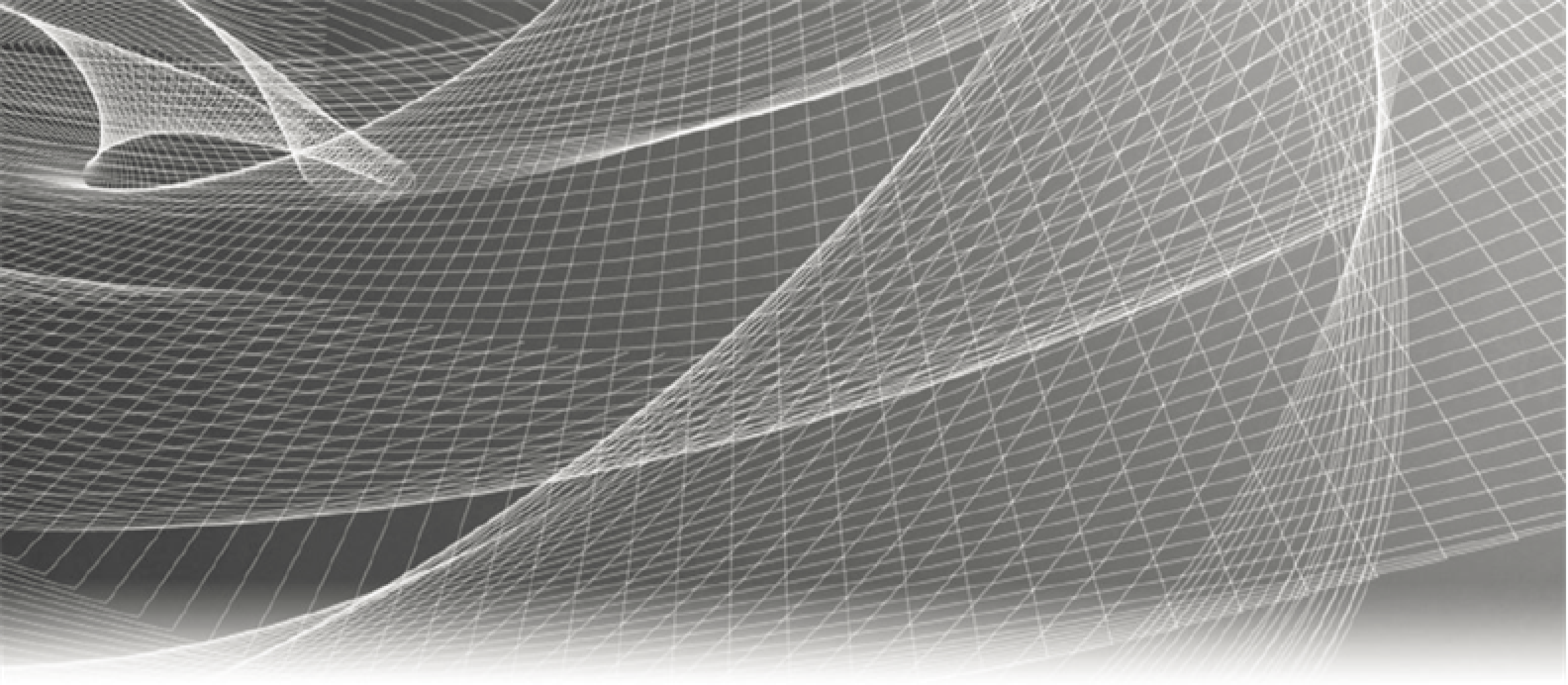

RSA Identity Governance and Lifecycle Database Setup and Management Guide

Version 7.2

Contact Information

RSA Link at https://community.rsa.com contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

License agreement

Third-party licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed by launching the RSA Identity Governance and Lifecycle product and selecting the About menu. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on encryption technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any Dell software described in this publication requires an applicable software license.

Dell Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." DELL INC. MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

February 2020

Contents

Preface

About This Guide

This guide provides instructions on how to set up an Oracle database in your infrastructure (or customer-supplied database) for RSA Identity Governance and Lifecycle. This guide assumes that the reader is authorized and qualified to configure and maintain the Oracle database.

Documentation Set

The latest product documentation is always available at https://community.rsa.com/community/products/governance-and-lifecycle.

Document	Description
Release Notes	What's new in the release, fixed issues, known issues and workarounds.
Installation Guide	Product installation instructions.
Upgrade and Migration Guide	Instructions for upgrading your product version and data.
Database Setup and Management Guide	Instructions for setting up and managing a customer-supplied Oracle database for RSA Identity Governance and Lifecycle.
Configuring WildFly Clusters	Instructions to set up and configure a WildFly application server cluster in an RSA Identity Governance and Lifecycle deployment.
Online Help	All concepts and instructions you need to configure and use the product.
Administrator's Guide	How to configure and manage RSA Identity Governance and Lifecycle. Contains a subset of the information provided in the Online Help.
Public Database Schema Reference	The public view of the database schema.

Support and Service

You can access community and support information on RSA Link at https://community.rsa.com/community/products/governance-and-lifecycle. RSA Link contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

You can also access the RSA Identity Governance and Lifecycle community at https://community.rsa.com/community/products/governance-and-lifecycle/client-partner-community. This private community is only available to RSA Identity Governance and Lifecycle customers, partners, and internal RSA staff.

Chapter 1: Introduction

Overview

Note: The Oracle database you maintain is referred to as the "remote database" in some sections in this document. The term is synonymous with the "customer-supplied database" term used in this guide and in other installation and upgrade documents. Both terms refer exclusively to a database you provide for your RSA Identity Governance and Lifecycle installation.

This chapter describes how to set up a database instance (RAC cluster option included) used by RSA Identity Governance and Lifecycle (application server) in a two-tier installation scenario where you supply and maintain an Oracle database in your hardware infrastructure.

It provides sample script download instructions and the steps required to configure or upgrade the database. The database must be configured before you install RSA Identity Governance and Lifecycle software.

RSA Identity Governance and Lifecycle is designed to use most (but not all) default Oracle database instance configuration settings and installation options. This document does not attempt to document every Oracle configuration setting or installation option for Oracle but instead covers only those settings or installation requirements over and above a default Oracle installation. Any modifications to the Oracle installation or configuration beyond the Oracle defaults and what is noted in this doucment may cause functional or performance issues with RSA Identity Governance and Lifecycle. Please contact RSA Support if you want to change any installation option or configuration beyond what is documented here to ensure a supportable configuration. Review Prepare the Oracle Database Instance for specific details on RSA Identity Governance and Lifecycle requirements.

The SQL commands and configurations documented here should be executed using a database account with sysdba privileges in the database.

This document describes the database objects, tables, and directories that must be created. These include the database user schemas that RSA Identity Governance and Lifecycle uses:

- RSA Identity Governance and Lifecycle user. The default name is AVUSER.
- RSA Identity Governance and Lifecycle reporting engine user. The default name is AVDWUSER.
- RSA Identity Governance and Lifecycle public database schema user. The default name is ACMDB.
- RSA Identity Governance and Lifecycle RSA Identity Governance and Lifecycle Statistics Report user. The data source is AVPERF. (This is required only if Oracle Statspack is installed on the database and you want to include Statspack data in RSA Identity Governance and Lifecycle Statistics Reports.) Failure to configure and use Statspack will limit reporting information for database diagnostics. The default oracle user name is PERFSTAT. See Installing Oracle Statspack to Enhance Database Diagnostics Capabilities for more information.

