MSSQLTips Free SQL Server Conference Registration + Travel Expenses click to learn more

Simple script to backup all SQL Server databases

Got Tempdb performance issues?
Quickly identify and resolve them with SQL Diagnostic Manager.
START MY FREE TRIAL
IDERA

Free Webcast - Simplify SQL Server Administration

Problem

Sometimes things that seem complicated are much easier then you think and this is the power of using T-SQL to take care of repetitive tasks. One of these tasks may be the need to backup all databases on your server. This is not a big deal if you have a handful of databases, but I have seen several servers where there are 100+ databases on the same instance of SQL Server. You could use SQL Server Management Studio to backup the databases or even use Maintenance Plans, but using T-SQL is a much simpler and faster approach.

Solution

With the use of T-SQL you can generate your backup commands and with the use of cursors you can cursor through all of your databases to back them up one by one. This is a very straight forward process and you only need a handful of commands to do this.

Steps to Backup All Server Databases

- Specify path to store database backups
- Specify backup file name format
- Select list of databases to backup
- Loop through list of databases
- Programmatically create database backup command using database name, path and file name format
- See attached SQL Server backup script

File Naming Format DBname_YYYYDDMM.BAK

Here is the script that will allow you to backup each database within your instance of SQL Server. You will need to change the **@path** to the appropriate backup directory.

```
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

-- specify database backup directory
SET @path = 'C:\Backup\'

-- specify filename format
SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)

DECLARE db_cursor CURSOR READ_ONLY FOR
SELECT name
FROM master.dbo.sysdatabases
WHERE name NOT IN ('master','model','msdb','tempdb')  -- exclude these databases

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
```

File Naming Format DBname_YYYYDDMM_HHMMSS.BAK

If you want to also include the time in the filename you can replace this line in the above script:

```
-- specify filename format
SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)
```

with this line:

```
-- specify filename format
SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112) + REPLACE(CONVERT(VARCHAR(20),GETDATE(),108),':','')
```

Notes

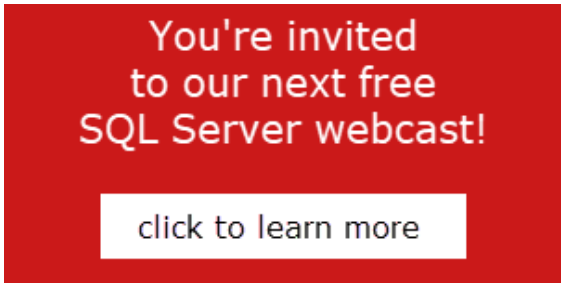
In this script we are bypassing the system databases, but these could easily be included as well. You could also change this into a stored procedure and pass in a database name or if left NULL it backups all databases. Any way you choose to use it, this script gives you the starting point to simply backup all of your databases.

Also, if you wanted to bypass some of your user databases you can include them in the NOT IN section as well.

Next Steps

- Add this script to your toolbox
- Modify this script and make it a stored procedure to include one or many parameters
- Enhance the script to use additional BACKUP options

Last Update: 2017-03-14



About the author

Greg Robidoux is the President of Edgewood Solutions and a co-founder of MSSQLTips.com.

View all my tips

Related Resources

- Simple script to backup all SQL Server databases...
- Automating Transaction Log Backups for All SQL Ser...
- Backup and Restore SQL Server databases programmat...
- Differential Database Backups for SQL Server...
- More SQL Server DBA Tips...

Post a comment or let the author know this tip helped.

All comments are reviewed, so stay on subject or we may delete your comment. Note: your email address is not published. Required fields are marked with an asterisk (*).

*Name *Email Notify for updates

*** NOTE *** - If you want to include code from SQL Server Management Studio (SSMS) in your post, please copy the code from SSMS and paste the code into a text editor like NotePad before copying the code below to remove the SSMS formatting.

p

Send me SQL tips:

I'm not a robot

reCAPTCHA
Privacy - Terms

Save Comment

Monday, March 19, 2018 - 12:00:53 PM - Greg Robidoux Back To Top

Hi pabbhi,

you can just change line in the code above to do copy only backups.

BACKUP DATABASE @name TO DISK = @fileName

to

BACKUP DATABASE @name TO DISK = @fileName WITH COPY_ONLY

-Greg

Monday, March 19, 2018 - 11:18:59 AM - pabbhi Back To Top

how to take copyonly backup for all databases

Saturday, March 03, 2018 - 7:29:05 AM - Greg Robidoux Back To Top

Hi RPi80

You could do this without the cursor and use a loop like Dave did in the script below. Both approaches will work.

The cursor won't damage any data in the database, but the problem you could have with cursors is that data can be updated by other processes and therefore cause an issue with the cursor data (like skipping a database). This is why this was changed to a read only cursor to make sure the data stays consistent, plus we are not planning on updating the data in the cursor so this is a better option. Also, since the backups will take some time to complete the read only option makes more sense. You could also look at other cursor options like static, forward_only, etc.

Take a look at these other articles for more information about how cursors work and the options:

https://sqlperformance.com/2012/09/t-sql-queries/cursor-options

https://sqlperformance.com/2015/09/t-sql-queries/cursor-options-2

https://docs.microsoft.com/en-us/sql/t-sql/language-elements/declare-cursor-transact-sql

-Greg

Saturday, March 03, 2018 - 5:47:04 AM - RPi80 Back To Top

Hi All!

@Greg Robidoux Great script!
I'm not familiar with TSQL as good as You, but this CURSOR (w/out READ_ONLY) option can anyhow damage any data in database while backup script is running? What is the difference between CURSOR and CURSOR READ_ONLY?

What You think about Dave's script (w/out using CURSOR at all)?

Robert

Thursday, February 15, 2018 - 1:23:31 AM - Ahsan Back To Top

how can i create a folder datewise through above statements and on that datefolder my backup should be placed

Friday, February 09, 2018 - 8:11:46 AM - Greg Robidoux Back To Top

Hi Ranu,

try converting things to NVARCHAR as follows:

DECLARE @name NVARCHAR(100) -- database name
DECLARE @path NVARCHAR(512) -- path for backup files
DECLARE @fileName NVARCHAR(512) -- filename for backup
DECLARE @fileDate NVARCHAR(40) -- used for file name

```
-- specify database backup directory
SET @path = 'C:\Backup\'

-- specify filename format
SELECT @fileDate = CONVERT(NVARCHAR(40),GETDATE(),112)

DECLARE db_cursor CURSOR READ_ONLY FOR
SELECT name
FROM master.dbo.sysdatabases
WHERE name NOT IN ('master','model','msdb','tempdb')  -- exclude these databases

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
```

Friday, February 09, 2018 - 1:24:54 AM - ranu gupta Back To Top

I work on same code. But I got a error "Msg 8114, Level 16, State 12, Line 16

Error converting data type varchar to nvarchar."

Wednesday, January 31, 2018 - 1:33:46 AM - Compressed Back To Top

Add WITH COMPRESSION to the end of BACKUP -command to ensure your backups doesn't eat all your diskspace.

Sunday, January 21, 2018 - 8:40:26 AM - Greg Robidoux Back To Top