July, 2016

Omega Core Audit ™

For Oracle Database

OMEGA Core Audit

For Oracle Database

Deployment Guide

2.8.1

www.dataplus-al.com

TABLE OF CONTENTS

1 CHAPTER 1: Omega Core Audit Overview

1.1 Introducing Omega Core Audit

Omega Core Audit provides an out-of-box, software-only security and compliance solution that helps customers approach the complex and difficult security challenges in the Oracle database systems today; protecting against outsider and/or insider threats, unauthorized access and informational breaches or manipulation; this by enforcing strong security controls and duty separation, in meeting regulatory compliance requirements, those being external or internal.

Omega Core Audit implements strong practices of Access Control, Audit Monitoring and Real-Time Protection, providing clear visibility and control into database activity, even for privileged accounts and more, the DBAs, thus leading to a safer and more secure information system.

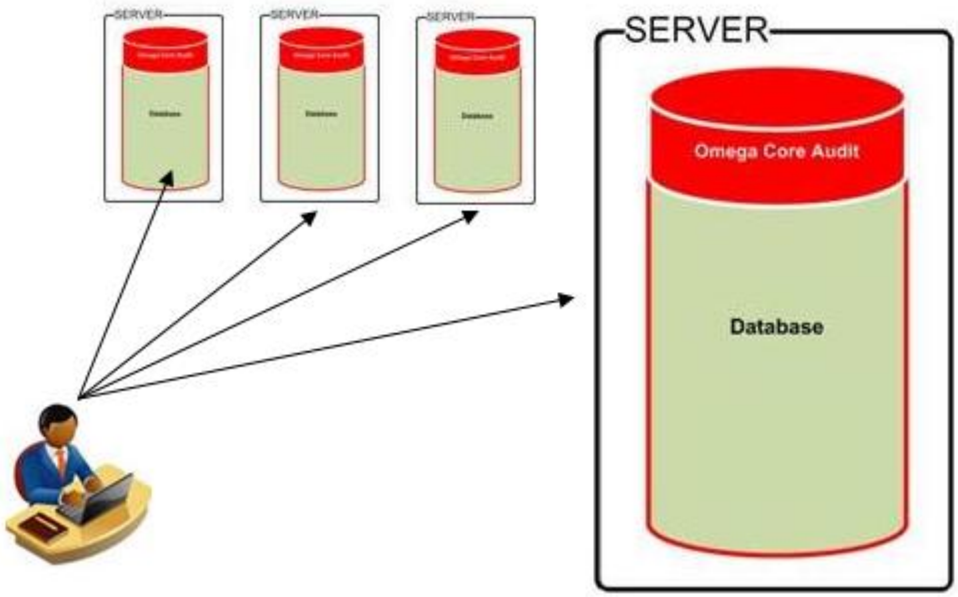
Omega Core Audit is a full back-end solution that is installed in minutes and easily managed by its applicative interface. It enhances the Oracle native security features with state-of-art and value-added programming and automation. It brings easiness to its users letting them focus only on the conceptual security tasks, without concentrating on complex technical security configurations, made easy and plainly presented to them via its rich user interface.

Security applied at the core - from within the database - ensures same rigid level of compliance from all possible connection directions, applications, users or devices and offers maximum accuracy and immediate auditing and protection action before user’s actions or transactions. It also requires no (or very minimal, industry recommended) changes in existing security configurations.

1.2 Omega Core Audit Architecture

The Omega Core Audit solution has three main components:

- **Omega Core Audit Engine**: An Oracle PL/SQL software package containing core audit and protection logic, Back-End installed into the target database under the SYS schema and running with its privileges.
- **Omega Core Audit Repository**: An Oracle Schema Repository containing all system data, Back-End installed into the target database.
- **Omega Core Audit Application**: A Windows-based client desktop Application, connecting to the target database and interacting with the Engine and Repository.



1.2.1 Omega Core Audit Engine

The Omega Core Auditing Engine is an Oracle PL/SQL software package installed under the SYS schema. The Engine contains the core logic of the access control, auditing and protection.

The Core Audit Engine Objects are:

OMEGA_CORE_AUDIT	Oracle Database PL/SQL Package containing core logic
OMEGACA_ACC_DB_AF_LOGON	Oracle Database Trigger on After Logon event
OMEGACA_RTP_DB_BF_DDL	Oracle Database Trigger on Before DDL event
OMEGACA_TRANS	Oracle Database scheduler job for audit trails purge

1.2.2 Omega Core Audit Repository

The Omega Core Auditing Repository is an Oracle Schema named OMEGACA, containing all objects for audit trail data and all configuration options needed by the Engine. It is installed from the back-end setup script.

The Core Audit Repository Objects are:

OMEGACA_TS	An Oracle tablespace storage object, see the Note below.
OMEGACA	An Oracle database schema, containing repository data and configurations.

1.2.3 Omega Core Audit Application

The Omega Core Audit Application is a typical Windows-based and database-enabled client desktop application that connects to each target database. It is used to configure the system for all its operations and also monitor the audited activity generated by the prior.

1.3 Compatibility and Requirements

The Omega Core Audit solution is compatible with and has the following technical requirements.

1.3.1 Supported Oracle Database Versions and Releases

Omega Core Audit Engine and Repository support the following Oracle Database Versions and Releases:

- Oracle Database 10g Release 2.
- Oracle Database 11g Release 1.
- Oracle Database 11g Release 2.
- Oracle Database 12c Release 1 (Traditional Auditing only).

1.3.2 Supported Oracle Database Editions

Omega Core Audit Engine and Repository supports the following Oracle Database Editions:

- Oracle Database EE - Enterprise Edition.
- Oracle Database SE - Standard Edition.
- Oracle Database SE1 - Standard Edition One.

Omega Core Audit features availability by Oracle Database Editions:

Omega CA Features	Oracle Database Edition
Access Control	EE, SE, SE1
Standard Audit	EE, SE, SE1
Real-Time Protection DDL	EE, SE, SE1
Real-Time Protection DML	EE, ---, ----
Security Management	EE, SE, SE1

1.3.3 Oracle Core Audit Application requirements

Omega Core Audit Application supports all the Oracle Database Versions and releases as those supported by the Engine and Repository.

