ORACLE®

INSURANCE

Oracle Insurance for Health

Release Installation

version 2.18

Part number: E39878-01

March 2013

ORACLE®

CHANGE HISTORY

Release	Version	Changes
10.12.2.0.0.0	2.17	Added change history paragraph.
10.12.3.0.0.0	2.18	Added appendices C&D about database parameters and privileges checkups
10.13.1.0.0.0	2.19	Changed appendix C about database parameter checkups

CONTENTS

Chapter 1

Introduction

This document contains the guidelines, for installation of a release for Oracle Health Insurance customers.

The document provides a technical description concerning the location of releases, how to install releases and how to handle errors during installation.

The files of a release are stored in a directory that has a name equal to the release name.
In this document the placeholder <release> is used for the name of a release. This placeholder has to be replaced by the release name. E.g. When you are installing a release with the release name '1999.01.0.000' and the instructions contain a directory <release> then the directory with the name 1999.01.0.000 is meant.

The term <appl> is used as a placeholder for the various possible applications within Oracle Health Insurance. It concerns ZRG (Back Office), FIN (Financial), SYS (System), REL (Relationship management), etc.

Chapter 2

Basic Assumptions

The following basic assumptions apply to the installation of releases.

DELIVERY

Effective OHI release 2012.01 releases for OHI Back Office and OHI Business Intelligence will be delivered via MyOracleSupport (MOS) in section "Patches & Updates". Previous releases and OHI Self Service are delivered via Oracle's BeehiveOnline website (see the applicable manual).

NAMING OF THE RELEASE

A – Type Releases

There are 3 release types:

Release type	Description
Major release	Regular release with corrective and adaptive functionality.
Patch set	Cumulative bundle of bug fixes and new functionality delivered monthly between major releases.
Interim patch	Bug fix for a certain problem. Can be delivered in a bundle with other bug fixes. Does not contain adaptive functionality.

☞ **N.B.:** Patch sets are *cumulative*: A patch set contains all of the previously delivered patch sets (to that major release) and interim patches.

Example

The OHI Back Office environment has version 2009.02.0.0000.
You want to upgrade OHI Back Office to release 2009.02.4.0001.
To do this, you have to install the following patch sets and interim patches:

1. Patch set 2009.02.4.0000.
   This patch set cumulatively contains all changes from the 2009.02.1, 2009.02.2, 2009.02.3 and all changes from the 2009.02.3.XXXX interim patches.

2. Interim patch 2009.02.4.0001.

B – Oracle Health Insurance Release Calendar

The Oracle Health Insurance Release calendar contains the dates for releasing major releases and patch sets and the end of support dates for major releases.

C - Major Releases

3 major releases are released each year, scheduled according to the Release calendar. Releases names are based on the year plus an additional release number.
Eg:

- 2009.01   First major release delivered in 2009
- 2009.02   Second major release delivered in 2009
- 2009.03   Third major release delivered in 2009

D – Dependency between Major- and Patch Release

A patch release (patch set or interim patch) can only be installed on the associated major release.

Example

Patch release **2009.01.0.**0001 can only be installed on release **2009.01.0**.0000.

E - 1 Release per Environment

There is only one Oracle Health Insurance major release active per "environment" (including patch sets and interim patches concerned), e.g.:

- Production   2009.01.X.XXXX
- Acceptance   2009.02.X.XXXX

F – Release Naming

Release naming is done using the following coding conventions (named positions are divided by a *dot*):

Position	Description
1 - 4	Year in which the release was/is going to be released
6-7	Sequence number of the regular release according to the release calendar
9	Patch set sequence number (0 = it does not concern a patch set)
11-14	Interim patch sequence number

INSTALLATION SEQUENCE

The following basic assumptions apply to the installation sequence of patch releases:

A- Major Release Dependency

Installation of Oracle Health Insurance releases must be done in order of *dependency*. This means that the major release is installed first and subsequently the patch sets and /or interim patches. It is not possible to install a patch set or interim patch when the associated major release has not yet been installed.

Example

Before *interim patch* **2009.01.0.0001** can be installed, *major* release **2009.01.0.**0000 has to be installed.

B - Sorting on Sequence Number

Sequence number orders patch sets and interim patches.

Example 1 – patch sets

2009.01.**1**.0000 is installed before 2009.01.**2**.0000.

Example 2 – interim patches

2009.01.1.0001 is installed before 2009.01.1.0002.

Mandatory Installation

Interim patches

When an interim patch is installed, all previous interim patches have to be installed on the same patch set or major release.

Patch set

It is mandatory to install a patch set (usually 3 or 4) before a major release is installed; this patch set will be mentioned explicitly in the release calendar.

Chapter 3

Installation of a Release

REQUIREMENTS

During the installation of an Oracle Health Insurance (patch) release no other
sessions should be active, performing actions on the Oracle Health Insurance objects
in the database; the installation operates on these objects and therefore requires
exclusive access.

It is therefore necessary that the following conditions be met:

Restricted Sessions

At the time of installation, only restricted sessions may be allowed; this ensures that
normal users will not have access to the database.
The Oracle Health Insurance owner account, under which the database installation is
performed, therefore has the system privilege of a RESTRICTED SESSION.

The status can be obtained in 2 ways:

1. The database has to be started in restricted mode (STARTUP RESTRICT).

2. When it is not possible to bring down the database first, ensure that no user
    sessions are active and that only restricted sessions are permitted (ALTER
    SYSTEM ENABLE RESTRICTED SESSION, followed by closing the non-
    restricted sessions).

When the database needs to become available again, change into non-restricted mode
(with a normal startup or ALTER SYSTEM DISABLE RESTRICTED SESSION).

Batch Scheduler

Before a (patch) release is installed, the OHI Back Office batch scheduler has to be
stopped.

Users

No Oracle Health Insurance users should be active. Apart from the installation
session of the Oracle Health Insurance owner account which should be active.

BACKUP REQUIREMENTS (ARCHIVELOG AND NOLOGGING)

OHI Back Office on one hand and OHI Business Intelligence on the other have
specific backup requirements.

This is related to the two differences between the OHI Back Office and OHI Business Intelligence environments:

1. **ARCHIVELOG VS. NOARCHIVELOG mode**
    An <u>OHI Back Office</u> environment has to run in <u>ARCHIVELOG</u> mode; this allows the performance of a point-in-time recovery.

    An <u>OHI Business Intelligence</u> environment normally runs in <u>NOARCHIVELOG</u> mode; in case of a recovery the last full backup needs to be restored.
    It is therefore, advisable for an OHI Business Intelligence environment to make a *full backup **before*** and ***after*** the installation.

2. **Use of NOLOGGING functionality**
    When NOLOGGING database functionality is used, modifications are not logged completely; this makes it impossible to use a restore of a backup before the start of the installation and the subsequent application of any unarchived redo logs to come to a consistent/usable situation as it existed during or after the release installation.
    In case NOLOGGING is used, you will have to make a *full back* up *before and after* the installation, to be able to restore database files from the backup, if necessary. A restore based on the backup made before the installation will not be successful, even when you run in ARCHIVELOG mode.

    It is also necessary for enabling the use of the NOLOGGING option the database (or the tablespaces we discuss below) is not running in FORCE LOGGING mode. This is normally not applicable, especially when no standby database is used. When FORCE LOGGING is active the installation may be delayed considerably but the requirement of a backup before and after the installation is not strictly necessary. Although it is very advisable to prevent long restore times in case a restore is needed for some reason.

    The NOLOGGING functionality is used for <u>OHI Business Intelligence</u> for <u>each installation</u>; the advice to make a *full backup before* and *after* the installation is therefore also applicable here.

    To <u>OHI Back Office</u> applies that NOLOGGING is normally used only for the <u>installation of major releases</u>; in this case a *full back up* has to be made *before* and *after* the installation. However, this is not enforced and for that reason we advise to use the FORCE LOGGING option during non major release installations.

IMPACT ON CHANGE LOGGING

When a new release is installed database objects may be changed, dropped, disabled, etc. This may impact change logging (modification logging) functionality which might impact custom functionality based on this. For more information please read the white paper 'OHI Back Office modification logging' which is present on BeehiveOnline.

SPECIFIC REQUIREMENTS FOR INSTALLATION OF MAJOR RELEASE

A number of specific requirements apply for the installation of an Oracle Health Insurance *major* release on an Oracle Health Insurance environment.

Space Management

Additional space needed

It is important to have a large amount of temporary sorting space available in the standard temporary table space. There also has to be sufficient undo space to process long-term operations in 1x. Finally, additional space is needed for the increased size of tables and indexes.

For the above components applies that a minimum of 2 times the (unfragmented) space of the largest index is required.
The following query determines the required MBs:

```
select round((max(bytes)/1024/1024))*2 "Required #MB temp space"
from   dba_segments
where  segment_type = 'INDEX'
and    owner       = '<Name Oracle Health Insurance owner account>'
/
```

Use temporary space during installation

The following query can be used to monitor the temp space during installation:

```
select tablespace_name
,      total_blocks
,      used_blocks
,      free_blocks
from   v$sort_segment
/
```

Other Database Settings

When large indexes are created during installation, it is important to temporarily (!) set the adjusted initialization parameters below (Automated SQL Execution Memory Management off, sort area to 200MB, hash area to 100MB):

```
workarea_size_policy=manual
sort_area_size=209715200
hash_area_size=104857600
```

☞ **Note:** Once the installation is finished, please make sure to return these 3 parameters to their original settings.

☞ **Note:** This only applies for installations on OHI Back Office environments or OHI Business Intelligence environments prior to release 2009.02.0.0000.

PARALLEL EXECUTION

During installation of the Oracle Health Insurance (patch) releases Parallel Execution is used to reduce the time needed for the installation.

To enable maximum use of this function, a number of settings have to be entered in the database.

These settings are active for OHI Business Intelligence by <u>default</u>, see:

Oracle Health Insurance - OHI Business Intelligence Administrator Reference

For OHI Back Office applies that these settings are *mandatory* during the installation of a <u>major</u> release. These settings are optional for the installation of <u>patch sets</u> and <u>interim patches</u>.
Parallel Execution is *not* permitted during OHI Back Office *runtime*.

General

To use Parallel Execution during installation of (patch) releases, the following database parameter have to be set:

parallel_max_servers=...

Set this parameter to twice the number of CPUs; see also the section *Different settings* for specific situations.

The following parameters should not be set:

parallel_min_servers
parallel_execution_message_size

The above settings manage of the default parallelization ratio (roughly twice the number of CPUs present) and will apply to most environments.

Note: Once the installation is finished, please make sure to return the parameters concerned to their original settings.

