Guide

Preferred Oracle Configuration

SimCorp Solutions
Based on version All
January 2019

SimCorp

PUBLISHED BY

SimCorp A/S
Weidekampsgade 16
2300 Copenhagen S
Denmark

Published	January 2019
Writer:	TECH-DBPERF
Based on:	All

If you have any comments to the contents of this document or suggestions for topics to be included in a future updating of this document, please do not hesitate to send them by mail to: SCDimension.doc@SimCorp.com.

Contents

1 Introduction

This document describes how to set up the Oracle database according to SimCorp's recommendations for running SimCorp Dimension. The Preferred Oracle Configuration is designed to provide a solid base for SimCorp Dimension installations and it is constantly being evaluated and refined to make sure that the database is configured optimally for SimCorp Dimension. The goal is to ensure that the database is set up so that application problems due unsuitable configurations might be avoided. The parameter values advised here come from a thorough knowledge of how the application interacts with the database and a proven usage by most clients and internally in SimCorp on several hundred test installations.

The document covers the SimCorp Dimension supported Oracle12c. Please refer to the ***SimCorp Dimension System Administrator's Manual*** for information regarding the exact Oracle version and minimum patch level required for the different versions of SimCorp Dimension. The manual also contains information about requirements for Oracle edition and options in relation to SimCorp Dimension feature usage.

The reader is assumed to be an experienced Oracle DBA as this document does not provide details of basic administrative tasks, but rather relies on existing Oracle documentation being followed where necessary. The reader is therefore also assumed to have access to My Oracle Support and Oracle Technology Network.

Note, for environments utilizing more advanced features, for example a Data Guard configuration, it may be the procedures described in this document require special consideration given the actual context. In this case the relevant Oracle documentation must be followed.

Chapter Planning the Database Configuration on page 9 covers basic setup considerations when configuring a database environment for SimCorp Dimension.

Chapter Oracle Software and Database Installation on page 19 provides guidelines for installing, creating and configuring a new database ready for SimCorp Dimension installation. The installation of the SimCorp Dimension application itself is not covered by this document.

Chapter Configuring Oracle RAC for SimCorp Dimension on page 21 details special considerations for installing and configuring a RAC database for SimCorp Dimension.

Chapter Oracle Upgrade and Patching on page 27 gives guidelines specifically for SimCorp Dimension when upgrading Oracle from version 12.1 to 12.2 and when patching Oracle.

Chapter Preferred Configuration for SimCorp Dimension on page 30 lists and explains those default Oracle configurations requiring change, in order to comply with the SimCorp Dimension Preferred Oracle Configuration.

Chapter System Operations on page 57 provides a few general notes in relation to administration and operation of the database.

The appendix contains a list highlighting the points from the different chapters and what constitutes the Preferred Oracle Configuration for SimCorp Dimension.

2 Planning the Database Configuration

Before creating the database for SimCorp Dimension you will need to consider, and take decisions in regards to the storage required by the database, the database character set to be used and the memory allocations for the database.

2.1 Storage

2.1.1 Oracle Software

It is possible to install Oracle on a host which is already running older versions of Oracle. If there are other versions of Oracle installed on the host, the Oracle software must be installed in a separate ORACLE_HOME. ORACLE_BASE is the root of all the Oracle software and otherwise related folders. ORACLE_HOME will therefore be placed in ORACLE_BASE. You should choose an ORACLE_HOME on a physical drive, which does not contain your operating system, swap files, or data files. For instance, on a Windows system your c: drive would normally contain the Windows-directory, your d: drive could contain your application files, which makes it a suitable choice for ORACLE_BASE, and the drive letters e: and above could be dedicated to data files.

For production systems, it is highly recommended that you install the Oracle instance on a dedicated disk sub-system optimized for online transaction processing I/O throughput. For ASM, please see section Automatic Storage Management (ASM).

2.1.2 Automatic Storage Management (ASM)

SimCorp Dimension has at this moment only been tested with ASM in a very limited manner (i.e. in connection with tests of SimCorp Dimension on RAC). But since ASM is treated in a black-box manner, ASM can be used if desired. If choosing to use ASM please refer to the points listed in chapter Configuring Oracle RAC for SimCorp Dimension regarding configuration of ASM (even if using a single instance database).

2.1.3 Tablespaces

With the database creation the mandatory tablespaces SYSTEM, SYSAUX, UNDO, and TEMP are created. If you use a custom method to create the database, you must ensure that they are all locally managed. SYSAUX and other utility tablespaces (e.g. USERS) should in addition be configured to use Automatic Segment Space Management (ASSM).

Typically, a high-end storage array is used for database storage and the file locations are typically less of an issue on a SAN than on ordinary external disks. You can consider separating groups of data files physically to avoid I/O bottlenecks. The load on TEMP and UNDO can be high so moving these away from the SimCorp Dimension data files is likely to be beneficial.

Creation of the SimCorp Dimension tablespaces is done with the installation of SimCorp Dimension, which is not covered by this manual. The following tablespace information is therefore provided for information and reference only.

The SimCorp Dimension tablespaces typically as a minimum include:

- SCDAT: Data owner’s tablespace. Contains application data.
- SCSYS: Application metadata.
- SCINDX: Data owner’s indexes.
- SCAUDIT: Application audit (history) data.
- SCAUDIX: Audit indexes.

They should be configured as Locally Managed and using ASSM. If uniform size is specified it must be minimum 128K if db_block_size is 8K (otherwise minimum 14*db_block_size) as SimCorp Dimension uses XMLTYPE data columns.

Besides these, one or more tablespaces may have to be configured for optional SimCorp Dimension functionality, e.g. Order Manager, Communication Server, and Data Extractor.

For more information on the SimCorp Dimension tablespaces, please refer to the ***SimCorp Dimension System Administrator’s Manual***.

2.1.4 Data Files

As the database grows over time, you will need to extend your data files and add new ones. It is recommended that you enable AUTOEXTEND on the data files in order to ease administration. You may specify a MAXSIZE and add new data files when the existing data files grow near their maximum size. SimCorp recommends an initial file size of 100 MB-1 GB for auto extendable files and that you continually monitor file sizes and disk space. You are encouraged to set up an alert notification scheme using Oracle Enterprise Manager Cloud Control or a similar administration tool.

If AUTOEXTEND is on, please ensure that NEXT size is of a reasonable volume. What is reasonable depends on the usage of the tablespace, but for SimCorp Dimension tablespaces this should be at least in the 100s of Megabytes rather than Kilobytes.

2.1.5 Redo Log Files and Archive Log Files

To avoid too frequent check pointing and log switches your redo log file size should be set to at least 500 MB and you ought to have at least 4 redo log groups. For very busy systems you may consider having even larger redo log files. A very rough rule of thumb is to aim for a log switch every half-hour during normal workload.

RAID-5 or RAID-6 **is not a good choice for redo log files because many small writes tend to be expensive in terms of the I/O overhead for parity calculation**.

Neither is Solid State Disk (SSD) a good choice for the redo log files, as although on average delivering good write performance, they may suffer write peaks which will severely increase wait time on ‘log file sync’. An exception to this is Exadata which is specially optimized for redo on SSD.

In production systems, it is **highly recommended** that you put the database in ARCHIVELOGMODE. You should decide where to store your archived log files. Preferably, you should put archived logs on separate disks or disks where the additional load will not have a negative impact on system performance.

2.2 Database Character Set

SimCorp Dimension supports the Unicode database character set AL32UTF8. All new installations of SimCorp Dimension will require the database to use this character set.

Existing installations are supported on the single byte Western European character set WE8MSWIN1252. In this case the database should be migrated to the Unicode character set as soon as convenient, and at latest before upgrading SimCorp Dimension to version 6.3/6.4.

2.2.1 Migrating from Single to Multi Byte Character Set

If the SimCorp Dimension database needs to be migrated to Unicode, Oracle's guidelines for doing so must be followed. If you have a SimCorp Dimension Data Warehouse remote database using single byte character set, you should ensure that it is migrated to Unicode before the SimCorp Dimension core database.

In short the migration process consists of the following activities, which are described in more detail in the following sections:

1. Preparation below
2. Scan data with DMU on the next page
3. Analyse scan result and solve issues on page 13
4. Repeat step 2 and 3 until no more issues
5. Migrate database to Unicode on page 14
6. Post migration tasks in SimCorp Dimension on page 16

2.2.1.1 Preparation

Oracle provides the tool Oracle Database Migration Assistant for Unicode (DMU) to assist the process. SimCorp recommends that you use the latest version[1] of the DMU tool. For more information on DMU please refer to: http://www.oracle.com/technetwork/database/database-technologies/globalization/dmu/overview/index.html.

Before starting the process, please ensure you have read the Database Migration Assistant for Unicode Guide (available under Documentation in the before mentioned link) and the Oracle Support Document 1637455.1 (Known Issues with The Database Migration Assistant for Unicode (DMU) Tool). It is also recommended to visit: http://www.oracle.com/technetwork/database/database-technologies/globalization/dmu/learnmore/faq-345828.html.

When you plan for the migration, you should be aware that preparing the database can be a lengthy affair before it is ready to be converted. It

[1] In case you are using version 2.1.1 SimCorp recommends that you at least apply patch 25607433. The patch solves the problem, as described later in this chapter, with primary key indexes getting misplaced in case a table is changed using CTAS.

depends on the findings from the DMU scan. You are strongly encouraged to work with a clone of the production system, in order to identify possible issues, test their resolution and plan the activities for the actual production system. It is SimCorp's recommendation that you plan to migrate the production database at a time which does not coincide with a SimCorp Dimension upgrade.

Before starting the process you are strongly recommended to ensure as much SimCorp Dimension data cleanup and archiving have been done as possible. You are also encouraged to have run SimCorp Dimension Cleanup Extraneous Objects and Verify Database Structure with the option "Execute Alter Table Where Needed".

Optionally, you can in advance of scanning and converting the database change the length semantics on VARCHAR2 columns from BYTE to CHAR. SimCorp Dimension has for some time used CHAR as the default, but likely there are older tables which haven't been changed yet. All columns must be changed as part of the Unicode migration and doing it in advance might ease the scanning (and conversion when using DMU), and shorten the workload during the actual production conversion. This can be done using SQL ALTER TABLE commands and modify the columns where possible. For some tables you may get the error "ORA-01404: ALTER COLUMN will make index too large". The tables with columns that are not possible to alter must be fixed afterwards using SimCorp Dimension Verify Database Structure. A SQL script is available from SimCorp on request, which can be used to change the length semantics for those columns where possible and list the tables where it is not possible. Then run Verify Database Structure with options "Force ALTER TABLE on all tables" and "Select tables". This will open a dialog for selecting the tables which could not be changed with the SQL ALTER TABLE command. Verify Database Structure will now force a full copy of the selected tables. This operation should be done in a SimCorp Dimension maintenance window.

If SimCorp Dimension is version 6.1 or below you should ensure you have Oracle bug fix for 21540128 FSM PARSER MISHANDLES OVERFLOW IF MULTI-BYTE CHARACTER ENDS ON BUFFER BOUNDARY. For version 6.2 (and above) the bug fix is included in the minimum requirements for SimCorp Dimension.

2.2.1.2 Scan data with DMU

The database must be scanned using DMU. Please refer to the Oracle documentation for the prerequisites and usage guidelines. The following options have special relevance for SimCorp Dimension.

As SimCorp Dimension does not store rowids, you can set the "Consider CTAS with Row Movement Disabled:" parameter to "Yes" under General Conversion Information in the database properties so that DMU will assign the CTAS conversion method for optimal conversion performance.

SimCorp has seen several issues in databases after Unicode migration where some characters were not migrated correctly and resulted in queries failing with "ORA-29275: partial multibyte character". SimCorp has together with Oracle Support verified that the problem can be avoided by changing the "Assumed Database Character Set" in DMU from blank to

WE8MSWIN1252 (which is the current database character set). The setting is placed on the General tab in database properties.

In some situations, the DMU application might choose the table conversion method "Copy data using CREATE TABLE AS SELECT" for reference-partitioned tables. This is not a valid approach for such tables because it will cause the DMU application to attempt to drop the referential integrity constraint that is used for the child tables which is not allowed. If this conversion method is chosen by the DMU application it will cause the conversion to fail (ORA-14650).

The following table groups in SimCorp Dimension are using reference partitioning:

- Audit trail and 4 eyes principle
- Calc figures
- Log messages

If any of the table groups are partitioned there are two workarounds:

- Disable partitioning of the table group prior to Unicode migration.
- In the DMU application select "Convert Database" which brings you to the "Conversion Details" screen. In the conversion step "Convert Application Tables" click on "Edit Table Conversion Plan". Once the DMU application has created the conversion plan you should manually select an appropriate conversion method other than "Copy data using CREATE TABLE AS SELECT" for all referenced-partitioned tables. The following query can generate a list of tables when executed as data owner: SELECT table_name FROM user_part_tables WHERE partitioning_type='REFERENCE'.

2.2.1.3 Analyse scan result and solve issues

If there is data in CDEF$.CONDITION which cannot be converted and the column using the condition is SYSCHGLOG in the system owner schema, e.g. SCSYS.SYSCHGLOG, you can drop the constraint prior to the migration. The constraint will be re-created during the next SimCorp Dimension patch or upgrade.

If there is data in REG$.SESSION_KEY please refer to Oracle Support Document 1490394.1 (Lossy or Convertible Conversion Data In SYS.REG$ Table).

If the DMU scan reports issues which require alterations to SimCorp Dimension data, the alteration as a rule should be done through the appropriate interfaces in SimCorp Dimension. An example could be Over Type Limit reported on a FREECOM VARCHAR2(4000) field. When changing data in SimCorp Dimension that is protected by audit trail, the "bad data" will be in the audit trail after you change the source table. If you e.g. change the name of a security, the corresponding entry in the table AUDITDETAIL column VALOLD would need to be changed (or deleted), before the migration can take place.

Over Type Limit (exceed data type limit) issues on the AUDITDETAIL columns VALOLD or VALNEW, should be solved by allowing DMU to truncate the values during the conversion. Alternatively use the SimCorp

Dimension Audit Cleanup task to delete the audit records. If changes to audit records, in this circumstance, violates your audit regulations, you have the option to archive the audit data for future audits, after which the audit records can be deleted.

If the DMU scan finds Invalid Data (invalid binary presentation) on SimCorp Dimension tables, you should contact SimCorp for advice on the solution. In general you should create a SR with SimCorp if the scan reports issues where you are in doubt of the solution. When creating the SR please also attach a Problem Data Report (can be created from the DMU Migration menu).

2.2.1.4 Migrate database to Unicode

Oracle offers several methods for doing the actual conversion. Which method is chosen depends on several factors, e.g. time frame and competences. The most typical methods are:

- Use DMU
- Use DataPump export/import
- Use a mix of DMU and DataPump export/import

Regardless of the method chosen data must be scanned and any issues resolved before the migration is initiated.

For CLOB columns, you should note that CLOBs are stored in an internal fixed-width Unicode character set AL16UTF16 after the conversion. This means that each character is stored as at least 2 bytes in the LOB segment and thereby the segment will increase to around double the size compared with single byte database. For more information please refer to the following My Oracle Support articles:

- CLOBs and NCLOBs character set storage in Oracle Release 8i, 9i, 10g and higher (Doc ID 257772.1)

It has been identified that some versions of the DMU application, when changing tables during conversion using CTAS, misplace the corresponding primary key index in the wrong tablespace; it is placed in the same tablespace as the table, regardless of what tablespace was used prior to the CTAS operation. When running "Verify Database Structure" after the Unicode migration SimCorp Dimension will identify the incorrect placement and rebuild the index into the correct tablespace. This will most likely extend the duration of "Verify Database Structure".

