’ is introduced in OHI Back Office for group contracts, this dimension property should be added to OHI Data Marts.

To add this dimension property, th window 'Other Flex Fields' (ZRGO031F) should be used. Query on 'DWH_COLLECTIEVE_CONTRACTEN' in field 'DM-tabel'. The new property must be added and a column should be defined where the dimension property ‘GROEPCODE’ is loaded into OHI Data Marts.

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



DM-tabel	Kolom	BO-tabel	Eigenschap	Type gegeven	Lengte		Datum ingang
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_01	VER_COLLECTIEVE_CONT	GROEPCODE	Alfanumeriek	30		01-01-2015
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_02	VER_COLLECTIEVE_CONT	CHAR_02	Alfanumeriek	30		01-01-2015
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_03	VER_COLLECTIEVE_CONT	CHAR_03	Alfanumeriek	30		01-01-2015
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_04	VER_COLLECTIEVE_CONT	CHAR_04	Alfanumeriek	30		01-01-2015
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_05	VER_COLLECTIEVE_CONT	CHAR_05	Alfanumeriek	30		01-01-2015
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_06	VER_COLLECTIEVE_CONT	Z555 SPECIALISME ASSISTENT	Alfanumeriek	5		01-01-2014
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_07						
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_08						
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_09						
DWH_COLLECTIEVE_CONT	CHAR_EIGENSCHAP_10						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_01	VER_COLLECTIEVE_CONT	BEGINDATUM PRESTATIE	Datum	10		01-01-2001
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_02						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_03						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_04						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_05						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_06						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_07						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_08						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_09						
DWH_COLLECTIEVE_CONT	DATE_EIGENSCHAP_10						
DWH_COLLECTIEVE_CONT	NUMBER_EIGENSCHAP_01	VER_COLLECTIEVE_CONT	AANDUIDING DIAGNOSECODELIJ	Numeriek	3	0	01-01-2014
DWH_COLLECTIEVE_CONT	NUMBER_EIGENSCHAP_02						

Generate views to create a functional overlay over the generic structure

It is possible to generate a view per dimension that contains all attributes of the dimension. For the generic dimension properties there is a generic naming convention (e.g. CHAR_EIGENSCHAP_01). With the view generator it is possible to create a view that contains all dimension properties, the attributes will obtain the functional name of the dimension property (as defined in 'Other Flex Fields' (ZRGO031F)) instead of the generic name. The view can be used to represent the data with column names that have a functional meaning.

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

```
exec wbx_view_generator.generate_dimensie_view(‘DWH_COLLECTIEVE_CONTRACTEN’);
```

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

Add dimension properties from generic structure to OBI EE repository

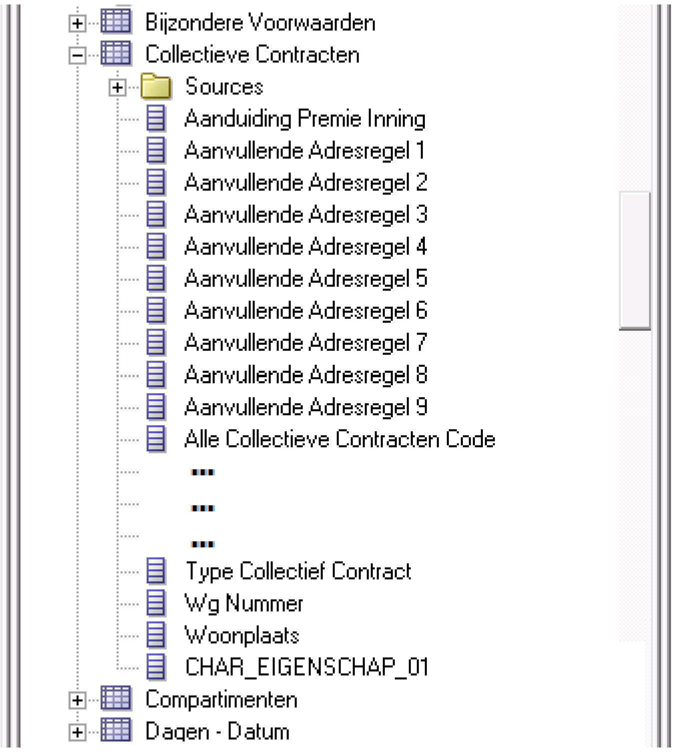
This paragraph describes how to add generically mapped dimension properties to the OBI EE repository business model.

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

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

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

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



2. <u>Name the column to the functional meaning</u>
   Change the name of CHAR_EIGENSCHAP_01 to ‘Groep Code’ in the logical table

3. <u>Add the new attribute to the presentation layer</u>
   Add ‘Groep Code’ to the Presentation table ‘Collectieve Contracten’

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

Part III - Technical management

New releases of Oracle Health Insurance Data Marts

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

The Oracle Health Insurance installation menu OHIPATCH must be used for the installation of new Oracle Health Insurance Data Marts releases or patches. For operation of the installation menu: see document 'PCM02101.pdf' (Oracle Health Insurance Installation of Releases).

Reorganization of tables

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

For the partitioned facts tables:

- DWH_AFGEWEZEN_DECLARATIES
- DWH_DECL_PRESTATIE_DETAILS
- DWH_DECLARATIES
- DWH_DECLARATIES_AGG
- DWH_DECLARATIE_BOEKINGEN
- DWH_DECLARATIE_BRON_KOLOMMEN
- DWH_DECLARATIE_DETAILS
- DWH_DECLARATIE_EIGENSCHAPPEN
- DWH_DEELNAME_MUTATIES
- DWH_ONDERHANDEN_WERK
- DWH_PREMIES
- DWH_VERBINTENISSEN
- DWH_VERPLICHTINGEN
- DWH_VERZEKERDEN
- DWH_VORDERINGEN
- 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_OPENSTAANDE_VERPLICHTINGEN
- DWH_OPENSTAANDE_VORDERINGEN
- DWH_TOE_UITTREDINGEN

this can be done using:

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

COMPRESSION OF PARTITIONED FACT TABLES

As of Oracle Health Insurance Data Marts 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 OHIPATCH.pl is available for this. This choice is the same for OHI Back Office and Oracle Health Insurance Data Marts, however the following submenu choices are only shown for Oracle Health Insurance Data Marts.

```
INFO : ============================================
INFO : = Redefine a table:                        =
INFO : = O - Compress Advanced                    =
INFO : = Q - Compress for Query (Exadata only)    =
INFO : = A - Compress for Archive (Exadata only)  =
INFO : = U - Uncompress  (revert compression)     =
INFO : ============================================
INFO : Which table redefinition would you like to execute (A, A, Q, U)?
```

If one of the compress actions (O, Q or A) 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 must be rebuilt. Option '870 - Rebuild unusable indexes' in OHIPATCH.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.

