Taking Scheduled Backups in SQL Express

Background

The purpose of this document is to walk the user through the process of scheduling SQL backups in SQL Express in conjunction with Windows Task Scheduler. This document is for instruction purposes only, and it remains the user’s responsibility to confirm that backups are functional and being taken periodically. To ensure that data can be fully recovered in the event of a server crash, backing up the SQL database is one of the most important tasks that must be done after installing either SOLIDWORKS PDM Professional or Standard. With PDM Professional, the SQL server management studio for SQL Standard includes the option to create maintenance plans that can be scheduled to run at preset times. With PDM Standard, the SQL server management studio for SQL Express does not include the option for maintenance plans. This does not mean that SQL backups cannot be scheduled to run but that more work may be required to setup these scheduled backups.

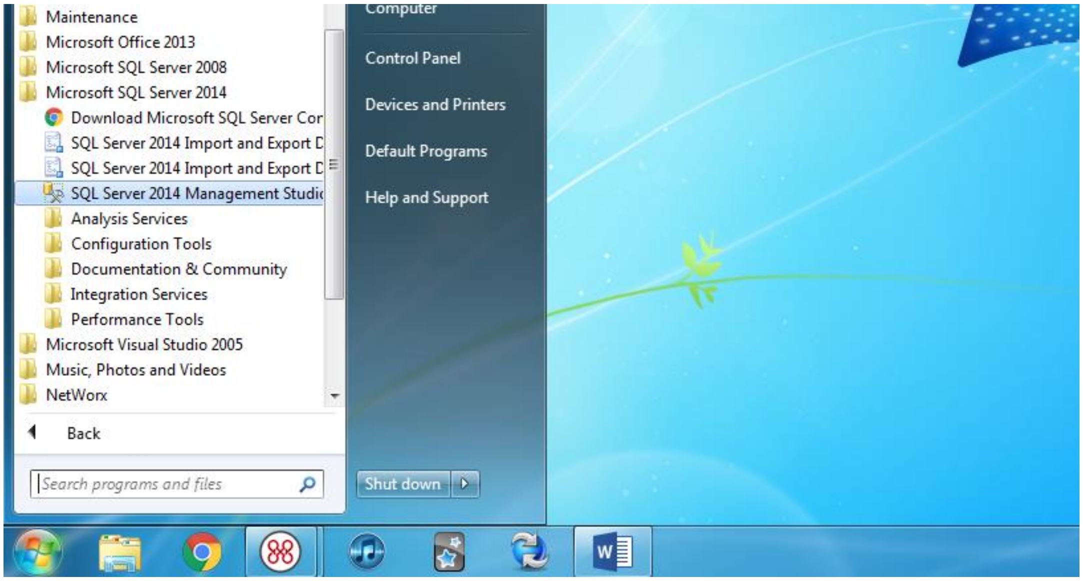
Programs Required

1. SQL Server Management Studio
2. Windows Task Scheduler
3. Text editor (e.g. Notepad)
4. Command Prompt

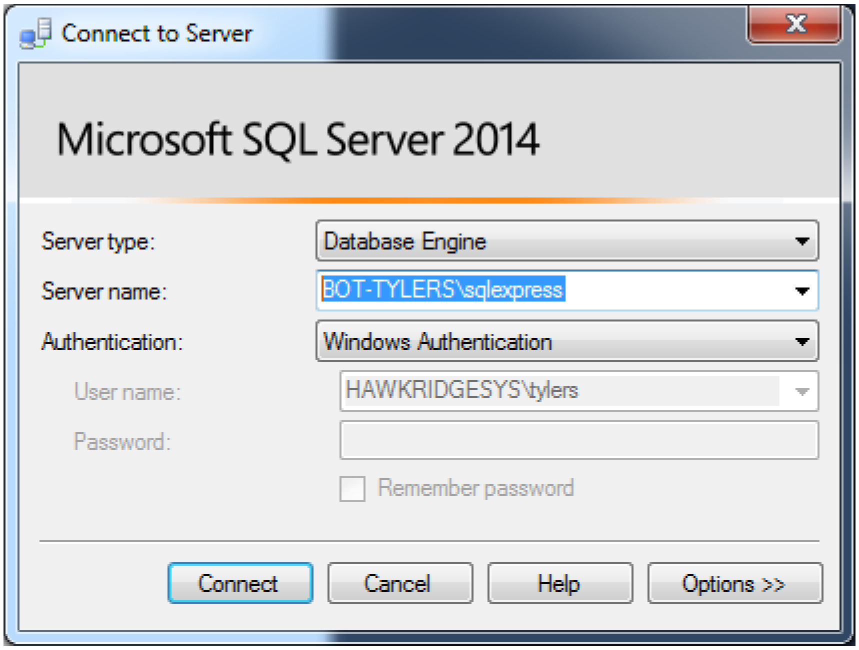
Procedure

Section 1 – Backup Query Creation

1. Open the SQL Server Management Studio by going to the Start Menu > All Programs > Microsoft SQL Server 20XX > SQL Server 20XX Management Studio


Maintenance
Microsoft Office 2013
Microsoft SQL Server 2008
Microsoft SQL Server 2014
Download Microsoft SQL Server Cor
SQL Server 2014 Import and Export D
SQL Server 2014 Import and Export D
SQL Server 2014 Management Studio
Analysis Services
Configuration Tools
Documentation & Community
Integration Services
Performance Tools
Microsoft Visual Studio 2005
Music, Photos and Videos
NetWorx
Back
Search programs and files
Computer
Control Panel
Devices and Printers
Default Programs
Help and Support
Shut down

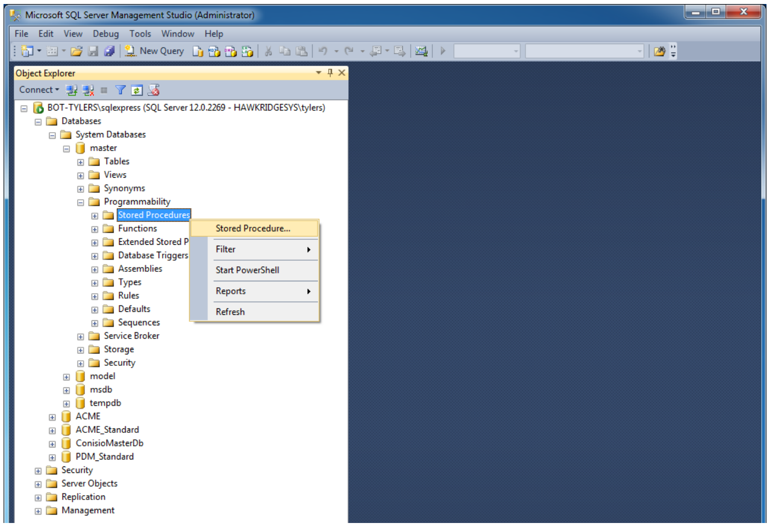
2. When the management studio opens and a login box appears, ensure that the ‘Server name’ box is directed towards the SQL Express instance. For example, [Server Name] \ [SQL Express Instance Name]