If you during the migration using DMU receive ORA-00904 errors on columns marked as unused, these can be ignored. The unused columns will automatically at a later point be removed by SimCorp Dimension, for example during upgrade.

Please ignore, if there following the database migration are invalid SimCorp Dimension owned database packages, procedures or other PL/SQL code. These will be recompiled by the application in the correct context.

2.2.1.4.1 When large tables containing LOB columns take too long to migrate using DMU

SimCorp recommends using DMU for the entire conversion process. However, conversion of large tables with LOBs might take very long time, and if DMU can't finish the conversion within an acceptable time frame, SimCorp suggests another approach. Large tables with LOBs can be exported prior to converting the database using DMU. Immediately following the database conversion, the tables can be imported again. This has proved to be much faster. To follow this approach the following must be done.

- Ensure that the length semantics VARCHAR2 columns on the affected tables is set to CHAR.
- Run DMU and scan the affected tables. Any data issues MUST be solved prior to exporting.
- Note the results of SELECT COUNT(*) from each of the affected tables. You need these figures for verification after the import.
- Export the affected tables. You should export with "content" set to "data_only". Since datapump doesn't parallelize operations on LOBs, you should set the "parallel" parameter to 2 (one main thread, and one worker thread). You can start multiple datapump sessions each working on a separate table to speed up the process.
- On the affected table the triggers should be dropped to allow data modification from outside SimCorp Dimension.
- Referential integrity constraints to the table should be dropped to allow truncation of the table.
- Drop primary key (type P) and unique (type U) constraints on the table to allow drop of indexes.
- Drop indexes on the table to speed up the import. Further, it is generally faster to let Align Objects re-create indexes than using DataPump.
- Disable all check constraints (type C) on the table except NOT NULL constraints. These must be left on the table to avoid a CTAS operation during Verify Database Structure after the migration. Similarly, you should disable referential constraints (type R) from the table.
- Truncate the tables
- Start DMU and convert the database to Unicode. Optionally, before pressing "Convert" button on the "Conversion Details" tab, follow the "Edit Table Conversion Plan" link in the upper right corner. In the window, find the affected tables and ensure that Oracle chooses "Update all rows" as the "Conversion Method". This is slightly faster than rebuilding using the table using CTAS.

- Immediately following the database conversion, import the affected tables. Like the export, "content" should be "data_only", and "parallel" should be "2". Any warnings about data loss due to character set conversions can be ignored because the tables were scanned (and fixed) previously.
- Note the results of SELECT COUNT(*) from each of the affected tables. Compare these figures with the figures you had before export.
- Enable all the check and referential constraints on the table with NOVALIDATE and RELY.
- Follow the post-conversion steps as otherwise described.

A script is available that can generate the DataPump commands and DDLs needed to follow this approach. Please contact your SimCorp representative. In addition, a paper (Benchmark of Unicode migration using DMU and DataPump) is also available on request.

2.2.1.5 Post migration tasks in SimCorp Dimension

In the single byte SimCorp Dimension database there are constraints to ensure multi byte character data cannot be inserted. Once the database character set has been migrated to Unicode these constraints need to be changed before it is possible to input multi byte characters from the application. Before the constraints can be changed, a new SimCorp Dimension ins file containing the Unicode option is required. Please request such from your local SimCorp office.

After the Unicode conversion, regardless of the migration method used, the following tasks must be executed in order:

1. As stated in section nls_length_semantics on page 34, if SimCorp Dimension is version 6.1 or lower set Oracle database init parameter NLS_LENGTH_SEMANTICS to CHAR, e.g.:

   ```
   SQL> ALTER SYSTEM SET NLS_LENGTH_SEMANTICS='CHAR'
   SCOPE=BOTH;
   ```

2. Ensure all VARCHAR2 columns are CHAR (and not BYTE) defined.
3. Ensure that database statistics have been purged, and regathered as described in the Oracle Database Migration Assistant for Unicode Guide chapter 5 "Advanced Topics in the DMU"
4. Force SimCorp Dimension to re-install the Align Objects components, e.g. if data owner is SCDAT:

   ```
   SQL> DROP PACKAGE SCDAT.t_version_s;
   ```

5. Check the new SimCorp Dimension ins-file contains Unicode=1
6. Apply the new ins file containing the Unicode module.
7. Run Verify Database Structure with Execute Alter Table Where Needed.

8. The system can now be opened for tests and validations, while the step below is executing.
9. Run Validate Basic Constraints.

If Align Objects (during apply of new ins-file or verify database structure) is running for a very long time spending time of recursive SQLs then you might be affected by the known issue "Missing statistic on Oracle scheduler background tables" as recorded in the System Administrator's Manual. If so, abort the current operation, apply the mentioned workaround, and re-do the aborted task.

2.3 Memory Sizing

Taking into account the physical memory available for the database instance, you need to decide about memory allocation for the Oracle SGA and PGA, i.e. the target for the Oracle memory.

When configuring the Oracle memory parameters you must take care not to over allocate, whilst considering all processes/applications which will be running on the database host. An instance configured with excessive values can cause serious degradation in performance and may lead to users receiving Oracle errors as they run out of process memory. As a rule of thumb, the sum SGA_TARGET + PGA_AGGREGATE_TARGET (or MEMORY_TARGET) should max be 70-80% of the physical memory available for Oracle.

Guidelines for how to find optimal values are provided in the comments given on the individual sizing parameters in section Preferred Oracle Initialization Parameters. The sizes provided in the table below are intended as an initial guideline. They are based on feedback from existing SimCorp clients on the sizes used in their production.

SimCorp Dimension size	SHARED_POOL_SIZE	DB_CACHE_SIZE	Sum of SGA and PGA
Small (up to 100 sessions)	4G	12G	20G
Medium (100 to 300 sessions)	4G	33G	50G
Large (300+ sessions)	4G+	40G+	60G+

Note in an Oracle RAC configuration the above guidelines should be applied per instance in the cluster.

2.4 Oracle 12c Multitenant Container and Pluggable Databases

Please note SimCorp Dimension is not supported in the Oracle Multitenant Architecture (nor single-tenant mode) when the database is on Oracle 12.1 because the application utilize features currently not supported in a 12.1 pluggable database. These are:

- DBMS_QOPATCH
- Database Change Notification

For more information see http://docs.oracle.com/database/121/READM/chapter12102.htm#READM120 and https://mikedietrichde.com/2017/02/08/dbms_qopatch-does-not-work-in-pdbs-right-now/.

SimCorp has verified the features availability in Oracle 12.2 and do support SimCorp Dimension in a pluggable database when the database has been upgraded to 12.2. This implies SimCorp Dimension must be upgraded to version 6.4 or higher and the database upgraded to 12.2 before moving to a pluggable database. The Oracle preferred settings for SimCorp Dimension described in this manual also applies to the pluggable database.

3 Oracle Software and Database Installation

This chapter provides an overview of the Oracle software installation and database creation process and highlights the specific requirements for SimCorp Dimension. It is assumed that the installation of the Oracle software and creation of the database is performed by an experienced DBA, in accordance with Oracle documentation. My Oracle Support note: Oracle Database - Client / Install / Deinstall / Cloning (Doc ID 1555012.2) is a good starting point.

3.1 Supported Platforms

SimCorp recommends that either Microsoft Windows, Linux or one of the wide-spread implementations of UNIX is chosen as the underlying platform for the Oracle database.

Configuration of the OS should be done in accordance with the guidelines and requirements for the chosen platform as provided by Oracle. The ***Database Installation Guide for <OS name>*** contains checklists to ensure the actual OS environment is configured as required when running an Oracle Database. The books are available at https://docs.oracle.com/en/database/oracle/oracle-database/12.2/install-and-upgrade.html.

The SimCorp Dimension database is also supported on Oracle Exadata and Oracle Database Appliance.

3.2 Download the Oracle Software

The Oracle Database software can be downloaded from http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html.

The downloaded software is delivered as zip files. Please always read and follow the instructions in the accompanying readme file. When installing the software ensure to install the appropriate Oracle edition (Standard or Enterprise Edition). If installing Enterprise Edition, ensure to only install the options you require. See the ***SimCorp Dimension System Administrator's Manual*** for a list of SimCorp Dimension features requiring Oracle Enterprise Edition and options.

Release Updates (RU)/Bundle Patches(BP) can be applied following the software installation. Please refer to the relevant version of ***SimCorp Dimension System Administrator's Manual*** for the minimum required Oracle RDBMS version and required patches. Oracle patches (including RU/BP) are downloaded through My Oracle Support. The accompanying readme file provides installation instructions.

3.3 Create the Database

Creating the database can be done using any method as provided by Oracle. The database must be installed with the SimCorp Dimension supported database character set (see Database Character Set in this manual)

Typically, three SimCorp Dimension environments will be required, all having a dedicated function in the environments life cycle. Each environment requires a dedicated database. The recommended naming convention follows the environment name as shown below:

Environment	DB Name	Oracle Net Alias
PROD	SCPROD	SCPROD
TEST	SCTEST	SCTEST
NEW	SCNEW	SCNEW

Please refer to the ***SimCorp Dimension System Administrator's Manual*** for a more detailed description of the environments, their purpose and usage.

3.4 Configure the Database for SimCorp Dimension Usage

Once the initial database has been installed it must be configured to comply with the SimCorp Dimension Preferred Oracle Configuration. Please refer to the chapter Preferred Configuration for SimCorp Dimension on page 30 in this manual.

3.5 Installation of Oracle Clients

The SimCorp Dimension software delivery includes an embedded Oracle Instant Client, so separate installation of Oracle client software is not required for the application. If an Oracle client is required for other purposes, please refer to Oracle installation guidelines for this.

4 Configuring Oracle RAC for SimCorp Dimension

SimCorp Dimension is supported on Oracle Real Application Cluster (RAC). In general when installing and configuring the RAC database Oracle's best practices should be observed.

Note Oracle RAC is a highly complex database solution which requires specialization skills in order to install, configure and operate. Choosing RAC for SimCorp Dimension should therefore only be done following careful considerations of the available skill set.

At My Oracle Support (MOS), Oracle provides a series of articles headed ***RAC and Oracle Clusterware Best Practices and Starter Kit***. Both the platform independent and the platform dependent versions of these articles must be thoroughly studied. Doc id 1096952.1 ***Master Note for Real Application Clusters (RAC) Oracle Clusterware and Oracle Grid Infrastructure*** provides links to all the starter kits, as well as links to additional notes, frequently asked RAC questions, tools and known issues. The ***Information Center: Oracle Scalability Grid Infrastructure / Clusterware and Real Application Clusters (RAC)*** (Doc ID 1452965.2) provides a useful portal for RAC related information.

The RAC configuration must use ASM (Automatic Storage Management). Please refer to MOS doc id 1187723.1 ***Master Note for Automatic Storage Management (ASM)*** to get links to more detailed documentation on installation, configuration, administration, monitoring and best practices for ASM.

If installing on a Microsoft Windows platform, it is SimCorp's experience that it is best to use a dedicated, purposely created Windows user for the Grid Infrastructure software installation, patching and upgrading. This helps prevent potential software ownership issues.

For SimCorp Dimension Preferred Oracle Configuration specifically, the following must be observed in regards to the RAC configuration:

1. Place redo logs on separate high speed disks (not RAID which requires parity calculation) – cf. section Redo Log Files and Archive Log Files
2. Use high speed interconnect between RAC nodes
3. Use ASSM tablespaces
4. LOG_ARCHIVE_FORMAT must contain thread# e.g. %t%s%r.arc
5. PARALLEL_EXECUTION_MESSAGE_SIZE must be at least 16K
6. Configure and set remaining SimCorp Dimension preferred values as described in chapter Preferred Configuration for SimCorp Dimension
7. Complete the remaining tasks as described in section Configure the Database for SimCorp Dimension Usage

And for configuration of ASM:

1. Increase the default memory for the ASM instance size if the database controlled by ASM will be greater than 5TB
2. Combining RAID and ASM is not supported if using a dedicated disk system, use either:
   A. ASM only (as recommend by Oracle)
   B. RAID 1 + ASM external redundancy
   C. RAID 5 + ASM external redundancy (however not for redo, please see Redo Log Files and Archive Log Files)
3. If the disk system is a SAN, use ASM external redundancy and let the SAN handle the RAID
4. Disks in a disk group should have the same size and performance characteristics (disk response time)

The RAC configuration should be scaled (cluster interconnect, number of nodes, CPU per node, memory etc.) in accordance with Oracle's directions and guidelines. The Oracle supplied tool Cluster Health Monitor can be used for cluster-wide analysis of OS resource consumption and is documented in the ***Oracle Clusterware Administration and Deployment Guide***. And in general, the RAC configuration and database can be administrated through Oracle Grid or Oracle Database Control.

Note When adding nodes to the cluster, the new nodes will only be used by SimCorp Dimension sessions connecting to the database thereafter. That is database sessions already connected to the database will not use the new nodes. E.g. if the new node is added to ease congestion of the SimCorp Dimension Service Platform sessions, and the aim is for the service sessions to utilise the new node, the SimCorp Dimension Service Platform must be restarted, thereby forcing a reconnect to the database.

4.1 Connecting to the RAC Cluster

There are a few considerations in regards to choice of connectivity strategy when SimCorp Dimension is using a RAC database.

Note SimCorp Dimension, with its stateful database connections, does not support Oracle TAF (transparent application failover). If the application loses connection to the database instance, the SimCorp Dimension session must be reinitialised and a new database connection made.

It is a prerequisite to configure Single Client Access Name (SCAN) if the SimCorp Dimension clients are to connect using the dbconnect entries specified in Cnf.ini file rather than the dbname entry. Please refer to the ***SimCorp Dimension System Administrator's Manual*** for a detailed description of the Cnf.ini connectivity parameters. See also the section SimCorp Dimension Configuration Examples .

A “Round robin” connect method can potentially be unsuitable in RAC setups with an even number of nodes. Each SimCorp Dimension session started makes two connections to the database at initialisation time, one session using OCI and another session using ODP.NET. The typical pattern is that the heaviest database activity is done via the OCI session. With an even number of nodes it is possible that the OCI sessions will be connected on one node and the ODP.NET sessions on another, creating an uneven load on the cluster nodes.

From version 5.8, it is possible for all database sessions created by the same one SimCorp Dimension session to reside on just one node in the RAC configuration. This removes the concern, and the Round robin connect method is equally suited uneven and even number of nodes. Round robin connectivity will in this way be applied per SimCorp Dimension Windows session rather than per database connected session. In order to utilize this option it is a pre-requisite that the SimCorp Dimension installation is configured to use the dbconnect parameters in the Cnf.ini file. If the dbname parameter is used, it will not be possible to maintain the RAC node affinity for the SimCorp Dimension sessions.

A “Load balancing” connect method can potentially be unsuitable for SimCorp Dimension server processes. Typically when SimCorp Dimension server/service processes connect to the database (at their start-up time) the load they place on the database will be minimal whilst they are in idle state. It is first when the processes are assigned a job that they can generate significant load on the database instance. Therefore instances in the cluster can initially appear least loaded and attract new connections, it is only when the SimCorp Dimension server processes enter a busy state, that the instance may become overloaded.