Different Settings

parallel_max_servers

The above settings will lead to accelerated installation only when sufficient CPU-, I/O- and memory capacity are available.

When this is *not* the case, *or* when other environments and/or applications are running on the server and you would like to prevent that parallel operations use up your server's capacity, the value may be set to an explicitly lower value than the default value.
This can be done with the parameter `parallel_max_servers`. Set this parameter to 2 times the amount of CPUs available for this environment; this will normally ensure an acceptable load to take place.

When there are for instance 6 CPUs and it is desired to fully load only 2 with parallel jobs, the value has to be set to 4. Generally, these 4 processes will keep two processors quite busy.

It is possible that the idea of your own environment is different as this all depends on the mutual relations of the speed of the I/O system and the CPUs speed.

When it becomes apparent during a (test) installation that the system is primarily waiting for I/O operations, it is possible that too many parallel processes are running. In this case, reduce the maximum number of parallel processes by using a lower value for `parallel_max_servers`

sort_area_size

It is possible that during an installation several parallel server processes temporarily use 200MB of working memory with the above setting for `sort_area_size`.

This can be the case for at most half of the number of parallel servers available (determined with `parallel_max_servers`). Therefore sufficient working memory needs to be available in the server. This will normally be available, as the application is not in use, but might be a problem on test servers. For test servers the *sort_area_size* can be decreased to 100MB.

This also applies to the `hash_area_size` parameter.

UNZIPPING A RELEASE

The file containing the Oracle Health Insurance (patch) release is a Unix zip file in the format < Oracle Health Insurance release>.zip.

Example

2009.01.0.0000.zip

This file can be unzipped by giving the following command in the Unix directory `patch`:

unzip <Oracle Health insurance release>

Example

unzip 2009.01.0.0000

DIRECTORY STRUCTURE

Once the sent file has been unzipped, a <release> directory is created containing the complete release.
You can now navigate to the release directory:

cd <Oracle Health Insurance release>

Example

cd 2009.01.0.0000

The directory structure of this Oracle Health Insurance release is:

Directory	Source files present
admin	Utilities
bin	Screens, menus (Forms) (object) libraries and Pro*C sources
conf	Configuration files
doc	Documentation relating to the release.
help	On-line help information
install	Database installation scripts
java	Java-related files
report	Reports and (Reports) library
sh	Unix shell scripts
sql	SQL modules
xml	XML-related files

Target Directories

After successful installation, the sources and executables are copied to their target directories.

After compilation / generation of sources, the following files are created:

Target Directory	Source file Extension	Executable File Extension	Module type
$OZG_BASE/bin	.fmb	.fmx	Oracle*Forms Form
	.pll	.plx	Oracle*Forms Library
	.mmb	.mmx	Oracle*Forms Menu
	.olb	<no executable>	Oracle*Forms Object Library
	.pc	<no extension>	Oracle Pro*C
$OZG_BASE/conf	<various>	<no executable>	Configuration files
$OZG_BASE/report	.rdf	.rep	Oracle*Reports Report
	OZGRLIB.pll	<no executable>	Oracle*Reports Library
$OZG_BASE/sql	.sql	<no executable>	SQL module
$OZG_BASE/sh	.sh, .pl, .ctl	<no executable>	OS shell scripts, Perl scripts, SQL*Loader Controlfiles
$OZG_BASE/xml	.xsd, .xsl, .xslt	<no executable>	XML-related files
$OZG_BASE/help	<various>	<various>	Online help information
$OZG_BASE/java	.ear, .java, .jar	<no executable>	Java-related files
$OZG_ADMIN	<various>	<no executable>	Utilities

⚠ **N.B. (1):** Compilation of OHI Back Office EI Media consists of importing the Media in the database (menu option 130).

⚠ **N.B. (2):** When a new version of the Oracle*Reports library file OZGRLIB.pll is installed, *all* Oracle*Reports source files (.rdf) have to be *recompiled* (.rep) with menu option 810.

⚠ **N.B. (3):** When a new version of the Oracle*Forms (object) library files OZGLIB.pll (and executable) and/or *.olb and/or reference forms OZGREF*.fmb are installed, *all* Oracle*Forms source files (.fmb) have to be *recompiled* (.fmx) with menu option 800.

RELEASE DOCUMENTATION

The release documentation is located in the documentation directory in HTML format (*<release> *.htm*). This documentation contains the description of all changes delivered via the release and a summary of the released sources.

Specific details, like test plans, specific installation issues etc., are recorded in this documentation.

The documentation also contains overview of files in the release.

Finally, it contains list of sources that have been obsoleted.

THE README.TXT FILE

The documentation directory contains a `readme.txt` document.
This document contains a short description of all steps, and the order in which they have to be performed, to install the release.

USE OF VARIABLES DURING INSTALLATION

Variables Definition

During installation when necessary variables will be used.

Different types of variables can be used: *generic* variables, which are generic for all environments, *environment-specific* variables, which are specific for a certain environment and *(patch) release-specific* variables, which are specific for a certain release.

Generic variables

The installation software expects definitions from the *generic* variables (that can be used for every (patch) release) in file `ozdefine.sql`, this file is expected to be in the `$OZG_ADMIN` directory.

An initial version of this file is available on the BeehiveOnline website. The settings in this file are *case insensitive*.

Environment-specific variables

There are, however, variables which have *different values in different environments*; such variables are defined by entering a call to an *environment-specific file* `ozdefine.sql.$ORACLE_SID` in the *generic* `ozdefine.sql`.

The call of ozdefine.sql `$ORACLE_SID` on runtime is subsequently replaced by the actual value for `$ORACLE_SID`, e.g. `PROD`, and will now automatically call the script `ozdefine.sql.prod`.

The variables vary per customer per environment and are supposed to be assigned values by the customer.

Example generic ozdefine.sql script

```
start $OZG_ADMIN/ozdefine.sql.$ORACLE_SID #do not replace $ORACLE_SID
```

Example environment specific script ozdefine.sql.PROD

```
define OMGEVING=P
define OZGPATCH_SUPPRESS_WARNING=N
```

See the template files on the BeehiveOnline website for clarification on the parameters (which are all required).

(Patch) release-specific variables

When variables are used in a (patch) release specific to this (patch) release (this may be the case in .sql scripts), these variables are entered in the file `readme.<PRODUCT>` (where PRODUCT is OZG for OHI Back Office and OBD for OHI Business Intelligence.
The file is located in the release directory.

This file has to be *modified* and then *copied* to a file named
`readme.<PRODUCT>.<$ORACLE_SID>`, eg. `readme.OZG.prod`, so that this file is specific for the release concerned in the specific environment (the Oracle Health Insurance .sql scripts also contain a call to `readme.OZG.$ORACLE_SID`, *not* to `readme.OZG`).

In this way it is possible to retrieve with which settings the installation has run *for any release on any environment for any product at any given moment*.

B.T.W. The installation menu does copying automatically; the installation menu will also provide the interface to modify the values for the variables.

Restarting

The data of the installed release are stored in the `ALG_RELEASES` table during installation. If this release has been installed before in the current environment, an error message will occur; *reinstallation* is an option.

Installation of a (patch) release can always be restarted.
When errors have occurred during the first installation, these can be solved after which the installation can be restarted. An example is a *space management error* upon creation of a table (e.g. insufficient space in the table space).

SQL SCRIPTS

SQL scripts may be part of a (patch) release.

SQL scripts should always contain:

totals of changed and rejected and / or erroneous records;

details of rejections or errors;

Start- and end time of the script;

Facilities for monitoring of the script by the DBA (not applicable for check scripts, as these only contain selections instead of updates/deletes)

The following functionality is optional (when judged to be necessary by Oracle):

the way in which data can be restored (manually);

details of the changed records (n/a for check scripts).

There are 2 categories of SQL scripts: Check scripts and automatic SQL scripts.

Check Scripts

Check scripts are meant to help install a (patch) release without functional installation errors; they check for the presence of inconsistent data; only selections are performed.
These scripts are always executed before the actual installation takes place. It is not always known prior to the installation if data inconsistencies exist and if they do exist, how they can be resolved. It is not always possible to detail how to resolve inconsistencies detected by the scripts. Check scripts are never allowed to adjust data.

Check scripts can also be started prior to an installation.

Oracle strives to deliver the check scripts to the customer as early as possible however, this is often not possible. The script cannot be delivered before a (patch) release is has been completed.

Automatic SQL Scripts

Automatic SQL scripts run automatically as part of the database installation.

These scripts perform updates and/or deletes on data.

WALLET ENTRY

Wallet entry for application owner account

From release 2012.01.0.0000 from OHI Back Office and OHI Business Intelligence, the database login for the application owner, like the account for the batch scheduler, must be defined in the Wallet. The variables `ozg_un` and `ozg_pw` are no longer stored in the `ozg.conf.$ORACLE_SID` configuration file. For backwards compatibility, older releases are still using the login defined in the configuration file.

The Wallet entry must be named **`<$ORACLE_SID>_install`**, and must login the application owner in the corresponding database. For example, when the database is named `PROD` and the owner is named `OZG_OWNER`, a Wallet entry and tnsnames entry `PROD_INSTALL` must be created, such that the following command:

```
sqlplus /@prod_install
```

logs in into the database and the sqlplus command:

```
SQL> show user
```

displays 'USER is "OZG_OWNER"'.

See the 'Oracle Health Insurance Back Office Installation, Configuration and DBA Manual' (cta13508.doc) for more details on how to create this secure wallet entry.

Chapter 4

Installation Menu OZGPATCH

INTRODUCTION

Installation menu OZGPATCH.pl is intended for the automated installation of Oracle Health Insurance (patch) releases.

The installation menu shows a menu with activities to be performed based on the releases properties file $OZG_PATCH/<release>/doc/<release>.prp.
It is determined per product line (one has a license for) which actions have to be performed. It can then be used to perform installation of objects in the database and on the file system.

The following sections describe the functionality of the script and examples indicate what the properties and options are.

Perl Modules

To run the menu, it is required to install a number of additional Perl modules; see

> Oracle Health Installation, Configuration and DBA Manual
> *Appendix F – Installing required Perl modules*

GENERIC MENU FUNCTIONALITY

Start

The OZGPATCH.pl script can be found in the $OZG_ADMIN directory.

The script is started under oracle with the name of the database to be patched as the parameter.

$ OZGPATCH.pl <name database>

It is not mandatory to be in a certain directory when OZGPATCH is used.

Configuration

First time that OZGPATCH is used for an Oracle Health Insurance environment, some questions will be asked.

OZGPATCH will ask for a new one if the password has been changed.