The OS, hardware and software requirements of Omega Core Audit Application are:

- All x86/x64 versions of Windows from XP and above supported by the Oracle 32bit database clients.
- All Oracle Database Clients from 10g R2 to 11gR2.
- Only 32 bits clients Oracle Clients are supported, even on 64bit Windows systems.

1.4 Limitations

Omega Core Audit has currently the following limitations:

1. Only databases opened in Read-Write mode are supported.
2. Unicode character sets are currently not supported. Current language characters support is for Western European Character sets only, however, even in databases with National Character Set of Unicode functionality is achieved almost intact, given that Database infrastructure names of users, objects, columns, ..., etc, (and database language) are set to Western European Languages.
   You must test in your own system to be sure on the compatibility!
3. Connectivity from Omega Core Audit Client Application is currently supported only on 32 Bit Oracle Clients.
4. The Real-Time Protection DML module is functional only on Enterprise Editions - this is a vendor limitation.

Check our website for news on current developments.

Note:
Omega Core Audit operation is unavoidably dependent on Oracle database limitations and bugs/issues, although in the later extensive effort is done in programming to proper handle and circumvent. You should be aware of such limitations especially on topics such Database Triggers, Audit and Fine-Grained Audit.

1.5 Engine and Repository Deployment technical details

The method for installing, upgrading and uninstalling the Omega Core Audit Engine and Repository is simply the execution of Oracle Scripts while connected on a SQL> terminal into the target Oracle database.

The script files provided in your downloaded software package are:

Omega_CA_[VS]_[MN]_[PT]_Install.sql	Install of Omega CA Engine and Repository
Omega_CA_[VS]_[MN]_[PT]_Upgrade.sql	Upgrade of Omega CA Engine and Repository
Omega_CA_[VS]_[MN]_[PT]_UnInstall.sql	Uninstall of Omega CA Engine and Repository

The acronyms are two digit numbers, 0 left padded and stand for:

VS	- Version
MN	- Maintenance Release
PT	- Patch Number

The Install, Upgrade and Uninstall scripts must always be executed with SYS/SYSDBA privileges!

To open a SQL command line connection to the target database on which you are installing, upgrading or uninstalling the Omega Core Audit, in your operating system command line, use one of the following the following:

1. sqlplus SYS@oratnsname as SYSDBA

where oratnsname stands for your Oracle network connection. You will be prompted for SYS password and logon and if the right password given the SQL> command line will be available.

2. sqlplus SYS / as SYSDBA
for local OS connection.

In both cases verify that you are being logged on with SYS/SYSDBA privileges by executing the following in the SQL> command line:

```
SQL>show user;
```

The output must be:

```
USER is "SYS"
```

To execute the provided script, first logon in a SQL> command line as described above, type the @ (at sign) and then the full path (including filename) to the script file. You can of course drag and drop the script file to the SQL terminal if a graphical interface allows that. It should look, for example, like:

```
*NIX:
SQL>@/dev100/omegaca/install/Omega_CA_[VS]_[MN]_[PT]_Install.sql
Win:
SQL>@c:\omegaca\install\Omega_CA_[VS]_[MN]_[PT]_Install.sql
```

Press Enter to execute the script. Scripts execution is automatic and requires no user intervention. All provided scripts will spool in respective ".log" files, named the same way as the ".sql" files.

For more information on how to execute Oracle SQL scripts, see the Oracle 11g R2 Book SQL*Plus® User's Guide and Reference, Chapter 5 Using Scripts in SQL*Plus.

2 CHAPTER 2: Installing Omega Core Audit

The Omega Core Audit deployment mostly consists in installing the Back-End part (Engine and Repository) into the target database. The Front-End part is a simple deployment of the windows desktop Application binaries and (one-time only) of the Oracle Client software for connectivity to the target database.

2.1 Omega Core Audit Engine and Repository Install

2.1.1 Engine and Repository Install Prerequisites

1. Required environment settings:

The following are minimal (and mostly security related industry recommended) changes that must be done, if not already present:

1. Ensure that the Oracle database initialization parameter "audit_trail"=DB_EXTENDED to have the Standard Audit effective. Change of this parameter requires a database restart to be effective!

See Appendix A - Utility Scripts 6.2 Oracle AUDIT_TRAIL parameter.

2. Ensure that the Oracle system triggers are firing. This is controlled by database "hidden" system parameter "_system_trig_enabled" which must always have a value of TRUE (the default) and no change is supposed to be. This parameter controls the firing of system-level triggers and thus the enforcement of the Access Control and Real-Time Protection DDL module policies!

See Appendix A - Utility Scripts 6.4 Oracle System Triggers Firing.

3. To achieve Auditing and Protection at application user level, in case of middle-tier systems that access the database with a single logon account and manage security on their own, modify the application so that it immediately after logon:

- Call the DBMS_SESSION.SET_IDENTIFIER to set the CLIENT_IDENTIFIER value on the middle-tier.
- Sets this value to the application username.

This will populate the CLIENT_ID fields of Standard Audit, Real-Time Protection DDL and DML module audit trails - obviously this field is missing in the Access Control trails. This will also make possible the usage of the Client Identifier factor for evaluating the CLIENT_IDENTIFIER value in the middle-tier (the application user) in Access Control, Real-Time Protection DDL and DML module policies.

For more on this topic see the:
Oracle Database Security Guide book, Chapter 3 Configuring Authentication, "Preserving User Identity in Multitiered Environments", "Using Client Identifiers to Identify Application Users Not Known to the Database" and there see the topic "Using the DBMS_SESSION PL/SQL Package to Set and Clear the Client Identifier".

2. The Omega Core Audit Repository Tablespace

The Omega Core Audit Repository Tablespace (OMEGACA_TS tablespace object) creation is commented in the install script and must be manually executed before installation according to your OS environment and growth needs.

You must execute the OMEGACA_TS tablespace creation separately before running the provided Install script!

The Name of the tablespace OMEGACA_TS is not an option, it must always be OMEGACA_TS and must not be changed! There are many other options you can set in a tablespace create command (except the Name). Consult the database DBA on this topic!

See Appendix A - Utility Scripts 6.1 Omega Core Audit Repository Tablespace.

3. Existing Oracle standard audits and FGA policies

Statement or object audits and/or existing fine-grained audit policies might have been created prior to the install of Omega Core Audit.

It is recommended that you drop such audits.

