Performance Management Advisors Deployment Guide

Advisors Database Deployment with Data Encryption

12/14/2025

Contents

Advisors Database Deployment with Data Encryption

This page describes Advisors deployment with data encryption for MS SQL and Oracle database servers.

Secure Deployment for MS SQL Server 2008

For MS SQL Server data encryption, Genesys recommends using MS SQL Server Transparent Data Encryption (TDE), which performs a real-time I/O encryption and decryption of the data and log files. This method has only a minor impact on performance, which is critical for the Advisors Suite.

It is important to mention that TDE is available only for MS SQL Server Enterprise edition. The data cannot be encrypted using TDE if any other MS SQL Server edition is used.

Advisors Suite MS SQL databases do not have any properties, such as READ-ONLY file groups, full text indexes, or filestreams that would prevent the TDE. Users must follow the standard Microsoft documentation related to this topic.

The Advisors Suite does not support MS SQL Server cell-level encryption.

Database Deployment with Data Encryption for Oracle

Oracle offers:

- Transparent Database Encryption (TDE) introduced in Oracle 10g, which allows the encryption of individual column content on the data file level.
- Tablespace encryption introduced in Oracle 11g, which allows the encryption of the entire content of a tablespace.

Genesys recommends TDE for Oracle tablespaces.

Initial Platform, Metric Graphing, and Genesys Adapter Metrics database scripts contain tablespace names in the form of variables in each create SQL statement for tables, primary keys, and indexes. The tables and indexes are distributed among several groupings based on Genesys' recommendations related to the data update patterns and its usage characteristics.

The Platform deployment script replaces the variables dynamically with the values you provide in the deployment script dialog. The deployment script generates a new `runObjCre.sql` script with the substituted variables. The deployment script executes `runObjCre.sql` and other SQL scripts in a certain order.

It is important to make a decision about what objects need encryption and what objects should go to

what tablespace before the deployment script execution.

If you decide to place all objects into one single encrypted tablespace, specify the tablespace as a user default data tablespace, and then read the script dialog prompts to ensure this tablespace is used for all objects (that is, on all prompts, specify the name of this tablespace, or simply press `Enter`). If you want to use different encrypted tablespaces for different groups of objects predefined in the scripts, you must specify the tablespace names you have chosen for this purpose on the corresponding prompts. Review the `Readme.txt` file supplied with the scripts to find out how the objects are grouped in the scripts.

Advanced Tablespace Customization

If a more granular customization is necessary (for instance, changing the table/index grouping or encrypting the data at the column level), you have an advanced option. Do not use this option if you do not have a clear understanding of the script's internals.

You will need to perform the following steps:

1. Run the deployment script from SQL*plus to generate `runObjCre.sql`.

2. Drop the previously-created user.

3. Customize the generated `runObjCre.sql` script.

4. Save it, and then execute the scripts in the following order:

    a. Run `runUsrCre.sql`

    b. Run `runObjCre.sql`

    c. Run all of the scripts listed in the bottom of the corresponding `<xxx>-<version>_Schema.sql` file after an object creation script.

