Technical Article

DriveLock Database Guide

Database Maintenance and Configuration

DriveLock SE 2022

Contents

1 Introduction

1.1 Purpose of this document

This documentation is a guide to the DriveLock database and is intended to be a knowledgebase source for DriveLock and database administrators.

2 DriveLock Database

2.1 Overview

The DriveLock Enterprise Service uses a database to store and manage data.

2.2 Database Security

The DriveLock Enterprise Service uses Windows Authentication to connect to the databases. SQL authentication is currently not supported.

The DriveLock Enterprise Service runs under the Windows account that was selected during installation. This account requires a login to the SQL server, which is linked to a user to the DriveLock database.

DES service account -> SQL server login -> DriveLock database user (for all DriveLock databases)

The Database Install Wizard sets up database login and users and provides options depending on the environment.

2.2.1 Roles

DriveLock uses a custom role **srcsystem** to grant permissions to execute stored procedures and to use custom table types. It will be created with the database schema. The DES database user must be member of this role.

2.2.2 SQL Server

For DES functionality, the database user mapped to the DES service account's login requires the following role memberships:

- **db_datareader**
- **db_datawriter**
- **srcsystem**

On SQL Servers, it is recommended to set up database maintenance, backup, and event grooming tasks directly on the SQL Server as a job.

2.2.3 SQL Express

SQL Express does not come with a SQL Agent, so it cannot schedule and run database maintenance or backup tasks. In this case, the DES can run database maintenance, backup, and event grooming tasks.

For this, the database user also requires to be a member of the **db_owner** role. The **db_owner** role gives the DES service account all permissions on the database.

2.3 Tenant databases

A new tenant database is added for each new tenant to strictly separate the data of each tenant. By default, the DriveLock Database Service runs under a "root" tenant. All tenants have the same database schema and version as the base DriveLock database.

The naming convention for tenant databases is:

<DriveLock database name>_<tenant name>

For example, DriveLock_tenant1

2.4 Considerations regarding agent configuration

The database size and growth depends a lot on the DriveLock agent configuration.

- Event reporting

  Event reporting is the most important configuration to consider, as it hard to estimate the number of incoming events. There are some features in DriveLock that can generate a huge number of events, for example file auditing (reporting of file filter events) or application control. Please refer to the DriveLock DB Sizing document for more details. Consider which events need to be collected and reported.

- Hard & software inventory

  This depending on the number and configuration of the computers. Once the inventory has been generated, the size will only change marginally, as the hardware and the software of the computers change.

- Centrally stored policies

  These are stored directly in the database. A lot of different large policies with many versions can use up some space, however this is foreseeable. For example, when a complex policy uses up 5MB of space, then a new version of it will require roughly additional 5MB of space in the database. **Note** that large files, for example Security Awareness videos or application hash databases can significantly increase the size of a policy. It is recommended to store these in designated separate policies. Space used up by large policies can be freed with deleting old versions of a policy.

3 Database installation and update

For more information on the database installation procedure, please refer to the DriveLock Installation Guide at https://drivelock.help/.

Note: Before changing anything regarding the DriveLock database, please make sure to create a backup.

3.1 Database Install Wizard

The Database Install Wizard guides you through the installation. The Installation Guide explains the required steps.

3.2 Database connection parameters

The DriveLock Enterprise Service uses registry keys to specify the connection parameters to the database. The keys are located in the windows registry at `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\CenterTools\DES`

- dbServer: the Instance name of the SQL Server
- dbName: the name of the main DriveLock database, default: DriveLock
- dbType: MSSQL

3.2.1 Moving or changing SQL server

In case the database is moved to another server or the database server is renamed, please edit the database connection parameters in the registry (see above) and restart the DriveLock Enterprise Service.

Make sure that the DES service account has a login on the server and has access and permissions on all DriveLock databases (see database security).

4 Database Maintenance

The maintenance of the database has two main tasks:

- Ensure performance despite of growth
  The performance of searching and reading data highly depends on indexes. Database indexes should be defragmented at regular intervals or rebuilt if needed.
- Limit database growth
  To limit database growth, event data must be deleted at regular intervals. This is referred to as database grooming. Events meeting the specified condition and references to these events in corresponding entity tables are removed during this operation.