Part IV – Set-up of Databsae Vault for Data Marts

Starting with OHI Data Marts release 10.19.1.0.0 an OHI Data Marts specific implementation of Oracle Database Vault is supported. This is a licensed option with the Oracle Database Enterprise Edition to implement more restrictive access to database objects.

Database Vault for OHI Data Marts – Organisational Impact

Typically Database Vault is used to prevent privileged, generic database administration accounts, to be used to access or manipulate OHI data as the OHI data is considered as ‘sensitive data’.

Usually such ‘DBA’ accounts have privileges to access all data in all database objects. As these more technical privileged accounts may be used by people in a service organisation, which is possibly externalized from the organization that uses the OHI application, there is less control over who can use these privileges. This while it is well known the OHI tables contain sensitive person identifying data as well as health and financial information regarding these persons, so very privacy sensitive information. As such it may be that security policies and legal obligations cannot be fulfilled when such generic accounts can fully access this sensitive data.

For these situations OHI supports the implementation of Oracle Database Vault, which means there will be a clear segregation of duties:

- Database administrators will typically execute the day to day database administration and management tasks.
- Application administrators will manage the access to the application data and as such will also be responsible for account and access management. This means at least creating and dropping user accounts, granting privileges and fulfilling unlock and password reset requests. As such only the application administrators can grant access to the application data and will know and control the credentials of accounts that provide access to the application data.

This has quite some impact on database and application management tasks, due to the segregation of duties. The following principles should be adhered to:

- The passwords of the Database Vault management accounts (‘owner’ and ‘account manager’ account, including backup accounts for these) may only be known to the application administrators after the Database Vault implementation has been activated (passwords typically have to be reset when these accounts were created by the database administrators).
- Only the application administrators create new accounts and may implement actions like reset, unlock and lock of a database account.
- Specifying passwords for ‘datasource’ accounts (for example for web services), wallet accounts, etc. must be done by the application administrators. Depending on which administrators implement application maintenance tasks this may mean that passwords have to be provided in ‘shared screen’ sessions, where the database administrator may execute technical configuration and the application administrator needs to provide the passwords for accounts when prompted for.
- The OHI accounts that can be used through a wallet that is restrictively accessible on application server level, but as such typically available for privileged operating system and database accounts, need extra protection. Having access to the wallet may not lead by default to having access to the sensitive data.
- Dual handed tasks should always remain dual handed (no password handover should occur unless directly reset afterwards).
- A robust and secure auditing of all database vault related command executing is required.

PREPARING YOUR DATABASE ENVIRONMENT FOR IMPLEMENTING DATABASE VAULT

This paragraph will describe how to install, configure and enable the Database Vault option for your OHI Data Marts database. This is necessary before an OHI specific implementation, meaning a pre-defined OHI Data Marts specific realm to protect the OHI Data Marts data from being accessed through regular DBA accounts, can be activated.

Installing Database Vault

Check the Oracle Documentation for the installation instructions.

The database vault components need to be installed on both the root container as well as in the pluggable database. You need both Oracle Label Security (OLS) and Database Vault (DV) although you do not need a license for OLS when it is only used for DV.

Check whether these components are present in your root container and pluggable database(s) by querying from within the root with opened pluggable database(s):

```
select comp_name, status, schema, other_schemas , con_id
from   cdb_registry where comp_id in ('OLS','DV')
order by con_id, comp_name
```

A possible result (query CDB_PDBS when you do not know which container id value belongs to which pluggable database):

COMP_NAME	STATUS	SCHEMA	OTHER_SCHEMAS	CON_ID
Oracle Database Vault	VALID	DVSYS	DVF	1
Oracle Label Security	VALID	LBACSYS		1
Oracle Database Vault	VALID	DVSYS	DVF	3
Oracle Label Security	VALID	LBACSYS		3

When DV is not present in your target container (check the CON_ID column above) or not valid run the scripts to create these components in the relevant container database (root, OHI pluggable or both).
The scripts to create the OLS and DV components are respectively:

@?/rdbms/admin/catols.sql

@?/rdbms/admin/catmac.sql

Typically your 12.2 created CDB$ROOT container will already have OLS and DV installed and you only need to install this in the PDB.

Run both scripts in sqlplus while connected to the target container (CDB$ROOT and/or some pluggable database container) and make sure you spool the result for in case a failure occurs.

In the example below, parameter 1 and 2 are explicitly undefined before starting catmac.sql for in case you have some sqlplus containing startup script which uses startup parameters. These undefine commands will not fail when you do not have this.

Parameter 1 specifies the default tablespace for the created users (usually ‘SYSAUX’) and the second parameter specifies the temporary tablespace (usually ‘TEMP’). The question mark notation is used to identify the database 12.2 Oracle Home.

You may run the scripts with the commands as shown below:

```
spool dv_install
@?/rdbms/admin/catols.sql
undefine 1
undefine 2
@?/rdbms/admin/catmac.sql
spool off
```

Or, as non-interactive (scriptable) alternative:

```
spool
dv_install
@?/rdbms/admin/catols.sql
undefine 1
undefine 2
@?/rdbms/admin/catmac.sql SYSAUX TEMP
spool off
```

Rerun the CDB_REGISTRY query above to check OLS and DB are installed and valid. The above scripts should be used to install Database Vault on both your CDB as your PDB’s.

Configuring Database Vault for CDB$ROOT

Once you completed the above steps on your CDB, databse vault should be installed and valid. The next step is to configure Database Vault within the CDB.

Check the Oracle Documentation for the registration instructions, meaning the configuration and enablement for the root container.

As part of the registration process, the Database Vault administrative accounts must be created.

These are user accounts that are granted Database Vault DV_OWNER and DV_ACCTMGR roles. As a safety measure, Oracle recommends that you create backups of these user accounts (for in case the password of the primary account is lost and must be reset).

ATTENTION: Passwords should be specified/changed by the ‘future’ Database Vault manager team, typically a different team than the DBA team that will manage the database.

