

Instruction   
for scheduled   
SQL database backup

Contents

[Prefix 3](#_Toc512618087)

[Introduction 4](#_Toc512618088)

[Description 5](#_Toc512618089)

[Complete database backup 5](#_Toc512618090)

[Transaction log backup 5](#_Toc512618091)

[Differential backup 6](#_Toc512618092)

[Schedule 7](#_Toc512618093)

[Maintenance plan 7](#_Toc512618094)

[Multiple sites 7](#_Toc512618095)

[Retention period and storage 7](#_Toc512618096)

[Transaction log backup 8](#_Toc512618097)

[Daily backup 8](#_Toc512618098)

[Bi-weekly backup 9](#_Toc512618099)

[Monthly backup 9](#_Toc512618100)

[Yearly backup 9](#_Toc512618101)

[Point in time recovery using transaction log 9](#_Toc512618102)

[How to set up 11](#_Toc512618103)

[Creating a new maintenance plan 11](#_Toc512618104)

[Add subplans to the maintenance plan 12](#_Toc512618105)

[Add a task to set up full database backup 15](#_Toc512618106)

[Add a task to set up transaction log backup 19](#_Toc512618107)

[Schedule the backups 22](#_Toc512618108)

[Full database backup schedule 22](#_Toc512618109)

[Transaction log schedule 24](#_Toc512618110)

[Save the maintenance plan 25](#_Toc512618111)

[Add cleanup subplan with task 26](#_Toc512618112)

[Remove old full database backups 26](#_Toc512618113)

[Remove old transaction log backups 30](#_Toc512618114)

[Copy to backup media 36](#_Toc512618115)

[Appendix 37](#_Toc512618116)

[Restoring the database 37](#_Toc512618117)

[As it was at the time of the full database backup 38](#_Toc512618118)

[At a point in time 42](#_Toc512618119)

[At a point in time where you miss the latest full database backup prior to the incident 48](#_Toc512618120)

[At a point in time after the latest transaction log backup 49](#_Toc512618121)

[At a point in time where one of the transaction log backups are damaged or missing 51](#_Toc512618122)

[References 52](#_Toc512618123)

# Prefix

This document use the Heading styles for each chapter. Heading 1 for the main chapters/sections of the document. Heading 2 for subchapters Heading 3 for sub subchapter and so on.

The style **Strong** is used to describe text that is to be typed exactly as shown.

The style Intense reference is used for text that you can enter according to your own choice.

The style Emphasisis is used for text that refer to a menu option or field to use.

A warning is written inside a red frame with bold characters.

Informational comments are written in a blue dashed frame

References to alternative ways of doing something is written in a green double frame.

# Introduction

This document describes our practice for database backups with regards to the Adonis databases on MSSQL.

The document has 2 main parts. The “Description”, which give some background details on the choices we have made and why we have made them. Then the “How to set up” that give a step by step instruction on how to set it up.

We always recommend that the SQL backup will be incorporated in your ordinary backup routines. In this document, we describe storing the SQL backup on the local drive of the SQL server. These files should then be copied/moved to another storage media. You may therefore pick up these files in your ordinary file backup to have them incorporated in your ordinary backup routines.

# Description

## Complete database backup

A complete database backup will be required in all situations. Either as a stand alone backup, or as a starting point for differential and transaction log backups. How often you should make the complete database backup may depend on your backup policy. But our suggestion would be once a day.

## Transaction log backup

The transaction log is used by the SQL server to log changes to the database before the database is updated. It is used by the SQL server to be able to rollback uncompleted transactions. The SQL server also use it to recover and ensure the integrity in the event that the server start up after being abruptly terminated.

If the transaction log file is not backed up, it will just grow until it completely fill the harddrive (or meet a size limit if that has been set). Unless simple recovery model is used.

Simple recovery model should not be used for production databases, as it does not provide the same level of protection against loss as a full recovery model.

When the transaction log file is backed up, it allow to reuse the space in the transaction log file that has been backed up. More frequent transaction log backups also allow for a smaller transaction log file.

So even if you don’t use transaction log backup in your backup/recovery plan, you should make regular transaction log backups, in order to allow for the re-use of the transaction log space.

The transaction log backup allow for point in time recovery, where the database can be restored to any point in time, not just to the time of the backup. So the transaction log backup can be part of the backup/recovery policy.

Be aware that the transaction log backup depends on a full database backup and all the prior transaction log backups from the full database backup up until the transaction log backup taken after the time we want to restore too.

A transaction log backup is a lot smaller then a full database backup, as it only contains the changes made to the database since the previous transaction log backup. Using more transaction log backups in the backup/recovery plan instead of full database backups can save backup storage. Transaction log backup also allow for point in time recovery, so the database can be restored to any time, not just the time when the backup was made. The drawback is that many transaction log backups may have to be restored to get to the restore time, if there is a long time since the full database backup.

If any of the transaction log backups are missing or damaged, the remaining transaction log backups after that time will also be useless.

### Differential backup

We usually don’t use differential backups. The reason is that differential backups also depend on the complete backup, like the transaction log backups. So it does not provide any advantage in that context. At the same time, the differential backups only allow for restore to the time that the differential backup was made. The transaction log backup will fill the purpose of the differential backup(less storage required then a complete backup), but the transaction log backup also provide functionality with the possibility of point in time restore.

It is possible though, that the differential backup could be used to reduce the storage requirement compared to transaction log backups. And that each differential backup may be set up to span a longer time period then the transaction log backups, allowing for restore of fewer backups to get to the desired time.

Our conclusion has been that the differential backup is not providing any significant advantages compared to complete backups and transaction log backups. While the transaction log backup have the additional advantage of point in time restore and re-use of transaction log space. A combination of all 3 can be beneficial in some situations, particularly with larger databases where it may take too long time to use a full database backup at a daily basis. However, we have so far not seen that situation with regard to Adonis.

## Schedule

Our suggestion is to have a daily complete backup to ensure that we have a starting point every day, and together with the transaction log backups we can recover to any time on that day. If a transaction log backup file is lost or damaged, it would not have a larger impact then 24 hours. We also suggest that the transaction log backup is taken every 2 hours. Frequent transaction log backup will also prevent the transaction log file from growing too large.

The sweet spot for how frequent you want the full backup and when to only depend on transaction log backup is the balance between reduced need for backup storage space but also minimizing the risk of having a lost or damaged transaction log backup file that prevent further restore or the need to restore a vast amount of transaction log backups to get to the restore point.

## Maintenance plan

With the SQL server maintenance plan, you can set up many different tasks which you can schedule at certain times. With the maintenance plan you can set up complete database backup, transaction log backup, cleanup of old backups and other maintenance tasks. It is a simple way to set up, schedule and maintain the complete database and transaction log backups.

Please note that the maintenance plan depend on the SQL Server Agent windows service. If that service is stopped, then the maintenance plan will not run at the scheduled time.

## Multiple sites

With installation on multiple sites, you should follow the backup procedures on each site where you have a database. That allow you to restore the system as it was on that specific site (in case not everything is replicated), and also to speed up any restore time.

## Retention period and storage

How long you are going to store the backups, which media to use and where it is stored has to be according to your backup/restore policy and requirements.

If you don’t have any backup/restore policy, we can suggest to

* Keep transaction log backups for 1 week or longer
* Keep the daily backups for 1-2 weeks or longer (daily backup)
* Keep one of the complete backups bi-weekly for 2 months (bi weekly backup)
* Keep one of the complete backups each month for 1 year (monthly backup)
* Keep one complete backup from the beginning/end of the (fiscal) year (yearly backup)

### Transaction log backup

The transaction log backup, in connection with a complete backup, allow for point in time recovery. With a daily complete backup and keeping transaction log backups for a week, you can restore to any point in time for that week.

Keeping all transaction log backup for a week and daily backups, provide a double backup for that week.

In the event where any of the complete backups during the week has failed or is damaged, you may use the daily backup from the day before and all the transaction log backups for that day and the transaction log backup of the following day to get to your desired point in time. See the example “At a point in time where you miss the latest full database backup prior to the incident” in the appendix.

If a transaction log backup has failed, is damaged or missing, you only lack point in time recovery between the transaction log backup before the one that is missing or damaged and the next daily backup. Ref. the example “At a point in time where one of the transaction log backups are damaged or missing” in the appendix.

### Daily backup

With a complete backup daily, kept for 1-2 weeks, you have a simple way of restoring the database to any day for that period. There will only be a gap of 24 hours between each backup. If a transaction log backup is missing or damaged, you still have a new starting point every day. It also reduces the amount of transaction log backups you may have to restore, compared to having fewer complete backups and relying more on transaction log backups.