Maintenance of the DriveLock database is performed by stored procedures that are installed with each DriveLock database:

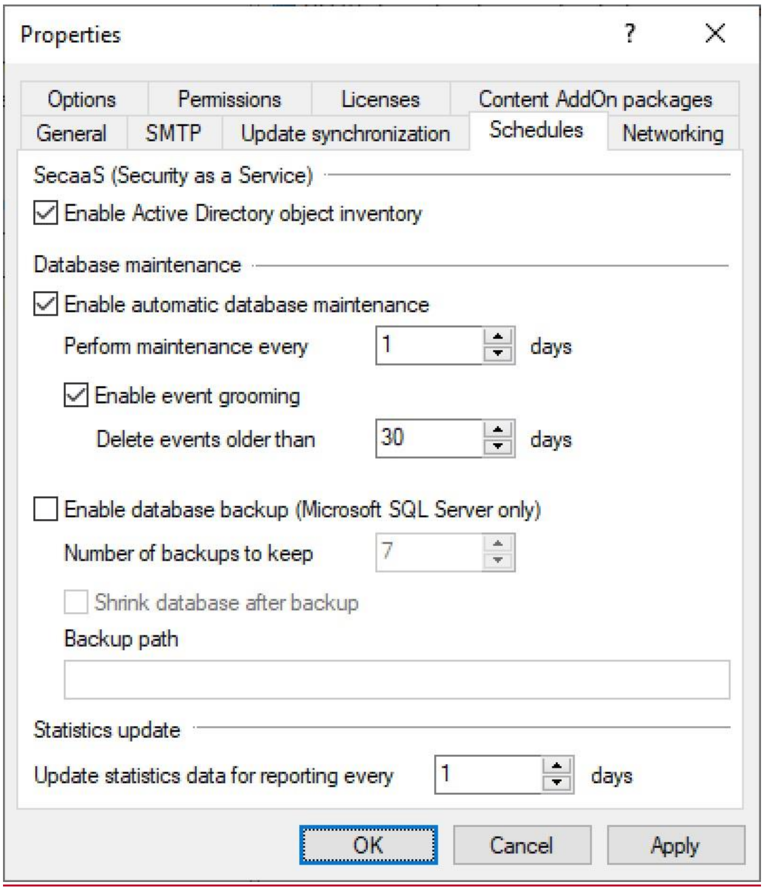
- ctsp_maintenance
  This handles the rebuilding/defragmenting of the database indexes and should run on the DriveLock database.
- ctsp_groomevents
  Default groom events job deletes events older than specified age in days.
- ctsp_delete_...
  Several stored procedures to delete data that otherwise could grow indefinitely.
- ctsp_backup
  Optional for local SQL Express databases to perform automated backups.
- ctsp_shrink
  Optional for SQL Express databases to shrink the database after backup.

These stored procedures are either directly started by the DriveLock Enterprise Service or can be configured to run directly on the SQL server in a scheduled job. If possible, it is always recommended to run these tasks directly on the SQL server.

Maintenance schedules should be configured to run at times of low database activity. During maintenance, the database is still functional and accessible, but query performance may decrease, and timeouts are more likely to occur.

4.1 Maintenance on MS SQL (full version):

On a full MS SQL server, it is recommended to run the maintenance tasks as jobs directly on the server. In this case, the tasks running from the DriveLock Enterprise Service must be deactivated:



SecaaS (Security as a Service)

☑ Enable Active Directory object inventory

Database maintenance

☑ Enable automatic database maintenance

Perform maintenance every 1 days

☑ Enable event grooming

Delete events older than 30 days

☐ Enable database backup (Microsoft SQL Server only)

Number of backups to keep 7

☐ Shrink database after backup

Backup path

Statistics update

Update statistics data for reporting every 1 days

Backups should be implemented for all databases, just like for all existing other databases on the server.

Shrinking is usually not necessary on a full MS SQL server. Once the DriveLock Enterprise Service is running in production mode for a while, and the grooming of old event data starts showing effect, the database sizes should be stable. The following table shows the recommended setup of the stored procedures, and on which databases they should run.