This should be done in the root container database by creating common accounts. Not in a pluggable database. Choose your own usernames for the Database Vault administrative accounts. You may use the example commands below.

```
GRANT CREATE SESSION, SET CONTAINER TO c##dbv_owner_root IDENTIFIED BY
&password_root CONTAINER = ALL;

GRANT CREATE SESSION, SET CONTAINER TO c##dbv_owner_root_backup IDENTIFIED BY
&password_root_bkp CONTAINER = ALL;

GRANT CREATE SESSION, SET CONTAINER TO c##dbv_acctmgr_root IDENTIFIED BY
&password_am_root CONTAINER = ALL;

GRANT CREATE SESSION, SET CONTAINER TO c##dbv_acctmgr_root_backup IDENTIFIED BY
&password_am_root_bkp CONTAINER = ALL;
```

After these accounts are created, configure the primary Database vault users:

```
BEGIN
  CONFIGURE_DV
  ( dvowner_uname    => 'c##dbv_owner_root'
  , dvacctmgr_uname => 'c##dbv_acctmgr_root'
  );
END;
/
```

When you run into the error below please close your pluggable database and re-execute the command:

```
ORA-65048: error encountered when processing the current DDL statement
in pluggable database <pdb_name>
……
ORA-65092: system privilege granted with a different scope to 'DBA'
```

If this won't solve the issue, try to compile invalid objects by running:

```
@?/rdbms/admin/utlrp
```

The roles DV_OWNER and DV_ACCTMGR must be granted manually to both backup users:

```
grant DV_OWNER to c##dbv_owner_root_backup;
grant DV_ACCTMGR to c##dbv_acctmgr_root_backup;
```

Recompile invalidated objects in the root container:

```
@?/rdbms/admin/utlrp
```

After this the view CDB_DV_STATUS should show 'true' as status for the DV_CONFIGURE_STATUS record for the root container.

Enable Database Vault in CBS$ROOT

Next we need to enable Database Vault in the root container. Connect as user c##dbv_owner_root and execute the following command to enable Database Vault:

```
exec dbms_macadm.enable_dv;
```

Continue with a restart of the root container database.

After this the view CDB_DV_STATUS should also show 'true' as status for the DV_ENABLE_STATUS record for the root container.

Configuring Database Vault in the PDB

The view CDB_DV_STATUS will show Database Vault is not yet configured and enabled in the pluggable database. It is an option to reuse the common DV accounts as created in the root container. We might prefer a solution where local accounts are created as this provides a more robust implementation because these accounts are stored within the pluggable databsae.

For configuring connect as SYS in the PDB and run the example commands below.

```
GRANT CREATE SESSION TO dbv_owner_root IDENTIFIED BY &password_root;
GRANT CREATE SESSION TO dbv_owner_root_backup IDENTIFIED BY &password_root_bkp;
GRANT CREATE SESSION TO dbv_acctmgr_root IDENTIFIED BY &password_am_root;


GRANT CREATE SESSION TO dbv_acctmgr_root_backup IDENTIFIED BY
&password_am_root_bkp;
```

After these accounts are created, configure the primary Database Vault users.

```
BEGIN
  CONFIGURE_DV
  ( dvowner_uname    => 'dbv_owner_root'
  , dvacctmgr_uname => 'dbv_acctmgr_root'
  );
END;
/
```

If the above script fails, try to restart the PDB. If this does not solve the issue, then make sure that there are no invalid database objects within the PDB.

Recompile invalidated objects in the pluggable database:

```
@?/rdbms/admin/utlrp
```

Enable Database Vault in the PDB

To conclude we need to enable Database Vault also the pluggable database. Only do this when you actually are going to implement Database Vault for OHI Data Marts in the next step. If you do activate Database Vault while not implementing it for OHI Data Marts you may already run into some ‘ORA-01031: insufficient privileges’ symptoms when not proceeding with the implementation, for example running OHIPATCH step 120.

Connect as user dbv_owner_root in the pluggable database and execute the following command to enable Database Vault:

```
exec dbms_macadm.enable_dv;
```

Next restart the PDB.

After this the view CDB_DV_STATUS should show ‘true’ as status for the DV_ENABLE_STATUS record for the pluggable database.

Activate the OHI DM Specific Realm

The database preparations need to ready at this point. Starting with OHI release 10.18.2.2.0 functionality within OHI is contained to create a tailored realm. This paragraph describes how.

Create OHI realm access role

Create within your PDB, while connected with DBA privileges (so typically a DBA team member does this), the role OHI_REALM_ACCESS and grant this role to the OHI Data Marts object owner with admin option. This must be done before the OHI DM realm is created.

This role will typically be granted to custom development users (see next step).

```
create role ohi_realm_access;
```

```
grant ohi_realm_access to &ohi_object_owner with admin option;
```

While still connected with DBA privileges grant select rights to DBA_SYNONYMS to the database owner user. This is needed to determine the owner of objects when creating the DV realm in the next step.

```
grant select on dba_synonyms to dbv_owner_root;
```

Execute privileges on a few OHI DM objects needs to be granted to Database Vault related accounts. The role OHI_REALM_ACCESS needs to be granted to all custom development users by executing a specific packaged procedure for this.

Please execute the commands below while connected as the OHI Data Marts object owner (Default OBD_OWN). The code below uses the local Database Vault root owner account. Adapt this if you have used a different account name. Also adapt the revoke from SYS if you did not create the role using the SYS account.

```
grant execute on alg_db_vault_pck to dbv_owner_root;
grant ohi_realm_access to dbv_owner_root;
grant execute on alg_dv_allow_owner_access_fnc to dvsys;
revoke ohi_realm_access from sys;
```

Grant inherit privilege and create the OHI specific

The commands below need to be run as the Database Vault root owner account in the pluggable database. The first command is needed in order to use the root owner privileges by the OHI packaged procedure that is run afterwards, it allows that package to run with root owner privileges.

Connect as the Database Vault root owner in the pluggable database and execute the following accounts (again dbv_owner_root is used, adapt if you use a different accounts name).

```
grant inherit privileges on user dbv_owner_root to &ohi_object_owner;
set serveroutput on
exec alg_db_vault_pck.cre_ohidm_realm;
```

When the last call has finished the OHI specific Database Vault realm with name ‘OHI Data Marts Data Protection’ is created. You can check the presence by quering the view DBA_DV_REALMS

PART V - APPENDICES

APPENDIX A: CLONING ORACLE HEALTH INSURANCE DATA MARTS ENVIRONMENTS

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

An Oracle Health Insurance Data Marts environment consists the Oracle Health Insurance Data Marts schema OBD_OWN.

