BACKUP APP V7

MICROSOFT SQL SERVER BACKUP AND RESTORE GUIDE

Revision History

Date	Descriptions	Type of modification
29 July 2016	First Draft	New
22 Aug 2016	Modify Ch 1.5, Appendix B	Modify
30 Aug 2016	Add Ch 1.6, Recovery Model	New
3 Feb 2017	Added instructions and screen shots for Encryption key handling in Ch. 4.1	New
28 Feb 2017	Added Encryption Type option in Ch. 4.1 Creating Backup Set for Microsoft SQL Server	New
20 Mar 2017	Added Ch.1 Overview; Added Ch.5 Backup Mode; Revised Create Backup Backup Set section	New / Modification
7 Apr 2017	Added Backup Mode section; Revised Appendix B Truncating Transaction Log section, Added relevant information for backup of transaction log	New / Modification

Table of Contents

1 Overview

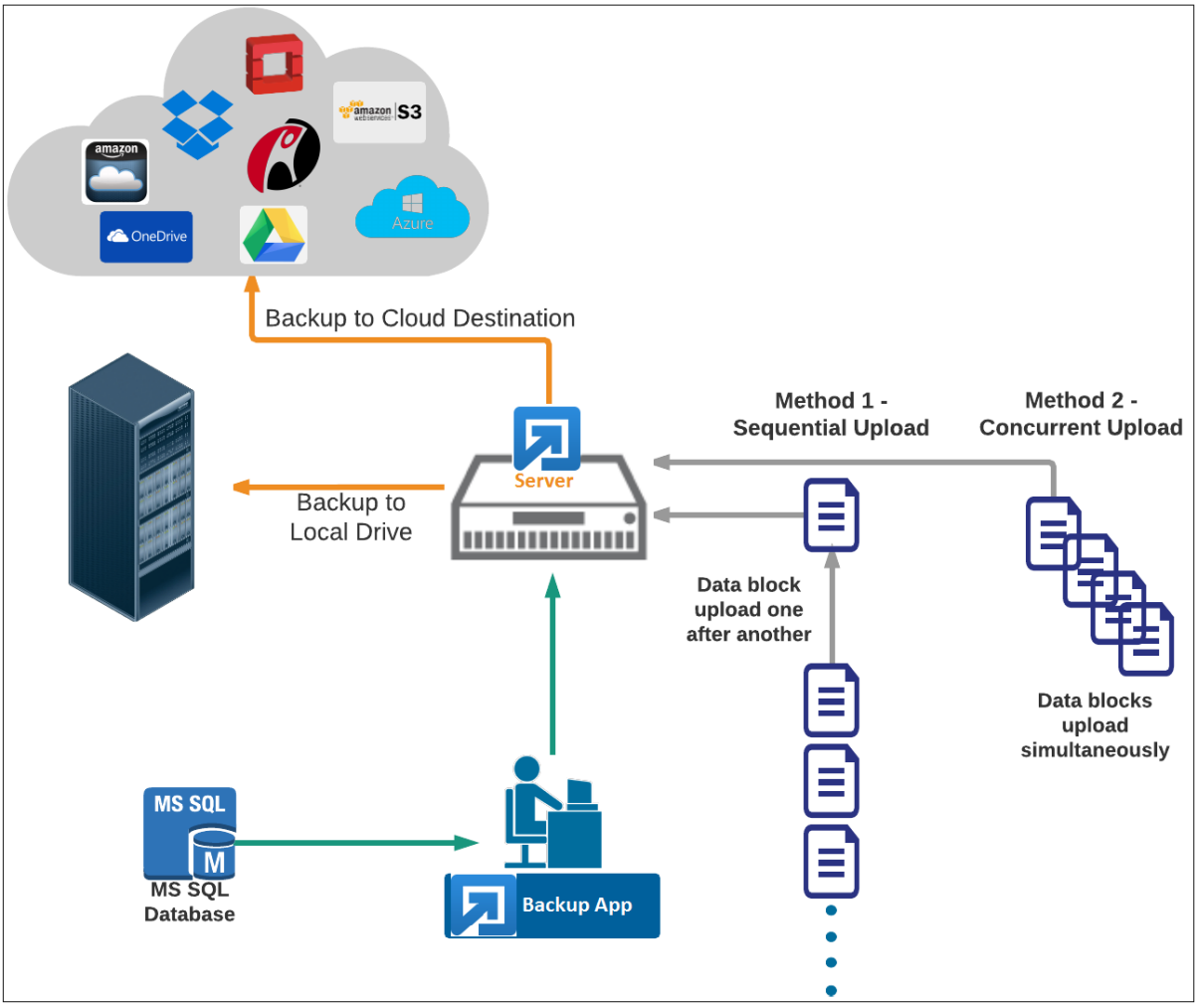
What is this software?

Backup App brings you specialized client backup software, namely Backup App, to provide a comprehensive backup solution for your MS SQL Server. The MS SQL Server module of Backup App provides you with a set of tools to protect your MS SQL Server, whether in VSS backup mode or ODBC backup mode.

System Architecture

Below is the system architecture diagram illustrating the major elements involved in the backup process among the MS SQL server, Backup App.

In this user guide, we will focus on the software installation, as well as the end-to-end backup and restore process using the Backup App as a client backup software.



2 Requirements

You are strongly recommended to configure or check all the requirements below before you proceed with the MS SQL server backup and restoration.