It may become necessary to create a service orientated connect method allowing separation of online SimCorp Dimension users from the SimCorp server/batch processes. This can provide a shield between the requirements of the online users and that of the server/batch processes, and can prevent them from competing for available resources.

In all cases, it is a must that the instance utilisation on each node is monitored and reviewed on an ongoing basis and there is a readiness to change connectivity strategy if required.

If the database runs on a Windows platform, the sqlnet.ora used by the Oracle Grid Infrastructure must include NTS authentication, e.g.: SQLNET.AUTHENTICATION_SERVICES=(NTS).

4.1.1 SimCorp Dimension Configuration Examples

In the following example we have a RAC database consisting of 5 nodes (instances). The database name is RACDB and the instances: RACDB1, RACDB2, RACDB3, RACDB4 and RACDB5.

The database connection does not need to specify which instance in the RAC cluster it wants to connect to. Instead you have a SCAN listener installed in the RAC cluster. The SCAN listener basically consist of 3 listener processes but all answering to the same ‘name’, in our example: raccluster-scan. The SCAN listener knows each of the local listeners running on RACDB1, RACDB2, RACDB3, RACDB4 and RACDB5. When you connect to the database you simply connect to the SCAN listener and it then guides your connection to the right instance. Details of how this is installed and configured can be found in the Oracle’s documentation on the RAC & Grid Clusterware installation and administration. The examples here concentrate on the SimCorp Dimension side of configuration.

Scenario 1:

In the most simple setup we just want our SimCorp Dimension sessions to be evenly distributed across the 5 instances, i.e. we will adapt a round-robin method for connection distribution, which means the first database session gets connected to RACDB1, the next gets connected to RACDB2 and so forth.

If we use tnsnames, this is how our connect identifier for RACDB would look:

```
RACDB.WORLD =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)
(HOST = raccluster-scan.raccluster.gdom.net)
(PORT = 1521))
)
(CONNECT_DATA =
(SERVICE_NAME = RACDB)
)
)
```

And in cnf.ini we would simply have our connect string to the database in the dbname entry:

```
; dbname : database source name
dbname=RACDB

; dbconnecthost, dbconnectport, dbconnectprotocol and
dbconnectservice or dbconnectsid : alternative data
source specification
;dbconnectsid=
;dbconnecthost=
;dbconnectport=
;dbconnectprotocol=
;dbconnectservice=
```

If we do not want to use tnsnames we can specify the connect details directly in the cnf.ini file. This will also ensure all database sessions from the same SimCorp Dimension Windows session connect to the same RAC node (ref Connecting to the RAC Cluster on page 22):

```
; dbname : database source name
; dbname=

; dbconnecthost, dbconnectport, dbconnectprotocol and
dbconnectservice or dbconnectsid : alternative data
source specification
; dbconnectsid=
dbconnecthost=raccluster-scan.raccluster.gdom.net
dbconnectport=1521
dbconnectprotocol=TCP
dbconnectservice=RACDB
```

If we want to use load balancing as our connect methodology, i.e. that the connections are distributed to the instances based on their load, we have to use tnsnames, for example:

```
RACDB.WORLD =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)
(HOST = raccluster-scan.raccluster.gdom.net)
(PORT = 1521)) )
(LOAD_BALANCE = ON)
(CONNECT_DATA =
(SERVICE_NAME = RACDB)  )  )
```

And the dbname=RACDB entry in cnf.ini.

Scenario 2:

The second scenario is a little more complex. Now we have decided that 2 of our 5 nodes should be dedicated to certain SimCorp Dimension sessions, let’s say our calculation and STP server’s database sessions. All other SimCorp Dimension sessions should connect to the remaining 3 nodes. In this case we will have to make 2 services on the RAC cluster. Let us call them SCDSRV and SCDUSR. Again, the details of how to do this can be found in the Oracle documentation.

For this scenario we have to use a tnsnames file, and have to manage two separate versions of the tnsnames file. The SimCorp Dimension servers need one and the users another. The cnf.ini file will be the same across the board:

```
; dbname : database source name
dbname=RACDB

; dbconnecthost, dbconnectport, dbconnectprotocol and
dbconnectservice or dbconnectsid : alternative data
source specification
;dbconnectsid=
;dbconnecthost=
;dbconnectport=
;dbconnectprotocol=
;dbconnectservice=
```

The servers will look at a tnsnames containing:

```
RACDB.WORLD =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)
(HOST = raccluster-scan.raccluster.gdom.net)
(PORT = 1521)) )
(CONNECT_DATA =
(SERVICE_NAME = SCDSRV)  )  )
```

Whereas the user sessions will look at a tnsnames containing:

```
RACDB.WORLD =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)
(HOST = raccluster-scan.raccluster.gdom.net)
(PORT = 1521)) )
(CONNECT_DATA =
(SERVICE_NAME = SCDUSR)  )  )
```

Which tnsnames the SimCorp Dimension client looks into is controlled by the TNS_ADMIN, as a registry entry in HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE.

Please note, in this last example the SimCorp Dimension Cnf.ini parameter tnsadmin cannot be used to specify location of tnsnames. In this case, if the parameter is set in the cnf.ini file, it must be removed and the clients must rely on the TNS_ADMIN entry in registry as described above. For more information on the tnsadmin Cnf.ini parameter, please refer to the ***SimCorp Dimension System Administrator's Manual***.

5 Oracle Upgrade and Patching

A given version of SimCorp Dimension is verified and supported to run on a minimum Oracle database version. To allow for the Oracle upgrade from one version to another, specific versions of SimCorp Dimension can be run on both new and the prior supported Oracle version. The System Administrators manual for the given SimCorp Dimension version provides the information on the supported Oracle versions. For example:

SimCorp Dimension introduced support for Oracle 12.2 from version 6.4. The minimum required Oracle 12.2 version (release update/bundle patch) is also listed in the System Administrators manual. SimCorp Dimension must be upgraded to minimum version 6.4 before you upgrade Oracle to 12.2. The current plan is to have dual support for Oracle 12.1 and 12.2 up to and including SimCorp Dimension version 19.04, implying Oracle must be upgraded to Oracle 12.2 before upgrade of SimCorp Dimension to version 19.07.

5.1 Roadmap for SimCorp Dimension alignment with new Oracle releases

Starting with Oracle 12.2 Oracle has changed to an annual release schedule for new database releases (cf. My Oracle Support article: Release Schedule of Current Database Releases (Doc ID 742060.1)). In addition the year of the release will now be part of its name, e.g. Oracle 18, Oracle 19 etc.

In regards to support of the new releases in SimCorp Dimension the goals being worked towards are:

- Provide support for a new Oracle release in the SimCorp Dimension version released 3 months after the on premise Oracle database release on Linux.
- An Oracle version must be supported in such a way that a SimCorp Dimension four versions upgrade still can be separated from the Oracle upgrade.
- After certification of a new Oracle version retrospective support is considered for the two previous SimCorp Dimension versions.
- If support for the new version is given retrospective, de-support of the older version will be brought forward.

The actual roadmap for future versions is published in the System Administrators manual, the chapter Future Plans.

5.2 Oracle Upgrade Tips and Experiences

Generally, the upgrade of Oracle must be done using the Oracle provided guidelines for upgrading to the new version. The Oracle Database Upgrade guide provides detailed information on the process of planning and executing Oracle Database upgrades. The guide for the actual version is available from https://docs.oracle.com/en/database/oracle/oracle-database/index.html using the Install and Upgrade link.

Internally SimCorp works with Oracle upgrades for a longer period prior to supporting the version on SimCorp Dimension. Following are some tips for the upgrade process, gained through upgrading more than 100 databases internally:

- Always get the latest version of the Preupgrade Tool (preupgrade.jar) – See My Oracle Support note: How to Download and Run Oracle's Database Pre-Upgrade Utility (Doc ID 884522.1).
- The Preupgrade Tool can output to XML format, which makes it suitable for batch processing, for example to automate the upgrade.
- Use Flashback Database and guaranteed restore point as fallback method (cf: https://mikedietrichde.com/2012/06/28/guaranteed-restore-points-as-fallback-method/). At SimCorp this proved to have an insignificant overhead (approx. 1 minute on the upgrade time and approx. 10 GB redo generation independent of database size).
- Remove APEX from the database or upgrade in advance (cf. https://mikedietrichde.com/2014/06/06/save-upgrade-downtime-upgrade-apex-upfront/). The experience in SimCorp: APEX added 25 minutes, which was a doubling of the total database upgrade time.

And for post upgrade:

- From Oracle 12.2 the database uses more memory, especially shared pool. Therefore, the minimum recommendations for shared_pool_size is also raised to 4GB. Generally, if you in the 12.1 database are running close to full Oracle memory utilization you are encouraged to increase memory target for the database.
- Transfer database audit records using DBMS_AUDIT_ MGMT.TRANSFER_UNIFIED_AUDIT_RECORDS into the new 12.2 format for greatly enhanced query performance on audit data (cf. https://docs.oracle.com/en/database/oracle/oracle-database/12.2/arpls/DBMS_AUDIT_MGMT.html#GUID-0465DB99-CD48-405C-ACAC-0BB5DC658A19)
    - Known issue: DBMS_AUDIT_MGMT.TRANSFER_UNIFIED_AUDIT_ RECORDS fails with ORA-12899 on CLIENT_PROGRAM_NAME (Enhancement Bug 27259651 in development at Oracle). Occurs when trying to migrate the unified audit trail into new 12.2 format if audit record exists in which the column CLIENT_PROGRAM_NAME exceeds 48 bytes. Only workaround is to purge audit trail up to and including the offending record.
- Consider changing partition interval for internal database audit tables using DBMS_AUDIT_MGMT.ALTER_PARTITION_INTERVAL (cf. https://docs.oracle.com/en/database/oracle/oracle-database/12.2/arpls/DBMS_AUDIT_MGMT.html#GUID-7A9B2516-82B6-4284-970D-78D962C96A11).
- The SimCorp Dimension created Oracle directory SCDTRACEDIR can be removed once the database is on 12.2 or higher.

The general experience from performance tests of SimCorp Dimension on Oracle 12.2 conducted at SimCorp, is the 12.2 performance is very similar to 12.1.

5.3 Patching Oracle

With the introduction of 12.2, Oracle has changed patching strategy. For Windows platform patches are released in bundle patches (BP) as for earlier Oracle versions. For all other platforms patches are now released in Release Updates (RU) and Release Update Revisions (RUR). More information can be found in My Oracle Support note: Oracle Database - Overview of Database Patch Delivery Methods for 12.2.0.1 and greater (Doc ID 2337415.1).

From Oracle 12.2 a given SimCorp Dimension version will require a minimum RU/BP. Meaning, at least that RU/BP must be installed in order for SimCorp Dimension to start. RU/BP are accumulative, and SimCorp Dimension is supported on more recent RU/BP than the minimum required. From version 6.4 going forward all SimCorp Dimension released versions are continuously tested on the latest four available RU/BP.

6 Preferred Configuration for SimCorp Dimension

SimCorp Dimension depends upon most Oracle parameters using their default values. However certain default parameter values have proven inadequate for the sound operation of SimCorp Dimension. This chapter is dedicated to describing these configuration parameters and their recommended values.

The preferred Oracle configuration is designed to provide a solid base for SimCorp Dimension installations. Preferred Oracle Configuration is constantly being evaluated and refined to make sure that the database is configured optimally for SimCorp Dimension. It is based on experiences from not only internal SimCorp usage, but also from many clients worldwide running SimCorp Dimension in many varied configurations.

Note SimCorp Dimension is thoroughly tested on the preferred Oracle configuration. SimCorp Dimension performance can be seriously affected by database configuration changes. As applications differ one application may suffer from a configuration change made in favour of another application. There may be various reasons for preferring other configurations, but be aware that by purposely choosing a divergent configuration you are responsible for ensuring that SimCorp Dimension performs properly on the changed configuration.

6.1 Preferred Oracle Initialization Parameters

The tables below provide an overview of the Oracle initialization parameter specified in Preferred Oracle Configuration for SimCorp Dimension. Each parameter change is explained in the sections following.

Static Parameters:	Non default value
audit_trail	DB
_b_tree_bitmap_plans	FALSE
control_file_record_keep_time	90
_cursor_obsolete_threshold	1024 (on 12.2 and onwards)
db_block_size	8192
open_cursors	500
recyclebin	OFF
_optim_peek_user_binds	FALSE
optimizer_adaptive_reporting_only	TRUE
temp_undo_enabled	TRUE

Variable Parameters:	**Non default value**
db_cache_size	set appropriate minimum value
job_queue_processes	set appropriate value
log_archive_format	%d%t%s%r.arc
log_buffer	set appropriate value
processes	set appropriate value
sga_target and pga_aggregate_target (or alternatively memory_target)	set appropriate values
shared_pool_size	set appropriate minimum value
undo_tablespace	set to actual UNDO tablespace name

Parameters only applicable in certain circumstances:	**Non default value**
nls_length_semantics	CHAR (see section nls_length_ semantics on page 34)

Note Please use the Oracle “ALTER SYSTEM RESET…” command if you have set a parameter to a given value and want it to revert to the Oracle default. Simply setting the parameter to the default value is not enough, as other hidden parameters may be affected. See for example the explanation provided by Kerry Osborne in his article: http://kerryosborne.oracle-guy.com/2008/11/reset-oracle-initora-spfile-parameters/.

Note When using the “ALTER SYSTEM RESET…” command please be aware that the default value won't take effect until the database is restarted.

6.1.1 audit_trail

SimCorp Dimension contains functionality for viewing database logon activity. This requires the AUDIT_TRAIL parameter set to DB. Please refer to the ***SimCorp Dimension System Administrator’s Manual***’s section on auditing database logon activity. In Oracle12c this functionality employs the traditional audit, implying the value relates to the database running traditional or mixed mode audit. If unified audit mode is enabled this parameter does not apply. Unified Audit is supported by SimCorp Dimension from version 5.9 (see section Oracle Unified Audit Oracle Unified Audit in this manual).

6.1.2 control_file_record_keep_time

One of the metrics collected when running the SimCorp Dimension task **Transfer Meta Data to SimCorp** is avg. number of archive logs per day in the last quarter year. For this reason you are asked to set CONTROL_FILE_ RECORD_KEEP_TIME=90. In addition, in reference to backup, if this parameter is left at its default (i.e. 7 days), you will not be able to make a restore of anything older than 7 days if the recovery catalogue is lost.

6.1.3 db_block_size

SimCorp Dimension is release tested, both functionality and performance wise, using databases with a db_block_size of 8K. Different block sizes have not been tested.

6.1.4 db_cache_size

When MEMORY_TARGET (or SGA_TARGET) is set, the value set in DB_ CACHE_SIZE indicates a minimum value for the Buffer Cache. Oracle then auto-tunes the size of the Buffer Cache depending on actual requirements. By setting a minimum value using DB_CACHE_SIZE Oracle can save on resources following database start up, by eliminating the initial resizing operations that otherwise would be needed to get the Buffer Cache adjusted. Also, experiences with SimCorp Dimension have shown that, in some cases, Oracle can favour the Shared Pool by increasing it to an extreme size, which in turn results in constant resize operations or starvation of the Buffer Cache. This has especially been observed in installations with heavy reporting load and extensive use of literals in user defined queries.