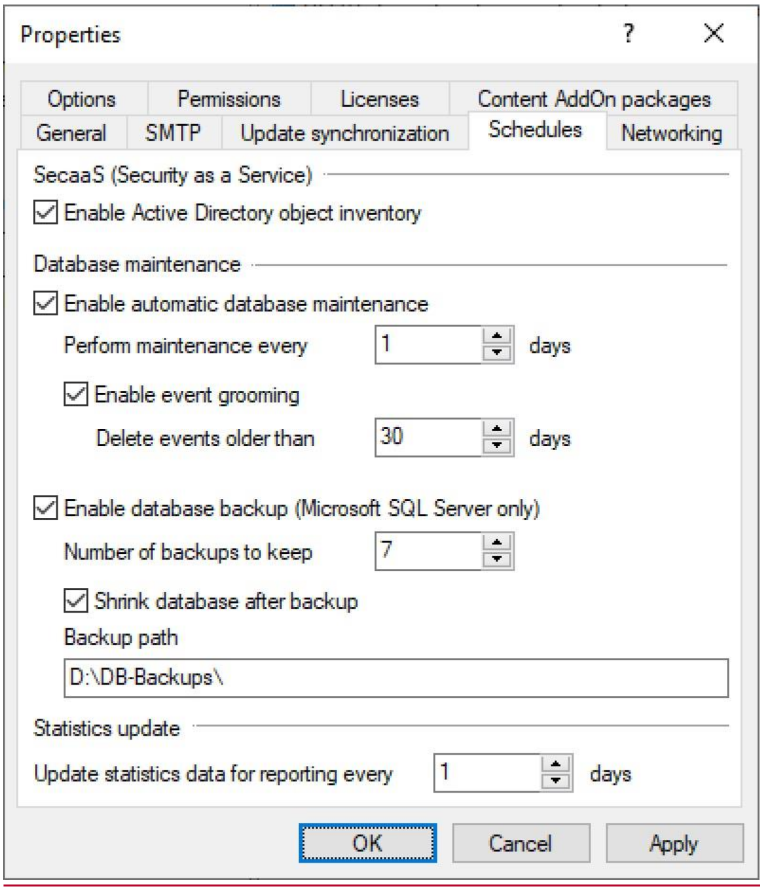
Stored procedure	Drive Lock	Tenant database (DriveLock_Tenant1)
ctsp_groomevents ctsp_delete_acprocesses ctsp_delete_datasnapshots ctsp_delete_computervulnerabilities ctsp_delete_threatdetections	x	x
ctsp_maintenance	x	x
ctsp_backup		

ctsp_shrink		

Database (index) maintenance, backups and database shrinking can of course also be implemented with the SQL server admin’s own methods.

4.2 Maintenance on MSSQL (EXPRESS version):

As the SQL Agent is not available on MS SQL EXPRESS editions, it is necessary that the tasks are executed by the DriveLock Enterprise Service.



SecaaS (Security as a Service)

☑ Enable Active Directory object inventory

Database maintenance

☑ Enable automatic database maintenance

Perform maintenance every 1 days

☑ Enable event grooming

Delete events older than 30 days

☑ Enable database backup (Microsoft SQL Server only)

Number of backups to keep 7

☑ Shrink database after backup

Backup path

D:\DB-Backups\

Statistics update

Update statistics data for reporting every 1 days

Note that the backup path has to be accessible by both the SQL Server and the DriveLock Enterprise Service.

The DriveLock Enterprise Service will manage the number of backups kept. Old backups will be deleted.

4.3 Stored procedures for maintenance

4.3.1 ctsp_maintenance

This stored procedure handles the maintenance of indexes in the DriveLock database. Example: To start the ctsp_maintenance stored procedure, use the following command in T-SQL:

```
EXEC ctsp_maintenance
```

This stored procedure has no parameters and works for all DriveLock databases. The executing database user requires db_owner permissions to run index maintenance.

4.3.2 ctsp_groomevents

This stored procedure deletes events that are older than a specified number of days. During this operation, events meeting the specified condition are removed.

Difference between events and audit events:

- Standard events are sent by DriveLock components, for example providing the information that an executable or a drive was blocked.
- Audit events are administrative events with the purpose of tracing administrative and security actions, triggered by the DriveLock accounts in the DOC and MMC, for example when changing a policy or permissions.

Stored procedure parameters:

```
@days:          age of the events in days
@batchSize:     number of events that are deleted in a batch
@audit:         flag for deleting only normal or audit events
                event: 0
                audit event: 1
```

Example: To start the ctsp_groomevents stored procedure and delete all events older than 30 days with a batch size of 10000, use the following command in T-SQL:

```
EXEC ctsp_groomevents 30, 10000, 0
```

Example: To start deleting audit events older than 90 days with a batch size of 10000:

```
EXEC ctsp_groomevents 90, 10000, 1
```

Deleting events and audit events can be done together in a job.

Note that this operation can run for a long time, depending on the amount of data to be deleted. The data is deleted in batches, so it is possible to stop and restart this without data loss or bigger rollbacks.

4.3.3 ctsp_delete_acprocesses

This stored procedure handles the deletion of old process data that is generated from application control events.

Stored procedure parameters:

```
@days:         age of the processes in days
@batchSize:    number of processes that are deleted in a batch
```

Example: To start the ctsp_delete_acprocesses stored procedure, use the following command in T-SQL:

```
EXEC ctsp_ delete_acprocesses 30, 10000
```

4.3.4 ctsp_delete_datasnapshots

This stored procedure handles the deletion of old data snapshots that are generated for time based graphs in DOC.

