



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Performance Management Advisors Deployment Guide

Creating a SQL Server Database

12/15/2025

Contents

- 1 Creating a SQL Server Database
 - 1.1 Create the DB
 - 1.2 Create login for DB
 - 1.3 Create linked servers for the DB
 - 1.4 Create objects in the DB
 - 1.5 Migration Scripts

Creating a SQL Server Database

If, due to security restrictions, administrator or security administrator access cannot be granted, the local DBA should implement the steps described in this section.

Create the DB

1. Connect to your SQL Server instance using Microsoft SQL Server Management Studio with the LoginID assigned to the SQL Server sysadmin server role. It can be sa or any other login assigned to the sysadmin server role and created for you for temporary use during the deployment.

2. In the object explorer right-click on Databases and choose New Database. Open the General screen and configure the following properties. See the Figure that follows—Database Properties – General—as an example.

a. Specify the database name.

[+] See examples of database names.

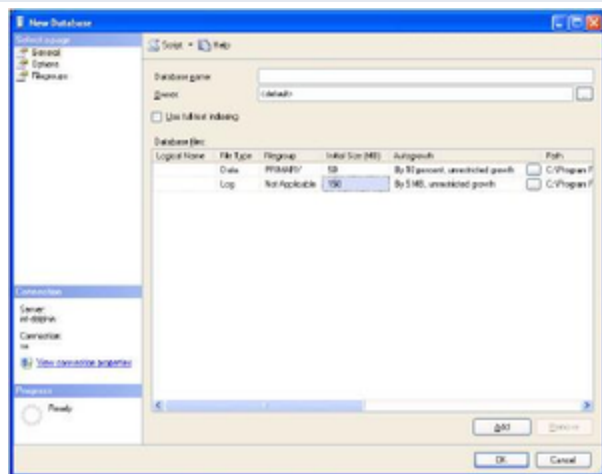
Advisors Component	Recommended DB name	Notes
Platform	advisors_platformdb	Required for Advisors implementations.
CCAdv/WA		Use the Platform and Metric Graphing databases.
FA		Uses the Platform database.
Metric Graphing	advisors_mgdb	Metric Graphing database. Required to run the CCAdv/WA dashboards and CCAdv XML Generator.
Advisors Genesys Adapter	advisors_gametricsdb	AGA metrics database. Used by AGA to transfer Genesys configuration and statistics values to XML Generator for CCAdv/WA. This database includes a table to support calling list statistics. This database is required for CCAdv/WA and WA server installations only.

b. Leave the owner as <default>.

c. Specify 50 Mb as the initial data file size with Autogrowth set to By 10%, unrestricted file growth.

d. Specify 150 Mb as the initial log file size with Autogrowth set to By 5MB, unrestricted file growth.

e. Change the pathnames to the data and log files if necessary.



Database Properties - General

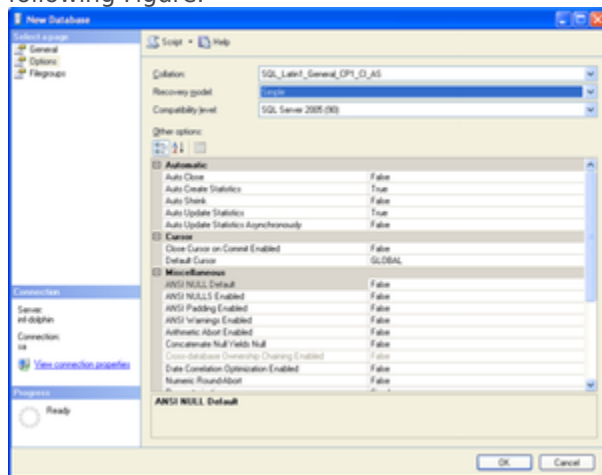
3. Open the Options screen.

- In the Collation field, select SQL_Latin1_General_CP1_CI_AS.
- In the Recovery model field, select Simple.
- Set Auto Create Statistics and Auto Update Statistics to the value true.

