Performance Management Advisors Deployment Guide

Least Privileges: How to Configure Advisors Database Accounts with Minimal Privileges

5/7/2025

Contents

Least Privileges: How to Configure Advisors Database Accounts with Minimal Privileges

In the general Advisors installation scenario with Oracle, the Oracle schema owners are also used by Advisors components to access the database during runtime.

Starting with Advisors release 8.5.202, access to an Oracle database 12c R2 can be configured in such a way that the Advisors components access the database through low-privileged, runtime users that are not schema owners. The runtime users are granted only DML privileges and the privileges to execute a selected list of stored procedures that operate only within the Advisors database environment.

Advisors installations with MS SQL Server could be configured to access the database through low-privileged runtime users in previous releases. In release 8.5.2, a dedicated security procedure has been added to help further restrict the privileges and allow access only to a minimum set of necessary stored procedures and functions, rather than access to all.

There can be different acceptable scenarios for configuring database accounts with reduced privileges to achieve the same goal. However, this page contains only recommended scenarios that were tested and have passed the evaluation.

This page describes how to configure users with least privileges, which can be used by Advisors components during runtime. You must set up the runtime users before you run the Advisors installation wizards.

The procedures on this page are divided by RDBMS type:

- Microsoft SQL Server
- Oracle

Microsoft SQL Server

This section includes information about the Advisors database users, and the privileges associated with each, for the following setup and installation tasks:

- Creating the Advisors Databases
- Creating the Database Objects
- Creating the Runtime User
- Running the Bulk Configuration Tool
- Running the Advisors Installation Wizards

Creating the Advisors Databases

You require one privileged database user. That user sets up all three Advisors databases. The privileged user requires privileges to create a database, create a login account, create a user, and to back up the database.

Creating Database Objects

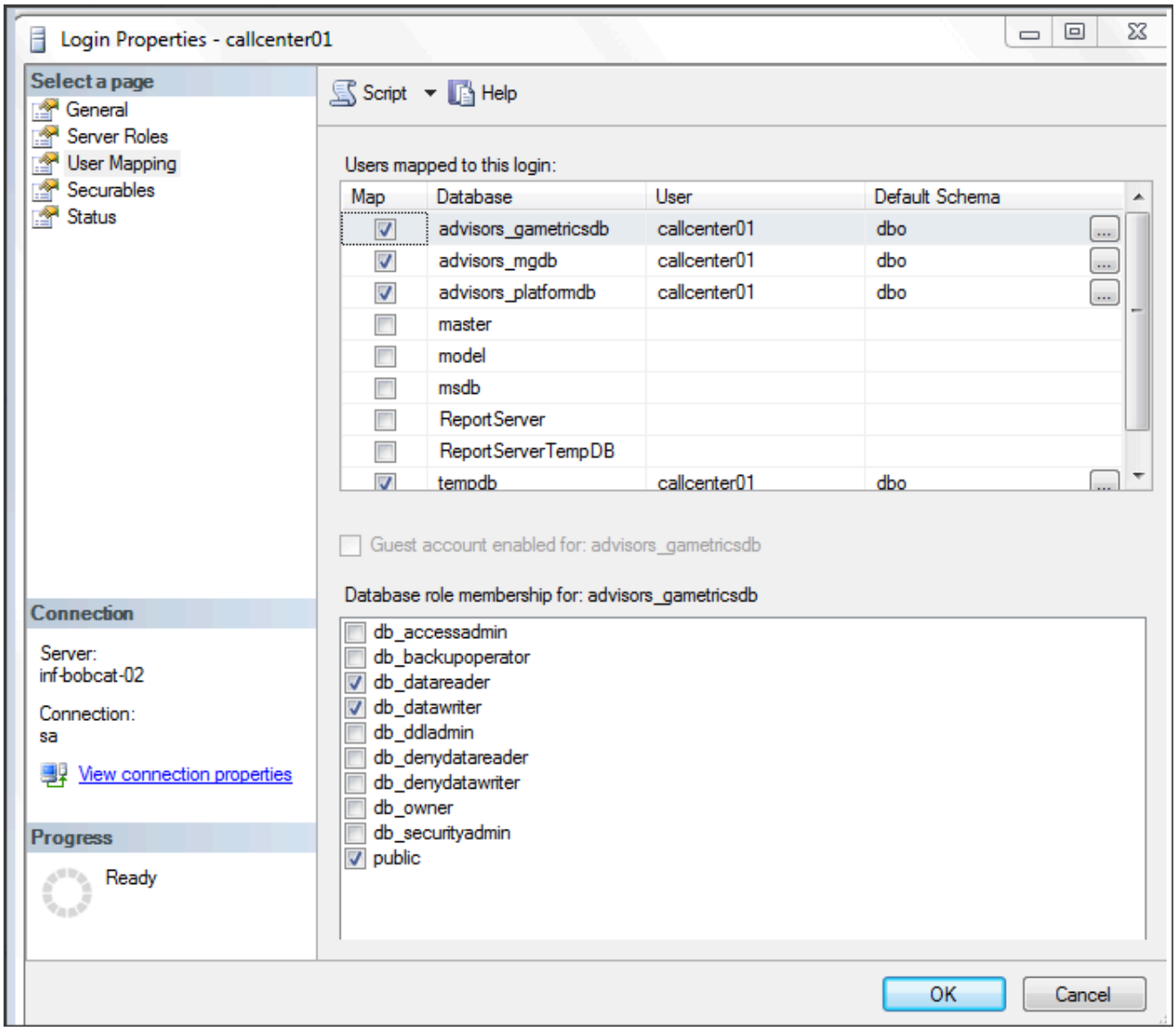
The database owner (db owner) of all three Advisors databases creates the database objects in each database. The db owner executes the "new-database" SQL scripts, which are provided in the Advisors Installation Package (IP) for each database:

- Advisors Genesys Adapter (AGA) metrics database creation script is located in the Advisors Genesys Adapter IP.
- Advisors Platform database creation script is located in the Advisors Platform IP.
- Advisors metric graphing database creation script is located in the Advisors Platform IP starting with Advisors release 8.5.202, and in the Contact Center Advisor/Workforce Advisor IP in earlier releases.

Creating the Runtime User

You must make the low-privileged user, to be used during Advisors application runtime, a member of the [db_datareader] and [db_datawriter] roles in each of the three Advisors databases. The low-privileged user account must have a default schema that holds all objects within each database.

For example, let's say that three databases are created by the "sa" user during the database creation stage. The "sa" user creates a "callcenter01" user login account, which is mapped to each of the three databases and is assigned a default schema, "dbo".



Once the user is added, the db owner must execute the spGrantExecute procedure, located in each of the three Advisors databases. The spGrantExecute procedure has the same name in each database, but has different content depending on the database that holds it. For example:

- AGA metrics database:

```
USE [advisors_gametricsdb]
 GO

 EXEC [dbo].[spGrantExecute]
 @UserName = N'callcenter01'

 GO
```

- Advisors metric graphing database:

```
USE [advisors_mgdb]

GO

EXEC [dbo].[spGrantExecute]
@UserName = N'callcenter01'

GO
```

- Advisors Platform database:

```
USE [advisors_platformdb]

GO

EXEC [dbo].[spGrantExecute]
@UserName = N'callcenter01'

GO
```

It is possible to set up a separate "data reader/data writer" user for each database. However, in that case, the Platform user must also be made a data reader in the Advisors metrics database, or, at a minimum, must be granted a `select` permission on all views contained in the AGA metrics database. A corresponding database user name must be provided in the spGrantExecute procedure and in the Advisors installation wizard prompts.

If a CISCO data source is present, the Platform user must be granted permissions as described elsewhere in guide.

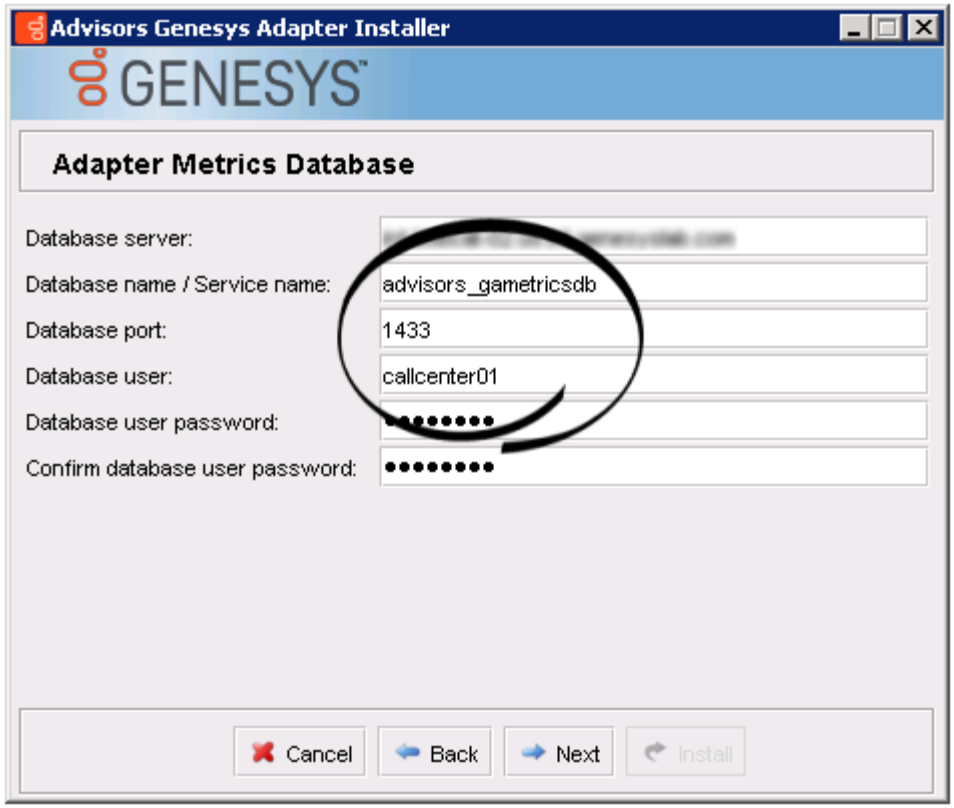
Running the Bulk Configuration Tool

The bulk configuration tool supplied with the Advisors Platform IP is used outside of the applications and is a candidate for a high-privileged user. The spGrantExecute procedure excludes the bulk configuration procedures. Genesys recommends that privileges to execute all procedures with names that start with "spBlk" be temporarily granted to a user when it is necessary to use the bulk configuration tool, and revoked once the Advisors configuration is complete and needs to be frozen. At this point, Genesys also recommends that you back up the Platform database.