The value for the parameter depends on the whole database configuration, the use of SimCorp Dimension, concurrent activity and the physical framework in which the database operates. Particularly SimCorp Dimension batch jobs and calculations servers running performance calculation jobs have been proven to benefit significantly with a large Buffer Cache.

You should monitor the resize operations and frequency through the V$MEMORY_DYNAMIC_COMPONENTS or V$MEMORY_RESIZE_OPS views. These views contain information about the last 800 resize operations. If resize operations are taking place for the component “DEFAULT buffer cache” frequently (e.g. several times each hour) you should reconsider the value you have specified for DB_CACHE_SIZE. In SimCorp Dimension databases DB_CACHE_SIZE is usually more than 50% of the SGA.

6.1.5 job_queue_processes

This parameter controls the maximum number of concurrently running Scheduler tasks. The value should be set with consideration to the available database server resources.

The recommended value is to set JOB_QUEUE_PROCEESSES to 2 x CPU_ COUNT.

For systems with a very high CPU_COUNT where the database host is shared amongst several databases, JOB_QUEUE_PROCEESSES should be reduced further taking into account the actual available CPU resources for the instance. E.g. if CPU_COUNT is 128, reflecting the underlying database host total number of CPU cores/threads, but the host is shared amongst 4 database instances, consider a JOB_QUEUE_PROCEESSES value of 64 (128x2/4).

6.1.6 log_archive_format

The specified format will name archive log files as [Database id][thread number][log sequence number][resetlogs ID].arc. The format can be adjusted for actual requirements. For example in RAC you need to include %t (cf. chapter Configuring Oracle RAC for SimCorp Dimension). Please refer to Oracle documentation for full details on specifying LOG_ARCHIVE_FORMAT.

6.1.7 log_buffer

The default value for LOG_BUFFER is 512 KB or 128 KB * CPU_COUNT, whichever is greater. SimCorp has experienced that a log buffer of minimum 16MB is required for SimCorp Dimension configurations in order to avoid log buffer space waits. In balance, to avoid high log file sync waits, it should be ensured that log_buffer is not too big as a very large log_buffer can have an adverse affect since waits will be longer when flushes occur. If the default value calculates to something larger than 16MB you should leave the parameter unset.

6.1.8 sga_target, pga_aggregate_target, memory_target

For large memory databases Oracle Automatic Shared Memory Management (ASMM) is recommended, i.e. setting SGA_TARGET and PGA_AGGREGATE_TARGET independently. When ASSM is used MEMORY_TARGET is not to be set.

If Oracle Automatic Memory Management (AMM) is used then MEMORY_TARGET should be set instead of SGA_TARGET and PGA_AGGREGATE_TARGET.

Ensure to leave adequate headroom as the physical memory also needs to accommodate other processes running on the database host (the OS, other applications etc.). The sum of SGA_TARGET and PGA_AGGREGATE_TARGET (or alternatively MEMORY_TARGET) should at max be 70-80% of the physical memory available.

If the database platform is Linux and HugePages[1] are configured, please note that MEMORY_TARGET and MEMORY_MAX_TARGET must not be set, as Automatic Memory Management and HugePages are incompatible. In this case SGA_TARGET and PGA_AGGREGATE_TARGET must be used.

You can use the Oracle miscellaneous memory advisors to detemine appropriate values. As a start use a 80:20 ratio between SGA_TARGET and PGA_AGGREGATE_TARGET.

The maximum amount of PGA memory consumed by the database instance is determined by the PGA_AGGREGATE_LIMIT parameter. The default value for this parameter is derived from the values of the other

[1] Oracle strongly recommends disabling Transparent HugePages and instead use standard HugePages, cf: https://docs.oracle.com/database/121/UNXAR/appi_vlm.htm#UNXAR428 and MOS ALERT: Disable Transparent HugePages on SLES11, RHEL6, RHEL7, OL6, OL7, and UEK2 and above (Doc ID 1557478.1).

parameters mentioned here. For example, if MEMORY_TARGET is not set the default value is 200% of PGA_AGGREGATE_TARGET. If PGA_AGGREGATE_TARGET is undersized you will run into ORA-04036: PGA memory used by the instance exceeds PGA_AGGREGATE_LIMIT. For more information please refer to the Oracle Database Reference manual.

6.1.9 nls_length_semantics

The default value for NLS_LENGTH_SEMANTICS is BYTE. If SimCorp Dimension is version 6.1 or less, this parameter must be set to CHAR if the SimCorp Dimension Unicode module is used (i.e. the database character set is Unicode). In this instance the application presumes character columns to be CHAR defined. If they are BYTE defined there is a serious risk of overflow errors in the application. As not all parts of the application uses CHAR semantics in the issued SQL or PL/SQL, setting NLS_LENGTH_SEMANTICS on instance level is required. From SimCorp Dimension version 6.2 it is no longer a requirement for this parameter to be set on the database.

6.1.10 open_cursors

The default value for OPEN_CURSORS is 50. SimCorp Dimension requires this parameter set to minimum 500.

6.1.11 processes

The PROCESSES parameter should be set to an appropriate value that will support the size of the SimCorp Dimension installation, seen from a number of users, servers and batch jobs perspective. The parameter specifies the maximum number of operating system user processes that can simultaneously connect to the database.

The Oracle default value of 100 is not enough for a SimCorp Dimension installation. Besides allowing for all the SimCorp Dimension connected processes the value must also allow for all Oracle background processes such as locks, job queue processes, and parallel execution processes. Each SimCorp Dimension session will create at least two database processes and may create additional processes. E.g. a user logged on to SimCorp Dimension will have two database connections and if a report is then executed, a third database session will be created for this user. If SimCorp Dimension is set up to utilise parallel processing a SimCorp Dimension session executing such functionality will lead to Oracle creating multiple background slave processes as well.

During SimCorp Dimension upgrade and patching structural changes to database objects are executed utilising the Oracle Scheduler feature. This means many changes are being done concurrently and this may increase the requirements to PROCESSES. As a general rule of thumb, and as a pre-requisite for SimCorp Dimension upgrade and patching, you should ensure that PROCESSES is set to minimum:

PARALLEL_MAX_SERVERS + JOB_QUEUE_PROCESSES + # background processes.

If this figure is higher than the requirements for the daily operations (as described above), you may consider using this figure through daily operations as well.

6.1.12 recyclebin

SimCorp Dimension cannot benefit from the Oracle recyclebin feature. On the contrary, if this parameter is set to TRUE excessive disk space may be required during SimCorp Dimension patching and upgrades, disk space which cannot be used for SimCorp Dimension business generated data. For this reason it is recommended to set RECYCLEBIN=OFF.

6.1.13 shared_pool_size

When MEMORY_TARGET (or SGA_TARGET) is set, the value set in SHARED_POOL_SIZE indicates a minimum value to be allocated for the Shared Pool in memory. Oracle then auto tunes the size of the Shared Pool depending on actual requirements. By setting this minimum value using SHARED_POOL_SIZE Oracle can save on resources following database start up, by eliminating the initial resizing operations that otherwise would be needed to adjust the Shared Pool accordingly.

Also and under certain conditions, functionality in SimCorp Dimension will error if the Shared Pool is too small. The SHARED_POOL_SIZE should be set to at least 4G.

In order to find an appropriate size for SHARED_POOL_SIZE, resize operations and their frequency can be monitored through the V$MEMORY_DYNAMIC_COMPONENTS or V$MEMORY_RESIZE_OPS views. The Oracle memory advisors can also be used; however if presented with what seems like an extreme value, you should be critical and investigate its cause. For example regular execution of badly designed reports using literals will show up as requiring a very large Shared Pool. In this case it may be better to attempt optimizing the reports and then use the memory advisors again.

6.1.14 temp_undo_enabled

SimCorp recommends that the Oracle12c new feature Temporary Undo is enabled.

Operations of SimCorp Dimension have shown significant saving in the redo generated when temp_undo_enabled=true. E.g. on databases with frequent use of global temporary tables (the INLISTS_ tables), by the SimCorp Dimension Service Platform, reduction of more than 50% in redo was seen. This in particular, reduce the storage requirements for archive redo logs. Other areas in SimCorp Dimension which have a significant usage of these table types are the performance measurement and portfolio calculations.

In addiction Oracle recommends: “If database applications make use of temporary objects (using global temporary tables or temporary table transformations), it is advisable to set this parameter's value to true”, cf. MOS article Doc ID 1570287.1.

When enabling Temporary Undo you may observe more space usage in the TEMP tablespace. This is because Oracle retains the TEMP segments for a user for the whole duration of the session, not just for the duration of the transaction as is the case when temp_undo_enabled is unset. This is expected behaviour. Predicting exactly how much temp undo get retained in all cases is unrealistic as this will be very dependent on database setup and what SimCorp Dimension functionality is run. If you should find the additional TEMP space outweighs the reduction you see in generation of redo/archive logs you are free to unset the temp_undo_enabled parameter (defaults to false).

When setting temp_undo_enabled to true please be aware of Oracle bug 19649152. This bug has been seen on a few cases in SimCorp to cause ORA-600 error. Oracle has fixed the bug in the 12.1.0.2 APR2015 PSU and one-off patches are available for many platforms. For more information please refer to MOS article: Bug 19649152 - ORA-600 [kturrur_0] on Primary DB with TEMP_UNDO_ENABLED=TRUE (Doc ID 19649152.8).

6.1.15 undo_tablespace

The name of the undo tablespace, e.g. UNDO

6.1.16 _optim_peek_user_binds, optimizer_adaptive_reporting_only and adaptive features

For certain SQL statements, a value of TRUE for the _OPTIM_PEEK_USER_BINDS parameter can yield very varied performance, as the execution plan is generated on the basis of the bind variable values submitted on the first passing of the statement. If the bind variable value in subsequent executions causes a significant change in the amount of rows visited (i.e. because of the actual data distribution), the execution plan chosen by the Optimizer in the first case, might no longer be optimal, and this can have a big impact on the execution time for the statement when the values of the bind variables are changed.

Especially in core SimCorp Dimension functionality such as business calculations, transaction saving etc., there is a high risk of performance being negatively impacted when this parameter is TRUE. In favour of performance stability the preferred setting for _OPTIM_PEEK_USER_BINDS has therefore been FALSE[1] .

In Oracle 12.1 many enhancements were introduced to the Oracle Optimizer. During the 12.1 verification process at SimCorp it became clear that the Automatic Reoptimization and Statistics Feedback feature[2] may, especially in an environment where the shared pool is under pressure, result in the same query seeing fluctuating runtimes. SimCorp worked together with Oracle support and the following was identified in regards to a (complex) statement after it is parsed:

1. During execution the Optimizer gathers information to determine if the query is a candidate for re-optimization.

[1] SimCorp Dimension does provide configuration possibilities to let selected functionality run in a context where the parameter gets set to TRUE for the session, thereby allowing the benefits of bind peeking in controlled areas of the system. A more detailed description can be obtained in the section on Bind Peeking, found in the SimCorp Dimension System Administrator's Manual.

[2] Ref: http://docs.oracle.com/database/121/TGSQL/tgsql_optcncpt.htm#TGSQL94983

2. At the end of execution the Optimizer compares its calculated cardinality estimates with the actual number of rows returned. If there is a significant difference the query is marked IS_REOPTIMIZABLE.
3. At the next execution the Optimizer looks for a plan replacement using the information gathered in the previous execution

This re-optimization process may be repeated several times, each time allowing the Optimizer to learn more and further improve the plan. When the Optimizer has found its final plan the query is no longer marked IS_ REOPTIMIZABLE.

Cases at SimCorp have seen up to 8 re-optimization attempts on the same query where subsequent executions progressively got longer runtimes than the earlier chosen plan and first at the 8th attempt would the Optimizer revert to the earlier identified "best" plan. This implied that every time a query had aged out of the shared pool and therefore required parsing, the runtimes of the following 7 executions differed significantly. The next to last execution before the Optimizer would revert to the earlier "better" plan could see execution times of more than +200% compared to the one using the better plan.

If the OPTIMIZER_ADAPTIVE_REPORTING_ONLY is changed from default FALSE to TRUE, the Optimizer always chooses the default plan as generated on the first parse.

The test cases also showed that the many attempts at re-optimization could to some extent be lessened by enabling bind peeking. The Optimizer would still go through the process after parse of the statement, but appeared to settle on the best plan after fewer attempts at re-optimization.

These cases have resulted in the following recommendations when running SimCorp Dimension on Oracle 12c:

- If ensuring stable reliable run-times is prioritised highest and _OPTIM_ PEEK_USER_BINDS=FALSE then OPTIMIZER_ADAPTIVE_REPORTING_ ONLY must be set to TRUE
- If memory allows for a large enough shared pool to reduce the parse frequency, and varied runtimes can be tolerated while the Optimizer through plan retries eventually finds the optimal plan, bind peeking can be enabled (i.e. unset _OPTIM_PEEK_USER_BINDS, which defaults to TRUE). In this case OPTIMIZER_ADAPTIVE_REPORTING_ONLY should remain unset at FALSE.

The table below summarises the current position in regards to the combination of the two parameter values:

	OPTIMIZER_ADAPTIVE_ REPORTING_ONLY=TRUE	OPTIMIZER_ADAPTIVE_ REPORTING_ONLY=FALSE
_OPTIM_PEEK_ USER_BINDS=FALSE	PREFERRED	NOT SUPPORTED

_OPTIM_PEEK_ USER_BINDS=TRUE	NOT TESTED	SUPPORTED

In Oracle 12.1 the SimCorp preferred value for OPTIMIZER_ADAPTIVE_ FEATURES is the default, i.e. TRUE.

In Oracle 12.2 Oracle has made further enhancements to the optimizer. The OPTIMIZER_ADAPTIVE_FEATURES parameter is split into two new 12.2 parameters: OPTIMIZER_ADAPTIVE_PLANS (default TRUE) and OPTIMIZER_ ADAPTIVE_STATISTICS (default FALSE). The SimCorp preferred value for the two new 12.2 parameters is the default.

During the verification of 12.2 SimCorp validated the performance of SimCorp Dimension on Oracle 12.2 compared with 12.1. Three different configurations on 12.2 were tested and compared with the preferred configuration on 12.1. The cases were:

1. _OPTIM_PEEK_USER_BINDS=FALSE and OPTIMIZER_ADAPTIVE_ REPORTING_ONLY=TRUE
2. _OPTIM_PEEK_USER_BINDS=TRUE and OPTIMIZER_ADAPTIVE_ REPORTING_ONLY=FALSE
3. _OPTIM_PEEK_USER_BINDS=FALSE and OPTIMIZER_ADAPTIVE_ REPORTING_ONLY=FALSE

The conclusion headlines:

- The tests do not give concern in regards to performance of SimCorp Dimension on Oracle 12.2.
- The general picture from the tests: 12.2 performance compares positively to performance on 12.1.
- The tests do not give reason to change the current guidelines.

6.1.17 _b_tree_bitmap_plans

It has been observed that in many situations the Oracle Optimizer will choose to use bitmap conversion in its execution plans. These execution plans have proven to be suboptimal and a generally better execution plan has been obtained by setting _B_TREE_BITMAP_PLANS=FALSE. For this reason a value of FALSE is recommended.

