

Microsoft SQL Server Scheduling



Table of Contents

<u>INTRODUCTION</u>	3
<u>MSSQL EXPRESS DATABASE SCHEDULING</u>	3
SQL QUERY FOR BACKUP	3
CREATION OF BATCH FILE	6
CREATE NEW SCHEDULER JOB	6
SELECT THE PROGRAM FOR SCHEDULING	7
SCHEDULER SETTINGS	8
SET USER CREDENTIALS FOR SCHEDULER	9
<u>MSSQL SERVER MAINTENANCE PLANS</u>	9
CREATE A NEW MAINTENANCE PLAN	10
SET THE JOB INFORMATION	11
SELECT THE MAINTENANCE TASKS	11
SPECIFY THE DATABASES	12
RESTORE AUTOMATICALLY	13

Introduction

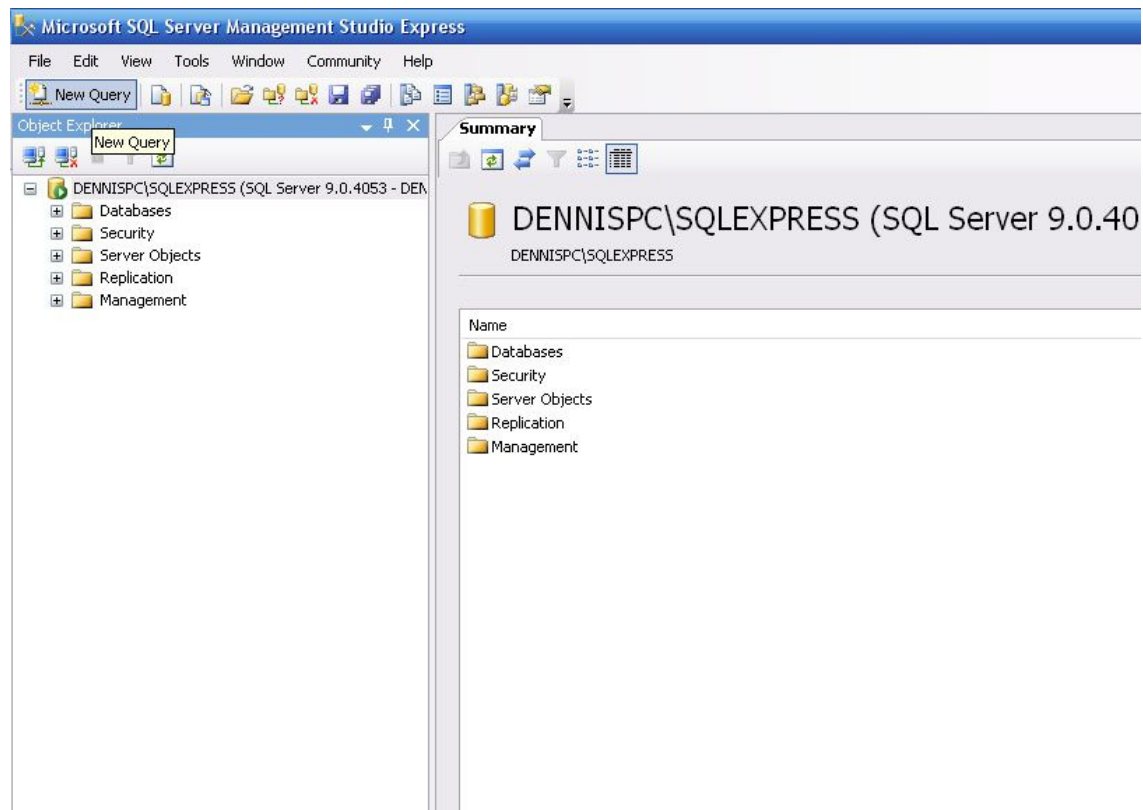
There are different ways in which we can schedule the backup routine. If you are using SQL server full version, you should have something called “Maintenance plans” and if its SQL Server express edition, a scheduled backup can be done by using Windows Scheduler.