### Bi-weekly backup

The bi-weekly backup is to shorten the gap between daily and monthly backups for the last 2 months. Reducing the required amount of backup storage required, compared with keeping daily complete backups.

### Monthly backup

The monthly backups are for longer retention times, but also to reduce the required amount of backup storage required compared to keeping complete daily or bi-weekly backups. With regards to monthly payroll cycles, the most ideal time would be that this backup is made after the monthly payroll has been processed, but before the payroll period is closed. But as the backup is usually scheduled, and there may be a variation in the time of the actual payroll completion, we suggest that this is scheduled after the day when the payroll is usually supposed to be completed. This make it possible to make a restore of the database at a time just after the payroll completion. And by keeping it for a year, to have such a backup available for each payroll period (assuming monthly payroll) throughout the year.

### Yearly backup

The yearly backup are for long term retention, if that is required. Similar to the monthly backups, with regard to payroll, you may want to schedule this around the time of the closing of the fiscal year. The most ideal time would be after you have completed all the processing of the fiscal year, but just before closing the year. But as it is scheduled and automatically run, we suggest scheduling this for a day after completion of the fiscal year and before the processing of the first payroll in the following fiscal year. It is possible that the retention of this backup would correspond to your requirements of retaining financial data.

## Point in time recovery using transaction log

With the maintenance plan we set up, with complete backup at 01:00 and transaction log backup every 2 hours, if something happen and the database has to be restored to the state it was at 10:40, it can be accomplished by restoring the full database backup from 01:00. Then all the 2 hourly transaction log backups between 01:00 and 10:00. Then the transaction log backup made after 10:40 will be restored not entirely, but specified to restore just until 10:40.

If we would be missing one of the transaction log backup files, say at 07:00, that would render all the transaction log backups after 07:00 useless, and the latest time we could restore to would be 07:00. If we would be missing the full database backup, then we could not use the transaction log backups for recovery either, as we would be missing the starting point.

Now say that we have the full database backup on 25. January, but missing the full backup (or it is damaged) on 26. January, but we have all the transaction log backups from the 25 and 26. January, we could restore to any point in time on the 26. January by restoring the full database backup from the 25th and all the transaction log backups on 25th and 26th until the time we want on the 26th.

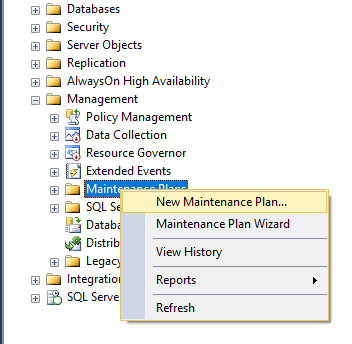
# How to set up

## Creating a new maintenance plan

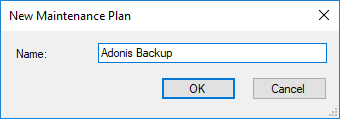
Start Microsoft SQL Server Management Studio and log on to the SQL Server.

Expand Management and expand Maintenance plan to view the existing maintenance plans. Rightclick on Maintenance plan

Select New Maintenance Plan ….

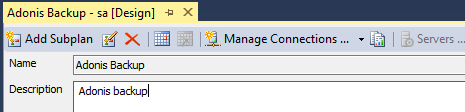


Give the maintenance plan a name (which not already exist)



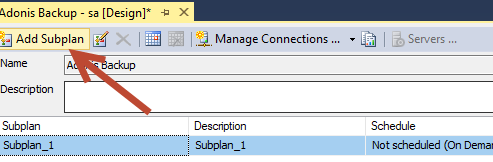
And click OK

Fill in a description



## Add subplans to the maintenance plan

Click Add Subplan on the toolbar

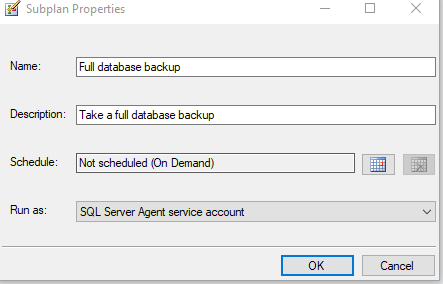


Name the subplan Full Database Backup

Give it a description Take a full database backup

You may set up the schedule now, but in this example we will come back to the schedule later.

Run as SQL Server Agent Service account.



Click OK

The account that the SQL Server Agent service is using must have sufficient rights in the file system to write to the destination folder where the backup files will be stored, and also have sufficient rights in the SQL server to make database backups of the Adonis database.

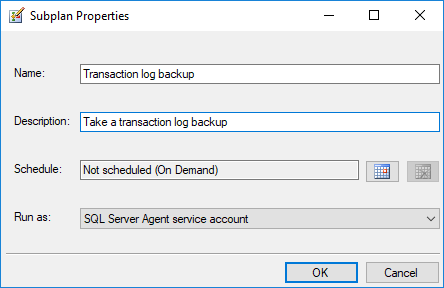
Click Add Subplan again

Name the subplan Transaction log Backup

Give it the description Take a Transaction log backup

You may set up the schedule now, but in this example we will come back to the schedule later.

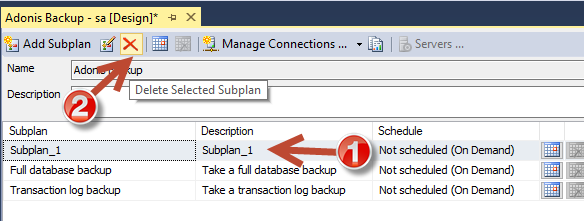
Run as SQL Server Agent Service account.

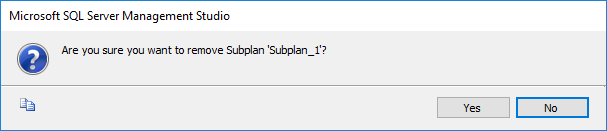


Click OK

You now have 3 subplans in this maintenance plan

Select Subplan\_1 in the list and click on the  icon in the toolbar to delete Subplan\_1



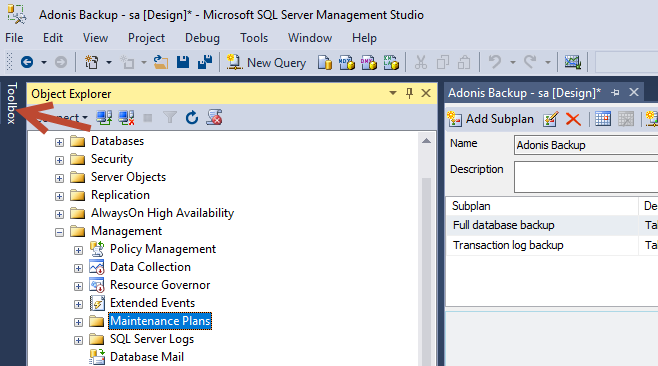


Confirm that you want to delete subplan\_1 by clicking Yes

## Add a task to set up full database backup

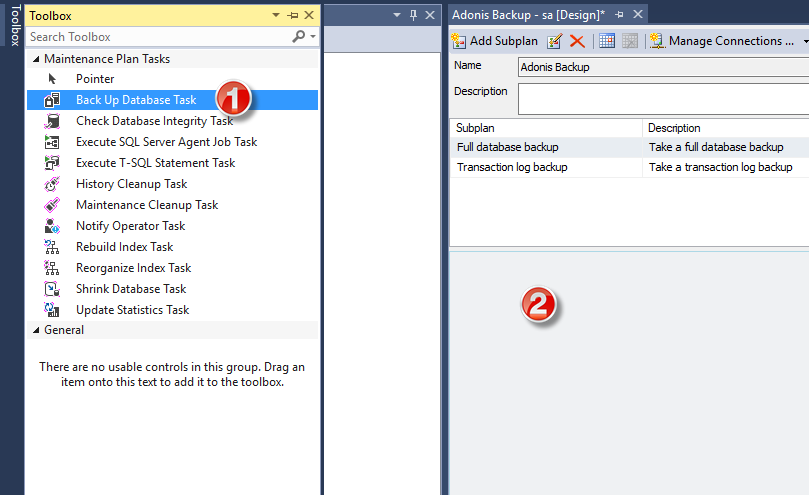
Select the Full database backup subplan

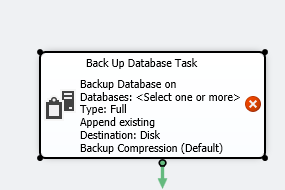
Click the Toolbox button at the upper far left



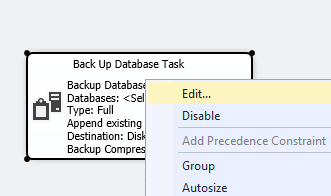
This will bring up a list of possible tasks you can add to the maintenanace plan.