Hardware Requirement

- Dual Core architecture or above [1]
- Minimum: 2 GB
- Recommended: 4 GB or more
- Minimum: 500 MB
- TCP/IP
- Java 1.7u76 or above [3]

Software Requirement

- **Windows platforms**:

  Vista Home Basic / Home Premium / Business / Enterprise / Ultimate

  7 Home Basic / Home Premium / Professional / Enterprise / Ultimate

  8 Pro / Enterprise

  8.1 Pro / Enterprise

  10 Pro / Enterprise

  Server 2008 Standard / Enterprise / Datacenter

  Server 2008 R2 Standard / Enterprise / Datacenter

  Server 2012 Standard / Essentials / Datacenter

  Server 2012 R2 Standard / Essentials / Datacenter

  Server 2016 Standard / Premium

  Small Business Server 2008 Standard / Essentials / Datacenter

  Small Business Server 2011 Standard / Essentials / Datacenter

- **Linux platforms**:

  CentOS 6

  CentOS 7

  Red Hat Enterprise Linux 6

  Red Hat Enterprise Linux 7

- **Unix platforms**:

  FreeBSD 9.0 / 9.1 / 9.2 / 10.0 [9]

  FreeBSD 10.1

  Solaris 10 x64

  Solaris 11 Express x64

  Solaris 11 x64

- **Mac OS X platforms**:

  Mac OS X 10.7.3 or above [10]

  OS X 10.8

  OS X 10.9

  OS X 10.10

  OS X 10.11

  macOS 10.12

Backup App Installation

Make sure the latest version of Backup App has been installed on the MS SQL server.

Backup App Add-On Module Configuration

Make sure the add-on module Microsoft SQL Server has been enabled in your Backup App user account. Contact your backup service provider for more details.

User Account Privileges

The operating system account that performs the backup and restoration must have the administrator permission (i.e. sufficient permission to access both SQL server & VSS). For the temporary folder, where the database and transaction log files will be spooled to, is located on a network drive, make sure the login account has sufficient permission to access the network resources.

Refer to the URLs below for more details.
https://support.microsoft.com/en-us/kb/2926557
https://technet.microsoft.com/en-us/library/cc966520.aspx

SQL Server VSS Writer (For VSS Backup Mode Only)

Ensure that the **SqlServerWriter** is installed and running on the SQL server, and the writer state is **Stable**. This can be verified by running the "**vssadmin list writers"** command in the Windows Command Prompt.

If you do not find the SqlServerWriter in the result, make sure the SQL Server VSS Writer has been started by following the instructions in Windows Services section below.

Example:

```
C:\Users\Administrator>vssadmin list writers
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line
tool
(C) Copyright 2001-2005 Microsoft Corp.

Writer name: 'Task Scheduler Writer'
   Writer Id: {d61d61c8-d73a-4eee-8cdd-f6f9786b7124}
   Writer Instance Id: {1bddd48e-5052-49db-9b07-b96f96727e6b}
   State: [1] Stable
   Last error: No error

Writer name: 'VSS Metadata Store Writer'
   Writer Id: {75dfb225-e2e4-4d39-9ac9-ffaff65ddf06}
   Writer Instance Id: {088e7a7d-09a8-4cc6-a609-ad90e75ddc93}
   State: [1] Stable
   Last error: No error

Writer name: 'Performance Counters Writer'
   Writer Id: {0bada1de-01a9-4625-8278-69e735f39dd2}
   Writer Instance Id: {f0086dda-9efc-47c5-8eb6-a944c3d09381}
   State: [1] Stable
   Last error: No error

Writer name: 'SqlServerWriter'
   Writer Id: {a65faa63-5ea8-4ebc-9dbd-a0c4db26912a}
   Writer Instance Id: {3de4f842-4d57-4198-9949-3b3f8c2629dc}
   State: [1] Stable
   Last error: No error

Writer name: 'System Writer'
   Writer Id: {e8132975-6f93-4464-a53e-1050253ae220}
   Writer Instance Id: {32d2fccc-624f-4baa-beb3-17b27fcae9ee}
   State: [1] Stable
   Last error: No error

Writer name: 'ASR Writer'
   Writer Id: {be000cbe-11fe-4426-9c58-531aa6355fc4}
   Writer Instance Id: {e8580fb0-b51f-40ab-91bf-4eff5107c4d1}
   State: [1] Stable
   Last error: No error

Writer name: 'WMI Writer'
   Writer Id: {a6ad56c2-b509-4e6c-bb19-49d8f43532f0}
   Writer Instance Id: {de1b6322-1d96-4f85-adbf-05cb517322ea}
   State: [1] Stable
   Last error: No error

Writer name: 'BITS Writer'
   Writer Id: {4969d978-be47-48b0-b100-f328f07ac1e0}
   Writer Instance Id: {a623b49f-a3d4-42d2-af9a-4e924fb31262}
   State: [1] Stable
   Last error: No error

Writer name: 'Registry Writer'
   Writer Id: {afbab4a2-367d-4d15-a586-71dbb18f8485}
```

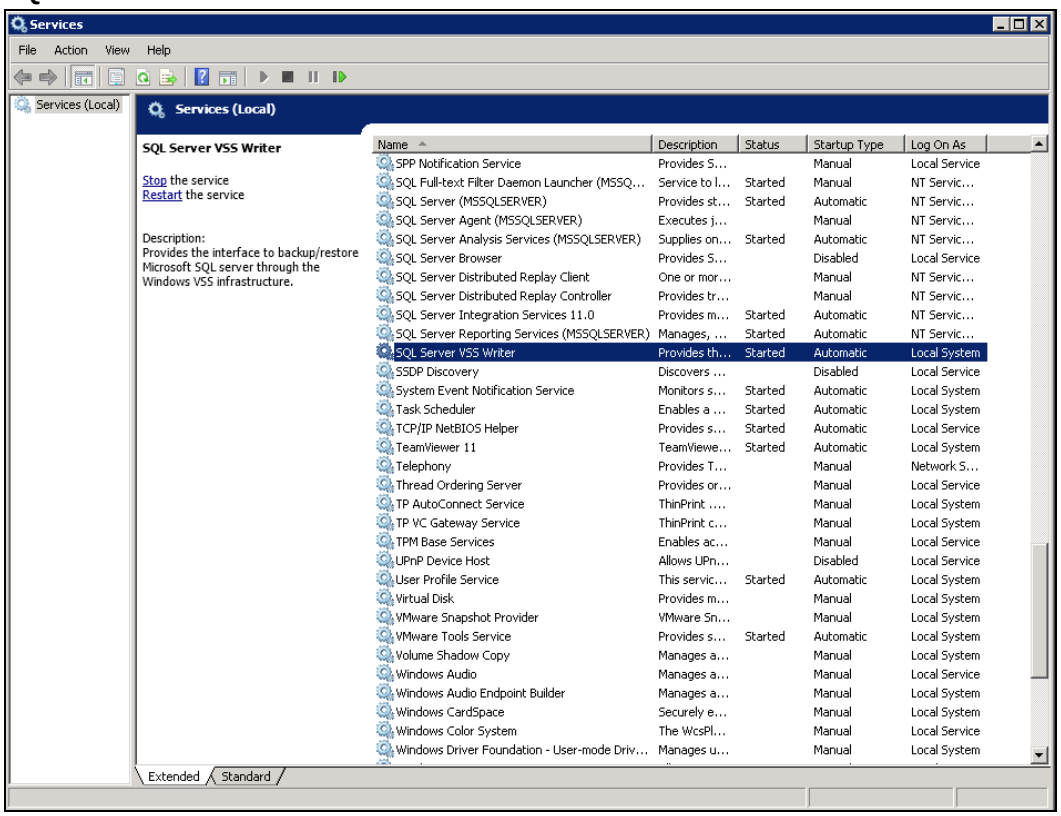
```
   Writer Instance Id: {cc6b42f1-ebd0-429f-b3d3-e860905d40d3}
   State: [1] Stable
   Last error: No error

Writer name: 'Shadow Copy Optimization Writer'
   Writer Id: {4dc3bdd4-ab48-4d07-adb0-3bee2926fd7f}
   Writer Instance Id: {957ff981-d54f-4a1f-8798-bd9bd76396bd}
   State: [1] Stable
   Last error: No error

Writer name: 'COM+ REGDB Writer'
   Writer Id: {542da469-d3e1-473c-9f4f-7847f01fc64f}
   Writer Instance Id: {801fea63-6bfc-406d-9a40-4ad5af484773}
   State: [1] Stable
   Last error: No error
```