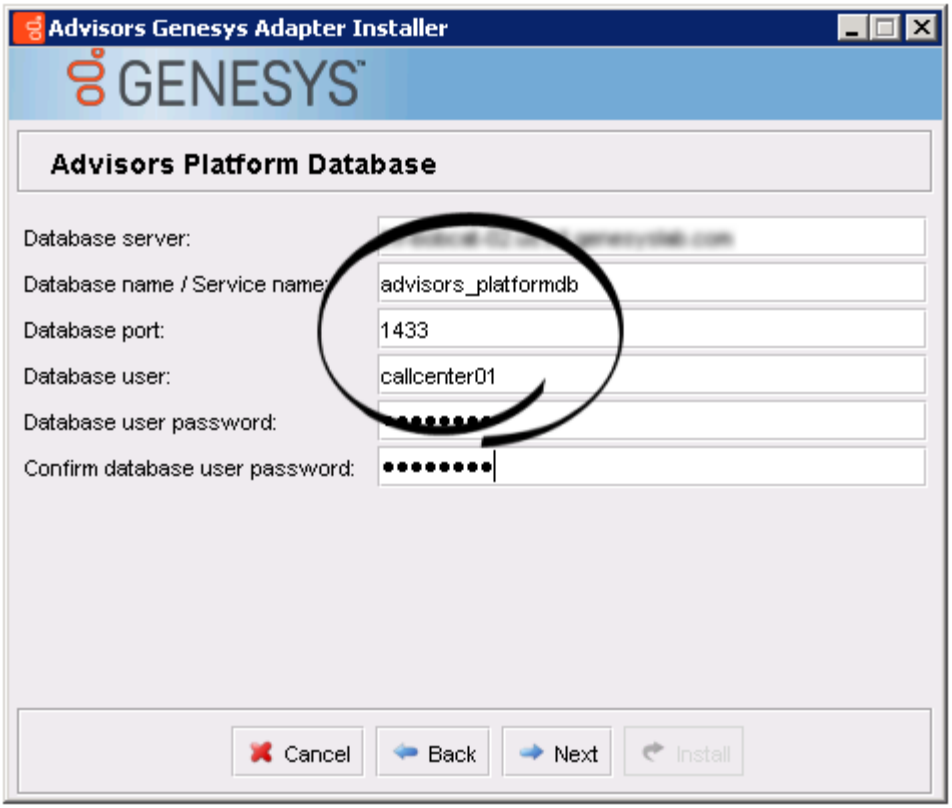
Running the Advisors Installation Wizards

Once the database setup is complete, you can run the Advisors installation wizards. Enter the runtime database user name(s) in the installation wizard prompts for each database. The following examples show the runtime user specified in all of the database user–related fields.

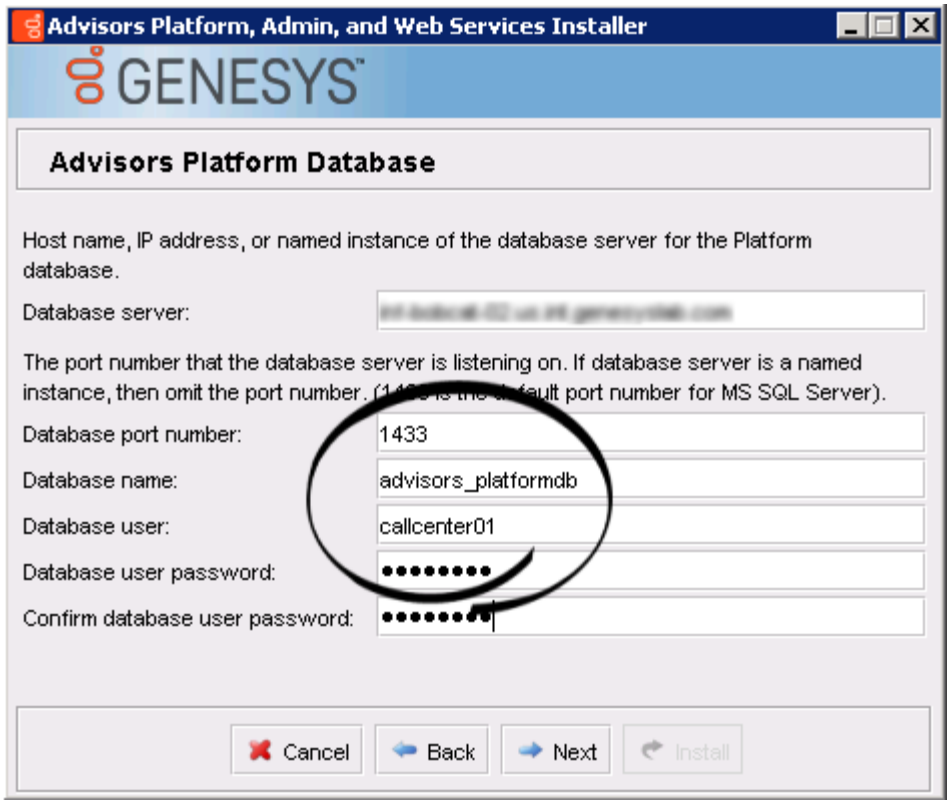
- Advisors Genesys Adapter installation wizard > AGA metrics database



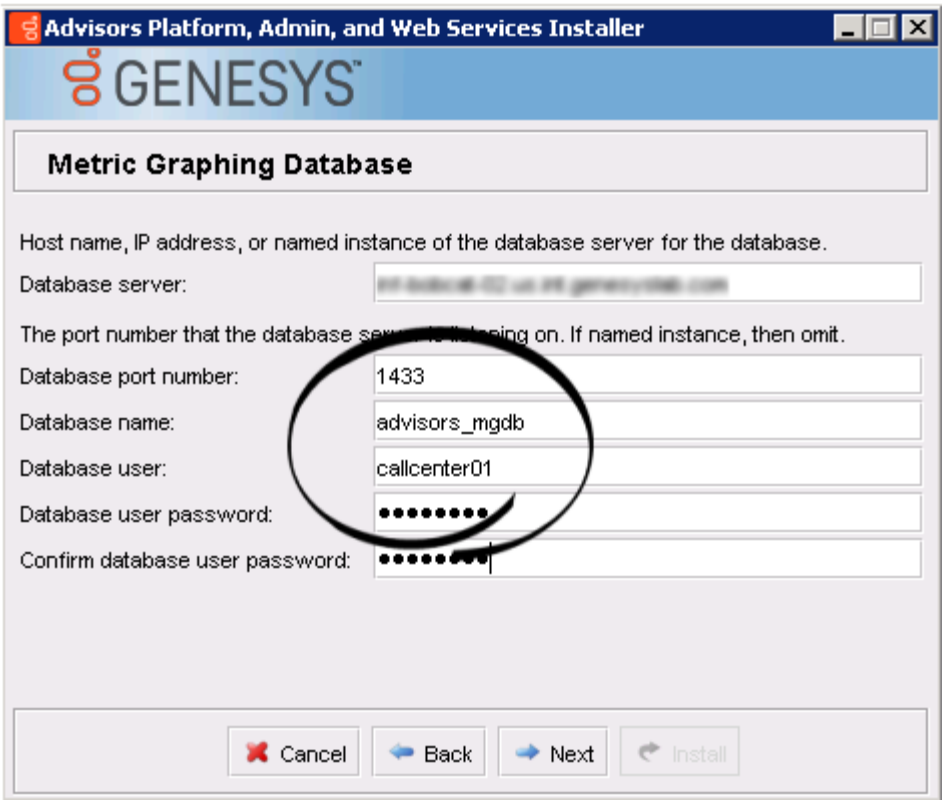
- Advisors Genesys Adapter installation wizard > Platform database



