

Oracle Health Insurance Data Marts

Administrator Reference

version 9.64

Part number: G24367-01

Januari 29, 2025

Copyright © 2011, 2025 Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

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

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

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

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

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

CHANGE HISTORY

Release	Version	Changes
10.12.2.0.0	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 different 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 properties 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
10.20.3.0.0	9.51	Removed LEEFTIJD_CATEGORIEEN.dat and POSTCODE_VERRIJKINGEN.dat.
10.20.6.0.0	9.52	- M-6241: Database Vault implementation for Data Marts enhanced. Reference to CTA13508 Oracle Health Insurance Back Office – Installation, Configuration and DBA Manual added. - M-6235: Improve error-handling Data Marts ETL batches
10.20.8.0.0	9.53	Grants on OBD_SELECT_USER changes: 'select on dba_tables' and 'select on dba_tab_statistics' added, 'select on sys.v_\$database' removed.
10.21.1.0.0	9.54	New part number and copyright year, no changes.
10.21.2.0.0	9.55	Definition of authorization roles OBD_ROL_ADMIN, OBD_ROL_SELECT and OBD_ETL_ROL changed.

Release	Version	Changes
10.21.4.0.0	9.56	- Reference to 'CTA13685.doc: Oracle Health Insurance Back Office - Security Guide' added. Paragraph 'Authorization' moved to this document - Paragraphs 'Default Secure Install' and 'Enable Security Audit' removed. - References to 'Oracle Warehouse Builder' removed - Parameters FINMOD and INDDCA removed - Description of external file WBX_OBIEE_TRANSLATIONS.csv removed - Obsolete references to OWB removed - View CTR_DECL_DETAILS_VW modified
10.21.6.0.0	9.57	Updated Compression; removed part on table reorganisation.
10.22.2.0.0	9.58	Removed old document codes from document references and replaced them with a reference to docs.oracle.com
10.22.3.0.0	9.59	Overview of views in 'Results of the compatibility checks' transferred to Online Help
10.23.1.0.0	9.60	No changes, republished with a new part number.
10.23.8.0.0	9.61	Role OBD_SELECT_ROLE removed and grants for OBD_SELECT_USER revisited
10.24.1.0.0	9.62	No changes, republished with new part number.
10.25.1.0.0	9.63	No changes, republished with new part number.
10.25.2.0.0	9.64	Added paragraph NOLOGGING clause of fact and dimension tables

CONTENTS

Change History	ii
Introduction	6
Important documentation	6
Software used	6
Overview	7
Part I - Installation	8
Introduction	8
Setting up Oracle Health Insurance Data Marts environment	8
OS set-up	8
Software installation	8
Database	8
Creation of an Oracle Health Insurance Data Marts database	8
Parameters	8
Tablespaces	9
Database users	10
Database link	10
Database directories	11
General instructions	11
Generation and installation of Oracle Health Insurance Data Marts objects	12
Installation	12
Release	12
System parameters	12
Set-up of Oracle Business Intelligence Enterprise Edition environment	14
Software installation	14
Oracle Business Intelligence Customizations	14
Creating a NON- PRIVILEGED user: OBI_SELECT_USER	14
Installation of Oracle Health Insurance Data Marts Repository (RPD)	14
Configure Usage Tracking	17
Upgrading Oracle Business Intelligence repository	18
Part II - Application management	22
Introduction	22
Functional management	22
Logging and validation	22
Logging of loading	22
WBX logging	23
Logging load scripts	24
Configuration of generic structure for claim properties	25
Introduction	25
Examples	25
When to place a claim property in a dimension table	28
Generate views to create a functional overlay over the generic structure	29

Add claim properties from generic structure to OBI EE repository	29
Configuration of generic structure for dimension properties	31
Introduction	31
Examples	31
Generate views to create a functional overlay over the generic structure	31
Add dimension properties from generic structure to OBI EE repository	32
Part III - Technical management	33
New releases of Oracle Health Insurance Data Marts	33
Compression of partitioned fact tables	33
NOLOGGING clause of fact and dimension tables	34
Part IV – Set-up of Database Vault for Data Marts	35
Part V - Appendices	36
Appendix A: Cloning Oracle Health Insurance Data Marts environments	36
Appendix B: Exadata Usage	36
Example of using I/O Calibration procedure	38
Appendix C: Active Data Guard	39
Appendix D: Oracle Access Manager / Oracle Identity Manager	40
Appendix E: Virtual Private Database	41
Introduction	41
OBIEE Users	41
Refreshing Authorizationrules	42
VPD Policies	42
Adding custom policies	44
Example of adding a custom policy for DWH_MERKEN	44

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. Oracle Health Insurance Installation, Configuration and DBA Manual (docs.oracle.com)
3. Oracle Health Insurance Back Office - Security Guide (docs.oracle.com)