List of (patch) releases to be installed

Subsequently, it has to be indicated which (patch) release has to be used to start the patching and which (patch) release has to end it.

The chosen values are stored in a configuration file and will be the default choices for the next installation.

Whenever a range of (patch) releases is selected, the installation menu will *propose* on what (patch) releases in this range need to be installed.
After this proposal, the user will get the opportunity to override the proposed selection.
Following algorithm is used for the proposed selection:

1. The provided begin and end release ("*boundaries*") will be selected (unless it's a patch set for which it is not necessary to install; see 4 & 5);
2. All *major releases* in the range will be selected;
3. All interim patches on the last major release or patch set in the range will be selected;
4. All *interim patches* will be selected when a range of patches is selected *for the same major release and patch set*;
5. In case of migrating to another (cumulative) patch set or major release, do *not select intermediate patches or patch sets;*
6. In case of migrating to another major release; if (*mandatory*) *patch set 4* is available this *will be selected*; subsequent patch sets will not be selected. If patch set 4 is not available, the *highest available patch set* for the major release will be selected.

Preconditions

Subsequently, a number of preconditions (i.e. availability of utilities needed and use of the recommended directory structure) will be checked.

When these checks do not result in any issues, the menu will appear and show the first (patch) release selected.

When the selected (patch) release does not belong to the correct product line (e.g. OHI Back Office and OHI Business Intelligence) this will be reported and the (patch) release will automatically be skipped.

File Authorization

The files created by the installation menu are created with rights (`umask 077`) so that these can only be read under the account used to start the installation menu (normally `oracle`).
Other accounts cannot view these files.

The Menu

The menu shows the activities for each release. The overview of the activities to be performed per product line can also be read in the `readme.txt` file.

Based on the Oracle Health Insurance release 2008.01.0.0000 a (fictitious) example of the menu is included:

```
+---------------------------------------------------------------+
|          Oracle Health Insurance installation menu            |
```

```
+----------------------------------------------------------------+
  Release 2009.01.0.0000, database prod


  (+) A v Preparation
  (+) B v Database installation
  (+) C v Filesystem installation
  (+) D v Completion
  (+) ?   Help


+----------------------------------------------------------------+
Make your choice [10]:
```

When this menu has been expanded completely, the following is visible:

```
+----------------------------------------------------------------+
|          Oracle Health Insurance installation menu          |
+----------------------------------------------------------------+
  Release 2008.01.0.0000, database prod


  (-) A v Preparation
        10 v Fill parameter file
        20 v Process specific installation instructions
        30 v Place installation modules
        90 o Perform object validation
       100 v Run SQL check scripts
  (-) B v Database installation
       110 v Run database installation script
       115 v Validate non-validated constraints
       120 v Process synonyms, grants and schema compilation
       130 v Import External Integration media
       140 v Import Discoverer EULs
  (-) C v Filesystem installation
         C1 v Complete filesystem installation WITOUT compilation/generation
         C2 v Complete filesystem installation WITH   compilation/generation


        200 v Place OS modules
        205 v Place OHI Business Intelligence OS modules
        210 v Place SQL modules
        215 v Place OHI Business Intelligence SQL modules
        220 v Place online help information
        230 v Drop obsolete objects
        240 v Place Java/XML-related files
        250 v Place database DDL scripts
        260 v Place configuration files
        270 v Deploy ear files
        300 v (Compile and) place (v 300CP) Pro*C modules
        400 v Place Reports printer definitions
        410 v (Compile and) place (v 410CP) Reports library
        420 v (Generate and) place (v 420CP) reports
        430 v (Generate and) place (v 430CP) OHI Business Intelligence reports
        500 v Place reference forms
        510 v Place Forms object library
        520 v (Generate and) place (v 520CP) Forms libraries
```

530 v (Generate and) place (v 530CP) forms and/or menu's
(-) D v Completion
    800 v Compile/Generate objects in $OZG_BASE/bin
    810 v Compile/Generate objects in $OZG_BASE/report
    820 o Gather table/index statistics
    830 o Show Business Rules validation status
    840 o Drop obsolete columns
    850 o Partition/compress tables
    860 o Move tables/indexes
    900 v Perform object validation

(-) ? Help
```
_____ Symbols _____
v Mandatory execution       * Successful execution
o Optional execution        x Errors after execution


_____ Menu control_____
- Open entire menu       + Close entire menu
A Submenu A open/close      C Submenu C open/close
B Submenu B open/close      D Submenu D open/close


_____ Menu options _____
L Show log file for last choice M Automode
O Skip default choice       R Read readme.txt file
S Reset log file(s)       T Show logfile timing
U Update client releasetoken    N Next (patch)release
Q Quit              Z Create logfiles zip
$ OS command line         @ SQL*Plus (ozg_owner)


_____ Info _____
Perl version: 5.8.5       RCS Version: 4.80
                  [-]
+----------------------------------------------------------------+
```
Make your choice [10]:

Collapsing and Expansion of the (Sub-)Menu

The entire menu can be expanded by keying in - (minus).
The entire menu can be collapsed by keying in + (plus).

A submenu can also be collapsed or expanded by entering the code in the submenu;
A, B, C, D or ?.
The codes entered are *case insensitive*.

Choosing Options

An option can be chosen by entering the option number, e.g. 100, 410, 530, etc.

The generic options can be chose by entering the indicated code: L, V, M, S, Q, etc.

These codes are also *case insensitive*.

Answering Questions

Answering a question can only be done with `Y`, `y`, `N`, `n` or <enter>.
For backward (Dutch) compatibility, `J` and `j` are also valid answers (to act as `Y`).
When <enter> is given, the answer proposed between `[ ]` is chosen.
The proposed answer does not have to be copied.

Log Files

Naming

When an activity is chosen and executed, a log file is created in the following format
`<release>.<activity>.log.<database SID>`.

When an activity is performed again, the existing log file will be renamed; a
date/time component will be placed after the name.

Example

2009.01.0.0000.110.log.prod

becomes

2009.01.0.0000.110.log.prod.Mon_01-Mar-2009_14:11:35

Showing logfiles

To view an activity's log file, the option `L` has to be chosen.
The log file of the last activated activity will then be shown; in this case the
"*context*" of the menu is set to this last activated activity.

When you want to view a log file which is not the last activated one, the *context*
should first be set to this activity.
This can be done by choosing the activity, and answering the following question '`Do
you want to execute activity … again?`' with `N` (No). It will subsequently be
possible to choose option `L` again.

To view *all* of a submenu's activities, again the *context* has to be set to the submenu,
by choosing it first (e.g. `A`; it does not matter if the submenu is expanded or collapsed
with this choice, the only requirement is that this is the current choice), and then
option `L`.
To view all log files for *all* activities for the *complete* menu, the main menu has to be
chosen first (with + or -) and subsequently option `L`.

The *active context* can be seen on the *right bottom of the menu*; in the expanded
menu example above the context is set to -, as indicated by `[-]` on the last line.

Viewer used

To show the log files the installation menu uses the `more` utility.

Errors

During the performance of a menu activity, the following errors may occur (which
will be registered in that activity's log file):

1. `ORA-`
   Oracle DBMS errors.
2. `IMP-`
   Errors during execution of the Oracle import utility.
3. `ERROR:`
   Errors during installation of an Oracle Health Insurance (patch) release.
   The installation will continue.
4. `PL/SQL ERROR`
   Errors during compilation of Oracle Health Insurance customer sources.
5. `SP2-`
   SQL*Plus errors during installation of objects in the database.
6. `RTC-` en `OMBXXXXX`
   Oracle Warehouse Builder Control Center Service errors during OWB
   mapping packages installation in the database with the `OMBPlus` utility.
7. `FATAL:`
   Fatal installation errors during an Oracle Health Insurance (patch) release
   installation. The installation is aborted.

At the bottom of the screen, at the end of the log file, before
"`Press <Enter>` " (press a key) it will state if any errors have occurred.
Before patching continues, each technical error will have to be restored.

Messages with the following prefix are meant as a warning and will for instance
occur when a patch is installed for the second time or when invalid objects still exist
at the beginning of an installation. An installation is not perceived as erroneous in
case of a warning:

- `WARN :`

Other messages with the following prefix are informative:

- `INFO :`

OPTIONS DESCRIPTION

Symbol Use

o = Optional execution

When an option is optional (and is not mandatory for this (patch) release) but has not
been executed yet, an o will appear in front of the option.

v = Mandatory execution

When an option is mandatory but has not yet been executed, a *v* will appear before
the option.

* = Successful execution

When an option has been executed successfully, an * will appear in front of the
option.
This applies to optional and mandatory options.

x = Errors after execution

When execution of an option has caused errors, an *x* will appear in front of the option.
This applies to optional and mandatory options.

Examples

Example 1 – Option 800 is mandatory

  800 v Compile/Generate objects in $OZG_BASE/bin

Example 2 – Option 800 is optional

  800 o Compile/Generate objects in $OZG_BASE/bin

Example 3 – Option 800 was executed successfully

  800 * Compile/Generate objects in $OZG_BASE/bin

Example 4 – Option 800 was not executed successfully

  800 x Compile/Generate objects in $OZG_BASE/bin

Sub-menus

The following rules apply for submenus

1. When 1 or more activities in the submenu are mandatory and have not yet been executed, the submenu will be indicated as mandatory and not yet executed.
2. When 1 or more activities in the submenu have not been executed successfully, the submenu will be indicated as not successful.
3. When there are no mandatory activities in the submenu and only successfully executed activities exist in the submenu, but only (still) optional activities, one or more of which have not yet been executed, then the submenu will be indicated as optional.
4. When only successfully executed activities are present in the submenu, the submenu will be indicated as successfully executed.

A Preparation

The options in this submenu are used to execute preparatory activities, like the preparation of installation modules and utilities, which may be used.

10 Fill parameter file

Filling of the release-specific parameter file, needed for the database installation of the (patch) release.

This action has to be performed manually, which is why confirmation of its execution is requested explicitly.

Editor used

The editor is used to fill in the parameter file the way it is set in the OS shell variable `$EDITOR`.

When it is not set, `vi` is used.

20 Process specific installation instructions

Specific installation instructions are shown; these have to be processed first.

This action has to be performed manually, which is why confirmation of its execution is requested explicitly.

30 Place installation modules

Placing installation modules in `$OZG_ADMIN` & `$OZG_BASE/<subdirs>`.

Care is taken during this activity that any higher versions of the objects concerned are not overwritten.
This check is performed because it concerns objects in a generic directory, which means that they apply to several releases.

90 Perform object validation

Performing an Object Validation; used to check the presence, versions and status of the Oracle Health Insurance objects.

Choose to check the database objects [D], the filesystem objects [F] or both [B].