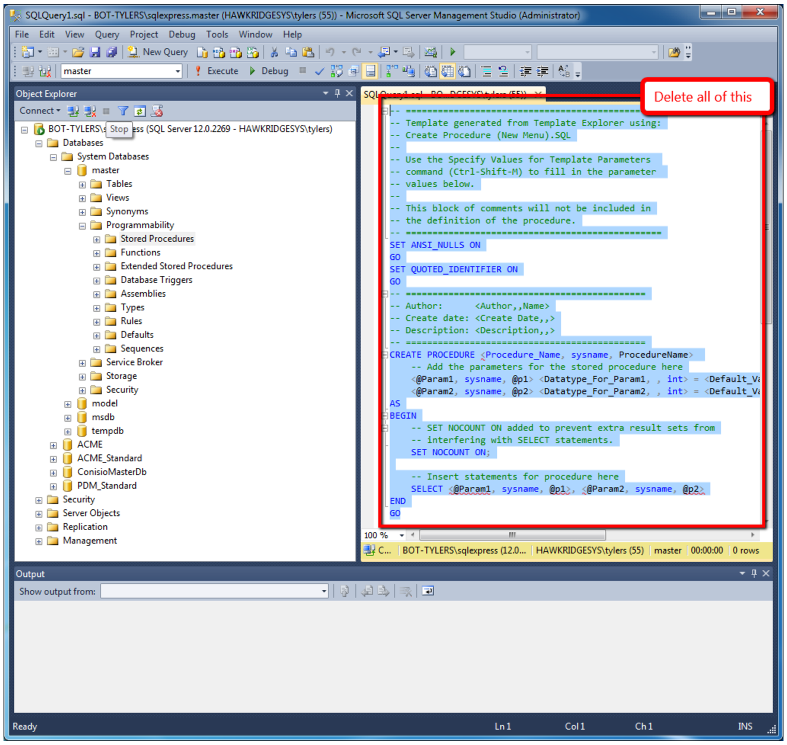
Server type: Database Engine
Server name: BOT-TYLERS\sqlexpress
Authentication: Windows Authentication
User name: HAWKRIDGESYS\tylers
Password:
□ Remember password

If you are using Windows Authentication, a password does not need to be input. If you are using SQL Authentication, you will need the password for the ‘sa’ account.

3. After logging in using the required SQL administrator credentials, locate the ‘Databases’ node under the SQL Express server name in the upper left hand corner of the management studio
4. After expanding the ‘Databases’ node, locate the ‘Stored Procedures’ node by expanding Databases > System Databases > master > Programmability > Stored Procedures
5. Right-click on ‘Stored Procedures’ and select the ‘Stored Procedure…’ option



6. A SQL Query box should appear on the right side of the management studio
7. Delete out any information that initially appears in the query box


SQLQuery1.sql - BOT-TYLERS\sqlexpress.master (HAWKRIDGESYS\tylers (55)) - Microsoft SQL Server Management Studio (Administrator)
Object Explorer
Output
Ready

Delete all of this

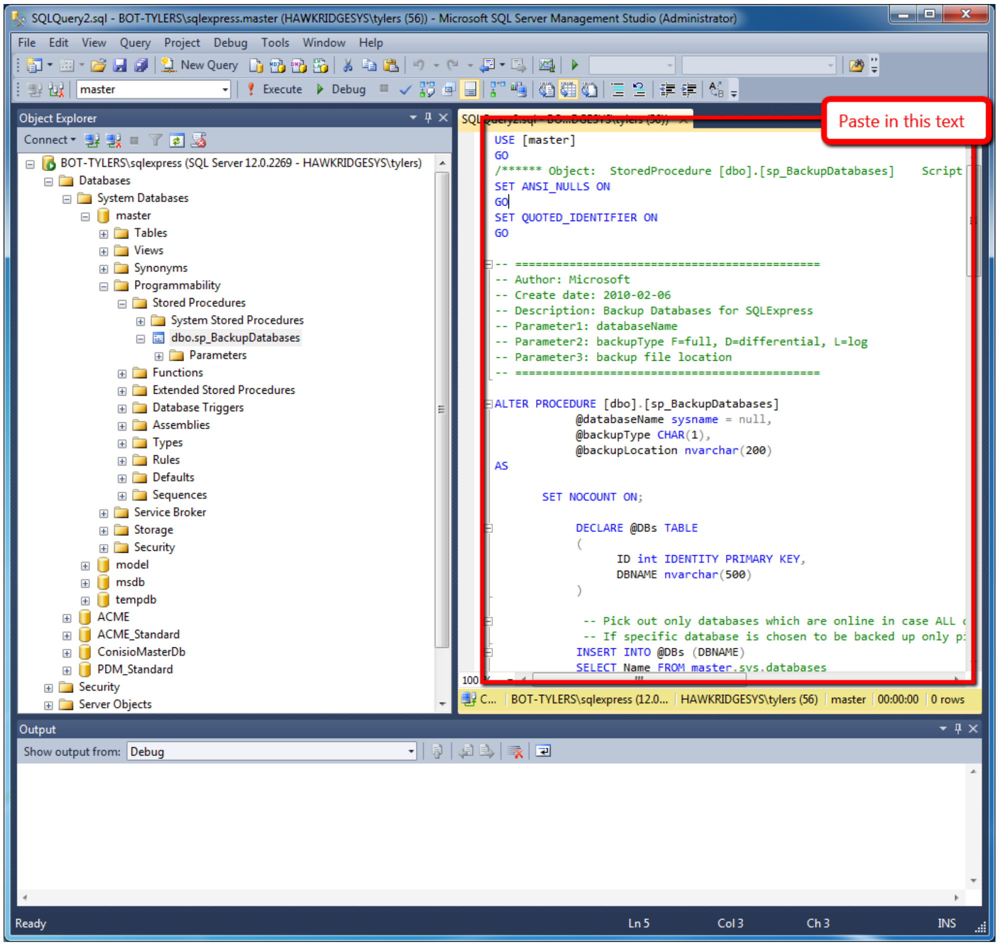
```
-- =============================================
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- =============================================
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE <Procedure_Name, sysname, ProcedureName>
    -- Add the parameters for the stored procedure here
    <@Param1, sysname, @p1> <Datatype_For_Param1, , int> = <Default_Va
    <@Param2, sysname, @p2> <Datatype_For_Param2, , int> = <Default_Va
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
END
GO
```