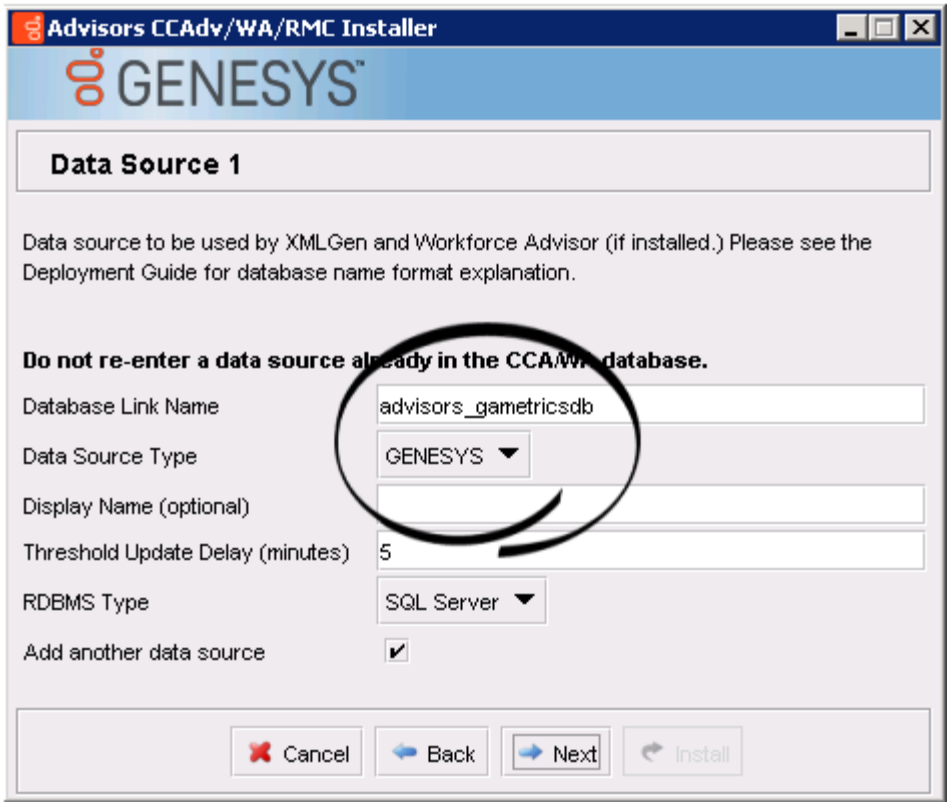
- Platform installation wizard > Platform database



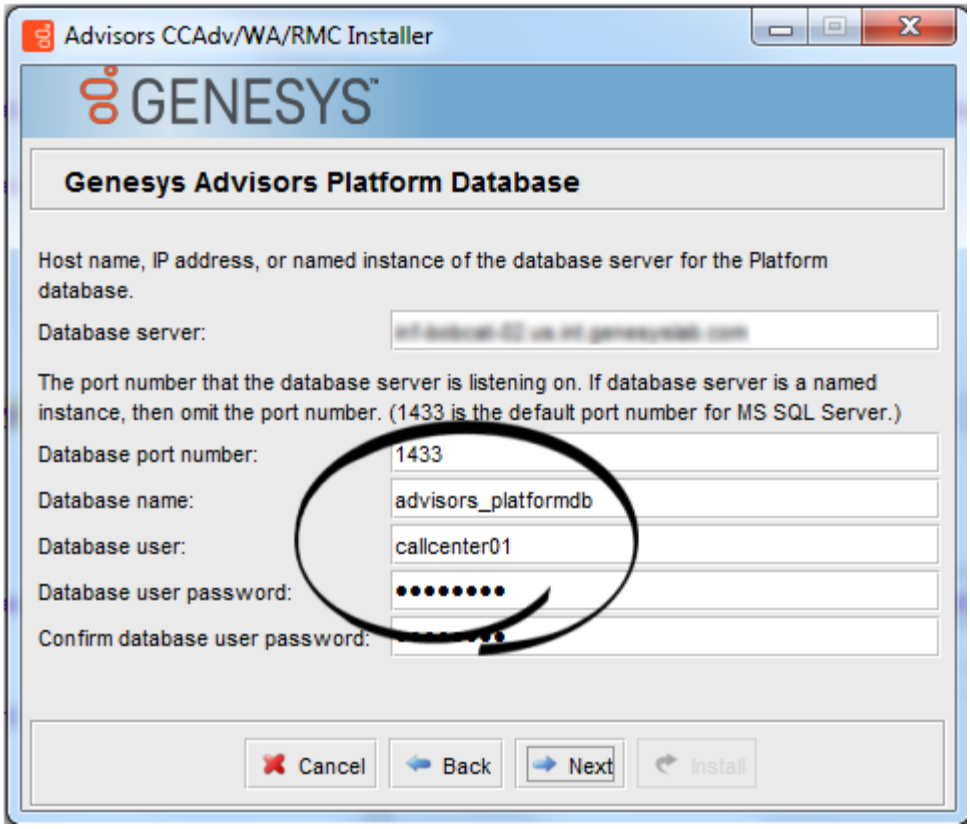
- Platform installation wizard > metric graphing database



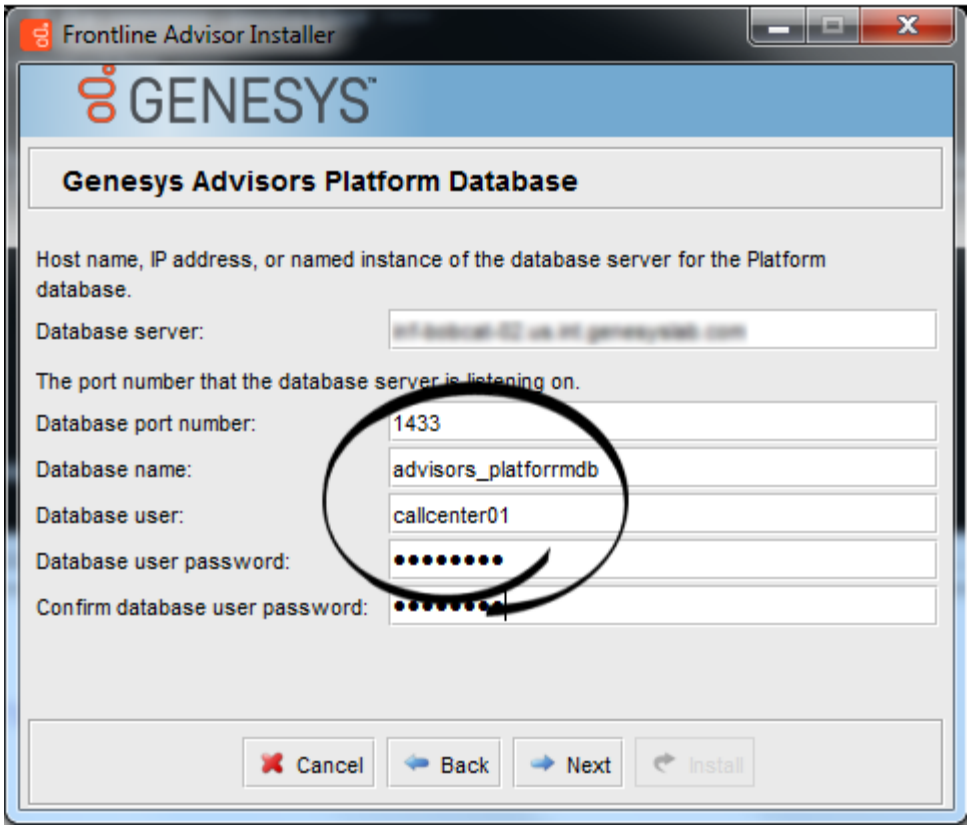
- CCAdv/WA installation wizard > Genesys data source



- CCAdv/WA installation wizard > Platform database



- Frontline Advisor installation wizard > Platform database



Oracle

This section includes the following topics:

- Prerequisites
- Creating the Runtime User
- What to do if something goes wrong
- Running the Advisors Installation Wizards
- Alternative Method to Configure Oracle Runtime Database Access
- Reusing Application and Database Roles

Prerequisites

- Use the Oracle 12c Release 2 RDBMS for your Advisors installation.
- Create three Advisors database users/schemas and the corresponding database objects using the procedures described in the base Oracle Database Installation section of this guide.