This document references the user schemas by their default names. You can use non-default user schema names; however, that will require additional configuration when installing RSA Identity Governance and Lifecycle.

Database Parameter Values Worksheet

Print the worksheet below, record parameter values (case-sensitive), and provide the worksheet to the person who installs RSA Identity Governance and Lifecycle. The installer require these values for the installation.

RSA Identity Governance and Lifecycle schema user (AVUSER):
RSA Identity Governance and Lifecycle reporting engine schema user (AVDWUSER):
RSA Identity Governance and Lifecycle public database schema user (ACMDB):
AVUSER password:
AVDWUSER password:
ACMDB password:
AVPERF username/password:
Oracle Database SID:
(RAC) TNSName that references the SCAN name:
Oracle service name:
Oracle listener Hostname:
Oracle Listener Port:

Download Installation Files for Your Database

Before You begin

- You must have a valid license for RSA Identity Governance and Lifecycle 7.2.
- Ensure that your customer-supplied Oracle 12.1.0.2.0 database has all up-to-date Oracle patches installed.

Procedure

1. Log in to RSA Link at https://community.rsa.com/community/products/governance-and-lifecycle.
2. Download the appropriate files for your deployment scenario:
   - Download the following files to the RSA Appliance or software bundle host machine.
     - **instantclient-basiclite-linux-x64-12.1.0.2.0.zip**
     - **instantclient-sqlplus-linux-x64-12.1.0.2.0.zip**
   - For a virtual application deployment, download the following file to the database installation machine: **RSA_IGL_DatabaseOnly.tar.bz2**.

Requirements for Database Deployment

Note: When deploying your database, you must synchronize the database system clock to the system clock on the application server where RSA Identity Governance and Lifecycle is installed.

The following hardware requirements apply to both RAC (remote) and Non-RAC deployment of the RSA Identity Governance and Lifecycle database:

- Cluster instances: At least two database cluster instances for load balancing and high availability. All cluster instances need to be configured to the same minimum or higher memory requirements.

- Memory: Minimum 32GB.
- Disk: Minimum 500GB, 1TB or greater recommended for the Oracle database files.
- Processors: Minimum 2 quad-core CPUs.
- An Oracle database (12cR1 version 12.1.0.2 for RAC or non-RAC implementation) running on a database machine that meets hardware requirements at the deployment site
- A database account with sysdba privileges to be able to perform the database system configuration and validation tasks described in this chapter
- A UNIX root user to create appropriate physical directories and users on the server

Hardware Recommendations

RSA recommends the following specifications for database performance:

- Public Network: Dual-load balanced 1Gb NICs
- Private Network: Minimum 1Gb NIC, 10Gb for production servers, jumbo frames set on switch

Storage Area Network (SAN) Recommendations

- Disks: Minimum of 8 SATA/SAS drives, for a total of 2TB or greater of disk space
- Volume Configuration: RAID 5 for development and production servers, RAID 5 and RAID 0+1 for production servers with higher performance demands
- Logical Unit Number (LUN) configuration: At least three LUNs should be configured and assigned iSCSI initiator names:
  - CRS1: CRS/Cluster Voting Disk, 10GB recommended
  - FS1: Shared file storage space, 200GB recommended
  - VOL1: Main tablespace volume, the remainder of the 2TB.
- Shared File storage space system: ASM Clustered File System (ACFS) share using Oracle ASM Configuration Assistant (asmca) in "ASM Cluster File Systems" tab. Apply the following options:
  - General Purpose File System
  - Register MountPoint, such as **/mnt/acfs-fs1** for the LUN named FS1

  Note: When Oracle CRS starts up, this volume is automatically mounted. No modifications to the /etc/fstab file are required.

Getting Sample Database Configuration Scripts

You can find the database sample configuration scripts located in the DatabaseSamples directory in the aveksa-supplement-*.zip file from the packages directory for this RSA Identity Governance and Lifecycle release. The file includes scripts that include SQL used to set up the database (table spaces, user schemas, and directories on the database server). The files may be generic or contain SQL for a specific version of Oracle.

Scripts:

- aveksa_sample_tablespace_ASM_scripts.sql — Includes sample "create tablespace" instructions for use with ASM.
- aveksa_sample_tablespace_file_scripts.sql — Includes sample "create tablespace" instructions for use with a file system.

- aveksa_sample_ora12_db_scripts.sql — Includes the various commands to create the Aveksa users with the required grants and settings on Oracle 12cR1.
- aveksa_sample_sys_scripts.sql — Includes the required command samples to complete the database configuration.
- aveksa_db_check.sql — Includes the required SQL to verify minimum file size requirements.
- aveksa_db_password_lifetime.sql — Includes a script to determine information about database user password expiration dates.

Note: If you are using an Oracle RAC implementation, ensure that the directories referenced in the scripts point to clustered file systems.

Chapter 2: Set Up the Database

Overview

RSA Identity Governance and Lifecycle requires an Oracle database instance pre-configured with the requirements outlined in this chapter. One requirement is that database users are created with necessary Oracle grant privileges. RSA Identity Governance and Lifecycle creates all of its required database objects such as tables, views, and packages when it is initially started and fully populates the database schema. Those objects may also be modified by upgrades of RSA Identity Governance and Lifecycle software through the RSA Identity Governance and Lifecycle migration process.

Note: The created product schemas, such as AVUSER, are reserved for product database objects. Creating additional database objects within the product schemas may affect the operation of the systems, migration, or patch applications.

Memory, system global area (SGA), and program global area (PGA) are important to having a high-performance system. Environments that collect and process large amounts of data may require higher defined values of SGA and PGA.

Deployment Summary

The deployment process includes the following steps:

1. Prepare the Oracle database instance:
    a. Use a designated instance name.
    b. Use a designated character set.
    c. Ensure a minimum memory configuration for both the SGA and PGA. Memory must meet our required minimums. You can use ASMM to set either SGA/PGA or use AMM (not recommended) to set memory_target and memory_max_target. The settings are mutually exclusive.
    d. Configure the database instance with all required packages.

See Prepare the Oracle Database Instance (If Upgrading from a Previous Product Version) or Prepare the Oracle Database Instance (For a New Product Installation).

2. Create the Aveksa objects:
   a. Create a set of table spaces with the specified naming conventions.
   b. Create a set of database directories.
   c. Map the database directories to specified physical directories.
   d. Create a database password profile.
   e. Configure the user schemas with all required settings.
   f. Create a context for RSA Identity Governance and Lifecycle report filtering values.

      See Create the Aveksa Objects.

3. Complete the required database instance configurations.
   - Disable the nightly job that gathers schema statistics.

   See Complete the Required Database Instance Configuration.

   See Verify Correct Database Configuration.

4. Verify correct database configuration.

   See Verify Correct Database Configuration.

5. Update the database.

   See Updating the Database.

Prepare the Oracle Database Instance (If Upgrading from a Previous Product Version)

Do the following on the database server:

1. Create or identify the instance name that is going to be used.
2. Ensure that the database instance uses the Unicode (AL32UTF8) character set.