See Appendix A - Utility Scripts 6.8 Oracle Statement and Objects Audits and 6.9 Oracle FGA Policies.

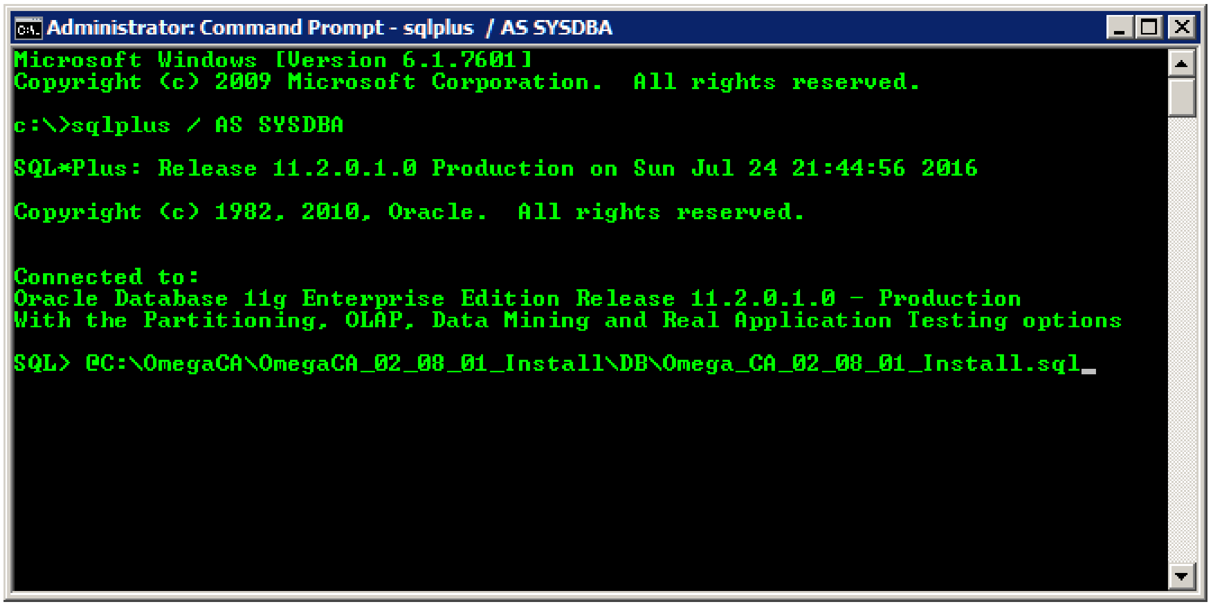
Also records may be present also in SYS.AUD$ and SYS.FGA_LOG$ tables. It is best to archive them and then delete.

2.1.2 Engine and Repository - Install

After you have successfully created the OMEGAC_TS tablespace, you can install the Omega Core Audit Repository and Engine.

To install the Omega Core Audit Engine and Repository, run Install script:

Omega_CA_[VS]_[MN]_[PT]_Install.sql



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\>sqlplus / AS SYSDBA

SQL*Plus: Release 11.2.0.1.0 Production on Sun Jul 24 21:44:56 2016

Copyright (c) 1982, 2010, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @C:\OmegaCA\OmegaCA_02_08_01_Install\DB\Omega_CA_02_08_01_Install.sql_
```

To install, pres Enter at the point shown in the figure above!

Important Note:

This script must be executed with SYS/SYSDBA privileges from a SQL Plus terminal connected to the database!

The installation is performed generally on the following order:

- Check existence of the Omega Core Audit Repository Tablespace
- The Repository Schema Owner OMEGACA is created.
- Dictionary objects (system views) privileges are granted to the Repository Schema Owner.
- The Omega Core Audit Roles are created.
- The Engine's is created with inactivated components.
- Engine privileges granted to the Repository Schema Owner.
- The Repository is created.
- The Repository deployed data are created.
- The Repository deployed reports are created.
- The Engine is compiled.
- The Repository is compiled.
- Repository Privileges are granted to the Omega Core Audit Roles.
- The Administrator user OMEGACAADM is created.
- The Repository Schema Owner OMEGACA is locked.
- The system version info is updated.

Check the content of the Omega_CA_[VS]_[MN]_[PT]_Install.log file after the install. Normally there should be no errors!

Check also for any INVALID Engine or Repository Object[s]. Normally there should be no INVALID objects, or at least must become VALID after one or two compilations.

See Appendix A - Utility Scripts 6.7 Omega Core Audit Objects Status.

Important Note:

For release specific instructions on installing read the Omega_CA_ReadMe_Install.txt document (delivered inside the Omega Core Audit Install software package) and any extra instruction that may appear on the website’s appropriate pages.

2.2 Omega Core Audit Engine and Repository - After the Install

Omega Core Audit will not activate any access control, audit or protection feature after the install!

You have to manually activate the Access Control and Real-Time Protection DDL modules and create policies on them and also create policies into the Standard Audit and Real-Time Protection DML modules. You must also activate the DB Audit Trails Purge Job.

See the System Components form and respective modules policy forms for the above.

2.2.1 Immediate Actions

Immediately after the install:

1. Change the password of the Repository Schema Owner OMEGACA, although account is locked during install.
2. Change the password of the Omega Core Audit Administrator account OMEGACAADM, created during the install. This account is active and unlocked!!!

In order to ensure the proper and optimal operation of the Omega Core Audit, the following minimal and recommended security settings must be applied.

2.2.2 Privilege encapsulation

After the install of the Omega Core Audit Back-End, you are strongly advised to revoke certain security privileges from all users and roles, Omega Core Audit will exercise them exclusively through its SYS side installed Engine with SYS/SYSDBA rights.

Do the Following (logged on as DBA or SYSDBA):

1. ADMINISTER DATABASE TRIGGER System Privilege

Grantees' of the ADMINISTER DATABASE TRIGGER System Privilege do bypass the protective actions of the Access Control module, although they are subject to audit trails generation.

Revoke of this System Privilege ensures enforcement of protective measures of the Access Control module policies! Thus it is important to be revoked from at least all Oracle users and roles (except SYS), DBAs included.

See Appendix A - Utility Scripts 6.3 Oracle ADMINISTER DATABASE TRIGGER System Privilege.

2. Audit System Privileges

It is advised to revoke the Standard Audit System Privileges from all Oracle users and roles (except SYS), DBAs included.

The System Privileges to be revoked are:

AUDIT ANY	Object Auditing System Privilege
AUDIT SYSTEM	Statement Auditing System Privilege

See Appendix A - Utility Scripts 6.5 Oracle AUDIT System Privileges.

3. Fine-Grained Audit Policies Privileges

It is advised to revoke the Fine-Grained Audit Policies (DBMS_FGA) privileges from all Oracle users and roles (except SYS), DBAs included. (Rarely) the privilege needed to be revoked is the EXECUTE on DBMS_FGA.

See Appendix A - Utility Scripts 6.5 Oracle DBMS_FGA Package Privileges.

Important Note:

"Privilege encapsulation" Revokes of the above have not created any invalid object or loss/crash/deviance of any functionality in all our tests performed on this topic!
However, as e measure of precaution, you are advised to first test the above on a non-production (test) system.

Although not recommended, any of the removed privileges can be granted back, when in very rare cases may be needed by features of the application, or the administrative or reporting activities. These important security functions being exercised only through Omega Core Audit Software Packages, further consolidates the appliance of the concept of separation of duty and as such, these controls are recommended to be active.

2.3 Omega Core Audit Application Install

The Omega Core Audit application enables management of multiple Oracle databases, where the Omega Core Audit Engine and repository are installed.

To connect to the target database with your Omega Core Audit Client application, first you need Oracle client connectivity on a Windows PC and then simply deploy the two application's files to any directory of the PC.

2.3.1 Oracle Client Connectivity

To connect to the database with your Omega Core Audit Client application first you need an Oracle Database Client installed. This is done only once and will not change with updates of the application, only very rarely when upgrade of the Oracle database client will be needed.

Important Notes:

1. Omega Core Audit client application's connectivity is supported only on 32bit version of Oracle Clients! This is a system limitation which we are working to resolve in the near future.
However Multiple Oracle Homes (Clients) environment is supported by the Omega Core Audit client application. In this case you can keep the existing Oracle Client 64 bit and use an Oracle Instant Client 32 bit, see below.

2. The application by default will use the default Oracle database client, the one which is defined by system environment "Path" variable. See the OCIDLL_Path initialization parameter to empty (default) in the Oracle_Client section of the OmegaCA.ini

3. You can manually use another Oracle Client Installed, included Instant Clients, by manually setting the initialization parameter OCIDLL_Path in the Oracle_Client section of the OmegaCA.ini initialization file.

4. It is advised to have the Oracle Clients in the same version with the Oracle Database, or when managing multiple databases of different versions, have the recent version of the Oracle Client.

2.3.2 Omega Core Audit Client Application

The two Omega Core Audit Client application files are:

OmegaCA.exe
Application executable contains all the application logic. You may optionally create a shortcut on the desktop pointing to the OmegaCA.exe file.

OmegaCA.ini
Configuration initializations parameters for Oracle database client usage and database connections. This file contains three types of sections.

1. The first section is named Last_Entry and contains one parameter named Last_DB. This is managed by the application only and should not be changed by users when changing the initialization file!

2. The second section is named Oracle_Client and contains one parameter named OCIDLL_Path. When this parameter is empty (the default) the Omega Core Audit client application will use the system environment default Oracle database client. If other Oracle database client connectivity is required, set this value to the full path of the oci.dll file (name of the file included). See the examples below:

Example of default Oracle_Client:

```
[Oracle_Client]
OCIDLL_Path=
```

Example of manual Oracle_Client:

```
[Oracle_Client]
OCIDLL_Path=c:\ora_instantclient_11_2\oci.dll
```

Important Notes:

1. Changing the value of the parameter OCIDLL_Path requires application startup to take effect!
2. When choosing a manual setting for the Oracle Client you must have environment variable NLS_LANG set. If the variable has not been created (for example, when using the Oracle Instant Client as a simple zip extraction no environment variable entry is created for NLS_LANG) you must do this manually in your Windows Environment Variables. Set the NLS_LANG value in the following format:

NLS_LANG = language_territory.charset
For example:
NLS_LANG= AMERICAN_AMERICA.WE8MSWIN1252

For more see the Oracle Documentation on "Choosing a Locale with the NLS_LANG Environment Variable"!

3. The third and (optionally) other sections configure database connections and can be named as per your choice. You can create multiple sections for database connections as per your needs.

Example of a database connection section (with two entries):