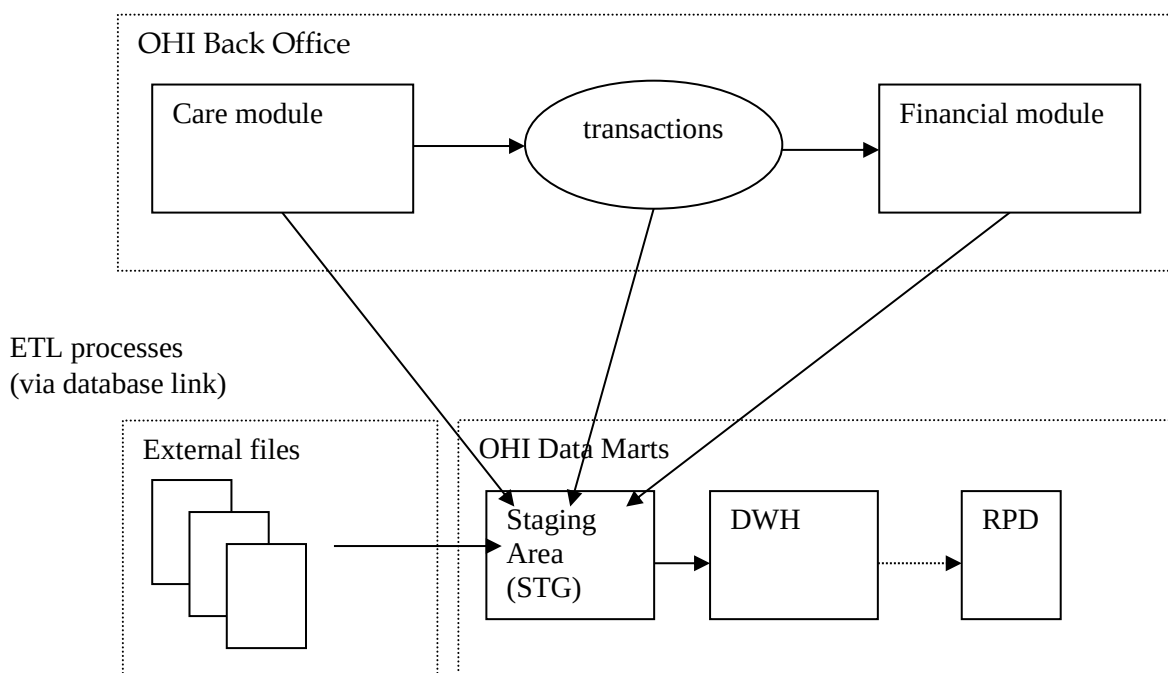
SOFTWARE USED



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

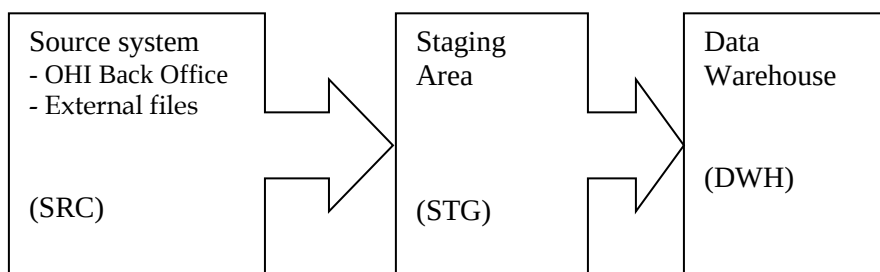
OVERVIEW

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.

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



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

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 Oracle Health Insurance Installation, Configuration and DBA Manual (docs.oracle.com).

Tablespaces

The tablespaces below must be created for data and indexes:

Tablespace
STG_STAD
STG_STAI
STG_DYND
STG_DYNI
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

Tablespace
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_OWEN	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)