8. After clearing the query window, paste in the following text:

```
USE [master]
GO
/****** Object:  StoredProcedure [dbo].[sp_BackupDatabases] ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO


-- =============================================
-- Author: Microsoft
-- Create date: 2010-02-06
-- Description: Backup Databases for SQLExpress
-- Parameter1: databaseName
-- Parameter2: backupType F=full, D=differential, L=log
-- Parameter3: backup file location
-- =============================================

CREATE PROCEDURE [dbo].[sp_BackupDatabases]
```

```
            @databaseName sysname = null,
            @backupType CHAR(1),
            @backupLocation nvarchar(200)
AS

       SET NOCOUNT ON;
           DECLARE @DBs TABLE
            (ID int IDENTITY PRIMARY KEY,DBNAME nvarchar(500))
           -- Pick out only databases which are online in case ALL
           -- databases are chosen to be backed up
           -- If specific database is chosen to be backed up only pick
           -- that out from @DBs
            INSERT INTO @DBs (DBNAME)
            SELECT Name FROM master.sys.databases
            where state=0
            AND name=@DatabaseName
            OR @DatabaseName IS NULL
            ORDER BY Name
           -- Filter out databases which do not need to backed up
            IF @backupType='F'
                  BEGIN
                  DELETE @DBs where DBNAME IN ('tempdb')
                  END
            ELSE IF @backupType='D'
                  BEGIN
                  DELETE @DBs where DBNAME IN ('tempdb','master')
                  END
            ELSE IF @backupType='L'
                  BEGIN
                  DELETE @DBs where DBNAME IN ('tempdb','master')
                  END
            ELSE
                  BEGIN
                  RETURN
                  END
           -- Declare variables
            DECLARE @BackupName varchar(100)
            DECLARE @BackupFile varchar(300)
            DECLARE @DBNAME varchar(300)
            DECLARE @sqlCommand NVARCHAR(1000)
        DECLARE @dateTime NVARCHAR(20)
            DECLARE @Loop int
            -- Loop through the databases one by one
            SELECT @Loop = min(ID) FROM @DBs

      WHILE @Loop IS NOT NULL
      BEGIN

-- Database Names have to be in [dbname] format since some have - or _ in
-- their name
      SET @DBNAME = '['+(SELECT DBNAME FROM @DBs WHERE ID = @Loop)+']'

-- Set the current date and time n yyyyhhmmss format
      SET @dateTime = REPLACE(CONVERT(VARCHAR, GETDATE(),101),'/','') +
'_' +  REPLACE(CONVERT(VARCHAR, GETDATE(),108),':','')

-- Create backup filename in path\filename.extension format for full,diff
-- and log backups
      IF @backupType = 'F'
            SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME,
'[',''),']','')+ '_FULL_'+ @dateTime+ '.BAK'
      ELSE IF @backupType = 'D'
            SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME,
'[',''),']','')+ '_DIFF_'+ @dateTime+ '.BAK'
      ELSE IF @backupType = 'L'
            SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME,
'[',''),']','')+ '_LOG_'+ @dateTime+ '.TRN'

-- Provide the backup a name for storing in the media
      IF @backupType = 'F'
            SET @BackupName = REPLACE(REPLACE(@DBNAME,'[',''),']','') +'
full backup for '+ @dateTime
      IF @backupType = 'D'
            SET @BackupName = REPLACE(REPLACE(@DBNAME,'[',''),']','') +'
differential backup for '+ @dateTime
      IF @backupType = 'L'
            SET @BackupName = REPLACE(REPLACE(@DBNAME,'[',''),']','') +'
log backup for '+ @dateTime

-- Generate the dynamic SQL command to be executed

       IF @backupType = 'F'
                  BEGIN
```

```
               SET @sqlCommand = 'BACKUP DATABASE ' +@DBNAME+  ' TO DISK =
'''+@BackupFile+ ''' WITH INIT, NAME= ''' +@BackupName+''', NOSKIP,
NOFORMAT'
                  END
       IF @backupType = 'D'
                  BEGIN
               SET @sqlCommand = 'BACKUP DATABASE ' +@DBNAME+  ' TO DISK =
'''+@BackupFile+ ''' WITH DIFFERENTIAL, INIT, NAME= ''' +@BackupName+''',
NOSKIP, NOFORMAT'
                  END
       IF @backupType = 'L'
                  BEGIN
               SET @sqlCommand = 'BACKUP LOG ' +@DBNAME+  ' TO DISK =
'''+@BackupFile+ ''' WITH INIT, NAME= ''' +@BackupName+''', NOSKIP,
NOFORMAT'
                  END

-- Execute the generated SQL command
       EXEC(@sqlCommand)

-- Goto the next database
SELECT @Loop = min(ID) FROM @DBs where ID>@Loop

END
```

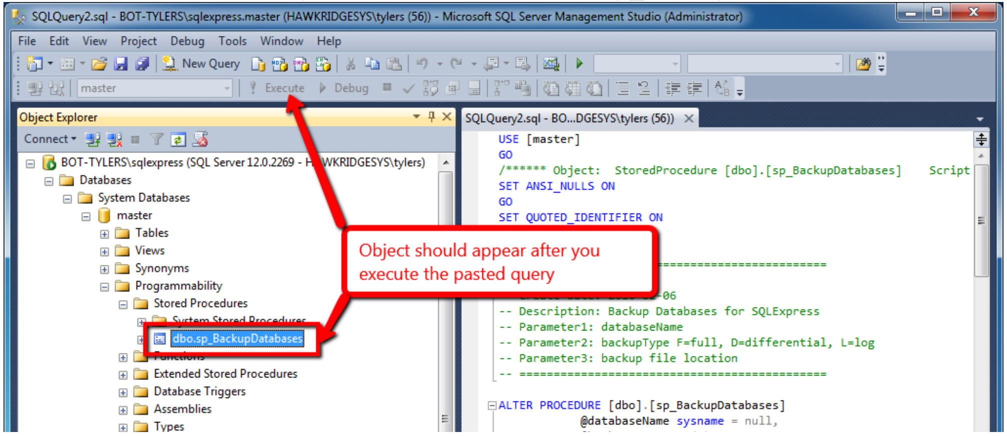


Paste in this text

```
USE [master]
GO
/****** Object:  StoredProcedure [dbo].[sp_BackupDatabases]    Script
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- ============================================
-- Author: Microsoft
-- Create date: 2010-02-06
-- Description: Backup Databases for SQLExpress
-- Parameter1: databaseName
-- Parameter2: backupType F=full, D=differential, L=log
-- Parameter3: backup file location
-- ============================================

ALTER PROCEDURE [dbo].[sp_BackupDatabases]
            @databaseName sysname = null,
            @backupType CHAR(1),
            @backupLocation nvarchar(200)
AS

       SET NOCOUNT ON;

            DECLARE @DBs TABLE
            (
                  ID int IDENTITY PRIMARY KEY,
                  DBNAME nvarchar(500)
            )

             -- Pick out only databases which are online in case ALL d
             -- If specific database is chosen to be backed up only p:
            INSERT INTO @DBs (DBNAME)
            SELECT Name FROM master.sys.databases
```

