Cadmatic

CADMATIC eShare

Database Administration Guide

2026H2

docs.cadmatic.com

Copyright © 2026 by Cadmatic

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the express written permission of the publisher.

Cadmatic is a registered trademark of Cadmatic Oy. All other marks are property of their respective owners.

Cadmatic
Linnankatu 52
20100 Turku
Finland
Tel. +358 2 412 4500
www.cadmatic.com

Cadmatic

Contents

1. Introduction 5
2. Database configuration 5
2.1. Permissions 5
2.1.1. Permissions for creating eShare databases 6
2.1.2. Permissions for running the eShare server 6
2.2. Transaction logs 6
2.3. Main database configuration 7
2.3.1. Database name 7
2.3.2. Connection string 7
2.3.3. Database schema 7
2.3.4. Database size 8
2.4. Project database configuration 8
2.4.1. Database name 8
2.4.2. Connection string 8
2.4.3. Database schema 8
2.4.4. Database properties 8
2.4.5. Database size 9
3. SQL server permissions required by eShare 9
3.1. Installation 9
3.2. Configuring the database connection string 10
3.3. Required permissions 10
3.3.1. Day-to-day operation 10
3.3.2. Creating projects 11
3.3.3. Installation and version upgrades 11
4. Database administration 12
4.1. Database creation 12
4.1.1. Creating the main database 12
4.1.2. Creating an SQL script that creates the main database 13
4.1.3. Creating a project database 13
4.1.4. Creating an SQL script that creates a project database 14
4.1.5. Adding a project database to the main database 14
4.1.6. Creating an SQL script that adds a project database to the main database 15
4.2. Database upgrade 15
4.2.1. Upgrading all databases 16
4.2.2. Creating an SQL script that upgrades all databases 16

4.2.3. Upgrading the main database 17
4.2.4. Creating an SQL script that upgrades the main database 17
4.2.5. Upgrading a project database 18
4.2.6. Creating an SQL script that upgrades a project database 18
4.3. Database removal 19
4.3.1. Removing a project database 19
4.3.2. Creating an SQL script that removes a project database 20
4.3.3. Removing a project database from main database 20
4.3.4. Creating an SQL script that removes a project database from main database 21
4.4. Database permissions 21
4.4.1. Checking permissions in a database 21
4.4.2. Granting permissions to a database 22
4.4.3. Creating an SQL script that grants permissions to a database 22
5. Server backups 23
5.1. Databases to back up 23
5.2. Directories to back up 24
5.3. Restoring eShare server from backups 24

1. Introduction

Cadmatic eShare stores data in Microsoft SQL Server databases. The database architecture of eShare is such that there is one main database for running the server, and then one database for each plant or marine design project. This distributed structure makes it very efficient to access a given database and search and join data, and it allows a database administrator to easily back up and remove specific projects.

Before eShare is installed on a site, it needs to be decided whether the database login that eShare uses is to have permissions that allow it to create, upgrade, and remove databases, or whether its permissions only allow it to update data in existing databases.

We recommend eShare to use a database login that does not have administrative privileges, but this does require a more active role from the eShare database administrator. For the most part, this document describes tasks that a database administrator needs to perform if eShare is working with minimal database permissions.

If, however, your site wants to deploy eShare in the simplest possible way and provide eShare with a database login that minimizes human administrative work, we recommend that you review the information in Database configuration, but basically all you need to install the system is a database login that has administrative privileges.

For information on supported versions and editions of Microsoft SQL Server, see docs.cadmatic.com/systemrequirements.

2. Database configuration

The following topics provide information on configuring the Cadmatic eShare databases.

2.1. Permissions

If the eShare installer is to be able to create the main database, or if the eShare server is to be able to create a new project database whenever a new eShare project needs to be established, the database user that eShare logs in as must have appropriate, elevated permissions.

If you manage the databases, you can provide the eShare login with more basic permissions.

