# USER GUIDE

| | |
|---|---|
| Product | Snow Inventory Oracle Scanner |
| Version | 5 |
| Release date | 2016-09-27 |
| Document date | 2020-05-26 |

# CONTENTS

# 1 INTRODUCTION

This document describes the Snow Inventory Oracle Scanner.

The Snow Inventory Oracle Scanner is a Java program. This makes the scanner platform independent and it can be run on a variety of platforms.

**NOTE**
All inventory results provide a snapshot of the environment from the time when the inventory scan was performed.

## 1.1 PREREQUISITES

### 1.1.1 DEPENDENCY ON OTHER SNOW PRODUCTS

The Snow Inventory Oracle Scanner is started from a Snow Inventory Agent. This version of the Inventory agent can only be used in a Snow Inventory Server 5 environment.

### 1.1.2 PERMISSIONS AND ENVIRONMENT VARIABLES

The Operating System User running the Inventory agent:

- must be a member of the operating system DBA group or use a dedicated database user described in Oracle inventory with user authentication.
- must have access to the sqlplus file and folder.

On all servers where the Snow Inventory Oracle Scanner is to be run:

- The **ORACLE_HOME** environment variable must be set, either in the configuration file of the agent or in the context of the Operating System User running the agent.
- The **PATH** environment variable must include the path to the Java Runtime Environment and the path to sqlplus.

For information on supported operating systems, see Snow System Requirements document, in the Snow Knowledge Base, which is available for customers and partners.

### 1.1.3 JAVA RUNTIME ENVIRONMENT

To run the Snow Inventory Oracle Scanner, the target computer is required to have Java Runtime Environment 6 (1.6), 7 or 8 installed. We recommend to use Java Runtime Environment 8 since there are performance gains in doing so.

**NOTE**
Due to an internal defect in Java, Java Runtime Environment 1.7.0_7 must not be used. See Known issues for more information.

## 1.2 PRIVILEGES

### 1.2.1 WINDOWS

In Windows the Oracle Scanner is run with an administrative user account (Local Admin on the server).

### 1.2.2 LINUX AND UNIX

In Linux and Unix there are two privilege options for running the Oracle Scanner, and each option is described in the sections that follow.

#### 1.2.2.1 FULL PRIVILEGES

For full privileges, either a user with root privileges or a user with privileges to run the Java runtime with the **sudo** program is required.

#### 1.2.2.2 PRINCIPLE OF LEAST PRIVILEGES

A user is required with **sudo** rights to the operating system commands outlined in the table below. The **NOPASSWD** option needs to be set on the sudoers file.

An Oracle database user is required for each database to be inventoried, and needs to be defined in the configuration file. Either the user is the same for all databases and configured using the <DefaultInstanceCredentials> element, or the user can be unique and configured using the <InstancesWithConfiguration> element.

To run Snow Inventory Oracle Scanner with operating-system authentication instead of authenticating by means of a dedicated Oracle-database user for each database instance, the local user that runs Snow Inventory Oracle Scanner must be member of the **dba** group.

Example from sudoers file for Solaris:

User **snow** has the rights to execute the commands with **sudo** and **no password**:

```
##
## User privilege specification
##
root ALL=(ALL) ALL
snow ALL=NOPASSWD: /usr/bin/pwdx
```

Example from sudoers file for Linux:

User **snow** has the rights to execute the commands with **sudo** and **no password**:

The following commands are used for determining **OracleHome** and the location of running processes.

```
##
## User-privilege specification
##
root ALL=(ALL) ALL
snow ALL=NOPASSWD: /usr/bin/ls
snow ALL=NOPASSWD: /usr/bin/ls -l /proc/[0-9]*/cwd
```

| Operating system | Command |
|---|---|
| **Solaris** | pwdx |
| **AIX and Linux** | ls<br><br>ls -l /proc/[0-9]*/cwd |
| **HP-UX** | pfiles |

## 1.3 FILES

| Executable | Description |
|---|---|
| **sios.jar** | This is the only required file for the Snow Inventory Oracle Scanner. It is an executable Java archive that performs inventory of Oracle databases on the current server that it is executed on.<br><br>Sios.jar is started from a Snow Inventory Agent. |
| **snowagent.config** | Configuration file used for configuration of the Snow Inventory Oracle Scanner. |
| **sios.rotation0.log** | Log file |
| **sios.properties** | Properties file containing information about installation date, last run date, Java version, output filename, and version. |

# 2 AUTOMATIC ORACLE INVENTORY

The Snow Inventory Oracle Scanner is designed to perform automatic Oracle inventory without the need of providing or creating a specific user account within the Oracle databases. It automatically discovers all running Oracle instances on the current server, detects what user is running each database instance, and switches to that specific user when inventorying the databases. No specific permissions or no specific user is needed for the databases that are to be inventoried.