This option is optional and can be used to check the environment before a release is installed.

This option is *always* visible in the menu.

100 Run SQL check scripts

Running of (patch) release SQL check scripts.

☞ **Note:** This option can and should be run as soon as the (patch)release becomes available; this way the possible reported issues can be fixed in time, before the actual installation of the complete (patch)release.

B Database Installation

The (patch) release database installation activities are executed through options in this submenu.

110 Run database installation script

Running of the database installation script.

115 Validate non-validated constraints

Validation of constraints with status NON-VALIDATED under the Oracle Health Insurance owner account.

This option does not apply to OHI Business Intelligence.

A constraint is always created ENABLED for an Oracle Health Insurance database installation; which means that the constraint applies to all *new* records.

To validate the constraint for the *existing* records this option is used.
The number of records which cannot be validated is indicated per table/constraint
where applicable.

This option is *always* mandatory; as all constraints present always have to be
validated.

120 Process synonyms, grants and schema compilation

The creation of private and public synonyms for the Oracle Health Insurance objects,
the distribution of grants to the Oracle Health Insurance database roles and
compilation of objects in the Oracle Health Insurance owner schema.

130 Import External Integration media

Importing of EI Media in the database.

This option only applies to OHI Back Office.

140 Import Discoverer EULs

Importing of Discoverer End User Layer(s).

This action has to be performed manually, which is why confirmation of its
execution is requested explicitly.

The Discoverer EULs are present in the (patch) release subdirectory `report`.

C Filesystem Installation

The (patch) release filesystem installation activities are executed through options in
this submenu.

C1 Complete filesystem installation WITHOUT compilation/generation

When this option is chosen, the below numbered filesystem installation activities will
be performed, *without* compilation/generation taking place (see the terms in brackets
for the numbered activities; these will *not* be executed here); *this will have to be done
at a later stage!*

C2 Complete filesystem installation WITH compilation/generation

When this option is chosen, the below numbered filesystem installation activities will
be performed, *including* compilation/generation taking place (see the terms in
brackets for the numbered activities; these *will* be executed here); *this completes the
filesystem installation!!*

200 Place OS modules

Placing OS modules in the `$OZG_BASE/sh` directory.

This option is not used to process the OHI Business Intelligence OS modules (see
205).

205 Place OHI Business Intelligence OS modules

Placing OHI Business Intelligence OS modules in the $OZG_BASE/sh directory.

210 Place SQL modules

Placing SQL modules in the $OZG_BASE/sql directory.

This option is not used to process the OHI Business Intelligence SQL modules (see 215).

215 Place OHI Business Intelligence SQL modules

Placing OHI Business Intelligence SQL modules in the $OZG_BASE/sql directory.

220 Place online help information

Placing online help information in the directory $OZG_BASE/help directory.

This option only applies to OHI Back Office.

230 Drop obsolete objects

The deletion of obsolete objects (sources & executables) on the filesystem.

Obsolete objects are objects that no longer apply. These objects have to be deleted from the runtime environment. The object source file and the object executable are deleted from the filesystem.

This only applies for objects not present in the database, but on the filesystem, only. Database objects are, when applicable, automatically deleted from the database installation and are not listed out either.

240 Place Java/XML-related files

Placing Java/XML-related files in the $OZG_BASE/java and $OZG_BASE/xml directories.

250 Place database DDL scripts

Placing database DDL scripts (for the creation of tables including the constraints, indexes and triggers concerned) in the $OZG_BASE/install directory.

The scripts are (for the time being) used when option 850 is chosen to partition table(s).

260 Place configuration files

Placing configuration files in the directory $OZG_BASE/conf directory.

270 Deploy ear files

Deploy ear files.

This option only applies to OHI Self Service. Depending on the initial choice to deploy the standard web application and/or the Portal web application one or two ear files are deployed.

300 (Compile and) place (300CP) Pro*C modules

The compilation (only in case of C2) and placing of the Pro*C modules in the `$OZG_BASE/bin` directory.

This option only applies to OHI Back Office.

400 Place Reports printer definitions

Placing of the Oracle*Reports printer definitions in the `$OZG_BASE/report` directory.

This option only applies to OHI Back Office.

410 (Compile and) place (410CP) Reports library

Compilation (only in case of C2) of the Oracle*Reports library.

Activity 410CP (is executed automatically as part of activity 410, but can also be (re)started separately) will subsequently place the files in the `$OZG_BASE/report` directory.

This option only applies to OHI Back Office.

420 (Generate and) place (420CP) reports

The compilation (only in case of C2) of the reports.

Activity 420CP (is executed automatically as part of activity 420, but can also be (re)started separately) will subsequently place the files in the `$OZG_BASE/report` directory.

This option is not used to process the OHI Business Intelligence reports (see 430).

This option only applies to OHI Back Office.

430 (Generate and) place (430CP) OHI Business Intelligence reports

Compilation (only in case of C2) of the OHI Business Intelligence reports.

Activity 430CP (is executed automatically as part of activity 430, but can also be (re)started separately) will subsequently place the files in the `$OZG_BASE/report` directory.

500 Place reference forms

Placing of the reference forms in the `$OZG_BASE/bin` directory.

This option only applies to OHI Back Office.

510 Place Forms object library

Placing of the Forms object library in the `$OZG_BASE/bin` directory.

This option only applies to OHI Back Office.

520 (Compile and) place (520CP) Forms libraries

Compilation (only in case of C2) of the Oracle*Forms library.

This option only applies to OHI Back Office.

Activity 520CP (is executed automatically as part of activity 520, but can also be (re)started separately) will subsequently place the files in the `$OZG_BASE/bin` directory.

530 (Generate and) place (530CP) forms and/or menu's

Compilation (only in case of C2) and placing of the forms and/or menus in the `$OZG_BASE/bin` directory.

Activity 530CP (is executed automatically as part of activity 530, but can also be (re)started separately) will subsequently place the files in the `$OZG_BASE/bin` directory.

This option only applies to OHI Back Office.

D Completion

The options in this submenu are used to perform activities that can be performed after the installation of 1 or more (patch) releases.

800 Compile/Generate objects in $OZG_BASE/bin

Compilation of sources in the `$OZG_BASE/bin` directory.

It is mandatory to execute this option when option C1 is chosen for the filesystem installation (no compilation/generation) and the options 300 or 520 or 530 are present. Compilation of *all* sources will then have to be chosen.
In other cases, this option is optional.

Even when other generic actions have been performed, e.g. the installation of a Developer Patch Set, it is still mandatory to execute this activity.
This is why this option is *always* visible in the menu.

This option offers the possibility to compile specific sources; the option to not compile all sources will have to be chosen, in that case.
Subsequently, a prefix and an extension can be given. When no value is given, no restriction applies for the determination of the sources to be compiled.

This option only applies to OHI Back Office.

Example 1 – give prefix and extension

Prefix `SYS11`, extension `pc` results in compilation of `SYS11*.pc`, causing source `SYS1108S.pc` to be selected.

Example 2 – give prefix

Prefix `OZGA` results in compilation of `OZGA*.*`, causing sources `OZGABOUT.fmb` and `OZGAMENU.mmb` to be selected.

Example 3 – give extension

Extension `pc` results in compilation of `*.pc`, causing sources `SYS1108S.pc` and `SYSS004S.pc` to be selected.

810 Compile/Generate all objects in $OZG_BASE/report

Compilation of all sources in the `$OZG_BASE/report` directory.

It is mandatory to execute this activity when option C1 is chosen for the filesystem installation (no compilation/generation) and the options 410 or 420 or 430 are present. Compilation of *all* sources will then have to be chosen.

In other cases, this option is optional.

Even when other generic actions have been performed, e.g. the installation of an Application Server Patch Set, it is still mandatory to execute this activity.
This is why this option is *always* visible in the menu.

This option offers the possibility to compile specific sources; the option to not compile all sources will have to be chosen, in that case.
Subsequently, a prefix and an extension can be given. When no value is given, no restriction applies for the determination of the sources to be compiled.

This option only applies to OHI Back Office.

> **Tip:** The compilation / generation steps 800 and 810 can be performed in parallel by starting the installation menu twice; activity 800 will be run in one session and activity 810 can run parallel in the other.

> **Tip:** When the timestamp/signature of a stored object, a screen and/or report refers to, is changed, the following error message will occur during use of the screen and/or report:

> ORA-04062: timestamp/signature of %s has been changed

> The following message is also possible, in that case:

> ORA-01403: no data found

> By recompiling the source(s) concerned, these messages are resolved.
> To *always* prevent this message, one can choose to always execute activities 800 and 810 after installation of a release.

820 Gather table/index statistics

Collecting table/index statistics under the Oracle Health Insurance owner account with the `$OZG_ADMIN/OZGISTAS.sql` script.

This option is also *always* visible in the menu; given that this activity will always have to be executed at the end of an installation. It is up to the installer to decide when this activity is to be performed.

Running this step on environments prior to release 2010.01.0.0000 cannot be performed in restricted mode.

830 Show Business Rules validation status

This option produces a report of the status of the validation of static Business Rules under the Oracle Health Insurance owner account.

This option is *always* mandatory; as all constraints present always have to be validated.

This option is also *always* visible in the menu; as this activity can always be executed during an installation.

This option only applies to OHI Back Office.

How to validate Business Rules

The actual validation *itself* is a batch in the OHI Back Office application ("Valideren Business Rules"), and should be started using the application menu.

Running this batch, that is; validating all Business Rules, may take a long time, therefore a parameter may be used to indicate how many hours this step may take at the maximum. If not all Business Rules can be validated in this period, the remainder may be validated at a later stage.
No later than *two weeks after the release goes live*, all Business Rules have to be validated.

A static Business Rule (= a complex constraint on a table) is always ENABLED for an Oracle Health Insurance database installation; i.e. the Business Rule applies to all *new* records.
To validate the Business Rule for the *existing* records, the batch is used.
The number of records that cannot be validated is indicated per table/Business Rule where applicable.

840 Drop obsolete columns

Deletion of obsolete columns under the Oracle Health Insurance owner account.

Obsolete columns are deleted *logically* during the database installation, which means that they can no longer be used (but are still present in the tables concerned).
This activity physically deletes the columns concerned.

This option is also *always* visible in the menu; as this activity can always be executed during an installation.

850 Partition/compress tables

The partitioning/departitioning or compression/uncompression of tables under the Oracle Health Insurance owner account.

A number of tables are recognized which may be partitioned (especially to benefit space management, performance (during installation) and to apply *Information Lifecycle Management*).

A partitioning key is determined (by Oracle) for these tables as well as a division into partitions.