4. Click OK.

5. If you want to use a separate schema as a container for the database objects related to the Advisors applications, implement steps 6 and 7. Otherwise proceed to the procedure on the *Create login for Advisors* tab on this page.

6. In the Object Explorer, expand Databases, <dbname_db>, Security, and Schemas. See the following Figure.



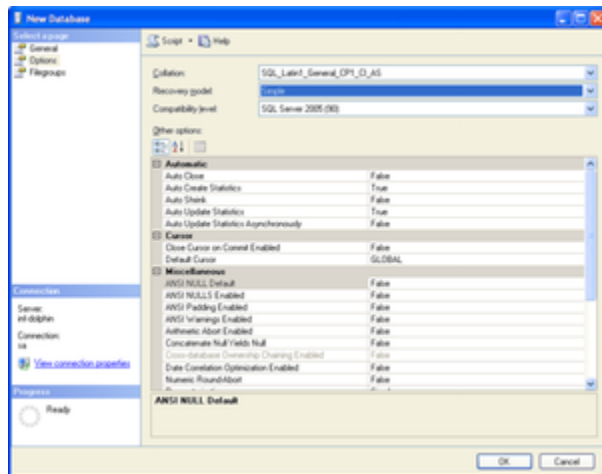
Database Properties - Options

7. Right-click on Schemas, choose New Schema, then specify the schema name. You can choose any schema name that corresponds to your company and SQL Server naming conventions; for example, callcenter01.

8. Click OK. The database is created and properties are configured.

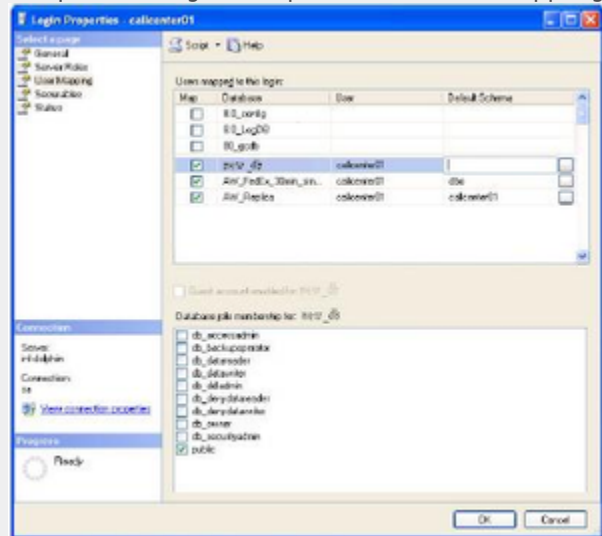
Create login for DB

1. In the Microsoft SQL Server Management Studio object explorer, select Server, and then Security.
2. Right-click Logins and choose New login. See the Figure that follows—Server-level Security.
 - a. Specify the login name (in this example, callcenter01).
 - b. Click SQL Server Authentication.
 - c. Specify a password that complies with your enterprise's security policy.
 - d. If strong passwords are part of the security policy, check the Enforce password policy check box.



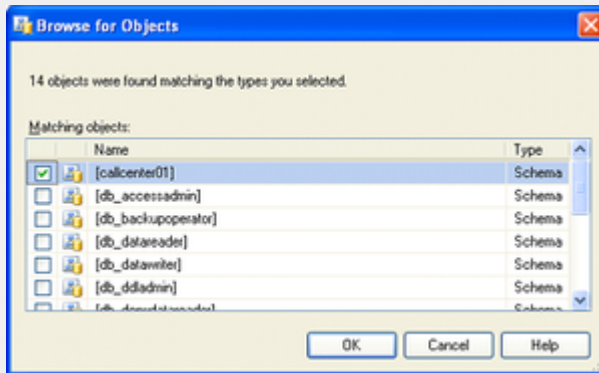
Server-level Security

3. Open the Login Properties - User Mapping screen.



Login Properties - User Mapping

- a. Map the user (callcenter01 in this example) to the newly created database by checking the appropriate check box.