   You can validate the character set by running the following SQL:

   ```
   select * from NLS_DATABASE_PARAMETERS where parameter='NLS_
   CHARACTERSET';
   ```

   Output: `NLS_CHARACTERSET AL32UTF8`

   RSA Identity Governance and Lifecycle will fail to start if this character set is not set for the database instance.

   This character set is not the default when configuring Oracle. NLS_LENGTH_SEMANTICS is required to be BYTE, do not change this setting to CHAR.

   Note: The following steps in this section assume that your database is initialized using an spfile and not a pfile. To determine if your database is using a pfile or an spfile, you can run the following commands via SQL*Plus. If the first command returns a value for "ifile," then this value is the name and location of the pfile for your system. If an "spfile" value is found, then this is the name and location of the spfile for your system.

```
show parameter ifile

show parameter spfile
```

One of these commands returns a value.

Convert the a pfile into an spfile if your system is using a pfile. For example:

```
shutdown immediate;

startup pfile=<ORACLE_HOME>/dbs/init<ORACLE_SID>.ora

create spfile='<ORACLE_HOME>/dbs/spfile<ORACLE_SID>.ora' FROM
pfile='<ORACLE_HOME>/dbs/init<ORACLE_SID>.ora';

shutdown immediate;

startup;
```

3. Configure memory management settings for Oracle.

Setting ASMM Values

RSA recommends that memory that can be dedicated to the Oracle instance is allocated in a 2:1 ratio between SGA and PGA. For example, if a host with 128GB of RAM is dedicated to run a single instance of Oracle, allocate 8GB of RAM for the operating system, and set SGA to 80GB, and PGA to 40GB (which would divide the remaining 120GB between SGA and PGA).

a. Determine the memory allocation by running the following SQL:

```
show parameter sga_max_size;

show parameter sga_target;

show parameter pga_aggregate_target;
```

b. Change the parameters with the recommended values by running the following commands:

```
alter system set sga_max_size=<sga_value> scope=spfile

alter system set sga_target=<sga_value> scope=spfile;

alter system set pga_aggregate_target=<pga_value> scope=spfile;

shutdown immediate;

startup;
```

c. Make sure that AMM is not enabled by verifying that the following parameters are not greater than 0:

```
show parameter memory_target

show parameter memory_max_target
```

d. If required, unset the parameter(s) by executing the following command and then restart the database:

```
alter system reset memory_target scope=spfile;

alter system reset memory_max_target scope=spfile;
```

Setting AMM Values

While using AMM has not been tested by RSA and some experts DO NOT recommend using AMM management with Linux HugePages, the following commands will help you set Oracle initialization parameters:

If X is the amount memory that can be dedicated to Oracle instance, RSA recommends that you set Oracle initialization parameters as follows:

Parameter Name	Value
memory_target	X
memory_max_target	X
sga_target	X * 2/6
pga_aggregate_target	X * 1/6

Memory must meet our required minimums. For example, if a host with 128GB of RAM is dedicated to run a single instance of Oracle, allocate 8GB of RAM for the operating system, and set memory_target and memory_max_target to 120GB, SGA minimum value to 40GB, and PGA minimum value to 20GB.

a. Determine the memory allocation by running the following SQL:

```
show parameter memory_target

show parameter memory_max_target

show parameter sga_target

show parameter pga_aggregate_target
```

b. Change the parameters with the recommended values by running the following commands:

```
alter system set memory_target=<memory_value> scope=spfile;

alter system set memory_max_target=<memory_value> scope=spfile;

alter system set sga_target=<sga_value> scope=spfile;

alter system set pga_aggregate_target=<pga_value> scope=spfile;

shutdown immediate;

startup;
```

4. Configure adequate space for all system tablespaces (see Configure Undo, Temp, and Redo Sizes).
5. Make sure that the database instance is configured with the XML_DB package. You can verify that XML DB has been installed by running the following SQL:

```
select comp_name from dba_registry where comp_name like '%XML%';
```

You should see results like the following:

```
COMP_NAME
```

```
Oracle XML Database
```

If this package does not exist, it can be installed with the database configuration assistant (dbca) or manually by executing the instructions found in the Oracle documentation at the following location:

```
http://download.oracle.com/docs/cd/B19306_
01/appdev.102/b14259/appaman.htm#CACIBCBA
```

6. Run the following commands to specify the optimizer settings:

   ```
   alter system set optimizer_index_cost_adj=30 scope=both;
   ```

   ```
   alter system set optimizer_index_caching=50 scope=both;
   ```

   ```
   alter system set optimizer_adaptive_features = false scope=spfile;
   ```

7. The value of the cursor sharing parameter cursor_sharing, which determines which kinds of SQL statement can share the same cursors. Run the following command:

   alter system set cursor_sharing=force scope=spfile;

8. Configure the database to accommodate a minimum of 500 processes by running the following command:

   ```
   alter system set processes=500 scope=spfile;
   ```

   If your database serves multiple application server nodes, multiply the process number by the number of nodes.

   If you set "sessions" in your previous product release, let Oracle use the default by running the following command:

   ```
   alter system reset sessions scope=spfile;
   ```

9. Configure the creation of deferred segments for tables to false, and also unset the db_file_multiblock _ read_count parameter if it was changed. This is required for the Oracle 12c R1 Enterprise version.

   ```
   alter system set DEFERRED_SEGMENT_CREATION = FALSE
   ```

   ```
   alter system reset db_file_multiblock_read_count scope=spfile;
   ```

10. Validate that database requirements are reflected in your base Oracle startup by validating the spfile used by your database instance. Run this command:

    ```
    SELECT NAME, Value FROM gv$parameter order by name
    ```

11. Restart the database server.

Prepare the Oracle Database Instance (For a New Product Installation)

Do the following on the database server:

1. Create or identify the instance name that is going to be used.
2. Ensure that the database instance uses the Unicode (AL32UTF8) character set.

   You can validate the character set by running the following SQL:

   ```
   select * from NLS_DATABASE_PARAMETERS where parameter='NLS_
   CHARACTERSET';
   ```

   Output: `NLS_CHARACTERSET AL32UTF8`

   RSA Identity Governance and Lifecycle will fail to start if this character set is not set for the database instance.

   This character set is not the default when configuring Oracle. NLS_LENGTH_SEMANTICS is required to be BYTE, do not change this setting to CHAR.

   Note: The following steps in this section assume that your database is initialized using an spfile and not a pfile. To determine if your database is using a pfile or an spfile, you can run the following commands via SQL*Plus. If the first command returns a value for "ifile", then the this value is the name and location of the pfile for your system. If an "spfile" value is found, then this is the name and location of the spfile for your system.

   ```
   show parameter ifile

   show parameter spfile
   ```

   One of these commands returns a value.