Stored procedure parameters:

```
@days:         age of the data snapshot in days
@batchSize:    number of data snapshots that are deleted in a batch
```

Example: To start the ctsp_delete_datasnapshots stored procedure, use the following command in T-SQL:

```
EXEC ctsp_ delete_datasnapshots 30, 10000
```

4.3.5 ctsp_delete_computervulnerabilities

This stored procedure handles the deletion of old vulnerability scanner detection data.

Stored procedure parameters:

```
@days:         age of the detected computer vulnerability in days
@batchSize:    number of computer vulnerabilities that are deleted in a batch
```

Example: To start the ctsp_delete_computervulnerabilities stored procedure, use the following command in T-SQL:

```
EXEC ctsp_delete_computervulnerabilities 90, 10000
```

4.3.6 ctsp_delete_threatdetections

This stored procedure handles the deletion of old Microsoft Defender threat detection data.

Stored procedure parameters:

```
@days:        age of the threat detections in days
@batchSize:   number of threat detections that are deleted in a batch
```

Example: To start the ctsp_delete_threatdetections stored procedure, use the following command in T-SQL:

```
EXEC ctsp_delete_threatdetections 90, 10000
```

4.3.7 ctsp_backup

This stored procedure creates a full backup of the database. The backup will be stored in the designated folder, the backup file name contains the database name and a timestamp.

Note: This stored procedure is not meant to be called manually or by a job, but by the DriveLock Enterprise Service, which will also manage the maximum number of available backups. It is intended for use on local SQL Express servers, where the SQL Agent is not available. On full SQL Servers, please contact your database administrator to create a backup job.

The only parameter is an NVARCHAR that specifies the backup path. The filename will be automatically appended by the stored procedure. Note that this path is from the SQL Server's point of view. The database user needs db_owner or db_backupoperator role membership to back up a database.

4.3.8 ctsp_shrink

This stored procedure will shrink the database and release the free space from the database files, hence reducing the database file size.

Note: This stored procedure is not meant to be called manually or by a job, but by the DriveLock Enterprise Service, it is intended for use on local SQL Express servers, where the SQL Agent is not available.

Database shrinking can only be started by a database sysadmin or the database owner. The DriveLock Enterprise Service account must be the owner of the database, otherwise this stored procedure will fail.

This stored procedure has no parameters. The database user needs **db_owner** role membership to shrink a database.

4.4 Recommended execution order for maintenance tasks

It is recommended to execute the maintenance stored procedures in the following order:

- Event grooming
- Delete old data (datasnapshots, application control processes, vulnerability scanner detections and MS Defender threat detections)
- Database index maintenance
- Database backup
- Database shrinking

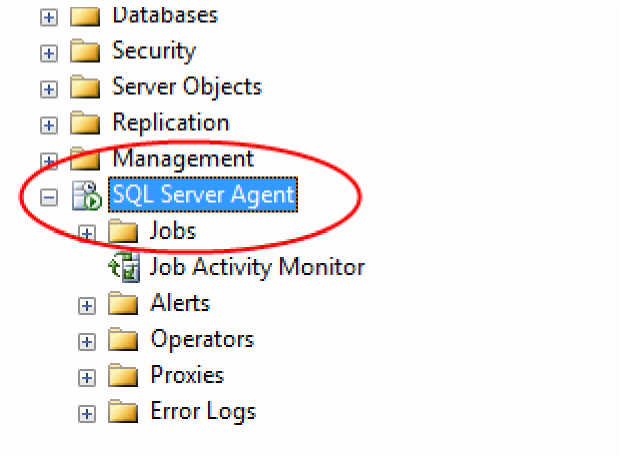
5 Appendix

5.1 Example of configuring a maintenance job (MS SQL Server)

This is an example on how to set up a maintenance job using Microsoft SQL Server Management Studio. Maintenance jobs in SQL Server can have many additional features (for example, e-mail notification if a job fails), but these are beyond the scope of this document. For additional information about these features, refer to the Microsoft SQL Server and Microsoft SQL Server Management Studio documentation. Maintenance jobs can also be implemented using the SQL Server maintenance plans feature.