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 CREATE VIEW SELECT ON DBA_TABLES SELECT ON DBA_TAB_STATISTICS

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_OWEN) 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_OWNS) 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_OWNS 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.
```

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
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 = ','

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	Claims from date
FINVDM	Financials from date
OHWVDM	Work in Progress Claims from date
PREVDM	Premiums from date
VBSVDM	Commitments from date
VZEVDM	Policies from date
ZVNVDM	Care authorization from date

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.

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-BI-11G-176387.PDF](http://www.oracle.com/technetwork/middleware/bi/customizing-oracle-bi-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 privileges 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 masked).

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.

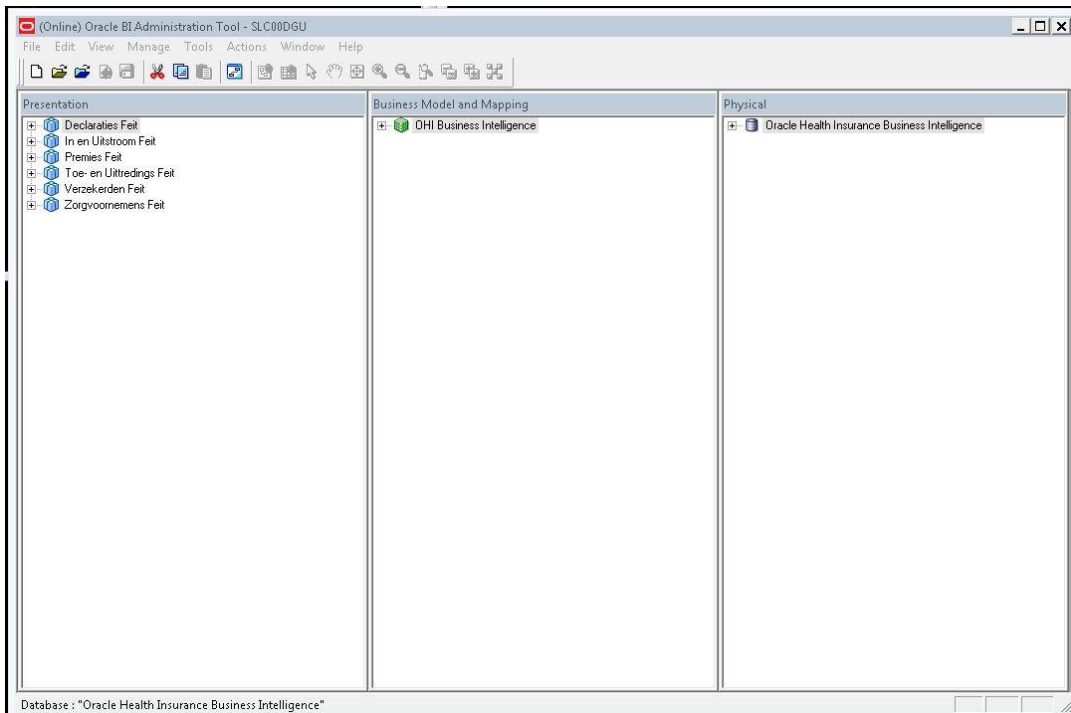
Before being able to use the DWH_OHI_BI.rpd file in your OBIEE environment, you will have to configure 3 connection pools using the Oracle BI Administration Tool.

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

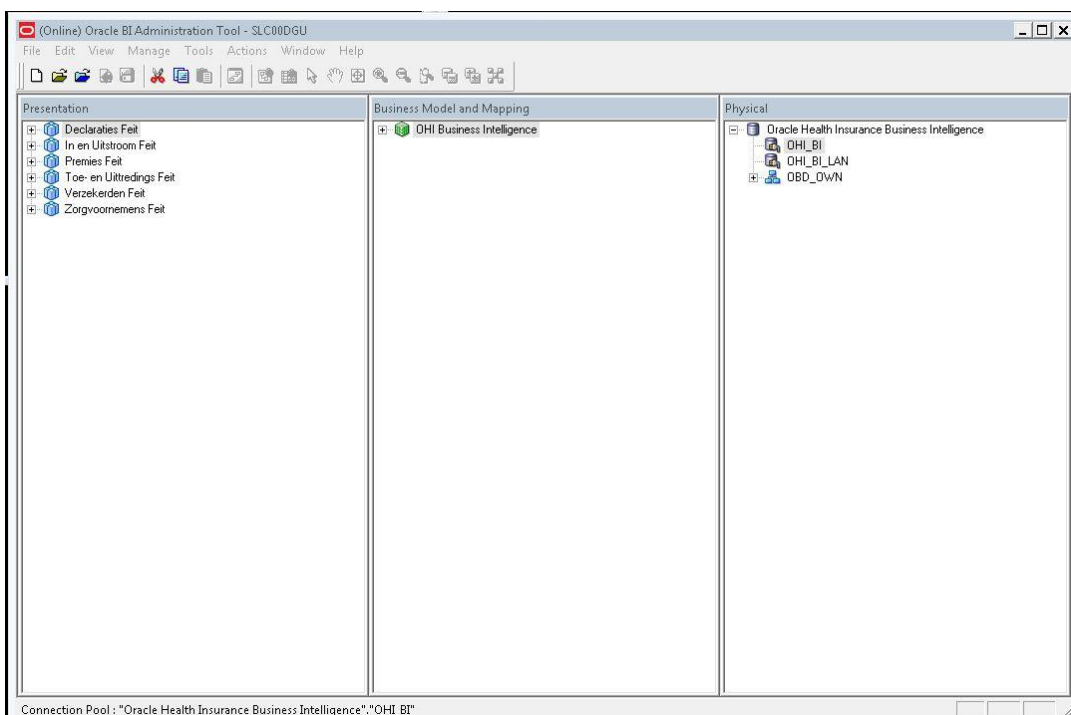
Open the DWH_OHI_BI.rpd offline using the Oracle BI Administration Tool:



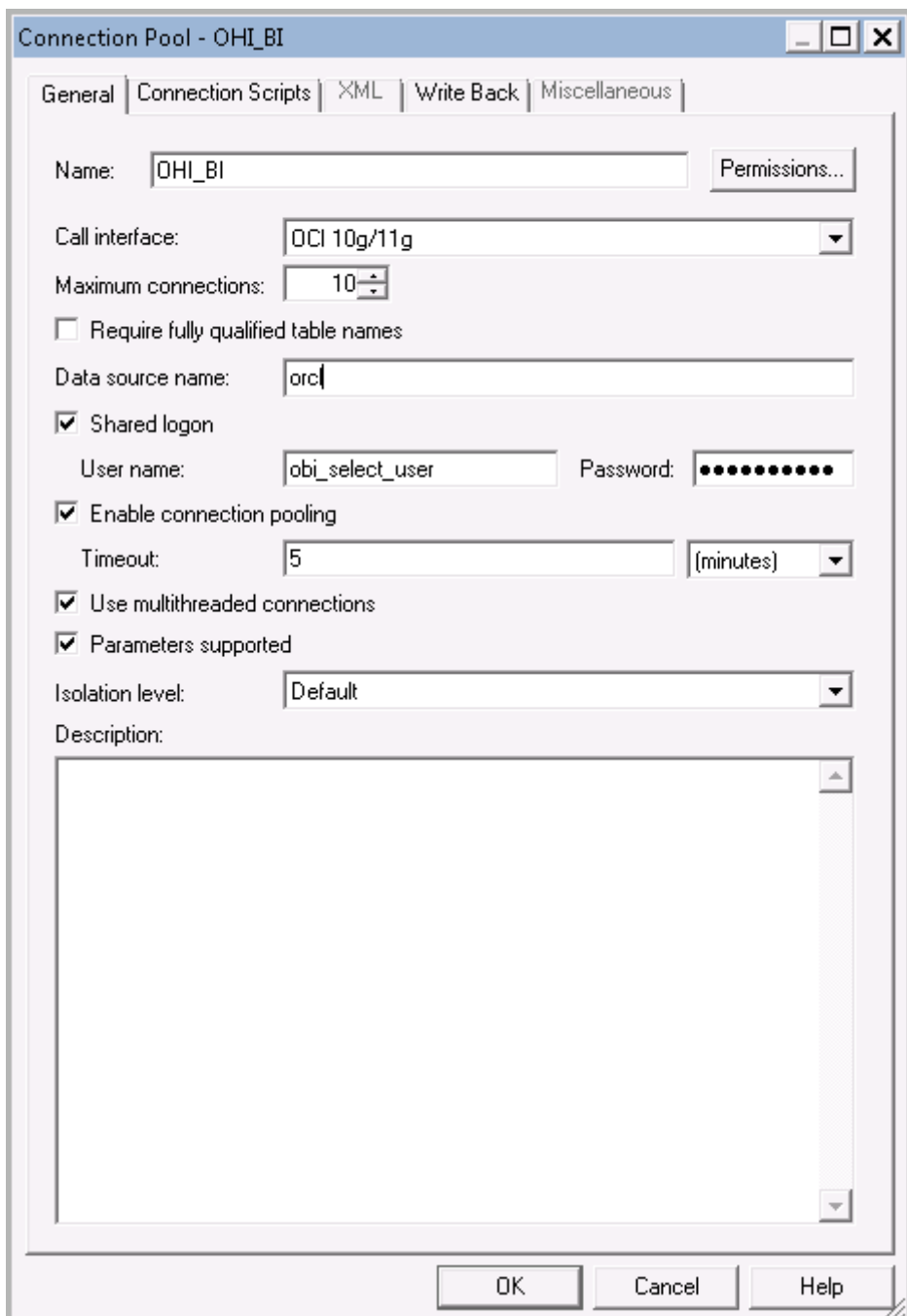
Enter adm1n1strator as the initial password.



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



Open the OHI_BI connection pool by double clicking OHI_BI.



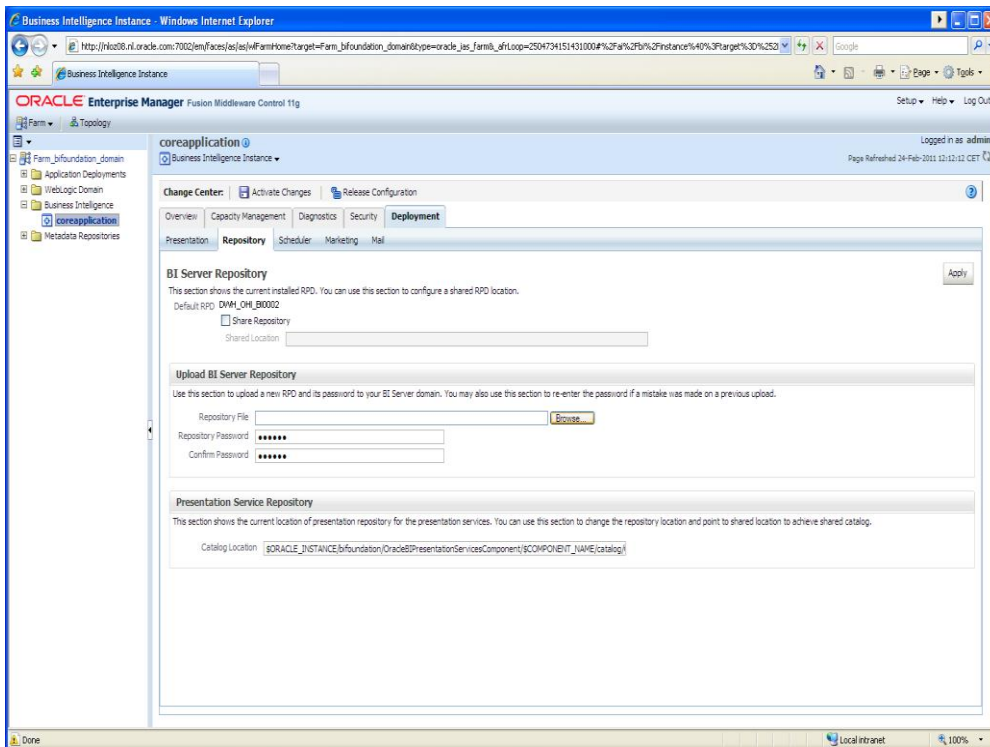
Change the data source name to your OHI Data Marts database.

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.



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 referred 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:

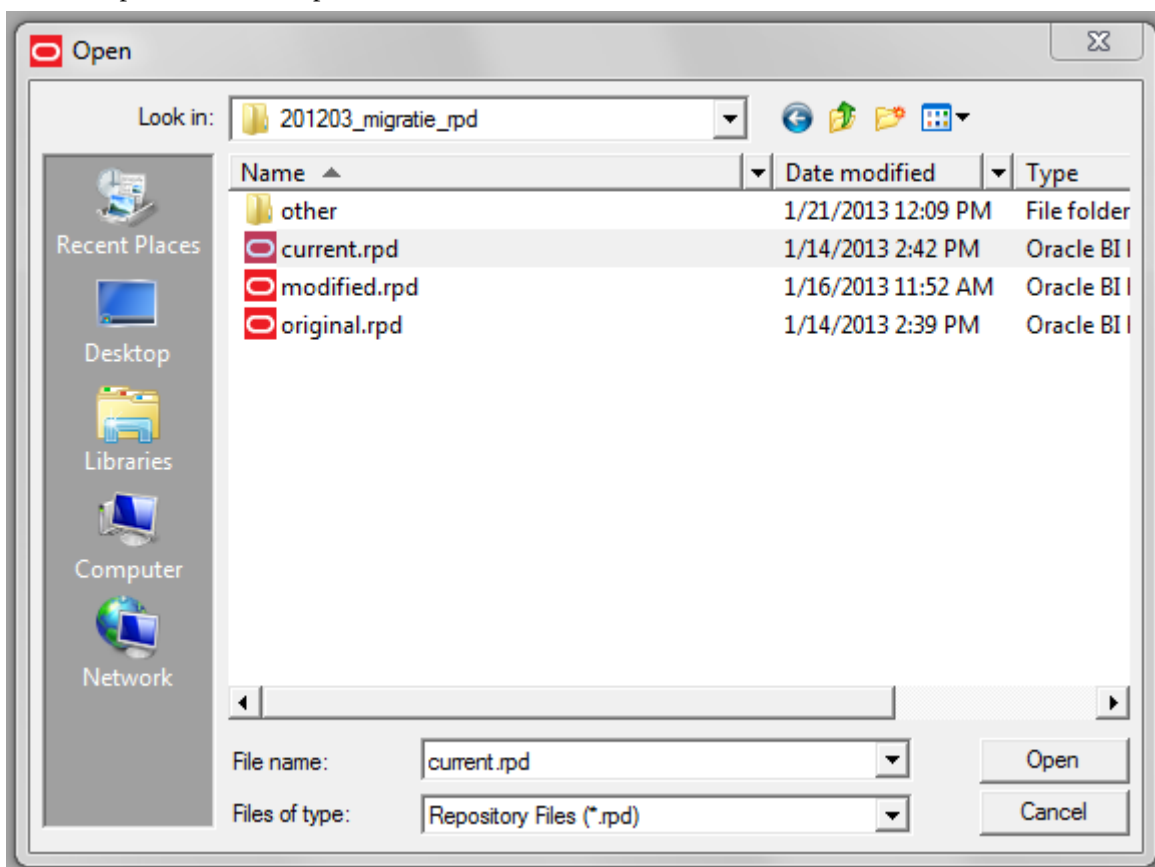
- **Original RPD:** The rpd of the release which is migrated from.
(This is the version provided in the previous release of OHI Data Marts and is unmodified by the Health Insurer, for instance of version 2012.03.01, in the example original.rpd)
- **Modified RPD:** The rpd of the release which is migrated 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.

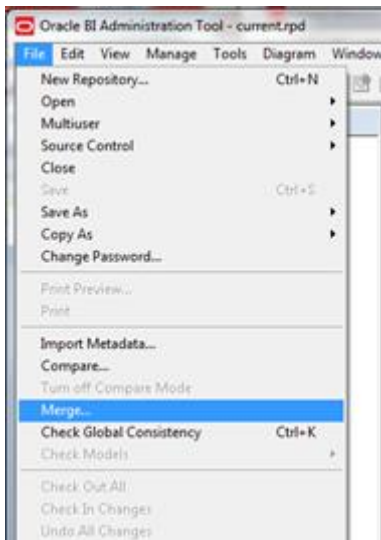
First Download and Install the Oracle Business Intelligence Developer Client Tools Installer. To download the correct version, please consult the certification matrix

(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.



2. File Merge



3. Tick equalize during merge
4. Open original and merge rpd
5. Type the passwords for original and modified repositories.
6. Save merged repository as: DWH_OHI_BI.rpd