Once partitioned or compressed, the same menu choice is available for departitioning/decompressing of the table.

For more information, see chapter Partitioning.

This option is also *always* visible in the menu.

860 Move tables/indexes

All tables and indexes must be stored in specific tablespaces. Warnings are raised during installation step 110 and during object validation (step 900) when tables and/or indexes are not in the correct tablespace. This option can be used to move tables and/or indexes to the correct tablespace.

This option cannot be used to move tables and/or indexes to an arbitrary tablespace.

This option is also *always* visible in the menu.

900 Perform object validation

Performing an Object Validation; used to check the presence, versions and status of the Oracle Health Insurance objects.

Choose to check the database objects, the filesystem objects or both.

This option is always mandatory.

Tables and indexes must be stored in specific tablespaces. When tables and/or indexes are not in the correct tablespace, then during the object validation warnings are reported. Installation step 860 can be used to move tables and/or indexes to the correct tablespaces.

Note that when the collection of statistics and the validation of Business Rules has not (yet) been performed/completed, this will be reported by the Object Validation.

☞   **Note:** Always *start* and *end* an installation session with an object validation.

L Show Logfile for Last Choice

Show the log file of the activity, which was chosen last.

M Automode

This option makes it possible to install the (patch) releases completely automatic.

The mandatory options in submenus A, B and C are executed automatically for the selected range of (patch) releases.

Automode can be enabled at any time; the installation will run automatically from that point onwards.

Automode will stop processing in case of an error.

Execute D menu activities in Automode

Automode will automatically skip the mandatory activities from the D submenu for each (patch) release it processes.
It will, however, keep track of what mandatory D menu activities should be executed.

When Automode finishes the automatic installation, it can then either

report the mandatory D menu activities that need to be executed, or

execute them.

This behavior is managed by the configuration file parameter
`ozg_run_dmenu_in_automode;`

When set to `N`, the mandatory D menu activities are reported, not executed (this is the default behavior).

When set to `Y`, the mandatory D menu activities are executed automatically.
Note that option 820 (for which the default choice is "N") is *not* executed automatically prior to 2010.02.0.0000 releases.

Conditions

Conditions for a completely automatic installation are that the following *manual activities* for the selected releases *must have been executed*

10

The release-specific parameter files have to be filled out previously for an automatic installation of database objects (submenu B).

20

The specific installation instructions have to be processed beforehand, because they may contain actions specified for a successful further installation of database- and/or customer objects.

140

The Discoverer EULs have to be imported beforehand, because these are manual actions and cannot be automatically verified to see if they have indeed been performed; a manual confirmation is requested explicitly.

Finally, it is recommended to run option 100 (running SQL check scripts) for *all* selected *(*patch) releases; given that the checks may not find anything, as else the database installation will not be executed.

O Skip Default Choice

The menu item is automatically preselected for "Make your choice …" (the so-called *default* choice); this is the first mandatory activity that has not yet been performed.
When an activity is performed, however, causing errors, *this* will always be shown first, as the errors have to be resolved first. This is why in that case the next activity is *not* automatically preselected.

There is always the option *not* to execute the default choice.
To prevent this from being the default choice (*within the same (patch) release*) *each time* for each following activity, this function can be used to indicate that this activity (within this (patch) release) should not (yet) be performed and should therefore not be selected as the default choice.

Example 1 – 120 postpone

When a range of patches is installed, the option exists to execute activity 120 only for the last patch.
By choosing O, when activity 120 is suggested as the default choice, the activity is skipped and (for this patch) no longer preselected as default.

Example 2 – 110 delivers errors

When activity 110 results in errors during patch installation, this activity will be suggested as the default choice (until errors have been resolved).
When the installation has to be completed, however (e.g. because the error concerns a known installation problem), O may be chosen again, so that the activity will be skipped (for this patch) and will no longer be preselected as default.

R Read Readme.txt File

Show the readme.txt file, the "plan of action for the installer".

S Reset Log File(s)

OZGPATCH determines whether or not an activity has been executed and the result of the possible execution of an activity based on the presence and content of the activity log file.

This option resets current log files for the selected activity (activities), i.e. a backup is made (conform reexecution of the activity, see section Log files).
This achieves that OZGPATCH sees the activity as not yet executed and will thus indicate if the activity has to be executed mandatorily or optionally.

By default, this option resets the logfile for the activity to which the *context* is set.

T Show Logfile Timing

Showing the log file in which the installation process' overall timing is registered.

A start and end time can be found for each chosen activity.
This log file can for instance be used to determine the total installation throughput time.

Warnings on mandatory activities which have not been executed for a specific (patch)release (e.g. on exiting the menu or navigating to a next (patch)release), are also recorded in this logfile.

U Update Client Releasetoken

Save the name of the *highest* release installed, for which (at least) the database installation has been successful, in the $OZG_BASE/ozg.conf configuration file.

This information is used by $OZG_ADMIN/OZG_CMD.sh to determine if a certain release has been installed. This data is needed to determine against which Oracle system software certain executables or utilities have to be started.
E.g. sqlplus has to be started up to (incl.) release 2009.03.0.0000 from the 10*g* database software, and from 2010.01.0.0000 from the 11*g* database software.

This activity is *automatically* performed by the installation menu in case of a successful database installation (activity 110) for a release.
The information with regards to the highest installed release is therefore not only available in the *database*, but also on the *filesystem* in the customer software in $OZG_BASE belonging to the release.
The activity can be performed manually, whenever desired, e.g. when the token has been changed/deleted as a result of manual management activities; the use of this activity will bring the highest release in the database and on the filesystem *in sync* again.

N Next (Patch) Release

When a range of (patch) releases is given during the start of OZGPATCH, it can be used to navigate to the next (patch) release.

Q Quit

Close the menu.

Z Create Logfiles Zip

Creation of a zip file with all installation log files present at that time, for the installation of the current release on the current database.

The zipfile is created in the $OZG_PATCH/<release> directory in the format <release>.log.<sid>.zip, e.g. 2009.02.2.0000.log.prod.zip.

$ OS Command Line

Starting an OS shell.
By leaving the shell, with an "exit" or "^D", you will return to the installation menu.

@ SQL*Plus (<owner>)

Starting SQL*Plus under the Oracle Health Insurance owner account.
Leaving SQL*Plus, with a "`quit`" or an "`exit`", you will return to the installation menu.

? Help

This submenu shows online help information regarding use of symbols, menu control, menu options and universal OZGPATCH options.

The version number of the used OZGPATCH version will also be shown.

LICENSE CHECK

An Oracle Health Insurance (patch) release contains all necessary installation objects for the different product lines (core + options).

When a certain installation activity does not apply based on the license present, this activity will not be shown.
When an activity does apply, but not all sources within this activity do, then only the sources for which a license is present will be installed.

FIRST USE ON AN ENVIRONMENT

The OZGPATCH installation menu has to be used for installations in an OHI Back Office, OHI Business Intelligence, or OHI Self Service environment.
Initially, the question "`Is this an OHI Business Intelligence environment?`" has to be answered with `Y` or `N`. When the answer is `N` the question "`Is this an OHI Self Service environment?`"has to be answered with Y or N. When the answer on both questions is N, the environment will be considered an OHI Back Office environment.

In case of an OHI Self Service environment two additional questions must be answered to determine which application type should be deployed: standard, portal or both.

The menu will automatically ensure that non-relevant menu options are not shown. Also, actions that do not apply (e.g. checks on the running batch scheduler) will not be executed.

☞ **Note:** During installation, checks are performed to validate the availability of an OHI BI license and the provided answers to the previous questions.

If things do not match, this will be signaled and you will be requested to restart the menu and re-answer the previous questions.

CONFIGURATION FILE

OZGPATCH uses the configuration file `$OZG_ADMIN/ozg.conf.$ORACLE_SID` in the background. For this file also applies that other (besides the installation account) accounts do not have rights (see section *File authorization*).

This file is automatically created when starting the installation menu for the first time.

Parameters

The table below indicates which parameters are present in the configuration file and what their content/function is.

Parameter	Description
ias_ctx_root	Context root of the standard (non-Portal) OHI Self Service application.
ias_pw	Application Server password under which the deployments are performed (only applicable for OHI Self Service).
ias_target	The context target of the OHI Self Service application.
ias_un	Application Server username under which the deployments are performed (only applicable for OHI Self Service).
ias_uri	Deployment URL needed for deployment of ear files of the standard (non-Portal) OHI Self Service application.
osi_deploy_wls	Applicable to OHI Self Service only • **N** Indicates that the standard web application ear file should *not* be deployed. • **Y** Indicates the standard web application ear file should be deployed.
osi_deploy_wlp	Applicable to OHI Self Service only • **N** Indicates that the Portal web application ear file should *not* be deployed. • **Y** Indicates the Portal web application ear file should be deployed.
ozg_begin_ree	The (patch) release chosen to start the most recent installation session.
ozg_eind_ree	The (patch) release chosen to end the most recent installation session.
ozg_eul	Indication Oracle Discoverer End User Layer: • **N** Indicates that no Discoverer End User Layer is used; even though it is possibly present in the (patch) release, it will not be installed. • **Y** Indicates that a Discoverer End User Layer is being used; when it is possibly present in the (patch) release, an option is shown which can confirm that it has been installed.

Parameter	Description
ozg_obd	Indication OHI Business Intelligence: • **OBD** Indicates that this ($ORACLE_SID) environment is an OHI Business Intelligence environment. • **N** Indicates that this ($ORACLE_SID) environment is an OHI Back Office environment and that no OHI Business Intelligence license is available; hence OHI Business Intelligence interface software will not be installed on this OHI Back Office environment. • **Y** Indicates that this ($ORACLE_SID) environment is an OHI Back Office environment and that that an OHI Business Intelligence license is available; hence OHI Business Intelligence interface software will be installed on this OHI Back Office environment. The values Y/N are automatically derived from the available licenses.
ozg_prd	Name of the product for the environment. Allowed values: • **OSI** indicates the environment is an OHI Self Service environment. • **OBD** indicates the environment is an OHI Business Intelligence environment. • **OZG** indicates the environment is an OHI Back Office environment.
ozg_run_dmenu_in_automode	• **N** Report mandatory D menu activities at the end of Automode installation • **Y** Automatically execute mandatory D menu activities at the end of Automode installation
ozg_820_hours_in_automode	Number of hours option 820 is allowed to run if this option is executed in Automode. Default is 1 hour.
rtr_pw	Oracle Warehouse Builder Runtime Repository user Password for the OHI Business Intelligence database account.
wlp_ctx_root	The context root of the Portal version of the OHI Self Service application.
wlp_uri	Deployment URL needed for deployment of ear files of the Portal version of the OHI Self Service application.
wlp_un	Weblogic Portal Server username under which the deployments are performed on the (only applicable for OHI Self Service).
wlp_pw	Weblogic Portal password under which the deployments are performed (only applicable for OHI Self Service).
wlp_target	The context target on Weblogic Portal of the OHI Self Service application.