C... | BOT-TYLERS\sqlexpress (12.0... | HAWKRIDGESYS\tylers (56) | master | 00:00:00 | 0 rows

9. After pasting this text into the query window, save the query and select the execute button near the upper right
10. Under 'Stored Procedures' you should now see 'dbo.sp_BackupDatabases'

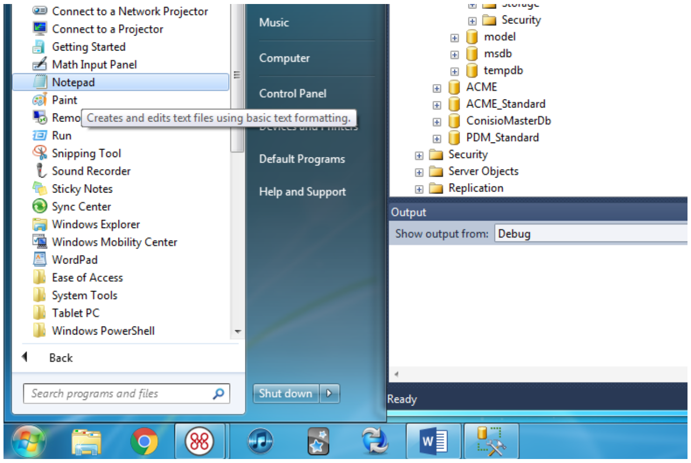

SQLQuery2.sql - BOT-TYLERS\sqlexpress.master (HAWKRIDGESYS\tylers (56)) - Microsoft SQL Server Management Studio (Administrator)
File Edit View Project Debug Tools Window Help
New Query
master
Execute
Debug
Object Explorer
Connect
BOT-TYLERS\sqlexpress (SQL Server 12.0.2269 - HAWKRIDGESYS\tylers)
Databases
System Databases
master
Tables
Views
Synonyms
Programmability
Stored Procedures
System Stored Procedures
dbo.sp_BackupDatabases
Functions
Extended Stored Procedures
Database Triggers
Assemblies
Types
SQLQuery2.sql - BO...DGESYS\tylers (56))

Object should appear after you execute the pasted query

```
USE [master]
GO
/****** Object:  StoredProcedure [dbo].[sp_BackupDatabases]    Script
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

                   ========================
                   -06
-- Description: Backup Databases for SQLExpress
-- Parameter1: databaseName
-- Parameter2: backupType F=full, D=differential, L=log
-- Parameter3: backup file location
-- ============================================

ALTER PROCEDURE [dbo].[sp_BackupDatabases]
            @databaseName sysname = null,
```

Section 2 – Batch File Creation

1. Open a text editor. For Notepad, it can be found by going to the Start Menu > All Programs > Accessories > Notepad



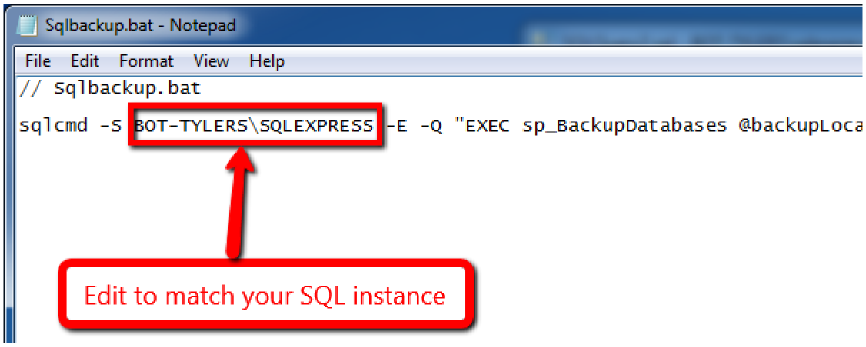
2. In a blank text file, copy the following text:

```
sqlcmd -S [SERVERNAME]\[SQLEXPRESS] -E -Q "EXEC sp_BackupDatabases
@backupLocation='C:\Program Files\Microsoft SQL
Server\MSSQL12.SQLEXPRESS\MSSQL\Backup\’, @databaseName=’[VAULTNAME]’,
@backupType='F'"
```

3. Edit the ‘[SERVERNAME]\[SQLEXPRESS]’ section to match what you input in the ‘Server name’ box during the login to the SQL Server Management Studio
4. Edit the ‘C:\Program Files’ section to match the directory on your machine where the ‘Backup’ folder is stored.

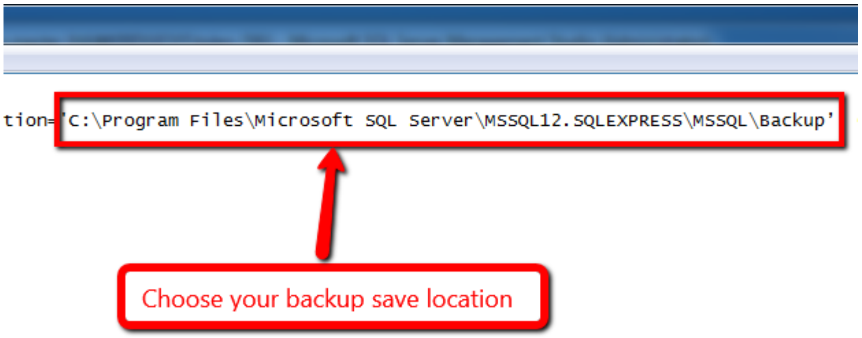
   *Remember that this backup is being stored on the machine where SQL is running, so it will be prudent to create an additional copy in a shared network location on another machine. If you do pursue this option, ensure that the user is logged into the SQL Server Management Studio using Windows authentication and that the user has permission to write files to the shared network location. Additional locations can be specified by creating multiple lines in your batch file, one for each unique location and vault.

5. Edit the ‘[VAULTNAME]’ section to match the vault name that you want to backup. If you have put a space in your vault name, your database name will automatically be renamed SWPDM_[VAULTNAME] with the space removed.