Seeing as environment-specific information is stored in the repository, several things must 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

APPENDIX B: EXADATA USAGE

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

1. Please refer to chapter C: "*Converting to Oracle RAC and Oracle Rac One Node from Single Instance Oracle Database.", "Oracle Real Application Clusters Installation Guide 11g Release 2".*
2. Please refer to chapter 8: "*Performing Oracle ASM Data Migration with RMAN*", "*Oracle Automatic Storage Management Administrator's Guide*."
3. Make sure the DB link SRC_OPENZORG.WORLD point to the correct Open Zorg environment.
4. Grant permissions as per administrator reference on the Data Marts Database.

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

5. Make sure the following patchset is installed on the Exadata Database Machine: 16568042
   Note 1: that this patchset is installed in 2012.01, however if the physical machine changes, the software of new and old machines need to be identical.
   Note 2: please note that no SQL statements of the patch need to be applied: they should be already applied in 2012.01 (ie only run Opatch apply)
6. Make sure java and xmldb are installed.
7. Gather schema statistics for OBD_OWN.
8. Make sure the tnsnames.ora entries on application server are pointing to the correct environment for Data Marts as well as Back Office.
9. Recompile schema OBD_OWN, after which make sure no invalid objects are found in schema OBD_OWN:

```
begin
dbms_utility.compile_schema( schema  => user , compile_all   =>TRUE,  reuse_settings =>TRUE);
end;
select count(*) from user_objects where status != 'VALID';
```

10. Create wallet entries on Application server for 3 users, and display mkstore is similar to below on the application server.

```
mkstore -wrl . -listCredential
<db> batch
```

```
<db>_batch batch
<db>_install obd_own
```

Also make sure there is a tnsnames entry for

```
<db>
<db>_batch
<db>_install
```

To check for multi instance rac node (setup 1, non preferred setup):

```
mkstore –wrl . –listCredential
oton1 batch
oton1_batch batch
oton1_install obd_own
```

where oton1 is the service_name which is used for extraction

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

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

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

where oton is the service_name which is used for extraction

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

11. Make sure the input files of the External tables, are in the OBD_INPUT location, if not place them in OBD_INPUT directory and make sure oracle has read, write permissions on that directory.
12. Make sure there is a batch scheduler running on the application server.

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

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

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

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

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

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

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

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

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

For DOP to work properly IO Calibration is needed. This can be achieved by executing the one time only process DBMS_RESOURCE_MANAGER.CALIBRATE_IO, failure to do so would lead to the message in the explain plan:

"*automatic DOP: skipped because of IO calibrate statistics are missing*"

The example below is derived from the documentation:
http://docs.oracle.com/cd/E11882_01/appdev.112/e10577/d_resmgr.htm#CJGHGFEA

Example of using I/O Calibration procedure

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

APPENDIX C: ACTIVE DATA GUARD

OHI Data Marts supports an environment within which Active Data Guard is being used. OHI Data Marts has been tested and certified against a physical standby database in read-only mode.

The process of creating a Physical Standby database is described in detail in chapter “3 Creating a Physical Standby Database” of the document “Oracle® Data Guard Concepts and Administration, 11g Release 2 (11.2)”, part number E41134-02.

More information on managing Archive Redo Logs can be read in chapter 13 “Managing Archived Redo Logs” of the document “Oracle® Database Administrator's Guide, 11g Release 2 (11.2)”, part number E25494-04.

APPENDIX D: ORACLE ACCESS MANAGER / ORACLE INDENTITY MANAGER

Oracle Business Intelligence Enterprise Edition (OBIEE) is supported in combination with Oracle Access Manager and Oracle Identity Manager. For the supported version for each of the products please review the certification matrix available on the page ‘Oracle Fusion Middleware Supported System Configurations’ (http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html).

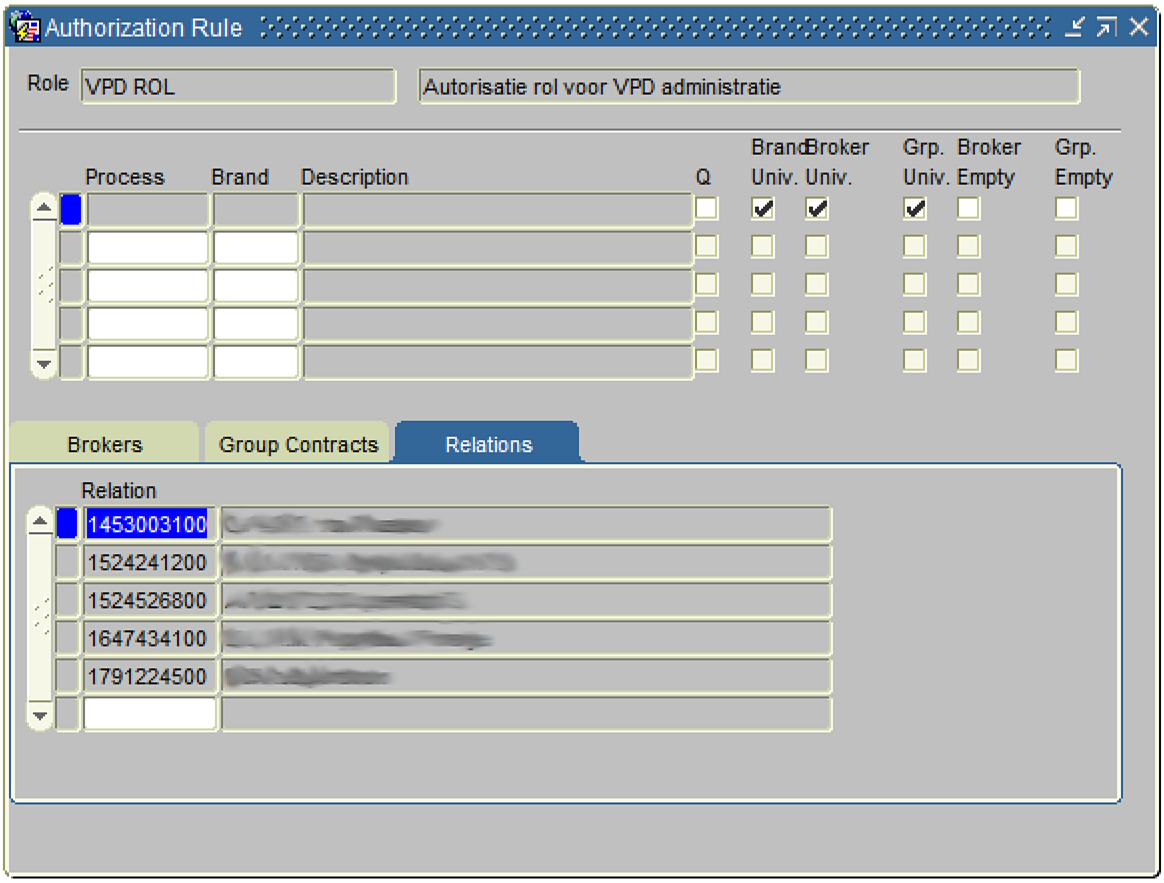
To setup OBIEE in combination with Oracle Access Manager 11g please review the Oracle Student Learning Installation and Deployment Guide, part number E20664-04. This guide provides detailed information on the installation and configuration of Oracle Access Manager 11g, as well as information on how to setup OAM as the SSO solution for OBIEE.