Windows Services

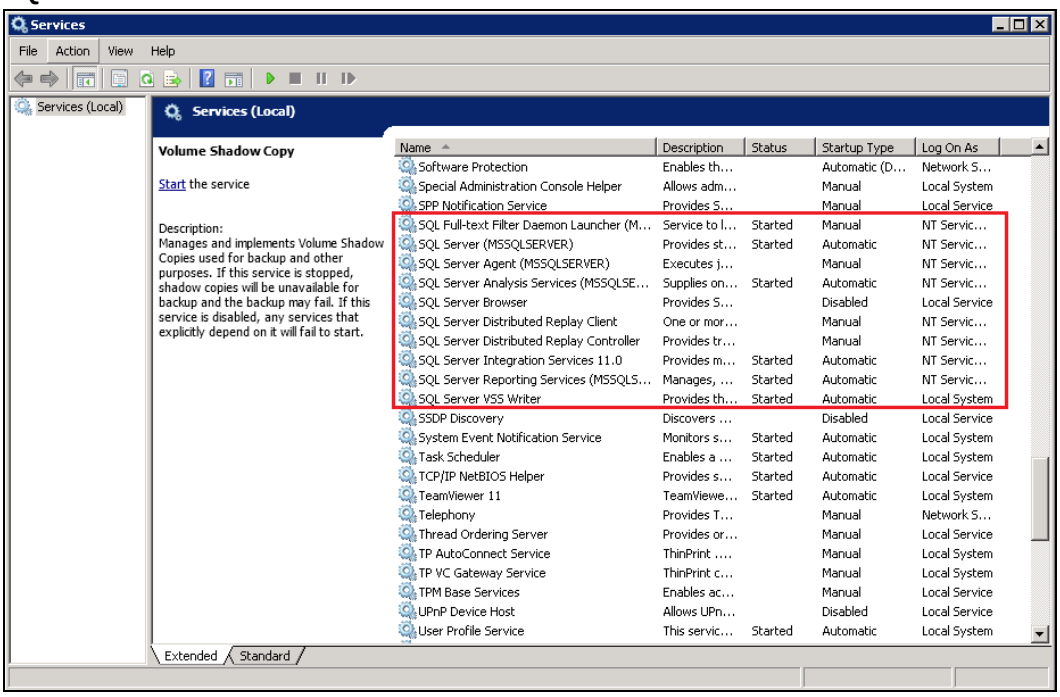
Ensure that the following services have been enabled in the Windows Services menu.

Launch **Services** in Windows by clicking **Start** then typing “Services” in the search box. All MS SQL server related services should be started by default, in case if it is not, turn it on by right clicking the item then selecting **Start**.

