SOLIDWORKS PDM – COMPLETE GUIDE TO PDM STANDARD BACKUPS

Contents

SOLIDWORKS PDM Standard is a great data management tool included with SOLIDWORKS Professional and SOLIDWORKS Premium. However, it is built on an SQL Express database, which unfortunately means there are no automated maintenance plans in SQL Studio Manager. This makes automating the SQL backup procedure more difficult, but, as we will explain in this document, it is possible.

First, we need to clarify some assumptions. The information below assumes you have SQL Express, SQL Studio Manager, and PDM Standard already installed on a server. If you need assistance installing the software, please feel free to contact our services team for information. If you recently completed a PDM Jumpstart with GoEngineer, all the necessary PDM components have been installed as a part of your services.

Even if you used GoEngineer's Jumpstart services for PDM Standard implementation, GoEngineer is not directly responsible for your backups and is not specifically recommending any of the methods below. It is your company's responsibility to look over the information and decide the best course of action for your team.

This document serves to explore three methods to automate the SQL backup process without installing any additional software. It is simply a method to accomplish backups without purchasing, downloading, or installing additional software on the server.

There are various utilities on the internet to accomplish backups, including a few paid and free utilities for backing up SQL Express.

Some options include:

- SQL Backup and FTP
- SQL Backup Master
- SQL Backup Free

Note: GoEngineer does not specifically recommend these products nor can guarantee effectiveness. These utilities are not GoEngineer or SOLIDWORKS products, and they are just for information purposes only.

Restore Components

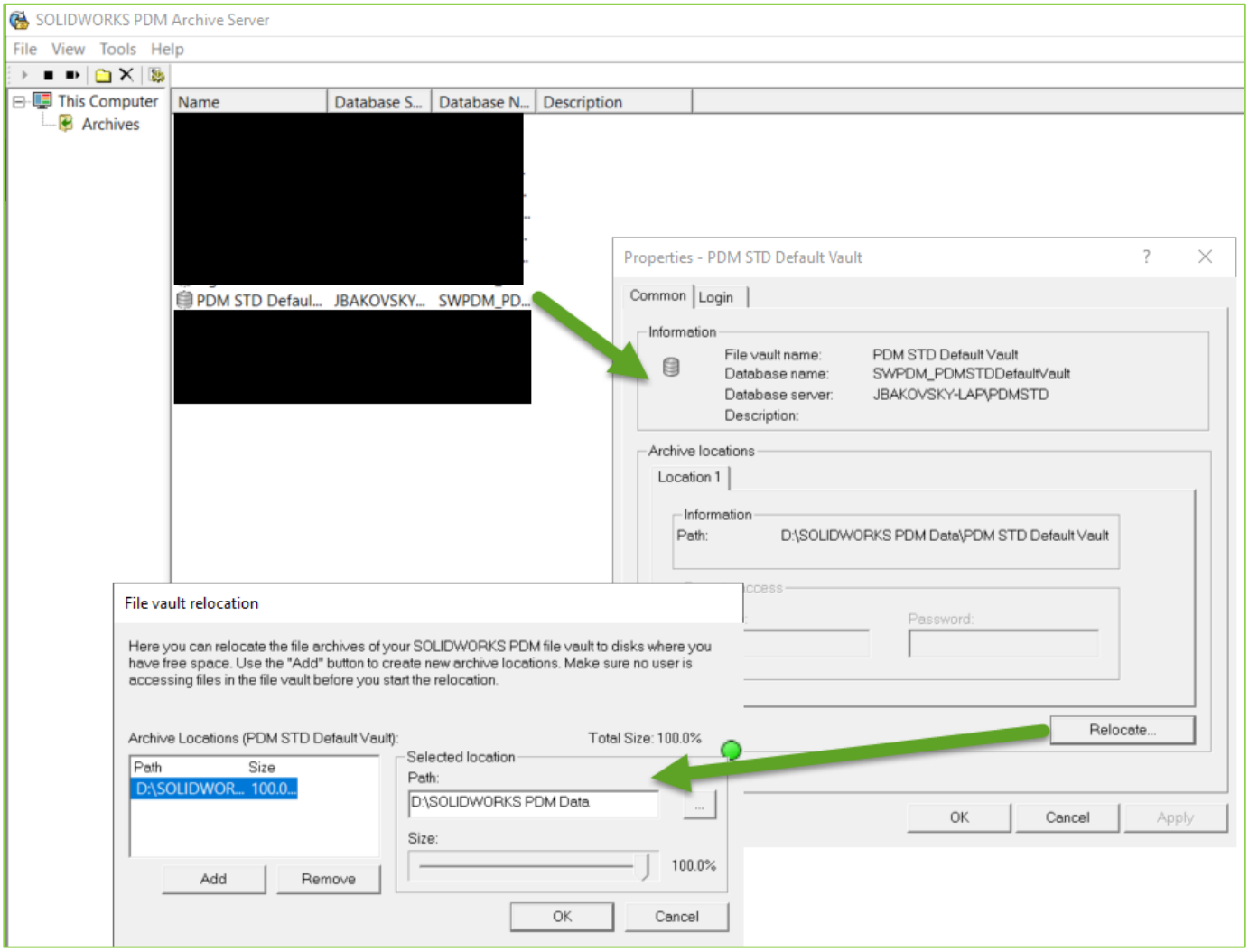
To better understand which PDM components we need to backup, we need to understand which PDM components are necessary for a restore in the event of a failure.