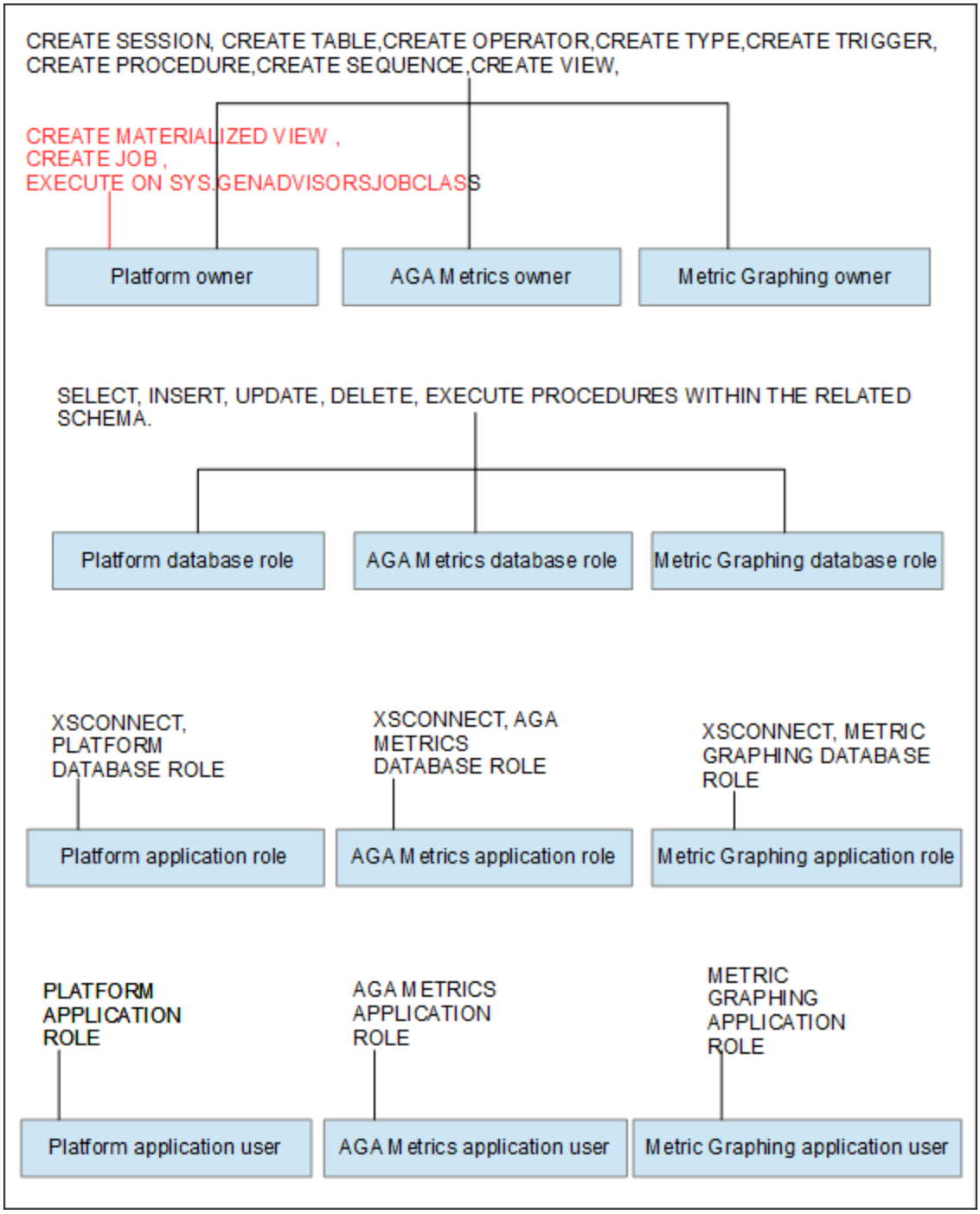
Creating the Runtime User

The solution described on this page is based on the Oracle Database Real Application Security feature and direct-login application users. Application users do not own database schemas by definition, but can create application sessions in the database. Application users can be assigned traditional database schemas owned by other users as their default schemas.

The overall procedure consists of four groups of tasks:

1. Application roles creation and direct-login application users creation with the XS_PRINCIPAL package.
2. Granting roles to direct-login application users with the XS_PRINCIPAL package.
3. Database roles creation and granting a set of restricted object-level privileges to the database roles.
4. Granting database roles to the corresponding application roles.

The following figure is a simplified schema showing the resulting schema owner privileges and the application user privileges.



The Platform schema owner might require an additional privilege if the Advisors application is installed using an Oracle database that does not have the JServer Java Virtual Machine installed.

`EXECUTE ON SYS.DBMS_LOCK` will be required in addition to the three privileges shown in red in the figure above. You must modify the `advisors-platform-<version>_UsersAndRoles.sql` script to accommodate the additional privilege.

Procedure:

Steps

1. Decide what you will use as names for the following entities:

   - The names and passwords for direct-login application users with a restricted set of privileges that Advisors components will use to access the database during runtime.
   - The names for the application roles that will be granted to the direct-login application users.
   - The names for regular database roles that will hold the restricted set of object-level privileges and that will be granted to application roles.

   You will also need to provide the names of schema owners that should have been created already, using the base database creation procedure (these are the Platform, AGA metrics, and Metric Graphing schema owners).

   For this example, we will use the following names:

   - Adv1PltOwner,Adv2AgaOwner,Adv3MgOwner as schema owners.
   - Adv1,Adv2,Adv3 as direct-login application users that will become Advisors runtime users.
   - AdvPlt_approle,AdvAga_approle,AdvMg_approle as application roles.
   - AdvPlt_dbrole,AdvAga_dbrole,AdvMg_dbrole as regular database roles.

2. Connect to SQL*Plus as a privileged user (such as "system") who has access to all three Advisors schemas. Execute the `advisors-platform-<version>_UsersAndRoles.sql` script, providing the names and passwords when prompted:

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> @advisors-platform-8.5.202.09_UsersAndRoles.sql
Platform schema owner: Adv1PltOwner
AGA Metrics schema owner: Adv2AgaOwner
MG Metrics schema owner: Adv3MgOwner
Platform runtime user name:  Adv1
AGA Metrics runtime user name:  Adv2
MG runtime user name:  Adv3
Platform application role: AdvPlt_approle
AGA Metrics application role: AdvAga_approle
MG application role: AdvMg_approle
Platform database role: AdvPlt_dbrole
AGA Metrics database role: AdvAga_dbrole
MG database role: AdvMg_dbrole
Enter value for platform_runtime_password: password1
Enter value for aga_runtime_password: password2
Enter value for mg_runtime_password: password3
```

   If you prefer, instead of executing the `advisors-platform-<version>_UsersAndRoles.sql` SQL*Plus script, you can use the Alternative Method to Configure Oracle Runtime Database Access procedure, described below. Using the alternative method, you execute the same commands that are provided in the SQL* Plus script, but in a more controlled way.

3. Once the setup is complete, use the following query to verify the direct-login application users (runtime users) that you have created:

```
SELECT * FROM DBA_XS_USERS;
```

The Name column contains the names of the direct-login application users that you created. The Schema column contains the default schema of the corresponding direct-login application user.

The user name must match the name that you planned for your runtime user. The default schema for the application user must be the name of the Platform, AGA metrics, or Metric Graphing schema that you added during the initial database creation. This will ensure that all of the database objects that the application accesses during runtime through the direct-login application user account will be pulled from the correct schema (Platform, AGA metrics, or Metric Graphing schema), while access control during runtime is restricted to the privileges assigned to the application user.

Considering the sample names used in this procedure, you should see results that are similar to the following:

NAME	SCHEMA
ADV1	ADV1PLTOWNER
ADV1	ADV1AGAOWNER
ADV1	ADV1MGOWNER

4. Verify your "direct login application user - application role" and "application role - db role" mappings using the following query:

   ```
   SELECT GRANTEE, GRANTED_ROLE FROM DBA_XS_ROLE_GRANTS ORDER BY GRANTEE;
   ```

   The application roles must be granted to the corresponding direct-login application users, while the DB roles are granted to the corresponding application roles. Considering the sample names used in this procedure, you should see results that are similar to the following:

GRANTEE	GRANTED_ROLE
ADV1	XSPUBLIC
ADV1	XSCONNECT
ADV1	ADVPLT_APPROLE
ADVPLT_APPROLE	ADVPLT_DBROLE
ADV2	XSPUBLIC
ADV2	XSCONNECT
ADV2	ADVAGA_APPROLE
ADVAGA_APPROLE	ADVAGA_DBROLE
ADV3	XSPUBLIC
ADV3	XSCONNECT
ADV3	ADVPLT_APPROLE
ADV3	ADVMG_APPROLE
ADVMG_APPROLE	ADVMG_DBROLE

What to do if something goes wrong

If it looks like something went wrong during your attempt to add the application users and the application and database roles, then you can remove those users and roles as shown in the samples below. For consistency, the following examples use the same names that were used in the preceding procedure. Removing application users and the application and database roles does not impact the initial database installation or the schema owner permissions.

Sample: Removing application users and application and database roles

```
EXEC SYS.XS_PRINCIPAL.DELETE_PRINCIPAL ('Adv1',xs_admin_util.cascade_option);
EXEC SYS.XS_PRINCIPAL.DELETE_PRINCIPAL ('Adv2',xs_admin_util.cascade_option);
EXEC SYS.XS_PRINCIPAL.DELETE_PRINCIPAL ('Adv3',xs_admin_util.cascade_option);
```

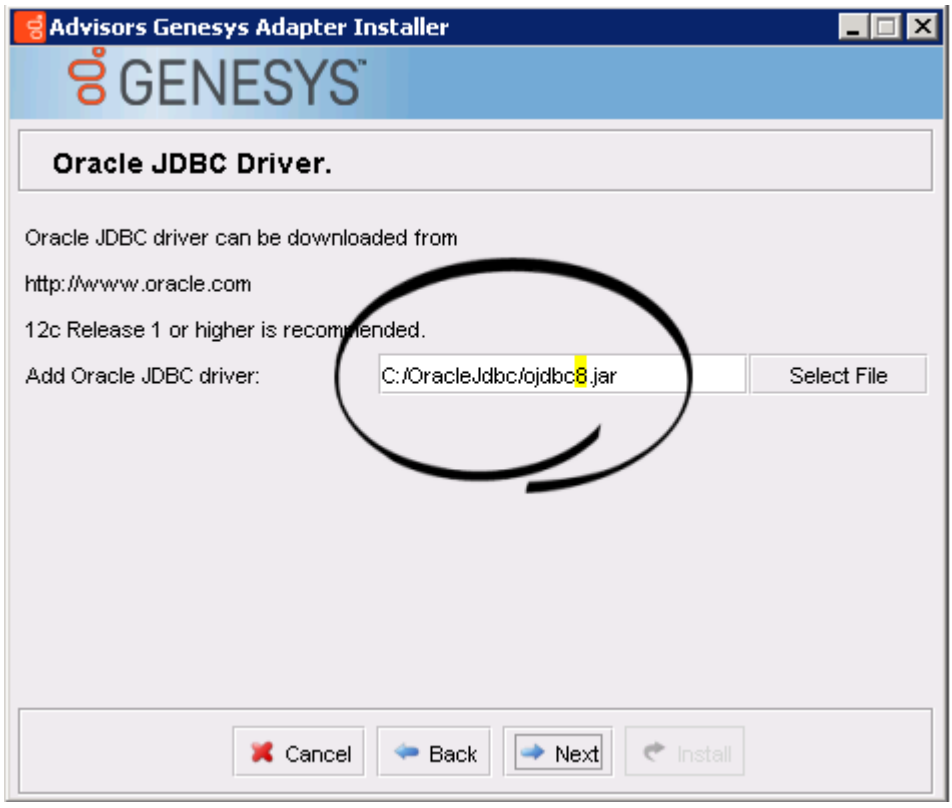
```
EXEC SYS.XS_PRINCIPAL.DELETE_PRINCIPAL('AdvPlt_role');
EXEC SYS.XS_PRINCIPAL.DELETE_PRINCIPAL('AdvAga_role');
EXEC SYS.XS_PRINCIPAL.DELETE_PRINCIPAL('AdvMg_role');