Select and drag Back Up Database Task and drop it on the empty area bellow the list of your subplans (2)





You have now added a database backup task to your Full Database Backup subplan. Rightclick this task and select Edit to start configuring the backup task.

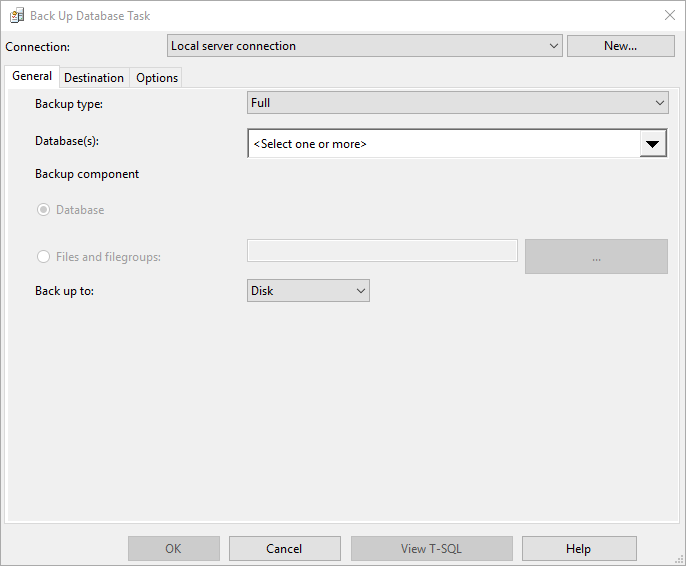


Connection Local server Connection is OK, since the task will run towards the server where we have added the maintenance plan.

Backup Type is Full since we are going to make a full (complete) database backup.

Select the dropdown for databases, and set a checkmark to the Adonis database to include the Adonis database in this backup task.

Backup to Disk as we are going to store the backup as a file in the file system.

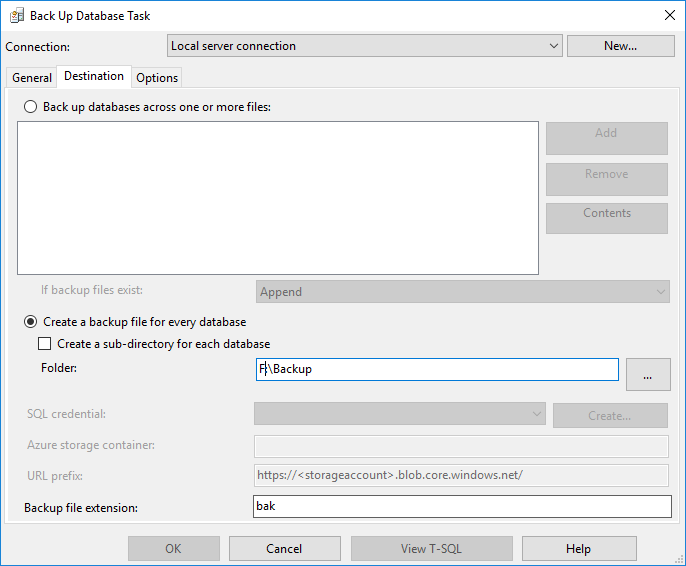


Click on the Destination page.

Make sure that you have selected Create a backup file for every database. Then select the drive and folder where you want to store you full database backup. Fill in the backup extension. By default the backup extension is BAK

Note that the extension is provided without the dot (just BAK, not .BAK)

You are free to use any extension you want, but when you use the SQL Server Management studio GUI to browse for files to restore, it will filter files with .BAK, .TRN and .LOG extensions. So by using another extension you may have to select to view All Files in order to have your backup file(s) listed.



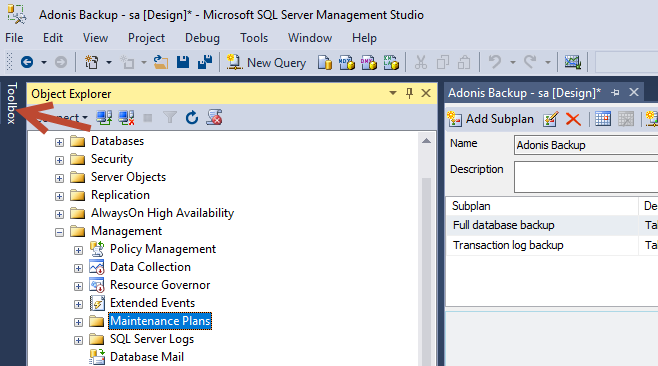
On the options page, there are additional options to control the use of compression, backup expiry, backup verification and backup encryption.

Click OK to close the window with the setup of this task.

## Add a task to set up transaction log backup

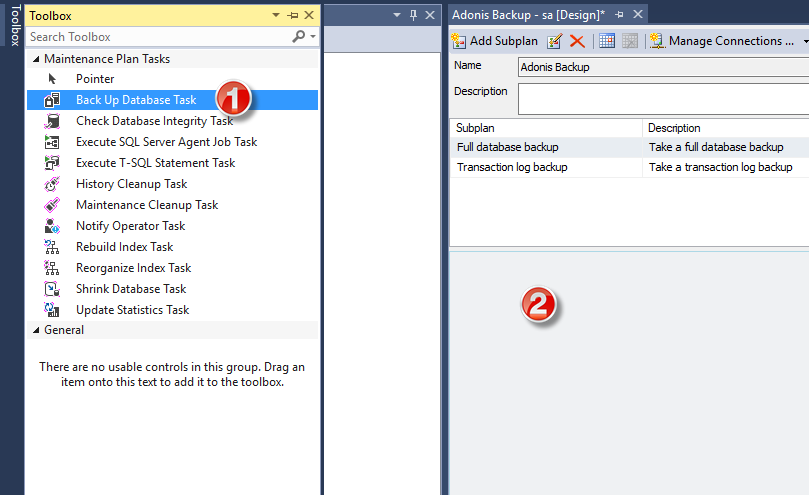
Select the Transaction log backup subplan

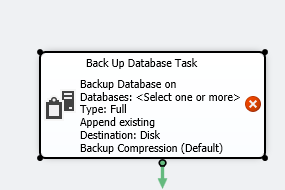
Click the Toolbox button at the upper far left



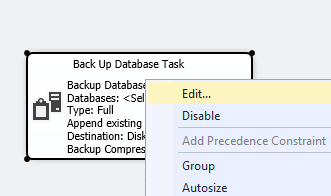
This will bring up a list of possible tasks you can add to the maintenance plan.

Select and drag Back Up Database Task and drop it on the empty area bellow the list of your subplans (2)





You have now added a database backup task to your Transaction log Backup subplan. Right click this task and select Edit to start configuring the backup task.

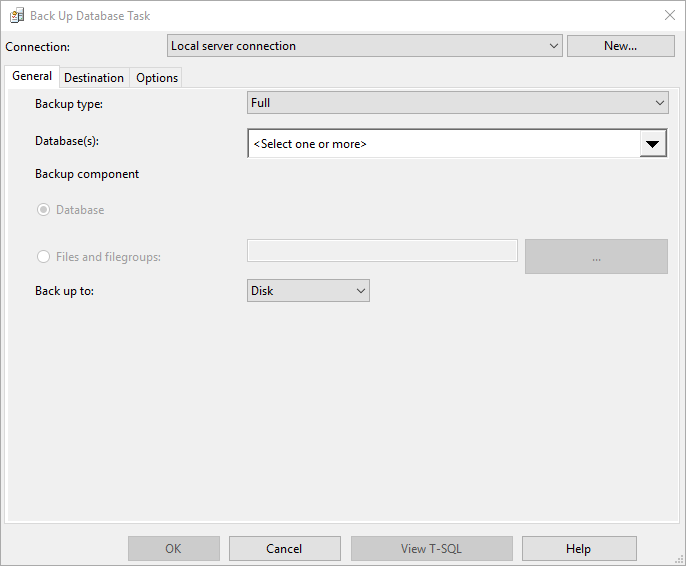


Connection Local server Connection is OK, since the task will run towards the server where we have added the maintenance plan.

Backup Type is Transaction log since we are going to make a transaction log backup.