There are three components necessary for a full PDM restore:

1. Backup of Archive Server Folders
2. Backup of Archive Server Settings
3. Backup of SQL Databases

Backup the Archive Server Folders

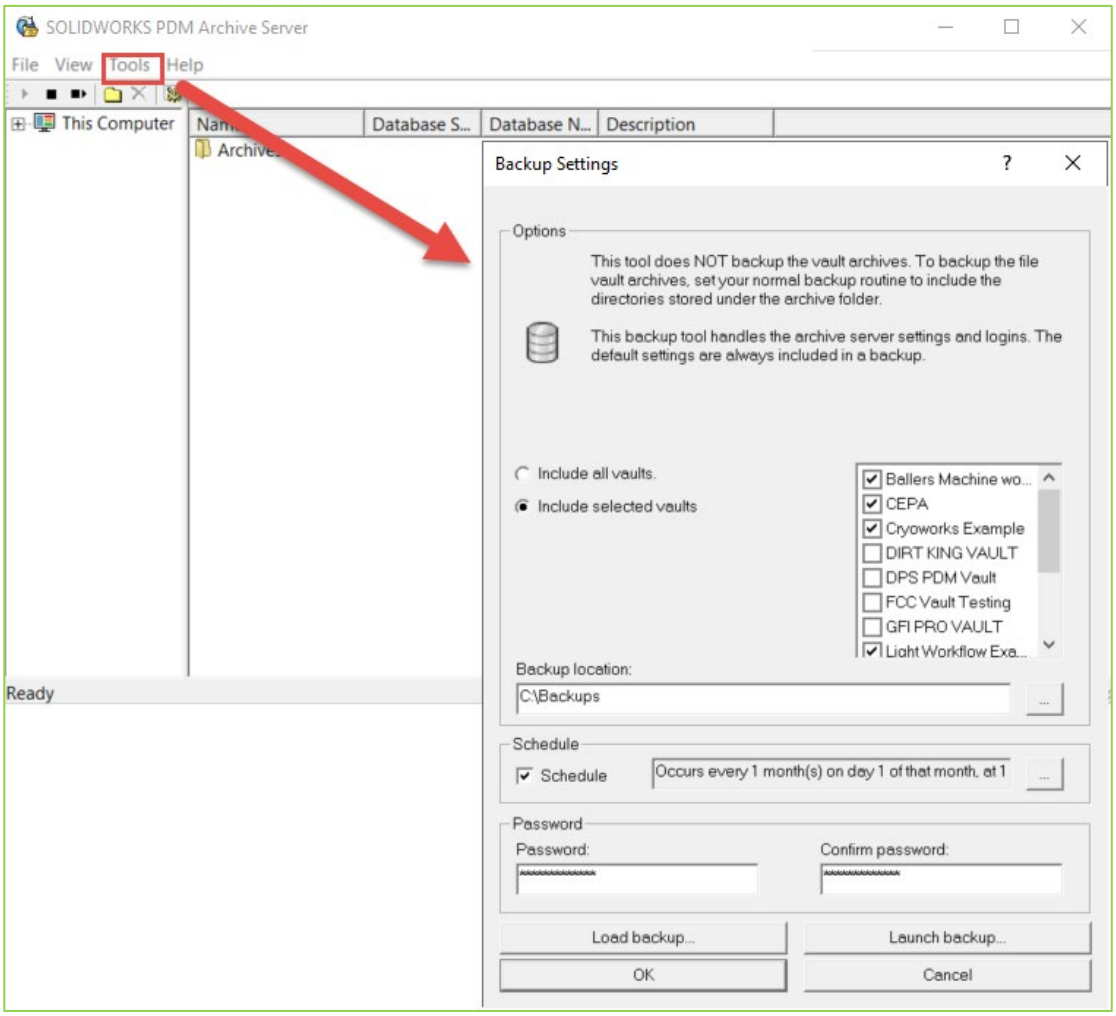
Please verify the location for the Archive Server Folders before attempting to back them up. It is possible to find the location of the Archive Servers by logging into the PDM Server. Press **Start** and search for the **SOLIDWORKS PDM Archive Server**. Expand Computers, Select **Archive**, then right mouse click on the vault you want to backup. Select **Properties** → **Relocate** and copy the Path shown below to see where your archive folders reside.



Contact your IT department to see what imaging software you will need to image and back up the folders in the specified Archive Server folder location.

Backup the Archive Server Settings

On the PDM Server, press **Start** and search for the **SOLIDWORKS PDM Archive Server**. Select **Tools → Backup Settings** as shown below. Set a location and set it on a routine schedule for backups. It is a good idea to backup these files in the same location as the SQL backups.



SQL Backups

Backup Method 1: Backup with multiple unique backup files. Backup up one database. It does not back up the ConioMasterDB by default.

Backup Method 2: Backup with a single backup for each database. No history. Most simple solution.

Backup Method 3: Backup with multiple unique backup files. More options are available for full, differential backups as well as options to specify additional databases to back up.
All backup methods require the SQL command utility to be installed. The sqlcmd utility lets you enter Transact-SQL statements, system procedures, and script files.