For detailed information on the Oracle Identity Managent Suite, please review the document Oracle Fusion Middleware Integration Overview for Oracle Identity Management Suite 11g Release 1 (11.1.1), part number E15477-03.

APPENDIX E: VIRTUAL PRIVATE DATABASE

Introduction

To make use of the feature Virtual Private Database (VPD) for masking privacy sensitive information one can specify Authorization Rules within OHI Back Office using the 'Authorization Rule'-form (ZRG7093F). These roles can be assigned to users/groups.



Role: VPD ROL — Autorisatie rol voor VPD administratie

Process	Brand	Description	Q	Brand Univ.	Broker Univ.	Grp. Univ.	Broker Empty	Grp. Empty
			□	☑	☑	☑	□	□
			□	□	□	□	□	□
			□	□	□	□	□	□
			□	□	□	□	□	□
			□	□	□	□	□	□

Relation	
1453003100	[redacted]
1524241200	[redacted]
1524526800	[redacted]
1647434100	[redacted]
1791224500	[redacted]

These rules are used within both OHI Back Office and OHI Data Marts to mask privacy sensitive information.

When logging into OBIEE the user-account used for OBIEE will be used to determine the authorization as is defined in OHI Back Office for the same user.

OBIEE Users

Within Weblogic Console users are to be defined with the exact same user name as they have been defined within OHI Back Office. These user need to belong at minimum to the group 'BIConsumers'.
See 'Introduction to Security in Oracle Business Intelligence' for more information the Oracle Business Intelligence security model.

When a user logs in to the OBIEE environment a procedure is called to identify the functional user (i.e. the username used to log into OBIEE). This user will then be used to evaluate the authorization rules against.

When there is a need to also apply VPD outside of the OBIEE (i.e. for users directly connecting to the database themselves using a database account) a LOGON trigger should be defined to ensure that the VPD policies also know for what user the authorization rules should be evaluated. The following is an example for a LOGON trigger.

```
create or replace trigger dwh_vpd_init_user
after logon
on database
enable
declare
  l_vpd_user stg_gebruiker_rollen.fun_oracle_user%type;
begin
  l_vpd_user := dwh_vpd_pck.set_user( USER );
end;
/
```

Make sure the user has been granted the OBD_ROL_SELECT and has not been granted ‘`exempt access policy`’.

Refreshing Authorizationrules

To be able to make use of VPD within the OHI Data Marts environment the authorization rules batch \`ZRGO023S\` (Refreshing Authorizationrules) needs to be executed. This batch load information about the authorization rules defined within OHI Back Office into the OHI Data Marts environment.
This batch can also to be scheduled using the menu options ‘Data Marts’ ➔ ‘Refreshing Authorizationrules’.

VPD Policies

For the following OHI Data Marts tables/columns VPD policies are defined

Data Marts table	**Privacy sensitive columns**
DWH_ADRESSEN	ADS_AANVULLENDE_ADRESREGEL_1, ADS_AANVULLENDE_ADRESREGEL_2, ADS_AANVULLENDE_ADRESREGEL_3, ADS_AANVULLENDE_ADRESREGEL_4, ADS_AANVULLENDE_ADRESREGEL_5, ADS_AANVULLENDE_ADRESREGEL_6, ADS_AANVULLENDE_ADRESREGEL_7, ADS_AANVULLENDE_ADRESREGEL_8, ADS_AANVULLENDE_ADRESREGEL_9, ADS_HUISNUMMER, ADS_HUIS_NR, ADS_HUIS_NR_TOEVOEGING, ADS_LAND_CODE, ADS_LAND_OMSCHRIJVING, ADS_POSTCODE, ADS_POSTCODE_LETTER, ADS_POSTCODE_NR, ADS_STRAATNAAM, ADS_WOONPLAATS
DWH_COLLECTIEVE_CONTRACTEN	CCT_AANVULLENDE_ADRESREGEL, CCT_AANVULLENDE_ADRESREGEL_2, CCT_AANVULLENDE_ADRESREGEL_3, CCT_AANVULLENDE_ADRESREGEL_4, CCT_AANVULLENDE_ADRESREGEL_5, CCT_AANVULLENDE_ADRESREGEL_6, CCT_AANVULLENDE_ADRESREGEL_7, CCT_AANVULLENDE_ADRESREGEL_8, CCT_AANVULLENDE_ADRESREGEL_9, CCT_HUISNUMMER, CCT_HUIS_NR, CCT_HUIS_NR_TOEVOEGING, CCT_LAND_CODE, CCT_LAND_OMSCHRIJVING, CCT_NAAM, CCT_OMSCHRIJVING, CCT_POSTCODE, CCT_POSTCODE_LETTER, CCT_POSTCODE_NR, CCT_STRAATNAAM, CCT_WOONPLAATS
DWH_CREDITEUREN	CDR_AANVULLENDE_ADRESREGEL, CDR_AANVULLENDE_ADRESREGEL_2, CDR_AANVULLENDE_ADRESREGEL_3, CDR_AANVULLENDE_ADRESREGEL_4, CDR_AANVULLENDE_ADRESREGEL_5, CDR_AANVULLENDE_ADRESREGEL_6, CDR_AANVULLENDE_ADRESREGEL_7, CDR_AANVULLENDE_ADRESREGEL_8, CDR_AANVULLENDE_ADRESREGEL_9, CDR_HUISNUMMER, CDR_HUIS_NR, CDR_HUIS_NR_TOEVOEGING, CDR_LAND_CODE, CDR_LAND_OMSCHRIJVING, CDR_NAAM, CDR_OMSCHRIJVING, CDR_POSTCODE, CDR_POSTCODE_LETTER, CDR_POSTCODE_NR, CDR_P_AANVULLENDE_ADRESREGEL, CDR_P_AANVULLENDE_ADRESREGEL_2, CDR_P_AANVULLENDE_ADRESREGEL_3, CDR_P_AANVULLENDE_ADRESREGEL_4, CDR_P_AANVULLENDE_ADRESREGEL_5, CDR_P_AANVULLENDE_ADRESREGEL_6, CDR_P_AANVULLENDE_ADRESREGEL_7, CDR_P_AANVULLENDE_ADRESREGEL_8, CDR_P_AANVULLENDE_ADRESREGEL_9, CDR_P_HUISNUMMER, CDR_P_HUIS_NR, CDR_P_HUIS_NR_TOEVOEGING, CDR_P_LAND_CODE, CDR_P_LAND_OMSCHRIJVING, CDR_P_POSTCODE, CDR_P_POSTCODE_LETTER, CDR_P_POSTCODE_NR, CDR_P_STRAATNAAM, CDR_P_WOONPLAATS, CDR_STRAATNAAM, CDR_WOONPLAATS