2.1.1. Permissions for creating eShare databases

Suitable server-level roles for creating and altering eShare databases include the following:

- sysadmin
- dbcreator

2.1.2. Permissions for running the eShare server

eShare server is a web server application—it responds to requests made by users, but it also runs scheduled maintenance tasks in the background. Accordingly, its application pool in the Internet Information Services (IIS) web server has to be configured to run with such credentials that allow it to access the main database and all project databases, or it needs to be using a connection string that provides such access.

Note: Installing eShare server enables IIS and adds the eShare application pool to it.

---

eShare server can read, modify, and remove data stored in the main database and in project databases. A suitable database-level role for performing these tasks would include the following:

- db_datareader + db_datawriter

eShare server might also modify the database schema, for example when creating a project, when a created project is accessed for the first time, and when starting the server after an upgrade. A suitable database-level role for performing these tasks would include the following:

- db_ddladmin

2.2. Transaction logs

Because eShare can create quite a lot of database traffic, the database transaction logs are by default set to use the Simple recovery model. This minimizes log space consumption, but on the other hand it reduces the chances of recovering from a critical failure.

If your site requires more comprehensive recovery handling or advanced mechanisms such as database mirroring, you can set the recovery model to ‘Full’.

If you do change the recovery model of a database from Simple to Full, you should periodically truncate the transaction logs or schedule transaction log backups to prevent transaction logs from filling up and causing excessive space consumption.

For information on recovery models, refer to Microsoft documentation at https://msdn.microsoft.com/en-us/library/ms189275.aspx.

2.3. Main database configuration

2.3.1. Database name

If the eShare installer creates the main database, the database name is "EShare".

If you create the main database, we recommend that you give it the default name "EShare".

2.3.2. Connection string

The connection string for accessing the main database is initially configured by providing the requested information in the eShare installation wizard. The wizard stores the connection string in the IIS server, and you can modify it afterwards if needed (for example, to change the login).

To modify the connection string, open the Connection Strings view of the eShare web site in the Internet Information Services (IIS) Manager application, and double-click the 'Main' row. You might need to restart the eShare server for the change to be applied.

If you are required to use Integrated Security (Trusted Connection) for the database connection, connection is established using the identity assigned for the application pool.

To configure the identity of the application pool, open the Application Pools view of Internet Information Services (IIS) Manager, right-click 'eShareApplicationPool' and select **Advanced Settings**, and then set Identity to the appropriate value (default: ApplicationPoolIdentity). You might need to restart the eShare server and recycle the application pool for the change to be applied.

Note: When you change the identity of the application pool, you assign the permissions of that identity to the whole eShare server; to minimize security risks, make sure that the identity does not have excessive permissions. You should also make sure that the identity is permitted to read from the eShare server folder and to read from and write to the ProgramData folder used by eShare server.

2.3.3. Database schema

The main database keeps track of projects, cross-project configurations, users, and permissions in the *dbo* schema, and that should be the default schema for the login that eShare uses. If the default

schema is not configured, *dbo* is used as the default schema.

2.3.4. Database size

The main database is usually relatively small, ranging from a couple of megabytes to a maximum of a few tens of megabytes, because it only contains data needed to administer and run the eShare server. In practice, the size depends on the number of users and projects that are being stored on the server.

2.4. Project database configuration

2.4.1. Database name

If eShare creates the project databases, the database names use the format *EShareProject_<id>*.

If you create the project databases, we recommend that you establish a naming standard that makes it easy to distinguish the project databases from other databases and also allows applying grouping for the databases via their naming. For example, if you use the format *EShare_<project name>*, the database name of project "PowerPlant" would be "EShare_PowerPlant". The name can be up to 123 characters long, and it can only consist of alphanumeric and underscore (_) characters.

2.4.2. Connection string

eShare uses the connection string defined for the main database as a template for connecting to project databases. That is, when connecting to a project database, eShare simply replaces the name of the main database in the connection string with the name of the relevant project database.

2.4.3. Database schema

A project database includes project configurations and data in the *dbo* schema, and it should be the default schema for the login that eShare uses. If the default schema is not configured, *dbo* is used as the default schema.