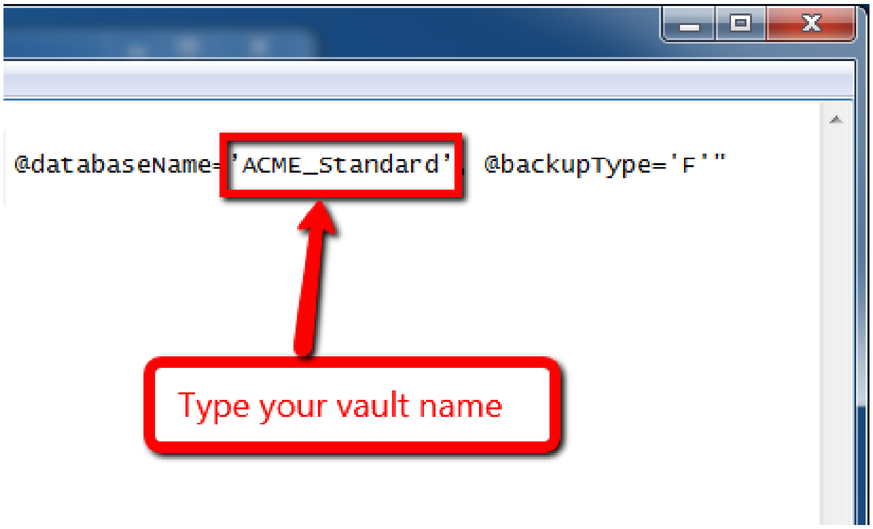
```
// sqlbackup.bat
sqlcmd -S BOT-TYLERS\SQLEXPRESS -E -Q "EXEC sp_BackupDatabases @backupLoca
```

Edit to match your SQL instance



```
tion='C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\Backup'
```

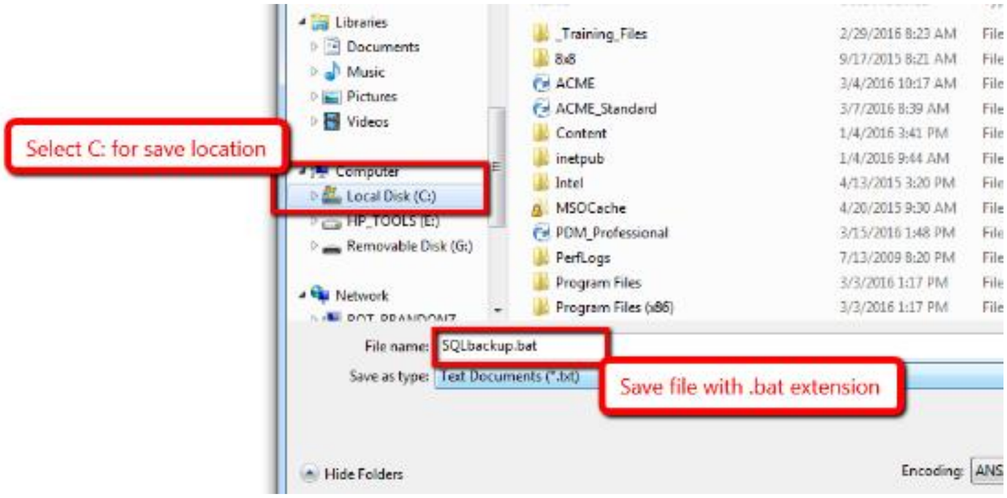
Choose your backup save location



```
@databaseName='ACME_Standard', @backupType='F'"
```

Type your vault name

6. Save the file as 'SQLbackup.bat' on the root level of your C: drive



_Training_Files	2/29/2016 8:23 AM	File
8x8	9/17/2015 8:21 AM	File
ACME	3/4/2016 10:17 AM	File
ACME_Standard	3/7/2016 8:39 AM	File
Content	1/4/2016 3:41 PM	File
inetpub	1/4/2016 9:44 AM	File
Intel	4/13/2015 3:20 PM	File
MSOCache	4/20/2015 9:30 AM	File
PDM_Professional	3/15/2016 1:48 PM	File
PerfLogs	7/13/2009 8:20 PM	File
Program Files	3/3/2016 1:17 PM	File
Program Files (x86)	3/3/2016 1:17 PM	File

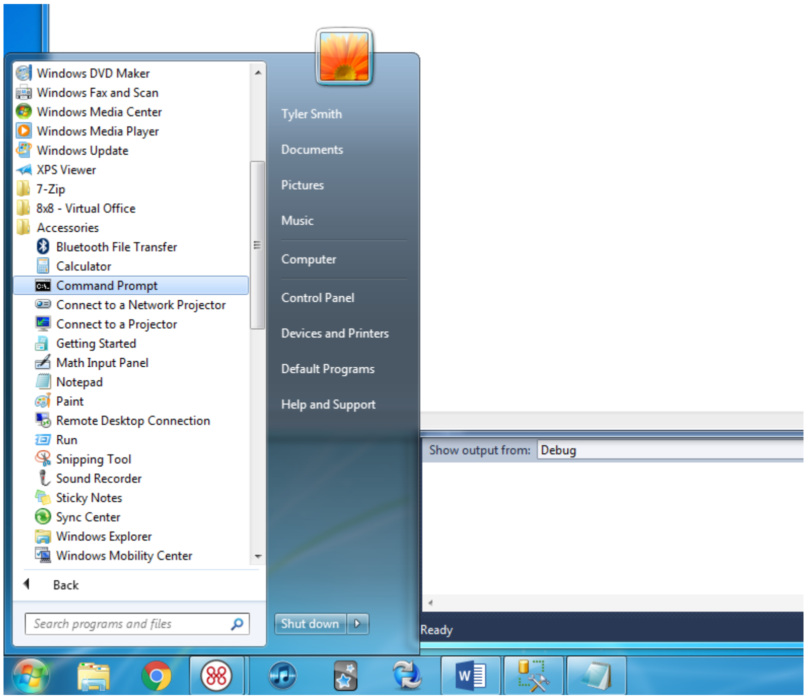
File name: SQLbackup.bat

Save as type: Text Documents (*.txt)