- **SQL Server VSS Writer**

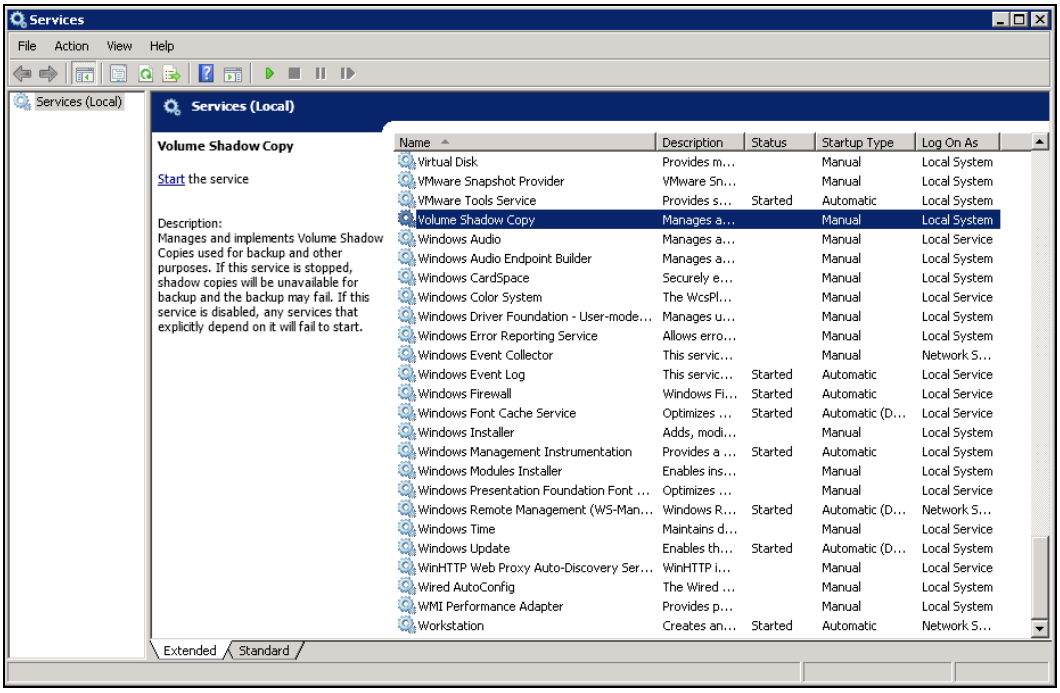


- **SQL Server Services**



- **Volume Shadow Copy**

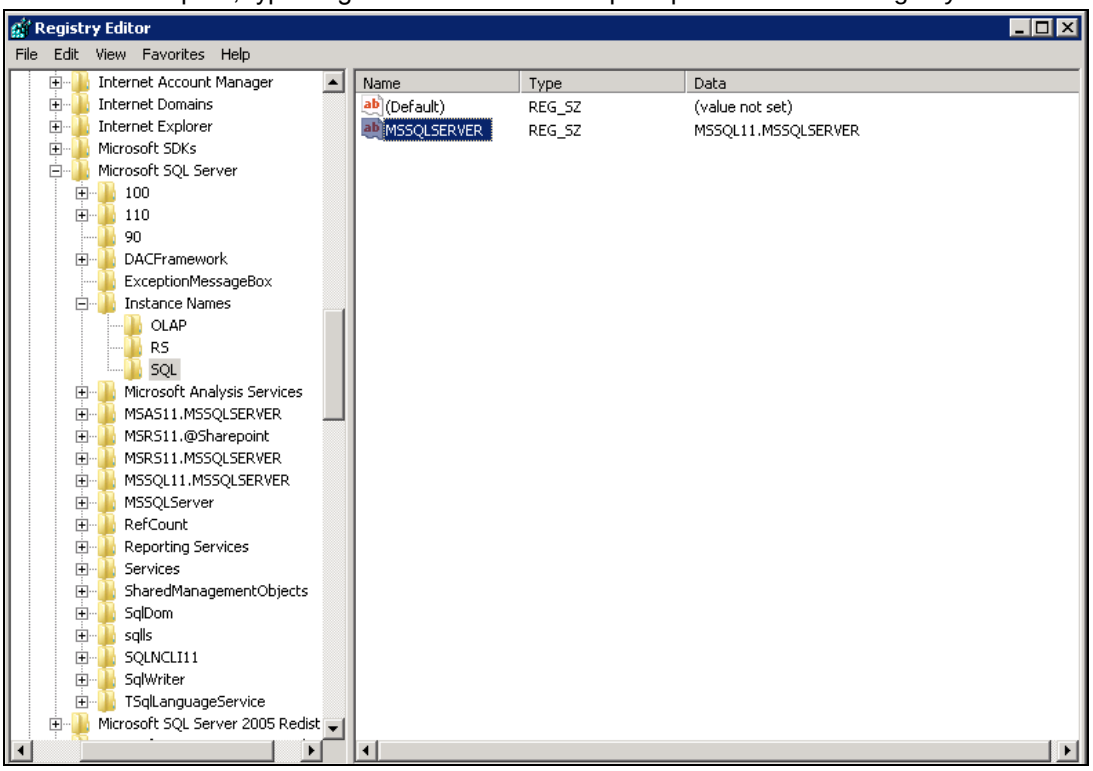
Make sure the Volume Shadow Copy service is not disabled.



MS SQL Server Registry

Make sure the MS SQL entry is present in the registry key **"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names\SQL"**

To access this path, type “regedit” in the command prompt to launch the Registry Editor.



Note: Pay extra attention when you are checking configuration in Registry Editor. Any unauthorized changes could cause interruption to the Windows operation.

Maximum Worker Thread

For SQL instance with large number of database (more than 500 databases), consider to increase the “Maximum Worker Thread” setting. Refer to the article below for further details.
https://msdn.microsoft.com/en-us/library/ms190219.aspx

- **MS SQL Recovery Model**

For transaction log backups the Recovery Model must be set to Full.

Refer to the URLs below for more details.
https://msdn.microsoft.com/en-us/library/ms189272.aspx
https://msdn.microsoft.com/en-us/library/ms189272(v=sql.105).aspx

3 Best Practice and Recommendation

Considerations for Backing up and Restoring of System Databases

Refer to the following tables for considerations for backup and restoration of system databases.

For backup of system databases

SQL server maintains a set of system level database which are essential for the operation of the server instance.

Several of the system databases must be backed up after every significant update, they include:

- master
- model
- msdb

For SQL database with replication enabled, make sure the “distribution” database is backed up as well.

This table summarizes all of the system databases.

System	Description	Backup	Suggestion
master	The database that records all of the system level information of a SQL server system.	Yes	To back up any database, the instance of SQL server must be running. Startup of an instance of SQL server requires that the master database is accessible and at least party usable. Back up the master database as often as necessary to protect the data sufficiently for your business needs. Microsoft recommends a regular backup schedule, which you can supplement with manual backup after any substantial update.
model	The template for all databases that are created on the instance of SQL server.	Yes	Backup the model database only when necessary, for example, after customizing its database options. Microsoft recommends that you create only full database backups of model, as required. Because model is small and rarely changes, backing up the log is unnecessary.
msdb	The msdb database is used by SQL Server Agent for scheduling alerts and jobs, and for recording operators. It also contains history	Yes	Back up msdb whenever it is updated.

	tables (e.g. backup / restore history table).		
tempdb	A workspace for holding temporary or intermediate result sets. This database is recreated every time an instance of SQL server is started.	No	The tempdb system database cannot be backed up.
distribution	The distribution database exists only if the server is configured as a replication distributor. It stores metadata and history data for all types of replication, and transactions for transactional replication.	Yes	Replicated databases and their associated system databases should be backed up regularly.