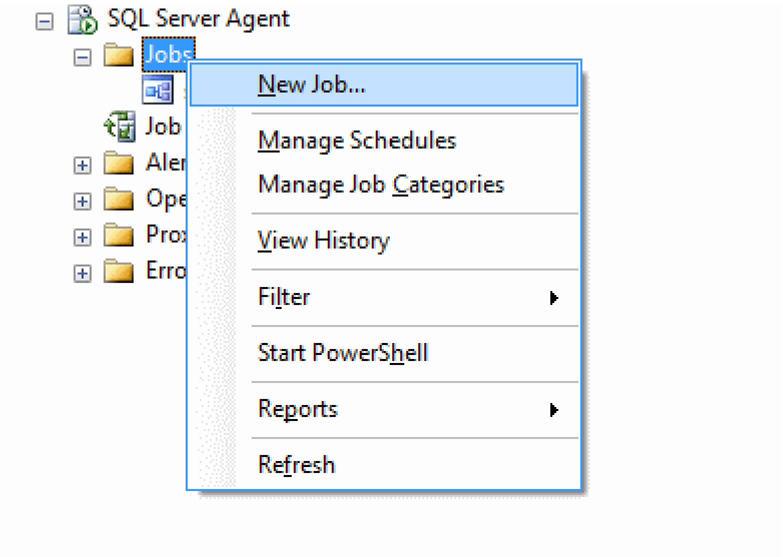
The following example describes how to configure a maintenance job using the Microsoft SQL Server Agent.

Database maintenance jobs require that the SQL Server Agent service is running.

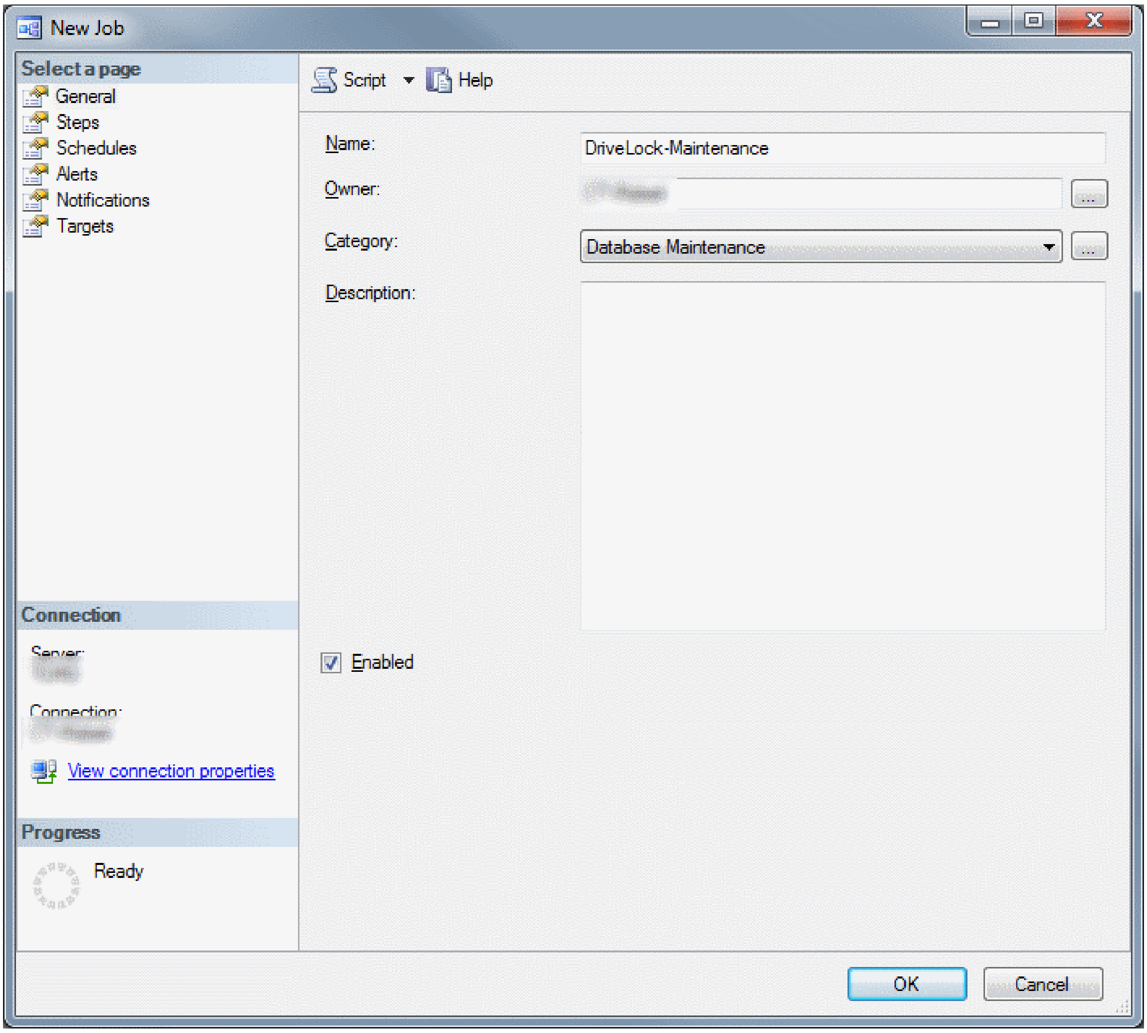


5.1.1 Creating a new job

Right-click **Jobs** and then click **New Job**.