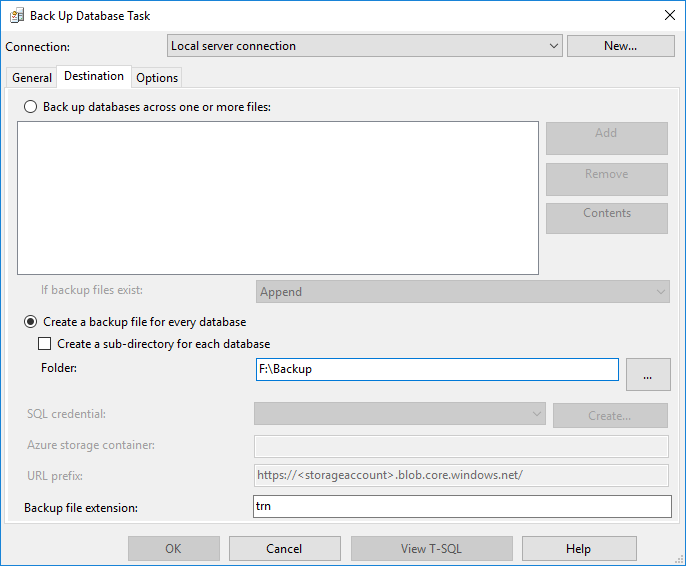
Select the dropdown for databases, and set a checkmark to the Adonis database to include the Adonis database in this backup task.

Backup to Disk as we are going to store the backup as a file in the file system.



Click on the Destination page.

Make sure that you have selected Create a backup file for every database. Then select the drive and folder where you want to store your transaction log backup. Fill in the backup extension. By default the backup extension is TRN



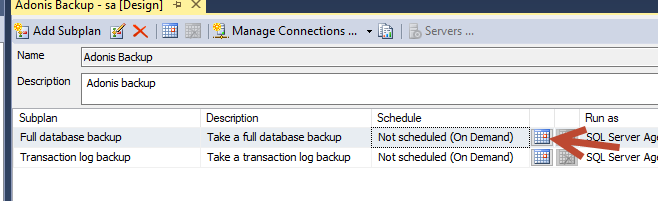
On the options page, there are additional options to control the use of compression, backup expiry, backup verification and backup encryption.

Click OK to close the window with the setup of this task.

## Schedule the backups

### Full database backup schedule

Click the calendar icon on the Full database backup line.



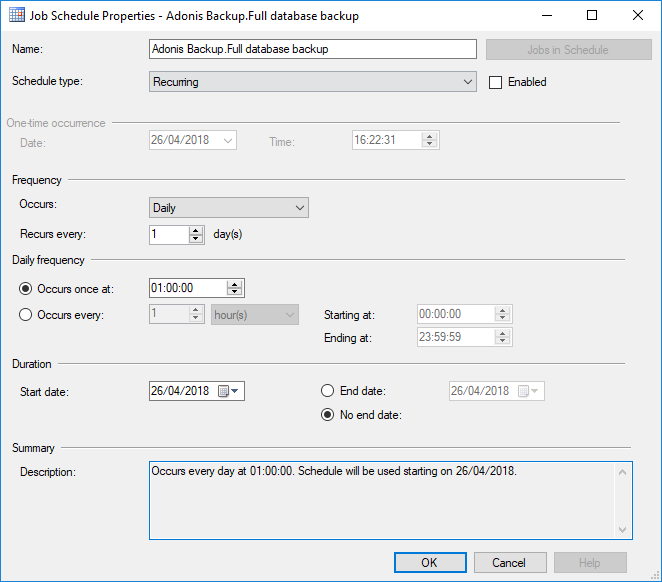
The schedule is set to be recurring and ensure that it is enabled.

Set the frequency to daily and to recur every 1 day

Set it to occurs once at 01:00:00

The start date is probably OK as it will default to today’s date. Otherwise set the date that you want to run the first backup.

Ensure that you have selected No end date



Click OK

### Transaction log schedule

Click the calendar icon on the Transaction log backup line.

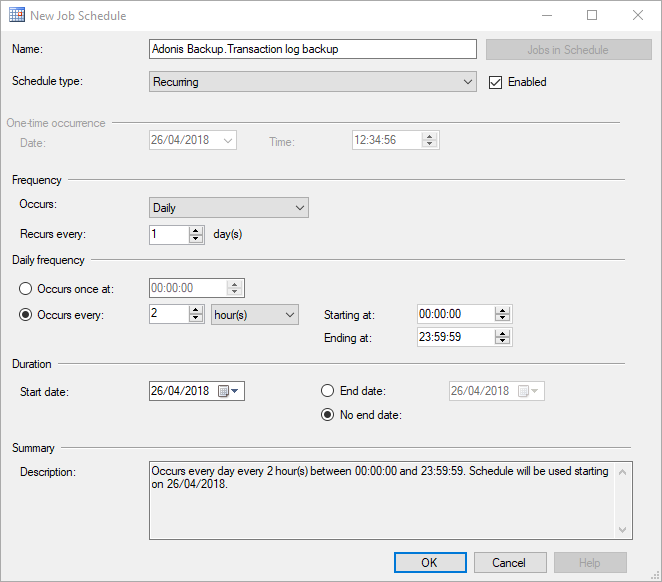
The schedule is set to be recurring and ensure that it is enabled.

Set the frequency to daily and to recur every 1 day

Set it to occur every 2 hours starting at 00:00:00 ending at 23:59:59

The start date is probably OK as it will default to today’s date. Otherwise set the date that you want to run the first backup.

Ensure that you have selected No end date

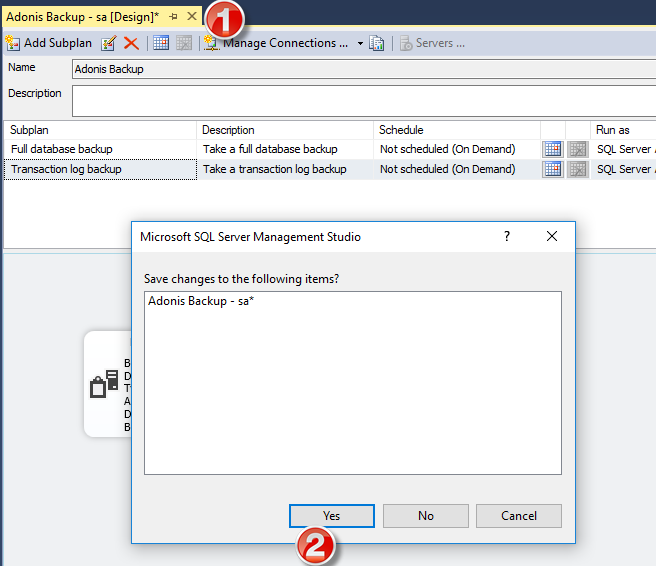


Click OK

If you have not yet taken a full database backup prior to running the transaction log backup, the SQL server may complain when it generate it, or give an error or warning that there is no full backup. There is not much use in Transaction log backups from before the first full database backup either, which is why it may complain. You could set the start date after the first full database backup, or just ignore the errors before the first full database backup.

## Save the maintenance plan

Click on the X on the maintenance plan tab to close it, and answer Yes to save changes



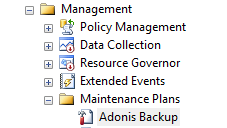
## Add cleanup subplan with task

### Remove old full database backups

To avoid filling up the harddrive with backups, we add a maintenance subplan to remove old backups.

If you want to do this cleanup in connection to your scheduled ordinary full backup, you could add these tasks to the Full Database Backup subplan instead.

Double click on the Adonis Backup maintenance plan we created earlier to open it again



Click the Add Subplan icon.

Give it a name Remove old backups

Give it a description Remove old full and transaction log backups

You may set a schedule now, but we will come back to that later.

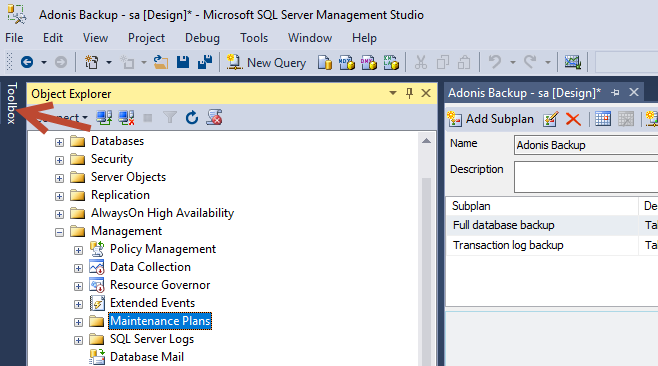
Run as SQL Server Agent service account

The account needs permission on the file system to list and delete the files.