DROP ROLE AdvPlt_dbrole;
DROP ROLE AdvAga_dbrole;
DROP ROLE AdvMg_dbrole;
```

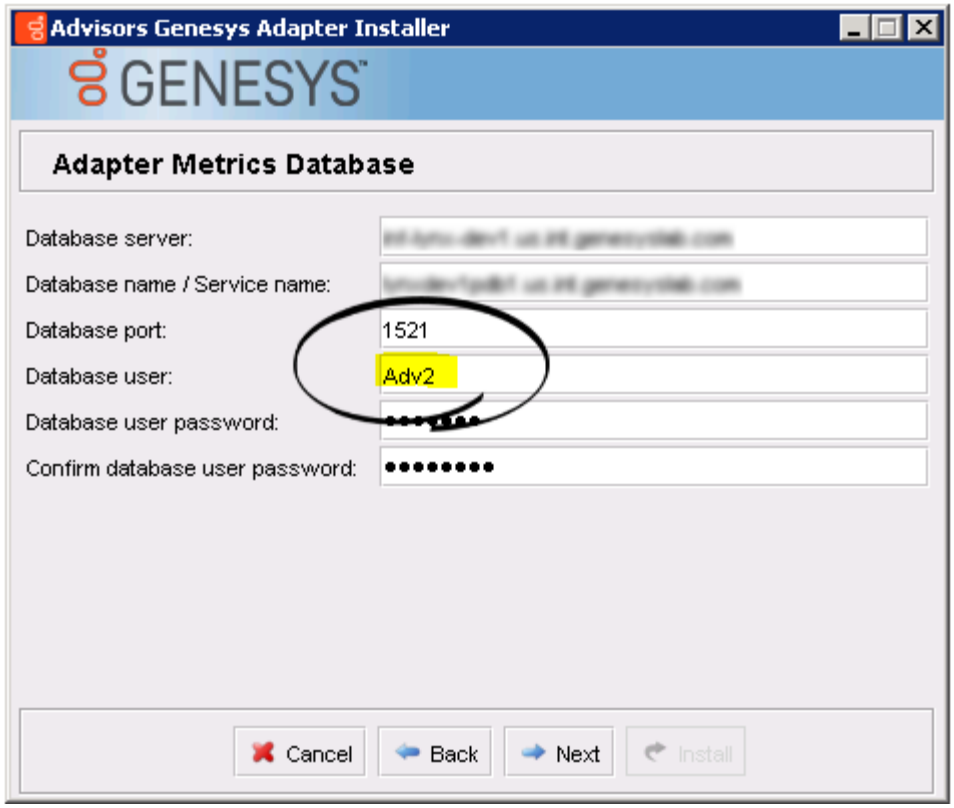
Running the Advisors Installation Wizards

Once the database setup is complete, you can run the Advisors installation wizards. Enter the runtime database user name(s) in the installation wizard prompts for each database. The following examples show the runtime user specified in all of the database user–related fields.

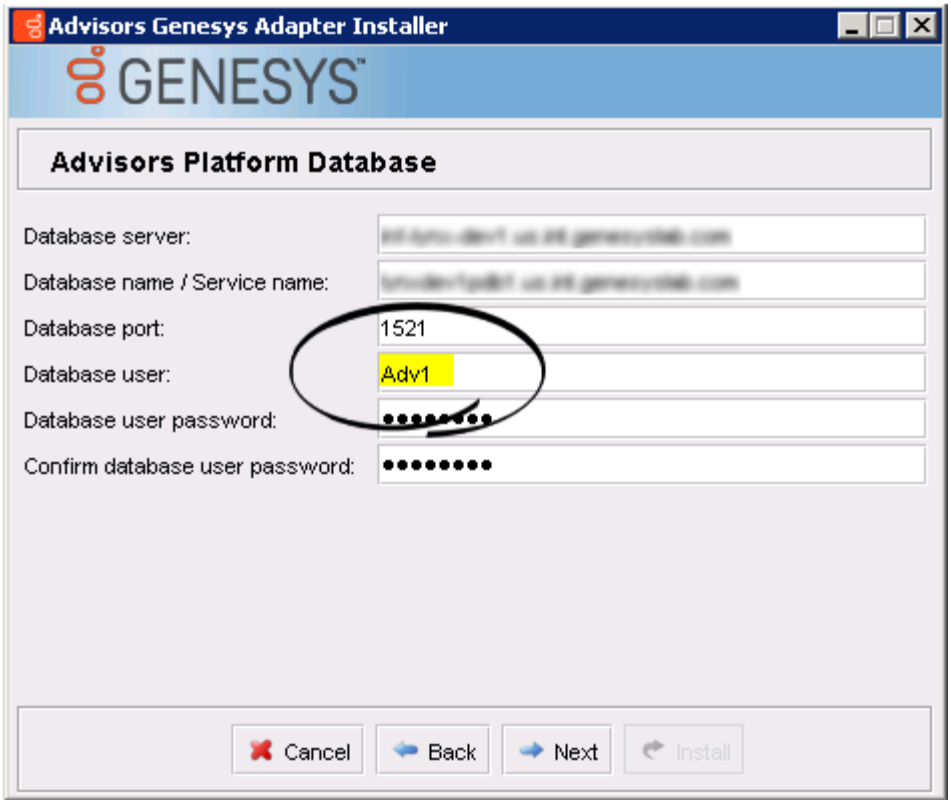
- Advisors Genesys Adapter installation wizard. Make sure you specify `ojdbc8.jar` as the Oracle JDBC driver.



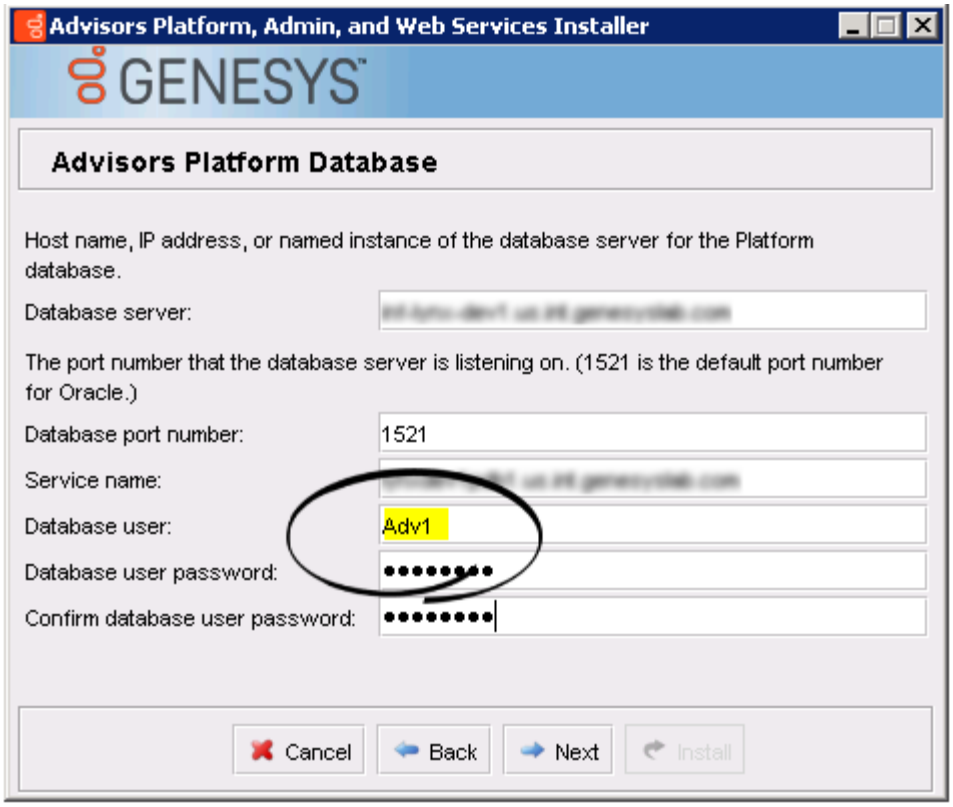
- Advisors Genesys Adapter installation wizard. The AGA runtime user is specified in the **Database user** field.



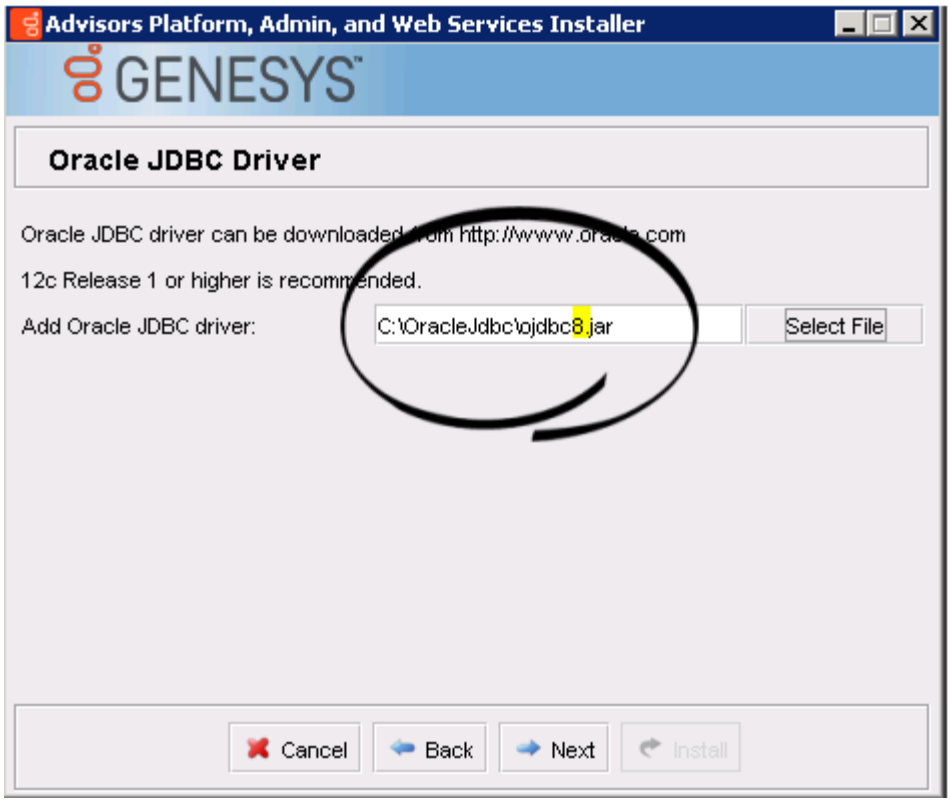
- Advisors Genesys Adapter installation wizard. The Platform runtime user is specified in the **Database user** field.



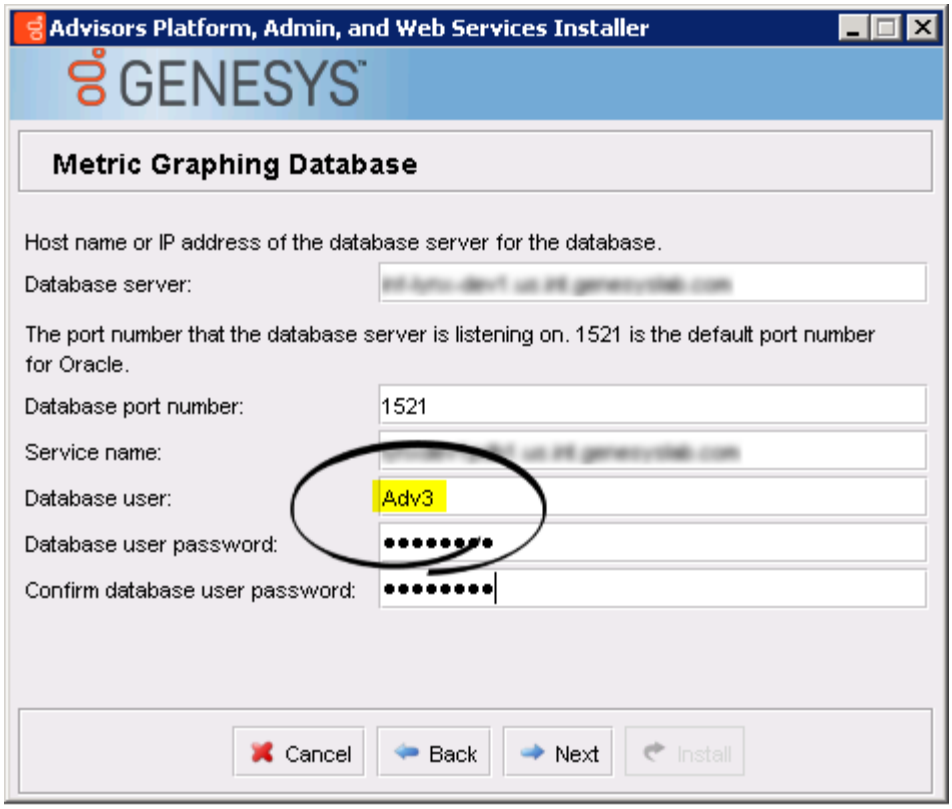
- Platform installation wizard. The Platform runtime user is specified in the **Database user** field.



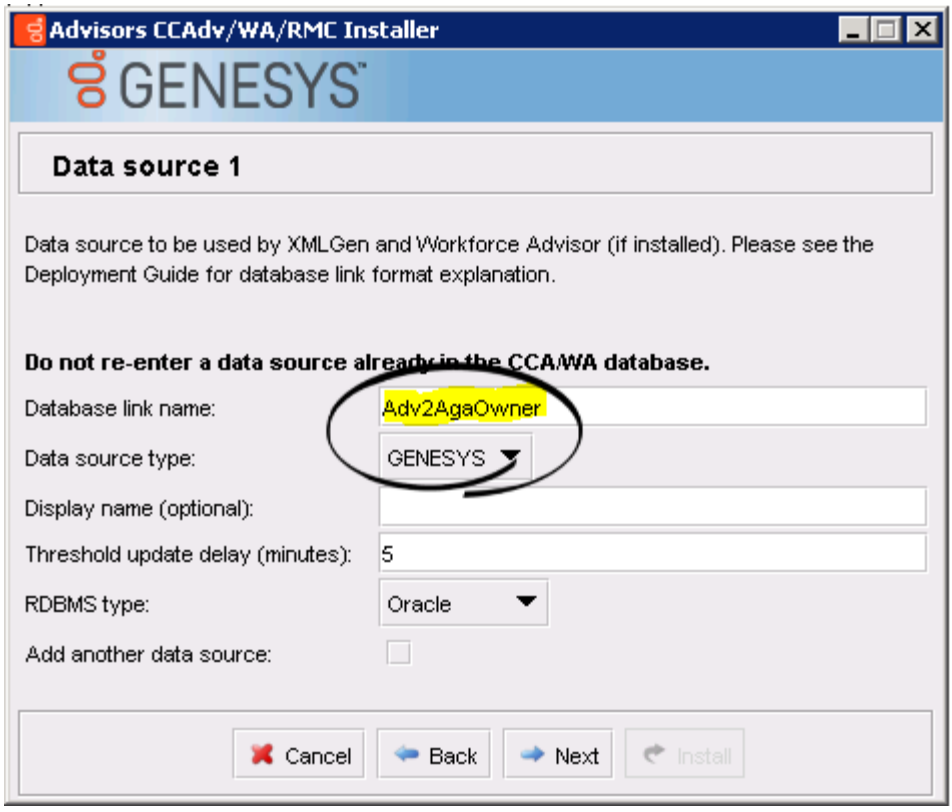
- Platform installation wizard. Make sure you specify `ojdbc8.jar` as the Oracle JDBC driver.



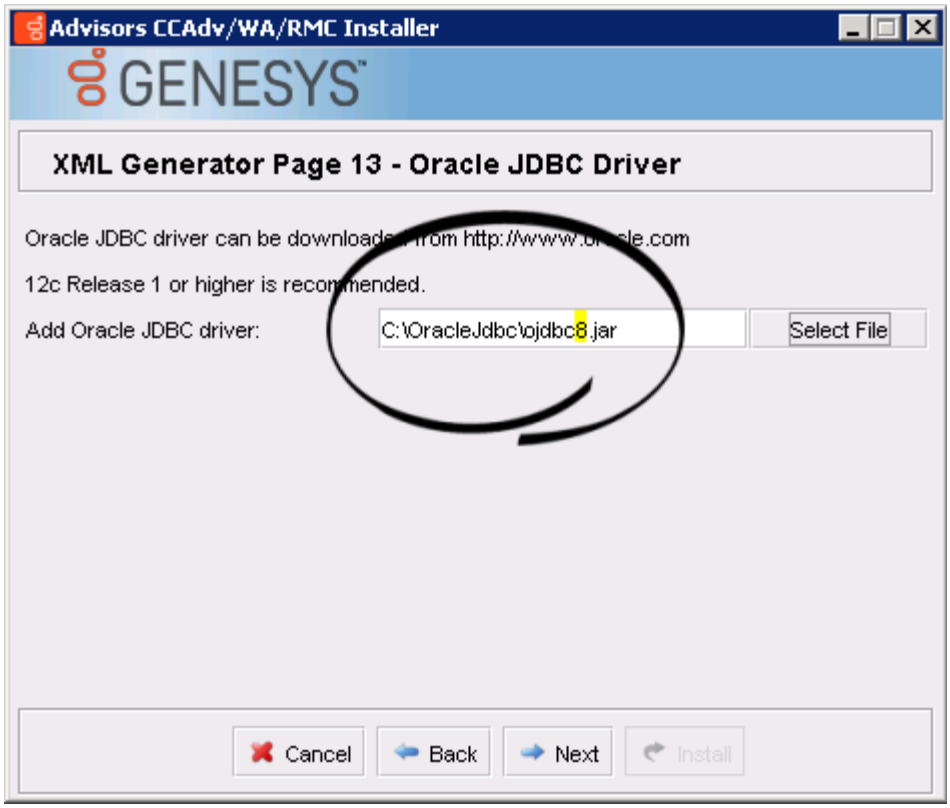
- Platform installation wizard. The metric graphing runtime user is specified in the **Database user** field.



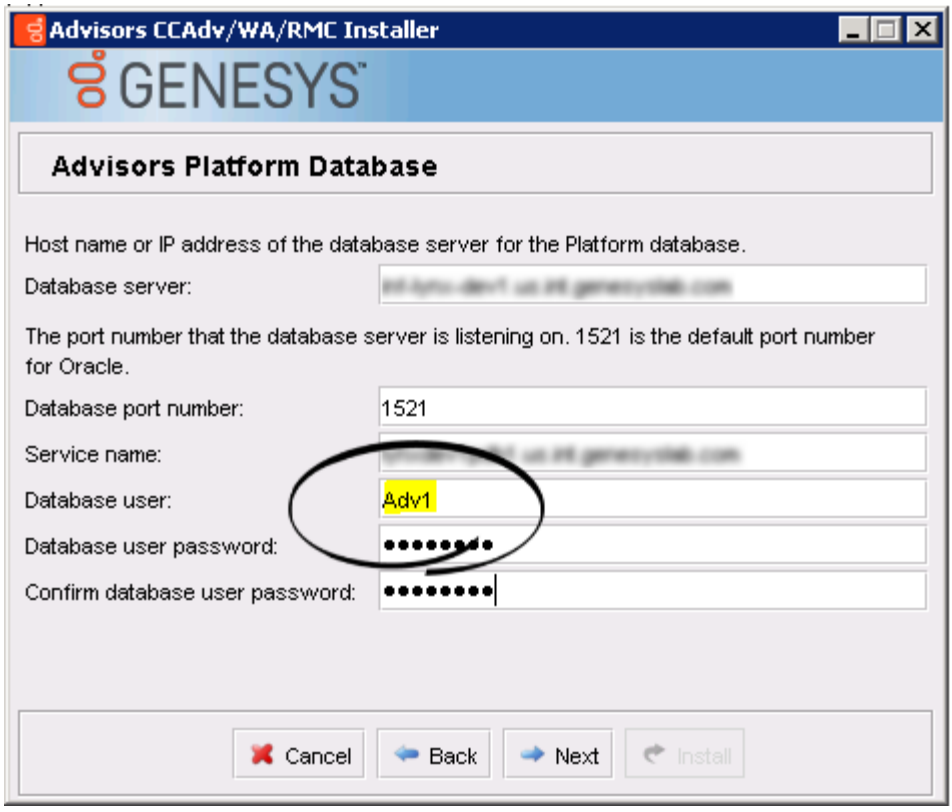
- CCAdv/WA/RMC installation wizard. The AGA schema owner is specified in the **Database link name** field.



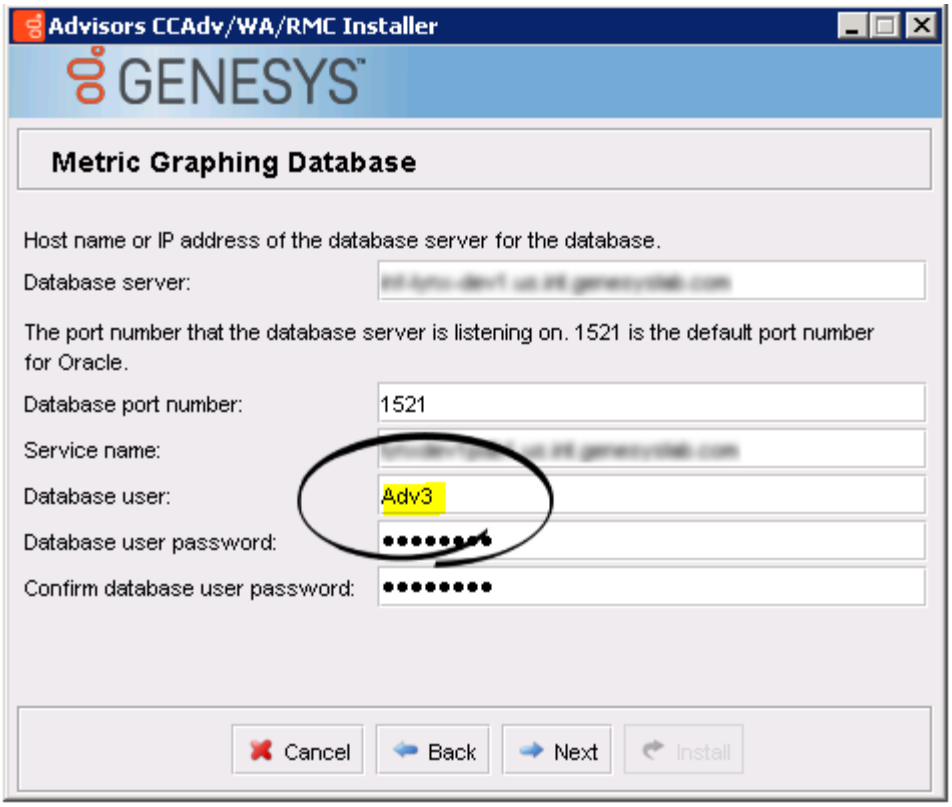
- CCAdv/WA/RMC installation wizard. Make sure you specify `ojdbc8.jar` as the Oracle JDBC driver.