2.4.4. Database properties

These properties are enabled by default in project databases—do not disable them:

- Allow Snapshot Isolation
- Is Read Committed Snapshot On

This property is selected by default in project databases—you can change it if needed:

- Recovery model: Simple

Note: If you do need to change the recovery model, see Transaction logs first.

2.4.5. Database size

Most of eShare data is stored in project-specific databases. When eShare imports a 3D model from the Cadmatic EBM format, the data is uncompressed, normalized, and inserted into the relevant project database. EBM is a very compact transfer and storage format for 3D model data, whereas a relational database is less optimal for this type of data. On the other hand, database provides indexing that greatly speeds up data searches.

Depending on the amount of data in the EBM file and on actions performed on the server, project databases might grow fairly large, ranging from a couple of hundred megabytes to several gigabytes. For example, importing a 50 MB *.ebm file might create an approximately 2 GB project database, but if a project administrator uploads a new model, modifies the model, or republishes the model, the project database can contain several models at the same time, and the database size can multiply.

To ensure smooth operation of the server, allow project databases to extend at least three times their original, estimated size.

3. SQL server permissions required by eShare

Cadmatic eShare server uses a Microsoft SQL Server database. Supported versions can be found in docs.cadmatic.com/systemrequirements.

The following topics provide information on the permissions for running Microsoft SQL Server database in Cadmatic eShare.

Note: The following applies to Cadmatic eShare version 2024T1 or newer. Previous versions may have required different permissions.

3.1. Installation

To install Cadmatic eShare for the first time, the used SQL Server login must have permission to create databases ("CREATE ANY DATABASE"). If the same login is used in the eShare connection

string (this is the default), no other permissions need to be granted, and **eShare can be used with this setup alone without further changes**.

With "CREATE ANY DATABASE", eShare has full rights to create new project databases and acts as db_owner of those projects. If this database ownership is not possible due to security concerns, the following sections will describe in detail how to change the connection string, as well as which specific permissions are needed for eShare to function.

3.2. Configuring the database connection string

The database connection string used by eShare is configured in Internet Information Services (IIS) Manager. The login specified in that connection string is the login that performs all eShare database operations.

Do the following:

1. Sign in to the eShare server computer with administrative privileges, and open the Internet Information Services (IIS) Manager application.
2. In the Connections pane, browse to **Sites > eShare**.
3. In the main view, open **Connection Strings**.
4. Double-click the Main row, and edit the connection string as required.
5. Restart the web server for the changes to take effect.

3.3. Required permissions

3.3.1. Day-to-day operation

Day-to-day operation of eShare requires the following permissions:

- On the main database:
  - CONNECT
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
- On the project database:

- CONNECT
- SELECT
- INSERT
- UPDATE
- DELETE
- ALTER on ModelObjectObject

Publishing:

- EXECUTE
- ALTER (on ModelObjectAttribute, ModelObjectAttributeStaging,ModelObject, ModelAttributeDefinition)
- REFERENCES (on ModelObjectAttribute, ModelObjectAttributeStaging,ModelObject,ModelObjectObject, ModelAttributeDefinition)

The publishing permissions are not required, if the IIS setting *Cadmatic.IsAttributeStagingTableUsed* is set to false. This, however, results in slower publishing. See Configuring Application Settings.

Note: If the eShare project was created in eShare user interface, the user accessing the database is automatically db_owner of the database, and none of these permissions need to be granted manually.

3.3.2. Creating projects

Each eShare project has its corresponding database. In order to create databases from the eShare user interface, the eShare login must have permission to create databases ("CREATE ANY DATABASE").

If this is not possible due to security policies, eShare projects must be created manually using the DatabaseManager tool supplied with eShare. See Database administration .

3.3.3. Installation and version upgrades

When installing or upgrading to a newer version of eShare, database revision upgrades generally assume all the above permissions + db_owner level permissions, as these upgrades may alter anything in the main or project databases.