■

SILENT INSTALLATION

It is possible to run the OZGPATCH installation menu in *silent* mode, which is without any manual intervention.

This enables the running of an installation session as called by a script, scheduling an installation using Unix crontab, etc.

To use silent installation, start the menu with the -silent tag, e.g.

$ OZGPATCH.pl –silent prod

When configured correctly, the installation will start immediately and fully automatic.

Automode

Internally, silent installation makes full use of the Automode feature; therefore all conditions that apply to Automode, apply to silent installation (e.g. perform manual activities before starting the automatic installation!).

Silent installation will also, like Automode, stop automatically when it runs into any error.

☞ In silent mode it is advised to set the ozg_run_dmenu_in_automode parameter to Y to fully automate the installation process.

☞ The difference between silent installation and Automode is that Automode is manually enabled (at any time) when an administrator uses the installation menu in interactive mode, while silent installation is a fully automated process from start to end (which enables automated scheduling & processing of (patch)release installations).

Conditions

Parameters

The ozg_begin_ree and ozg_eind_ree parameters in the configuration file *have to be specified* for silent mode; the installation menu uses the values of these parameters to determine at which (patch)release to start and end the automatic installation.

Chapter 5

Error Handling

GENERAL

Successful Installation

When the installation log files no longer contain errors, the installation of the (patch) release has been performed successfully.

Actions Based on Errors

If errors occur during a release installation, a number of steps have to be taken, depending on the location of the error.

Always ensure that a backup can be restored, for the database as well as the filesystem.

First determine if the error can be resolved by you. (e.g. in case of lack of disk space).
All (patch) releases can, unless stated otherwise, be restarted; once the cause of the error has been resolved, the (patch) release will have to be installed again.

If an error cannot be resolved, first determine if is a known installation problem; consult the BeehiveOnline website to check.
When it is not a known problem please contact Oracle Health Insurance Support. They can determine what the consequences are and if the release installation can be continued. It may be possible to resolve the issue over the phone, if not, an incident has to be reported which will be dealt with urgently.
When an error involved an `ORA-600` (i.e. `ORA-006XX`) or `ORA-07445` message, this error *always* has to be reported with a Service Request at Oracle Support Services too.

RESTORE DELIVERIES

When errors occurring during a (patch) release installation have to be resolved by Oracle, this will be dealt with by:

A fix in a later (patch) release.
Not only restored errors will be delivered again, other parts of the installation which have not been installed as a result of the errors, will be too.

A fix by redelivery of the (patch) release.
When a (patch) release cannot be installed as a result of a blocking error (installation will not continue), the complete patch (release) will be redelivered.
The patch (release) will then be redelivered to MOS Patches & Updates.
Reinstallation only has to be executed by customers who had errors. When no errors occurred during the installation of the original release, the new delivery does not

have to be reinstalled .An explanation will be placed on MOS Patches & Updates for each redelivery, for the (patch) release concerned, stating ' Known installation problems'

Example

When in the 2009.01.0.0000 release, a SQL script is delivered and the customer encounters performance problems during installation such that the installation cannot be completed within reasonable limits.

Oracle will fix the SQL script; the modified script will be delivered in a subsequent patch. E.g. in 2009.01.0.0001.

Apart from the fix in patch 2009.01.0.0001 the patch/release which contained the original script will also be redelivered at the customer's requests.
A new patch/release (2009.01.0.0000) will in that case be made available on BeehiveOnline stating 'Known installation problems'.
It is not mandatory for customers to install this redelivered patch. They may also use the subsequent patch (2009.01.0.0001).

Customers experiencing problems are offered a viable way to overcome the problem. Customers who do not experience problems do not have re-install an earlier patch/release.

Note: When, after the original 2009.01.0.0000, the 2009.01.0.0001 has already been installed and the decision is made to install the adapted 2009.01.0.0000, then all patches delivered after that patch concerned have to be redelivered, too. So, after the restored 2009.01.0.0000 the 2009.01.0.0001 has to be reinstalled, too.

Preconditions

The above construction is ONLY used for SQL scripts causing insurmountable performance problems. When a script for instance runs longer than 8 hours, without any indications of completion of the script, a request may be made for the redelivery of a (patch) release. When a script contains an error which is corrected in a later patch (when there is no performance issue) NO redelivery will take place. Only when a situation occurs which cannot be resolved by a delivery in a later patch, will this mechanism be used for errors.

In relation to the times stated in the preconditions, it may differ per script what does and what does not take a long time. The customer has to clearly mention why a patch redelivery request is justified. Based on this, Oracle will decide if the request will be honored.

Chapter 6

Specific Installation Instructions

PREPARATION & COMPLETION

Before a release is installed in an environment, a number of activities can and have to be performed to prepare the environment.
A good preparation can reduce the actual installation time (as it can 'keep running' without interruptions).

Once a release has been installed, a number of things have to be done *before the environment can be released to the end users*.

Preparation

Following activities can be performed in preparation of the release installation.

It is not required to wait with these activities until the actual installation of the (patch)release; the mentioned activities can all be performed (long) before.

1. Ask Functional Specialists to provide the values for parameters in the `readme.<PRODUCT>` files.
   These values can be entered in activity 10 in the installation menu.

2. Perform the installation instructions.
   This is activity 20 in the installation menu.

3. Run the check scripts twice: 1x well before the release is installed (so that there is ample time to solve any issues and so that this does not affect actual installation time), and 1x right before the release installation to determine if no data have been added which violates the checks.
   This is activity 100 in the installation menu.

   Note that during the database installation (activity 110 in the installation menu) the check scripts are rerun automatically; when this results in errors, the database installation stopped.

4. Perform **Object Validation** to determine if the environment is initially correct and fix reported problems before installation
   This is activity 900 in the installation menu.

   Start the **Object Validation** *via the installation menu* to check if all is set up correctly for *installation*.

Completion

After installing (patch)releases, execute following activities:

1. Perform the activities from the installation submenu D (Completion).

2. Switch the database from installation mode to normal mode (e.g. by unsetting `parallel` settings).

3. Start the application to check that the environment can be accessed.

4. Start the **Object Validation** *via the application* to check if all is set up correctly for *runtime*.

5. Fix any remaining errors; if this is not possible, contact Oracle Health Insurance Helpdesk to decide if the errors are blocking (and the environment can thus not be delivered to the end users), or non-blocking.

INSTALLING DESIGNER EXPORTS

Introduction

For the development (maintenance and new build) of OHI Back Office the CASE tool Oracle*Designer is used. In Oracle*Designer functional (e.g. in the form of Business Functions) and technical information (e.g. table definitions) is stored for the OHI Back Office modules and data structures.

This information is made available to customers, for each Oracle Health Insurance major release the Oracle*Designer repository is exported.

The exported information is stored in dump files, which are merged in a zip file. The naming convention for this zip file is d2<yymmdd>.zip, where <yymmdd> stands for the date the file has been created. For example: d2000315.zip.

This file is available on the BeehiveOnline website. It has to be unzipped by the customer after which the dump file can be imported in the company's Oracle Designer repository. If this dump file has been read in before, the 'containers' concerned will already exist. In that case, these have to be deleted or renamed first.

Activities

How does importing into the company's own Oracle*Designer (customer) environment work?

1. Starting up Repository Object Navigator
   Start Oracle*Designer and log in as the Repository Owner.
   Start the Repository Object Navigator (RON).

2. Delete existing Containers
   Existing Oracle Health Insurance Containers can be deleted with menu option **Utilities/Force Delete**.

3. Read-in export
   The dump file can now be read in. For this, choose the menu option **Utilities/Import**.

INSTALLING AN INITIAL RELEASE

Introduction

An initial release is used for the creation of an empty Oracle Health Insurance environment. A special patch will be delivered on request containing all sources needed for an empty Oracle Health Insurance environment.
Empty means that only a few system tables (e.g. system messages) are populated.

For the installation of an initial release different installation activities apply than for the installation of a regular release.

Activities

Install the database and application server software, create a database and create the correct OHI directory structure on the file system. See the following document for more information:

Oracle Health Installation, Configuration and DBA Manual

Execute the steps below to create an empty Oracle Health Insurance environment (an 'empty' OHI database containing the complete data structure with only the initial seed data for starting the application):

1. Unzip the <release>.zip file in the usual way in `$OZG_PATCH`.

2. Change directory to the <release> directory.

3. Run the following script with SQL*Plus under the `SYS` account to create generic objects, accounts, roles, directories, privileges, etc.:

    ```
    @sql/OZGI001S.sql (for OHI Back Office)

    @sql/OBDI001S.sql (for OHI Business Intelligence)

    @sql/SSEI001S.sql (for OHI Self Service)
    ```

    First, set the Unix environment to the correct OHI database with

    ```
    . ozg_init.env <sid>
    . ozg_init.env $OZG_ORATAB_DB10G
    ```

    Example:

    ```
    . ozg_init.env prod
    . ozg_init.env $OZG_ORATAB_DB10G
    ```

4. Run the following script (in the $OZG_PATCH/<release> directory) to create the objects needed for the installation. Run this script with SQL*Plus under the Oracle Health Insurance owner account (e.g. `OZG_OWNER` for OHI Back Office, `OBD_OWN` for OHI Business Intelligence or `SSE_OWNER` for OHI Self Service):

    ```
    @sql/OZGI004S.sql (for OHI Back Office)

    @sql/OBDI004S.sql (for OHI Business Intelligence)
    ```

```
@sql/SSEI004S.sql (for OHI Self Service)
```

First, set the Unix environment to the correct Oracle Health Insurance database with

```
. ozg_init.env <sid>
```

Example:

```
. ozg_init.env prod
```

After this a list of plsql objects will remain invalid, you can ignore this. During the further installation additional missing objects will be created so these objects can become valid.