   Convert the a pfile into an spfile if your system is using a pfile. For example:

   ```
   shutdown immediate;

   startup pfile=<ORACLE_HOME>/dbs/init<ORACLE_SID>.ora

   create spfile='<ORACLE_HOME>/dbs/spfile<ORACLE_SID>.ora' FROM
   pfile='<ORACLE_HOME>/dbs/init<ORACLE_SID>.ora';

   shutdown immediate;

   startup;
   ```

3. Configure memory management settings for Oracle.

   ### Setting ASMM Values

   RSA recommends that the memory that can be dedicated to Oracle instance is allocated in a 2:1 ratio

between SGA and PGA. For example, if a host with 128GB of RAM is dedicated to run a single instance of Oracle, allocate 8GB of RAM for the operating system, and set SGA to 80GB, and PGA to 40GB (which would divide the remaining 120GB between SGA and PGA).
Memory must meet our required minimums. On a RAC system with multiple oracle instances, every instance must meet the same SGA/PGA calculated values.

a. Determine the memory allocation by running the following SQL:

```
show parameter sga_max_size;

show parameter sga_target;

show parameter pga_aggregate_target;
```

b. Change the parameters with the recommended values by running the following commands:

```
alter system set sga_max_size=<sga_value> scope=spfile

alter system set sga_target=<sga_value> scope=spfile;

alter system set pga_aggregate_target=<pga_value> scope=spfile;

shutdown immediate;

startup;
```

c. Make sure that AMM is not enabled by verifying that the following parameters are not greater than 0:

```
show parameter memory_target

show parameter memory_max_target
```

d. If required, unset the parameter(s) by executing the following command and then restart the database:

```
alter system reset memory_target scope=spfile;

alter system reset memory_max_target scope=spfile;
```

Setting AMM Values

While using AMM has not been tested by RSA and some experts DO NOT recommend using AMM management with Linux HugePages, the following commands will help you set Oracle initialization parameters:

If X is the amount memory that can be dedicated to Oracle instance, RSA recommends that you set Oracle initialization parameters as follows:

Parameter Name	Value
memory_target	X
memory_max_target	X
sga_target	X * 2/6
pga_aggregate_target	X * 1/6

Memory must meet our required minimums. On a RAC system with multiple Oracle instances, every instance must meet the same SGA/PGA calculated values. For example, if a host with 128GB of RAM is dedicated to run a single instance of Oracle, allocate 8GB of RAM for the operating system, and set memory_target and memory_max_target to 120GB, SGA minimum value to 40GB, and PGA minimum value to 20GB.

a. Determine the memory allocation by running the following SQL:

```
show parameter memory_target

show parameter memory_max_target

show parameter sga_target

show parameter pga_aggregate_target
```

b. Change the parameters with the recommended values by running the following commands:

```
alter system set memory_target=<memory_value> scope=spfile;

alter system set memory_max_target=<memory_value> scope=spfile;

alter system set sga_target=<sga_value> scope=spfile;

alter system set pga_aggregate_target=<pga_value> scope=spfile;

shutdown immediate;

startup;
```

4. Configure adequate space for all system tablespaces (see Configure Undo, Temp, and Redo Sizes).
5. Make sure that the database instance is configured with the XML_DB package. You can verify that XML DB has been installed by running the following SQL:

```
select comp_name from dba_registry where comp_name like '%XML%';
```

You should see results like the following:

```
COMP_NAME

Oracle XML Database
```

If this package does not exist, it can be installed with the database configuration assistant (dbca) or manually by executing the instructions found in the Oracle documentation at the following location:

```
http://download.oracle.com/docs/cd/B19306_
01/appdev.102/b14259/appaman.htm#CACIBCBA
```

6. Run the following commands to specify the optimizer settings:

alter system set optimizer_index_cost_adj=30 scope=both;

alter system set optimizer_index_caching=50 scope=both;

alter system set optimizer_adaptive_features = false scope=spfile;

7. Keep a default size for the log buffer and log_checkpoint_interval which should be suitable for most

workloads.

Consider increasing log_buffer parameter (up to 200M) whenever you experience high redo log space requests or redo log space wait time, the log_checkpoint_interval parameter should be set so you only encounter a checkpoint at each log switch:

```
alter system set log_buffer=209715200
```

```
alter system set log_checkpoint_interval=188743680
```

8. Configure the creation of deferred segments for tables to false.

   ```
   alter system set DEFERRED_SEGMENT_CREATION = FALSE
   ```

9. Validate that database requirements are reflected in your base Oracle startup by validating the spfile used by your database instance. Run this command:

   ```
   SELECT NAME, Value FROM gv$parameter order by name
   ```

10. Restart the database server.

Create the Required Objects

Create all required table spaces, user schemas, and directories on the database server.

Create Tablespaces

The standard RSA Identity Governance and Lifecycle database setup has eight tablespaces, four for data and four for indices. RSA Identity Governance and Lifecycle uses the well-known tablespace names when creating database objects such as tables and views. Configure these tablespaces with storage settings appropriate for your installation. The following table provides the initial, extended, and fixed size recommendations for the tablespaces.

If you do not want to use autoextend in your database, use the fixed size. If possible, RSA recommends to set the autoextend maxsize value to "unlimited." This requires you to monitor the available space in the OS or ASM as the data file grows.

Tablespace Name	Initial Size	Extend Size	Fixed Size
DATA_256K	25M	25M	500M
DATA_1M	200M	200M	10G

Tablespace Name	Initial Size	Extend Size	Fixed Size
DATA_25M	25M	25M	500M
DATA_50M	200M	200M	10G
INDX_256K	100M	100M	10G
INDX_1M	100M	100M	10G
INDX_25M	400M	200M	10G
INDX_50M	100M	100M	10G

Your database can have different tablespace names or fewer tablespaces. However, because RSA Identity Governance and Lifecycle uses the eight known tablespace names internally, additional configuration is required when installing or upgrading the software. For examples, use the following sample scripts. You must edit them for your configuration before using them:

- **aveksa_sample_tablespace_file_scripts.sql** (creating tablespaces using the file system)
- **aveksa_sample_tablespace_ASM_scripts.sql** (creating tablespaces using ASM with recommended sizes)

You must map your tablespaces to the names known to RSA Identity Governance and Lifecycle.

Configure Undo, Temp, and Redo Sizes

RSA Identity Governance and Lifecycle requires minimum sizes for Oracle's undo, temp and redo logs to perform effectively. Temp and undo each require at least 96 GB. Redo requires at least 2.4 GB.

The following tables provide examples of the database commands to increase sizes for the database files for both Oracle ASM and file system implementations. These examples assume you know the filesystem paths for your Oracle installation and database instance. Ensure there is adequate disk space for the changes.

Increase Temp Log Sizes

The examples in this table show how to increase the temp filespace from the default 32 GB to 96 GB by adding two 32 GB temp files.

ASM	`alter tablespace "temp" add tempfile '+dg01' size 128m reuse autoextend on next 256m maxsize 32767m;` `alter tablespace "temp" add tempfile '+dg01' size 128m reuse autoextend on next 256m maxsize 32767m;`

File System	`alter tablespace "temp" add tempfile '{oracle_base}/oradata/{db_unique_name}/temp02.dbf' size 128m reuse autoextend on next 256m maxsize 32767m;`
	`alter tablespace "temp" add tempfile '{oracle_base}/oradata/{db_unique_name}/temp03.dbf' size 128m reuse autoextend on next 256m maxsize 32767m;`

Increase Undo Log Sizes

The examples in this table show how to increase the undo filespace from the default 32GB to 96GB by adding two 32 GB temp files.