For restoring of system databases

System database	Restoration suggestion
master	To restore any database, the instance of SQL server must be running. Startup of an instance of SQL server requires that the master database is accessible and at least party usable. Restore or rebuild the master database completely if master becomes unusable.
model	Restore the model database if: ➢ The master database has been rebuilt. ➢ The model database has been damaged, for example due to media failure. ➢ The model database has been modified, in this case, it is necessary to restore model from a backup when you rebuild master, because the Rebuild Master utility deletes and recreates model.
msdb	Restore the msdb database if the master database has been rebuilt.
distribution	For restore strategies of distribution database, please refer to the following online document from Microsoft for more details: http://msdn.microsoft.com/enus/library/ms152560.aspx

4 Limitation

SQL Server Version for Backup & Restore

- **Automated Restore Option**
  If you have chosen the automated restoration to the Original SQL server or Alternate SQL server of your selection, the restoration can only be done in a SQL server version that is the same as the one used for performing the backup.

- **Manual Raw-file Restore Option**
  If you have chosen to restore the raw file, the raw database file(s) can be manually restored to the same or newer SQL server version that you used to perform the backup.

Local Drive Backup Source Only (VSS Mode Backup Only)

For VSS mode backup, it only allows backup of database in the local drive but not network drive.

Restoration to Other SQL Server

i. If you would like to restore database to an alternate SQL server, you can only choose to restore one database to restore at a time.

ii. If you would like to restore database to an alternate SQL server, make sure you choose to restore raw file by enabling the checkbox **Restore raw file**.

File System for Database Snapshot

You cannot create database snapshots on FAT32 file system or RAW partitions. The sparse files used by database snapshots are provided by the NTFS file system.

Standalone Environment Only

Backup App does not support backup of MS SQL server in cluster environment, only standalone environment is allowed.

5 Backup Mode

Starting from Backup App v7.11.0.0, you can choose from one of the two backup modes when creating a backup set for MS SQL server. The information below provides you with more details on each backup mode.

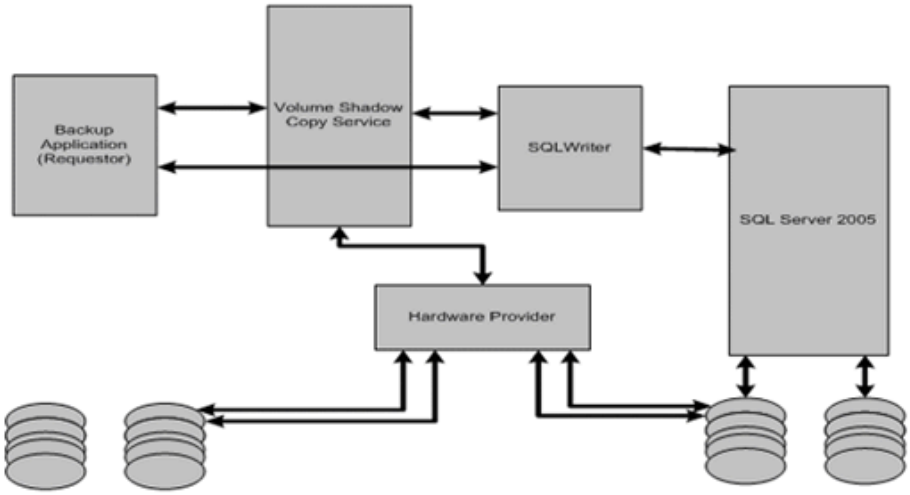
Note

For MS SQL server backup sets which are upgraded from v6, the default backup mode will be ODBC.

VSS Mode

Introduction

VSS-based backup utilizing the Microsoft SQL Server VSS Writer to obtain a consistent snapshot of the MS SQL databases, no spooling / staging of database file(s) is required during the backup process.



(Diagram from Microsoft)

Temporary Folder Requirement

- **Location for temporary folder**

  The temporary directory folder is used by Backup App for storing backup set index files and incremental/differential delta files. To ensure optimal backup/restoration performance, it is recommended that the temporary directory folder is set to a local drive. The temporary folder should not be located on Windows system partition or the database partition to minimize any potential performance impact on Windows and or database.

- **Temporary folder capacity**

  With VSS-based backup, the disk space of the temporary folder required for storing the VSS image is significantly smaller than using the ODBC spooling backup method. As the extra space is not require to hold the full database.

  It is recommended that the temporary directory should have at least free disk space of 50% of the total database size. The rationale behind this recommended free disk space is the default in-file delta ratio settings is 50%, therefore Backup App could generate incremental

or differential delta file(s) of up to 50% of the total database size. The actual free disk space required depends on various factors including the size of the database, number of backup destinations, backup frequency, in-file delta settings etc.

Pros

- **Fast and minimal interruption**
  The database snapshot capture process is fast and can be taken place on a running server, as you may continue to work when the snapshot capturing is taking place, there may be another process that holds your input in some memory section until the snapshot capture is completed. That said, the whole snapshot capture is fast, so there is no need for you to stop working and it causes minimal interruption to your business operation.
- **Significantly lesser disk burden**
  VSS Snapshot typically requires much less additional disk space than clones which is the traditional backup method by spooling database into the temporary folder. Often times, the capacity of the database to back up is huge and therefore the temporary folder would overload with the equal or even larger disk space if traditional backup method is used. By utilizing the VSS technology, it helps your system greatly reduce disk capacity burden and promote optimized performance.

Cons

- **No Transaction Log Backup**
  MS SQL does not support transaction log backup when VSS is used, therefore, transaction log backup will have to be done manually.
- **Workaround is time consuming**
  In order to truncate the transaction logs, you have to either change the Recovery model to Simple or perform a manual log truncation, which could be time consuming.

Transaction Log Handling

VSS based backup no longer requires backup of the backup of transaction log files, however for databases configured in either full or bulk-logging recovery model, this may eventually result in transaction logs filling up the available disk space on the volume of the MS SQL Server. https://technet.microsoft.com/en-us/library/cc966520.aspx.

To prevent this from occurring, it is recommended to change the recovery model of database selected for backup to simple recovery model.