```
[ProductionDB]
HOST=192.168.1.10
PORT=1521
SID=ORCL
SERVICE_NAME=

[TestDB]
HOST=192.168.1.111
PORT=1521
SID=ORCL
SERVICE_NAME=ORCL
```

Where:

- HOST - IP Address or Hostname of the database server.
- PORT - Oracle Listener Port.
- SID - Database Instance Name, same as database name for single instance (no RAC). Used by default.
- SERVICE_NAME - Database Service name. Used for RAC compliance. Same as SID when (no RAC). Effective only when SID is left NULL

Important Notes:

1. Use only alphanumeric and underscore for parameter values.
2. Do not change section and parameter names, only their values where allowed.
3. Do not change anything in the first section "Last_Entry".
4. Do not use the names of the first two sections for the database connection sections.
5. Use only unique names for all database connection sections.

3 CHAPTER 3: Upgrading Omega Core Audit

The Omega Core Audit upgrade mostly consists in upgrading the Back-End part, Engine and Repository into the target database. The Front-End part is a simple replace of the Application binary OmegaCA.exe.

3.1 Omega Core Audit Engine and Repository Upgrade

3.1.1 Engine and Repository Upgrade Prerequisites

The following are the Omega Core Audit upgrade prerequisites:

1. Backup your Omega Core Audit environment

Take a backup of the Omega Core Audit Engine and Repository before performing any upgrade!
For details on this topic please refer to the Chapter System Wide Functionalities, Utilities and Guidelines, topic System Backup in the book Omega Core Audit User's Guide.

2. Time-Frame for Upgrade

It is advised that Omega Core Audit upgrade is performed at time-frames of low system activity and not on peak hours.

3. Deactivation of System Components

Before applying of this Upgrade, certain system components may need to be disabled. Top candidates would be the Access Control and real-Time Protection modules as a whole. The DB Purge Job is also one.

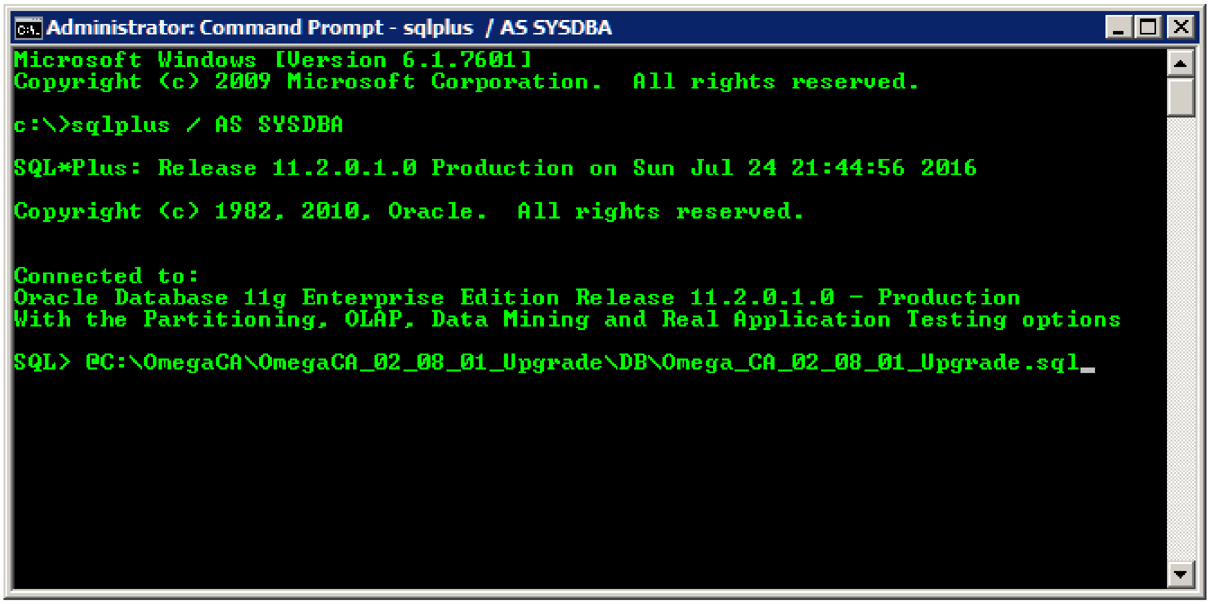
For example, upgrading parts of the system related to the Real-Time Protection DDL module, would certainly call for the disable of this module or otherwise it would lead to SQL errors.

The list of the system components, types of policies, or parameter settings, that need to be disabled before the Upgrade will be release specific as in the "Upgrade Suspend List" section of the Omega_CA_ReadMe_Upgrade.txt document.

3.1.2 Engine and Repository - Upgrade

To upgrade the Omega Core Audit Engine and Repository, run the Upgrade script:

Omega_CA_[VS]_[MN]_[PT]_Upgrade.sql



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\>sqlplus / AS SYSDBA

SQL*Plus: Release 11.2.0.1.0 Production on Sun Jul 24 21:44:56 2016

Copyright (c) 1982, 2010, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @C:\OmegaCA\OmegaCA_02_08_01_Upgrade\DB\Omega_CA_02_08_01_Upgrade.sql_
```

To upgrade, pres Enter at the point shown in the figure above!

This script must be executed with SYS/SYSDBA privileges from any SQL Plus terminal connected to the database!

Check the content of the Omega_CA_[VS]_[MN]_[PT]_Upgrade.log file after the upgrade. Normally there should be no errors!

Check also for any INVALID Engine or Repository Object[s]. Normally there should be no INVALID objects, or at least must become VALID after one or two compilations.

See Appendix A - Utility Scripts 6.7 Omega Core Audit Objects Status.

Important Note:

For release specific instructions on upgrading read the Omega_CA_ReadMe_Upgrade.txt document (delivered inside the Omega Core Audit Upgrade software package) and any extra instruction that may appear on the website’s appropriate Upgrades pages.

3.2 Omega Core Audit Engine and Repository - After the Upgrade

Remember to re-activate modules, policy status and settings that you de-activated before the Upgrade, as they are listed into the release specific "Upgrade Suspend List" section of the Omega_CA_ReadMe_Upgrade.txt document.

3.3 Omega Core Audit Application Upgrade

Upgrade of the Omega Core Audit Application will simply consist in replacing your current desktop-deployed OmegaCA.exe with the new version downloaded into the Upgrade Software Package.

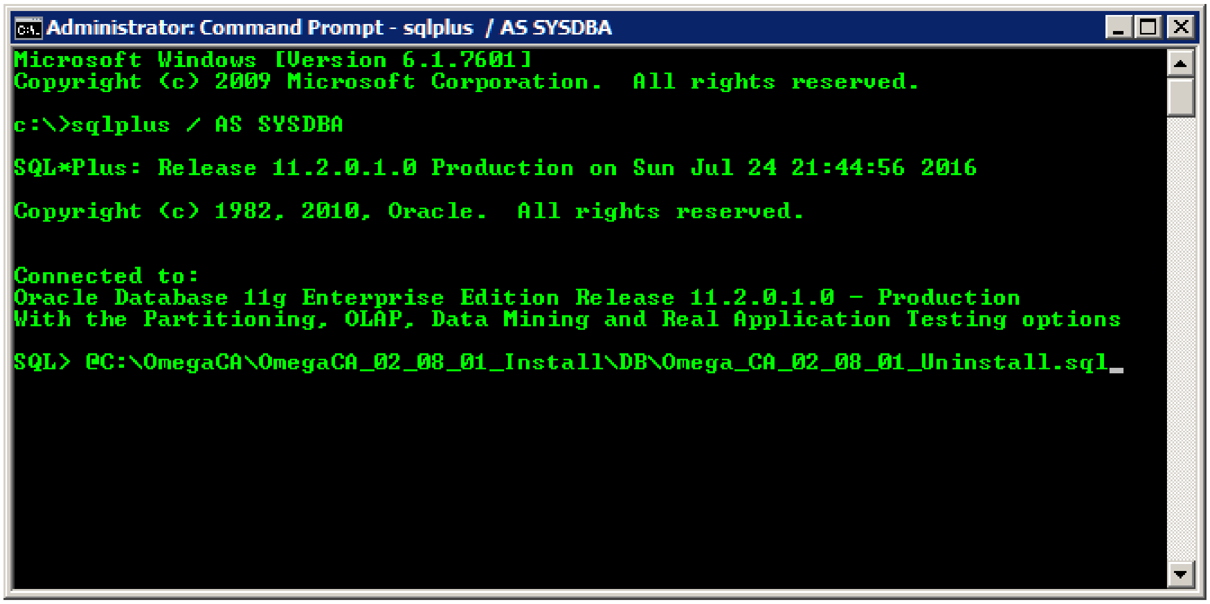
4 CHAPTER 4: Uninstalling Omega Core Audit

The Omega Core Audit un-installation mostly consists in uninstalling the Back-End part, Engine and Repository in the target database. The Front-End part is a simple removal of the Application binaries and (optionally) the Oracle client.