ASM	`alter tablespace "undotbs1" add datafile '+dg01' size 128m reuse autoextend on next 256m maxsize 32767m;` `alter tablespace "undotbs1" add datafile '+dg01' size 128m reuse autoextend on next 256m maxsize 32767m;`

File System	`alter tablespace "undotbs1" add datafile '{oracle_base}/oradata/{db_unique_name}/undotbs02.dbf' size 128m reuse autoextend on next 256m maxsize 32767m;`
	`alter tablespace "undotbs1" add datafile '{oracle_base}/oradata/{db_unique_name}/undotbs03.dbf' size 128m reuse autoextend on next 256m maxsize 32767m;`

Increase Redo Log Sizes

To review your system's redo files, execute the following SQL:

```
select * from v$logfile;
```

RSA Identity Governance and Lifecycle requires six redo logs of at least 800 MB each. By default, the Oracle database is configured with three redo logs of 50 MB each. The examples in the following table demonstrate how to increase the redo log sizes by dropping the three default redo logs and adding six redo logs of 800 MB each.

ASM

```
// Extend the existing redo logs.

alter Database Clear Logfile '+DG01/avdb/onlinelog/group_1.263.763560403';

alter Database Drop Logfile Group 1;

alter Database Add Logfile Thread 1 Group 1 '+DG01/avdb/onlinelog/group_1'
Size 800m;

alter Database Clear Logfile '+DG01/avdb/onlinelog/group_2.262.763560421';

alter Database Drop Logfile Group 2;

alter Database Add Logfile Thread 1 Group 2 '+DG01/avdb/onlinelog/group_2'
Size 800m;

alter Database Clear Logfile '+DG01/avdb/onlinelog/group_3.261.763560439';

alter Database Drop Logfile Group 3;

alter Database Add Logfile Thread 1 Group 3 '+DG01/avdb/onlinelog/group_3'
Size 800m;

// Add three additional redo logs.

alter Database Add Logfile Thread 1 Group 4 '+DG01/avdb/onlinelog/group_4'
Size 800m;

alter Database Add Logfile Thread 1 Group 5 '+DG01/avdb/onlinelog/group_5'
Size 800m;
```

```
alter Database Add Logfile Thread 1 Group 6 '+DG01/avdb/onlinelog/group_6'
Size 800m;
```

File System

```
// Extend the existing redo logs.

alter Database Clear Logfile '{ORACLE_BASE}/oradata/{DB_UNIQUE_
NAME}/GROUP_1';

alter Database Drop Logfile Group 1;

alter Database Add Logfile Thread 1 Group 1 '{ORACLE_BASE}/oradata/{DB_
UNIQUE_NAME}/group_1' Size 800m;

alter Database Clear Logfile {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/GROUP_
2';

alter Database Drop Logfile Group 2;

alter Database Add Logfile Thread 1 Group 2 '{ORACLE_BASE}/oradata/{DB_
UNIQUE_NAME}/group_2' Size 800m;

alter Database Clear Logfile '{ORACLE_BASE}/oradata/{DB_UNIQUE_
NAME}/REDO03.LOG';

alter Database Drop Logfile Group 3;

alter Database Add Logfile Thread 1 Group 3 '{ORACLE_BASE}/oradata/{DB_
UNIQUE_NAME}/group_3' Size 800m

// Add three additional redo logs.

alter Database Add Logfile Thread 1 Group 4 '{ORACLE_BASE}/oradata/{DB_
UNIQUE_NAME}/group_4' Size 800m

alter Database Add Logfile Thread 1 Group 5 '{ORACLE_BASE}/oradata/{DB_
UNIQUE_NAME}/group_5' Size 800m

alter Database Add Logfile Thread 1 Group 6 '{ORACLE_BASE}/oradata/{DB_
UNIQUE_NAME}/group_6' Size 800m
```

Create the Export/Import Database Directory

The Export/Import directory is used for the storage and retrieval of database backups (if RSA Identity Governance and Lifecycle provided backup and restore scripts are used). This directory is optional if you are using your own service tools or scripts to backup and restore the database. Otherwise, this directory is mandatory.

You must create the directory in the filesystem on the database server before you begin database instance configuration. See Map the Export/Import Directory for more information. Create the directory, for example: `/home/aveksa/AveksaExportImportDir`

Directory Permissions

The database user must be provided Read-Write permissions to the Export/Import directory. It is accessed by the user running the associated backup/restore scripts.

Directory Size Allotment

For the Export/Import directory, make sure the disk where the directory is located can accommodate at least the estimated size of your exported database backup.

RAC Database Implementation Only

In an Oracle RAC installation, the Export/Import directory must point to clustered file system locations. The following instructions use an example environment variable, CLUSTERED_FILE_SHARE, which can be freely substituted for whatever common location is defined for your installation.

1. Make sure the mount point is owned by oracle (and not root). For example:

   ```
   chown -R oracle:oinstall /mnt/acfs-fs1/*
   ```

2. Set the environment variable on all nodes to the mount point. For example:

   ```
   export CLUSTERED_FILE_SHARE=/mnt/acfs-fs1
   ```

   ```
   cd ${CLUSTERED_FILE_SHARE}
   ```

3. Create the directory on the file share. This is only required from one node. For example:

   ```
   mkdir AveksaExportImportDir
   ```

4. Create symbolic links to the mount point. This creates the expected directory location within the /home/aveksa root. For example:

   ```
   ln -s ${CLUSTERED_FILE_SHARE}/AveksaExportImportDir
   /home/oracle/AveksaExportImportDir
   ```

Map the Export/Import Directory

See the aveksa_sample_ora12_db_scripts.sql for installations using Oracle 12cR1 (12.1.0.2) for examples on how to map the directory:

Map the directory variable to the physical directories previously created as described in Create the Aveksa Export/Import Database Directory:

```
create or replace directory AVEKSA_EXPORTIMPORT_DIRECTORY as
‘/home/oracle/AveksaExportImportDir’;
```

Create a Database User Password Profile

This section describes how to create a database profile for the users that connect to the database.

Oracle 12cR1 has a default password expiration of 180 days. If a database user password were to expire, RSA Identity Governance and Lifecycle would fail to connect to the database.

If you choose to have a password policy that expires for the RSA Identity Governance and Lifecycle users, you will have to reconfigure the user database password settings when passwords expire. The sample file aveksa_db_password_lifetime.sql shows how to obtain the password lifetime information for the RSA Identity Governance and Lifecycle database users.

See the aveksa_sample_ora12_db_scripts.sql for installations using Oracle 12cR1 (12.1.0.2, 64-bit) script for examples on how to configure a database user password profile:

Enter the following command to create the profile:

```
Create Profile ACMPROFILE LIMIT PASSWORD_LIFE_TIME UNLIMITED;
```

RSA does not require that you include the Oracle SYS user in the profile. The Oracle SYS user password, therefore, will expire at some point. RSA recommends you do.the following: change the password in the database and the application, reset the password to its current value, or include the SYS user in the profile.

Create User Schemas

See the aveksa_sample_ora12_db_scripts.sql script for installations using Oracle 12cR1 (12.1.0.2) for examples on how to configure user schemas:

Create the following user schemas:

`AVUSER` (the RSA Identity Governance and Lifecycle user)

`AVDWUSER` (the RSA Identity Governance and Lifecycle reporting engine)

`ACMDB` (the RSA Identity Governance and Lifecycle public database schema)

If you are not using the default user schema names (AVUSER and others), substitute your user schema names in the scripts provided. Observe Oracle password guidelines when creating the user schemas. For example, although Oracle accepts a '$' character in a password, it does not recommend that you use the character. See Oracle’s password guidelines for more information .