For **SQL Server 2014 and lower versions**, the utility is shipped as part of the product. No download or separate installation is required.

For **SQL Server 2016 and 2017**, the sqlcmd utility is offered as a separate download. For more information, review sqlcmd Utility.

Command Line Utilities 15 download: https://go.microsoft.com/fwlink/?linkid=2142258. For **SQL Server 2019**, the utility is shipped as part of the product. No download or separate installation is required.

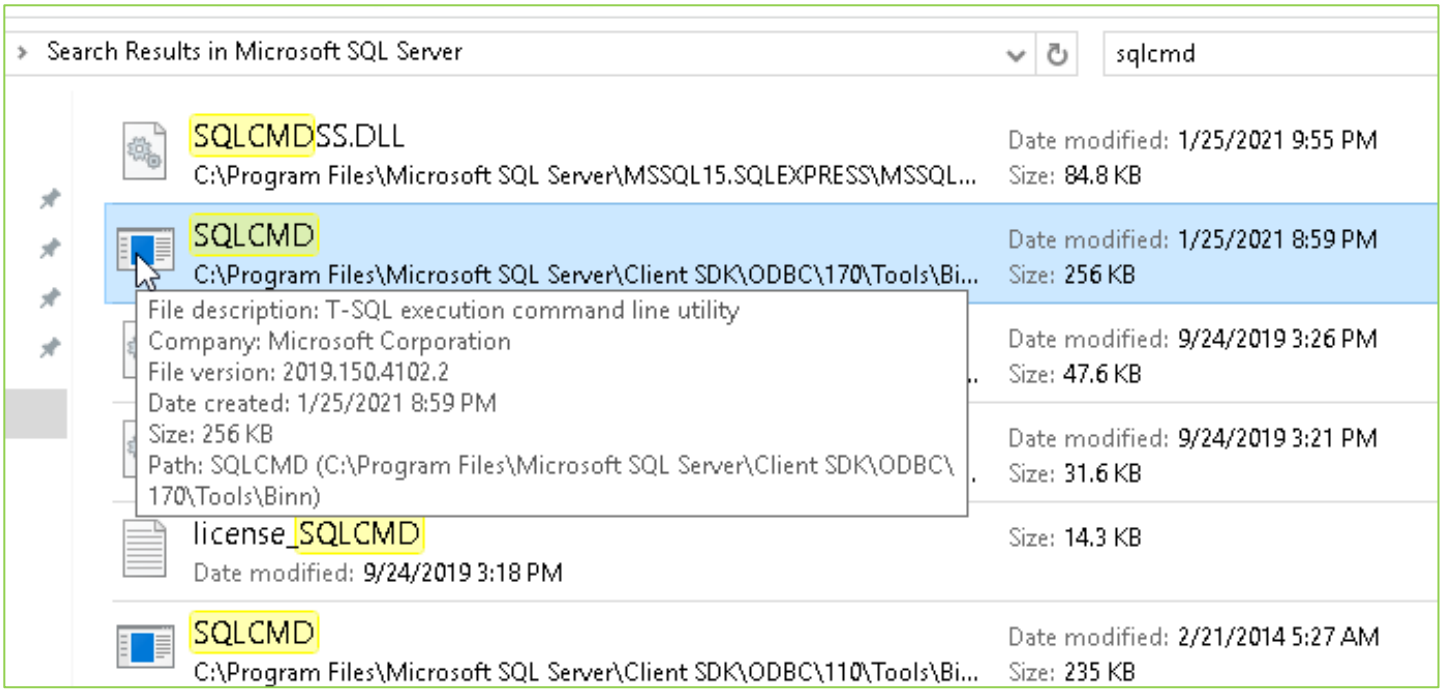


Figure 1: Installation location for SQL Server 2019

Backup Method 1

GoEngineer Article: SOLIDWORKS PDM Standard – Backup SQL

Summary:

Uses two scripts to produce a backup of a single database. One script contains SQL commands to create the database backup file. The second script is a Windows batch file that calls the SQL script and runs a command to remove older backup files. For backup history, a parameter in the Windows batch file controls the number of backups to keep.
If multiple vaults are present or if the ConisioMaster and vault databases are to be backed up, another set of SQL and Windows scripts must be customized for each additional database(s).

Variables are used to set Instance\database name, backup file location\name, script location, and the number of backups to keep.

The Windows batch file can be run manually or can be automated using Windows Task Scheduler. Information on how to automate using the Windows Task Scheduler can be found in the *How to set up the batch file to automate using the Windows Task Scheduler* section.

Using this method, you end up with:

A folder containing multiple unique backup files. The number depends on the “backups_to_keep” variable value.

We have pre-written the batch file and SQL file here. Simply download this file and change the required values to match your database(s) indicated in the original content.

Backup Method 2

GoEngineer Article: SOLIDWORKS PDM Standard Automated SQL Backup

Summary:

This method assumes you are looking to automate the SQL backup procedure for the PDM databases in SQL.

This method uses a single Windows batch file containing SQL commands to create the database backup file. Each line will back up a single database. Multiple lines can be added to back up multiple databases in a single script. No backup history is preserved as the backup files are overwritten each time the script is run. Instance\database name and backup file location\name are fixed parameters in each command line.