The Oracle Scanner is designed to be lightweight and scalable regarding size and number of objects in database. Resource consumption is mainly CPU bound since metadata is often available in memory.

The Snow Inventory Oracle Scanner is non-invasive and read-only, meaning that only read operations are performed when querying the database(s). It performs only SELECT queries based on PL/SQL blocks to extract necessary inventory data that is used for analyzing needs of a license.

Only Oracle internal objects listed in Oracle permissions required are accessed. The Oracle Scanner never reads user or application data, so there should be no data security concerns related to automatic Oracle inventory.

Snow Software recommends the use of Automatic Oracle Inventory rather than setting up specific user accounts and permission for each Oracle database.

## 2.1 ORACLE INVENTORY WITH USER AUTHENTICATION

A user account needs to be created with SELECT permissions on ALL TABLES and DICTIONARY in all databases. The account also needs to be able to CREATE SESSION. This is done by granting the user “SELECT ANY” privileges on “TABLE” and “DICTIONARY”, see the example below.

The user needs to be created and permissions need to be granted on each database that is to be inventoried.

**EXAMPLE**
Create the user <Oracle Scanner User> and grant SELECT ANY privileges in earlier versions than Oracle 12 (non-container):

```
CREATE USER <Oracle Scanner User> identified by <Password>;
GRANT CREATE SESSION TO <Oracle Scanner User>;
GRANT SELECT ANY TABLE to <Oracle Scanner User>;
GRANT SELECT ANY DICTIONARY to <Oracle Scanner User>;
```

For Oracle 12, the user needs to be created as a common user, which is done by typing “c##” before the username. Also, CONTAINER=ALL must be added to each line, see example below.

**EXAMPLE**
Create the user <Oracle Scanner User> and grant SELECT ANY privileges in a container database:

```
CREATE USER c##<Oracle Scanner User> identified by <Password> CONTAINER=ALL;
GRANT CREATE SESSION TO c##<Oracle Scanner User> CONTAINER=ALL;
GRANT SELECT ANY TABLE to c##<Oracle Scanner User> CONTAINER=ALL;
GRANT SELECT ANY DICTIONARY to c##<Oracle Scanner User> CONTAINER=ALL;
```

## 2.2 ORACLE PERMISSIONS REQUIRED

When performing automatic Oracle inventory via the user that runs the database instance, the required permissions are already granted to the user. When performing Oracle inventory using a specific Oracle user, the user needs to be granted the permissions required as illustrated in the section above (Oracle inventory with user authentication).

In both cases the following objects are accessed and the user needs to be able to run SELECT queries towards these tables/views. Also, if Oracle built-in roles and privileges cannot be used due to security reasons, grants can be made to the following objects.

- all_sdo_geom_metadata
- all_views
- cdb_pdb_history
- dba_advisor_tasks
- dba_aws
- dba_cpu_usage_statistics
- dba_cubes
- dba_encrypted_columns
- dba_feature_usage_statistics
- dba_flashback_archive
- dba_flashback_archive_tables
- dba_lob_partitions
- dba_lobs
- dba_lob_subpartitions
- dba_objects
- dba_object_tables
- dba_recyclebin
- dba_registry
- dba_segments
- dba_sql_profiles

- dba_sqlset
- dba_sqlset_references
- dba_tables
- dba_tablespaces
- dba_tab_partitions
- dba_tab_subpartitions
- dba_users
- dba_workload_captures
- dba_workload_filters
- dba_workload_replays
- dmsys.dm$model
- dmsys.dm$object
- dual
- dvsys.dba_dv_realm
- dvsys.dba_dv_realm_auth
- global_name
- gv$instance
- gv$parameter
- lbacsys.lbac$polt
- mdsys.sdo_feature_usage
- mdsys.sdo_geom_metadata_table
- mgmt_targets
- odm_document
- odm.odm_mining_model
- odm_record
- olapsys.dba$olap_cubes
- redaction_policies
- sys.dba_mining_models
- sys.dba_users
- sysman.mgmt_fu_registrations
- sysman.mgmt_license_definitions
- sysman.mgmt_licenses
- sysman.mgmt_targets
- sys.model$

- timesten.tt_gridid
- timesten.tt_gridinfo
- user_role_privsuser_role_privs
- v$archive_dest_status
- v$block_change_tracking
- v$containers
- v$database
- v$instance
- v$license
- v$option
- v$parameter
- v$pdbs
- v$session
- v$session_connect_info
- v$version

### 2.2.1 EXAMPLE: ORACLE PERMISSIONS REQUIRED

In this scenario, the Oracle built-in roles and privileges cannot be used due to security reasons.