 A screenshot of the 'Merge Type' dialog box in the Oracle BI Administration Tool. The dialog has two radio buttons at the top: 'Full Repository Merge' (selected) and 'Patch Repository Merge'. Below are four sections, each with a text field for a repository path and a 'Select' button, followed by a 'Repository Password' field with masked characters. At the bottom, there is a checkbox labeled 'Equalize during merge' which is checked.

Merge Type: ☒ Full Repository Merge ☐ Patch Repository Merge

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

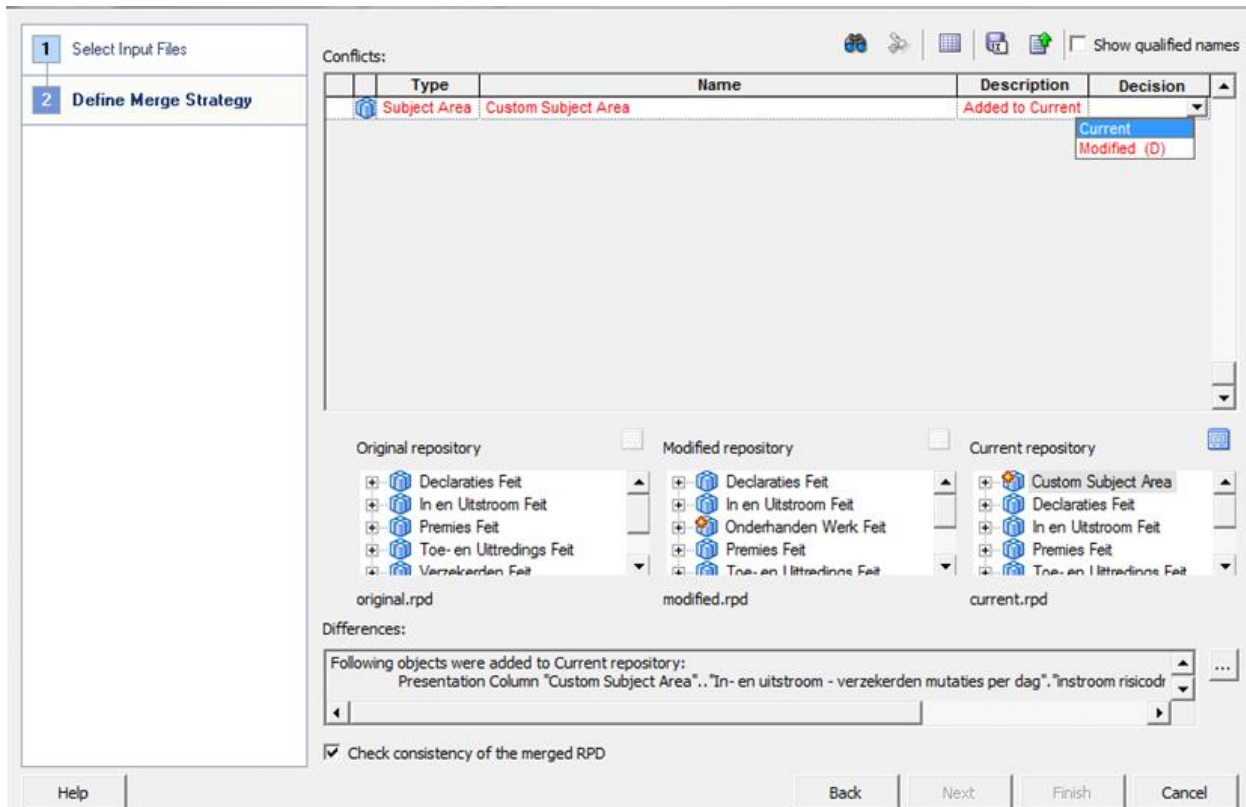
Modified Repository: C:\Users\DUVANL\Desktop\201203_migratie_rpd\modified.rpd Select
 Repository Password:

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

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

☒ Equalize during merge

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

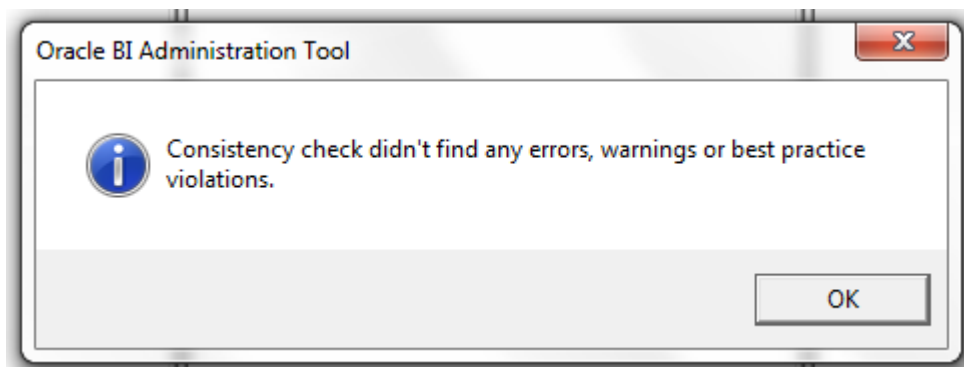


-- choose current to keep the changes in Current repository (Health Insurer RPD)

-- 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.



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:

DWH 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 ZRG0E01S.
SAV_ID_STG	ID of script request ZRG0S01S.
SAV_ID_DWH	ID of script request ZRG0D01S.
DECLARATIES_GELADEN	Indication (J/N [meaning Y/N]) of whether the claims fact is loaded in this run.
VERZEKERDEN_GELADEN	Indication (J/N [meaning Y/N]) of whether the policies fact is loaded in this run.
PREMIES_GELADEN	Indication (J/N [meaning Y/N]) of whether the premiums fact is loaded in this run.
TOE_EN_UITTREDINGEN_GELADEN	Indication (J/N [meaning Y/N]) of whether the In- and Out-fluxs fact is loaded in this run.
ZORGVOORNEMENS_GELADEN	Indication (J/N [meaning Y/N]) of whether the care authorizations fact is loaded in this run.
ABONNEMENTSHONORARIUM_GELADEN	Indication (J/N [meaning Y/N]) of whether the per capita agreement fact is loaded in this run.
VERBINTENISSEN_GELADEN	Indication (J/N [meaning Y/N]) of whether the provider relationships fact is loaded in this run.
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_UTGEVOERD	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

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_GESELECTEERDE_RIJEN	Number of selected rows
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
SCRIPT	Script that executed the load run
MAPPING_NAAM	Name of the mapping
TIJDS TIP_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
ID	Unique ID, generated from sequence
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

Column	Comments
EINDTIJD	End time of the object
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
MAP_SQLERRS	Stacktrace of the SQL-error message
MAP_SQLERRB	Backtrace of the SQL-error message

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.