Data Marts table	Privacy sensitive columns
DWH_DEBITEUREN	DBR_AANVULLENDE_ADRESREGEL, DBR_AANVULLENDE_ADRESREGEL_2, DBR_AANVULLENDE_ADRESREGEL_3, DBR_AANVULLENDE_ADRESREGEL_4, DBR_AANVULLENDE_ADRESREGEL_5, DBR_AANVULLENDE_ADRESREGEL_6, DBR_AANVULLENDE_ADRESREGEL_7, DBR_AANVULLENDE_ADRESREGEL_8, DBR_AANVULLENDE_ADRESREGEL_9, DBR_HUISNUMMER, DBR_HUIS_NR, DBR_HUIS_NR_TOEVOEGING, DBR_LAND_CODE, DBR_LAND_OMSCHRIJVING, DBR_NAAM, DBR_OMSCHRIJVING, DBR_POSTCODE, DBR_POSTCODE_LETTER, DBR_POSTCODE_NR, DBR_P_AANVULLENDE_ADRESREGEL, DBR_P_AANVULLENDE_ADRESREGEL_2, DBR_P_AANVULLENDE_ADRESREGEL_3, DBR_P_AANVULLENDE_ADRESREGEL_4, DBR_P_AANVULLENDE_ADRESREGEL_5, DBR_P_AANVULLENDE_ADRESREGEL_6, DBR_P_AANVULLENDE_ADRESREGEL_7, DBR_P_AANVULLENDE_ADRESREGEL_8, DBR_P_AANVULLENDE_ADRESREGEL_9, DBR_P_HUISNUMMER, DBR_P_HUIS_NR, DBR_P_HUIS_NR_TOEVOEGING, DBR_P_LAND_CODE, DBR_P_LAND_OMSCHRIJVING, DBR_P_POSTCODE, DBR_P_POSTCODE_LETTER, DBR_P_POSTCODE_NR, DBR_P_STRAATNAAM, DBR_P_WOONPLAATS, DBR_STRAATNAAM, DBR_WOONPLAATS
DWH_DECLARANTEN	DCT_AANVULLENDE_ADRESREGEL, DCT_AANVULLENDE_ADRESREGEL_2, DCT_AANVULLENDE_ADRESREGEL_3, DCT_AANVULLENDE_ADRESREGEL_4, DCT_AANVULLENDE_ADRESREGEL_5, DCT_AANVULLENDE_ADRESREGEL_6, DCT_AANVULLENDE_ADRESREGEL_7, DCT_AANVULLENDE_ADRESREGEL_8, DCT_AANVULLENDE_ADRESREGEL_9, DCT_HUISNUMMER, DCT_HUIS_NR, DCT_HUIS_NR_TOEVOEGING, DCT_IND_UNIVERSEEL, DCT_LAND_CODE, DCT_LAND_OMSCHRIJVING, DCT_OMSCHRIJVING, DCT_POSTCODE, DCT_POSTCODE_LETTER, DCT_POSTCODE_NR, DCT_P_AANVULLENDE_ADRESREGEL, DCT_P_AANVULLENDE_ADRESREGEL_2, DCT_P_AANVULLENDE_ADRESREGEL_3, DCT_P_AANVULLENDE_ADRESREGEL_4, DCT_P_AANVULLENDE_ADRESREGEL_5, DCT_P_AANVULLENDE_ADRESREGEL_6, DCT_P_AANVULLENDE_ADRESREGEL_7, DCT_P_AANVULLENDE_ADRESREGEL_8, DCT_P_AANVULLENDE_ADRESREGEL_9, DCT_P_HUISNUMMER, DCT_P_HUIS_NR, DCT_P_HUIS_NR_TOEVOEGING, DCT_P_LAND_CODE, DCT_P_LAND_OMSCHRIJVING, DCT_P_POSTCODE, DCT_P_POSTCODE_LETTER, DCT_P_POSTCODE_NR, DCT_P_STRAATNAAM, DCT_P_WOONPLAATS, DCT_STRAATNAAM, DCT_WOONPLAATS
DWH_DECLARATIES	AW_BSN_GEINDICEERDE_CLIENT, AW_BSN_PARTNER, AW_BSN_VERZEKERDE, AW_RELATIECODERING, HUISNUMMER, LANDCODE_BRON, PATIENTNUMMER, POSTCODE_BRON, POSTCODE_LETTER_BRON, POSTCODE_NR_BRON, REKENINGGEGEVEN, RT_LANDCODE, WOONPLAATS_BRON
DWH_PREMIES	LAN_CODE_BRON, POSTCODE_BRON, POSTCODE_LETTER_BRON, POSTCODE_NR_BRON, WOONPLAATS_BRON
DWH_TUSSENPERSONEN	TUP_AANVULLENDE_ADRESREGEL,TUP_AANVULLENDE_ADRESREGEL_2,TUP_AANVULLENDE_ADRESREGEL_3,TUP_AANVULLENDE_ADRESREGEL_4,TUP_AANVULLENDE_ADRESREGEL_5,TUP_AANVULLENDE_ADRESREGEL_6,TUP_AANVULLENDE_ADRESREGEL_7,TUP_AANVULLENDE_ADRESREGEL_8,TUP_AANVULLENDE_ADRESREGEL_9,TUP_HUISNUMMER,TUP_HUIS_NR,TUP_HUIS_NR_TOEVOEGING,TUP_IND_ABZ_BERICHT,TUP_LAND_CODE,TUP_LAND_OMSCHRIJVING,TUP_OMSCHRIJVING,TUP_POSTCODE,TUP_POSTCODE_LETTER,TUP_POSTCODE_NR,TUP_STRAATNAAM,TUP_WOONPLAATS
DWH_VERZEKERDEN	EXCASSOREKENING_DECLARATIES, HOEDANIGHEID, HUISNUMMER, INCASSOREKENING_DECLARATIES, LAN_CODE_BRON, PCE_LETTER_BRON, POSTCODE_BRON, REKENINGGEGEVEN, WOONPLAATS_BRON, WTZ_GROEP_CODE
DWH_ZORG_CLIENTEN	ZCL_AANVULLENDE_ADRESREGEL, ZCL_AANVULLENDE_ADRESREGEL_2, ZCL_AANVULLENDE_ADRESREGEL_3, ZCL_AANVULLENDE_ADRESREGEL_4, ZCL_AANVULLENDE_ADRESREGEL_5, ZCL_AANVULLENDE_ADRESREGEL_6, ZCL_AANVULLENDE_ADRESREGEL_7, ZCL_AANVULLENDE_ADRESREGEL_8, ZCL_AANVULLENDE_ADRESREGEL_9, ZCL_ACHTERNAAM, ZCL_A_NUMMER, ZCL_EMAIL_ADRES, ZCL_HUISNUMMER, ZCL_HUIS_NR, ZCL_HUIS_NR_TOEVOEGING, ZCL_LAND_CODE, ZCL_LAND_OMSCHRIJVING, ZCL_NAAM_NAAMAANDUIDING, ZCL_NAT_CODE, ZCL_NAT_OMS, ZCL_PAR_ACHTERNAAM, ZCL_PAR_VOORVOEGSEL, ZCL_POSTCODE, ZCL_POSTCODE_LETTER, ZCL_POSTCODE_NR, ZCL_P_AANVULLENDE_ADRESREGEL, ZCL_P_AANVULLENDE_ADRESREGEL_2, ZCL_P_AANVULLENDE_ADRESREGEL_3, ZCL_P_AANVULLENDE_ADRESREGEL_4, ZCL_P_AANVULLENDE_ADRESREGEL_5, ZCL_P_AANVULLENDE_ADRESREGEL_6, ZCL_P_AANVULLENDE_ADRESREGEL_7, ZCL_P_AANVULLENDE_ADRESREGEL_8, ZCL_P_AANVULLENDE_ADRESREGEL_9, ZCL_P_HUISNUMMER, ZCL_P_HUIS_NR, ZCL_P_HUIS_NR_TOEVOEGING, ZCL_P_LAND_CODE, ZCL_P_LAND_OMSCHRIJVING, ZCL_P_POSTCODE, ZCL_P_POSTCODE_LETTER, ZCL_P_POSTCODE_NR, ZCL_P_STRAATNAAM, ZCL_P_WOONPLAATS, ZCL_SOFI_NR, ZCL_STRAATNAAM, ZCL_TELEFOON_NR, ZCL_TITULATUUR, ZCL_VOORLETTERS, ZCL_VOORNAMEN, ZCL_VOORVOEGSELS, ZCL_WOONPLAATS, ZCL_ZCT_DERDEN_CODERING_1, ZCL_ZCT_DERDEN_CODERING_2, ZCL_ZCT_DERDEN_CODERING_3, ZCL_ZCT_DERDEN_CODERING_4, ZCL_ZCT_DERDEN_CODERING_5