MSSQL Express database scheduling

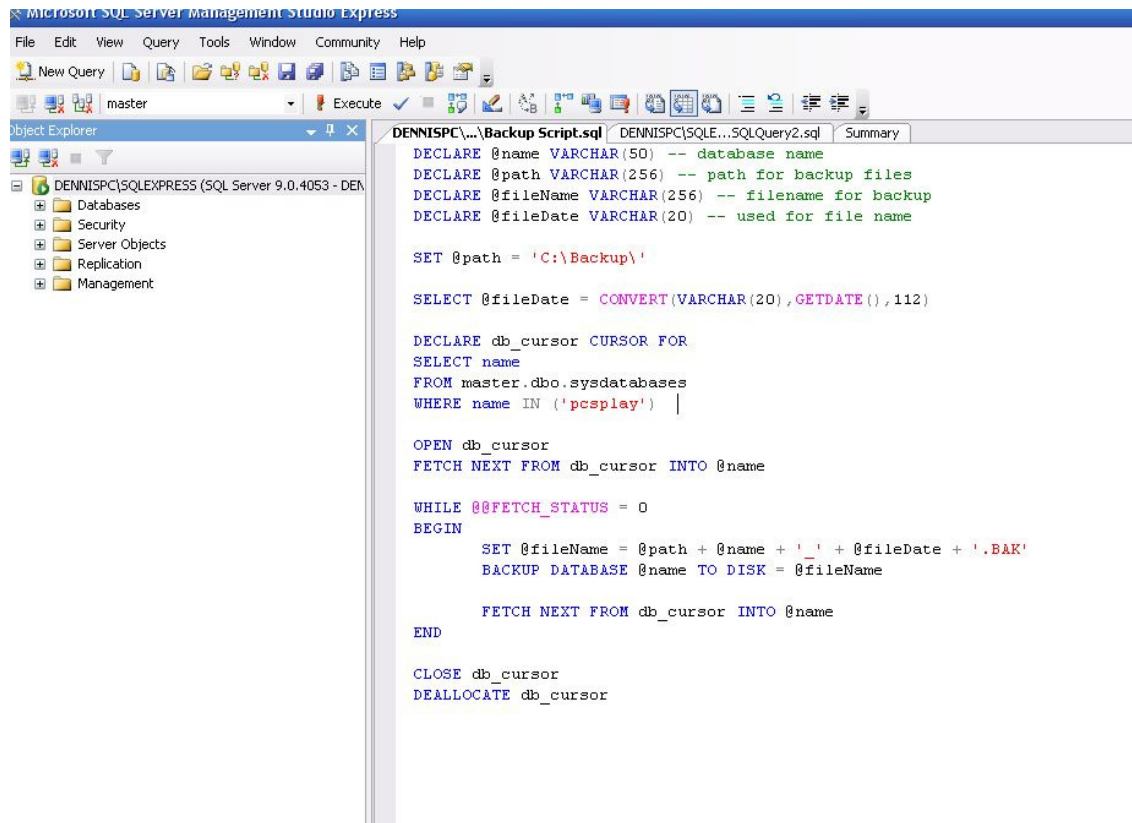
As the express edition does not have the backup scheduling possibility, it can be done by following the steps below.

SQL query for backup

In order to have scheduling, the first step is to create a script. For this purpose, you need to connect to the management studio and click on ‘new query’. This will open up a new window where you can type in the scripts.