Type the job name and then select *Database Maintenance* as the category. The owner of the job needs to have permission to read/write data and to defragment indexes.



Name: DriveLock-Maintenance

Owner: [redacted]

Category: Database Maintenance

Description:

☑ Enabled

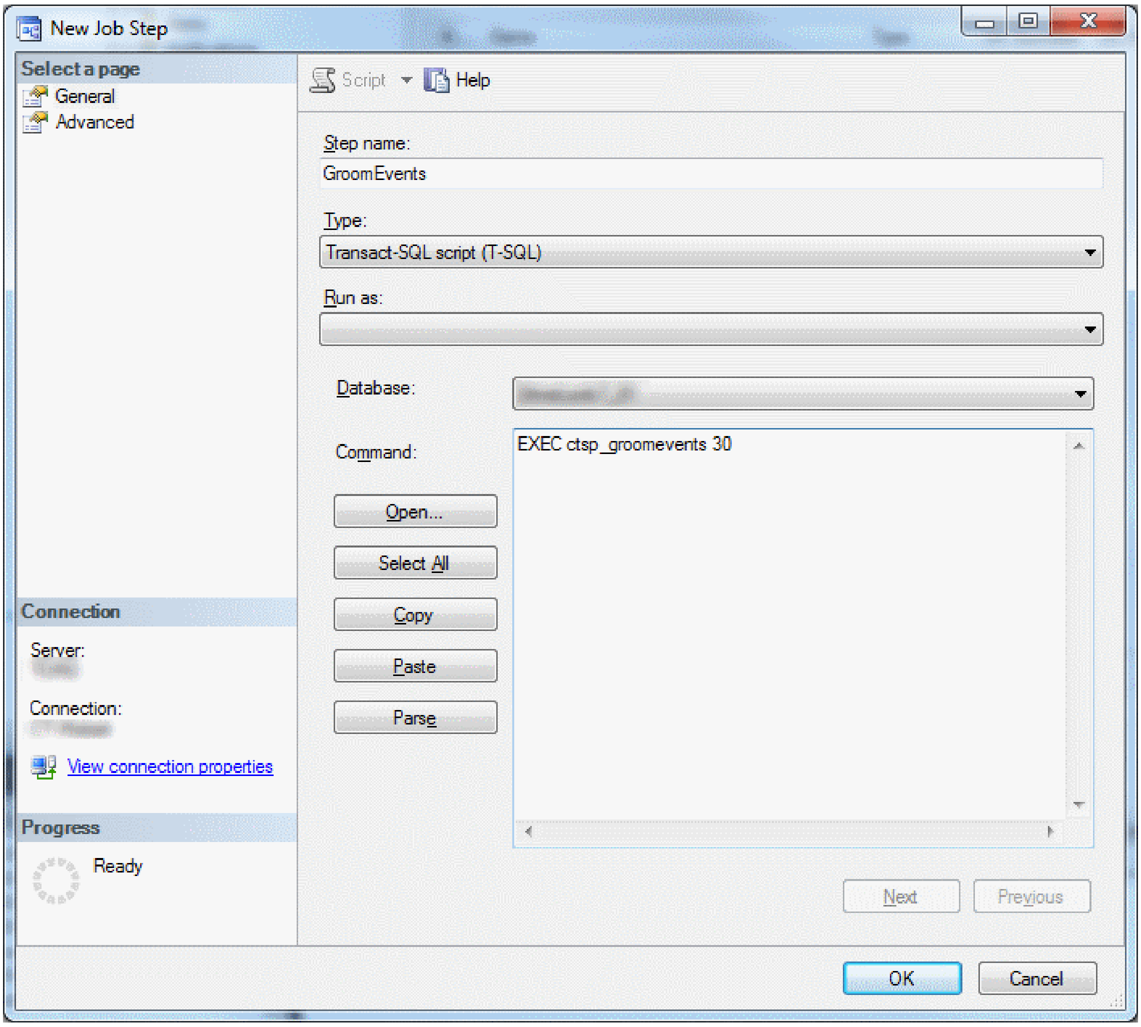
Server: [redacted]

Connection: [redacted]

Creating job steps

On the Steps page, add a new Job Step. Type a name for the step, select “*Transact-SQL script (T-SQL)*” as the type, select the DriveLock database and type the command that will be executed each time the step is executed. Create one job step for “*event grooming*” and second one for “*database maintenance*”.