Click OK

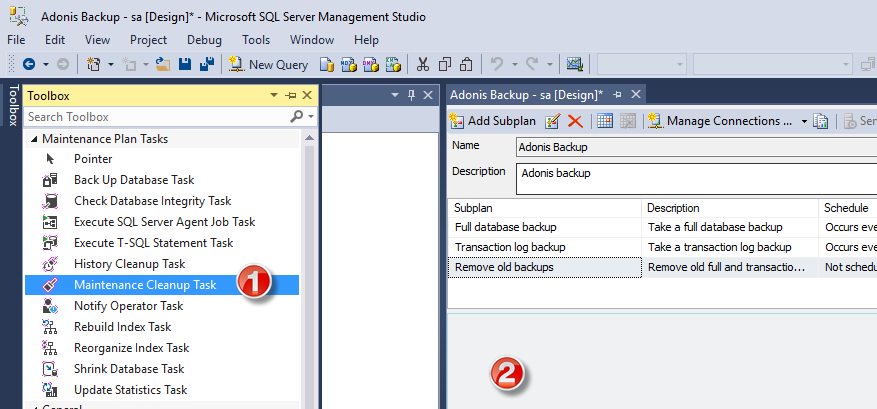
Select the newly created Remove old backups subplan

Click the Toolbox button at the upper far left

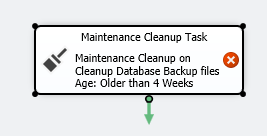


This will bring up a list of possible tasks you can add to the maintenanace plan.

Drag the Maintenance Cleanup Task and drop it on the empty area bellow the list of your subplans (2)



Rightclick on the newly added maintenance cleanup task



And select Edit

Local server connection is OK, as we will run it on the same server where we create the maintenance plan.

Select Backup files

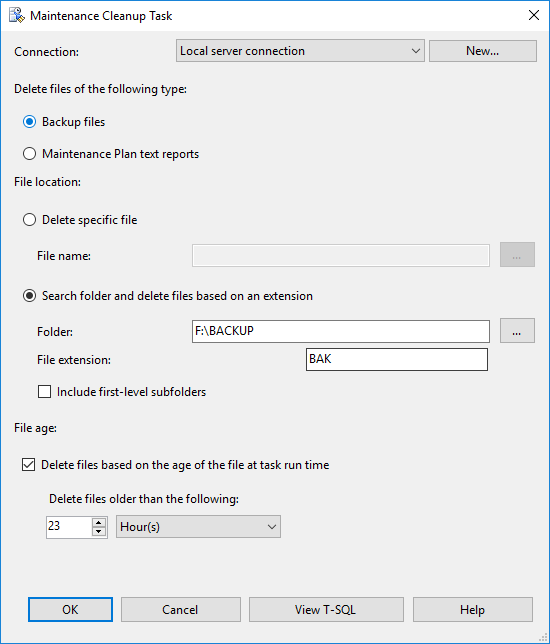
Select Search folder and delete files based on an extension Browse to and select the folder(or type in the location) where you stored the backup (ref the chapter “Add a task to set up full database backup”)

Enter BAK as the extension (ref the chapter “Add a task to set up full database backup”).

Please note that the extension is provided without the dot. It is just BAK not .BAK

Be aware that this will remove ALL the files in this folder with the specified extension that is older than the provided age limit. If there are other files with this extension in the folder that you want to keep, then use another extension or directory that does not contain such files, when you set up the backup task and this cleanup task.

Set a checkmark to select Delete files based on the age of the file at task run time then choose 23 Hours



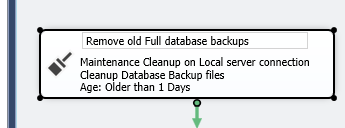
Click OK

We delete the files from this folder after 1 day, based on the assumption that you daily have made a copy of it to another backup media as part of your ordinary file backup routines. And that the retention and storage location of those backups comply with your backup/recovery plan.

We delete the backup files older than 23 hours, as the file may be less than one day old since it take some time to make the backup, and we may run the delete a bit ahead of the backup time to reduce the space consumption compared to making a new backup and then run the delete.

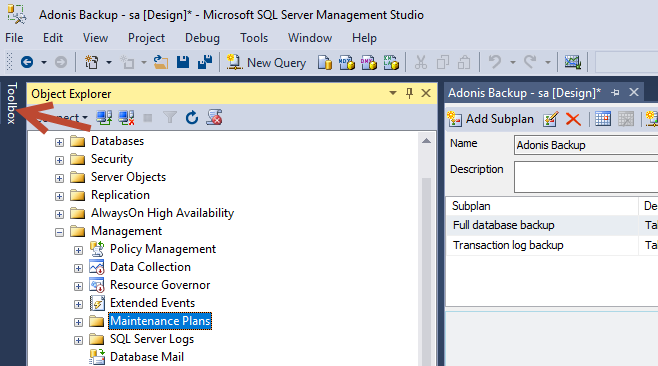
Rightclick on the task and select Rename

Give it a new name Remove old Full database backups



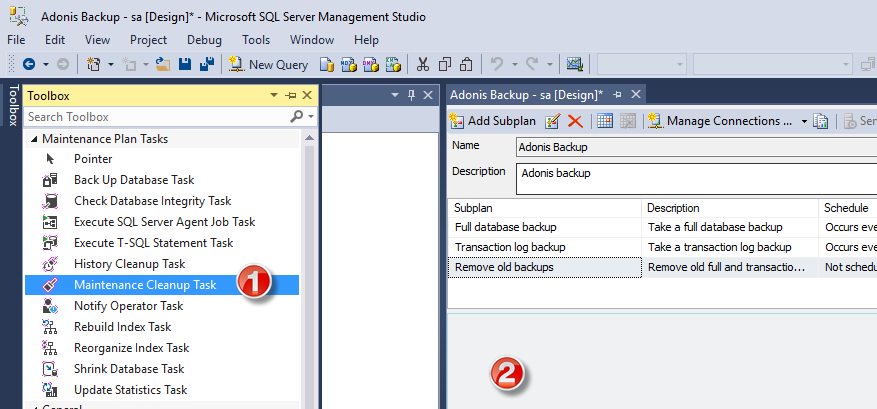
### Remove old transaction log backups

Click the Toolbox button at the upper far left

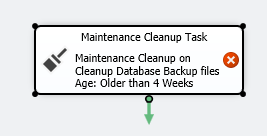


This will bring up a list of possible tasks you can add to the maintenance plan.

Drag the Maintenance Cleanup Task and drop it on the empty area bellow the list of your subplans (2)



Rightclick on the newly added maintenance cleanup task



And select Edit

Local server connection is OK, as we will run it on the same server where we create the maintenance plan.

Select Backup files

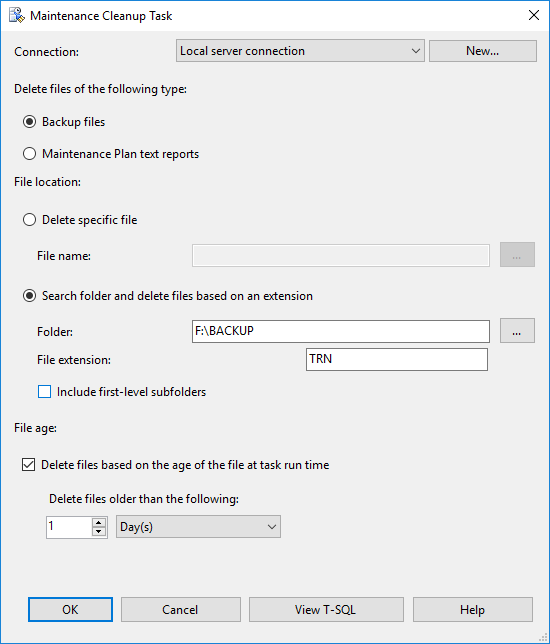
Select Search folder and delete files based on an extension Browse to and select the folder(or type in the location) where you stored the backup (ref the chapter “Add a task to set up full database backup”)

Enter TRN as the extension (ref the chapter “Add a task to set up transaction log backup”).

Please note that the extension is provided without the dot. It is just TRN not .TRN

Be aware that this will remove ALL the files in this folder with the specified extension that is older than the provided age limit. If there are other files with this extension in the folder that you want to keep, then use another extension or directory that does not contain such files, when you set up the backup task and this cleanup task.