Refer to the following steps for details:

1. In SQL Server Management Studio, expand **Databases**, select a user database, or expand **System Databases** and select a system database.
2. Right-click the corresponding database, then click **Properties** to open the **Database Properties** dialog box.
3. In the **Select a page** pane, click **Options**.
4. The current recovery model is displayed in the **Recovery model** list box. Modify the recovery model by selecting **Simple** from the model list.

Important: Only modify the recovery model of a live database during low activities hour. It is also recommended to perform a full backup before changing the recovery model.

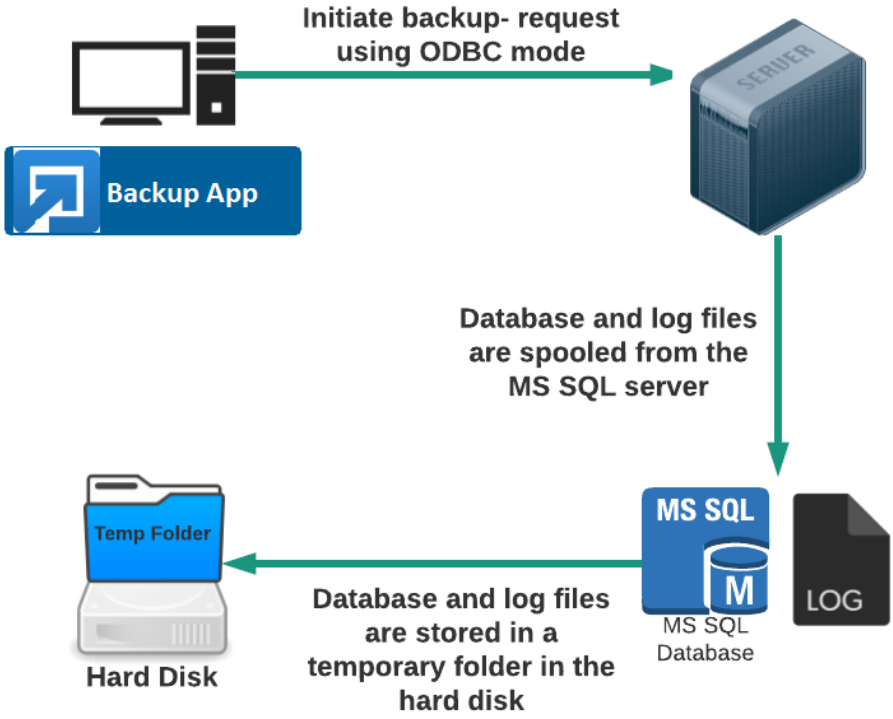
For MS SQL Server setups where you cannot modify the recovery model of the database, please refer to Appendix B for details on how to truncate transaction log (e.g. perform a

transaction log backup manually.

ODBC Mode

Introduction

By using the ODBC mode for MS SQL backup, databases files are spooled to a temporary directory before being uploaded to the backup destination.



Temporary Folder Requirement

- **Location for temporary folder**

  The temporary directory folder is used by Backup App for storing; the database files, incremental/differential delta files, and backup set index files. To ensure optimal backup/restoration performance, it is recommended that the temporary directory folder is set to a local drive. The temporary folder should not be located on Windows system partition or the database partition to minimize any potential performance impact on Windows and or database.

- **Temporary folder capacity**

  ODBC backup requires a significantly larger disk space of temporary folder as it need to store the database files spooled during the backup process.

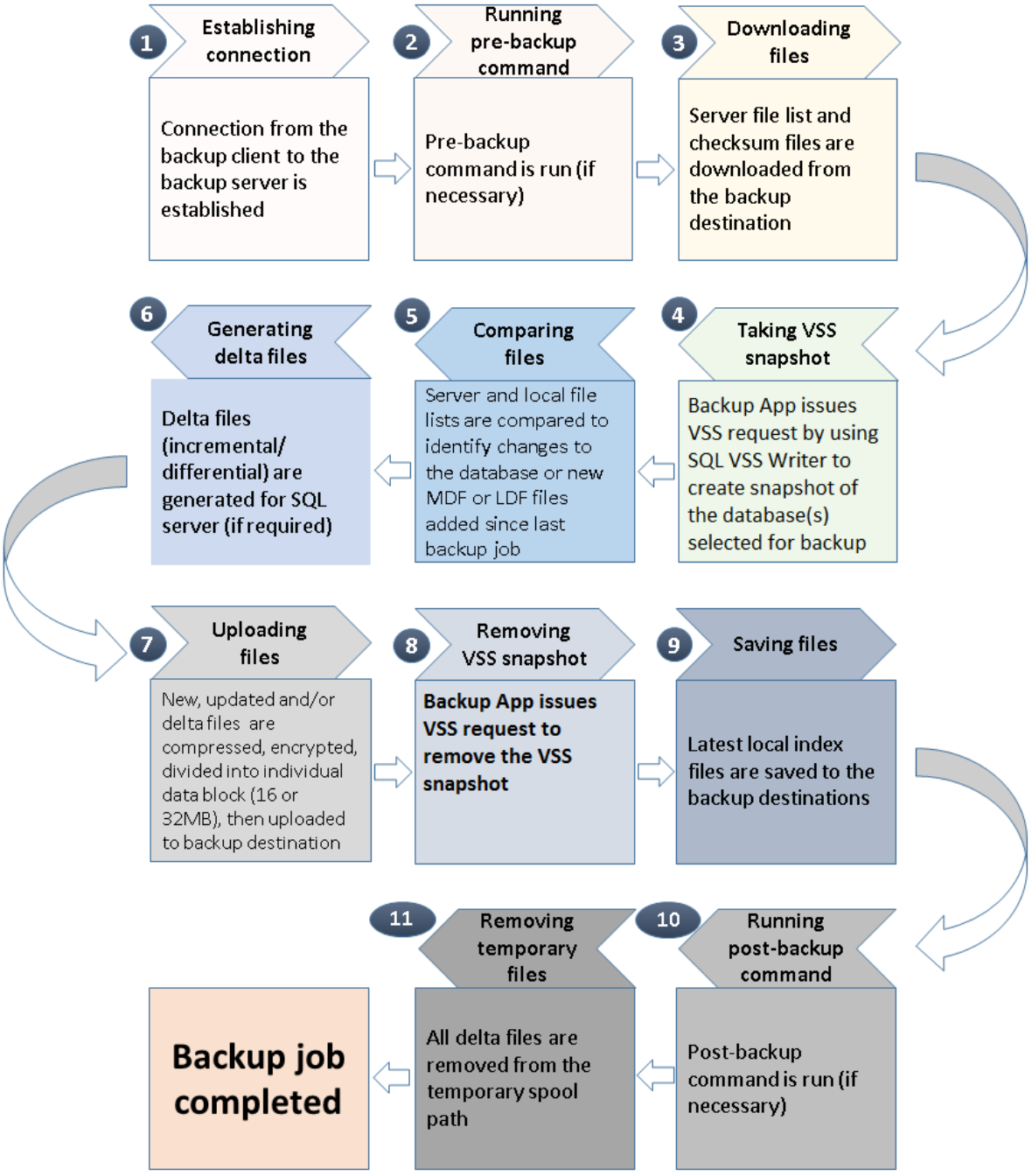
  It is recommended that the temporary directory have disk space of at least 150% of the total database size. For each database backup, Backup App will spool the database files to the temporary directory before they are uploaded to the backup destination. Also, additional space is required for in-file delta generation the default in-file delta ratio settings is 50%, therefore Backup App could generate incremental or differential delta file(s) of up to 50% of the total database size. The actual disk space required depends on various factors, including the size of the database, number of backup destinations, backup frequency, in-file delta settings etc.