- CCAdv/WA/RMC installation wizard. The Platform runtime user is specified in the **Database user** field.



- CCAdv/WA/RMC installation wizard. The metric graphing runtime user is specified in the **Database user** field.



Alternative Method to Configure Oracle Runtime Database Access

If you prefer to use a more controlled security setup, then instead of executing the SQL*Plus `advisors-platform-<version>_UsersAndRoles.sql` script as described in the procedure above, you can run the script in sections.

1. Connect to Oracle SQL Developer as a privileged user (such as system) who has access to all three Advisors schemas.
2. Copy the entire contents of the script section below and paste it into the Oracle SQL Developer query window. Highlight Section 1 and execute. Answer all 12 prompts. This will provide the substitutions for all variables contained in the next sections of the script, which you will execute later.
   If you make a mistake with the substitute variables, repeat Section 1.

   The scripts from all sections must be executed within the same session; that is, all queries must be run from the same SQL Developer window. The only exception is the object permission script that is generated in Section 3, which can be executed from any session, including your current session.
3. Once you are satisfied with the substitution, execute all of Section 2. Provide passwords, where prompted.
4. Highlight Section 3, and execute. This will generate Section 4.
5. Copy the results that were generated after you executed the Section 3 queries (that is, Section 4), and execute those as a privileged user (such as system).
6. Execute Section 5.