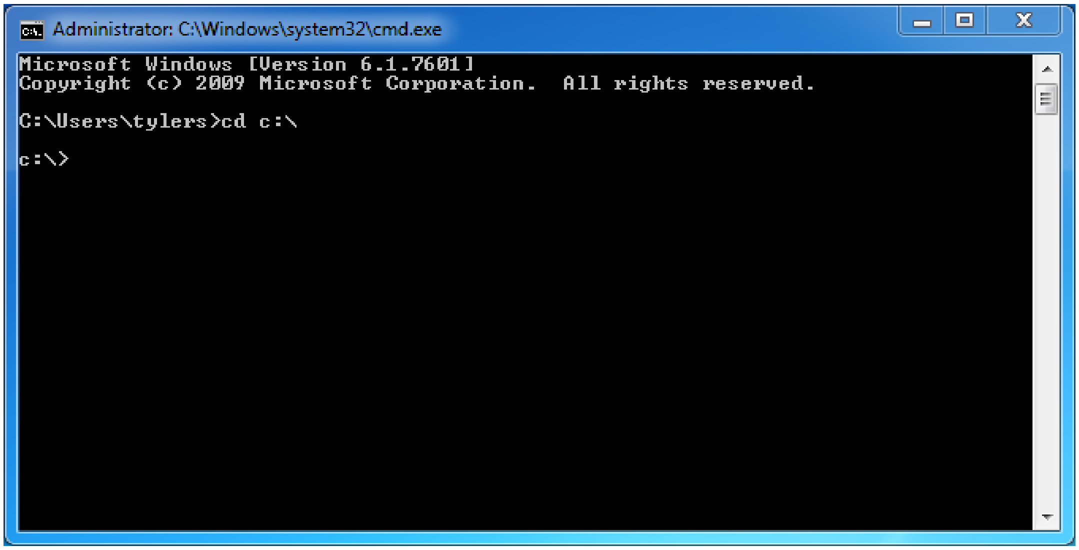
Select C: for save location

Save file with .bat extension

7. Open a command prompt by going to the Start Menu > All Programs > Accessories > Command Prompt

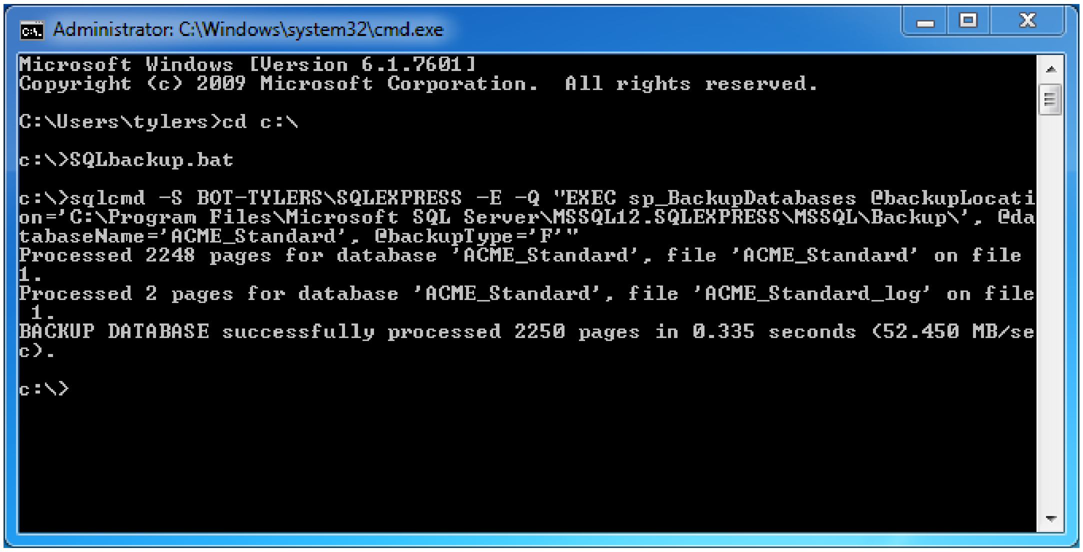


8. Type 'cd c:\' into the command prompt



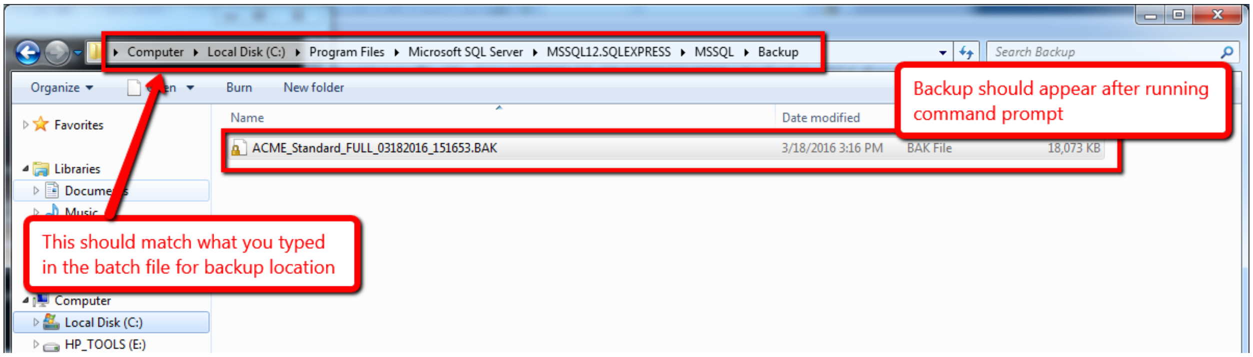
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\tylers>cd c:\

c:\>
```

9. Type 'SQLbackup.bat' and press enter



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\tylers>cd c:\

c:\>SQLbackup.bat

c:\>sqlcmd -S BOT-TYLERS\SQLEXPRESS -E -Q "EXEC sp_BackupDatabases @backupLocati
on='C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\Backup\', @da
tabaseName='ACME_Standard', @backupType='F'"
Processed 2248 pages for database 'ACME_Standard', file 'ACME_Standard' on file
1.
Processed 2 pages for database 'ACME_Standard', file 'ACME_Standard_log' on file
 1.
BACKUP DATABASE successfully processed 2250 pages in 0.335 seconds (52.450 MB/se
c).

c:\>
```

10. Go to the Backups folder in the SQL server backup folder and ensure that a backup file has been placed in that folder



Computer ▸ Local Disk (C:) ▸ Program Files ▸ Microsoft SQL Server ▸ MSSQL12.SQLEXPRESS ▸ MSSQL ▸ Backup

Name	Date modified	Type	Size
ACME_Standard_FULL_03182016_151653.BAK	3/18/2016 3:16 PM	BAK File	18,073 KB

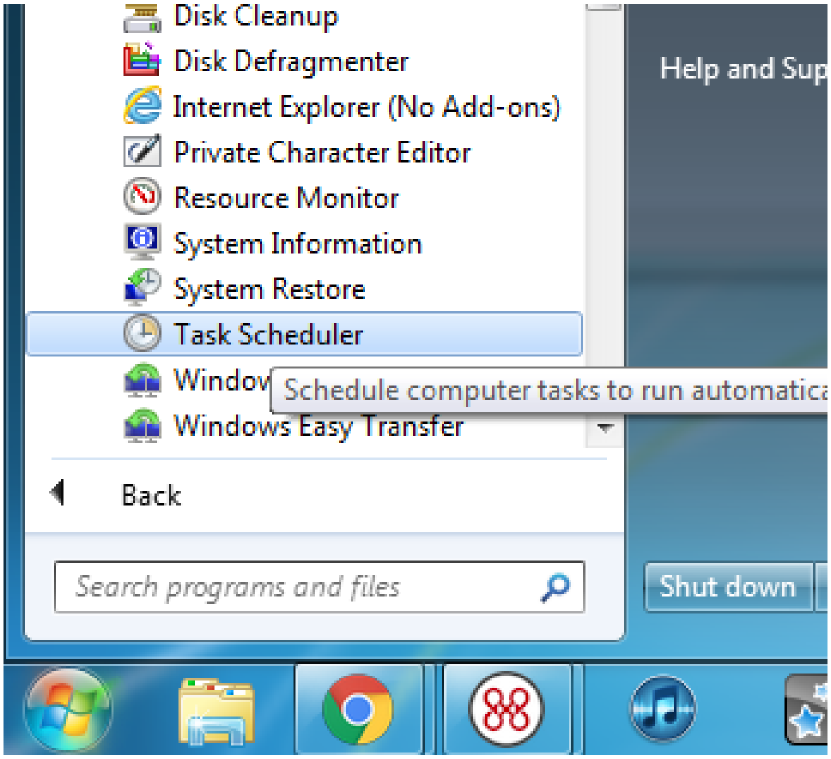
Backup should appear after running command prompt