The Windows batch file can be run manually or can be automated using Windows Task Scheduler. Information on how to automate using the Windows Task Scheduler can be found in the *How to set up the batch file to automate using the Windows Task Scheduler* section.

Backup Method 3

Summary:

Backup Method 3 uses a single Windows batch file containing SQL commands to create the database backup file. A single command line can be used to back up all databases in the instance or multiple lines can be added to back up a single selected database per command. For backup history, a parameter controls the number of backups to keep.

Variables are used to set Instance\database name, backup file location\name, script location, and the number of backups to keep. Additional parameters can be set on the command line to pass to the stored procedure.

The Windows batch file can be run manually or can be automated using Windows Task Scheduler. Information on how to automate using the Windows Task Scheduler can be found in the *How to set up the batch file to automate using the Windows Task Scheduler* section.

Prerequisites:

1. This method requires the SQL command utility to be installed. This is included with the installation of SQL Express 2014 and 2019. It must be downloaded and installed separately for SQL 2016 and 2017.

2. A stored procedure must be added to the master database in the SQL instance.

Using this method, you end up with:

A folder containing multiple unique backup files. The number depends on the “backups_to_keep” variable value. There are two options for executing method 3, simply comment out the option you do not plan on using.

1. Option 1 will back up all databases in the instance including system databases.

2. Option 2 will back up only the ConisioMaster and Primary vault databases.

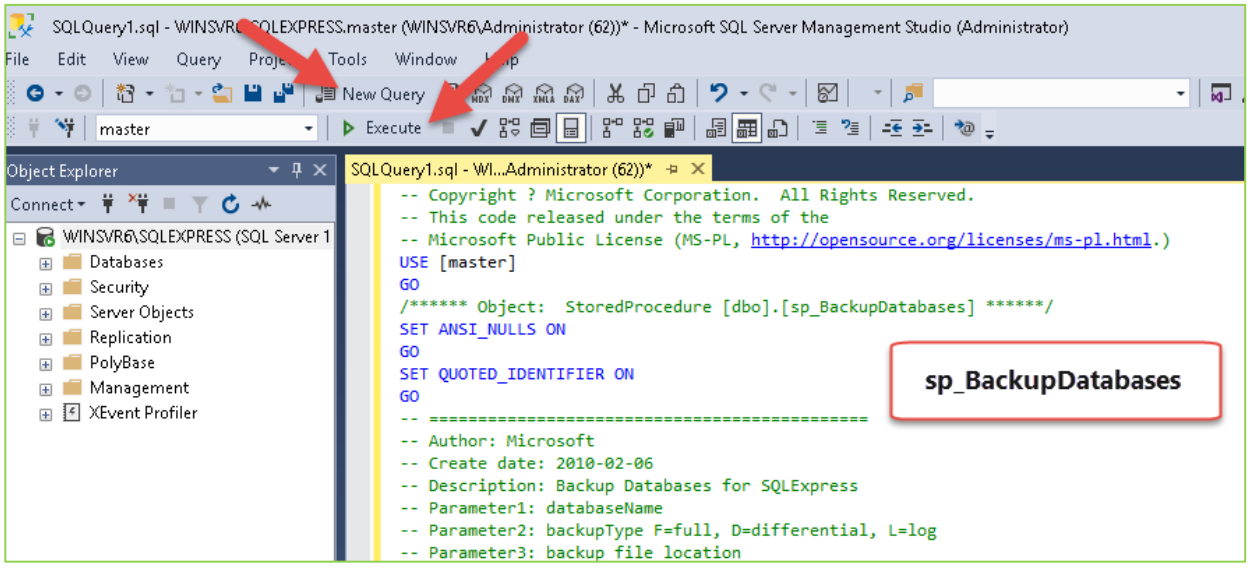
A prewritten download for both Method 3 Options 1 and 2 can be found here. You can edit either of these documents with your backup location and SQL instance. If you would like to set this up yourself, follow the below steps. Only one batch file is necessary to run this backup.

Step A: Create a stored procedure to back up your databases

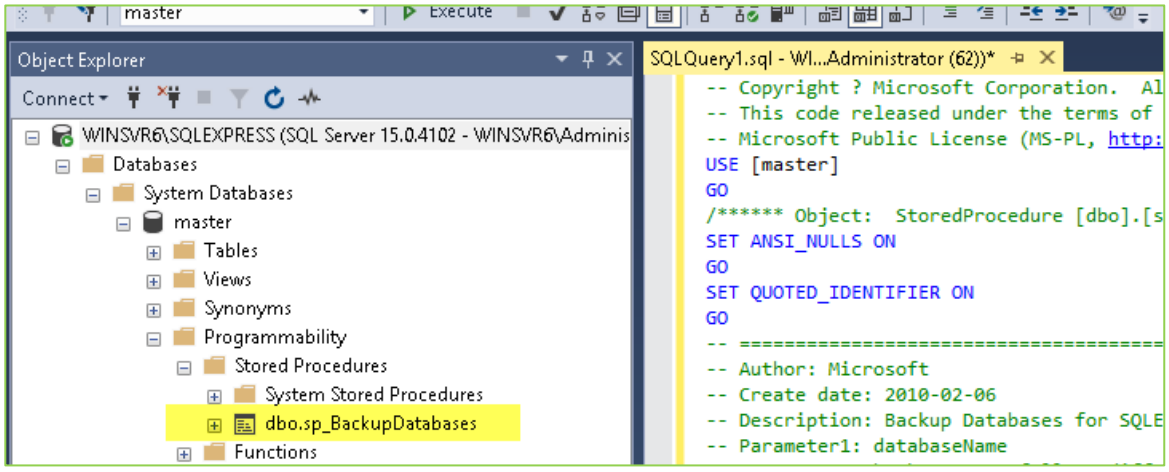
Connect to your SQL express instance and create sp_BackupDatabases stored procedure in your **master** database using the script at the following location: SQL_Express_Backups