```
--1

SET HEADING OFF
SET LINE 512
SET FEEDBACK OFF
Accept PLATFORM_USERNAME char Prompt 'Platform schema owner: '
Accept AGA_USERNAME char Prompt 'AGA Metrics schema owner: '
Accept MG_USERNAME char Prompt 'MG Metrics schema owner: '
Accept PLATFORM_RUNTIME_USERNAME char Prompt 'Platform runtime user name: '
Accept AGA_RUNTIME_USERNAME char Prompt 'AGA Metrics runtime user name: '
Accept MG_RUNTIME_USERNAME char Prompt 'MG runtime user name: '
Accept PLATFORM_APPLICATION_ROLE char Prompt 'Platform application role: '
Accept AGA_APPLICATION_ROLE char Prompt 'AGA Metrics application role: '
Accept MG_APPLICATION_ROLE char Prompt 'MG application role: '
Accept PLATFORM_DATABASE_ROLE char Prompt 'Platform database role: '
Accept AGA_DATABASE_ROLE char Prompt 'AGA Metrics database role: '
Accept MG_DATABASE_ROLE char Prompt 'MG database role: '

--2

SET VERIFY OFF;
EXEC SYS.XS_PRINCIPAL.CREATE_USER (name => '&&PLATFORM_RUNTIME_USERNAME', schema => '&&PLATFORM_USERNAME');
EXEC SYS.XS_PRINCIPAL.CREATE_USER (name => '&&AGA_RUNTIME_USERNAME', schema => '&&AGA_USERNAME');
EXEC SYS.XS_PRINCIPAL.CREATE_USER (name => '&&MG_RUNTIME_USERNAME', schema => '&&MG_USERNAME');

EXEC SYS.XS_PRINCIPAL.SET_PASSWORD('&&PLATFORM_RUNTIME_USERNAME', '&&PLATFORM_RUNTIME_password');
EXEC SYS.XS_PRINCIPAL.SET_PASSWORD('&&AGA_RUNTIME_USERNAME', '&&AGA_RUNTIME_password');
EXEC SYS.XS_PRINCIPAL.SET_PASSWORD('&&MG_RUNTIME_USERNAME', '&&MG_RUNTIME_password');

EXEC SYS.XS_PRINCIPAL.CREATE_ROLE(NAME => '&&PLATFORM_APPLICATION_ROLE', ENABLED => TRUE);
EXEC SYS.XS_PRINCIPAL.CREATE_ROLE(NAME => '&&AGA_APPLICATION_ROLE', ENABLED => TRUE);
EXEC SYS.XS_PRINCIPAL.CREATE_ROLE(NAME => '&&MG_APPLICATION_ROLE', ENABLED => TRUE);