This should match what you typed in the batch file for backup location

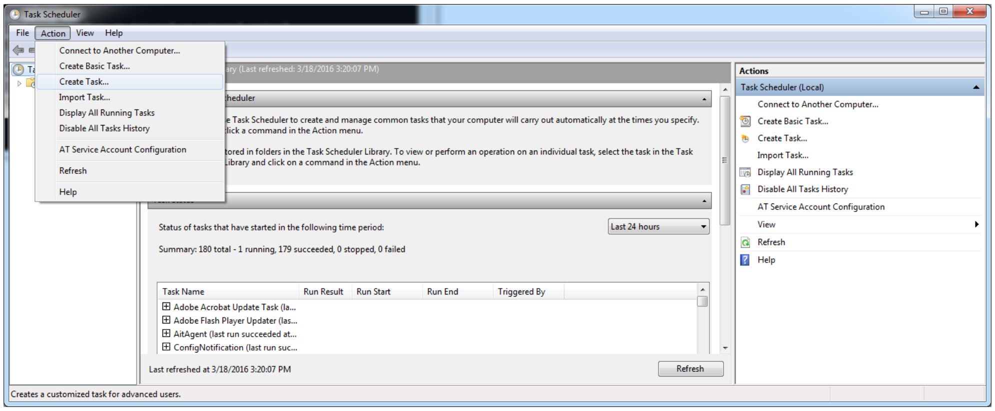
11. If so, proceed to Section 3
12. If not, ensure that the '@backupLocation' portion of your batch file matches the folder that you want to backup to go into

Section 3 – Scheduled Task Creation

1. Open Windows Task Scheduler by going to the Start Menu > All Programs > Accessories > System Tools > Task Scheduler



2. Select ‘Action’ in the top tool bar and ‘Create Task’ in the dropdown


Task Scheduler
File
Action
View
Help
Connect to Another Computer...
Create Basic Task...
Create Task...
Import Task...
Display All Running Tasks
Disable All Tasks History
AT Service Account Configuration
Refresh
Help
Actions
Task Scheduler (Local)
Connect to Another Computer...
Create Basic Task...
Create Task...
Import Task...
Display All Running Tasks
Disable All Tasks History
AT Service Account Configuration
View
Refresh
Help
Last 24 hours
Refresh

ary (Last refreshed: 3/18/2016 3:20:07 PM)

heduler

e Task Scheduler to create and manage common tasks that your computer will carry out automatically at the times you specify. lick a command in the Action menu.

tored in folders in the Task Scheduler Library. To view or perform an operation on an individual task, select the task in the Task Library and click on a command in the Action menu.

Status of tasks that have started in the following time period:

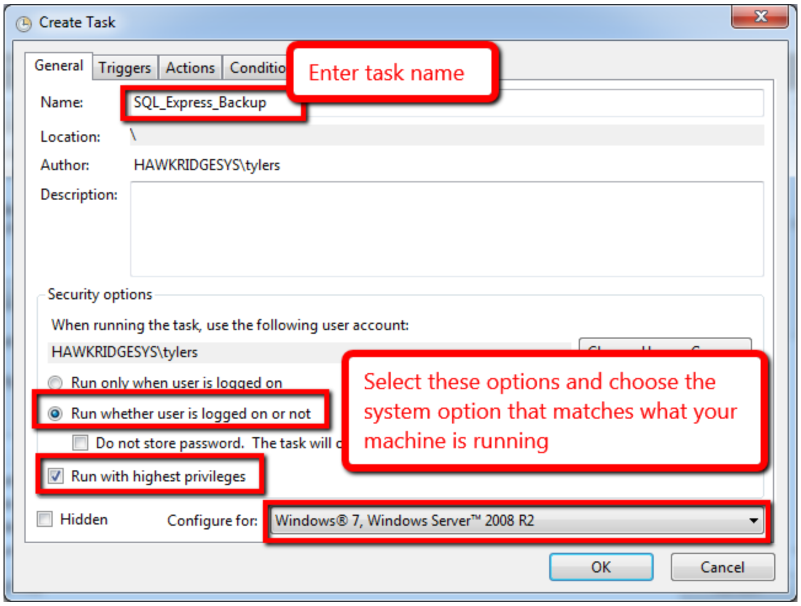
Summary: 180 total - 1 running, 179 succeeded, 0 stopped, 0 failed

Task Name	Run Result	Run Start	Run End	Triggered By
⊞ Adobe Acrobat Update Task (la...				
⊞ Adobe Flash Player Updater (las...				
⊞ AitAgent (last run succeeded at...				
⊞ ConfigNotification (last run suc...				

Last refreshed at 3/18/2016 3:20:07 PM

Creates a customized task for advanced users.

3. Name the action ‘SQL_Express_Backup’ on the ‘General’ tab
4. Also on the ‘General’ tab, select ‘Run whether user is logged on or not’, ‘Run with highest privileges’, and ‘Configure for:’ the applicable system that you are running



Name: SQL_Express_Backup

Location: \

Author: HAWKRIDGESYS\tylers

Description:

Security options

When running the task, use the following user account:

HAWKRIDGESYS\tylers

○ Run only when user is logged on

● Run whether user is logged on or not

□ Do not store password. The task will o

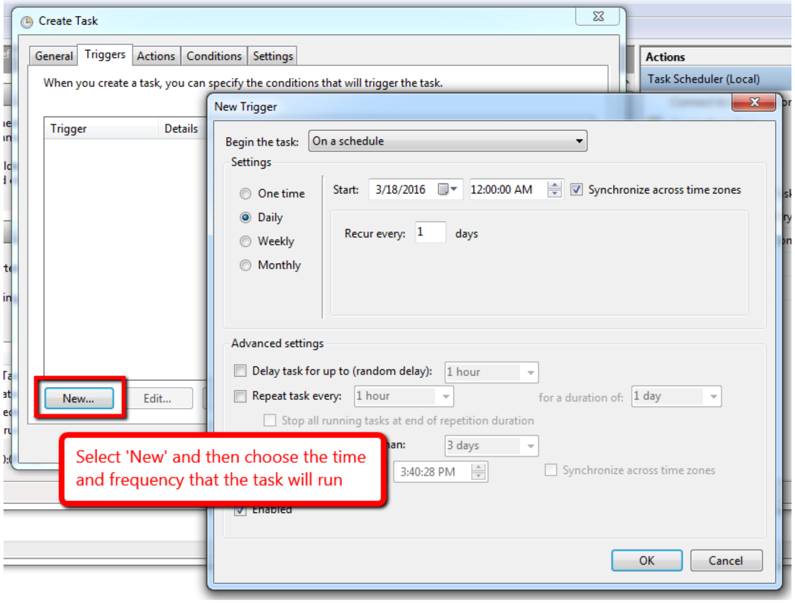
☑ Run with highest privileges

□ Hidden Configure for: Windows® 7, Windows Server™ 2008 R2

Enter task name

Select these options and choose the system option that matches what your machine is running

5. Select the ‘Triggers’ tab and ‘New’
6. Choose when the task should run and how often. It is recommended that you backup daily



When you create a task, you can specify the conditions that will trigger the task.