Set a checkmark to select Delete files based on the age of the file at task run time then choose 1 Day

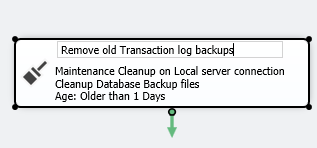


Click OK

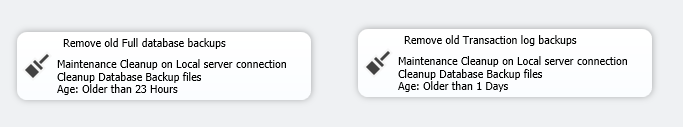
We delete the files from this folder after 1 day, based on the assumption that you daily have made a copy of it to another backup media as part of your ordinary file backup routines. And that the retention and storage location of those backups comply with your backup/recovery plan.

Rightclick on the task and select Rename

Give it a new name Remove old Transaction log backups



You now have 2 tasks in this subplan



Click on the calendar icon next to the Remove old backups subplan to schedule the execution.

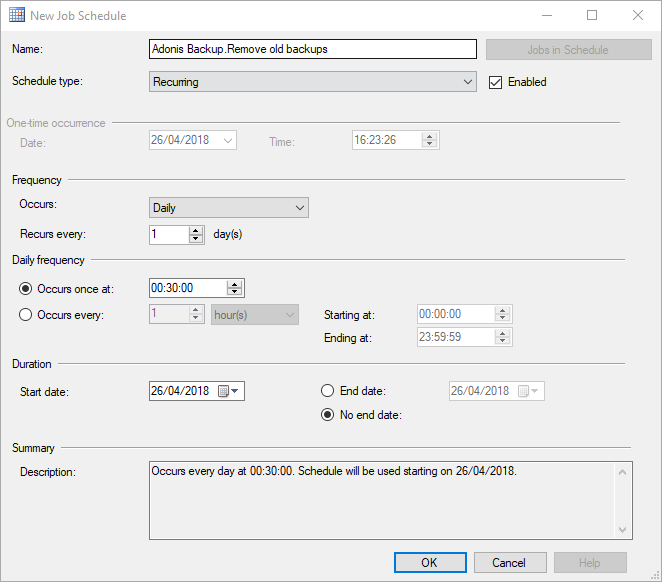
The schedule is set to be recurring and ensure that it is enabled.

Set the frequency to daily and to recur every 1 day

Set it to occur once at 00:30:00

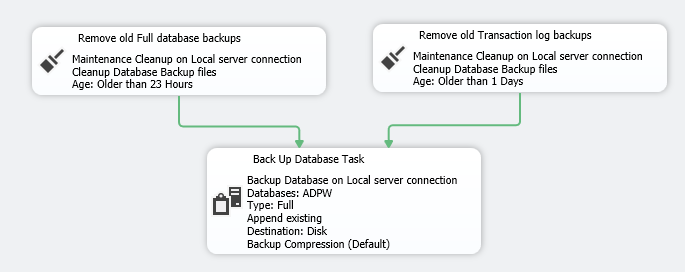
The start date is probably OK as it will default to today’s date. Otherwise set the date that you want to run the first backup.

Ensure that you have selected No end date

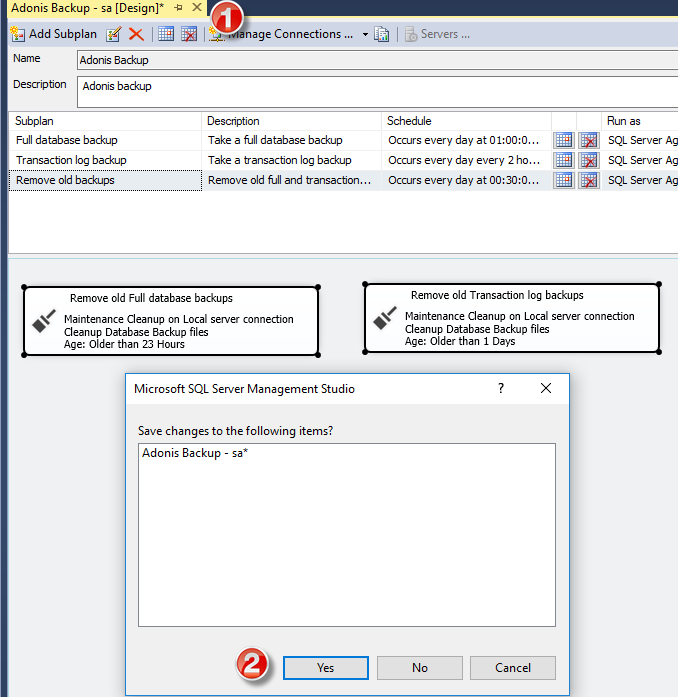


Click OK

As an alternative, you might have added the cleanup tasks to the Full Database Backup subplan. In that case you can select each of the cleanup tasks and drag the arrow to connect it to the backup task. That will ensure that both cleanup tasks are done prior to the backup task.



Click on the X on the maintenance plan tab to close it, and answer Yes to save changes



## Copy to backup media

Use your ordinary backup software to back up the files from the folder in this maintenance plan on a daily basis

The backup we created in the maintenance plan should not be considered as a full backup solution ON IT’S OWN for any disaster recovery situation, because we here stored the backup files on the SQL server itself and only kept them for 1 day. If something would happen to the file system or server, that may also impact the access or integrity of these backup files as well.

The backup files we stored in this maintenance plan together with the content of the Adonis share is sufficient to recover from a disaster, but you need to store them on another media too, in case something happen with the server that make the files unusable/inaccessible. And for a longer retention period then the 1 day we set up in this document.

Include the directory used in this maintenance plan in your daily file backup, to comply with your ordinary backup/recovery policy and to have a sufficient backup solution.

# Appendix

## Restoring the database

In this chapter we provide examples of several restore scenarios.

In all of these examples, we assume you have already connected to the SQL server with SQL Server Management Studio. We also assume that you restore the database on the same server where the backup was made (if not, there may be some additional steps to add the SQL server Logins, if required).

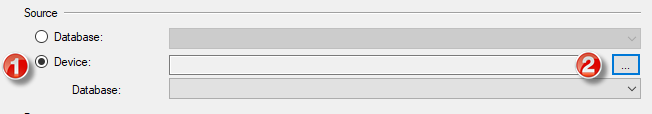
We have also assumed that you want to restore the database as an additional restored database residing next to your original database, just for inspection of the data(or fetching data from a limited set of records), instead of replacing your entire existing database.

### As it was at the time of the full database backup

In this example, we provide instructions on how to restore a full database backup. This make a restore of the database as it was on the time when the full database backup was started.

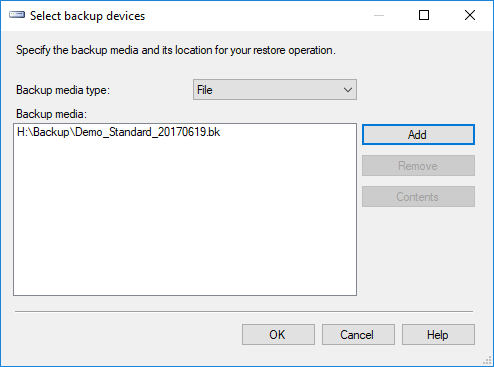
Right click on Databases. Select Restore Database….

In the Source section select Device and click the button with the 3 dots.



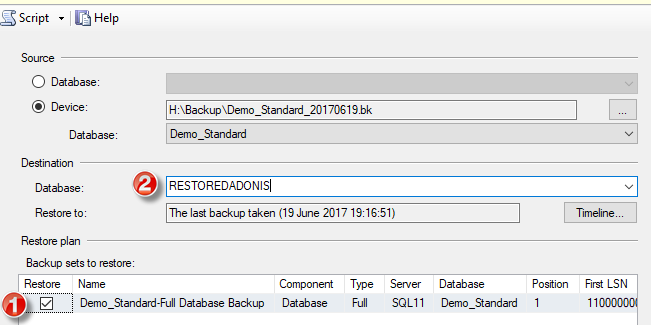
Select File as the backup media type. Click Add.

Browse to the backup file you want to restore and select it. Then click OK

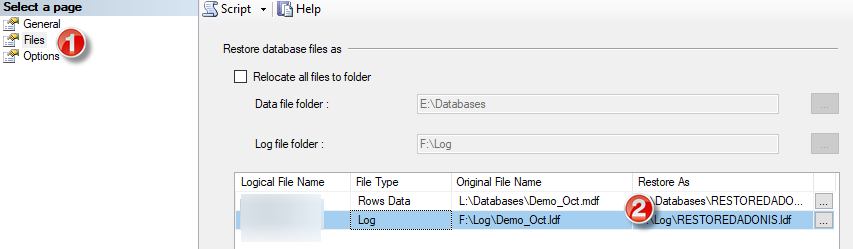


Click OK

In the restore plan section, make sure you have a checkmark on the backup, so it is select for restore. Then in the destination section provide the database name that you want to call the restored database. In this example we used RESTOREDADONIS



Select the Files page to verify the location and name of the database and transaction log files. If you want to store them on a different location or with another name, you may change that in the Restore As column



The file name is not always changed to match what you name the restored database. We always recommend to match the file name with the database name to avoid maintenance errors and incidents.