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

Selectiegericht eigenschappen

Selectiegericht 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_EIGENSCH	PATIENTNUMMER		
4	AANGEVRAAGD					DWH_DECLARATIE_EIGE	NUMBER_EIGENSCHAP_06
4	MACHTIGINGSNUMMER	Alfanumeriek	15	DWH_DECLARATIE_EIGENSCH	MACHTIGINGSNUMMER		
5	BEGINDATUM ZRGVERLENING	Datum	20			DWH_DECLARATIE_EIGE	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_PRESTATIECODELUST	Numeriek	3	0 DWH_DECLARATIE_EIGENSCH	AAND_PRESTATIECODELUST		
8	PRESTATIECODE	Alfanumeriek	14	DWH_DECLARATIE_EIGENSCH	PRESTATIECODE		
9	VOORSCHRIJVER			DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER		
10	SPECIALISME	Alfanumeriek	5	DWH_DECLARATIE_EIGENSCH	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGE	CHAR_EIGENSCHAP_07
11	SOORT BUZONDERE TANDHEEL	Numeriek	3	0 DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BUZ_TANDH		
12	SOORT PRESTATIE	Numeriek	2	0 DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE		
13	LAND	Alfanumeriek	2			DWH_DECLARATIE_EIGE	CHAR_EIGENSCHAP_01
13	UZOV	Numeriek	4	0 DWH_EWE_AWBZ	AWE_UZOV	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATIE	Numeriek	4	2 DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_02
15	INDICATIE BUZONDERE TANDHEEL	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.gebruiksgroep / dwh_declaratie_eigenschappen.gebruiksgroep). The combination of the generic column and the column dwh_declaratie_details.gebruiksgroep, 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 gebruiksgroep 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	Kolom
1	DCL_TND GEBITSELEMENT	Numeriek	2	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_EIGENSCHAPPEN	PATIENTNUMMER	
4	AANGEVRAAGD					DWH_DECLARATIE_EIGENSCHAPPEN
4	MACHTIGINGSNUMMER	Alfanumeriek	15	DWH_DECLARATIE_EIGENSCHAPPEN	MACHTIGINGSNUMMER	NUMBER_EIGENSCHAP_06
5	BEGINDATUM ZRGVERLENING	Datum	20			DWH_DECLARATIE_EIGENSCHAPPEN
5	INDICATIE BOVEN/ONDER TAND	Numeriek	1	DWH_EWE_TANDHEELKUNDE	TAE_IND_B_O_TANDHEELKUNDE	DATE_EIGENSCHAP_04
6	DCL_TND_VLAK_CODE	Alfanumeriek	6	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE	DWH_EWE_GENERIEK
7	AAND_PRESTATIECODELUST	Numeriek	3	DWH_DECLARATIE_EIGENSCHAPPEN	AAND_PRESTATIECODELUST	CHAR_EIGENSCHAP_07
8	PRESTATIECODE	Alfanumeriek	14	DWH_DECLARATIE_EIGENSCHAPPEN	PRESTATIECODE	
9	VOORSCHRIJVER			DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER	
10	SPECIALISME	Alfanumeriek	5	DWH_DECLARATIE_EIGENSCHAPPEN	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGENSCHAPPEN
11	SOORT BUZONDERE TANDHEELKUNDE	Numeriek	3	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BUZ_TANDH	CHAR_EIGENSCHAP_07
12	SOORT PRESTATIE	Numeriek	2	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE	
13	LAND	Alfanumeriek	2			DWH_DECLARATIE_EIGENSCHAPPEN