4. Database administration

The installation directory of the Cadmatic eShare server contains the DatabaseManager.exe command-line tool that you can use to manage eShare databases. The default location of this tool is *C:\Program Files\Cadmatic\eShareAdminTools\eShareDBTools*; in this document the location is defined as *<eShare>\eShareAdminTools\eShareDBTools*.

Many of the administrative tasks that you can perform with this tool can be done in two ways, whichever you prefer:

- Perform the whole task with *DatabaseManager.exe*.
- Use *DatabaseManager.exe* to generate an SQL script file, and then run the script on SQL Server.

For a complete list of commands that the tool supports, run the command `DatabaseManager.exe help`.

For information on a specific command, run the command `DatabaseManager.exe <command> help`. For example, `DatabaseManager.exe drop help`.

4.1. Database creation

The main database must exist and contain a fully created database schema when the eShare server is started.

Project databases can be created as empty, for example with Microsoft SQL Server Management Studio, and eShare will create the database schema before adding project data to the database. Or, you can create a project database and its schema using *DatabaseManager.exe*.

4.1.1. Creating the main database

Perform the following to create the main database and its schema.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   `DatabaseManager.exe create -t Main -c <connection string> -d <database to create>`

Example

- A command to create the main database "EShare" on the default instance of SQL Server running on localhost, using your credentials:
  ```
  DatabaseManager.exe create -t Main -c "Data Source=localhost;
  Integrated Security=True" -d EShare
  ```

4.1.2. Creating an SQL script that creates the main database

Perform the following to create an SQL script file that you can then run on SQL Server to create the main database and its schema.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   ```
   DatabaseManager.exe create -t Main -f <sql file to create> -d
   <database to create>
   ```

Example

- A command to create the script file *CreateMainDatabase.sql* for creating the main database "EShare":
  ```
  DatabaseManager.exe create -t Main -f
  "C:\DatabaseScripts\eShare\CreateMainDatabase.sql" -d EShare
  ```

4.1.3. Creating a project database

Perform the following to create a project database and its schema.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   ```
   DatabaseManager.exe create -t Project -c <connection string> -d
   <database to create>
   ```

Example

- A command to create the project database "EShare_PowerPlant" on the default instance of SQL Server running on localhost, using your credentials:
  ```
  DatabaseManager.exe create -t Project -c "Data Source=localhost;
  Integrated Security=True" -d EShare_PowerPlant
  ```

4.1.4. Creating an SQL script that creates a project database

Perform the following to create an SQL script file that you can then run on SQL Server to create a project database and its schema.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   ```
   DatabaseManager.exe create -t Project -f <sql file to create> -d
   <database to create>
   ```

Example

- A command to create the script file *CreateProjectPowerPlant.sql* for creating the project database "EShare_PowerPlant":
  ```
  DatabaseManager.exe create -t Project -f
  "C:\DatabaseScripts\eShare\CreateProjectPowerPlant.sql" -d EShare_
  PowerPlant
  ```

4.1.5. Adding a project database to the main database

Perform the following to add a specified project database to the main database. The command also has checks for if the project is already in the main database, or if the project database does not exist.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   ```
   DatabaseManager.exe addProjectToMain -p <project database to add> -n
   <project name> -c <connection string> -d <main database name>
   ```

Example

- A command to add the project database "EShare_PowerPlant" to the main database "EShare" on the default instance of SQL Server running on localhost, using your credentials:
  ```
  DatabaseManager.exe addProjectToMain -p EShare_PowerPlant -n
  PowerPlant -c "Data Source=localhost; Integrated Security=True" -d
  EShare
  ```