Select the Options page.

Ensure that the recovery state is set to restore with recovery

Remove the option to make tail log backup before restore

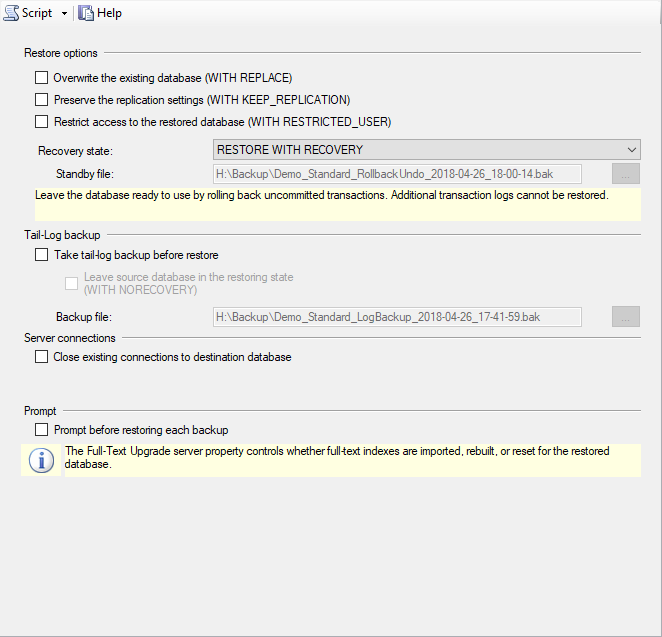
Make sure that the Take tail-log backup before restore option is empty(not set)

We don’t want to make a tail log backup.

Some SQL server editions may in some situations suggest that you make a tail log backup and also change the state of the original database to a restore state, and have these options turned on(checked) by default. See the References chapter for link to more details.

We want to leave the existing database in the same condition as it is currently in, as we are restoring the database as a new database with a different name, and we want to restore the database as it was on the backup time, not as it is right now. Therefore we remove the checkmark for tail log backup if it has been set by default.

Make sure that the overwrite the existing database option is empty(not set). We don’t want to accidentally overwrite any existing database.



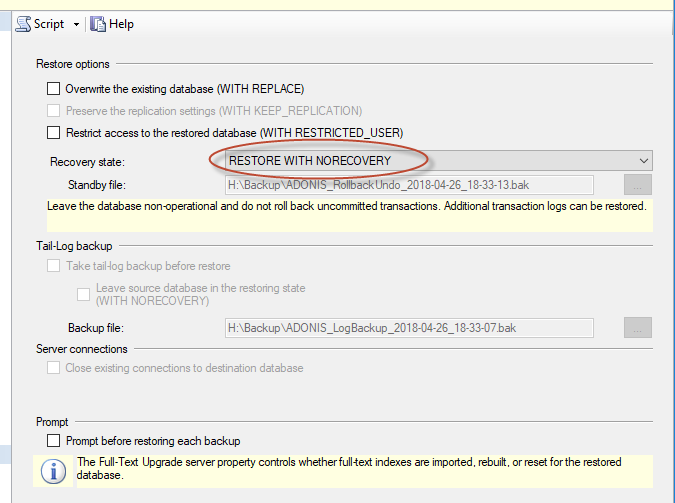
Click OK to start the database restore.

### At a point in time

In this example, we provide instructions on how to restore a database to any given time in the past. Assuming that you have the required full database backup and all the required transaction log backup(s).

For this example, let us make the assumption that you run a full backup every day at 01:00:00 and a transaction log backup every 2 hours from 00:00:00. Your users made an error deleting some records that should not have been deleted at 10:46:42 on the 24. April and you decide you want a restored database as the database was on 24. April at 10:46:00 (to have some slack with regard to the actual delete time).

Here you will first restore your latest full database backup prior to the point in time that you want to restore to, following the same instructions as in the chapter to restore “As it was at the time of the full database backup”, with the exception of the recovery state where we will now select Restore with no recovery

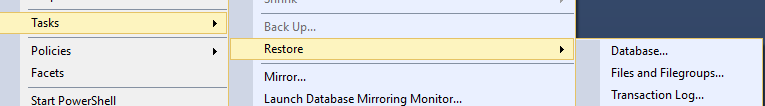


The full backup that you restore is the one made on 24. April at 01:00:00.

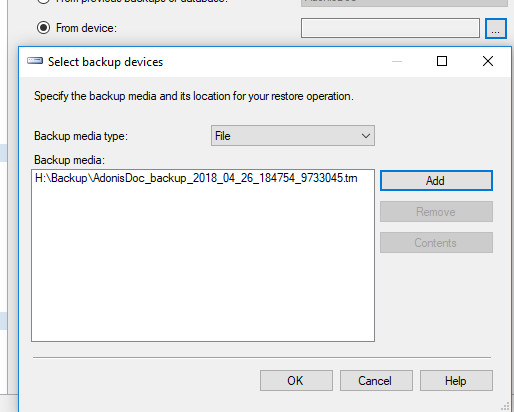
Once you have restored the full backup, you rightclick on the new database name (which is in the restore state)



and select Task>Restore>Transaction log



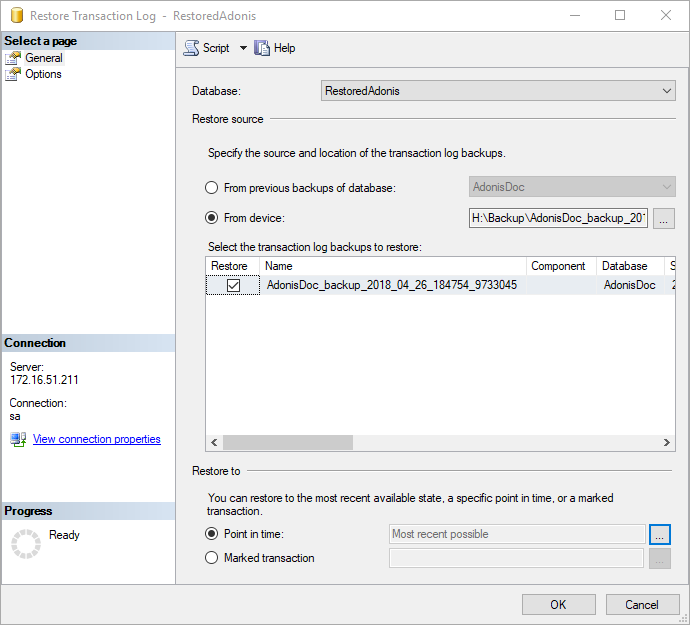
Select from Device, click the button with the 3 dots. Click Add and browse to the transaction log file that was created on 24. April at 02:00:00 (the screenshot bellow show 21. April 18:47:54, but you would select the one at 24. April approximately 02:00)



Click OK

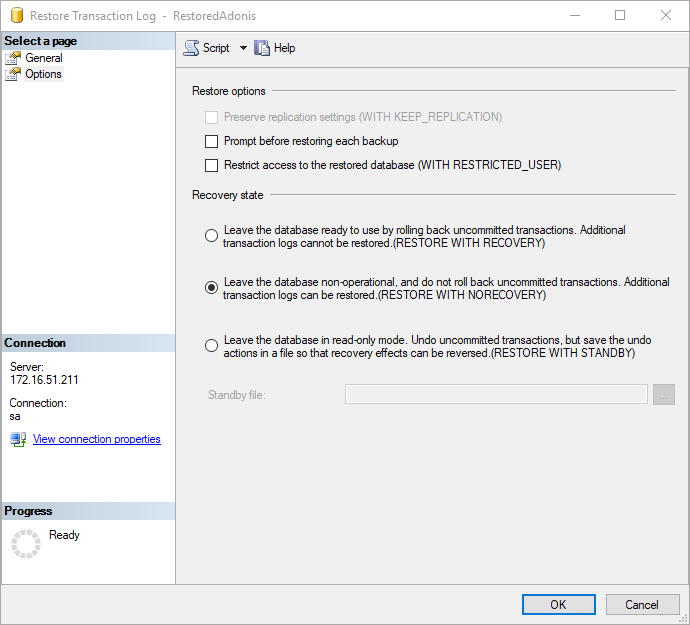
If this is the same server as where the backup was made, the backup history is still stored on the SQL server and you have the transaction log files in the same location as when they where made, you may select “From previous backups of database” instead and then select the name of the original database. Then skip all these steps to restore one transaction log backup at a time, and just provide Point in time to 24. April 10:46:00

Point in time is set to the Most recent possible



Select the Options page.