1. Open the SQL Server Management Studio and connect to the SQL Express instance.

2. Start a new query and paste the contents “SQL_Express_Backups.sql” into the empty query.



3. Executing the query adds the stored procedure to the master database. Just check to see that it exists in this location below. You do not need to open the table, just verify that it exists.



Step C: Create batch file using text editor

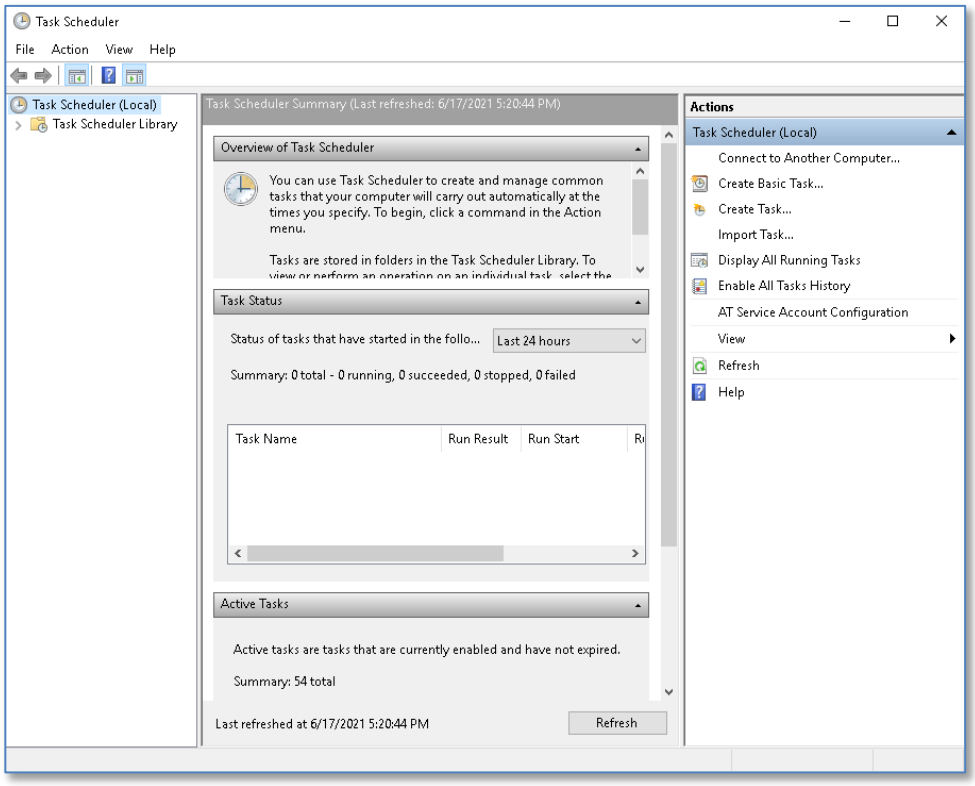
In a text editor, create a batch file that is named *PDM_Std_backup.bat*, and then copy the text from either of these files for option one and two in method 3 and save as a batch file (.bat).

This backup script is set up to use Windows authentication to connect to SQL. If you are using SQL authentication, ensure that access to the folder is restricted to authorized users as the passwords are stored in clear text.