Note: Additional configuration is required when installing RSA Identity Governance and Lifecycle with non-default schema names.

Command examples:

```
Create USER AVUSER identified by <password> profile ACMPROFILE;

ALTER USER AVUSER DEFAULT TABLESPACE DATA_1M TEMPORARY
TABLESPACE TEMP;

Create USER AVDWUSER identified by <password> profile
ACMPROFILE;

ALTER USER AVDWUSER DEFAULT TABLESPACE DATA_1M TEMPORARY
TABLESPACE TEMP;

Create USER ACMDB identified by <password> profile ACMPROFILE;

ALTER USER ACMDB DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE
TEMP ACCOUNT UNLOCK;
```

Record the passwords in the worksheet provided in Database Parameter Values Worksheet. The passwords are required for the RSA Identity Governance and Lifecycle installation.

Configure the User Schema Privilege Grants

See the aveksa_sample_ora12_db_scripts.sql script for installations using Oracle 12cR1 (12.1.0.2) for examples on how to configure the AVUSER, AVDWUSER, and ACMDB user schemas.

Privilege grants:

- AVUSER

  `grant unlimited tablespace to AVUSER;`

  `grant create session to AVUSER;` (Used for Application server access)

  `grant create table to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant create view to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant create trigger to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant create sequence to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant create synonym to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant create procedure to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant create type to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant create job to AVUSER;` (Database object used for RSA Identity Governance and Lifecycle runtime and migration)

  `grant read, write on directory AVEKSA_EXPORTIMPORT_DIRECTORY to AVUSER;` (Optional. Used for import/export of Oracle database dumps.)

  `grant select on dba_free_space to AVUSER;` (Provides database information for Statistics Reports)

  `grant select on gv_$parameter to AVUSER;` (Provides database information for Statistics Reports)

  `grant execute ON XDB.DBMS_XMLPARSER to AVUSER;` (Used to process XML data attributes and documents)

  `grant execute ON SYS.DBMS_XMLGEN to AVUSER;` (Used to process XML data attributes and documents)

  `grant execute ON SYS.DBMS_LOB to AVUSER;` (Used to process large sql statements)

  `grant execute ON SYS.DBMS_SQL to AVUSER;` (Used to process large sql statements)

  `grant execute ON SYS.DBMS_SCHEDULER to AVUSER;` (Data model manipulation for custom attributes)

- AVDWUSER

  `grant unlimited tablespace to AVDWUSER;` (Optional)

  `grant create session to AVDWUSER;`

```
grant create synonym to AVDWUSER;

grant create procedure to AVDWUSER;
```

- ACMDB

  `grant unlimited tablespace to ACMDB;` (Optional)

  ```
  grant create session to ACMDB;

  grant create synonym to ACMDB;

  grant create procedure to ACMDB;
  ```

Note: Object privileges must be granted directly to the user. Use of Oracle roles for privileges may not be sufficient for DBMS objects.

Additional grant information:

- Aveksa Statistics Reports requires grants on the dba_free_space and gv_$parameter system views to obtain useful diagnostic information:

  ```
  grant select on SYS.DBA_FREE_SPACE to AVUSER;

  grant select on SYS.GV_$PARAMETER to AVUSER;
  ```

- A grant to execute SYS.DBMS_LOB by the public should also be verified. On some systems the packages DBMS_XMLPARSER and DBMS_XSLPROCESSOR must be recompiled:

  ```
  grant execute on SYS.DBMS_LOB to PUBLIC;

  alter PACKAGE xdb.DBMS_XMLPARSER compile body;

  alter PACKAGE xdb.DBMS_XSLPROCESSOR compile body;
  ```

- A grant to execute the XDB.DBMS_XMLPARSER package is required by AVUSER for XML processing and metadata operations.

  ```
  grant execute ON XDB.DBMS_XMLPARSER TO AVUSER;
  ```

- AVUSER requires a grant for write-access to the sys.PLAN_TABLE table. In SQL*Plus, log in as SYSDBA and execute sql-script "$ORACLE_HOME/rdbms/admin/catplan.sql" as:

  ```
  @$ORACLE_HOME/rdbms/admin/catplan.sql
  ```

Create a Report Context

The report context holds the filtering values for RSA Identity Governance and Lifecycle reports.

Procedure:

```
create or replace context AV_REPORT_CONTEXT using
AVUSER.Reporting_Pkg;
```

Complete the Required Database Instance Configuration

See aveksa_sample_sys_scripts.sql for a sample script on how to perform this step. This script must be edited for your configuration before you use it. Complete database instance configuration as sysdba.

Procedure

Disable the daily job that gathers schema statistics. The RSA database executes its own database statistics collections and can conflict with this out-of-box Oracle setting.

```
execute dbms_auto_task_admin.disable(client_name => 'auto optimizer stats
collection', operation => NULL, window_name => NULL);
```

To verify, execute the following SQL:

```
SELECT client_name, status FROM dba_autotask_client where client_name =
'auto optimizer stats collection';
```

Confirm that the job is disabled as show in the example output:

```
CLIENT_NAME STATUS

-------------------------------------------------- --------

auto optimizer stats collection DISABLED
```

Installing Oracle Statspack to Enhance Database Diagnostics Capabilities

Oracle Statspack collects and compiles performance and execution statistics data that RSA Identity Governance and Lifecycle can include in reports generated by its Aveksa Statistics Report feature.

Installation of Statspack is optional but highly recommended; it provides diagnostics data that can indicate the causes of database performance issues that may arise.

You install Statspack on the database using the spcreate.sql script provided by Oracle. This script is executed as 'sys' on the database. The script prompts you for information and creates the STATSPACK schema owner, privilege grants, and objects. See Oracle documentation for complete information on Statspack installation and Statspack capabilities.

For example, to run the script in SQL*Plus:

Create a “perfstat” table space using the spcreate.sql script for Oracle12.1.0.2.

```
SQL> connect / as sysdba

SQL> @?/rdbms/admin/spcreate
```

The default username is “perfstat.” See the *Installation Guide* for information on changing the username and the Statspack user password in RSA Identity Governance and Lifecycle.

RSA recommends that the perfstat user is also configured with the same ACMPROFILE as configured for database users (as described in Create a Database User Password Profile), to prevent issues regarding password expiration.