A briefing on the script typed below; Initial lines declare few variables and then it sets the path to “c:\backup\”. If you want to change the backup path, you can change it here. Further below the script, we are setting the database to be backed up, in this case, its PCSPlay. If you want to change the database name, you can do it here. After that it’s the actual backup command of SQL Server, which starts with “BACKUP DATABASE”



```
Microsoft SQL Server Management Studio Express
File Edit View Query Tools Window Community Help
New Query | master | Execute
Object Explorer
DENNISPC\SQLEXPRESS (SQL Server 9.0.4053 - DEN
  Databases
  Security
  Server Objects
  Replication
  Management
DENNISPC\...Backup Script.sql | DENNISPC\SQLE...SQLQuery2.sql | Summary
DECLARE @name VARCHAR(50) -- database name
DECLARE @path VARCHAR(256) -- path for backup files
DECLARE @fileName VARCHAR(256) -- filename for backup
DECLARE @fileDate VARCHAR(20) -- used for file name

SET @path = 'C:\Backup\'

SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)

DECLARE db_cursor CURSOR FOR
SELECT name
FROM master.dbo.sysdatabases
WHERE name IN ('pcsplay')

OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @name

WHILE @@FETCH_STATUS = 0
BEGIN
    SET @fileName = @path + @name + '_' + @fileDate + '.BAK'
    BACKUP DATABASE @name TO DISK = @fileName

    FETCH NEXT FROM db_cursor INTO @name
END

CLOSE db_cursor
DEALLOCATE db_cursor
```

(NOTE: See next page for this script in a ‘copy-able’ format.)

Once it’s done, we can run the script by pressing the “Execute” button on the top menu. This may take some time depending on the database size, but will give you an output saying “what was the size of the database” and all.

```

DECLARE @name VARCHAR(50) -- database name
DECLARE @path VARCHAR(256) -- path for backup files
DECLARE @fileName VARCHAR(256) -- filename for backup
DECLARE @fileDate VARCHAR(20) -- used for file name

SET @path = 'C:\Backup\'

SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)

DECLARE db_cursor CURSOR FOR
SELECT name
FROM master.dbo.sysdatabases
WHERE name IN ('pcsplay')

OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @name

WHILE @@FETCH_STATUS = 0
BEGIN
    SET @fileName = @path + @name + '_' + @fileDate + '.BAK'
    BACKUP DATABASE @name TO DISK = @fileName

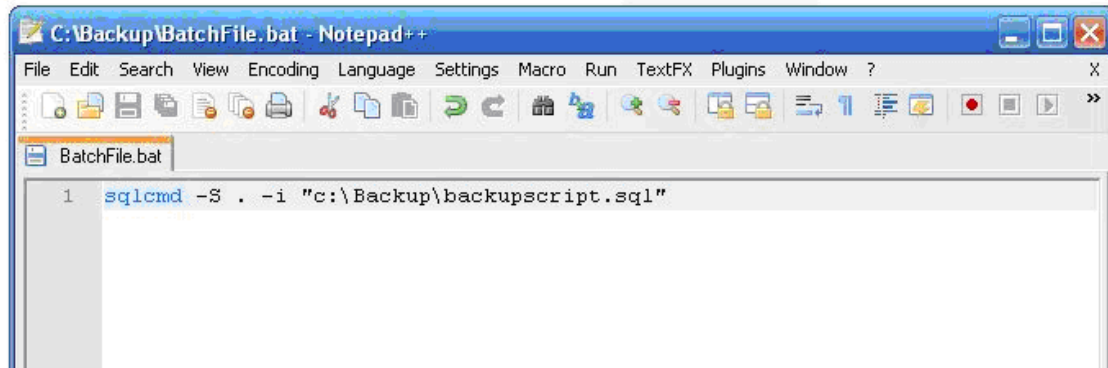
    FETCH NEXT FROM db_cursor INTO @name
END

CLOSE db_cursor
DEALLOCATE db_cursor

```

Creation of batch file

Further we will create a batch file, which we can run in the command mode. SQL Server exposes an object by name "SQLCmd" which can be run on our command window. The option to take the command window is by going to "Start" and go to "Run" and type in "Cmd". You can set the location to where your script is. You can open a notepad and type in the command and save it as ".bat" file.