Data Marts table	Privacy sensitive columns
DWH_ZORG_RELATIES	KOE_CODE, KOE_OMSCHRIJVING, ZRE_AANVULLENDE_ADRESREGEL, ZRE_AANVULLENDE_ADRESREGEL_2, ZRE_AANVULLENDE_ADRESREGEL_3, ZRE_AANVULLENDE_ADRESREGEL_4, ZRE_AANVULLENDE_ADRESREGEL_5, ZRE_AANVULLENDE_ADRESREGEL_6, ZRE_AANVULLENDE_ADRESREGEL_7, ZRE_AANVULLENDE_ADRESREGEL_8, ZRE_AANVULLENDE_ADRESREGEL_9, ZRE_CTG_NR, ZRE_EXTERN_KENMERK, ZRE_HAGRO_OMSCHRIJVING, ZRE_HUISNUMMER, ZRE_HUIS_NR, ZRE_HUIS_NR_TOEVOEGING, ZRE_LAND_CODE, ZRE_LAND_OMSCHRIJVING, ZRE_POSTCODE, ZRE_POSTCODE_LETTER, ZRE_POSTCODE_NR, ZRE_P_AANVULLENDE_ADRESREGEL, ZRE_P_AANVULLENDE_ADRESREGEL_2, ZRE_P_AANVULLENDE_ADRESREGEL_3, ZRE_P_AANVULLENDE_ADRESREGEL_4, ZRE_P_AANVULLENDE_ADRESREGEL_5, ZRE_P_AANVULLENDE_ADRESREGEL_6, ZRE_P_AANVULLENDE_ADRESREGEL_7, ZRE_P_AANVULLENDE_ADRESREGEL_8, ZRE_P_AANVULLENDE_ADRESREGEL_9, ZRE_P_HUISNUMMER, ZRE_P_HUIS_NR, ZRE_P_HUIS_NR_TOEVOEGING, ZRE_P_LAND_CODE, ZRE_P_LAND_OMSCHRIJVING, ZRE_P_POSTCODE, ZRE_P_POSTCODE_LETTER, ZRE_P_POSTCODE_NR, ZRE_P_STRAATNAAM, ZRE_P_WOONPLAATS, ZRE_REL_OMSCHRIJVING, ZRE_STRAATNAAM, ZRE_WOONPLAATS, ZRE_ZCL_CODE, ZRE_ZCL_OMSCHRIJVING, ZRE_ZKR_CODE
DWH_ZORG_WERKGEVERS	WGR_AANVULLENDE_ADRESREGEL, WGR_AANVULLENDE_ADRESREGEL_2, WGR_AANVULLENDE_ADRESREGEL_3, WGR_AANVULLENDE_ADRESREGEL_4, WGR_AANVULLENDE_ADRESREGEL_5, WGR_AANVULLENDE_ADRESREGEL_6, WGR_AANVULLENDE_ADRESREGEL_7, WGR_AANVULLENDE_ADRESREGEL_8, WGR_AANVULLENDE_ADRESREGEL_9, WGR_HUISNUMMER, WGR_HUIS_NR, WGR_HUIS_NR_TOEVOEGING, WGR_LAND_CODE, WGR_LAND_OMSCHRIJVING, WGR_OMSCHRIJVING, WGR_POSTCODE, WGR_POSTCODE_LETTER, WGR_POSTCODE_NR, WGR_STRAATNAAM, WGR_TELEFOON_NR, WGR_WOONPLAATS