4.1 Omega Core Audit Engine and Repository Uninstall

To uninstall the Omega Core Audit Engine and Repository, run the Uninstall script:

Omega_CA_[VS]_[MN]_[PT]_Uninstall.sql



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\>sqlplus / AS SYSDBA

SQL*Plus: Release 11.2.0.1.0 Production on Sun Jul 24 21:44:56 2016

Copyright (c) 1982, 2010, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @C:\OmegaCA\OmegaCA_02_08_01_Install\DB\Omega_CA_02_08_01_Uninstall.sql_
```

To uninstall, pres Enter at the point shown in the figure above!

This script must be executed with SYS/SYSDBA privileges from any SQL Plus terminal connected to the database!

The un-installation is performed on the following order:

- The Engine's active components are disabled.
- The Engine's components are dropped.
- The Omega Core Audit Roles are dropped.
- The Repository Schema is dropped.

Check the content of the Omega_CA_[VS]_[MN]_[PT]_Uninstall.log file after the install. Normally there should be no errors!
There should be no Omega Core Audit objects left in the SYS schema and also the OMEGACA schema user should have been dropped!

Important Note:

For release specific instructions on uninstalling read the Omega_CA_ReadMe_Deployment.txt file (delivered inside the Omega Core Audit Install software package).

4.2 Omega Core Audit Engine and Repository - After the Uninstall

Uninstall will not:

1. Drop Oracle audit settings for both user statements and objects that may have been created by the Omega Core Audit operations in the Standard Audit module.
2. Drop or disable Oracle fine-grained audit policies that may have been created by the Omega Core Audit operations in the Real-Time Protection DML module.
3. Drop the Tablespace used for the Repository.

To manually remove them, refer to Appendix A - Utility Scripts 6.5 Oracle AUDIT System Privileges, 6.6 Oracle DBMS_FGA Package Privileges and 6.1 Omega Core Audit Repository Tablespace.

4.3 Omega Core Application Uninstall

To uninstall the Omega Core Audit Application, simply delete your PC deployed files OmegaCA.exe and OmegaCA.ini and any shortcut on our desktop.

You might optionally uninstall the Oracle Client if not needed.

5 CHAPTER 5: Support and Licensing

5.1 Support

For technical support please use the support area on our site www.dataplus-al.com/support. Here you can find product documentation, knowledge base, raise support service requests and more.

Browse our site www.dataplus-al.com for updated information.

You can also send an e-mail to support@dataplus-al.com.

Support and upgrades are offered to registered and commercials users only. However, just by registering for free you will have free and unlimited access to latest software packages download, documentations and utilities.

We do commit to offer support to evaluation users too and also arrange remote trainings, demos and implementations on their systems.

DATAPLUS
Tirana, Albania
Street Address: Bul. Zog I, P. "Edicom", 8F.
E-Mail: info@dataplus-al.com
Cel: +355 68 2061664
Tel: +355 42419275

5.2 Licensing

Omega Core Audit license is perpetual, allows the customer to use the licensed software indefinitely and not tied to the product version. However, you are only entitled to new versions and upgrades only if you have a valid maintenance and support contract in place.

Omega Core Auditing for Oracle licensing model is based on the Edition of your Oracle database server, the number and type of the processors and the number of cores per processor, where this late applies, of the machine where the database server resides.

Enterprise Edition Per-core licensing:

If the edition of database monitored is the Enterprise Edition, the licensing model is based on the number and type of processors and the number of cores per processor, where this late applies.
The licensing formula is:

Number of Licenses = (Processors) * (Cores per Processor) * CPLF

Where:

Processors	Number of physical processors in the machine
Cores per Processor	Number of cores per processor (1 for single processors)
CPLF	Core Processor Licensing Factor

The CPLF (Core Processor Licensing Factor) value varies from 0.5 to 1.0 for different processors vendors and models and can be seen on the table at the end of this topic.

If the "Number of Licenses" has a fractional part, it will be rounded up to the next whole number.
This means a minimal number of 1 (one) license is required!

For example, an Enterprise Edition residing on a multi-core system with 2 x 4-core processors would require 2*4*0.5 = 4 licenses (for a CPLF of 0.5).

CPLF (Core Processor Licensing Factor) Table:

Vendor and Processor	CPLF
Sun and Fujitsu SPARC64 VI, VII	0.75
Sun UltraSPARC IV, IV+, or earlier Multicore chips	0.75
Sun UltraSPARC T2	0.75
HP PA-RISC	0.75
IBM POWER5+ or earlier Multicore chips	0.75
Intel Itanium Series 93XX (For servers purchased on or after Dec 1st, 2010)	1.0
Intel Itanium Series 95XX	1.0
IBM POWER6	1.0
IBM POWER7, IBM POWER7+	1.0
IBM POWER8	1.0
IBM System z (z10 and earlier)	1.0
All Other Multicore chips	0.5
All Single Core Chips	1.0

Standard Edition Per-socket licensing:

If the edition of database monitored is the Standard Edition or Standard Edition One, the licensing model is based on the number of processors. A processor is counted equivalent to an occupied socket; however, in the case of multi-chip modules, each chip in the multi-chip module is counted as one occupied socket.
This means that in the formula above (for the Enterprise Edition) the CPLF and Number Cores/Processor are always 1 (one)!

For example, a Standard Edition (One) residing on 2 Processors system you would require 2 licenses.

Note:

Processor technologies of "Soft partitioning" such as VMWare and Microsoft Virtual Server, or others of the same kind, are not recognized in the formula. The pricing is calculated based on the physical processor(s) of the underlying machine hardware.

6 Appendix A - Utility Scripts

6.1 Omega Core Audit Repository Tablespace

Description:
The following commands create and drop the Omega Core Audit Repository Tablespace OMEGACA_TS.

Create OMEGACA_TS TableSpace

A commented template of OMEGACA_TS TableSpace creation SQL command is found in the Install script.