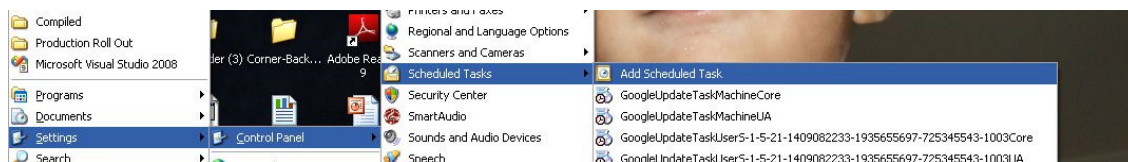
Supplied by a school: A bit of feedback for future users. In the Creation of the Batch File (above). When I ran the batch file it was unable to connect to the SQL server. Instead of "." I had to use ".\SQLEXPRESS". My finished product:

```
sqlcmd -S .\sqlexpress -i "c:\Backup\SQLBackup.sql" -o "c:\Backup\LastLog.txt"
```

LastLog.txt is a log file that overwrites each time the output of the command. It's a quick way for me to go in and see if the last run was successful.

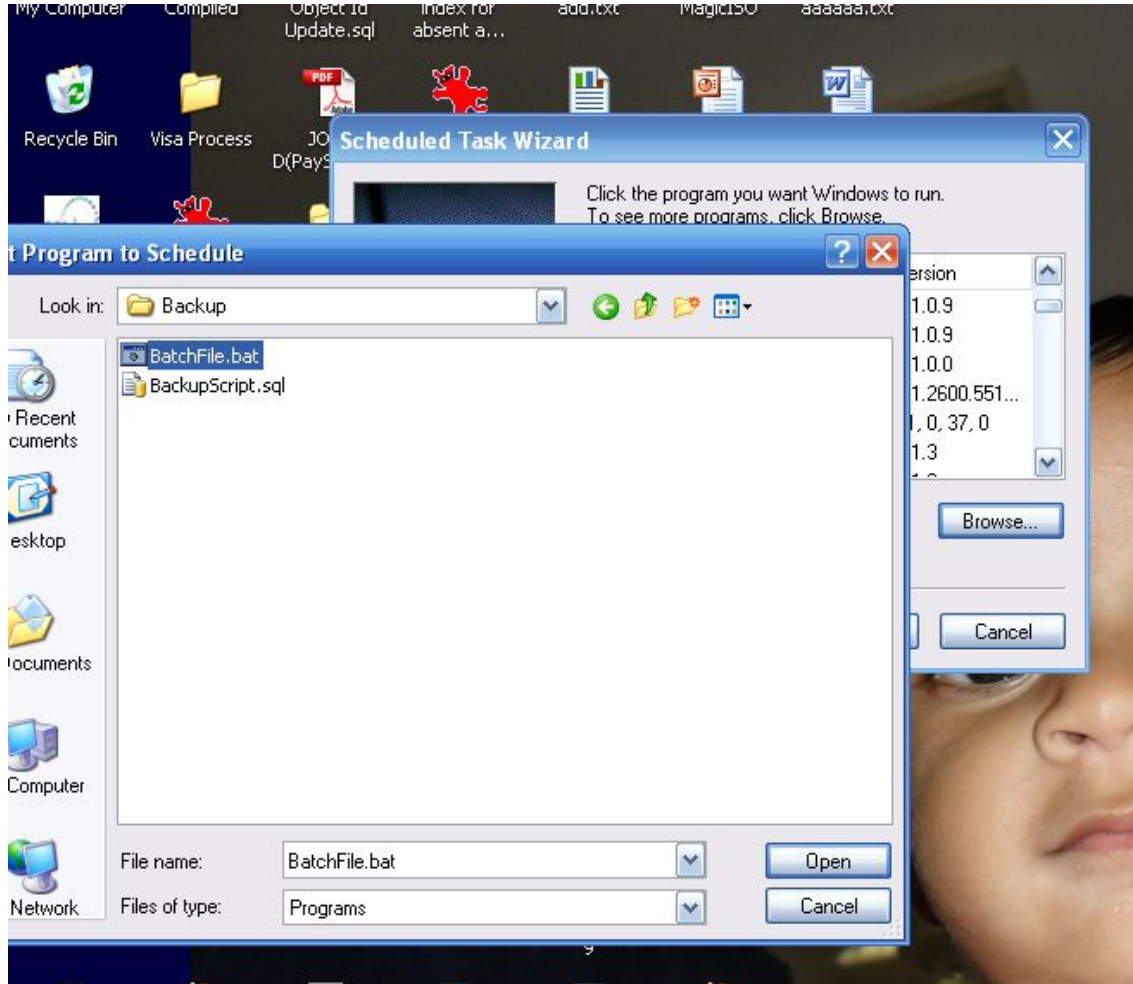
Create new scheduler job

Once it's finished, we can start setting a windows scheduler to run our batch file. Scheduled tasks should be available in control panel similar to below.



Select the program for scheduling

When we add a new scheduled task, we need to select the task which we need to run as a schedule. In this section, as we have created a batch file, we are just pointing it to it.



Scheduler Settings

We need to complete the basic settings for the scheduler, where we need to give it a name, and the frequency at which the program should run.