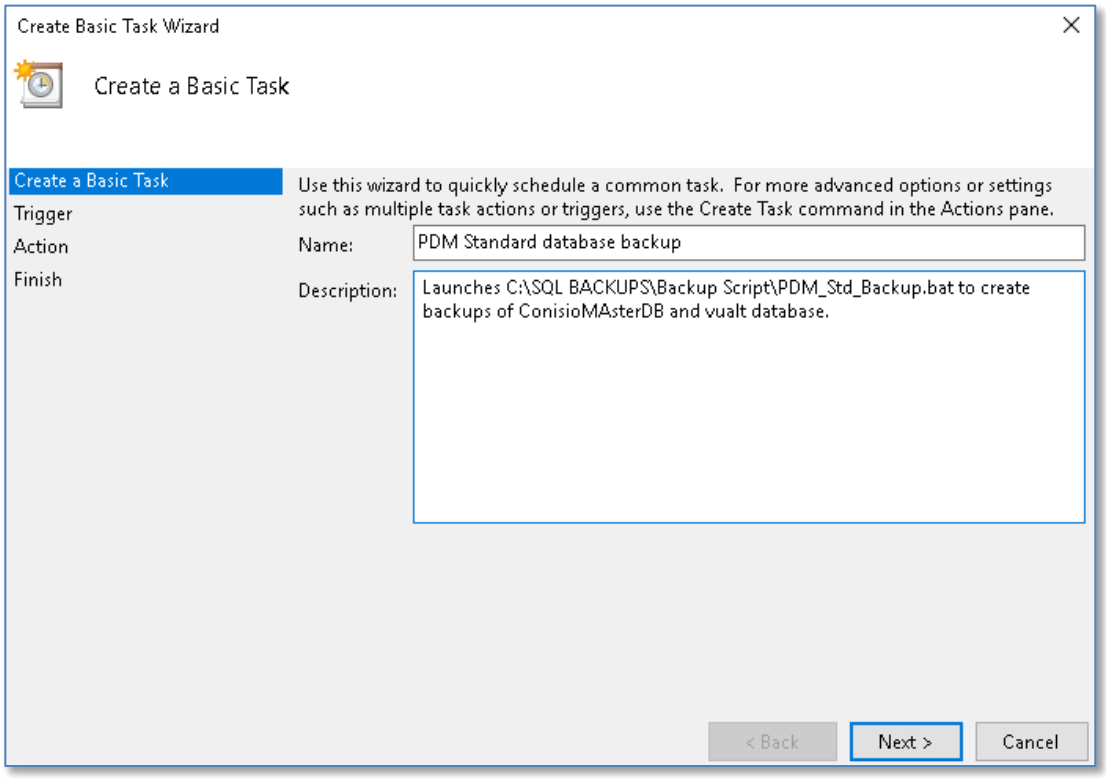
How to set up a batch file to automate using the Windows Task Scheduler

All the batch files created in the document can be automated using the Windows Task Scheduler. Follow these step-by-step instructions:

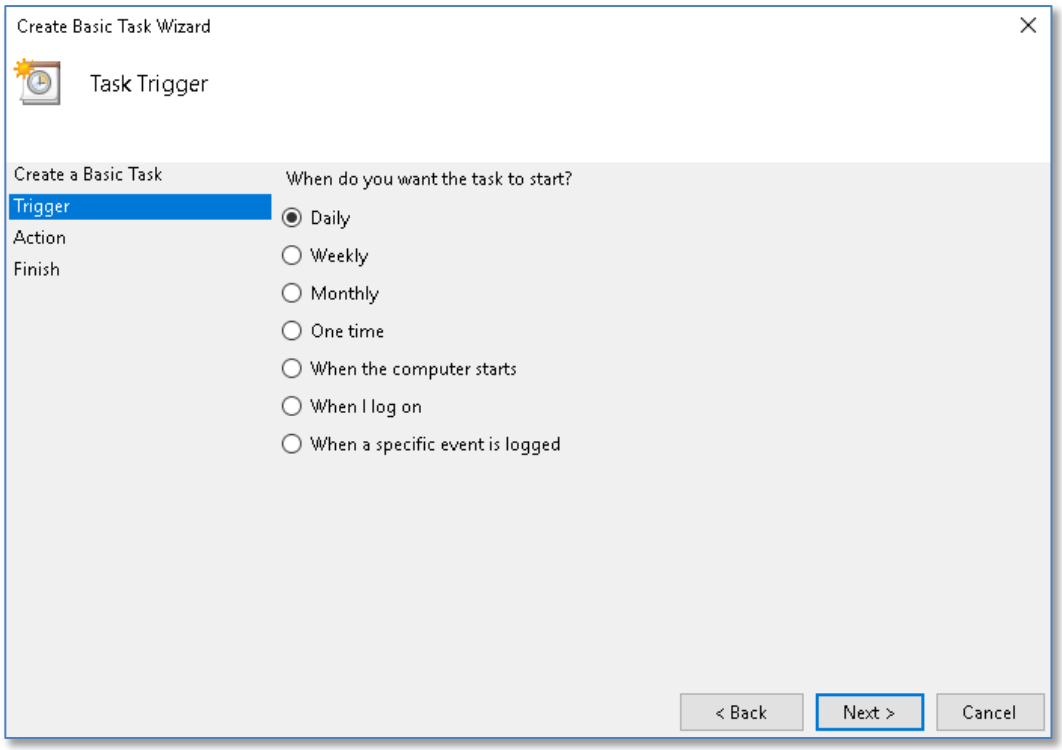
1. Start the Windows Task Scheduler and select **Create Basic Task…**