Using step 880 (Enable/disable VPD) of OHIPATCH.pl the VPD policies can be enabled (Policy group ‘OHI_RELATIONS’, policy ‘REL_ANONIEM’).

Adding custom policies

If there is a need to add additional VPD policies that are not offered within OHI Data Marts, one could create additional policies keeping in mind the following rules

- Do not add custom policies to the policy group 'OHI_RELATIONS' and/or policy name 'REL_ANONIEM'. These are used and reserved for OHI Data Marts
- Custom policies should not negatively impact performance
- Custom policies are only allowed to mask columns and should never remove entire rows (i.e. always specify what columns should be masked with the table or tables for which the custom policy is defined using the sec_relevant_cols option). Hiding/removing rows will result incomplete data when reporting on OHI Data Marts.
- Only SELECT policies are allowed
- Do not apply masking on columns that are most likely to be used within predicates (i.e. foreign key relation columns)

Example of adding a custom policy for DWH_MERKEN

In this example a custom policy is created for the dimension DWH_MERKEN within which data for the columns AOR_OMSCHRIJVING, MRK_OMSCHRIJVING, AGB_OMSCHRIJVING and ZVR_NAAM will be masked (for all users) when data related to `FINPR`, `PRE` or `ZFWCD` is selected.

Please consult the Oracle Database Security Guide for additional information on the topic ‘Using Oracle Virtual Private Database to Control Data Access’ at https://docs.oracle.com/database/121/DBSEG/vpd.htm#DBSEG007.

Creating new policy group (CUSTOM_VPD_GROUP)

```
begin
  dbms_rls.enable_grouped_policy
  ( object_name => 'DWH_MERKEN'
  , group_name  => 'CUSTOM_VPD_GROUP'
  , policy_name => 'CUSTOM_VPD_POLICY'
  , enable      => true
  );
end;
/
```

Creating new policy (CUSTOM_VPD_POLICY) within policy group

Thefollwing statement creates a SELECT policy for the table DWH_MERKEN for which data for the columns AOR_OMSCHRIJVING, MRK_OMSCHRIJVING, AGB_OMSCHRIJVING and ZVR_NAAM is to be masked.

The PL/SQL package/function DWH_CUSTOM_VPD_PCK.ADD_PREDICATE_SEL is specified as the function which generates the predicate for this policy.

```
begin
  dbms_rls.add_grouped_policy
  ( object_name           => 'DWH_MERKEN'
  , policy_group          => 'CUSTOM_VPD_GROUP'
  , policy_name           => 'CUSTOM_VPD_POLICY'
  , policy_function       => 'DWH_CUSTOM_VPD_PCK.ADD_PREDICATE_SEL'
  , statement_types       => 'SELECT'
  , enable                => false
  , sec_relevant_cols     =>
'AOR_OMSCHRIJVING,MRK_OMSCHRIJVING,AGB_OMSCHRIJVING,ZVR_NAAM'
  , sec_relevant_cols_opt => dbms_rls.ALL_ROWS
  );
end;
/
```

PL/SQL package

The PL/SQL package that is linked to the VPD policy will be executed for every row/column returned from a query that (in this example) uses the table DWH_MERKEN, by calling the function add_predicate_sel.

The function add_predicate_sel first checks for which table it is being called, this way this package can be used for more then just the DWH_MERKEN table, but could also be used for other tables containing information related to DWH_MERKEN. If this function returns condition which evaluates to **False** access is granted and the data will be masked.

The function check_mrk performs the actual check whether or not access is allowed.

```
create or replace package dwh_custom_vpd_pck
is

  function add_predicate_sel
  ( pi_schema_name in varchar2
  , pi_object_name in varchar2
  ) return varchar2
  ;

end dwh_custom_vpd_pck;
/
```

```
create or replace package body dwh_custom_vpd_pck
is

  function check_mrk
  ( pi_mrk_code in dwh_merken.mrk_code%type
  ) return varchar2
  is
    l_retval varchar2(1) := 'N'; -- By default no access
  begin
    if pi_mrk_code in ('FINPR', 'PRE', 'ZFWCD')
    then
      l_retval := 'J'; -- Allow access
    end if;
    return l_retval;
  end;

  function add_predicate_sel
  ( pi_schema_name in varchar2
  , pi_object_name in varchar2
  ) return varchar2
  is
    l_retval    varchar2(4000) := ' 1=1 ';
    l_tablename varchar2(60);
  begin
    l_tablename := upper(pi_object_name);
    if l_tablename = 'DWH_MERKEN'
    then
      l_retval := ' (dwh_custom_vpd_pck.check_mrk(MRK_CODE) = ''J'') ';
    end if;
    return l_retval;
  end add_predicate_sel;

end dwh_custom_vpd_pck;
/
```

Ensure that the role OBD_ROL_SELECT has been granted execute privileges on the package DWH_CUSTOM_VPD_PCK and a public synonym has been created.

```
grant execute on obd_own.dwh_custom_vpd_pck to OBD_ROL_SELECT;
create public synonym dwh_custom_vpd_pck for obd_own.dwh_custom_vpd_pck;
```

Enabling the policy

By default, when adding a policy the policy is not enabled. Using the following statement, the policy on DWH_MERKEN is activated.

```
begin
  dbms_rls.enable_grouped_policy
  ( object_name => 'DWH_MERKEN'
  , group_name  => 'CUSTOM_VPD_GROUP'
  , policy_name => 'CUSTOM_VPD_POLICY'
  , enable      => true
  );
end;
/
```

For this moment on any query on DWH_MERKEN each for within DWH_MERKEN will be evaluated and data will be masked accordingly.