Set User credentials for scheduler

We need to provide an authenticated user for this to run. Normally this should a powerful user who has permission to run a service in here, the process will then work without fail.

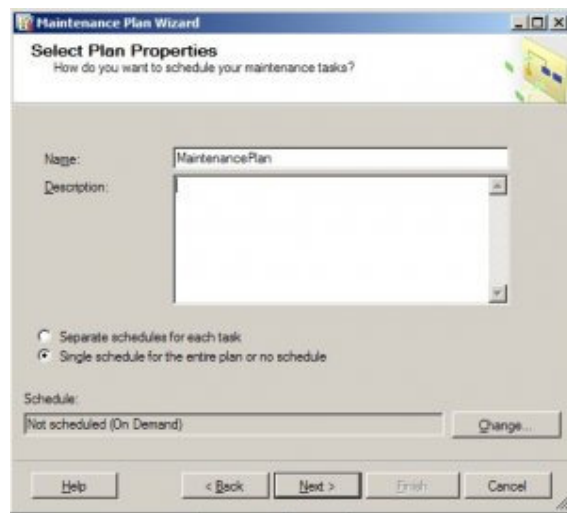
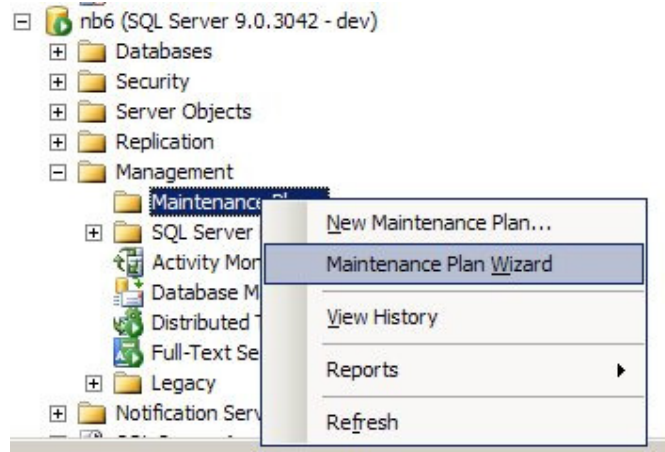


MSSQL Server maintenance plans

SQL Server has an inbuilt provision for maintenance. This can be used for database backups or for doing other tasks like "Rebuilt index", "truncate database" etc. Once a maintenance plan is created, it will create a "Job" under the management studio. This job will be used by the "SQL Agent" in order to get it executed.

Create a new maintenance plan

The provision to create a new maintenance plan is available under the “Management” sub menu inside the Management Studio.



Set the job information

Once we have created a name, we need to define the properties such as if it should be recurring and what frequency is required, this is done in the snapshot below.