13	UZOV	Numeriek	4	DWH_EWE_AWBZ	AWE_UZOV	CHAR_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATIE	Numeriek	4	DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEL	DWH_EWE_GENERIEK
15	INDICATE BUZONDERE TANDHEELKUNDE	Alfanumeriek	1			NUMBER_EIGENSCHAP_02
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	Kolom
1	DCL_TND GEBITSELEMENT	Numeriek	2	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_EIGENSCHAPPEN	PATIENTNUMMER	
4	AANGEVRAAGD					DWH_DECLARATIE_EIGENSCHAPPEN
4	MACHTIGINGSNUMMER	Alfanumeriek	15	DWH_DECLARATIE_EIGENSCHAPPEN	MACHTIGINGSNUMMER	NUMBER_EIGENSCHAP_06
5	BEGINDATUM ZRGVERLENING	Datum	20			DWH_DECLARATIE_EIGENSCHAPPEN
5	INDICATIE BOVEN/ONDER TAND	Numeriek	1	DWH_EWE_TANDHEELKUNDE	TAE_IND_B_O_TANDHEELKUNDE	DATE_EIGENSCHAP_04
6	DCL_TND_VLAK_CODE	Alfanumeriek	6	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE	DWH_EWE_GENERIEK
7	AAND_PRESTATIECODELUST	Numeriek	3	DWH_DECLARATIE_EIGENSCHAPPEN	AAND_PRESTATIECODELUST	CHAR_EIGENSCHAP_07
8	PRESTATIECODE	Alfanumeriek	14	DWH_DECLARATIE_EIGENSCHAPPEN	PRESTATIECODE	
9	VOORSCHRIJVER			DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER	
10	SPECIALISME	Alfanumeriek	5	DWH_DECLARATIE_EIGENSCHAPPEN	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGENSCHAPPEN
11	SOORT BUZONDERE TANDHEELKUNDE	Numeriek	3	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BUZ_TANDH	CHAR_EIGENSCHAP_07
12	SOORT PRESTATIE	Numeriek	2	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE	
13	LAND	Alfanumeriek	2			DWH_DECLARATIE_EIGENSCHAPPEN
13	UZOV	Numeriek	4	DWH_EWE_AWBZ	AWE_UZOV	CHAR_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATIE	Numeriek	4	DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEL	DWH_EWE_GENERIEK
15	INDICATE BUZONDERE TANDHEELKUNDE	Alfanumeriek	1			NUMBER_EIGENSCHAP_02
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!

Selectiegericht eigenschappen				Data Marts Specifiek		Data Marts Generiek	
Nr.	Eigenschap	Type gegeven	Lengte	Tabel	Kolom	Tabel	Kolom
1	DCL_TND GEBITSELEMENT	Numeriek	2	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_EIGENSCHAPPEN	PATIENTNUMMER		
4	MACHTIGINGSNUMMER	Alfanumeriek	15	DWH_DECLARATIE_EIGENSCHAPPEN	MACHTIGINGSNUMMER		
5	BEGINDATUM ZRGVERLENING	Datum	20			DWH_DECLARATIE_EIGENSCHAPPEN	DATE_EIGENSCHAP_04
6	DCL_TND_VLAK_CODE	Alfanumeriek	6	DWH_EWE_TANDHEELKUNDE	TAE_DCL_TND_VLAK_CODE	DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_07
7	AAND_PRESTATIECODELUST	Numeriek	3	DWH_DECLARATIE_EIGENSCHAPPEN	AAND_PRESTATIECODELUST		
8	PRESTATIECODE	Alfanumeriek	14	DWH_DECLARATIE_EIGENSCHAPPEN	PRESTATIECODE		
9	VOORSCHRIJVER			DWH_DECLARATIE_DETAILS	ZRE_REL_NR_VOORSCHRIJVER		
10	SPECIALISME	Alfanumeriek	5	DWH_DECLARATIE_EIGENSCHAPPEN	SPEC_VOORSCHRIJVER	DWH_DECLARATIE_EIGENSCHAPPEN	CHAR_EIGENSCHAP_07
11	SOORT BUZONDERE TANDHEELKUNDE	Numeriek	3	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_BUZ_TANDH		