The following example script creates the user **OSCAN**, and then adds the required Oracle permissions, which are used for non-container databases:

```
CREATE USER OSCAN IDENTIFIED BY <passwd>;
GRANT CREATE SESSION TO OSCAN;
GRANT EXECUTE ON SYS.DBMSOUTPUT_LINESARRAY TO OSCAN;
GRANT EXECUTE ON SYS.DBMS_APPLICATION_INFO TO OSCAN;
GRANT EXECUTE ON SYS.DBMS_OUTPUT TO OSCAN;
GRANT SELECT ON SYS.ALL_VIEWS TO OSCAN;
GRANT SELECT ON SYS.CDB_PDB_HISTORY TO OSCAN;
GRANT SELECT ON SYS.DBA_ADVISOR_TASKS TO OSCAN;
GRANT SELECT ON SYS.DBA_AWS TO OSCAN;
GRANT SELECT ON SYS.DBA_CPU_USAGE_STATISTICS TO OSCAN;
GRANT SELECT ON SYS.DBA_CUBES TO OSCAN;
GRANT SELECT ON SYS.DBA_ENCRYPTED_COLUMNS TO OSCAN;
GRANT SELECT ON SYS.DBA_FEATURE_USAGE_STATISTICS TO OSCAN;
GRANT SELECT ON SYS.DBA_FLASHBACK_ARCHIVE TO OSCAN;
GRANT SELECT ON SYS.DBA_FLASHBACK_ARCHIVE_TABLES TO OSCAN;
GRANT SELECT ON SYS.DBA_FLASHBACK_ARCHIVE_TS TO OSCAN;
GRANT SELECT ON SYS.DBA_LOBS TO OSCAN;
GRANT SELECT ON SYS.DBA_LOB_PARTITIONS TO OSCAN;
GRANT SELECT ON SYS.DBA_LOB_SUBPARTITIONS TO OSCAN;
GRANT SELECT ON SYS.DBA_MINING_MODELS TO OSCAN;
GRANT SELECT ON SYS.DBA_OBJECTS TO OSCAN;
GRANT SELECT ON SYS.DBA_OBJECT_TABLES TO OSCAN;
GRANT SELECT ON SYS.DBA_RECYCLEBIN TO OSCAN;
GRANT SELECT ON SYS.DBA_REGISTRY TO OSCAN;
GRANT SELECT ON SYS.DBA_SEGMENTS TO OSCAN;
GRANT SELECT ON SYS.DBA_SQLSET TO OSCAN;
```

```
GRANT SELECT ON SYS.DBA_SQLSET_REFERENCES TO OSCAN;
GRANT SELECT ON SYS.DBA_SQL_PROFILES TO OSCAN;
GRANT SELECT ON SYS.DBA_TABLES TO OSCAN;
GRANT SELECT ON SYS.DBA_TABLESPACES TO OSCAN;
GRANT SELECT ON SYS.DBA_TAB_COLUMNS TO OSCAN;
GRANT SELECT ON SYS.DBA_TAB_PARTITIONS TO OSCAN;
GRANT SELECT ON SYS.DBA_TAB_SUBPARTITIONS TO OSCAN;
GRANT SELECT ON SYS.DBA_USERS TO OSCAN;
GRANT SELECT ON SYS.DBA_WORKLOAD_CAPTURES TO OSCAN;
GRANT SELECT ON SYS.DBA_WORKLOAD_FILTERS TO OSCAN;
GRANT SELECT ON SYS.DBA_WORKLOAD_REPLAYS TO OSCAN;
GRANT SELECT ON SYS.DUAL TO OSCAN;
GRANT SELECT ON SYS.GLOBAL_NAME TO OSCAN;
GRANT SELECT ON SYS.GV_$INSTANCE TO OSCAN;
GRANT SELECT ON SYS.GV_$PARAMETER TO OSCAN;
GRANT SELECT ON SYS.REDACTION_POLICIES TO OSCAN;
GRANT SELECT ON SYS.V_$ARCHIVE_DEST_STATUS TO OSCAN;
GRANT SELECT ON SYS.V_$BLOCK_CHANGE_TRACKING TO OSCAN;
GRANT SELECT ON SYS.V_$CONTAINERS TO OSCAN;
GRANT SELECT ON SYS.V_$DATABASE TO OSCAN;
GRANT SELECT ON SYS.V_$INSTANCE TO OSCAN;
GRANT SELECT ON SYS.V_$LICENSE TO OSCAN;
GRANT SELECT ON SYS.V_$OPTION TO OSCAN;
GRANT SELECT ON SYS.V_$PARAMETER TO OSCAN;
GRANT SELECT ON SYS.V_$PDBS TO OSCAN;
GRANT SELECT ON SYS.V_$SESSION TO OSCAN;
GRANT SELECT ON SYS.V_$VERSION TO OSCAN;
GRANT SELECT ON OLAPSYS.DBA$OLAP_CUBES TO OSCAN WITH GRANT OPTION;
GRANT SELECT ON SYSTEM.PRODUCT_PRIVS TO OSCAN WITH GRANT OPTION;
```