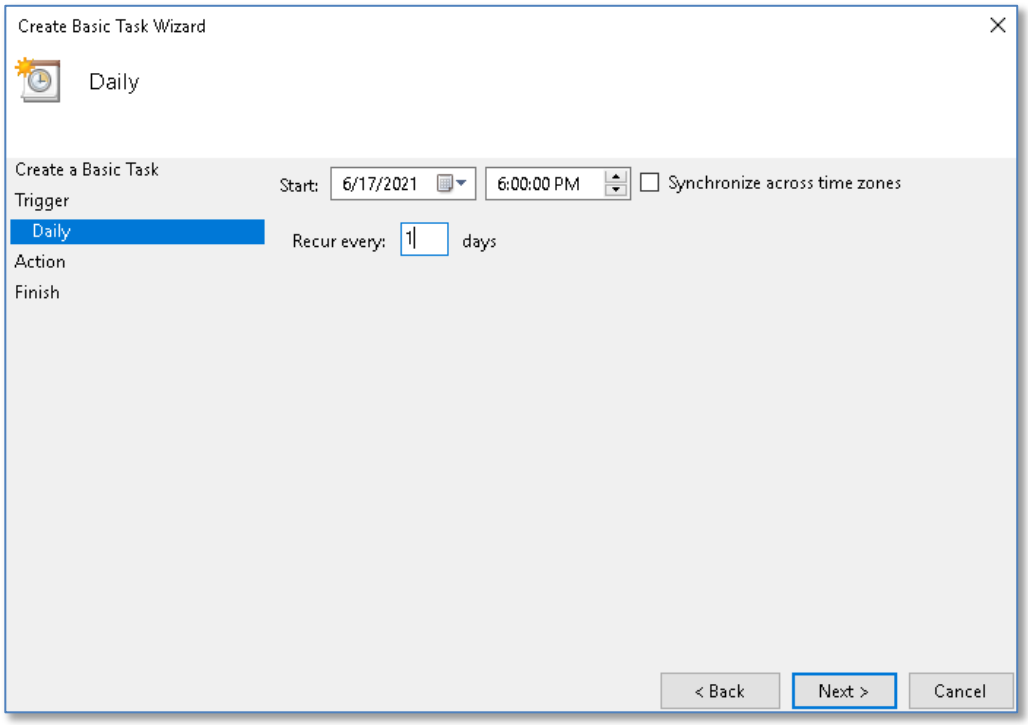
2. Name the task.



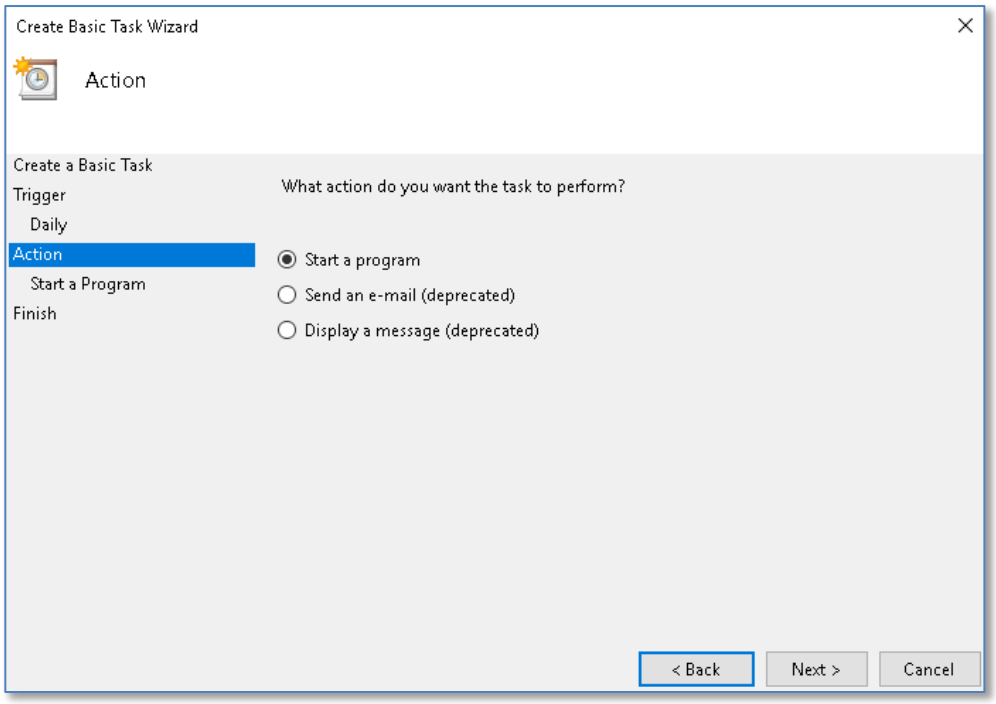
3. Set the trigger to run the task daily (or whatever interval you would like).



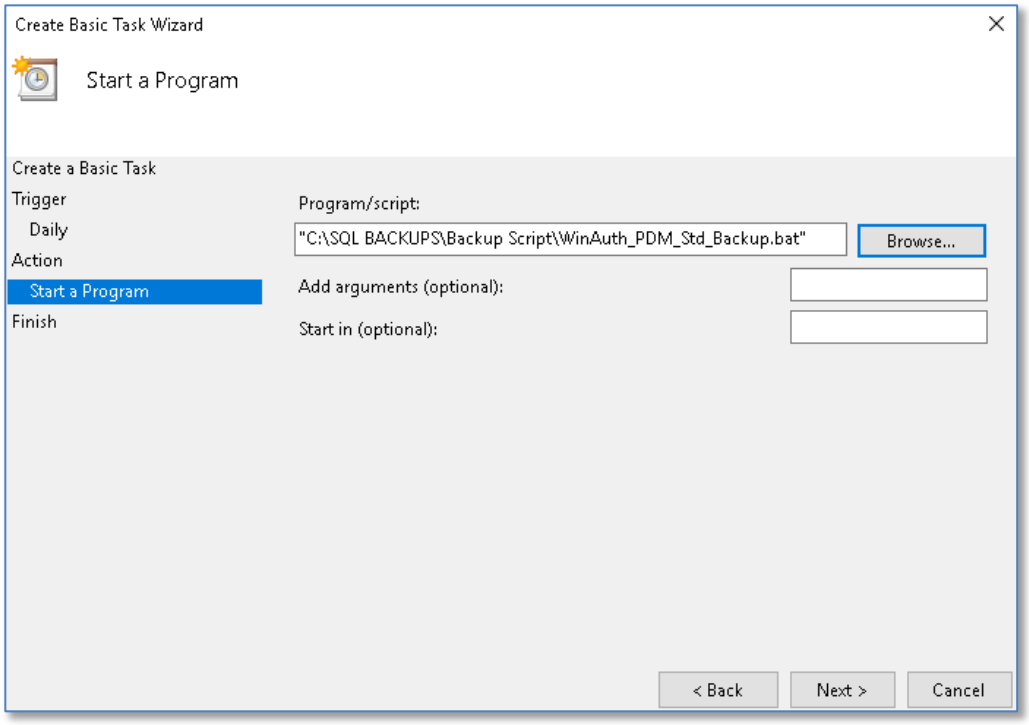
4. Set a time for the task to start.