Make sure that you select the option to leave the database non-operational



We select to leave the database non-operational, since we are going to restore more transaction log backups. When we are restoring the last one, we will choose to leave the database ready to use.

Click OK

Now repeat these steps for the transaction log restore for the transaction log backup made at 04:00, 06:00, 08:00 and 10:00. All of them with the option to leave the database non-operational (RESTORE WITH NORECOVERY).

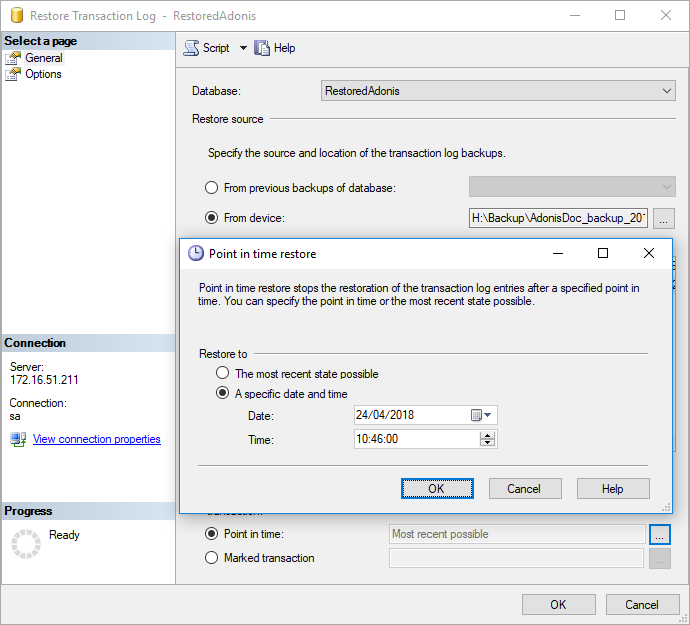
You must run the restores in chronological order

If you click the Script button at the top, before clicking OK to start the transaction log restore, the SQL server management studio will generate the SQL statement used for the restore command. It may be quicker to make a copy this SQL statement for each transaction log backup file, replace the filename in the command, and run these statements, instead of going through these dialogues for every transaction log backup.

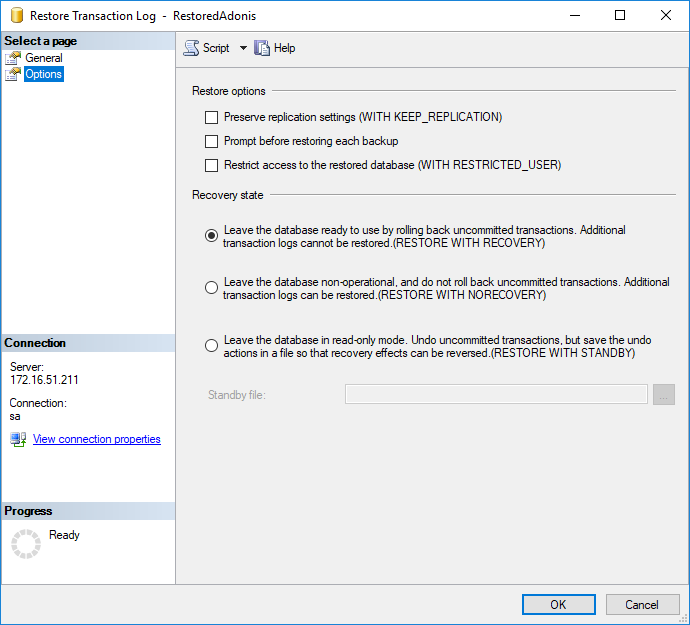
The restore command may look something like

RESTORE LOG [RestoredAdonis] FROM DISK = N'H:\Backup\AdonisDoc\_backup\_2018\_04\_26\_184754\_9733045.trn' WITH FILE = 1, NORECOVERY, NOUNLOAD, STATS = 10

Finally, you will do the same for the transaction log backup from 12:00, but this time you will click on the 3 dots next to the Point in time and select to restore to a specific date and time. Specify 24. April and 10:46:00



Select the options page and this time select the option to leave the database operational (RESTORE WITH RECOVERY)



Click OK

You have now restored the database as it was 24. April 10:46:00

### At a point in time where you miss the latest full database backup prior to the incident

In this example, we provide the instructions on how to restore to a point in time, but the latest full database backup prior to the incident is damaged or missing.

For this example, let us make the assumption that you run a full backup every day at 01:00:00 and a transaction log backup every 2 hours from 00:00:00. Your users made an error deleting some records that should not have been deleted at 10:46:42 on the 24. April and you decide you want a restored database as the database was on 24. April at 10:46:00 (to have some slack with regards to the actual delete time). You then discover that the full database backup on 24. April 01:00:00 is missing or you are unable to restore that backup.

Here you will handle this the exact same way as the chapter “At a point in time”, with the difference that you would here use the full database backup from 23. April 01:00:00 (instead of the one from 24. April), and then the transaction log backups on 23. April at 02:00, 04:00, 06:00, 08:00, 10:00, 12:00, 14:00, 16:00, 18:00, 20:00 followed by the transaction log backups on 24. April 00:00, 02:00, 04:00, 06:00, 08:00 and 10:00, before you finally restore the transaction log backup 24 April at 12:00 to point in time 10:46:00.

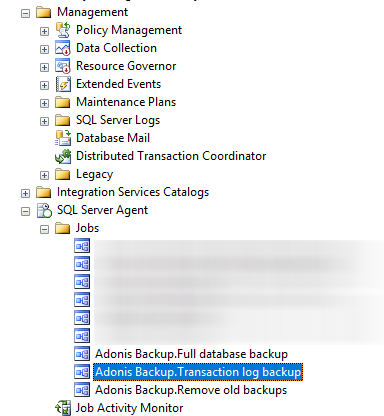
### At a point in time after the latest transaction log backup

In this example, we provide the instruction on how to restore to a point in time, that is after the time when your latest transaction log backup was made.

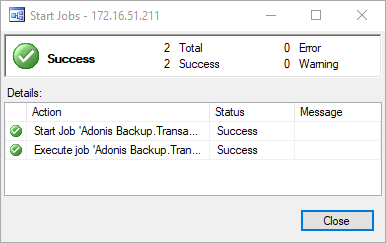
For this example, let us make the assumption that you run a full backup every day at 01:00:00 and a transaction log backup every 2 hours from 00:00:00. Your users made an error deleting some records that should not have been deleted at 10:46:42 on the 24. April and you decide you want a restored database as the database was on 24. April at 10:46:00 (to have some slack with regards to the actual delete time). The time is now only 11:15, so the latest transaction log backup was at 10:00, which is before your desired restore point.

To handle this, you will first manually run a new transaction log backup, to get all the changes past your desired restore point, and then handle it exactly as in the chapter “At a point in time”, with the difference that the last transaction log you restore(the one to point in time) use your manually created transaction log backup at 11:15, instead of the one at 12:00 (which has not been made yet)

To manually run the transaction log backup, you can expand the SQL Server Agent Jobs, Right click on your maintenance plan subplan for Transaction log backup



and select Start Job at Step ….



Click Close when it has completed.

Now follow the instruction in the chapter “At a point in time” to do the restore.

### At a point in time where one of the transaction log backups are damaged or missing

In this example, we provide the instructions with regard to a damaged or missing transaction log backup.

For this example, let us make the assumption that you run a full backup every day at 01:00:00 and a transaction log backup every 2 hours from 00:00:00. Your users made an error deleting some records that should not have been deleted at 10:46:42 on the 24. April and you decide you want a restored database as the database was on 24. April at 10:46:00 (to have some slack with regards to the actual delete time). During the restore, you then discover that the transaction log backup on 24. April 08:00:00 is damaged or missing(the transaction log backup has been made, but is missing), so you are unable to restore that transaction log backup.

In this situation, you will not be able to restore to any time between 24. April 06:00:00 and 25. April 01:00:00 (where you have your next full database backup).

Your option here will be to follow the instruction in the chapter “At a point in time”, with the difference that when you get to the 06:00 transaction log backup you will select to leave the database operational (RESTORE WITH RECOVERY), and will not restore any transaction log backups after the 06:00 transaction log backup.

## References

Tail log backup

<https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/tail-log-backups-sql-server?view=sql-server-2017>