The following example script creates the user **C##OSCAN** that needs to be created as a common user, and then adds the required Oracle permissions, which is used for container databases:

```
CREATE USER C##OSCAN IDENTIFIED BY <passwd> CONTAINER=ALL;
ALTER USER C##OSCAN SET CONTAINER_DATA=ALL CONTAINER=CURRENT;
GRANT CONNECT TO C##OSCAN CONTAINER=ALL;
GRANT CREATE SESSION TO C##OSCAN CONTAINER=ALL;
GRANT EXECUTE ON SYS.DBMSOUTPUT_LINESARRAY TO C##OSCAN CONTAINER=ALL;
GRANT EXECUTE ON SYS.DBMS_APPLICATION_INFO TO C##OSCAN CONTAINER=ALL;
GRANT EXECUTE ON SYS.DBMS_OUTPUT TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.ALL_VIEWS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.CDB_PDB_HISTORY TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_ADVISOR_TASKS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_AWS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_CPU_USAGE_STATISTICS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_CUBES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_ENCRYPTED_COLUMNS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_FEATURE_USAGE_STATISTICS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_FLASHBACK_ARCHIVE TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_FLASHBACK_ARCHIVE_TABLES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_FLASHBACK_ARCHIVE_TS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_LOBS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_LOB_PARTITIONS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_LOB_SUBPARTITIONS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_MINING_MODELS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_OBJECTS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_OBJECT_TABLES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_RECYCLEBIN TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_REGISTRY TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_SEGMENTS TO C##OSCAN CONTAINER=ALL;
```

```
GRANT SELECT ON SYS.DBA_SQLSET TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_SQLSET_REFERENCES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_SQL_PROFILES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_TABLES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_TABLESPACES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_TAB_COLUMNS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_TAB_PARTITIONS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_TAB_SUBPARTITIONS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_USERS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_WORKLOAD_CAPTURES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_WORKLOAD_FILTERS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DBA_WORKLOAD_REPLAYS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.DUAL TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.GLOBAL_NAME TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.GV_$INSTANCE TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.GV_$PARAMETER TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.REDACTION_POLICIES TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$ARCHIVE_DEST_STATUS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$BLOCK_CHANGE_TRACKING TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$CONTAINERS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$DATABASE TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$INSTANCE TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$LICENSE TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$OPTION TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$PARAMETER TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$PDBS TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$SESSION TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON SYS.V_$VERSION TO C##OSCAN CONTAINER=ALL;
GRANT SELECT ON OLAPSYS.DBA$OLAP_CUBES TO C##OSCAN WITH GRANT OPTION
CONTAINER=ALL;
GRANT SELECT ON SYSTEM.PRODUCT_PRIVS TO C##OSCAN WITH GRANT OPTION CONTAINER=ALL;
```

**NOTE**
These privileges allow the Snow Inventory Oracle Scanner to read metadata about database objects, but not the data inside them. They are needed in order to determine the license needs, if there for example exist partitioned or compressed tables.

## 2.3 DATABASE VAULT

When Database Vault is enabled, then SYS or the specific Oracle user must:

- have **PARTICIPANT** or **OWNER** authorization on 'Oracle Data Dictionary" realm
- have **PARTICIPANT** or **OWNER** authorization on 'Oracle Database Vault' realm
- be granted the **DV_SECANALYST** role for querying Oracle Database Vault-supplied views
- be granted **SELECT** on **LBAC$POLT** from **LBACSYS**

### 2.3.1 EXAMPLE: DATABASE VAULT

Example using **OSCAN**.