Browse for Objects

- Choose dbo as a default schema if you skipped steps 5 and 6 in the procedure on the *Create the DB* tab on this page. Otherwise select the name of the created schema.
- Click OK, then confirm your selection by highlighting it and clicking OK again in the *Select Schema* dialog. This returns you to the *User Mapping* screen.
- Add the user to one or more database roles by checking the relevant check box in the lower panel of the *Login Properties – User Mapping* window. Select either:
 - The db_owner database role
 - All three of the db_datareader, db_datawriter, and db_ddladmin roles

NEW

Starting with Advisors release 8.5.202, you have an option to configure the database user with least privileges. See [Least Privileges: How to Configure Advisors Database Accounts with Minimal Privileges](#) for more information.

If you choose db_datareader, db_datawriter, db_ddladmin option, ensure that, after you create all of the database objects, you then complete the step described in the *Assigning Additional User Permissions* section on the *Create objects in the DB* tab on this page.

The login to be used by the database is now created and configured.

Create linked servers for the DB

Before you start the procedure, identify the data sources that must be accessed. If the customer uses a Cisco environment, then a linked server is necessary for each MSSQL Server used by the CCAdv/WA CISCO ICM databases. Before each linked server is configured, the CISCO ICM database administrator must create a login on each such MSSQL Server and a corresponding AWDB user linked to it. The user must have Read permission on the following AWDB views and a table:

- Agent_Skill_Group_Real_Time
- Call_Type
- Call_Type_Real_Time
- Logical_Interface_Controller
- Peripheral

- Peripheral_Real_Time
- Service
- Service_Real_Time
- Skill_Group
- Skill_Group_Real_Time
- Service_Member
- Controller_Time table

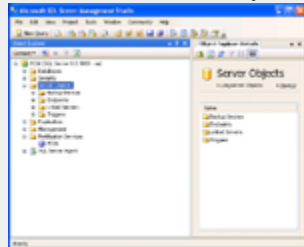
A linked server is normally not required to access the Advisors Genesys Adapter metrics database except in some uncommon cases when the Genesys Adapter metrics database and platform database reside on separate MSSQL Servers. However, each view in the Genesys Adapter metrics database must be accessible by the user defined in the Advisors Platform database. The platform user must be granted access to Genesys Adapter metrics database views that have the same names as the preceding list of CISCO ICM views. The Genesys Adapter metrics database also contains two additional views:

- Virtual_Queue_Set1_Real_Time
- Controller_Time

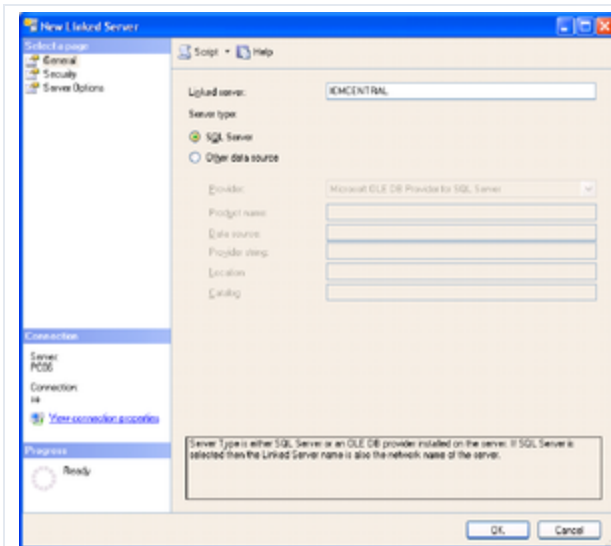
These two views must be accessible by the Platform user, also.

The user can be given the preceding object-level permissions or assigned to an equivalent user-defined database role. If your enterprise's security policy allows it, the user can be assigned to any database standard role that includes the above minimum permissions. For example, the user can be assigned to the standard `db_datareader` role.