For example, if you want to customize tablespaces for the Platform schema, you need to determine the additional script set and the script order by opening the `advisors-platform-<version>_Schema.sql` file of the corresponding version. The content will be different in different versions. As an example of a versioned script, let's take `advisors-platform-8.5.202.09_Schema.sql`. The bottom of the file will contain the following:

```
spool off
spool runObjCre.log
@@runObjCre.sql
@@advisors-platform-8.5.202.09_ROUTINE1.sql
@@advisors-platform-8.5.202.09_PIMPORT_J.sql
@@advisors-platform-8.5.202.09_ROUTINE2.sql
@@advisors-platform-8.5.202.09_FA_ROUTINE.sql
@@advisors-platform-8.5.202.09_INIT_DATA.sql
@@advisors-platform-8.5.202.09_CUSTOM_ROUTINE.sql
@@advisors-platform-8.5.202.09_CREATE_JOBS.sql
exec spCompileInvalid();
var r number;
var m varchar2(4000);
exec spAllRelatedMetrics(:r,:m);
@@advisors-platform-8.5.202.09_RECREATE_MV.sql
exec spCompileInvalid();
INSERT INTO PATCH_LOG (PATCH_LOG_ID,PATCH_DESC,PATCH_NAME,PATCH_APPLIED_USER,PATCH_APPLIED_DATE)
VALUES(SEQ_PATCH_LOG.NEXTVAL,'Advisors Platform 8.5.202.09','advisors-platform-8.5.202.09_ObjectsPlus.sql',USER,SYSDATE);
COMMIT;
/
BEGIN
   spCompileInvalid();
   dbms_output.put_line('Schema creation completed. Final compilation was successful.');
   dbms_output.put_line('Previous compilation warnings, if any, can be ignored.');
   EXCEPTION
    WHEN OTHERS THEN
     dbms_output.put_line('Schema creation completed with compilation errors.');
     dbms_output.put_line('Compile each invalid object manually to identify ');
     dbms_output.put_line('the reason of the problem.');
     dbms_output.put_line(sqlerrm);
END;
/
spool off
exit
```

Based on this example, after you execute `runObjCre.sql`, you will also have to execute the following:

```
@advisors-platform-8.5.202.09_ROUTINE1.sql
@advisors-platform-8.5.202.09_PIMPORT_J.sql
@advisors-platform-8.5.202.09_ROUTINE2.sql
@advisors-platform-8.5.202.09_FA_ROUTINE.sql
```

```
@advisors-platform-8.5.202.09_INIT_DATA.sql
@advisors-platform-8.5.202.09_CUSTOM_ROUTINE.sql
@advisors-platform-8.5.202.09_CREATE_JOBS.sql
exec spCompileInvalid();
var r number;
var m varchar2(4000);
exec spAllRelatedMetrics(:r,:m);
@advisors-platform-8.5.202.09_RECREATE_MV.sql
exec spCompileInvalid();
```

To verify that an Advisors schema is secured with TDE encryption, do the following:

1. Run the following query to verify that the tablespaces to which the user has access are created as encrypted:
   ```
   SELECT TABLESPACE_NAME,ENCRYPTED FROM USER_TABLESPACES WHERE ENCRYPTED='YES';
   ```

2. Run the following query to verify that all of the tables are created in the encrypted tablespaces:
   ```
   SELECT DISTINCT TABLESPACE_NAME FROM USER_TABLES;
   ```

List of Function-Based Indexes

TDE limitations related to the column-based encryption of the content with function-based indexes are applicable to the Advisors Suite. The Advisors schema contains a number of function-based indexes that need to be modified or dropped if the column-based encryption of the related columns is chosen. See the following Table.

Platform Schema

Index	Table	Column expression
IX_APPLICATION_NAME	APPLICATION – Contains application group metadata	UPPER("NAME")
IX_CALL_APP_UP	CALL_APPLICATION – Contains metadata for queues, call types, services, interaction queues	UPPER("NAME")
IX_CALL_CENTER_NAME	CALL_CENTER – Contains contact center metadata	UPPER("NAME")
IX_CALL_CREGION_NAME	REGIONS – Contains metadata for geographic regions, reporting regions and operating units	UPPER("NAME"), UPPER("TYPE")

Index	Table	Column expression
`IX_CG_UP`	CONTACT_GROUP – Contains metadata for workforce contact groups	`UPPER("NAME")`
`IX_CG_ORIGIN`	`CONTACT_GROUP`	`UPPER("WFM_EQUIVALENT_ID"),` `UPPER("SOURCE_SYSTEM")`
`IX_CONTACT`	CONTACT – Contains Advisors users contact data	`UPPER("EMAIL")`
`IX_PG_NAME`	PG – Contains metadata for peripheral gateways	`UPPER("PG_NAME")`
`IX_USERS_USERNAME`	USERS – Contains the list of Advisor users	`UPPER("USERNAME")`
`IX_KEY_ACTION_NAME`	`KEY_ACTION`	`UPPER("NAME")`
`IX_ADAPTER_INST_HOST_PORT`	`ADAPTER_INSTANCES`	`UPPER("HOST")`