Performance benefits by setting _B_TREE_BITMAP_PLANS=FALSE, has typically been in the Data Extract and Report areas. If experience show the issues to be confined to a manageable amount of SimCorp Dimension defined data extracts (DEX) or reports, it is alternatively possible to leave _ B_TREE_BITMAP_PLANS unset at the database level, and then specify a database hint on the definition within SimCorp Dimension in the form of:

```
 /*+ OPT_PARAM('_B_TREE_BITMAP_PLANS','FALSE')*/
```

E.g. in **Extract Definitions** on the Extra tab, choose Hint in the Optimizer mode dropdown list, and add the hint on the input field, or in **Report**

Properties , menu Functions, Oracle Hints, choose Hint in the Optimizer mode dropdown list, and add[1] the hint on the input field.

6.1.18 _cursor_obsolete_threshold

In 12.2 Oracle has changed the default value for this parameter to 8192. However Oracle advices to adjust this value in 12.2 and onwards to the 12.1 default (1024) to mitigate concurrency issues on the library cache. For more information please see https://mikedietrichde.com/2018/09/11/oracle-12-2-and-higher-set-_cursor_obsolete_threshold-to-old-default/and My Oracle Support: High Version Counts For SQL Statements (>1024) Post Upgrade To 12.2 and Above Causing Database Slow Performance (Doc ID 2431353.1).

6.2 Database Automatic Jobs

When creating the Oracle database, by default Oracle schedules and enables the following automated maintenance tasks: Optimizer Statistics Gathering, Segment Advisor and Automatic SQL Tuning.

These tasks are set to run each day of the week in a daily maintenance window defined on weekdays for a 4 hour period from 22:00-02:00 and at weekend days for a 20 hour period from 06:00-02:00. The daily maintenance windows are associated with a default maintenance resource plan, but the plan is not activated by default. The plan defines the percentage of CPU resources to allocate and prioritise between the maintenance tasks. You are encouraged to activate this plan. If not, the automated maintenance tasks will utilise all available CPU resources during the maintenance windows whilst the jobs are running. You are still strongly advised to monitor resource utilisation during the maintenance windows, even if the resource plan has been activated.

Please note that during the maintenance windows on weekends, which have a duration of 20 hours, if for example the statistics job completes faster, it is by default configured to restart every 4 hours.

You must also take into account the maintenance tasks run by SimCorp Dimension. In a typical SimCorp Dimension production set-up evening, night time and weekends are used for heavy batch and reporting loads. Therefore you may need to change the time and/or frequency when the Oracle maintenance tasks are run in order to accommodate the load on the database from SimCorp Dimension.

With knowledge of the SimCorp Dimension production cycle and after having monitored and gathered activity performance information from the database in the maintenance window periods you may want to customize the maintenance jobs, windows and schedules to fit your actual database load.

Finally a word of warning: Oracle11g introduced several new automation options for SQL tuning through various checker processes and the possibility to automatically implement the resultant suggested changes. If

[1] In Report Properties you must leave off the hint markers (e.g. no /*+ */) as these are added automatically at runtime by the report interface

you, as an experienced DBA, decide to enable these checkers you should do so only after very careful consideration in regards to SimCorp Dimension. And all system implemented changes should be documented and communicated to SimCorp Support in cases of reporting database issues. And please also note that changes to the SimCorp Dimension data schemas are only supported if the change is implemented through SimCorp Dimension itself.

6.3 Database Optimizer Statistics

6.3.1 Changes to the Default Parameters for Statistics Gathering Job

Oracle automatically captures Optimizer statistics in the predefined daily maintenance windows. The built-in logic of the job ensures that statistics are only gathered on objects with missing or stale statistics.

For SimCorp Dimension operations it is recommended that the parameters **DEGREE** and **METHOD_OPT** are changed and if the Partitioning module is installed for SimCorp Dimension, also the **CONCURRENT** parameter.

Statistics preference:	Value changed to:
DEGREE	AUTO_DEGREE (or an appropriate value > 1)
METHOD_OPT	FOR ALL INDEXED COLUMNS SIZE AUTO
CONCURRENT	If partitioning is used: ALL

Note The default value for preference PUBLISH is TRUE. If this is changed to FALSE it can lead to serious performance issues following SimCorp Dimension upgrade and patching, as the application during these operations issues statistics gathering commands and rely on the statistics being published.

To get the current (global) values for these parameters use the following SQL:

```
Select DBMS_STATS.Get_Prefs('DEGREE') Degree,
       DBMS_STATS.Get_Prefs('METHOD_OPT') GlobalMethod,
       DBMS_STATS.Get_Prefs('CONCURRENT') Concurrent
From DUAL;
```

To see schema level preferences add the schema name to the Get_Prefs call, e.g.:

```
DBMS_STATS.Get_Prefs('DEGREE',‘SCDAT’)
```

Add table name to view preferences on table level, .e.g.:

```
DBMS_STATS.Get_Prefs('DEGREE',‘SCDAT’,’TRANSMAIN’)
```

For detailed documentation of the DBMS_STATS package refer to the Oracle database documentation: ***PL/SQL Packages and Types Reference***.

The SimCorp Dimension preference is to set these parameters on global level. If they are not set on global level the settings are lost for any new tables created following, see My Oracle Support note: After Setting DBMS_STATS.SET_SCHEMA_PREFS, GET_PREFS does not Return the Current Configuration Values (Doc ID 1338709.1). This will cause problems for SimCorp Dimension, for example after upgrades when SimCorp Dimension tables have been altered using CTAS.

Use the DBMS_STATS to change the default value, e.g.:

```
Exec DBMS_STATS.Set_Global_Prefs('DEGREE','DBMS_
STATS.AUTO_DEGREE');
Exec DBMS_STATS.Set_Global_Prefs('METHOD_OPT','FOR ALL
INDEXED COLUMNS SIZE AUTO');
Exec DBMS_STATS.Set_Global_Prefs('CONCURRENT','ALL');
```

This preference is to avoid ending up with tables where statistics are gathered using the default database method (FOR ALL COLUMNS SIZE AUTO). This has on many occasions been observed to lead to suboptimal execution plans. When preferences are changed on schema (or table) level it only affects existing objects in the schema. Any new objects added to the schema following the change will still inherit its statistics preference from the global settings. If these parameters are only changed on schema or table level (not global) it implies a manual maintenance burden, as the schema (or table) preferences must be reset after each SimCorp Dimension upgrade (and potentially patch application). In addition schema statistics must be deleted and re-gathered using METHOD_OPT FOR ALL INDEXED COLUMNS SIZE SKEWONLY (ref section Re-establish Statistics after Changes to Statistics Preferences) as both the database default and the SimCorp Preferred Configuration method will fail to gather correct statistics for those tables updated during the upgrade (or patch) using “full copy” method.

The following sub sections explain the motivation behind each of the preferred values:

6.3.1.1 Degree

The out-off-the-box default value for the statistics preference parameter DEGREE is NULL, which means Oracle will use the table properties parallel clause parameter to determine the degree to utilise during statistics gathering. As all SimCorp Dimension tables are created with NOPARALLEL clause this implies a degree of 1 meaning only one single process will work on gathering statistics if DEGREE is left to NULL. This seriously prolongs the time required to gather statistics on the SimCorp Dimension schemas. If left un-changed, the timeframe for the allocated maintenance window may not be enough to complete the operation and over time the table statistics will grow increasingly stale resulting in poor system performance. So to allow the statistics gathering job to utilise parallel queries, statistics preference DEGREE should beset to an appropriate value.

6.3.1.2 Method_opt

The reason for using FOR ALL INDEXED COLUMNS SIZE AUTO as the preferred value, is that SimCorp has extensive experience from many client sides, showing that statistic/histograms on all columns appear to have caused flawed decisions by the Oracle optimizer for SimCorp Dimension. In these investigated cases, it has been seen that removing the histograms from all columns and 'only' having histograms on the indexed columns (on the tables involved in the query), changed the Optimizers decision for the better. So these experiences, together with the knowledge that by far the majority of queries from SimCorp Dimension have predicates on indexed columns, and that in most cases bind variables are used and that bind peeking is disabled, have led to this general recommendation.

6.3.1.3 Concurrent Statistics

The default value for the global preference CONCURRENT is OFF, meaning Oracle collects statistics on one object at a time. Setting CONCURRENT allows Oracle to gather statistics on multiple tables or on multiple partitions within a table concurrently. For SimCorp Dimension, especially the time for gathering statistics on sub partitions is greatly improved by setting the global preference, CONCURRENT with the DBMS_STATS package.

The global preference for CONCURRENT can take several values. The SimCorp Dimension preferred value (if partitioning is implemented) is ALL, meaning concurrency is enabled both during manual and automatic statistics gathering. The Oracle Scheduler determines how many concurrent tasks to run based on the value of JOB_QUEUE_PROCESSES.

Further the Resource Manager must be enabled and additional prerequisites apply. By default the Resource Manager is disabled. If you do not have a resource plan Oracle recommends that you enable the Resource Manager with the system-supplied DEFAULT_PLAN, e.g.:

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'DEFAULT_PLAN'
scope=both;
```

Note, if concurrent statistics is enabled Manage Any Queue privilege is a prerequisite for database users to gather table statistics. This implies, if Concurrent=ALL, Manage Any Queue must be granted to the SimCorp Dimension:

- user role (e.g. SCROLE)
- data owner (e.g. SCDAT)
- system owner (e.g. SCSYS)
- any data extract and data warehouse schemas created from the installation

The Manage Any Queue privilege to SimCorp Dimension schemas and role is from version 6.4 no longer maintained by the application, meaning **Verify Database Privileges** will neither grant nor revoke the privilege. Instead the privilege must be maintained by the DBA in accordance to the above.

If Manage Any Queue privilege has not been granted and Concurrent is ALL, any attempt from the application to gather statistics will fail with:

ORA-20000: Unable to gather statistics concurrently: insufficient privileges

For further guide lines please refer to the documentation[1] on enabling concurrent statistics gathering in Oracle12c.

6.3.2 Re-establish Statistics after Changes to Statistics Preferences

In several instances severe performance issues have been observed in SimCorp Dimension relating to the Oracle optimizer choosing inadequate execution plans as a result of statistics having been gathered using the out-of-the-box settings. Per default Oracle will, for example, gather histogram statistics on all columns, not just indexed columns. Even if the job defaults are changed to the preferred settings, as specified in the previous section, the 'old' histogram statistics on non-indexed columns will remain. As the job is now configured to only gather statistics on the indexed columns, the histograms on the non-indexed columns will become increasingly stale. And this can have serious implications with regard to the Optimizer's judgement.

It is therefore recommended, after changing statistics preferences, to delete statistics from the SimCorp Dimension schemas. Updated statistics may then be gathered afresh.

To re-establish the statistics the following can be used as a guideline:

1. Log on the database as SYS
2. For each SimCorp Dimension schema[2] run: DBMS_STATS.DELETE_SCHEMA_STATS ('<schema>'');
3. Ensure the default parameters have been changed to the preferred settings as specified in section Changes to the Default Parameters for Statistics Gathering Job
4. Start the statistics job by running: DBMS_STATS.GATHER_DATABASE_STATS;

As all statistics are deleted first, it may take considerable time to complete step 4. It is therefore recommended to run and time this operation in a representative test system, thereby finding an appropriate window, during which the operation can be completed in the production environment. To

[1] E.g. the Database SQL Tuning Guide, chapter 12.4.7.2 Enabling Concurrent Statistics Gathering – Prerequisites (https://docs.oracle.com/database/121/TGSQL/tgsql_stats.htm#TGSQL428).

[2] E.g.: SCDAT, SCSYS and all data extract schemas

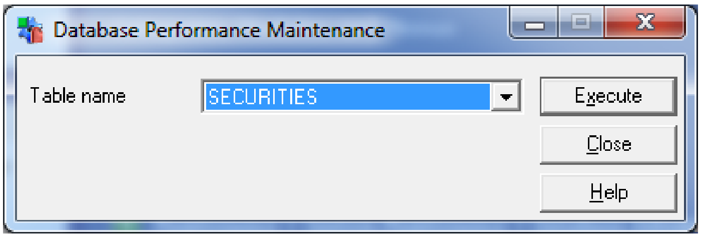
reduce the run time for step 4, the following can be used in step 2 instead of a complete deletion of all schema statistics:

```
DECLARE
CURSOR tCur IS
SELECT STAT.Owner, STAT.Table_Name tab, STAT.Column_
Name Col
FROM   ALL_TAB_COL_STATISTICS STAT, ALL_TAB_COLUMNS
TABC
WHERE  STAT.Owner IN ('SCDAT','SCSYS')
AND    STAT.Owner = TABC.Owner
AND    STAT.Table_Name = TABC.Table_Name
AND    STAT.Owner = TABC.Owner
AND    STAT.Column_Name = TABC.Column_Name
AND    substr(STAT.Table_Name,1,4) != 'BIN$'
MINUS
SELECT Table_Owner Owner, Table_Name Tab, Column_name
col
FROM   ALL_IND_COLUMNS
WHERE  Table_Owner IN ('SCDAT','SCSYS')
ORDER BY 1,2,3;
BEGIN
  FOR tRec IN tCur LOOP
    DBMS_STATS.Delete_Column_Stats
(ownname=>tRec.Owner,tabname=>tRec.tab,colname=>tRec.Co
l,force=>TRUE);
  END LOOP;
END;
/
```

Note If the operation is done following creation of a new database and import of data, the DBMS_STATS.GATHER_DATABASE_STATS call should be run with METHOD_OPT set to FOR ALL INDEXED COLUMNS SIZE SKEWONLY. Using the AUTO option in a newly created database is risky since histograms might not be gathered on columns due to lack of workload information.

6.3.3 Collecting Schema Statistics Manually from SimCorp Dimension

You may encounter situations where it is necessary to generate fresh Optimizer statistics on a singular table. You can use Database Performance Maintenance in SimCorp Dimension for this purpose as shown below. Statistics will be gathered for the table using the configured defaults, e.g. for DEGREE and METHOD_OPT. In addition, the job will delete already existing statistics on the table before gathering the new statistics.



Alternatively, you can use Database Control, PL/SQL or any other tool capable of executing DBMS_STATS to gather statistics on individual tables.

6.3.4 Stale Index Statistics are not Refreshed in some Situations

During testing, it has been discovered that stale index statistics are not refreshed during the automatic statistics gathering job (auto optimizer stats collection) unless the corresponding table statistics are stale as well. According to Oracle Support this is intended behaviour. When using SimCorp Dimension you might end up in a situation where the table statistics are fine, but the index statistics are stale. In that case, manual intervention is needed in order for the statistics to be refreshed. SimCorp is working on optimizing our code, so this should not occur, and have requested an enhancement to Oracle Database so stale index statistics are refreshed, regardless of the state of the corresponding table statistics.

In addition, it has come to our attention that there has been a change of behaviour of the procedure DBMS_STAT.GATHER_DATABASE_STATS from Oracle Database 11g to Oracle Database 12.1.0.1. In the 11g version, the procedure would gather statistics on indexes if the index statistics were stale, regardless of the state of the table statistics. In Database 12.1.0.1 and onwards stale index statistics are only refreshed if the corresponding table statistics are stale as well.

Currently, it has not been decided if and when the enhancement will be part of the Oracle Database. Consequently, SimCorp recommends that you manually gather statistics for those indexes that have stale statistics. In addition, SimCorp recommends that the gathering is executed in the same maintenance window as the automatic statistics gathering job.

6.3.5 Statistics Advisor

Part of the work verifying Oracle Database 12.2 at SimCorp involved executing the new Statistics Advisor on a SimCorp Dimension database to learn from any recommendations made by Oracle. Below is a list of findings from the advisor together with the comments from SimCorp for each finding.

The test was executed on a SimCorp Dimension database that was strictly configured according to Preferred Oracle Configuration.

Rule Name: UseDefaultPreference

Rule Description: Use Default Preference for Stats Collection

Finding: Global preference METHOD_OPT is set to a non-default value 'FOR ALL INDEXED COLUMNS SIZE AUTO'.

The predicates used in SELECT statements are almost always on indexed columns which means that it has no real benefit with histograms on non-indexed columns; knowing they are never used. SimCorp is aware of the possibility of using table preferences rather than global preferences. Whether it will be implemented is determined by development priorities.

Rule Name: AvoidSetProcedures

Rule Description: Avoid Set Statistics Procedures

Finding: There are x SET_[COLUMN|INDEX|TABLE|SYSTEM]_STATS procedures being used for statistics gathering.

SimCorp recommends fixing statistics on four tables in the schema SYS due to a known issue with bad cardinality estimates on queries issued by DBMS_SCHEDULER. Please refer to the known issues section in System Administrators Manual for details. In all other cases SimCorp Dimension does not set or recommend setting statistics manually.

Rule Name: UseDefaultParams

Rule Description: Use Default Parameters in Statistics Collection Procedures

Finding: There are x statistics operation(s) using nondefault parameters.

In certain situations, SimCorp Dimension will gather statistics using other preferences than the default. An example is during upgrade where statistics are gathered with METHOD_OPT=FOR ALL INDEXED COLUMNS SKEWONLY after a table has been re-created using CTAS. The “SIZE AUTO” feature doesn’t work properly on a newly created table because it is a new object, and thus, Oracle has no information about the data characteristics and will not create histograms. Hence, “SKEWONLY” is used to get decent histograms until better stats can be gathered at a later point using “SIZE AUTO”. The same applies when adding indexed columns. Another example is tables used by Align Objects. Due to the characteristics of these data, statistics gathering are using non-default parameters.

Rule Name: UseGatherSchemaStats

Rule Description: Use gather_schema_stats procedure

Finding: There are x uses of GATHER_TABLE_STATS.

The options available when using GATHER_TABLE_STATS is sufficient for SimCorp Dimension. SimCorp is aware of the additional options available with GATHER_SCHEMA_STATS and might choose to implement use hereof in future versions.

Rule Name: AvoidUnnecessaryStatsCollection

Rule Description: Avoid unnecessary statistics collection

Finding: Statistics are being gathered on x object(s) when they are not stale.

In some situations, SimCorp Dimension might choose to gather statistics even if the statistics are not recognized by Oracle Database as stale. The staleness information is persisted from memory at regular intervals, but until that the staleness information is not updated in the USER_TAB_ STATISTICS view. In Oracle Database 12.2 the staleness information should be immediately available in the USER_TAB_STATISTICS view, and SimCorp is currently nvestigating whether that information can be used to rule out situations where statistics are gathered on non-stale objects.

Rule Name: AvoidStaleStats

Rule Description: Avoid objects with stale or no statistics

Finding: There are x object(s) with stale statistics.

During normal operations, it is the database itself that is responsible for re-gathering statistics when they are stale, and it is normally done automatically in predefined maintenance windows. It is normal that statistics get stale during the use of the database, whereas the re-gathering of statistics is only done in the before-mentioned maintenance windows.

Following a SimCorp Dimension upgrade there might be objects in the database with stale statistics. Some data might be changed after the execution of Align Objects (which does the statistics gathering), and in some cases SimCorp Dimension doesn't deem it necessary to gather statistics. Also, previous attempts to implement statistics gathering as the last part of the upgrade had a severe performance penalty. The last part of the upgrade is recreation of the main workspace, and it does make sufficient data changes for some tables to become stale. However, the tables are small, and the data changes doesn't have an impact on the quality of the statistics.

Rule Name: UnlockNonVolatileTable

Rule Description: Statistics for objects with non-volatile should not be locked

Finding: Statistics are locked on x table(s) which are not volatile.

SimCorp Dimension is not locking statistics tables. However, SimCorp Dimension does use Advanced Queueing functionality and the underlying queue tables are created with locked statistics by the database itself. There is an inconsistency between the implementation of Advanced Queueing in the database, and the recommendations of the statistics advisor.

Rule Name: MaintainStatsConsistency

Rule Description: Statistics of dependent objects should be consistent

Finding: There are x indexes which have inconsistent statistics with their column statistics.

When a table is undergoing changes during Align Objects the process is split into pieces to parallel the work using the database scheduler. A consequence of this is that statistics on indexes that are used to maintain referential integrity constraints are gathered when they are created, whereas the table statistics are gathered at a later point in time. This is considered by Oracle Statistics Advisor as an inconsistency, however, since Align Objects is running in maintenance windows only there can be no change to the data in the tables in the period between index creation (and implicit statistics gathering) and gathering of table statistics (with cascade set to false). Hence, it is no practical impact on the quality of the statistics. In some situations, the index statistics can become stale without the same applies for the corresponding table. Since the GATHER_

DATABASE|SCHEMA|TABLE_STATS are only looking at the staleness of the table statistics, the database might end up in a situation where the stale (or missing) index stats are never re-gathered. SimCorp is aware of several situations in which the issue has been the root cause of performance problems.

Since the workaround is to gather those statistics using GATHER_INDEX_STATS, you might end up with inconsistent column statistics (table vs. index), however, statistics that (possibly) are slightly inaccurate is better than no statistics at all. Please refer to System Administrators Manual for more details of the known issue.

Rule Name: MaintainStatsConsistency

Rule Description: Statistics of dependent objects should be consistent

Finding: There are x object(s) which have inconsistent statistics for number of distinct values with the columns in their referenced tables.

Tables prefixed AO_ and DDXML contain data needed by Align Objects. The data stored in these tables has special characteristics and thus SimCorp has chosen to gather statistics in a non-default manner.

Rule Name: MaintainStatsConsistency

Rule Description: Statistics of dependent objects should be consistent

Finding: There are x object(s) which have inconsistent statistics for column minimum values with the columns in their referenced tables.

For tables prefixed AO_ and DDXML see note on “There are x object(s) which have inconsistent statistics for number of distinct values with the columns in their referenced tables”.

Rule Name: MaintainStatsConsistency

Rule Description: Statistics of dependent objects should be consistent

Finding: There are x object(s) which have inconsistent statistics for column maximum values with the columns in their referenced tables.

For tables prefixed AO_ and DDXML see note on “There are x object(s) which have inconsistent statistics for number of distinct values with the columns in their referenced tables”.

Rule Name: UseIncremental

Rule Description: Statistics should be maintained incrementally when it is beneficial

Finding: Incremental option should be used on x object(s) for statistics gathering.

Recently, SimCorp has evaluated the use of incremental statistics in Oracle Database 12.1. The feature was not found production-ready due to an extensive list of bugs and known issues; especially the impact of the bugs on the database taken into consideration. For Oracle Database 12.2 it appears that the feature is much more mature and with the introduction of the HYPERLOGLOG algorithm the foot print in SYSAUX tablespace is considerably smaller, see the section Incremental Statistics Gathering on the next page.

Rule Name: AvoidAnalyzeTable

Rule Description: Avoid using analyze table commands for statistics collection

Finding: There are x object(s) using "ANALYZE TABLE <table_name> [COMPUTE | ESTIMATE] STATISTICS" to gather statistics.

SimCorp Dimension does not use ANALYZE TABLE command. Any such occurrences listed must originate from users manually executing the command outside of SimCorp Dimension.

6.3.6 Incremental Statistics Gathering

From a SimCorp Dimension point incremental statistics gathering is recommended for partitioned tables only. It saves time and resources during statistics gathering. Candidate table groups in SimCorp Dimension are Logging and Audit Trail. The table groups cover the following tables:

- Audit Trail: AUDITMASTER, AUDITDETAIL, AUDITSECURITIES, FOUREYESAPPROVED, FOUREYESUNAPPROVED, AUDITMASTEREXT, AUDITDETAILEXT
- Logging: LOGMESSAGES, LOGCONTEXTWFITEMS, LOGMSGPROPKEYPAIRS, LOGMSGPROPKEYPAIRS

It is advised that incremental statistics is enabled in 12.2 databases only. The list of bugs in 12.1 and lower is very long, and the issues encountered severely affect performance.

The default behaviour is that on partitioned tables with incremental statistics, the statistics are consider stale as soon as there are any DML changes to that partition. The preference “INCREMENTAL_STALENESS” should be used to ensure that statistics are not stale until a certain stale percent has been reached. The default of 10 % should be sufficient, but the preference for “STALE_PERCENT” can be changed for customized needs.

It makes the most sense to enable incremental statistics on range partitioned tables, or referenced-partitioned tables where the parent table is range partitioned, or list partitioned tables where the lists are created as month intervals.

In a testing setup with 6 partitions in LOGMESSAGES (6 months of data, average partition row count = 292.111.099) the space needed in SYSAUX for the synopsis table was:

- Adaptive Sampling (12.1): 15 MB
- HyperLogLog (12.2): 2 MB

Therefore please note, incremental statistics comes at the cost of SYSAUX space usage, and you should monitor the disk usage, and make tests in your own environment to verify the results.

The preferences should not be set at global level (SCHEMA or DATABASE) as it has too wide impact. Instead the preference must be set at table level using DBMS_STATS.SET_TABLE_PREFS. You should here note, that the preference will only apply on the current object. If the table is full copied (create table as select) for example in an SimCorp Dimension upgrade, DBMS_STATS.SET_TABLE_PREFS must be re-executed again following. SimCorp is evaluating the possibility of making this part of the code in a future version.

The following provides an example on how to enable (or disable) the feature for the LOGMESSAGES table in the data owner schema SCDAT:

To enable:

```
EXEC DBMS_STATS.SET_TABLE_PREFS (ownname=>'SCDAT',
tabname=>'LOGMESSAGES', pname=>'INCREMENTAL',
pvalue=>'TRUE');
EXEC DBMS_STATS.SET_TABLE_PREFS (ownname=>'SCDAT',
tabname=>'LOGMESSAGES', pname=>'INCREMENTAL_STALENESS',
pvalue=>'USE_STALE_PERCENT');
EXEC DBMS_STATS.SET_TABLE_PREFS (ownname=>'SCDAT',
tabname=>'LOGMESSAGES', pname=>'PUBLISH',
pvalue=>'TRUE');
EXEC DBMS_STATS.SET_TABLE_PREFS (ownname=>'SCDAT',
tabname=>'LOGMESSAGES', pname=>'APPROXIMATE_NDV_
ALGORITHM', pvalue=>'HYPERLOGLOG');
```

To disable:

```
EXEC DBMS_STATS.SET_TABLE_PREFS (ownname=>'SCDAT',
tabname=>'LOGMESSAGES', pname=>'INCREMENTAL',
pvalue=>'FALSE');
```

6.4 Other Parameters for Special Purposes

6.4.1 Automatic Workload Repository

The Automatic Workload Repository (AWR) is an Oracle built-in repository used to store detailed database performance related statistics allowing advanced performance reporting and troubleshooting. The AWR information is gathered and stored continuously and kept for a configured period of time (retention period). AWR requires the Diagnostic Pack option of Oracle Enterprise Edition. If your Oracle license includes the Diagnostic Pack option, SimCorp strongly encourages you to use AWR.

Please ensure you set a retention period of min. 31 days, preferably 180 days, and with snapshots taken at least every 60 minutes, e.g.:

```
Exec DBMS_WORKLOAD_REPOSITORY.Modify_Snapshot_Settings
(INTERVAL=>60, RETENTION => 259200);
```

Please note the Oracle CONTROL_MANAGEMENT_PACK_ACCESS initialisation parameter controls which manageability packs that are active. The default value is DIAGNOSTIC+TUNING, thereby assuming you have licence options for both the Diagnostic Pack and the Tuning Pack. If your license conditions do not include these options you are encouraged to, at least, consider obtaining a licence for the Diagnostic Pack. This can greatly ease investigations into performance related issues. CONTROL_ MANAGEMENT_PACK_ACCESS can, besides the default, take the values: DIAGNOSTIC or NONE.

6.4.2 Database Profiles

During installation of SimCorp Dimension two database profiles are created. The profile names are as defined in the application's Cnf.ini file as e.g:

- userprofile=SCUSERPROFILE
- systemprofile=SCSYSPROFILE

The user profile is assigned regular SimCorp Dimension users who sign on as interactive sessions. The system profile is assigned special purpose SimCorp Dimension users . These users cannot start interactive sessions, but are used for automation, e.g. SERVER and BATCH used for SimCorp Dimension services and batch jobs.

It is generally recommended you leave these two profiles to be managed through SimCorp Dimension. If changes are required then please note:

- Setting IDLE_TIME is not supported and SimCorp Dimension will not function if set. Instead set an idle timeout for SimCorp Dimension users through System Log Off Options in the application.
- PASSWORD_VERIFY_FUNCTION is supported as long as the applications generated passwords can pass. The Default password flag must be set in System Security Options in the application.
- PASSWORD_LIFE_TIME is supported, however if set in the system profile, passwords for SERVER and BATCH must be reset through the application before the time out. E.g. an auto job executing scd.exe -job=SYNCDBPASSWORDS.
- If required set, the following can be set through Security Options in the application:
  - PASSWORD_GRACE_TIME
  - PASSWORD_REUSE_TIME
  - PASSWORD_REUSE_MAX
  - FAILED_LOGIN_ATTEMPTS

Please refer to the SimCorp Dimension System Administrator's Manual for more about the application usage and configuration.

6.4.3 cpu_count

There can be environmental reasons to specify a non-default value for CPU_COUNT. Per default this parameter is the total number of CPU cores/threads the database sees on the host. The benefit of using the default setting is that Oracle continuously monitors the number of CPU cores/threads reported by the operating system and thus uses the current value, allowing dynamic CPU reconfiguration.

CPU_COUNT has an impact on the default PARALLEL_MAX_SERVERS value and DOP (default degree of parallelism) used for statements running parallel query/execution. When SimCorp Dimension calculates the degree to use, it also uses CPU_COUNT in its calculation.

If the database is running on a host housing multiple databases, not necessarily all CPUs are available for the database. In this case it may be desirable (or even required) to lower CPU_COUNT to a value reflecting the amount of CPU cores/threads that are specifically available for the database. For example, a host with 60 CPU cores is shared between 3 test databases and the 3 database instances consume CPU equally. In this case CPU_COUNT in the 3 test database can be reduced to 20, thereby controlling the amount of spawned processes during parallel query/execution.

Many other parameter values are derived from CPU_COUNT and many of these control resource consumption. By letting Oracle believe it has more resources than actually available there is a serious risk of resource starvation on the host.

6.4.4 filesystemio_options

With Direct I/O, Oracle can bypass the Operating System’s file cache. This is desirable for various reasons. Most importantly, it makes better use of the available memory because with an OS file cache, data blocks are buffered twice; in Oracle’s memory and in the OS’s memory.

If you are migrating from buffered I/O to Direct I/O, you should take the following into consideration:

1. The memory that the OS used for its file cache should be given to Oracle instead. Read I/Os will be slower because each I/O will be a true physical I/O, not satisfied by an OS cache lookup. By giving the memory to Oracle, the number of I/Os will decrease so the net result will be fewer but slower read I/Os and a better utilisation of the available memory.
2. You cannot use “day 1” for reasonable comparison since enabling Direct I/O requires a database bounce. You will therefore start with a cold buffer cache and an increased amount of physical I/O until the buffer cache is warm.

On Windows, Direct I/O is automatically enabled. On other platforms, Direct I/O can be enabled by setting FILESYSTEMIO_OPTIONS to an OS dependent value along with other potential changes to the OS configuration (disabling OS file caching on file systems).

Please consult the Oracle platform specific documentation for instructions on how to enable Direct I/O correctly.

6.4.5 Flashback Recovery Area

A flashback recovery area offers a number of benefits in terms of manageability and backup/recovery options. SimCorp Dimension does not have any specific requirements in regards to set up a flash recovery area, but if you can afford the disk space and wish to use a flashback recovery area, you are encouraged to do so. For more information see My Oracle Support article: What is a Flash Recovery Area and how to configure it? [ID 305648.1]

6.4.6 Flashback Database

Enabling and using the Oracle Flashback Database feature is strongly discouraged in SimCorp Dimension production environments during operations. This is due to the complex nature of the SimCorp Dimension application and business usage. To give just one example, if flashback is done to a point prior to a print of a dealer slip, the transaction data will be reverted but the dealer slip still exists as the print cannot be reverted.

During maintenance operations (patch apply, upgrade, changing partition options etc.) Oracle Flashback Database can be a useful feature as it offers significantly faster restores. Likewise in test system Oracle Flashback Database can be used for repeat test scenarios where the exact same point of origin is required. In these situations provided sufficient flashback retention target and available storage is available, Oracle Flashback Database can be enabled.

6.4.7 parallel_max_servers and parallel_servers_target

The preferred values for PARALLEL_MAX_SERVERS is the Oracle default value (i.e. leave this parameters unset).

There are certain situations and conditions where adjusting this parameter may seem relevant. E.g. if the database instance shares physical host resources with other database instances, PARALLEL_MAX_SERVERS (and JOB_QUEUE_PROCESSES, ref section job_queue_processes) can be reduced during SimCorp Dimension upgrades to limit the amount of concurrently running database sessions and parallel processing.

Please note, setting PARALLEL_MAX_SERVERS too low can seriously affect the time required for upgrading SimCorp Dimension.

As an alternative to setting a non-default value for PARALLEL_MAX_ SERVERS, adjusting CPU_COUNT (ref section cpu_count) may be a better choice. Alternatively the SimCorp Dimension installation can be configured (through Cnf.ini file parameters) to control the parallel degree issued by statements from the application. Please refer to the sections on Parallel Options and Align Objects Settings in the ***SimCorp Dimension System Administrator's Manual***.

Before considering to change PARALLEL_MAX_SERVERS please read the ***SimCorp Dimension Upgrade*** technical papers: ***Controlling database resources*** and ***Understanding the upgrade engine***, available on the SimCorp Client Support web site.

Note, from version 6.0 the SimCorp Dimension upgrade engine will apply statement queuing hint on parallel query and execution statements. This means if a statement with parallel hint is issued and all Oracle parallel processes are currently engaged with different statements, then Oracle will queue the statement waiting for the parallel processes to become available, rather than executing the statement in a single process. Oracle uses the PARALLEL_SERVERS_TARGET parameter as a gauge when to queue a given statement. This enhancement to SimCorp Dimension removes the risk of a statement, which should have benefitted from parallel processing, being executed by a single process and at same time ensures the system is not flooded with parallel processes. The preferred value for PARALLEL_SERVERS_TARGET is to let Oracle default as with PARALLEL_MAX_SERVERS.

6.4.8 plsql_code_type

Especially one parameter, PLSQL_CODE_TYPE, is promoted by Oracle as a performance enhancing parameter. This feature gives the possibility for PLSQL code to be natively compiled machine code, thereby bypassing the PL/SQL interpreter engine and the overhead incurred when interpreted code is executed.

Tests of SimCorp Dimension have, however, failed to show any significant differences in SimCorp Dimension execution times with native PLSQL code. It is therefore, at present, not possible to conclude whether changing this parameter from INTERPRETED (default) to NATIVE will have an effect on SimCorp Dimension performance. But you may want to conduct your own tests to establish if your configuration would benefit from using native compilation. SimCorp would be happy to hear any resulting conclusions and experiences.

Note Testing the impact of PLSQL_CODE_TYPE=NATIVE must be done in a test system, otherwise identical to the production configuration. Subsequent implementation in the Production environment would be dependent on any observed performance benefits.

Please note that if you change PLSQL_CODE_TYPE from INTERPRETED to NATIVE, it will not have any effect on already complied PL/SQL library units. The units have to be natively compiled beforehand, as described in the Oracle documentation.

6.4.9 remote_os_authent and os_authent_prefix

The parameter REMOTE_OS_AUTHENT must be set to TRUE if SimCorp Dimension is configured for OS authentication. And in this context, please note that SimCorp Dimension does not support OS_AUTHENT_PREFIX being a blank character.

Oracle has deprecated remote OS authentication and the solution in SimCorp Dimension is only maintained for backward compatibility. As an alternative for single sign on solution SimCorp Dimension can be configured to use Oracle Kerberos Authentication if the Active Directory Authentication module is installed. In this case REMOTE_OS_AUTHENT must be FALSE (default value) and OS_AUTHENT_PREFIX must be set to a blank character.

Please refer to the ***SimCorp Dimension System Administrator's Manual***, the section on unattended logon for more information.

6.4.10 Report Performance

Very occasionally SimCorp has had reports of clients solving specific performance issues by setting the parameters listed below:

- _new_initial_join_orders = false
- _push_join_predicate = false

If there are specific reports or data extracts relying on these parameters, they can be provided as an Oracle OPT_PARAM hint directly on the definition in SimCorp Dimension, for example: /*+ OPT_PARAM('_new_ initial_join_orders ','FALSE') */. It is not recommended to set these parameters on the database level.

6.4.11 Result Cache

SimCorp recommend that the miscellaneous result cache parameters are kept unchanged (i.e. at default). Some SimCorp Dimension statements take advantage of manual hinting for result caching. Setting, for example, result_cache_max_size to 0 will disable the usage of the server side result cache and the intended performance gain will be lost.

6.4.12 shared_servers and dispatchers

SimCorp Dimension sessions rely on dedicated database connections. If the Oracle parameter DISPATCHERS is set, SHARED_SERVERS will default to 1 meaning shared server architecture is enabled. If the service used for the SimCorp Dimension connections is part of the dispatchers list the connection will be done using shared servers. This can have a serious impact on runtime performance. Please refer to the known issue described in the ***SimCorp Dimension System Administrator’s Manual***.

6.4.13 SQL Profiles

Most database statements from SimCorp Dimension are dynamically constructed by the application database interface. Especially the statement conditions are built by piecing information from several sources. For example, in a data grid form the resulting condition consists of the following parts AND’ed together: a static condition from the application, a user dependent authorisation condition and a user specified condition (e.g. data entered in a screen's “search mode” where-after Enter is selected). For this reason it is very unlikely a SQL profile/plan will work indefinitely for a given SimCorp Dimension generated SQL. At most, profiles can be used as a very temporary solution to an acute issue, until the right solution is found, but they should never become part of a permanent solution.

6.4.14 SQLNET Parameters

Running SimCorp Dimension does not as a rule pre-requisite Sqlnet.ora configuration, as in most cases the SQLNET parameter default values will suffice. However it is strongly recommended to enable dead connection detection with SQLNET.EXPIRE_TIME as described SQLNET Parameters. Additionally some other SQLNET parameters are listed, which may be useful to configure given certain conditions. The information should be viewed as ‘known issues’ and these parameters only need to be set if the described situation applies.

The Sqlnet.ora used by SimCorp Dimension (i.e. the client side) is the one situated in the path indicated by the tnsadmin entry in the installation’s Cnf.ini. If this entry is missing or empty, a TNS_ADMIN string entry in Windows Registry location HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE must point to the folder containing the Sqlnet.ora file (or an environment variable for TNS_ADMIN must exist).

The Sqlnet.ora used by the database (i.e. server side) is as configured by the DBA. Default location is the NETWORK ADMIN folder in ORACLE_ HOME, but this may vary.

6.4.14.1 SQLNET.EXPIRE_TIME=10

A number of issues have been seen where connections to the Oracle database server have been lost/cut without the Oracle database server discovering the missing connection. Setting SQLNET.EXPIRE_TIME to e.g. 10 in server side sqlnet.ora file will force a check each 10 minutes for dead connection on the server side and if found cleanup these dead connections. See more in http://www.oracle.com/technetwork/database/enterprise-edition/oraclenetdcd-2179641.pdf.

6.4.14.2 SQLNET.SEND_TIMEOUT=720

If you have patched the database up to 11.2.0.4 or 12.1 or have a new installation of these versions, you may find that you start getting ORA-12592 errors reported on the Client side. There may also be other errors or issues such as aborts or timeouts (Client and/or DB side). This can be due to Oracle Bug 18841764 - Network related error like ORA-12592 or ORA-3137 or ORA-3106 may be signaled (Doc ID 18841764.8).

The parameter is set in both client and server side Sqlnet.ora. The value itself is irrelevant but should be a large value. This has almost no impact on anything other than disabling the bug, and so can be used as a permanent workaround until you can receive a patch from Oracle for the bug. At SimCorp a value of 720 (12 minutes) has been sufficient work-around. Some customers have reported setting this to 6000 to work-around the bug.

6.4.14.3 SQLNET.INBOUND_CONNECT_TIMEOUT=0

The SimCorp Dimension Order Manager Server will not function correctly if SQLNET.INBOUND_CONNECT_TIMEOUT is set in the Sqlnet.ora used by this server. The parameter must either be set to 0 or not specified at all. For more information about this issue please refer to the Known Issues chapter of the System Administrators Manual.

6.4.14.4 DIAG_SIGHANDLER_ENABLED=FALSE

If this entry is not present in the client side Sqlnet.ora SimCorp Dimension sessions fault handling will not work as expected and can seriously affect the troubleshooting process. The issue is described in the System Administrators Manual and in Oracle MOS article: Application Fails with ORA-24550: signal received: Unhandled exception: Code=e0434f4d Flags=1 (Doc ID 779996.1).

7 System Operations

7.1 Backup & Recovery

Preferred Oracle Configuration for SimCorp Dimension does not deliver an out-of-the-box backup solution, since backup strategies are customer dependent.

When creating a backup strategy for the Oracle environment, you need to include backup for the SimCorp Dimension environment. It is imperative to have a consistent backup of both the Oracle instance and SimCorp Dimension application folder structure, taken at the same time. This is due to a tight link (patch version) between the SimCorp Dimension database schema and the SimCorp Dimension program files. It is not possible to start SimCorp Dimension if the restored program files belong to a different patch version to that of the restored database schema.

It is therefore crucial that a synchronised backup scheme for SimCorp Dimension and the database is employed, and tested regularly. An RMAN solution is highly recommended. Database Control/Grid can be used to easily deploy and schedule an Oracle recommended backup procedure.

Otherwise please refer to Oracle documentation and guidelines on backup & recovery.

7.2 Standby Database and Oracle Active Data Guard

SimCorp Dimension supports the use of a physical standby database, using an Oracle Data Guard solution in Maximum Performance mode. And with the Oracle Enterprise Edition Active Data Guard option it also becomes possible to execute reports from SimCorp Dimension on the standby database while in standby mode. Please refer to the ***SimCorp Dimension System Administrator's Manual*** for more information on this module.

If you are implementing an Oracle Data Guard solution (even if you are not planning to use the Active Data Guard option) you are recommended to read the document ***Oracle ADG Preferred installation and configuration for SimCorp Dimension***, available from SimCorp on request.

Please note, in a situation where failover from primary to the standby database becomes required, all SimCorp Dimension sessions must make new database connections. Automatic session failover to the standby database from SimCorp Dimension is not supported.

7.3 Users & Privileges

When creating the database, a number of special purpose Oracle users are created by default. Depending on the method used to create the database, these users may or may not have expired passwords and locked accounts. SimCorp Dimension has no direct requirements to log in as these users, and as such you should ensure that their accounts are only enabled if otherwise required.

For example, if using Oracle Database Configuration Assistant and General Purpose or Transaction Processing template to create the database all users (except for SYS and SYSTEM) will be created with account status EXPIRED & LOCKED.

If you are to use the EM repository, the SYSMAN and MGMT_VIEW accounts must be opened, and the DBSNMP opened for the Agent.

It is recommended that you manually REVOKE EXECUTE from PUBLIC on the following objects:

- UTL_FILE
- UTL_HTTP
- UTL_SMTP
- UTL_TCP
- DBMS_JOB
- DBMS_LOB
- DBMS_RANDOM

But do be aware that these privileges can be required at a later time, e.g. when setting up or de-installing Database Control/Grid and the underlying repository.

Please note, if you revoke UTL_FILE from PUBLIC you must ensure to:

- Grant execute on UTL_FILE to XDB and do a recompile of invalidated objects.
- If you have installed Workspace Manager, grant execute on UTL_FILE to the WMSYS user and do a recompile of invalidated objects.

In addition if you REVOKE EXECUTE on DBMS_JOB, DBMS_LOB and DBMS_RANDOM from PUBLIC, ensure to:

```
Grant execute on DBMS_JOB to DBSNMP;
Grant execute on DBMS_LOB to WMSYS and XDB;
```

There may very well be other Oracle functionality schemas relying on the permissions which by default are granted to for PUBLIC. Please keep this in mind and consult the Oracle documentation or My Oracle Support for further information.

SimCorp Dimension uses Oracle static data dictionary USER_* and ALL_* views. Oracle implements access to these views through PUBLIC grants, and as such SimCorp Dimension does not support these being revoked from PUBLIC. The concept of the static data dictionary views are explained here: http://docs.oracle.com/database/121/REFRN/GUID-10024282-6729-4C66-8679-FD653C9C7DE7.htm#REFRN-GUID-10024282-6729-4C66-8679-FD653C9C7DE7.

In the Oracle database by default, PUBLIC has the INHERIT PRIVILEGE on all users. Please note removing the privilege from PUBLIC on any of the SimCorp Dimension schemas is not supported.

Please refer to ***SimCorp Dimension System Administrator's Manual*** for information about SimCorp Dimension users, roles, privileges, connection methods etc.

7.4 Oracle Security Issues

It is highly recommended that you keep yourself up-to-date with critical security issues as they emerge. For more information, check http://www.oracle.com/technology/deploy/security/alerts.htm at regular intervals.

7.5 Table Compression

With the Enterprise Edition Oracle Advanced Compression option license it is possible to use the Oracle OLTP Compression feature.

Studies at SimCorp have beside the reduced storage requirements, shown a positive impact on query times against compressed tables. However the same study also shows that DML statements are significantly slower for compressed tables compared with non-compressed.

The study consisted of investigating, for a group of selected tables, the impact of compression on:

- the size of the object
- time to create table as select (CTAS)
- time to insert as select (IAS)
- time for gathering statistics on the object
- time for simple select from the object (count(*), min, max and sum)
- time to insert, update and delete from the object

These tests showed that compression reduced the storage requirement for the objects by between 60-80 %. The maintenance operations of creating copies of the objects (typically applied during SimCorp Dimension upgrade or data extract) showed performance penalty between 40-50% in the runtime for the operation when the table was compressed, whereas collecting statistics on the compressed objects was 20-30% faster.

Simple selects against the compressed tables were 2 to 2½ times faster compared with non-compressed case.

Inserts on the compressed tables required between 300-500% and updates between 100-250% more time than on their non-compressed counterparts, while the runtime penalty for the deletes were more moderate at between 0-15%.

The conclusion from the study is that table compression is not recommended on the complete set of SimCorp Dimension tables. Instead it is recommended, if you want to use the compression feature, to single out the tables where the benefits from faster queries weigh out the penalty of slower DML and place those tables in a storage customised tablespace. Tables suitable for compression are therefore those with high amounts of queries but relatively low amount of inserts or updates per transaction.

Tables definitely unsuited for table compression are the result tables, where typically very large amounts of insert or deletes are done in the same transaction, e.g. : PERFAGGR, PERFREP, LIMUTILCOMPLSUB, CALCFIGUREDETAILS. Also please note, in Oracle11g[1] , that neither are tables with more than 255 columns (e.g. TRANSMAIN, SECURITIES, HOLDINGS) suitable for compression, as table compression does not work for tables with columns greater than 255 (please refer to My Oracle Support article: Table Compression Does Not Work If Total Number Of Columns Is Greater Than 255 [ID 1464662.1]).

When you have singled out the tables you would like to compress, you will need a compressed tablespace and then use the SimCorp Dimension Table Storage Customisation task to transfer the tables from the default data owner tablespace to the compressed tablespace.

The below steps take you through the procedure:

1. Create the compressed tablespace:

   ```
   Create tablespace SCDATCOMP datafile
   'E:\oracle\oradata\scnew\SCDATCOMP01.dbf' size 2048M
   autoextend on next 500M DEFAULT ROW STORE COMPRESS
   ADVANCED;
   ```

2. Grant quota to the SimCorp Dimension data owner:

   ```
   Alter user SCDAT quota unlimited on SCDATCOMP;
   ```

3. Log on SimCorp Dimension and open the Table Storage Customisation task:

   In the Table field select the table you want to place in the compressed tablespace. In the Table Object field choose Table and in the Tablespace field choose the compressed tablespace.

4. Save the Table Storage Customisation changes:

   When you save the new entries, SimCorp Dimension will ensure the table is safely transferred to the compressed tablespace, and will hereafter be maintained by SimCorp Dimension in that tablespace. This operation should be done when no other users are logged on.

Note, if you move a table directly in the database without the use of the SimCorp Dimension Table Storage Customisation task, SimCorp Dimension will at the next patch, upgrade or Verify Database Structure task move the table back to the data owner default tablespace.

Note, the initial relocation of the table to the compressed tablespace will require space to hold a full copy of the table while being moved. Once the table is altered the additional space will be free to be used by other objects

[1] For Oracle12c, refer to My Oracle Support article: Advanced Compression Restriction on 255 Columns Lifted For Oracle 12c [Doc ID 1612095.1]

in the tablespace. If several tables are to be compressed it may be helpful to relocate the tables singly.

7.6 Oracle Unified Audit

SimCorp Dimension (from version 5.9) supports the new Unified Audit in Oracle12c. This means that the database audit reports available from Security Options in SimCorp Dimension will use UNIFIED_AUDIT_TRAIL, provided the database has been migrated to Unified Audit. If the database is not yet migrated or is running “mixed mode” the reports will be made, as in earlier versions, based on DBA_AUDIT_SESSION.

If the following query returns TRUE the database has been migrated to use Unified Audit. If it returns FALSE the database use traditional audit or runs in mixed audit mode.

```
Select value from V$OPTION where parameter = 'Unified
Auditing';
```

The SimCorp Dimension Service Manager can display the auditing session id (AUDSID) for a chosen session. You should here beware, that in mixed mode auditing, the UNIFIED_AUDIT_SESSIONID value in the USERENV namespace is different from the value that is recorded by the SESSIONID parameter. Hence, if you are using mixed mode auditing and want to find the correct audit session ID, you should use the USERENV UNIFIED_AUDIT_SESSIONID parameter, not the SESSIONID parameter. In pure unified auditing, the SESSIONID and UNIFIED_AUDIT_SESSIONID values are the same.

Before migrating the database to use Unified Audit trail, you are strongly encouraged to read the following article from My Oracle Support (MOS): Performance Issues While Monitoring the Unified Audit Trail of an Oracle12c Database (Doc ID 2063340.1). As this article implies, bad performance when querying UNIFIED_AUDIT_TRAIL may be decreased by cleaning up the audit trail (reducing the number of audit records). This also apply, if you experience longer response times when running the audit reports from Security Options in SimCorp Dimension after having migrated the database to use unified audit. For Oracle 12.2 databases please refer to the Oracle Upgrade and Patching on page 27 chapter in this manual, and the post upgrade tasks of transferring database audit records into the new 12.2 format for greatly enhanced query performance on audit data.

In general SimCorp recommends that Oracle's best practices and guidelines for auditing is followed, including creation of audit policies and that unified audit records are archived and purged at regular intervals. See the Oracle Database Security Guide chapter 21 (https://docs.oracle.com/database/121/DBSEG/auditing.htm#DBSEG343) and chapter 23 (https://docs.oracle.com/database/121/DBSEG/audit_admin.htm#DBSEG1026).

7.7 Oracle Database Vault

The SimCorp Dimension application schemas can from version 5.9 be protected in Oracle Database Vault. For a detailed guideline on how to configure this without affecting SimCorp Dimension functionality, please requisite the following technical paper: ***Oracle Database Vault for use with SimCorp Dimension***.

7.8 Oracle System Generated Files

You are strongly encouraged to implement good house-keeping routines to manage Oracle generated logs and trace files, preventing them from growing to unmanageable sizes or otherwise occupy storage for no practical purpose. See for example My Oracle Support article ***Automatic Diagnostic Cleanup - Auto purge (Doc ID 1196437.1)*** for information about configuring a purge policy for dump and incident files.

The **alert.log** and **listener.log** require you to manually rename, delete, or purge them in order to prevent from growing indefinitely. Automatic clean-up routines can be build using scripts or 3rd party tools. For example, on Linux/Unix platforms **logrotate** can be used to rename the **alert.log** at regular intervals, keeping a history of it in small, manageable, and numbered files. A technical paper ***Oracle Cleanup Procedures for UNIX/Linux*** showing examples of this is available from SimCorp.

7.9 Data Encryption

SimCorp Dimension is supported using both Oracle SQLNet encryption and Transparent Data Encryption (TDE).

The SQLNet Encryption feature encrypts plain text data while transported over network, so an unauthorized party cannot read the data whilst in transport. The feature provides protection against data modification and reply attacks.

The TDE feature prevents the possibility of bypassing the database layer and reading sensitive information directly from storage. This protects application data, also in backups and exports.

A technical paper: ***Encrypting Oracle Data for SimCorp Dimension***, is available from SimCorp, providing an overview of how these Oracle features can be used with SimCorp Dimension.

8 Appendix A: Preferred Oracle Compliance Template

The following table can be used as a quick template to cross check compliance of the actual database with the Preferred Oracle Configuration for SimCorp Dimension. Use for example the Compliance field to note if the specified configuration point is:

- Compliant (it meets or exceeds recommendation)
- Non-compliant but accepted for a given reason
- Non-compliant and should be changed

The bracketed numbers refer to the chapters and sections in this manual containing the detailed description of the requirement.

Preferred configuration	Compliance
Tablespaces:	
All tablespaces are Locally Managed	
TEMP is created as a TEMPFILE	
SYSAUX and other utility tablespaces (e.g. USERS) employ Automatic Segment Space Management (ASSM)	
If SimCorp Dimension tablespaces use uniform size it must be minimum 128K if db_ block_size is 8K (otherwise minimum 14*db_block_size)	
Data Files	
Data files AUTOEXTEND by a reasonable volume (100s MB, not KB)	
Redo Log Files and Archive Log Files:	
Redo log file size should be set to at least 500 MB	
Should have at least 4 redo log groups	
Aim for a log switch every half-hour during normal workload	
Redo logs stored on dedicated fast disks (e.g. RAID1 or RAID1+0 and not on RAID requiring parity calculation)	
The database is in ARCHIVELOGMODE, and archived logs on separate disks (or disks where the additional load will not have a negative impact on system performance)	
Database Character Set:	
AL32UTF8 or WE8MSWIN1252 if SimCorp Dimension is version 6.2 or less. AL32UTF8 if SimCorp Dimension is version 6.3 or higher	
Memory Sizing:	

Preferred configuration	**Compliance**
MEMORY_TARGET (or the sum SGA_TARGET + PGA_AGGREGATE_TARGET) should max be 80% of the physical memory	
MEMORY_TARGET (or the sum SGA_TARGET + PGA_AGGREGATE_TARGET) minimum: • 20G for small installations (less than 100 sessions) • 50G for medium sized installations (typically 100 to 300 sessions) • 60+G for large installations (typically more than 300 sessions)	
DB_CACHE_SIZE min 50% of SGA size	
SHARED_POOL_SIZE min 4G	
If RAC above mentioned minimum memory configurations are per instance	
Oracle 12c Multitenant Container and Pluggable Databases:	
SimCorp Dimension is not supported on the Oracle 12.1 multitenant architecture. For Oracle 12.2 the database can be a pluggable database.	
Configuring Oracle RAC for SimCorp Dimension:	
High speed interconnect between nodes	
Use ASSM tablespaces	
Must use ASM	
LOG_ARCHIVE_FORMAT must contain thread#	
PARALLEL_EXECUTION_MESSAGE_SIZE must be at least 16K	
ASM :	
Increase the default memory for the ASM instance size if the database controlled by ASM will be greater than 5TB	
Combining RAID and ASM is not supported if using a dedicated disk system, use either: • ASM only (as recommend by Oracle) • RAID 1 + ASM external redundancy • RAID 5 + ASM external redundancy (however not for redo, please see 2.1.5)	
If the disk system is SAN, use ASM external redundancy and let the SAN handle the RAID	
Disks in a disk group must have the same size and performance characteristics (disk response time)	
Preferred Oracle Initialization Parameters:	
audit_trail=DB	

Preferred configuration	Compliance
control_file_record_keep_time minimum 90	
db_block_size=8192	
log_buffer minimum 16MB. Leave the parameter unset if the default value calculates to something greater than 16MB.	
open_cursors=500	
recyclebin=OFF	
_b_tree_bitmap_plans=FALSE	
If _optim_peek_user_binds=FALSE then optimizer_adaptive_reporting_only=TRUE If _optim_peek_user_binds=TRUE then optimizer_adaptive_reporting_only=FALSE	
_cursor_obsolete_threshold=1024	
job_queue_processes maximum 2xCPU_COUNT	
processes at least the number of background processes + number of concurrent SimCorp Dimension database connections (or PARALLEL_MAX_SERVERS + JOB_QUEUE_PROCESSES whichever is largest).	
nls_length_semantics=CHAR if database character set is Unicode (AL32UTF8) and if SimCorp Dimension is version 6.1 or less.	
temp_undo_enabled=TRUE	
Database Automatic Jobs:	
Oracle default maintenance resource plan has been activated	
Scheduling of the automatic Oracle maintenance tasks does not conflict with database resource intensive SimCorp Dimension processing (e.g. heavy batch and reporting loads)	
Database Optimizer Statistics (configuration):	
The preferences have been set on global level	
DEGREE is changed to AUTO_DEGREE (or appropriate value > 1)	
METHOD_OPT is changed to FOR ALL INDEXES COLUMNS SIZE AUTO	
If partitioning module is installed in SimCorp Dimension CONCURRENT is ALL and the Resource Manager is enabled.	
If the preferences were changed to the preferred values subsequent to SimCorp Dimension operations in the database, statistics were deleted and re-gathered following the change of the preferences in accordance with the guide lines in section Re-establish Statistics after Changes to Statistics Preferences	
Other Parameters for Special Purposes:	

Preferred configuration	**Compliance**
If Oracle licensed with Diagnostic Pack option, configure the AWR snapshot settings: interval at least every 60 minutes and retention period minimum 31 days, preferably 180 days	
Set an appropriate value for cpu_count If the database host is shared amongst several databases	
Flashback database is not enabled in production during daily operations	
If the OS is configured and supports direct I/O set Oracle memory allocation accordingly and ensure Oracle init parameter filesystemio_options=SETALL	
If the SimCorp Dimension Active Directory Authentication module is used (i.e. Kerberos authentication is used for logon in SimCorp Dimension) os_authent_prefix must be set to a blank character	
Set SQLNET.EXPIRE_TIME=10 in the database server side Sqlnet.ora.	

9 Appendix B: Known Issues

This appendix contains reference to Oracle known issues which have been seen to affect SimCorp Dimension, and other Oracle articles with contents of interest. Please also refer to the SimCorp Dimension System Administrators Manual which contains several Oracle known issues related to SimCorp Dimension operation.

Please refer to the My Oracle Support for the documents:

- ORA-03135 When Connecting to the Database (Doc ID 404724.1)
- SQLPlus 12c Memory usage Grows (Leaks) While Running Long Query (Doc ID 1919000.1)

The following article (https://mikedietrichde.com/2017/09/20/oracle-12-2-adaptive-features-enabled-windows-12-1-0-2-bps/) by Oracle Master Product Manager Mike Dietrich may have interest if you have experienced runtime issues caused by misleading Oracle directives. SimCorp are in our verification of Oracle 12.2 aiming at running 12.2 with the default values for parameters optimizer_adaptive_plans and optimizer_adaptive_ statistics. As the article here explains, these parameters can now also be implemented on 12.1 database. If you have experiences from running SimCorp Dimension on the 12.1 database with these new parameters implemented, SimCorp will be very interested in your feedback.

ABOUT SIMCORP

SimCorp provides integrated, best-in-class investment management solutions to the world's leading asset managers, fund managers, asset servicers, pension and insurance funds, wealth managers and sovereign wealth funds. Whether deployed on premise or as an ASP solution, its core system, SimCorp Dimension, supports the entire investment value chain and range of instruments, all based on a market-leading IBOR. SimCorp invests more than 20% of its annual revenue in R&D, helping clients develop their business and stay ahead of ever-changing industry demands. Listed on NASDAQ Copenhagen, SimCorp is a global company, regionally covering all of Europe, North America, and Asia Pacific.

For more information, please visit **www.simcorp.com.**

ONE SYSTEM FOR A COMPLEX WORLD

LEGAL NOTICE

The contents of this publication are for general information and illustrative purposes only and are used at the reader's own risk. SimCorp uses all reasonable endeavors to ensure the accuracy of the information. However, SimCorp does not guarantee or warrant the accuracy, completeness, factual correctness, or reliability of any information in this publication and does not accept liability for errors, omissions, inaccuracies, or typographical errors. The views and opinions expressed in this publication are not necessarily those of SimCorp. © 2014 SimCorp A/S. All rights reserved. Without limiting rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form, by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose without the express written permission of SimCorp A/S. SimCorp, the SimCorp logo, SimCorp Dimension, and SimCorp Services are either registered trademarks or trademarks of SimCorp A/S in Denmark and/or other countries. Refer to www.simcorp. com/trademarks for a full list of SimCorp A/S trademarks. Other trademarks referred to in this document are the property of their respective owners.