Log into the database instance as a user who has been granted the **DV_OWNER** or **DV_ADMIN** role.

```
sqlplus /nolog
conn DV_OWNER/<password>
```

```
grant DV_SECANALYST to OSCAN;
exec DVSYS.DBMS_MACADM.ADD_AUTH_TO_REALM('Oracle Data Dictionary','OSCAN');
exec DVSYS.DBMS_MACADM.ADD_AUTH_TO_REALM('Oracle Database Vault','OSCAN');
conn LBACSYS/<password>
grant select on LBACSYS.LBAC$POLT to OSCAN;
```

# 3 INSTALLATION AND CONFIGURATION

## 3.1 INSTALLATION

The Snow Inventory Oracle Scanner is integrated into all Snow Inventory Agents. When requesting a Snow Inventory Agent for the desired operating systems the customer needs to specify that the Oracle Management Option needs to be included.

## 3.2 CONFIGURATION

Configuration of the Snow Inventory Oracle Scanner is performed in the Snow Inventory Admin Console in Snow Management and Configuration Center.

## 3.3 SCHEDULING

Scheduling is described in the document of the specific Snow Inventory Agent that is used.

## 3.4 OUTPUT

The inventory result from the Snow Inventory Oracle Scanner will be integrated with the information collected by the Snow Inventory Agent.

## 3.5 UPGRADE

Upgrade of the Snow Inventory Oracle Scanner is performed in the Snow Inventory Admin Console in Snow Management and Configuration Center. For detailed information on how to upgrade, see *User guide, Snow Inventory Admin Console*.

# 4 KNOWN ISSUES

The Java Runtime Environment (JRE) 1.7.0_07 contains a defect with the consequence that the Oracle inventory does not work. Any server running JRE 1.7.0_07 must upgrade to a later version for the Snow Inventory Oracle scanner to work properly.

# 5 SCANNING DETAILS OF OPTIONS AND PACKS

This section provides detailed information on how the Snow Inventory Oracle Scanner identifies an Oracle database component as installed and used. The described components are the ones for which information is most frequently requested by customers.

## 5.1 GENERAL INFORMATION

The view DBA_FEATURE_USAGE_STATISTICS is used to get information on **First used** and **Last used** for an option, which is then shown in Snow License Manager (on the **Options** tab in the **Oracle database details** view).

The DBA_FEATURE_USAGE_STATISTICS is not used to trigger an option as “installed" or "used”. This means that there can be options that are not used (used=false) but still show a time value in the **Last used** field, as well as there can be options that are used (used=true) but show an empty **LastUsed** field.

**NOTE**
For some options and features there are additional requirements regarding Command-Line usage of PL/SQL packages and services, where usage is not measurable today. Instead, these have to be confirmed used or not by the customer. For more information, see documentation on Oracle Database Licensing Information.

## 5.2 MANAGEMENT PACKS

To determine if Management Packs are being used, the following actions are performed:

- Check if pack access is granted/agreed (dbconsole+EM)
- Check for EM repository
- Check for CONTROL_MANAGEMENT_PACK_ACCESS setting
- Check for DDL_LOGGING
- Check for SQL_PROFILES
- Check for SQLTUNINGSETS
- Check for SQLTUNINGADVISOR
- Check for SQLACCESSADVISOR

Information from the DBA_FEATURE_USAGE_STATISTICS view is also gathered as additional details.

## 5.3 OPTION: ADVANCED COMPRESSION

To determine if the **Advanced compression** option is being used, the following items are checked for usage:

- Advanced Row Compression*
- Advanced LOB Compression*
- Advanced LOB Deduplication*
- Data Guard Redo Transport Compression
- Data Pump Data Compression
- RMAN Backup Compression
- Heat Map
- Optimization for Flashback Data Archive History Tables

*) SecureFiles (Compression and Deduplication) is checked by looking at the DBA_LOBS, DBA_LOB_PARTIONS, and DBA_LOB_SUBPARTIONS views. If objects are found that don't belong to SYSTEM, SYS, or SYSMAN they will be triggered as used (used=true).

If any of the items above return that they have been used, we will consider that the **Advanced compression** option has been activated. However, the first three items will not be considered as used if only system users are using them.