1. In the Microsoft SQL Server Management Studio object explorer, click Server Objects.



2. Right-click on **Linked Servers** and choose **New Linked Server...** The **New Linked Servers** screen displays.



New Linked Server Screen

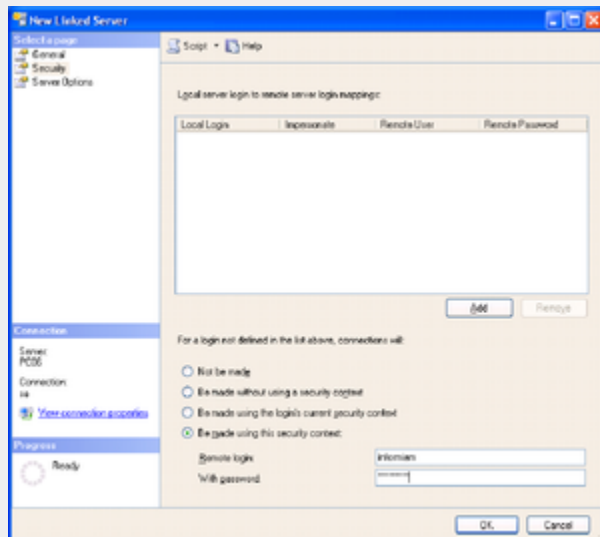
3. Under Server type, select SQL Server.

4. Specify the name of the external SQL database server to be accessed, and click OK.

The New Linked Server – Security screen displays.

5. On the Security screen:

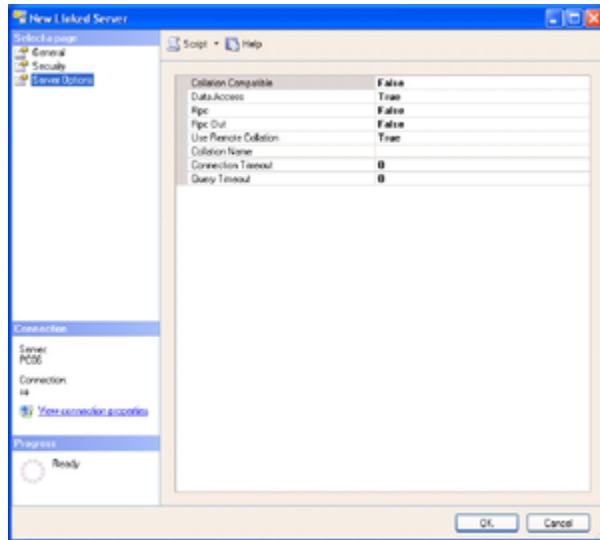
- Select Be made using this security context.
- Specify the remote login and password created by the external administrator for access to the external database.



New Linked Server – Security

6. On the Server Options screen:

- Check the Data Access check box and User Remote Collation check box.
- Click OK.



New Linked Server - Server Options

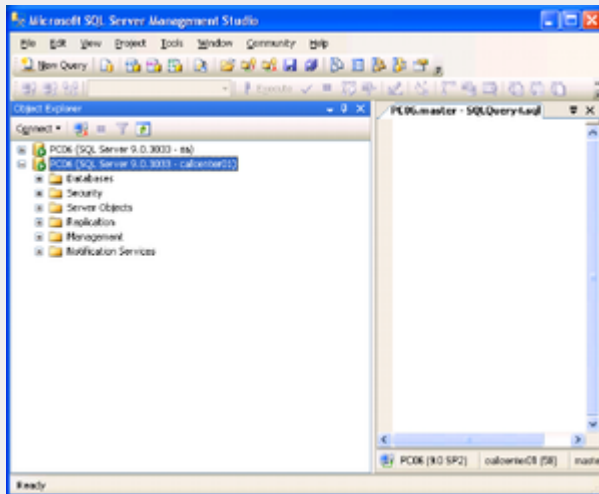
7. To test the linked server connectivity, run some SQL statements from the Microsoft SQL Server Management Studio.

- a. Enter the correct connection details and click Connect.