As sys dba, execute the sql:

```
alter user perfstat profile ACMPROFILE;
```

Verify Correct Database Configuration

Use the following commands to verify that the database used by RSA Identity Governance and Lifecycle has been configured correctly:

Verify that the import/export described in Deployment Summary is defined:

```
select * from all_directories where directory_name in ('AVEKSA_
EXPORTIMPORT_DIRECTORY');
```

Verify that the tables spaces described in Create Tablespaces exist:

```
SELECT * FROM USER_TABLESPACES WHERE TABLESPACE_NAME IN ('DATA_
256K','DATA_1M','DATA_25M','DATA_50M','INDX_256K','INDX_
1M','INDX_25M',
'INDX_50M');
```

Validate the XML package exists:

```
select username from all_users where username='XDB';
```

Validate the schemas described in exist:

```
select * from all_users where username in ('AVUSER', 'AVDWUSER',
'ACMDB');
```

Verify the timezone settings within the database. As sys dba, execute the following SQL statements:

1. SELECT DBTIMEZONE FROM DUAL;
2. SELECT avuser.Utilities_Pkg.Get_DBTimezone_Value FROM DUAL;

If the values from those two queries are not exactly the same, execute the following SQL statements:

1. alter database set time_zone='<value you got from the previous second query>';
2. shutdown immediate;
3. startup;

Verify that the value has in fact been updated to the value you have specified by executing the SQL statement:

SELECT DBTIMEZONE FROM DUAL;

Updating the Database for RSA Identity Governance and Lifecycle Upgrades

If you are upgrading from a pre-5.0 version of RSA Identity Governance and Lifecycle, you must update these additional privileged grants (required for Aveksa Statistics Report generation) for AVUSER:

```
grant select on dba_free_space to AVUSER;
```

```
grant select on gv_$parameter to AVUSER;
```

Chapter 3: Maintaining the Database

Back Up the Customer-Supplied Database

It is important to back up your database regularly or before making major changes to your system. You perform the backup procedure for the Oracle database on an database server machine using Oracle's expdp data pump utility. It creates a .dmp file of the AVUSER schema.

The dump is essentially a snapshot of the database containing all of the application data and some environment data about a particular system environment. In the case where you intend to import a dump from one machine to another (which must be running the same RSA Identity Governance and Lifecycle version from which the dump was created), you may be required to perform additional configuration on the target machine.

The dumps are upwardly compatible between Oracle versions. They are not backwards compatible when used to import to an older version of Oracle.

Before You Begin

- Ensure that the AVEKSA_EXPORTIMPORT_DIRECTORY directory has been created as described in Set Up the Database.
- Run the following query to identify if the directory structure exists, who owns it, and the directory that it points to on the database server:

  ```
  select owner,directory_name, directory_path from all_directories
  where directory_name = 'AVEKSA_EXPORTIMPORT_DIRECTORY';
  ```

- Run the following query to verify that the AVUSER or the schema owner has the appropriate privileges. The schema needs both Read and Write privileges.

  ```
  select * from all_tab_privs
  where table_name='AVEKSA_EXPORTIMPORT_DIRECTORY';
  ```

Procedure

1. Shut down the RSA Identity Governance and Lifecycle server before you export a database.
2. Shut down AFX if it is installed.
3. Run the following command from the database server machine:

   ```
   expdp avuser/<password>@<Oracle_SID> DumpFile=<FileName>.dmp
   Directory=Aveksa_ExportImport_Directory Schemas=avuser
   LogFile=<FileName>.log
   ```

   Where:

`Expdp` is the Oracle data pump utility.

`avuser/<password>` is the connection string.

`DumpFile` is the output file name; here set with a date stamp.

`Directory` is an internal Oracle directory object mapped to a physical UNIX directory. It would typically be the AveksaExportImportDir directory created when the customer-provided database was set up.

`Schemas` is the database, `avuser` for example.

`Logfile` is the name of the log file generated for the export.

Starting with the 7.0.1 product release, a new data encryption handling mechanism is in place which uses on disk (outside of the DB) data in conjunction with DB data to perform data encryption. When exporting DB data and moving it to a new installation it is no longer sufficient just to export/"pack up" the DB data and reimport it. The encryption key data stored on disk must also be "packed up" (see Step 2) and moved/reinstalled along with the DB data.

4. Zip up the master key data in the master key storage directory. This directory is identified by the rsavialg.security.keydir environment variable. The default directory is /home/oracle/security.

   You will unzip the master key data as part of the steps to import the database described in Importing AVUSER Schema/Data for a Customer-Supplied Database Restoration/Load.

   Note: In a clustered environment, if separate copies of the key data are stored on each cluster node, then only one copy of a key data (any node's key data) needs to be backed up as all areas should contain the same key data. However, when reinstalling the data, if a node has a local directory specified for storage of keys, then the key data should be reinstalled to each of these local directories (as specified by the rsavialg.security.keydir environment variable which each node has set).

Importing AVUSER Schema/Data for a Customer-Supplied Database Restoration/Load

You can import a AVUSER schema/data back up for restoring a database or loading it on a new machine. You perform the import procedure for the RSA Identity Governance and Lifecycle Oracle database on the database server machine using Oracle's impdp data pump utility. It uses a .dmp file of the AVUSER schema created from the export process. If you intend to import a dump from one machine to another, you may be required to perform additional configuration on the target machine. The target machine must be running the same RSA Identity Governance and Lifecycle version as the version from which the dump was created.

The server nodes may need to be updated; this is particularly true when moving clustered environments. Or directory specific locations configured for collectors may need to change. After you have imported the database, you must validate that the data is compatible with your database. See Validate Compatibility of the Database Import on page 30 for instructions.

The dumps are upwardly compatible between Oracle versions. They are not backwards compatible when used to import to an older version of Oracle.

Note: Ensure that the AVEKSA_EXPORTIMPORT_DIRECTORY directory has been created as described in Set Up the Database on page 10 and the database process has Read-Write permissions to the directory.

Procedure

1. Shut down RSA Identity Governance and Lifecycle.

   Starting with the 7.0.1 product release, a new data encryption handling mechanism is in place which uses on disk (outside of the DB) data in conjunction with DB data to perform data encryption. When exporting DB data and moving it to a new installation it is no longer sufficient just to export/"pack up" the DB data and reimport it. The encryption key data stored on disk must also be "packed up" and moved/reinstalled along with the DB data (see Step 2).

2. Unzip the master key data in the master key storage directory.

   This is the data that was zipped up in the procedure described in "Export the AVUSER Schema/Data for an RSA Database Backup." (The master key storage directory is identified by the rsavialg.security.keydir environment variable. The default directory is /home/oracle/security.)

   Note: In a clustered environment, if separate copies of the key data are stored on each cluster node, then only one copy of a key data (any node's key data) needs to be backed up as all areas should contain the same key data. However, when reinstalling the data , if a node has a local directory specified for storage of keys, then the key data should be reinstalled to each of these local directories (as specified by the rsavialg.security.keydir environment variable which each node has set).

3. Remove the avuser user from the database:

   ```
   drop user AVUSER cascade;
   ```

4. Create the avuser user:

   ```
   Create USER AVUSER identified by <password> profile ACMPROFILE;
   ```

   ```
   ALTER USER AVUSER DEFAULT TABLESPACE DATA_1M TEMPORARY TABLESPACE
   TEMP;
   ```

5. Specify avuser privilege grants as specified in Configure the User Schema Privilege Grants on page 24.
6. Import the schema/data:

   ```
   impdp avuser/<password>@<Oracle_SID> DumpFile=<SomeFileName>.dmp
   Directory=Aveksa_ExportImport_Directory Schemas=avuser
   LogFile=<SomeFileName>.log
   ```

7. If the database does not require migration, then run these commands as avuser to refresh database statistics:

   ```
   EXEC DBMS_STATS.GATHER_SCHEMA_STATS('AVUSER');
   ```

   ```
   EXEC DATABASE_STATISTICS.AFTER_IMPORT;
   ```

   If the database does require migration, you will be prompted to migrate the database when you access RSA Identity Governance and Lifecycle. At the prompt, enter the password **AuthorizeMigration** and click **Migrate Schema**.