Job Step for event grooming:



Step name: GroomEvents

Type: Transact-SQL script (T-SQL)

Run as:

Database: [redacted]

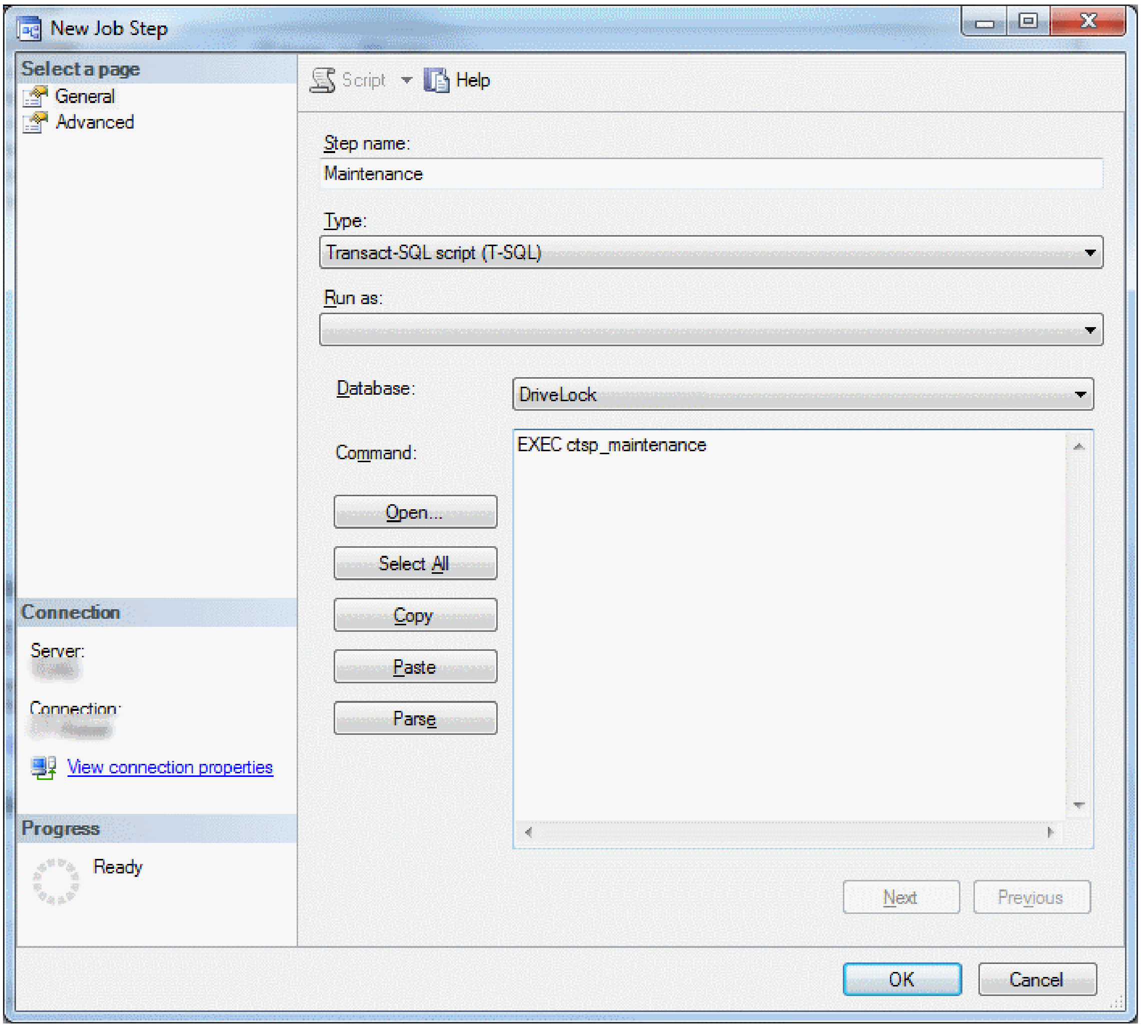
Command:

```
EXEC ctsp_groomevents 30
```

Server: [redacted]

Connection: [redacted]

Job Step for maintenance:



Step name: Maintenance

Type: Transact-SQL script (T-SQL)

Run as:

Database: DriveLock

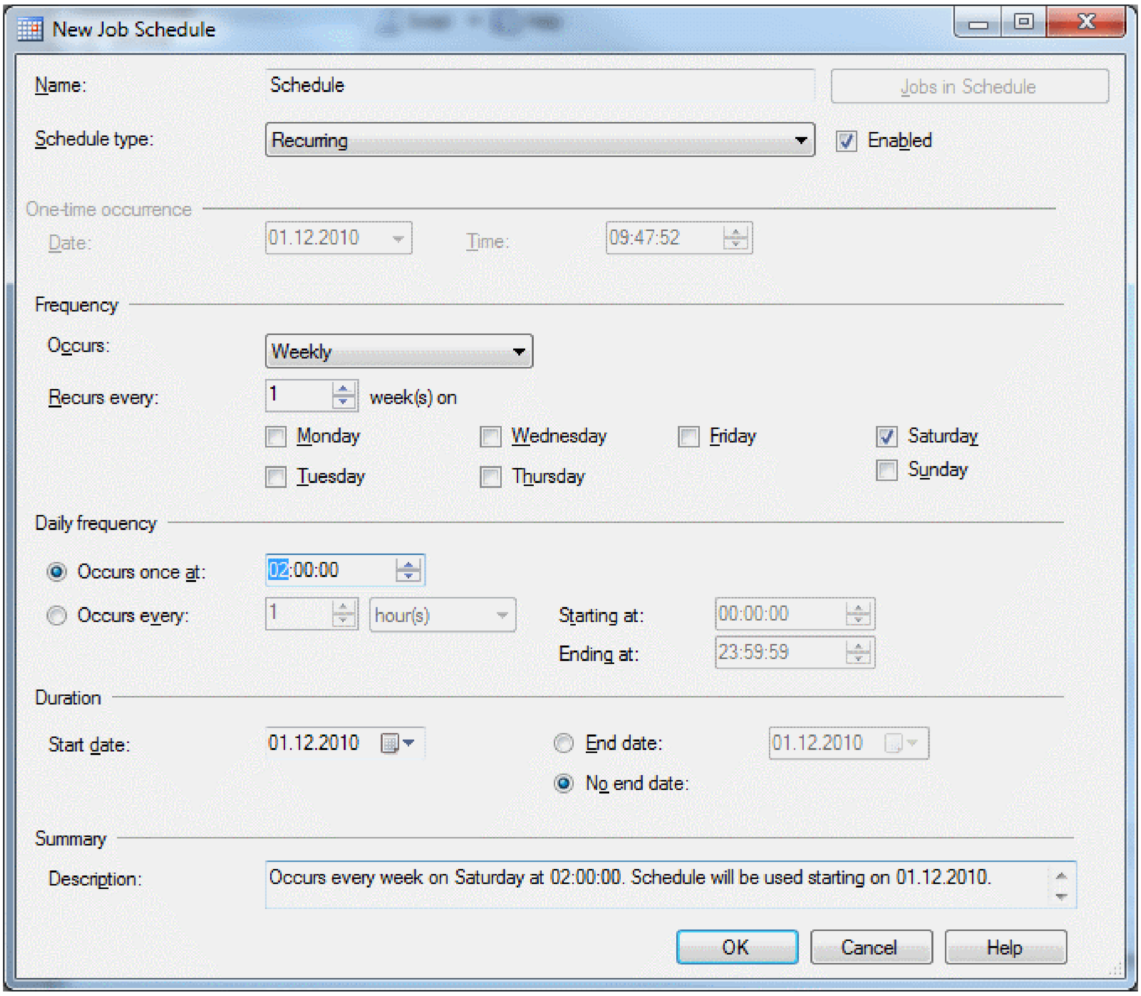
Command:

```
EXEC ctsp_maintenance
```

Click “OK” to save the Job Steps.

Creating a schedule

On the Schedules page, add a new schedule. Type a name for the schedule and then configure the frequency, execution time and start date of the job.



New Job Schedule

Name: Schedule

Jobs in Schedule

Schedule type: Recurring ☑ Enabled

One-time occurrence
Date: 01.12.2010 Time: 09:47:52

Frequency
Occurs: Weekly
Recurs every: 1 week(s) on
□ Monday □ Wednesday □ Friday ☑ Saturday
□ Tuesday □ Thursday □ Sunday

Daily frequency
● Occurs once at: 02:00:00
○ Occurs every: 1 hour(s) Starting at: 00:00:00
Ending at: 23:59:59

Duration
Start date: 01.12.2010 ○ End date: 01.12.2010
● No end date:

Summary
Description: Occurs every week on Saturday at 02:00:00. Schedule will be used starting on 01.12.2010.

Click “OK” to create the schedule and again “OK” to finalize the creation of the job.

Copyright