4.1.6. Creating an SQL script that adds a project database to the main database

Perform the following to create an SQL script file that you can then run on SQL Server to add a specified project database to the main database.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   ```
   DatabaseManager.exe addProjectToMain -p <project database to add> -n <project name> -f <sql file to create> -d <main database name>
   ```

Example

- A command to create the script file *AddProjectPowerPlantToEShare.sql* for adding the project database "EShare_PowerPlant" to the main database "EShare":
  ```
  DatabaseManager.exe addProjectToMain -p EShare_PowerPlant -n PowerPlant -f "C:\DatabaseScripts\eShare\AddProjectPowerPlantToEShare.sql" -d EShare
  ```

4.2. Database upgrade

When eShare server is upgraded to a newer version, there might be a need to update database properties or schemas. These updates are performed automatically if the database login that the installer uses has the required permissions. But, if the installer is unable to perform these updates, you must manually upgrade the databases to be compatible with the new version of Cadmatic eShare. You can upgrade all databases at the same time or one database at a time, as appropriate.

If a required database upgrade is not performed, the following will occur:

- If schemas are not up-to-date, eShare does not start.
- If the properties of the main database are not up-to-date, eShare displays a notification to the system administrator on the front page and on administrative pages.
- If the properties of a project database are not up-to-date, eShare displays a notification to the system administrator and to project administrators on the project administration page.

4.2.1. Upgrading all databases

Perform the following to upgrade the main database and all related project databases.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   `DatabaseManager.exe upgradeall -c <connection string> -d <main database>`

Example

- A command to upgrade all databases that relate to the main database "EShare" on the default instance of SQL Server running on localhost:
  `DatabaseManager.exe upgradeall -c "Data Source=localhost; Integrated Security=True" -d EShare`

4.2.2. Creating an SQL script that upgrades all databases

Perform the following to create an SQL script file that you can then run on SQL Server to upgrade the main database and all related project databases.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   `DatabaseManager.exe upgradeall -c <connection string> -f <sql file to create> -d <main database>`

   Note: Connection string is needed because DatabaseManager.exe checks the current state of the database to be able to create a proper upgrade script.

Example

- A command to create the script file *UpgradeAllDatabases.sql* for upgrading all databases that relate to the main database "EShare" on the default instance of SQL Server running on localhost:
  `DatabaseManager.exe upgradeall -c "Data Source=localhost; Integrated Security=True" -f "C:\DatabaseScripts\eShare\UpgradeAllDatabases.sql" -d EShare`

4.2.3. Upgrading the main database

Perform the following to upgrade just the main database.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   `DatabaseManager.exe upgrade -t Main -c <connection string> -d <main database>`

Example

- A command to upgrade the main database "EShare" on the default instance of SQL Server running on localhost:
  `DatabaseManager.exe upgrade -t Main -c "Data Source=localhost; Integrated Security=True" -d EShare`

4.2.4. Creating an SQL script that upgrades the main database

Perform the following to create an SQL script file that you can then run on SQL Server to upgrade just the main database.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   `DatabaseManager.exe upgrade -t Main -c <connection string> -f <sql file to create> -d <main database>`

Note: Connection string is needed because *DatabaseManager.exe* checks the current state of the database to be able to create a proper upgrade script.

Example

- A command to create the script file *UpgradeMainDatabase.sql* for upgrading the main database "EShare" on the default instance of SQL Server running on localhost:
  `DatabaseManager.exe upgrade -t Main -c "Data Source=localhost; Integrated Security=True" -f "C:\DatabaseScripts\eShare\UpgradeMainDatabase.sql" -d EShare`

4.2.5. Upgrading a project database

Perform the following to upgrade a specific project database.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   `DatabaseManager.exe upgrade -t Project -c <connection string> -d <database to upgrade>`

Example

- A command to upgrade the project database "EShare_PowerPlant" on the default instance of SQL Server running on localhost:
  `DatabaseManager.exe upgrade -t Project -c "Data Source=localhost; Integrated Security=True" -d EShare_PowerPlant`

4.2.6. Creating an SQL script that upgrades a project database

Perform the following to create an SQL script file that you can then run on SQL Server to upgrade a specific project database.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   `DatabaseManager.exe upgrade -t Project -c <connection string> -f <sql file to create> -d <database to upgrade>`

   Note: Connection string is needed because *DatabaseManager.exe* checks the current state of the database to be able to create a proper upgrade script.

Example

- A command to create the script file *UpgradeProjectPowerPlant.sql* for upgrading the project database "EShare_PowerPlant" on the default instance of SQL Server running on localhost:
  `DatabaseManager.exe upgrade -t Project -c "Data Source=localhost; Integrated Security=True" -f "C:\DatabaseScripts\eShare\UpgradeProjectPowerPlant.sql" -d EShare_PowerPlant`

4.3. Database removal

When an eShare project administrator removes a project from Cadmatic eShare, by default eShare tries to remove (drop) also the project database. If eShare does not have sufficient permissions to remove databases, you must remove redundant project databases manually, using Microsoft SQL Server Management Studio or the DatabaseManager.exe tool.

Note: If eShare is not supposed to remove databases, you should prevent it from even trying. You can do this in the Application Settings view of Internet Information Services (IIS) Manager, by setting *Cadmatic.DropProjectDatabases* to "false".

When a database administrator removes the project database of a project that still exists in eShare, the project reference is removed from the main database and the project is no longer listed in the user interface.

Your site should have an established procedure for archiving projects before they are removed.

4.3.1. Removing a project database

Perform the following to remove a project database.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   ```
   DatabaseManager.exe drop -t Project -c <connection string> -d
   <database to drop>
   ```