8. Restart RSA Identity Governance and Lifecycle.

Validate Compatibility of the Database Import

After you import the database on your system, determine whether the database dump is compatible for your database. Determine whether these system settings from the imported dump are set as follows:

- isAppliance = No
- isRemoteDB = Yes
- isSoftAppliance = Yes

To determine these values, run the following SQL as avuser:

```
select * from T_SYSTEM_SETTINGS where PARAMETER like 'is%';
```

If the values above are not set to the correct values, run the following SQL to set them to the correct values:

```
update T_SYSTEM_SETTINGS set VALUE='N' where PARAMETER='isAppliance';

update T_SYSTEM_SETTINGS set VALUE='yes' where PARAMETER='isRemoteDB';

update T_SYSTEM_SETTINGS set VALUE='yes' where
PARAMETER='isSoftAppliance';
```

If the exported database file is from an appliance and it is imported into a remote database, ensure that the system setting for "RemoteDB" on the target system that uses the remote database is enabled. After saving the RemoteDB setting you must restart the RSA Identity Governance and Lifecycle application server.

Removing User Schemas from the Database

All RSA Identity Governance and Lifecycle data is contained in three user schemas. As the database administrator you can remove these schemas. This procedure is required if you want to create a new database instance.

Procedure

Login to the database as an administrator and run the following commands:

```
drop user AVUSER cascade;

drop user AVDWUSER cascade;

drop user ACMDB cascade;
```

After you remove the database users and the associated schemas, you can recreate those users and privileges. Restart RSA Identity Governance and Lifecycle to update the new schemas on startup.

For more information and procedures to recreate users and privileges, see:

- Create User Schemas on page 23
- Configure the User Schema Privilege Grants on page 24
- Create a Report Context on page 25

Database Segment Maintenance

When using the data purging and archiving features in RSA Identity Governance and Lifecycle, you must ensure that the tables are properly maintained. When data is removed from a table in Oracle, the database retains the space that had stored the deleted data. As a result, after removing data, the system may not see a significant change in performance, because some SQL statements scan the empty blocks. Oracle provides commands to appropriately maintain these objects, allowing for increased performance when accessing the database.

RSA Identity Governance and Lifecycle has added the ArchivePurge_PostCare_Pkg package, which uses Oracle's Segment Advisor to analyze the tables and indexes and to identify which objects need to be rebuilt. The Segment Advisor contains Oracle's internal knowledge to optimize performance. While you can use the Segment Advisor outside of the package provided by RSA, the package uses the Segment Advisor to target the tables that have been purged from the database. Since analyzing the database can consume time and resources, this targeted approach limits the analysis to the tables that are most affected. This package also adheres to the purge time limit that is configured in the application.

While this process can initially be time consuming, especially for older databases with large amounts of data from prior product versions, running this package provides the following benefits:

- **Reduction in disk space.** Oracle tables and indexes do not release any space they have allocated after data is removed from a table. Once the space has been released, it is available for use by other tables, which eliminates the need to increase data storage.
- **Improved performance.** When data is deleted from a table, the space that is retained by the table is empty, but Oracle may still scan those blocks during the execution of a query. Releasing the space from the table or index reduces the number of blocks that Oracle has to scan, which improves performance.

Best Practices

RSA recommends the following practices with regards to performing this maintenance:

- Run the Analysis and Fulfillment process during times of low system activity. If possible, put the application in maintenance mode, as the process can consume a significant amount of CPU and IO.
- If possible, for initial runs of the Segment Advisor, increase the maximum purge time on the Data Management tab.
- Run this process on a monthly basis to review and implement recommendations.
- Run this process when there are significant changes in the data set. For example, run the process after multiple archives have been created and purged, or when the purging process is executed several times during a week, to reduce the data footprint.

Package Details

The ArchivePurge_PostCare_Pkg contains the following:

FUNCTION SA_ANALYZE_FULFILL RETURNS NUMBER

This function executes the Segment Advisor and executes each recommendation generated by the Segment Advisor run. The function returns a number, which is a Run ID. Keep this Run ID to confirm that all work has been completed.

FUNCTION SA_ANALYZE RETURNS NUMBER

This function executes the Segment Advisor and stores the recommendations in the RSA Identity Governance and Lifecycle tables. The function returns a number which is a Run ID. Keep this Run ID to identify the list of recommendations and use when implementing the recommendations.

Argument Name	IN_RUN_ID
Type	NUMBER(38)
In/Out Default?	IN

Note: At this time, do not pass the parameter in.

PROCEDURE SA_FULFILL

This parameter is the value returned by a prior SA_ANALYZE or SA_ANALYZE_FULFILL run.

Argument Name	IN_RUN_ID
Type	NUMBER(38)
In/Out Default?	IN

Run the Database Segment Package

Perform the following steps to use the ArchivePurge_PostCare_Pkg package.

Procedure

1. Run SA_ANALYZE_FULFILL:
   a. Call the package to run the advisor and fulfill the recommendations.

      For example, using the typical syntax for SQL*Plus:

      ```
      set serveroutput on;
      exec dbms_output.put_line(ArchivePurge_PostCare_Pkg.sa_analyze_
      fulfill);
      ```

   b. After completion, run the following query to determine if all recommendations have completed. The following query uses the <RunId> returned from the previous call:

      ```
      Select count(*) from t_sa_run_details
      Where run_id = < RunId >
      and script_finished is null;
      ```

   c. If the count returns a value greater than 0, there are additional calls to fulfill those recommendations. Use the SA_FULFILL procedure to fulfill the additional recommendations.

      ```
      exec ArchivePurge_PostCare_Pkg.sa_fulfill( < RunID > );
      ```

   d. Repeat steps b and c until all recommendations are complete.
2. Run SA_ANALYZE and SA_FULFILL:

   a. Call the package to run the advisor and full fill the recommendations.

      For example, using the typical syntax for SQL*Plus:

      ```
      set serveroutput on;
      exec dbms_output.put_line(ArchivePurge_PostCare_Pkg.sa_analyze);
      ```

   b. After completion, run the following query to determine the number of recommendations. The following query uses the <RunId> returned from the previous call:

      ```
      Select count(*) from t_sa_run_details
      Where run_id = < RunId >
      and script_finished is null;
      ```

   c. If the count returns a value greater than 0, there are additional calls to fulfill those recommendations. Use the SA_FULFILL procedure to fulfill the additional recommendations.

```
exec ArchivePurge_PostCare_Pkg.sa_fulfill( < RunID > );
```

d. Repeat steps b and c until all recommendations are complete.

3. To get a list of recommendations from a run of a procedure, use the following SQL with the RunId returned from the package:

```
SELECT
object_name,
object_type,
finding_info,
script_finished,
st.column_value sql_commands

FROM
t_sa_run_details rd,
TABLE ( utilities_pkg.tokenizer(script_to_run,';') ) st

where rd.run_id = 1
order by action_id;
```

Where:

- **Object Name** is the name of the object that the recommendation is maintaining.
- **Object Type** is the type of object, such as a table, index, or LOB.
- **Finding Info** is what is gained by implementing the recommendation.
- **Script Finished** is the date and time when the recommendation was completed.
- S**QL Commands** are the SQL commands that are run to implement the recommendation.

In some cases, implementing a recommendation may require multiple steps such as shrinking a table. In that type of recommendation, this query might return multiple records for the same Object Name / Object Type / Finding Info / Script Finished values.