```
-- Create Omega CA Tablespace OMEGACA_TS
CREATE TABLESPACE OMEGACA_TS
LOGGING
DATAFILE '<DATA_FILE>' SIZE <INIT_SIZE>
REUSE AUTOEXTEND ON NEXT <NEXT_SIZE>
MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL;
```

Set the following parameters according to your needs:

- DATA_FILE full path and name of the Datafile, OMEGACA_TS_01 suggested!
- INIT_SIZE Initial Datafile size in M (Megabytes) or G (Gigabytes).
- NEXT_SIZE Next Datafile size increase in M (Megabytes) or G (Gigabytes).

Example of OMEGACA TableSpace creation on a *NIX environment:

```
CREATE TABLESPACE OMEGACA_TS
LOGGING
DATAFILE '/u11/oracle/oradata/omegacoreaudit/omegaca_ts_01.dbf' SIZE 5 G
REUSE AUTOEXTEND ON NEXT 100 M
MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL;
```

The above would be the case of creating the OMEGACA_TS with one single datafile name omegaca_ts_01.dbf, in the folder /u11/oracle/oradata/omegacoreaudit, with an initial size 5 G and increment of 100 M.

Drop OMEGACA_TS TableSpace

Dropping the OMAGACA_TS tablespace:

```
SQL>drop tablespace OMAGACA_TS including contents;
```

or

```
SQL>drop tablespace OMAGACA_TS including contents and datafiles;
```

For more details see the Oracle Book Database SQL Language Reference, DROP TABLESPACE command!

6.2 Oracle AUDIT_TRAIL parameter

Description:
Shows the value of the AUDIT_TRAIL parameter that controls Oracle statement and object audit activation.

To check the value of the AUDIT_TRAIL parameter, run the query:

```
select t.name, t.value
from v$parameter t
where t.name = 'audit_trail'
```

There should be one row returned:

```
NAME          VALUE
audit_trail   DB_EXTENDED
```

To set the value of the AUDIT_TRAIL parameter to DB_EXTENDED, run the command:

```
SQL>alter system set AUDIT_TRAIL=DB_EXTENDED scope=spfile;
```

Because this parameter is a static one, you need to restart the instance for the action to take effect.

6.3 Oracle ADMINISTER DATABASE TRIGGER System Privilege

Description:
Show grantees of the ADMINISTER DATABASE TRIGGER system privilege. The REVOKE_ADM_DB_TRIGGER column shows the respective REVOKE command for each row.

To list Grantees of the ADMINISTER DATABASE TRIGGER system privilege, run the query:

```
select t.grantee, t.privilege, t.admin_option,
'REVOKE ADMINISTER DATABASE TRIGGER from ' || t.grantee || ';'
as REVOKE_ADM_DB_TRIGGER
from dba_sys_privs t
where t.privilege = 'ADMINISTER DATABASE TRIGGER'
and t.grantee <> 'SYS'
```

6.4 Oracle System Triggers Firing

Description:
Shows the System triggers firing status, controlled by the Oracle database “hidden” system parameter “_system_trig_enabled”.

To check for Oracle System Triggers Enabled Firing status, run the query:

```
select t.ksppinm name, x.ksppstvl value
from x$ksppi t,  x$ksppcv x
where t.indx = x.indx
and t.inst_id = userenv('Instance')
and t.ksppinm = '_system_trig_enabled'
and x.inst_id = userenv('Instance')
```

There should be one row returned:

NAME	VALUE
_system_trig_enabled	TRUE

6.5 Oracle AUDIT System Privileges

Description:
Show grantees of the AUDIT-related system privileges. The REVOKE_AUD_PRIV column shows the respective REVOKE command for each row.

To list Grantees of AUDIT-related system privileges, run the query:

```
select t.grantee, t.privilege, t.admin_option,
'REVOKE ' || t.privilege || ' from ' || t.grantee || ';'
as REVOKE_AUD_PRIV
from dba_sys_privs t
where t.privilege in ('AUDIT SYSTEM', 'AUDIT ANY')
and t.grantee <> 'SYS'
```

6.6 Oracle DBMS_FGA Package Privileges

Description:
The following commands show the status of Omega Core Audit Engine and Repository objects.

To check for grantees of the DBMS_FGA Package, run the query:

```
select t.grantee, t.owner, t.table_name, t.privilege,
'REVOKE ' ||  t.privilege || ' on ' || t.table_name || ' from ' || t.grantee || ';'
as REVOKE_FGA_PRIV
from dba_tab_privs t
where t.owner = 'SYS'
and t.table_name = 'DBMS_FGA'
order by t.table_name, t.grantee
```

6.7 Omega Core Audit Objects Status

Description:
The following commands show the status of Omega Core Audit Engine and Repository objects.

To check for status of Engine objects run the query:

```
select t.object_name, t.object_type, t.status
from dba_objects t
where t.owner = 'SYS'
and (t.object_name like 'OMEGACA%' or t.object_name = 'OMEGA_CORE_AUDIT')
```

There should be only VALID status Engine objects!

To check for INVALID Repository objects query:

```
select t.object_name, t.object_type, t.status
from dba_objects t
where t.owner = 'OMEGACA'
and t.status <> 'VALID'
```

There should be no records returned!

6.8 Oracle Statements and Objects Audits

Description:
The following commands list the Oracle statements and objects audits and generate a removal command for each row in the NOAUDIT_CMD column.

ORACLE STATEMENT AUDITS

To list all Oracle user-assigned Statement Audits, run the query:

```
select t.user_name, t.proxy_name, t.audit_option, t.success, t.failure,
'NOAUDIT ' || t.audit_option || ' BY ' || t.user_name || ';' as NOAUDIT_CMD
from dba_stmt_audit_opts t
where t.user_name is not null
order by t.user_name, t.audit_option
```

To list all Oracle user-wide Statement Audits, run the query:

```
select t.user_name, t.proxy_name, t.audit_option, t.success, t.failure,
'NOAUDIT ' || t.audit_option || ';' as NOAUDIT_CMD
from dba_stmt_audit_opts t
where t.user_name is null
order by t.user_name, t.audit_option
```

Warning:
These may be default Oracle audits (statement audits only).

ORACLE OBJECT AUDITS

To list all Oracle Objects Audits, run the query:

```
select A.owner, A.object_name, A.object_type, A.AUDIT_OPTION,
'NOAUDIT '
|| A.AUDIT_OPTION
|| ' ON ' ||
CASE A.object_type
WHEN 'DIRECTORY' THEN 'DIRECTORY ' || A.object_name || ';'
ELSE A.owner || '.' || A.object_name || ';'
END
as NOAUDIT_CMD

from
(
-- Begin Make ALTER
select t.owner, t.object_name, t.object_type, 'ALTER' as AUDIT_OPTION,
CASE Substr(t.ALT,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Success,
CASE Substr(t.ALT,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.ALT <> '-/-'
-- End Make ALTER

UNION ALL

-- Begin Make AUDIT
select t.owner, t.object_name, t.object_type, 'AUDIT' as AUDIT_OPTION,
CASE Substr(t.AUD,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Success,
CASE Substr(t.AUD,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.AUD <> '-/-'
-- End Make AUDIT

UNION ALL

-- Begin Make COMMENT
select t.owner, t.object_name, t.object_type, 'COMMENT' as AUDIT_OPTION,
CASE Substr(t.COM,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Success,
CASE Substr(t.COM,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.COM <> '-/-'
-- End Make COMMENT

UNION ALL

-- Begin Make DELETE
select t.owner, t.object_name, t.object_type, 'DELETE' as AUDIT_OPTION,
Case Substr(t.DEL,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.DEL,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.DEL <> '-/-'
-- End Make DELETE

UNION ALL

-- Begin Make GRANT
select t.owner, t.object_name, t.object_type, 'GRANT' as AUDIT_OPTION,
Case Substr(t.GRA,1,1)
```

```
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.GRA,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.GRA <> '-/-'
-- End Make GRANT

UNION ALL

-- Begin Make INDEX
select t.owner, t.object_name, t.object_type, 'INDEX' as AUDIT_OPTION,
Case Substr(t.IND,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.IND,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.IND <> '-/-'
-- End Make INDEX

UNION ALL

-- Begin Make INSERT
select t.owner, t.object_name, t.object_type, 'INSERT' as AUDIT_OPTION,
Case Substr(t.INS,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.INS,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.INS <> '-/-'
-- End Make INSERT

UNION ALL

-- Begin Make LOCK
select t.owner, t.object_name, t.object_type, 'LOCK' as AUDIT_OPTION,
Case Substr(t.LOC,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.LOC,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.LOC <> '-/-'
-- End Make LOCK

UNION ALL

-- Begin Make RENAME
select t.owner, t.object_name, t.object_type, 'RENAME' as AUDIT_OPTION,
Case Substr(t.REN,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
```

```
Else 'ERROR' end as Success,
CASE Substr(t.REN,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.REN <> '-/-'
-- End Make RENAME

UNION ALL

-- Begin Make SELECT
select t.owner, t.object_name, t.object_type, 'SELECT' as AUDIT_OPTION,
Case Substr(t.SEL,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.SEL,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.SEL <> '-/-'
-- End Make SELECT

UNION ALL

-- Begin Make UPDATE
select t.owner, t.object_name, t.object_type, 'UPDATE' as AUDIT_OPTION,
Case Substr(t.UPD,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.UPD,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.UPD <> '-/-'
-- End Make UPDATE

-- REF,  backward compatibility only !
UNION ALL

-- Begin Make EXECUTE
select t.owner, t.object_name, t.object_type, 'EXECUTE' as AUDIT_OPTION,
Case Substr(t.EXE,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.EXE,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.EXE <> '-/-'
-- End Make EXECUTE

UNION ALL

-- Begin Make CREATE
select t.owner, t.object_name, t.object_type, 'CREATE' as AUDIT_OPTION,
Case Substr(t.CRE,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
```

```
Else 'ERROR' end as Success,
CASE Substr(t.CRE,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.CRE <> '-/-'
-- End Make CREATE

UNION ALL

-- Begin Make READ
select t.owner, t.object_name, t.object_type, 'READ' as AUDIT_OPTION,
Case Substr(t.REA,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.REA,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.REA <> '-/-'
-- End Make READ

UNION ALL

-- Begin Make WRITE
select t.owner, t.object_name, t.object_type, 'WRITE' as AUDIT_OPTION,
Case Substr(t.WRI,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.WRI,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.WRI <> '-/-'
-- End Make WRITE

UNION ALL

-- Begin Make FLASHBACK
select t.owner, t.object_name, t.object_type, 'FLASHBACK' as AUDIT_OPTION,
Case Substr(t.FBK,1,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' end as Success,
CASE Substr(t.FBK,3,1)
WHEN '-' THEN 'NOT SET' WHEN 'A' THEN 'BY ACCESS' WHEN 'S' THEN 'BY SESSION'
Else 'ERROR' END as Failure
from dba_obj_audit_opts t
where t.FBK <> '-/-'
-- End Make FLASHBACK

) A

order by A.owner, A.object_name
```

For more details on both, see the Oracle Book Database SQL Language Reference, NOAUDIT command!

6.9 Oracle FGA Policies

Description:
The following commands list the Oracle FGA policies and generate a disable/removal command for each one respectively into the in the FGA_DISABLE_CMD and FGA_DROP_CMD columns.

To list all Oracle FGA (fine-grained) audit policies, run the query:

```
select t.object_schema, t.object_name, t.policy_name,
'exec DBMS_FGA.DISABLE_POLICY('
|| '''' || t.object_schema || '''' || ', '
|| '''' || t.object_name || ''''   || ', '
|| '''' || t.policy_name || ''''
||  ');' as FGA_DISABLE_CMD,
'exec DBMS_FGA.DROP_POLICY('
|| '''' || t.object_schema || '''' || ', '
|| '''' || t.object_name || ''''   || ', '
|| '''' || t.policy_name || ''''
||  ');' as FGA_DROP_CMD
from dba_audit_policies t
order by t.policy_name
```

For more details on FGA Disable and Drop both, see the Oracle Book Database PL/SQL Packages and Types Reference, DBMS_FGA package!