

Scheduled SQL Express Database Backups

Published: April 17, 2017

Overview

SQL databases are the foundation for many Wonderware software programs. They are constantly updated and changed as projects evolve. Backing these up becomes important in the event of computer hardware failure. Instead of ad-hoc backing these up when remembered, setup a scheduled task and a simple batch file to do this automatically. This applies to all Microsoft SQL Express versions and does not require Wonderware to be present. SQL Standard has its own backup method using Agent which isn't covered in this guide.

Batch File

Microsoft SQL gets installed with a command utility called sqlcmd.exe. While this command has many uses, we'll show the database backup feature.

1. Syntax

```
sqlcmd -S [server name] -Q "backup database [database name] to disk=[filepath]"
```

There are three variables in this command.

1. [server name] – replace with IP or node name of a SQL machine
2. [database name] – name of database as seen in Management Studio

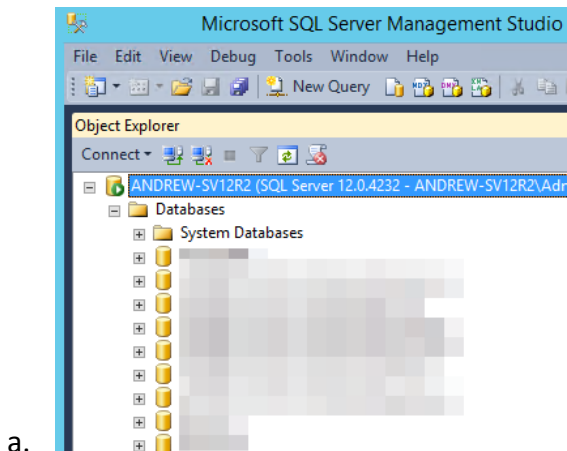


Figure 1 SQL Management Studio

3. [filepath] – This is a drive letter or UNC path down to the file and extension.
 - a. Example: C:\sqlbackups\database.bak



2. Batch

A batch file contains one or many commands to be run one at a time. If we have one or many databases to backup, we'll make a line command for each. This file can be created anywhere on a machine but must be named with a "bat" extension. Editing the file is done through typical text editors like notepad. Here is an example of backing up two databases:

```
sqlcmd -S localhost -Q "backup database Runtime to
disk='C:\sqlbackups\Runtime.bak'"
sqlcmd -S localhost -Q "backup database ProductionGalaxy to
disk='\\server\fileshare\sqlbackups\ProductionGalaxy.bak'"
```

Both of these commands reside in one batch files. Notice the disk filepath has single quotation marks while the entire '-Q' parameter gets double quotation marks.

Let's take this a step further. Each time this runs, it will overwrite the previous file which doesn't give us revisions through time. We'll add date and time to the file which can give us a bit of history.

```
set hr=%time:~0,2%
if "%hr:~0,1%" equ " " set hr=0%hr:~1,1%
set currentdatetime=%date:~-4,4%%date:~-10,2%%date:~-
7,2%_%hr%%time:~3,2%%time:~6,2%
sqlcmd -S localhost -Q "backup database Runtime to
disk='C:\sqlbackups\Runtime_%currentdatetime%.bak'"
```

Now we have a single database backed up to a filename containing date and time. The first 3 lines do not need any modification and give us a full date/time string. The fourth line runs the backup with the date/time expressed in the variable %currentdatetime%.

Task Scheduler

Now we need to run this batch file repeatedly without human interaction. Task Scheduler is a part of Microsoft Windows operating systems and is found in Control Panel > Administrative Tools. Right click onto Task Schedule Library and create a basic task. Design the schedule to your needs. When you arrive at the action step, select 'start a program' and browse for the batch file we created. Ignore the 'add arguments' and 'start in' fields.

To keep this automated such that no user must be logged in, modify the task so that 'run this task whether user is logged in or not' and type the user's password.