Example

- A command to remove the project database "EShare_PowerPlant" from the default instance of SQL Server running on localhost:
  ```
  DatabaseManager.exe drop -t Project -c "Data Source=localhost;
  Integrated Security=True" -d EShare_PowerPlant
  ```

4.3.2. Creating an SQL script that removes a project database

Perform the following to create an SQL script file that you can then run on SQL Server to remove a project database.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   `DatabaseManager.exe drop -t Project -f <file to create> -d <database to drop>`

Example

- A command to create the script file *RemoveProjectPowerPlant.sql* for removing the project database "EShare_PowerPlant":
  `DatabaseManager.exe drop -t Project -f "C:\DatabaseScripts\eShare\RemoveProjectPowerPlant.sql" -d EShare_PowerPlant`

4.3.3. Removing a project database from main database

Perform the following to remove a project database from main database.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   `DatabaseManager.exe removeProjectFromMain -n <project name in eShare> -c <database connection string> -f <output file for the script> -d <main database name>`

Example

- A command to remove the project database of "PowerPlant" from the main database on the default instance of SQL Server running on localhost, using your credentials:
  `DatabaseManager.exe removeProjectFromMain -n PowerPlant -c "Data Source=localhost; Integrated Security=True" -d EShare`

4.3.4. Creating an SQL script that removes a project database from main database

Perform the following to create an SQL script file that you can then run on SQL Server to remove a project database from main database.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   ```
   DatabaseManager.exe removeProjectFromMain -n <project name in eShare>
   -f <output file for the script> -d <main database name>
   ```

Example

- A command to create the script file *RemoveProjectPowerPlantFromEShare.sql* for removing the project database of "PowerPlant" from main database:
  ```
  DatabaseManager.exe removeProjectFromMain -n PowerPlant -f
  "C:\DatabaseScripts\eShare\RemoveProjectPowerPlantFromEShare.sql" -d
  EShare
  ```

4.4. Database permissions

4.4.1. Checking permissions in a database

Perform the following to check permissions in a database. This command can be used for debugging, as it lists permissions needed, checks if they have been granted and why these permissions are needed.

Do the following:

1. In a command-prompt window, go to *<eShare>\eShareAdminTools\eShareDBTools*.
2. Run this command:
   ```
   DatabaseManager.exe checkPermissions -l -c <database connection
   string> -d <database name>
   ```

Example

- A command to check permissions in database "EShare" on the default instance of SQL Server running on localhost:

```
DatabaseManager.exe checkPermissions -l -c "Data Source=localhost;
Integrated Security=True" -d EShare
```

4.4.2. Granting permissions to a database

Warning: Consider generating a script and running it manually instead. Allowing the tool to write permissions directly to the database is dangerous.

Perform the following to grant permissions in a database.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   ```
   DatabaseManager.exe grantPermissions -l <eShare login> -u <user> -c
   <database connection string> -d <database name> --
   IUnderstandThisIsDangerous
   ```

Note: Option "--IUnderstandThisIsDangerous" must be specified to write permissions directly to database.