5. In case of an OHI Back Office installation, run the following script (in the initial release directory) to create the batch queue. Run this script with SQL*Plus under the Oracle Health Insurance owner account (e.g. OZG_OWNER for OHI Back Office):

```
@sql/OZGI006S.sql
```

First, set the Unix environment to the correct Oracle Health Insurance database with

```
. ozg_init.env <sid>
```

Example:

```
. ozg_init.env prod
```

6. In case of an OHI Business Intelligence installation, create a private database link named SRC_OPENZORG to a valid OHI Back Office schema in the OHI Business Intelligence schema (OBD_OWN) with the command below:

```
create database link SRC_OPENZORG
connect to [name OHI Back Office account]
identified by [password OHI Back Office account]
using '[sid]'
                                    /
```

For example:

```
create database link SRC_OPENZORG
connect to ozg_owner identified by ozg_owner
using 'prod'
/
```

7. Copy from $OZG_PATCH/<release>/admin to $OZG_ADMIN.

    *   OZGPATCH.pl
    *   OZGPLIB.pm
    *   OZG_CMD.pl

8. Copy from $OZG_PATCH/<release>/sql/ to $OZG_BASE/sql.

    *   OZGC001S.sql
    *   OZGC003S.sql
    *   OZGC007S.sql
    *   OZGCOMPILE.ins

Depending on the environment copy the files:

- OHI Back Office
  - OZGSYNONYMS.ins
  - OZGGRANTS.ins
- OHI Business Intelligence
  - OBDSYNONYMS.ins
  - OBSGRANTS.ins
- OHI Self Service
  - SSESYNONYMS.ins

9. In case of an OHI Back Office installation copy from `$OZG_PATCH/<release>/sh/` to `$OZG_BASE/sh`.

- OZG_PROC.sh

10. In case of an OHI Back Office installation copy from `$OZG_PATCH/<release>/admin` to `$OZG_ADMIN` all files which have name that start with the prefix below (so use d11g* to identify them all).

- d11g

11. Create an empty file `ozg.conf` in folder `$OZG_BASE`.

12. Start `OZGPATCH.pl` and perform the (patch) release installation.

You will be asked what kind of environment you are installing. A default username and password will be suggested. Be aware that the suggested defaults are in uppercase. These will not be accepted when you use case sensitive usernames and passwords in the database and the owner account is in small case.

Make sure you did install the application server software and did run at least the configuration wizard to create the `$ORACLE_INSTANCE` environment. This folder structure is needed to make the check on the Developer software version succeed.

Run option U so the menu places relevant information in file $OZG_BASE/ozg.conf.

During installation of the database objects the message will appear that the previous release has not been installed. This message can be ignored. For other messages, the usual procedure applies.

EARLIER INSTALLATION OF PATCH RELEASES

Oracle may in specific cases indicate that it is necessary to install a delivered patch release before other, previously released, patch releases. In this case, the following rules apply:

1. Early installation of a patch is only permitted when written permission has been given by Oracle. This permission is sent via e-mail and is registered by Oracle.

2. The permission only applies to the patch indicated and not to any subsequent patches.

3. The re-creation of a patch at the customer's request (customer-specific delivery of one or more incidents) is not supported.

Chapter 7

Performing Installation of Multiple (Patch) Releases

Using the installation menu OZGPATCH, it is possible to install a range of (patch) releases sequentially.

METHODS

When installing a range of (patch) releases, 2 methods can be used:

1 - Automode

When running in Automode, all options will be performed automatically. Automode will stop when an error occurs.

The main advantage of this mode is that no manual actions are necessary (only in case of errors or manual activities, see Automode), so this is the easiest way of installing (patch) releases.

2 - Manual Installation

The main advantage of this option is that a few steps can be skipped when installing multiple (patch) releases.
This means manual installation for a large number of large (patch) releases can save installation time.

The installation procedure in case of manual installation should be:

1. For every (patch) release in the range:

2. Perform all installation activities in submenu's A and B.

3. For submenu C, *only* use option C1
   This will prevent intermediate compilation/generation of executables.

4. At the end of the complete (range) installation, perform all activities in the D submenu.
   In submenu D, activities 800 and 810 will then take care of the compilation/generation of all executables.

ERRORS DURING INSTALLATION

When installing a *range* of (patch) releases, it is important to understand all errors (that may occur *during* the installation process) should be gone *at the end* of the complete (range) installation.

For example, when installing patches 1 to 10, an error may occur in patch 3. However, if this error is fixed in patch 8, the error will *automatically be fixed during the installation of the range* of (patch) releases.

It is therefore *not necessary and not required* to contact the Oracle Health Insurance Helpdesk for these *intermediate* errors; only errors that (still) exist <u>at the end</u> of the complete (range) installation must be reported (if they cannot be solved by the DBA; see Error handling).

Appendix A

Partitioning

INTRODUCTION

Conditions

If Partitioning is used in an Oracle database, 2 conditions have to be met:

1. A license for the *Oracle Partitioning Option* is required;

2. The Oracle Partitioning Option has to be installed.

When the Partitioning Option is installed but *is not used*, it can be **deinstalled** using the *Oracle Universal Installer*. When the Partitioning Option is not installed, only the **departitioning** functionality of this menu choice is available.
As long as the Partitioning Option is active but not used, a warning will appear on installation of Oracle Health Insurance (patch) releases.

OHI Business Intelligence

Partitioning is required for OHI Business Intelligence.

OHI Back Office

For OHI Back Office Partitioning is supported from release 2007.01.0.0000 on. Partitioning is *optional*; the customer has the choice if partitioning is to be used or not.

Partitioning offers advantages in the following areas:

1. The time needed for the database installation activities in Oracle Health Insurance (patch) releases will decrease (e.g. as a result of the creation of several smaller partitioned indexes instead of 1 large index);

2. The OHI Business Intelligence runtime performance will improve (e.g. because more efficient access to certain data, present in a certain partition, is possible);

3. The ability to implement Information Lifecycle Management (with the possibility to place/move data (= partitions) to secondary storage).

OHI Self Service

Partitioning is not yet supported for OHI Self Service.

FUNCTIONALITY

OHI Business Intelligence

For OHI Business Intelligence the tables are created partitioned in the designated table spaces.

See also

Oracle Health Insurance - OHI Business Intelligence Administrator Reference

OHI Back Office

For OHI Back Office the tables are *created* non-partitioned, but that they can be *partitioned* at any time in the *future*. These partitions can then be moved (by the DBA) to other tablespaces. The tables are partitioned by running installation menu activity 850.
When this option is used, the database has to be in installation mode (i.e. restricted mode, no end users, parallel mode, specific parameter settings, etc.).

N.B.: This script uses NOLOGGING. A full backup has to be made BEFORE and AFTER it has run.

The tables are partitioned in the tablespace in which *they belong* (this information is present in the `ALG_TABELLEN` table); for an overview of the (18) mandatory OHI Back Office tablespaces see:

Oracle Health Insurance Installation, Configuration and DBA Manual

The DBA can move the partitions created to *other* tablespaces. The name of the target tablespace must be equal to the *source table space + suffix* name, e.g. source table space `OZG_FACT_ZRG_TAB` and partition table space `OZG_FACT_ZRG_TAB`**`_PARTQ42006_DECL`**.

Note that the objects, once partitioned, can also be de*partitioned* using the same option 850.

Oracle ILM Assistant

When the partitions need to be divided over a number of table spaces to enable Information Life cycle Management, Oracle advises to use *Oracle ILM Assistant.*

Oracle ILM Assistant is a GUI tool to manage the ILM environment. The tool is free and can be downloaded from OTN. To use ILM Assistant *Oracle Application Express* is required. This product is free as well and can also be downloaded from OTN.

The tool offers the possibility to set *lifecycle* and *retention policies*, to *classify* data, and gives advice for moving data (*data migration*) to another storage tier.
In the current release the tool does not perform the actual migration; it generates *scripts* to do this.

For more information on this tool, i.e. screenshots, please refer to OTN (Oracle Technology Network).

Also see this documentation:

Oracle Information Lifecycle Management OTN website:
http://www.oracle.com/technology/deploy/ilm/index.html

Information Lifecycle Management for Business Data
An Oracle White Paper (available on OTN)
August 2006

Implementing ILM using Oracle Database 10g Release 2
An Oracle White Paper (available on OTN)
August 2006

Information Lifecycle Management (ILM) for Business Data
Presentation (available on OTN)
October 2006

Implementing ILM using the Oracle ILM Assistant
Presentation (available on OTN)
October 2006

Installation Checklist

Following is a *quick reference checklist* for installing (a range of) OHI (patch)releases.
All steps are hyperlinks to the relevant paragraphs in this document.

1 - Get the Release(s)

Decide which (patch)release(s) to install.

Download the releases.

Unzip the files in $OZG_PATCH directory on the application server.

2 - Prepare for Installation

Make sure non-validated Business Rules are validated.

Make sure statistics have been gathered.

Make a backup.

Read *Known Bugs* list(s) for the release(s) on BeehiveOnline.

Get functional input to fill the parameter file for the release(s).

Read *Specific installation instructions* for the release(s).

Configure the database & application server for installation.

Fill in requirements for major release installation, including Parallel Execution.

Start the installation menu.

Perform *Object Validation* in installation mode.

3 - Perform the Installation(s)

Perform installation of (all) the release(s).

Use Automode or Silent Mode if required.

Fix errors & rerun installation(s) if necessary.

4 - Complete the Installation(s):

Reset the database & application server to runtime mode.

Gather statistics.

Make sure non-validated Business Rules are validated.

51

Perform *Object Validation* in runtime mode.

Install Designer exports.

Address *all* remaining errors.

Appendix C

Database Parameters checkup

Before the actual database installation in step 110 of the patch menu is performed, the database installation is checked on many recommended and mandatory parameter settings.

This appendix gives an overview of all checks performed, one generic checkup and one specific checkup per product (OHI Back Office, OHI Business Intelligence and OHI Self Service). Combine both parts per product to have the complete checkup.

Checks are performed with output written to console and log file. Each check reports its severity either as FATAL, ERROR, WARNING or INFO.