Pros
➢ **Support Automated Transaction Logs Backup** Schedule backup of transaction log can be configured so that the transaction logs can be backed up periodically and the transaction logs are truncated automatically after each backup job. ➢ **Support Point in Time Recovery** The ability to restore to a point in time for all of your transaction log backups. ➢ **Support Backup of High Transaction Databases** For databases which supports a high number of transaction which may require frequent backups. Transaction log backups at regular intervals are more suitable and less resource intensive than VSS based backups, i.e. transaction log backup every 60 minutes, 30 minutes, 15 minutes etc depending on the database transaction volume.
Cons
➢ **Large disk space required** Since the database files will be spooled to a temporary folder before uploading to backup destination, investment on hard disk could be high if your MS SQL database size is large. ➢ **Slower backup process** By utilizing the conventional spooling method, it could take a long time to back up the database and the speed is subject to various factors, including database size, network transfer speed, backup frequency, etc.

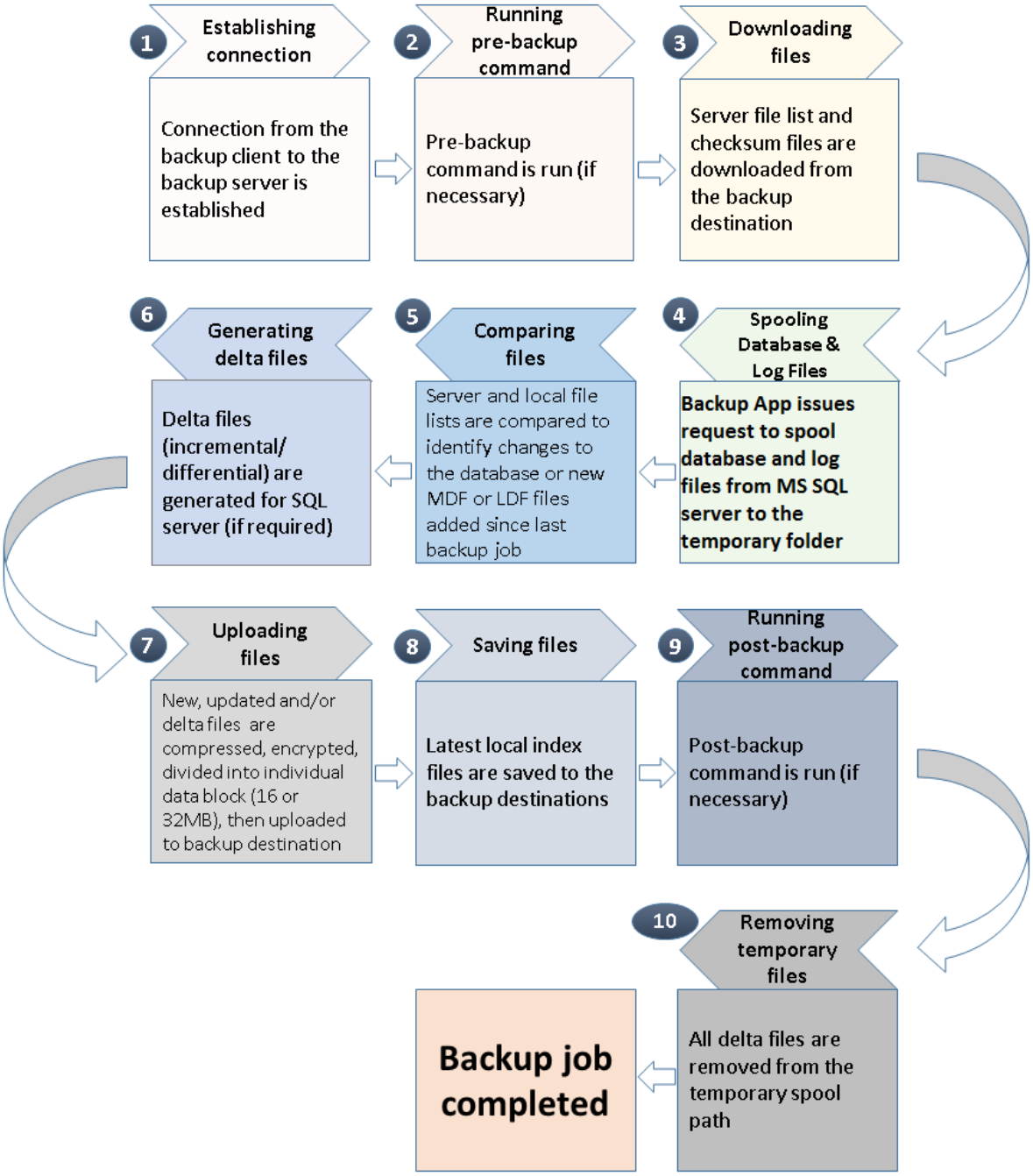
6 Overview of MS SQL Server Backup Process