Example

- A command to grant permissions to "eshare_user" to database "eShare_tankerdemo" on the default instance of SQL Server running on localhost:
  ```
  DatabaseManager.exe grantPermissions -l eshare -d eshare_tankerdemo -
  u eshare_user -c "server=localhost\SQLEXPRESS; Integrated
  Security=true; initial catalog=EShare; TrustServerCertificate=True; ;
  MultipleActiveResultSets=True" --IUnderstandThisIsDangerous
  ```

4.4.3. Creating an SQL script that grants permissions to a database

Perform the following to to create an SQL script file that you can then run on SQL Server to grant permissions in a database.

Do the following:

1. In a command-prompt window, go to *<eShare>**eShareAdminTools**eShareDBTools*\.
2. Run this command:
   ```
   DatabaseManager.exe grantPermissions -l <eShare login> -u <user> -f
   <output file for the script> -d <database name>
   ```

Example

- A command to create the script file *GrantPermissions.sql* for granting permissions to "eshare_user" to "eShare_tankerdemo" database:
  ```
  DatabaseManager.exe grantPermissions -l eShare -u eshare_user -f "C:\DatabaseScripts\eShare\GrantPermissions.sql" -d eshare_tankerdemo
  ```

5. Server backups

Cadmatic eShare has a tool for backing up and restoring eShare servers, server and project databases, server configuration files, and individual eShare projects: eShare Backup Restore Tool. It is strongly recommended to use the eShare Backup Restore Tool for all backups and restoring.

User can also use standard database and file backup mechanisms to back up eShare data. For example, use the Microsoft SQL Server Management Studio application of Microsoft SQL Server to back up and restore databases. The SQL Server Agent application of full versions of SQL Server can be configured to create scheduled backups, but you can also schedule backups with the Task Scheduler function built into Windows.

5.1. Databases to back up 23
5.2. Directories to back up 24
5.3. Restoring eShare server from backups 24

5.1. Databases to back up

Make sure you regularly back up the following databases.

- The main database. The default name of this database is "EShare", but if you create the database manually the name might be different. This database contains information about the projects and users in the system, and the permissions that each user has in specific projects.
- Each project database. The default name has the format "EShareProject_<id>", but if you create project databases manually the name is probably different. You can find the name of each project database on the project administration page, under General > Database Name.

5.2. Directories to back up

Make sure you regularly back up the following folders.

- Installation directory of eShare server. The default location is *C:\Program Files\Cadmatic\eShare*. This directory contains, for example, server configuration files such as *ConnectionStrings.config* that specifies the connection string used to connect to the main and project databases, and *AppSettings.config* that specifies the settings for connecting to the license server.
- Program data directory of eShare server. The default location is *C:\ProgramData\Cadmatic\eShare*. This directory contains project-related files and maintenance/error logs.

5.3. Restoring eShare server from backups

Perform the following to restore Cadmatic eShare server from backups. It is strongly recommended to use the eShare Backup Restore Tool.

Prerequisites

- You have database and directory backups, as described in Databases to back up and Directories to back up.

Do the following:

1. Install eShare server on a computer that does not have eShare installed, as described in *Cadmatic eShare Installation Guide*, using the same version of eShare server that was used to create the backups. In the installation wizard, we recommend that you select the option to manually create the databases.
2. Restore the main eShare database and the project databases from backups.
3. Restore eShare server data from backups. The most important files and folders to restore are:
    - The application configuration files *ConnectionStrings.config* and *AppSettings.config*.
    - The entire *C:\ProgramData\Cadmatic\eShare* directory (project data).