Begin the task: On a schedule

Settings

○ One time
● Daily
○ Weekly
○ Monthly

Start: 3/18/2016 12:00:00 AM ☑ Synchronize across time zones

Recur every: 1 days

Advanced settings

□ Delay task for up to (random delay): 1 hour

□ Repeat task every: 1 hour for a duration of: 1 day

□ Stop all running tasks at end of repetition duration

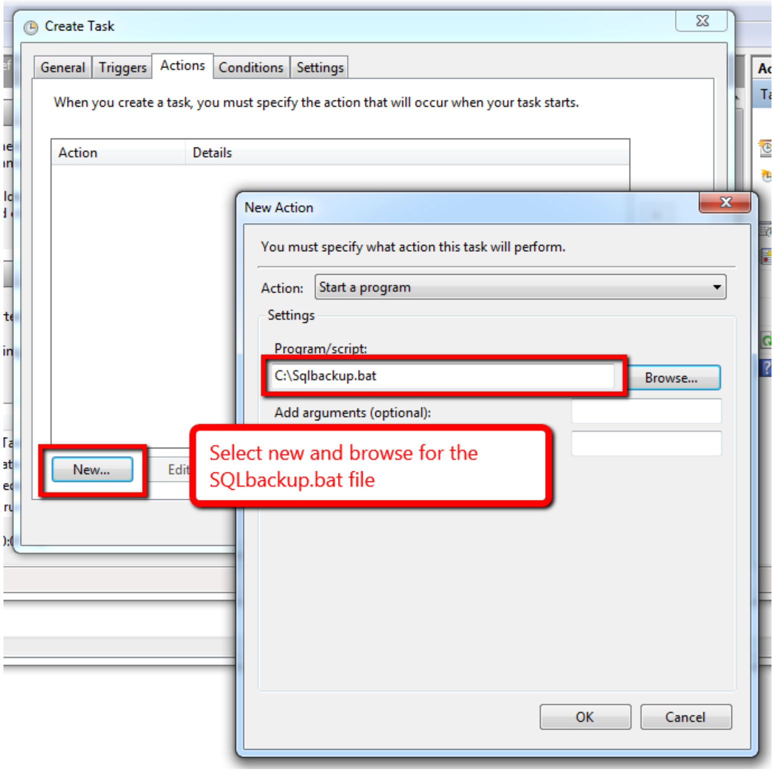
an: 3 days

3:40:28 PM □ Synchronize across time zones

☑ Enabled

Select 'New' and then choose the time and frequency that the task will run

7. Select the ‘Actions’ tab and ‘New’
8. In the popup, select ‘Start a program’ for Action: and use the browse button to locate the batch file that you created in Sections 2



When you create a task, you must specify the action that will occur when your task starts.

Action	Details

You must specify what action this task will perform.

Action: Start a program

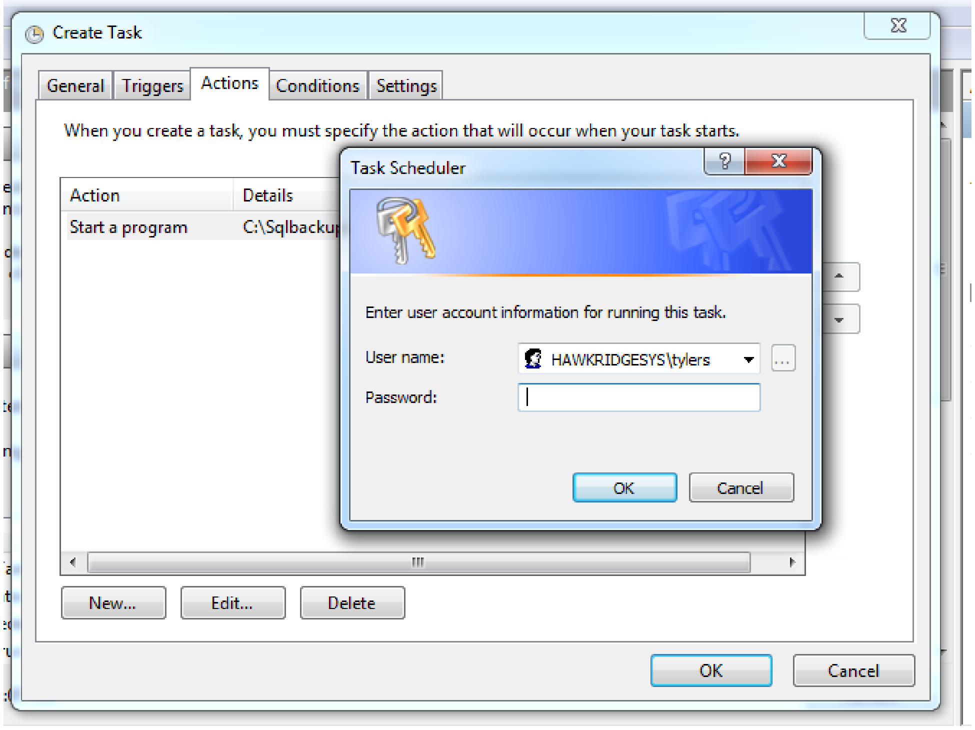
Settings

Program/script: C:\Sqlbackup.bat

Add arguments (optional):

Select new and browse for the SQLbackup.bat file

9. Select ‘OK’ at the bottom of the screen to complete the scheduled task creation
10. To save the new task that you just created, you will be prompted to input your user password



When you create a task, you must specify the action that will occur when your task starts.

Action	Details
Start a program	C:\Sqlbackup

Enter user account information for running this task.

User name: HAWKRIDGESYS\tylers

Password:

11. The final step is to ensure that the backups are being created and are useful. Periodically check that new backups are being taken by the Task Scheduler. Also, to ensure that the backups that

are being taken are of use and not corrupted, try restoring the database of a test server using the backup files that are being created automatically.

Reference Articles

Portions of this instruction document were taken from Microsoft Knowledge Base article 2019698.

1. https://support.microsoft.com/en-us/kb/2019698
2. http://www.makeuseof.com/tag/write-simple-batch-bat-file/
3. https://msdn.microsoft.com/en-us/library/ms345415.aspx
4. https://technet.microsoft.com/en-us/library/cc721931.aspx