EXEC SYS.XS_PRINCIPAL.GRANT_ROLES('&&PLATFORM_RUNTIME_USERNAME', 'XSCONNECT');
EXEC SYS.XS_PRINCIPAL.GRANT_ROLES('&&AGA_RUNTIME_USERNAME', 'XSCONNECT');
EXEC SYS.XS_PRINCIPAL.GRANT_ROLES('&&MG_RUNTIME_USERNAME', 'XSCONNECT');

EXEC SYS.XS_PRINCIPAL.GRANT_ROLES('&&PLATFORM_RUNTIME_USERNAME', '&&PLATFORM_APPLICATION_ROLE');
EXEC SYS.XS_PRINCIPAL.GRANT_ROLES('&&AGA_RUNTIME_USERNAME', '&&AGA_APPLICATION_ROLE');
EXEC SYS.XS_PRINCIPAL.GRANT_ROLES('&&MG_RUNTIME_USERNAME', '&&MG_APPLICATION_ROLE');
EXEC SYS.XS_PRINCIPAL.GRANT_ROLES('&&MG_RUNTIME_USERNAME', '&&PLATFORM_APPLICATION_ROLE');
```

```
    CREATE ROLE &&PLATFORM_DATABASE_ROLE;
    CREATE ROLE &&AGA_DATABASE_ROLE;
    CREATE ROLE &&MG_DATABASE_ROLE;

    --3
    --Grant permissions to database objects

    SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON '||OWNER||'.'||TABLE_NAME||' TO &&AGA_DATABASE_ROLE;' FROM DBA_TABLES WHERE
OWNER=UPPER('&&AGA_USERNAME')
    UNION
    SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON '||OWNER||'.'||VIEW_NAME||' TO &&AGA_DATABASE_ROLE;' FROM DBA_VIEWS WHERE
OWNER=UPPER('&&AGA_USERNAME')
    UNION
    SELECT 'GRANT SELECT ON '||OWNER||'.'||VIEW_NAME||' TO &&PLATFORM_USERNAME;' FROM DBA_VIEWS WHERE OWNER=UPPER('&&AGA_USERNAME')
    UNION
    SELECT 'GRANT SELECT ON '||OWNER||'."'||VIEW_NAME||'" TO &&PLATFORM_USERNAME WITH GRANT OPTION;' FROM DBA_VIEWS WHERE
OWNER='&&AGA_USERNAME'
    UNION
    SELECT 'GRANT EXECUTE ON '||OWNER||'.'||OBJECT_NAME||' TO &&AGA_DATABASE_ROLE;' FROM DBA_PROCEDURES WHERE
OWNER=UPPER('&&AGA_USERNAME') AND OBJECT_TYPE<>'PACKAGE' AND OBJECT_TYPE<>'TYPE' AND OBJECT_TYPE<>'TRIGGER'
    UNION
    SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON '||OWNER||'.'||TABLE_NAME||' TO &&MG_DATABASE_ROLE;' FROM DBA_TABLES WHERE
OWNER=UPPER('&&MG_USERNAME')
    UNION
    SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON '||OWNER||'.'||VIEW_NAME||' TO &&MG_DATABASE_ROLE;' FROM DBA_VIEWS WHERE
OWNER=UPPER('&&MG_USERNAME')
    UNION
    SELECT 'GRANT EXECUTE ON '||OWNER||'.'||OBJECT_NAME||' TO &&MG_DATABASE_ROLE;' FROM DBA_PROCEDURES WHERE OWNER=UPPER('&&MG_USERNAME')
AND OBJECT_TYPE<>'PACKAGE' AND OBJECT_TYPE<>'TYPE' AND OBJECT_TYPE<>'TRIGGER'
    UNION
    SELECT 'GRANT SELECT ON '||OWNER||'.'||OBJECT_NAME||' TO &&MG_DATABASE_ROLE;' FROM DBA_OBJECTS WHERE OWNER=UPPER('&&MG_USERNAME') AND
OBJECT_TYPE='SEQUENCE'
    UNION
    SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON '||OWNER||'."'||TABLE_NAME||'" TO &&PLATFORM_DATABASE_ROLE;' FROM DBA_TABLES WHERE
OWNER=UPPER('&&PLATFORM_USERNAME')
    UNION
    SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON '||OWNER||'.'||VIEW_NAME||' TO &&PLATFORM_DATABASE_ROLE;' FROM DBA_VIEWS WHERE
OWNER=UPPER('&&PLATFORM_USERNAME')
    AND VIEW_NAME NOT IN (SELECT VIEW_NAME FROM DBA_VIEWS WHERE OWNER=UPPER('&&AGA_USERNAME')) AND VIEW_NAME NOT LIKE '%REAL_TIME%' AND
VIEW_NAME NOT LIKE '%LOGICAL_CONTROLLER%' AND VIEW_NAME NOT LIKE '%DS_SERVICE_MEMBER%'
    AND VIEW_NAME NOT LIKE 'AGENT_SKILL_GROUP_REAL_TIME%' AND VIEW_NAME NOT LIKE 'INTERACTION_QUEUE_REAL_TIME%' AND VIEW_NAME NOT LIKE
'SKILL_GROUP%' AND VIEW_NAME NOT LIKE 'CALL_TYPE%'
    AND VIEW_NAME NOT LIKE 'SERVICE%' AND VIEW_NAME NOT LIKE 'INTERACTION_QUEUE%' AND VIEW_NAME NOT LIKE 'PERIPHERAL%' AND VIEW_NAME NOT
```

```
LIKE 'CONTROLLER_TIME%'AND VIEW_NAME NOT LIKE 'QUEUE_SET%'
    UNION
    SELECT 'GRANT SELECT ON '||OWNER||'.'||VIEW_NAME||' TO &&PLATFORM_DATABASE_ROLE;' FROM DBA_VIEWS WHERE
OWNER=UPPER('&&PLATFORM_USERNAME')
    AND (VIEW_NAME IN (SELECT VIEW_NAME FROM DBA_VIEWS WHERE OWNER=UPPER('&&AGA_USERNAME')) OR VIEW_NAME LIKE '%REAL_TIME%' OR VIEW_NAME
LIKE '%LOGICAL_CONTROLLER%' OR VIEW_NAME LIKE '%DS_SERVICE_MEMBER%'
    OR VIEW_NAME LIKE 'AGENT_SKILL_GROUP_REAL_TIME%' OR VIEW_NAME LIKE 'INTERACTION_QUEUE_REAL_TIME%' OR VIEW_NAME LIKE 'SKILL_GROUP%' OR
VIEW_NAME LIKE 'CALL_TYPE%'
    OR VIEW_NAME LIKE 'SERVICE%' OR VIEW_NAME LIKE 'INTERACTION_QUEUE%' OR VIEW_NAME LIKE 'PERIPHERAL%' OR VIEW_NAME LIKE
'CONTROLLER_TIME%'OR VIEW_NAME LIKE 'QUEUE_SET%')
    UNION
    SELECT DISTINCT 'GRANT EXECUTE ON '||OWNER||'.'||OBJECT_NAME||' TO &&PLATFORM_DATABASE_ROLE;' FROM DBA_PROCEDURES WHERE
OWNER=UPPER('&&PLATFORM_USERNAME')AND OBJECT_TYPE='PACKAGE'
    UNION
    SELECT 'GRANT EXECUTE ON '||OWNER||'."'||OBJECT_NAME||'" TO &&PLATFORM_DATABASE_ROLE;' FROM DBA_PROCEDURES WHERE
OWNER=UPPER('&&PLATFORM_USERNAME') AND OBJECT_TYPE<>'PACKAGE' AND OBJECT_TYPE<>'TYPE' AND OBJECT_TYPE<>'TRIGGER'
    AND UPPER(OBJECT_NAME) NOT LIKE 'SPBLK%'
    UNION
    SELECT 'GRANT SELECT ON '||OWNER||'.'||OBJECT_NAME||' TO &&PLATFORM_DATABASE_ROLE;' FROM DBA_OBJECTS WHERE
OWNER=UPPER('&&PLATFORM_USERNAME') AND OBJECT_TYPE='SEQUENCE';

    --5

    GRANT &&PLATFORM_DATABASE_ROLE TO &&PLATFORM_APPLICATION_ROLE;
    GRANT &&AGA_DATABASE_ROLE TO &&AGA_APPLICATION_ROLE;
    GRANT &&MG_DATABASE_ROLE TO &&MG_APPLICATION_ROLE;
```

Reusing Application and Database Roles

If you plan to have several Advisors installations that will use the same Oracle database, you can reuse the roles. You can also reuse the roles in application upgrades.

If you reuse the roles, then the following part can be omitted from section 1.

```
EXEC SYS.XS_PRINCIPAL.CREATE_ROLE(NAME => '&&PLATFORM_APPLICATION_ROLE', ENABLED => TRUE);
EXEC SYS.XS_PRINCIPAL.CREATE_ROLE(NAME => '&&AGA_APPLICATION_ROLE', ENABLED => TRUE);
EXEC SYS.XS_PRINCIPAL.CREATE_ROLE(NAME => '&&MG_APPLICATION_ROLE', ENABLED => TRUE);

CREATE ROLE &&PLATFORM_DATABASE_ROLE;
CREATE ROLE &&AGA_DATABASE_ROLE;
CREATE ROLE &&MG_DATABASE_ROLE;
```