12	SOORT PRESTATIE	Numeriek	2	DWH_EWE_TANDHEELKUNDE	TAE_SOORT_PRESTATIE		
13	LAND	Alfanumeriek	2			DWH_DECLARATIE_EIGENSCHAPPEN	CHAR_EIGENSCHAP_01
13	UZOV	Numeriek	4	DWH_EWE_AWBZ	AWE_UZOV	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_01
14	BTW-PERCENTAGE DECLARATIE	Numeriek	4	DWH_EWE_DBC_ZORGPROD	DZP_BTW_PERC_DECLARATIEBEI	DWH_EWE_GENERIEK	NUMBER_EIGENSCHAP_02
15	INDICATIE BUZONDERE TANDHEELKUNDE	Alfanumeriek	1			DWH_EWE_GENERIEK	CHAR_EIGENSCHAP_06
16	TARIEF PRESTATIE (INCL. BTW)	Numeriek	10				

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_OW 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:

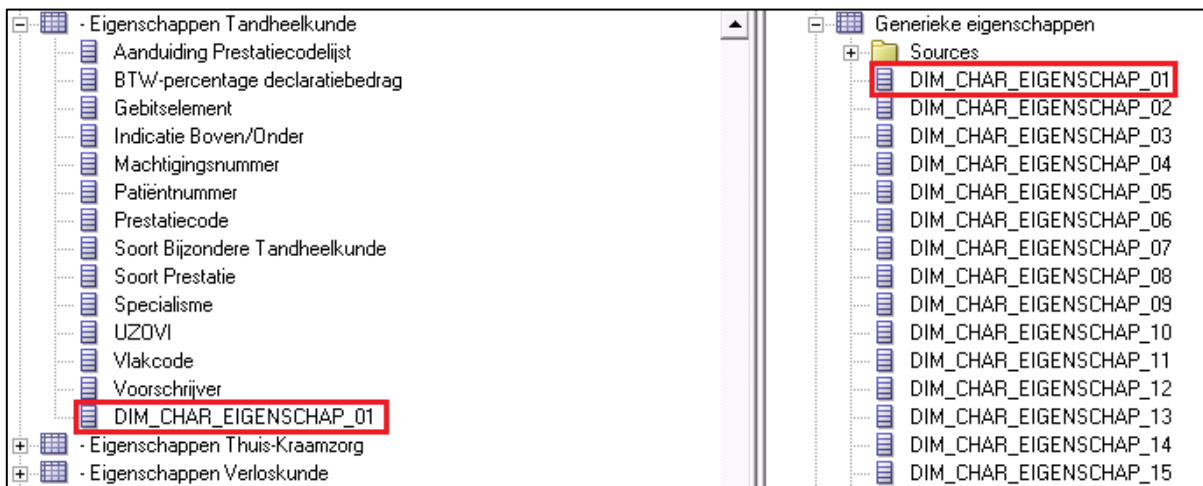
In Procedure Claim Group 'Tandheelkunde' the claim property 'EXAMPLE123' is added. This column is mapped to 'CHAR_EIGENSCHAP_01' in DWH_EWE_GENERIEK. The claim property 'EXAMPLE123' should be added to the presentation table 'Eigenschappen Tandheelkunde' and should be named 'Example 123'.



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_'.

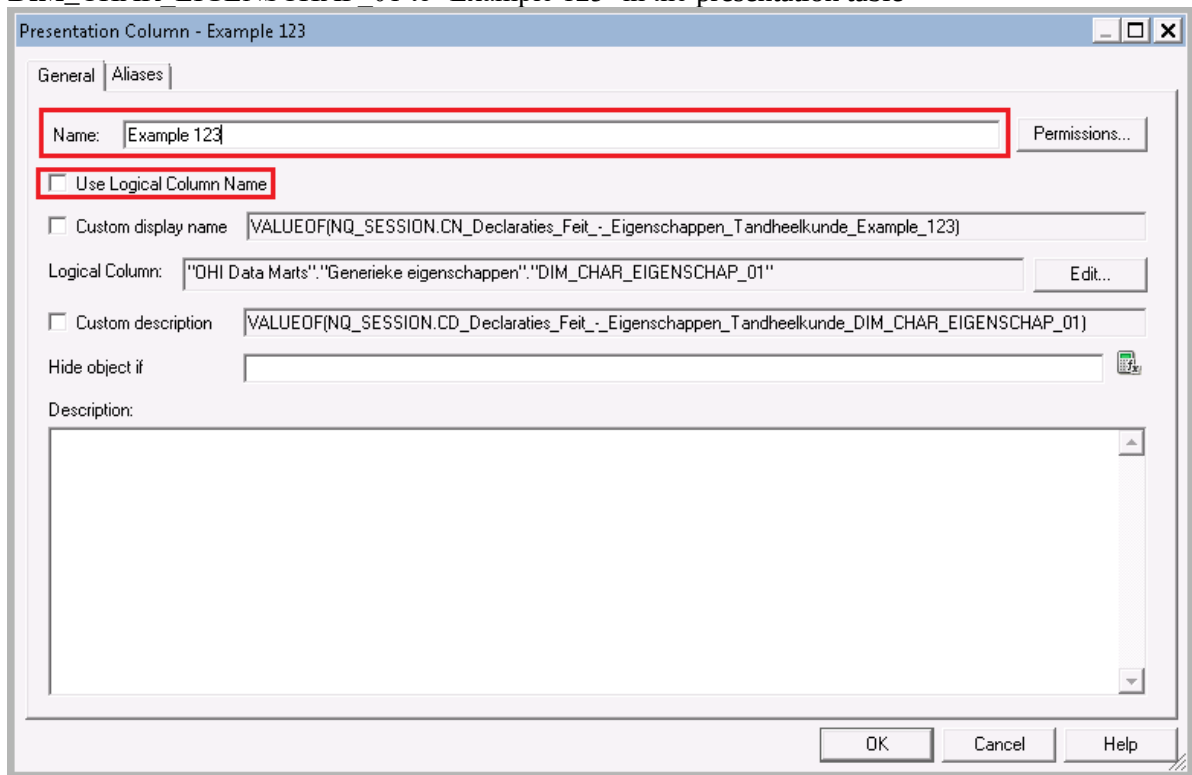
1. Add the generic column to the presentation model

Drag the attribute 'DIM_CHAR_EIGENSCHAP_01' from the Logical Table 'Generieke eigenschappen' to the Presentation Table '- Eigenschappen Tandheelkunde'. This should result in the following situation



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



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).

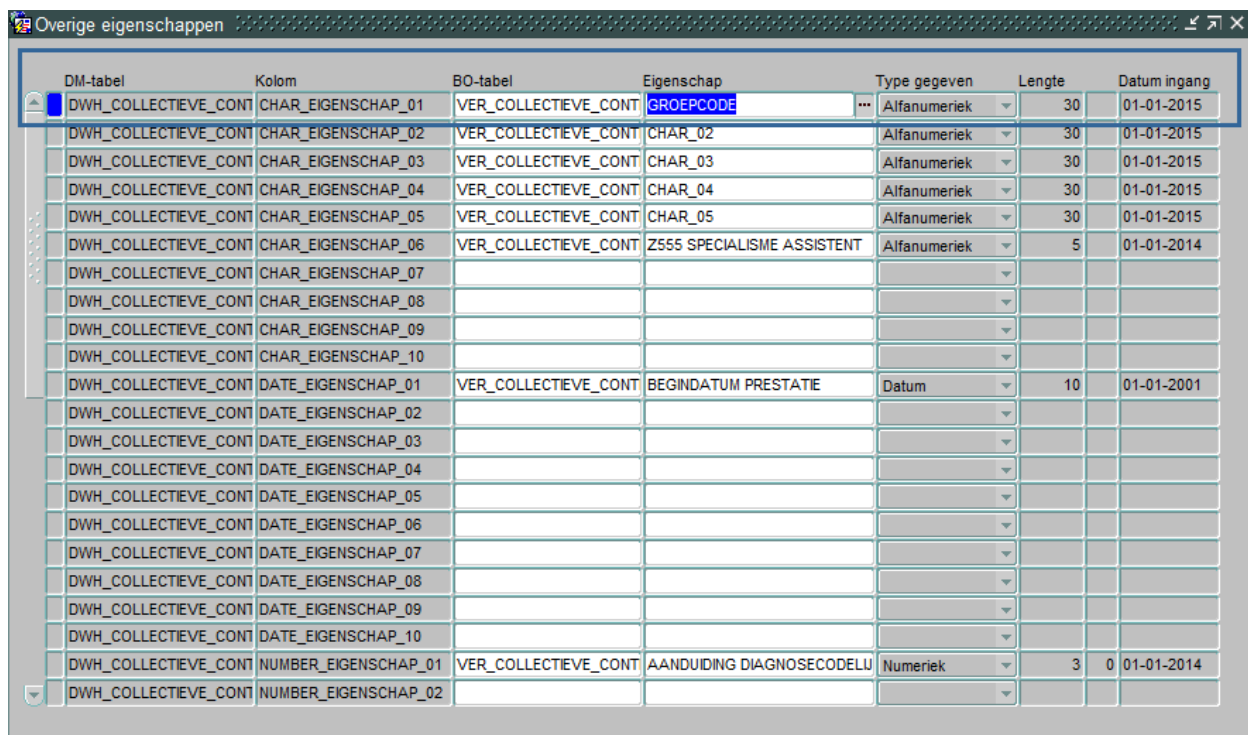
Examples

Example

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, the 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 DIAGNOSECODELU	Numeriek	3	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_OW 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.

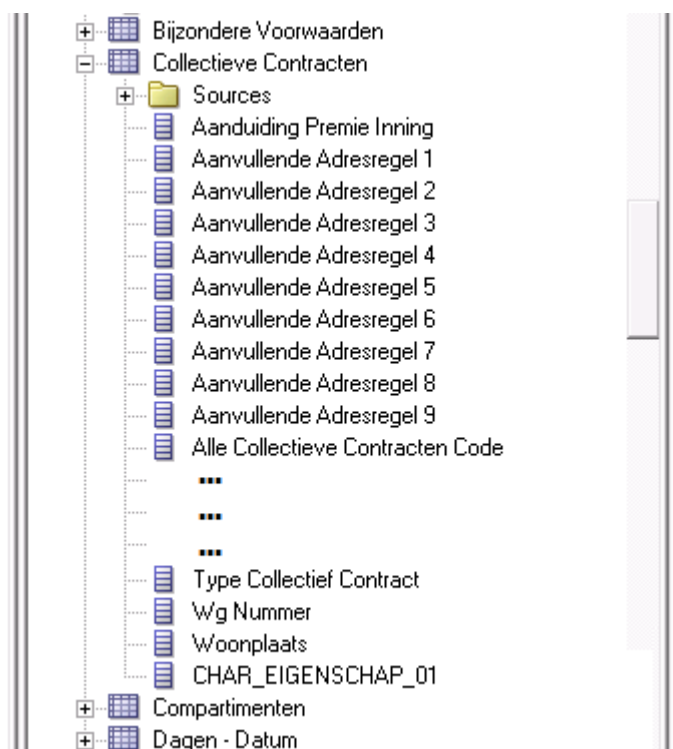
All generic dimension property attributes are already added to the physical layer of the OBI EE repository.

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. Add the generic column to the business model

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. Name the column to the functional meaning

Change the name of CHAR_EIGENSCHAP_01 to 'Groep Code' in the logical table

3. Add the new attribute to the presentation layer

Add 'Groep Code' to the Presentation table 'Collectieve Contracten'

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

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 Oracle Health Insurance Back Office – Release Installation Guide (docs.oracle.com).

COMPRESSION OF PARTITIONED FACT TABLES

As of Oracle Health Insurance Data Marts release 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 exists for both OHI Back Office and OHI Data Marts, but the implementation and submenu choices are different. For OHI Data Marts, the following menus are shown:

```
INFO : =====
INFO : = (Un)Compress a table:                =
INFO : = Q - Compress table          (HCC QUERY HIGH)      =
INFO : = U - Uncompress table        (incl. indexes)        =
INFO : =====
INFO : Which compression action would you like to execute (Q, U)?
Action? >
```

Or:

```
INFO : =====
INFO : = (Un)Compress a table:                =
INFO : = A - Compress table          (ADVANCED LOW)         =
INFO : = U - Uncompress table        (incl. indexes)        =
INFO : =====
INFO : Which compression action would you like to execute (A, U)?
Action? >
```

The type of compression is determined by the platform of the OHI DM database. On Engineered Systems (ExaData, ExaCS, ODA, ADW, etc.) Hybrid Columnar Compression (HCC) Query High is used. On other platforms, where HCC is not available, Advanced Compression is used. Please note that Advanced Compression is a licensed database option.



Note: Advanced Compression is a licensed database option.

If one of the compress actions (Q or A) is chosen, a list of tables that can be compressed will be shown. By typing in the table name the choice for that table is confirmed and the (sub)partitions of the table are compressed. The indexes of the table are not compressed. 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 to do 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 a list of tables that can be uncompressed will be shown. The rest of the procedure is the same as for compressing tables. After uncompressing tables, the local partitioned indexes will also have to be rebuilt.



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



Note: To correct tables that have been compressed by compression types that are no longer supported, the list of candidate tables will include tables that have (indexes or partitions with) an unsupported compression type.

NOLOGGING CLAUSE OF FACT AND DIMENSION TABLES

All tables in the OHI Data Marts database are created with the NOLOGGING option.

In NOLOGGING mode, data is modified with minimal logging (to mark new extents INVALID and to record dictionary changes). When applied during media recovery, the extent invalidation records mark a range of blocks as logically corrupt, because the redo data is not fully logged. Therefore, if you cannot afford to lose the database object, you should take a backup after the NOLOGGING operation.

NOLOGGING is effective in only a subset of the operations that can be applied on an object where NOLOGGING is activated. Only the following DML operations support the NOLOGGING mode:

- Direct-path INSERT (serial or parallel) resulting either from an INSERT or a MERGE statement.
- NOLOGGING is not applicable to any UPDATE operations resulting from the MERGE statement.

The FORCE_LOGGING database parameter can be used to override the NOLOGGING mode of the specified tables in the DML operation. If FORCE_LOGGING is set to YES full logging will be executed for the DML operation, regardless of the NOLOGGING mode of the tables involved.

Since the FORCE_LOGGING database parameter is set at CDB-level and should always be set to YES for an OHI Back Office environment, this parameter can not be set to NO in the situation that the OHI Back Office database and the OHI Data Marts database reside in the same CDB.

So if you want to set FORCE_LOGGING to NO to take maximum advantage of the NOLOGGING of the OHI Data Marts tables you should create a separate CDB for the OHI Data Marts database. An advantage of this choice is space saving (less redo) and a potential faster execution of DML during the load process of OHI Data Marts.

A consequence of this choice is that you have to design an appropriate backup/restore strategy. If FORCE_LOGGING is set to NO in the OHI Data Marts CDB at certain moments (after loading has been executed) a new backup should be created since it is impossible to rely on the previous backup and applying the archived redo logs on that backup.



Note: For more information about Backup and Recovery see the Backup and Recovery User's Guide 19c, E96241-14

PART IV – SET-UP OF DATABASE VAULT FOR DATA MARTS

ATTENTION: This functionality is still under construction and not yet available for general use.

Starting with OHI Back Office release 10.20.6.0.0 an OHI Back Office specific implementation of Oracle Database Vault is expected to be supported in combination with an implementation for OHI Data Marts.

Oracle Database Vault is a licensed option with the Oracle Database Enterprise Edition to implement more restrictive access to database objects and especially implement an additional protection for the data stored in these objects.

For more details see: Oracle Health Insurance Installation, Configuration and DBA Manual, Chapter 9: Installing and Configuring Oracle Database Vault for OHI Back Office (docs.oracle.com).

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.
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 otон1 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 otон 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 acquire 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 significant 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 parallelism, maximum utilization of resources failed.

The first situation is 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 queuing 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 IDENTITY 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 Management 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.

Process	Brand	Description	Q	BrandBroker Univ. Univ.	Grp. Broker Univ. Empty	Grp. Empty
			☐	☒ ☒	☒ ☐	☐
			☐	☐ ☐	☐ ☐	☐
			☐	☐ ☐	☐ ☐	☐
			☐	☐ ☐	☐ ☐	☐
			☐	☐ ☐	☐ ☐	☐

Relation
1453003100
1524241200
1524526800
1647434100
1791224500

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 login
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, WIZ_GROEP_CODE
DWH_ZORG_CLIENEN	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 in 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_ols.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

The following 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_ols.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_ols.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 than 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_ols.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.