Connect to the Database Engine

The New Query screen displays.



Microsoft SQL Management Studio - New Query

- b. Click New Query.
- c. Type a query using the following notation:
 - `Select <...> from <Linked Server Name>.<Remote Database Name>.<Remote Database Owner>.<Remote Table Name>, or`
 - `Select <...> from openquery(<Linked Server Name>, 'select <...> from >.<Remote Database Name>.<Remote Database Owner>.<Remote Table Name> [with (<locking hint>)]`

For example, for Cisco:

`Select * from ICM_AWDB1.company_awdb.dbo.Controller_Time, or`

`Select * from OpenQuery([ICM_AWDB1], 'select * from company_awdb.dbo.Controller_Time (nolock)')`

8. For each external data source, repeat this procedure.

Create objects in the DB

This step must be run either with the system administrator account or with a user having db_owner permissions to the database. In addition, the user must have the same default schema as that assigned to the Advisors user (created in the *Create login for Advisors* tab on this page).

The db_owner role can be given temporarily to the Advisors User for the purpose of running these steps.

1. From Microsoft SQL Server Management Studio, click File. Connect to the database engine as a user meeting the criteria described above.
2. Make sure that you choose the correct database from the list of available databases.
3. From the `../sql_files` folder in the distribution folder, run the SQL script `[databasename]-new-database-<version>.sql` against the newly created database. This script creates the database user objects and populates some tables with default configuration data.

4. Scroll down the query results tab and check for errors. Ignore warnings. The objects are created.

Assigning Additional User Permissions

NEW Starting with Advisors release 8.5.202, you have an option to configure the database user with least privileges. See [Least Privileges: How to Configure Advisors Database Accounts with Minimal Privileges](#) for more information. Otherwise, complete the database user permissions setup as described below.

Assigning additional user permissions is necessary if the created database user is assigned to db_datareader, db_datawriter, and ddl_admin roles but is not assigned to the db_owner role.

The user assigned to db_datareader, db_datawriter, and ddl_admin roles must be granted execute permissions only on all user stored procedures that exist in the database after the objects are created.

You can use the SQL Server interface to assign the permissions or create a grant permissions script and execute it against the newly created database. The following statement when executed against the newly created database will produce a set of grant permission statements.

To run the script press CTRL/T, then CTRL/E.

Copy the result from the result pane. That is, click on the Result pane, and then click CTRL/A, then CTRL/C. Paste the content (CTRL/V) into the query pane and execute the following script. Before executing the script, remember to change <database user> to the ID for your database user.

```
select 'grant execute on [' + routine_catalog + '].[' + routine_schema + '].[' + routine_name + ']' to
<database user>' from
INFORMATION_SCHEMA.ROUTINES where ROUTINE_TYPE='PROCEDURE'
```

Migration Scripts

Platform database deployment/migration in MSSQL is performed by executing the platform-new-database-<version>.sql script supplied in the distribution for releases up to, and including, Release 8.1.4. Starting in Release 8.1.5, the script is labeled advisors-platform-new-database-<version>.sql. The same script can be applied to a new empty database or a database of any previous version, unless a separate migration script is supplied. If present, the separate migration script has to be used for migrations. The script name has the following pattern: advisors-platform-migrateSchema_<from version>-<to version>.sql. The script can be applied to any version starting from, and including, the <from version> up to, and including, the <to version> specified in the script name. Always check the Release Notes and "readme" files for details, specifics, and exceptions to the rules described in the documentation guides.

Migration for other databases is performed by executing migration scripts supplied in the distribution.

These follow this pattern:

<database-name>-migration-<old-version>-to-<new-version>.sql

The example below is for the FA database:

fa-database-migration-3.1-to-3.3.sql

fa-database-migration-3.3-to-8.0.sql

fa-database-migration-8.0-to-8.1.sql
fa-database-migration-8.1-to-8.1.1.sql

To migrate a database across more than one update, run the scripts in sequence from earliest to latest.