Generic OHI checks

FATAL	The variable OMGEVING in configuration file "ozdefine.sql.<ORACLE_SID>" must be either 'A' or 'P'
FATAL	The variable OZGPATCH_SUPPRESS_WARNING in configuration file "ozdefine.sql.<ORACLE_SID>" must be either 'N' or 'Y'
FATAL	Installed release must be higher than current present release
FATAL	Installation cannot performed as user SYS or SYSTEM
FATAL	Oracle RDBMS Enterprise Edition 11.2.0 is required, minimum patchset 11.2.0.3
FATAL	Oracle Component XDB version 11.2.0 must be installed and VALID
FATAL	Oracle Component JAVAVM version 11.2.0 must be installed and VALID
FATAL	During non-Major installations in ARCHIVELOG mode; FORCE_LOGGING=NO is not allowed
FATAL	Any depricated database parameter must be defaulted (e.g. not set) on all instances
FATAL	Database parameter _COMPLEX_VIEW_MERGING must be defaulted (e.g. not set) on all instances
FATAL	Database parameter event 10195 may not be set on any instance
FATAL	Database parameter JOB_QUEUE_PROCESSES must be >= 10 on all instances
FATAL	When ASMM is not used; database parameter LARGE_POOL_SIZE must be >= 16MB on all instances
FATAL	Database parameter NLS_SORT must be set to BINARY on all instances
FATAL	Database parameter OPEN_CURSORS must be >= 500 on all instances
FATAL	Database parameter PGA_AGGREGATE_TARGET must be > 0 on all instances
FATAL	Database parameter PROCESSES must be >= 200 on all instances

FATAL	Database parameter SESSION_CACHED_CURSORS must be >= 500 on all instances
FATAL	Database parameter STATISTICS_LEVEL must be set to TYPICAL on all instances
FATAL	Database parameter UNDO_MANAGEMENT must be set to AUTO on all instances
FATAL	Any other not prescribed optimizer parameter must not be set on all instances
FATAL	Database parameter _OPTIMIZER_JOIN_FACTORIZATION may not be set (e.g. =FALSE) on all instances
FATAL	The Default Tablespace of the schema owner must exist and be available
FATAL	The Temporary Tablespace of the schema owner must be equal to the DEFAULT_TEMP_TABLESPACE of the database
FATAL	Database system statistics must have been collected
FATAL	All tablespaces must be locally managed
FATAL	The SYSTEM tablespace must use extent allocation type SYSTEM or USER
FATAL	In RAC; Partitioning option must be consistant across all instances
ERROR	During installation; non-RESTRICTED sessions are not allowed
ERROR	User (e.g. Functionarissen) sessions are not allowed
ERROR	Database parameter OPTIMIZER_INDEX_COST_ADJ must be 25 on all instances
ERROR	Database parameter OPTIMIZER_FEATURES_ENABLE must be at least '11.2' on all instances
ERROR	No PLAN_TABLE table may exist for the schema owner
ERROR	All data dictionary statistics must have been created / gathered
ERROR	All fixed objects statistics must have been created / gathered
ERROR	All mandatory Object Privilege must be present; see "Mandatory Privilege" overview
WARNING	Reported when installed release is already present
WARNING	Reported when the required previous release has not been installed before
WARNING	Reported when objects are still invalid after (re)compilation
WARNING	Reported during Major installations in ARCHIVELOG mode; FORCE_LOGGING=NO is not recommended
WARNING	Reported when database parameter DB_CACHE_SIZE is not set to the recommeded multiple of 16Mb on all instances
WARNING	Reported when database parameter RESOURCE_MANAGER_PLAN is not set to the recommended SYSTEM_PLAN or MIXED_WORKLOAD_PLAN on all instances

WARNING	When ASMM is not used; reported when database parameter SHARED_POOL_SIZE is not set to the recommended multiple of 16Mb on all instances
WARNING	Reported when database character set is not set to the supported, either WE8ISO8859P1 or WE8ISO8859P15 or AL32UTF8
INFO	Reported if ARCHIVELOG mode is used
INFO	Reported when FORCE LOGGING is set to YES
INFO	Reported all compressed objects, and recommendation in case to re-apply compression
INFO	Reported the number of SQL Profiles found / registered
INFO	Reported (first 20 by name) objects with missing dictionary statistics

--Info: Table last updated OHI release 2012.03

Specific OHI Back Office checks

FATAL	Queue SYS_BATCH_QUEUE must exist and belong to the OZG schema owner
FATAL	Database parameter AQ_TM_PROCESSES must be 1 on all instances
FATAL	Database parameter DB_BLOCK_SIZE must be 8k
FATAL	Database parameter DML_LOCKS must be >= 500 on all instances
FATAL	Database parameter OPTIMIZER_MODE must be set to FIRST_ROWS_10 on all instances
FATAL	During runtime; database parameter PARALLEL_MAX_SERVERS must be set to 0 on all instances
FATAL	During installation; database parameter PARALLEL_MAX_SERVERS must be set > 1 on all instances
FATAL	During runtime; database parameter WORKAREA_SIZE_POLICY must be set to AUTO on all instances
FATAL	During installation; database parameter WORKAREA_SIZE_POLICY must be set to MANUAL on all instances
FATAL	All user tablespaces must use ASSM, 8K block size and SYSTEM extent allocation type
ERROR	All mandatory Privileges for OZG schema owner must be present; see "Mandatory Privilege" overview
ERROR	All mandatory Privileges for BATCH account must be present; see "Mandatory Privilege" overview
ERROR	All mandatory Privileges for REPORTS account must be present; see "Mandatory Privilege" overview

ERROR	All mandatory Privileges for OBD_SELECT_USER account must be present; see "Mandatory Privilege" overview
WARNING	Reported the number of not validated Business Rules
WARNING	Reported when Partitioning Option is installed, but not or partially used; recommended to execute step 850
WARNING	Reported when the tablespace for the partitioned table is not correct
WARNING	Recommendation to set the retention time for queue SYS_BATCH_QUEUE to two weeks
WARNING	Reported on any table or index with incorrect tablespace
WARNING	Recommendation to set INI_TRANS to value >=4 for all normal data tables (e.g. tablenames without '$' on 4th position)
WARNING	Recommendation to set PCT_FREE to value 15 for all normal data tables (e.g. tablenames without '$' on 4th position)
WARNING	Recommendation to set PCT_FREE to value 5 for mutation log tables (e.g. tablenames with '$' on 4th position)
WARNING	Recommendation to revoke all granted but unnessessary privileges for OZG schema owner
WARNING	Recommendation to revoke all granted but unnessessary privileges for BATCH account
WARNING	Recommendation to revoke all granted but unnessessary privileges for REPORTS account
WARNING	Recommendation to revoke all granted but unnessessary privileges for OBD_SELECT_USER account

--Info: Table last updated OHI release 2012.03

Specific OHI Business Intelligence checks

FATAL	Database parameter DB_BLOCK_SIZE must be either 8k or 16k
FATAL	Database parameter OPTIMIZER_MODE must be set to ALL_ROWS on all instances
FATAL	Database parameter PARALLEL_MAX_SERVERS must be set > 1 on all instances
FATAL	Database parameter STAR_TRANSFORMATION_ENABLED must be set to TRUE on all instances
FATAL	Database parameter WORKAREA_SIZE_POLICY must be set to AUTO on all instances
FATAL	All user tablespaces must use ASSM, either 8K or 16k block size and must use either SYSTEM or UNIFORM extent allocation type
ERROR	Only one database link to OZG may exist
ERROR	OHI BackOffice release must correspond to the Business Intelligence release
ERROR	The job ALG_PERL_JOB must exist

ERROR	All mandatory Privileges for OBD schema owner must be present; see "Mandatory Privilege" overview
WARNING	Reported when the most recent OHI BI data load process did not complete (successfully)
WARNING	Recommendation to set PCT_INCREASE to value 0 for all normal data tables (e.g. tablenames without '$' on 4th position)
WARNING	Recommendation to revoke all granted but unnessessary privileges for OBD schema owner
INFO	Reported database link SRC_OPENZORG to which host
INFO	Reported the OHI BackOffice release found
INFO	Reported the status of the OHI BackOffice data load process

--Info: Table last updated OHI release 2012.03

Specific OHI Self Service checks

FATAL	All SSE% and SSR% tablespaces must use ASSM
ERROR	All mandatory Privileges for OSI schema owner must be present; see "Mandatory Privilege" overview
WARNING	Database parameter DB_BLOCK_SIZE must be 8k
WARNING	Recommendation to revoke all granted but unnessessary privileges for OSI schema owner

--Info: Table last updated OHI release 2012.03

Appendix D

Database Mandatory User Privileges

During the Database Parameters checkup, described in Appendix C, mandatory privileges for system/owner accounts are checked. Missing mandatory privileges are reported with an ERROR, unnessessary privileges are reported as WARNING.

This appendix gives an overview of all mandatory user privileges required for OHI to function correctly.

PrivType	Privilege	BackOffice Owner	Business Intelligence Owner	Self Service Owner	BATCH account	REPORTS account	OBD_SELECT_USER
DB Priv	ADVISOR	x	x	x			
DB Priv	ALTER SESSION	x	x	x	x	x	x
DB Priv	CREATE JOB	x	x	x			
DB Priv	CREATE PROCEDURE	x	x	x			
DB Priv	CREATE PUBLIC SYNONYM	x	x	x			
DB Priv	CREATE SEQUENCE	x	x	x			
DB Priv	CREATE SESSION	x	x	x	x	x	x
DB Priv	CREATE SYNONYM	x	x	x			
DB Priv	CREATE TABLE	x	x	x			
DB Priv	CREATE TRIGGER	x	x	x			
DB Priv	CREATE VIEW	x	x	x			
DB Priv	DROP PUBLIC SYNONYM	x	x	x			
DB Priv	MANAGE SCHEDULER	x	x	x			
DB Priv	RESTRICTED SESSION	x	x	x			
DB Priv	SELECT ANY DICTIONARY	x	x	x	x		
DB ROLE	SELECT_CATALOG_ROLE	x	x	x			
DB Priv	CREATE DATABASE LINK		x				
DB Priv	CREATE DIMENSION		x				
DB Priv	CREATE EXTERNAL JOB		x				
DB Priv	ALTER SYSTEM	x					
DB ROLE	AQ_ADMINISTRATOR_ROLE	x					
DB Priv	CREATE ANY CONTEXT	x		x			
DB Priv	CREATE TYPE	x					
DB Priv	DROP ANY CONTEXT	x		x			
DB Priv	JAVAUSERPRIV	x					
DB Priv	RESOURCE			x			

PrivType	Privilege	BackOffice Owner	Business Intelligence Owner	Self Service Owner	BATCH account	REPORTS account	OBD_SELECT_USER
OHI ROLE	OZG_ROL_BATCH				x		
OHI ROLE	OBD_SELECT_ROLE						x
SysObjPriv	EXECUTE ON DBMS_ALERT TO <user>	x	x	x			
SysObjPriv	EXECUTE ON DBMS_SYSTEM TO <user>	x	x	x			
SysObjPriv	EXECUTE ON DBMS_WORKLOAD_REPOSITORY TO <user>	x	x	x			
SysObjPriv	EXECUTE ON DBMS_LOCK TO PUBLIC	x					
SysObjPriv	EXECUTE ON DBMS_AQ TO <user>	x					
SysObjPriv	EXECUTE ON DBMS_AQADM TO <user>	x					
SysObjPriv	SELECT ON V_$DATABASE TO OBD_SELECT_USER						x
SysObjPriv	EXECUTE ON DBMS_CRYPTO <user>	x					

--Info: Table last updated OHI release 10.13.1