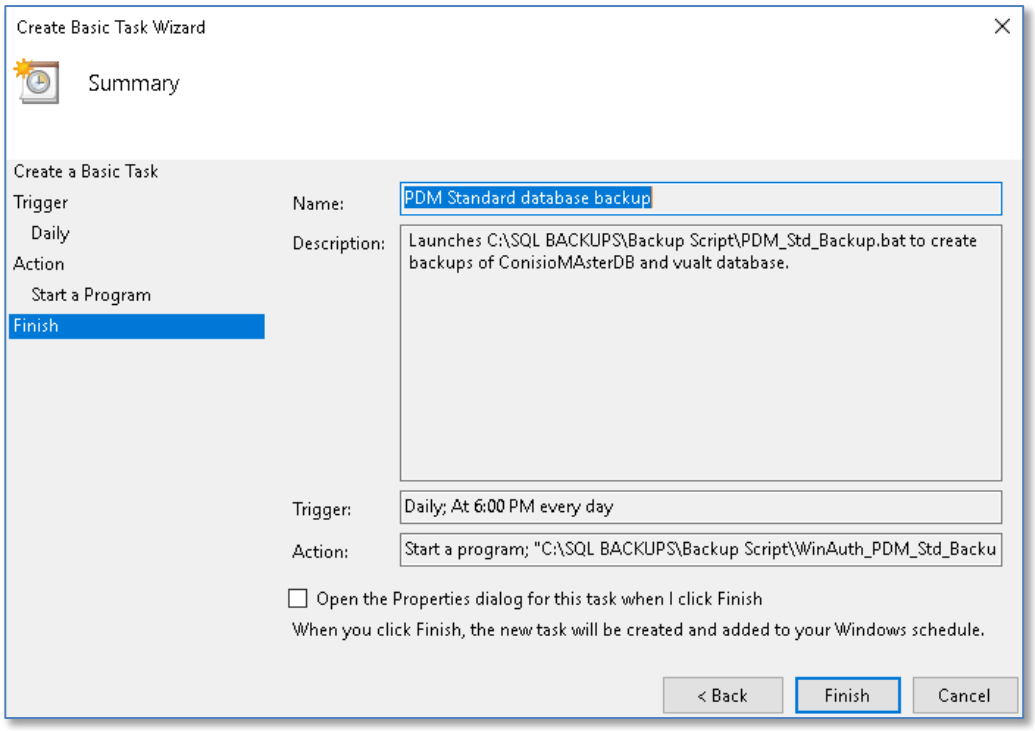
5. Set the type of action to perform to **Start a program**.



6. Browse to the backup script created earlier. This will be any of the batch (.bat) files that were created in any of the above methods.



7. Review the information in the summary. Select **Finish** to create the Windows schedule.



Appendix

References:
https://docs.microsoft.com/en-us/troubleshoot/sql/admin/schedule-automate-backup-database

Copy and pasted stored procedure from Microsoft below.

sp_BackupDatabases stored procedure

```
-- Copyright ? Microsoft Corporation.  All Rights Reserved.
-- This code released under the terms of the
-- Microsoft Public License (MS-PL, http://opensource.org/licenses/ms-pl.html.)
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= ISNULL(@databaseName ,name)
        ORDER BY Name
        -- Filter out databases which do not need to backed up
        IF @backupType='F'
            BEGIN
            DELETE @DBs where DBNAME IN ('tempdb','Northwind','pubs','AdventureWorks')
            END
        ELSE IF @backupType='D'
            BEGIN
            DELETE @DBs where DBNAME IN ('tempdb','Northwind','pubs','master','AdventureWorks')
            END
        ELSE IF @backupType='L'
            BEGIN
            DELETE @DBs where DBNAME IN ('tempdb','Northwind','pubs','master','AdventureWorks')
            END
        ELSE
            BEGIN
            RETURN
            END
        -- Declare variables
        DECLARE @BackupName nvarchar(100)
        DECLARE @BackupFile nvarchar(300)
        DECLARE @DBNAME nvarchar(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
```