New Job Schedule

Name: Jobs in Schedule

Schedule type: Recurring Enabled

One time occurrence

Date: 11/21/2005 Time: 2:22:01 PM

Frequency

Occurs: Weekly

Occurs every: 1 week(s) on

Monday Wednesday Friday Saturday
 Tuesday Thursday Sunday

Daily frequency

Occurs once at: 12:00:00 AM
 Occurs every: 1 hour(s) Starting at: 12:00:00 AM
Ending at: 11:59:59 PM

Duration

Start date: 11/21/2005 End date: 11/21/2005
 No end date

Summary

Description: Occurs every week at 12:00:00 AM. Schedule will be used starting on 11/21/2005.

OK Cancel Help

Select the maintenance tasks

As mentioned previously, the maintenance tasks in SQL Server can be used for a variety of tasks like shrinking the database, rebuilding indexes etc. In our case, we are only selecting the option "Back Up Database (full)".

Maintenance Plan Wizard

Select Maintenance Tasks
Which tasks should this plan perform?

Select one or more maintenance tasks:

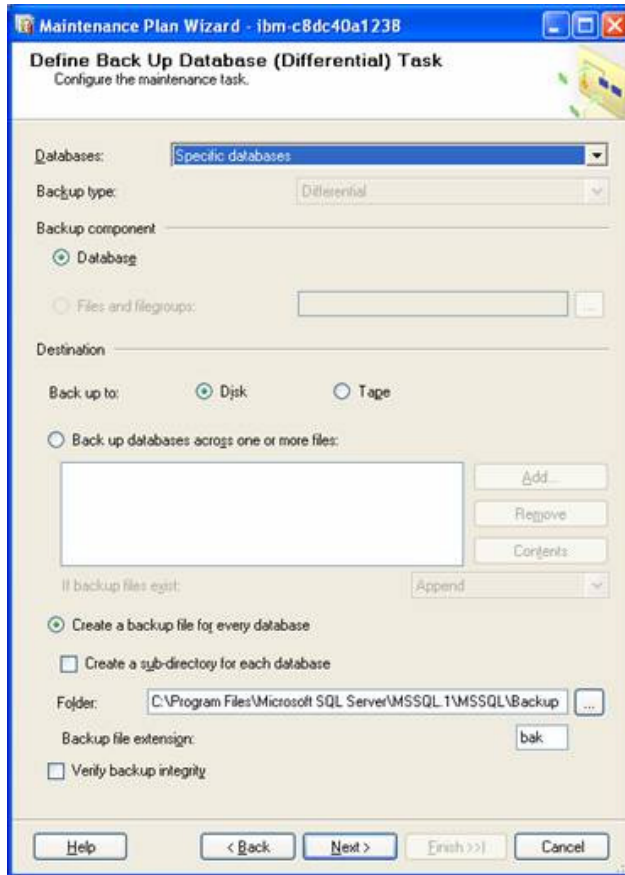
- Check Database Integrity
- Shrink Database
- Reorganize Index
- Rebuild Index
- Update Statistics
- Clean Up History
- Execute SQL Server Agent Job
- Back Up Database (Full)
- Back Up Database (Differential)
- Back Up Database (Transaction Log)
- Maintenance Cleanup Task

The Back Up Database (Full) task allows you to specify the source databases, destination files or tapes, and overwrite options for a full backup.

Help < Back Next > Finish Cancel

Specify the databases

We can specify all databases or a set of databases in the database selection screen after we have selected the tasks. We can also set here where it should be stored physically.



Once its all done, we need to make sure that the sql agent is running (this is a service) and can be viewed either in the “SQL Server Configuration manager” or under the services in the computer.

Restore automatically

As the maintenance plans in SQL server doesn't support restoring objects like in backup, we would be able to try it with SQL scripts as below. It's just a T-SQL script, so either this can be done in the maintenance plan or as a windows scheduler job.