The following steps are performed during an SQL server backup job:

VSS Backup Mode



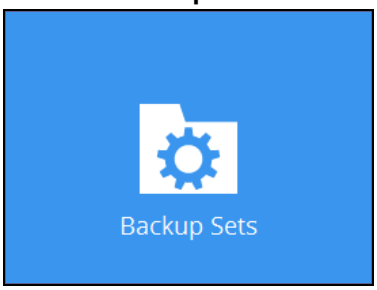
ODBC Backup Mode



7 Performing Backup for Microsoft SQL Server

Creating Backup Set for Microsoft SQL Server

1. Click the **Backup Sets** icon on the main interface of Backup App.

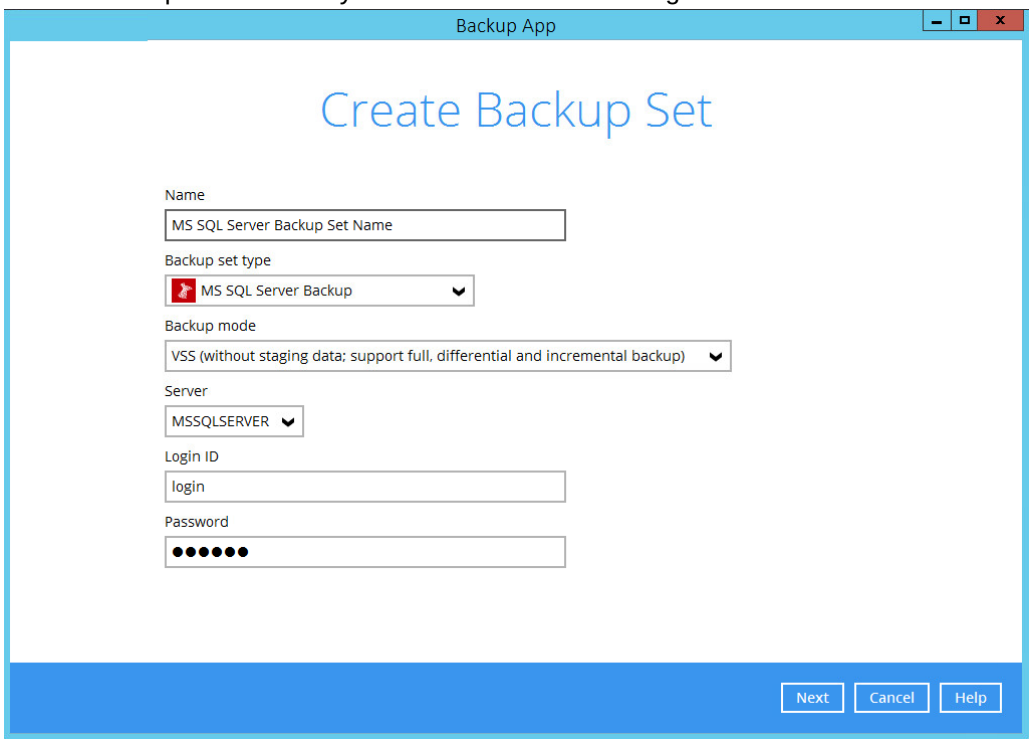


2. Create a new backup set by clicking the "**+**" icon next to **Add new backup set**.

3. Select the Backup set type as **MS SQL Server Backup**.

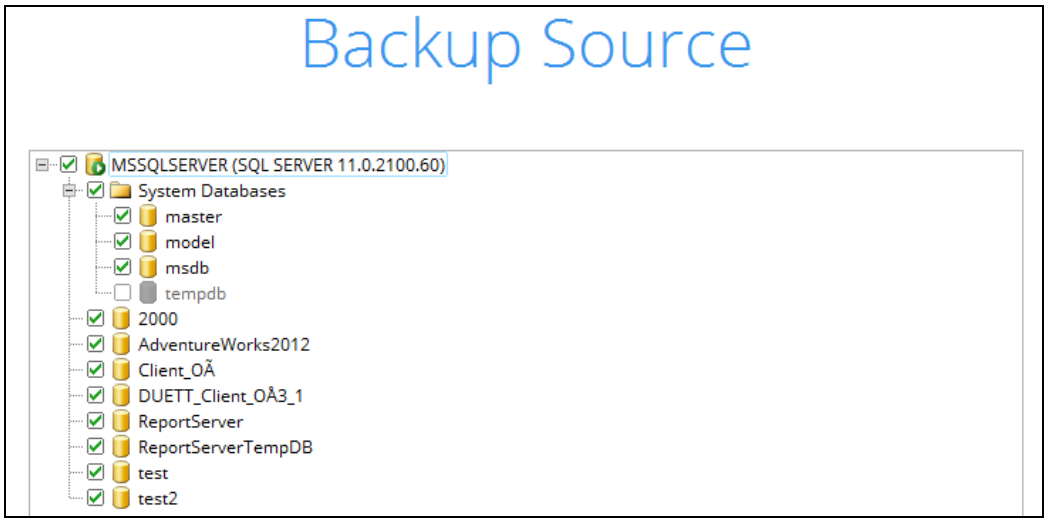
- **Name** – enter a meaningful backup set name
- **Backup mode** – choose between VSS mode and ODBC mode. Refer to the Backup Mode section for details on the differences between the two modes.
- **Server** - Backup App supports backup of multiple SQL instance in one backup set. In this **Server** drop-down menu, you can choose to back up multiple SQL instances or a specific instance of your choice.
- **Login** - Enter the login ID for the chosen instance.
- **Password** – Enter the password for the chosen instance.

Click **Next** to proceed when you are done with the settings.



4. In the **Backup Source** menu, select the database you would like to back up, then click **Next** to proceed.

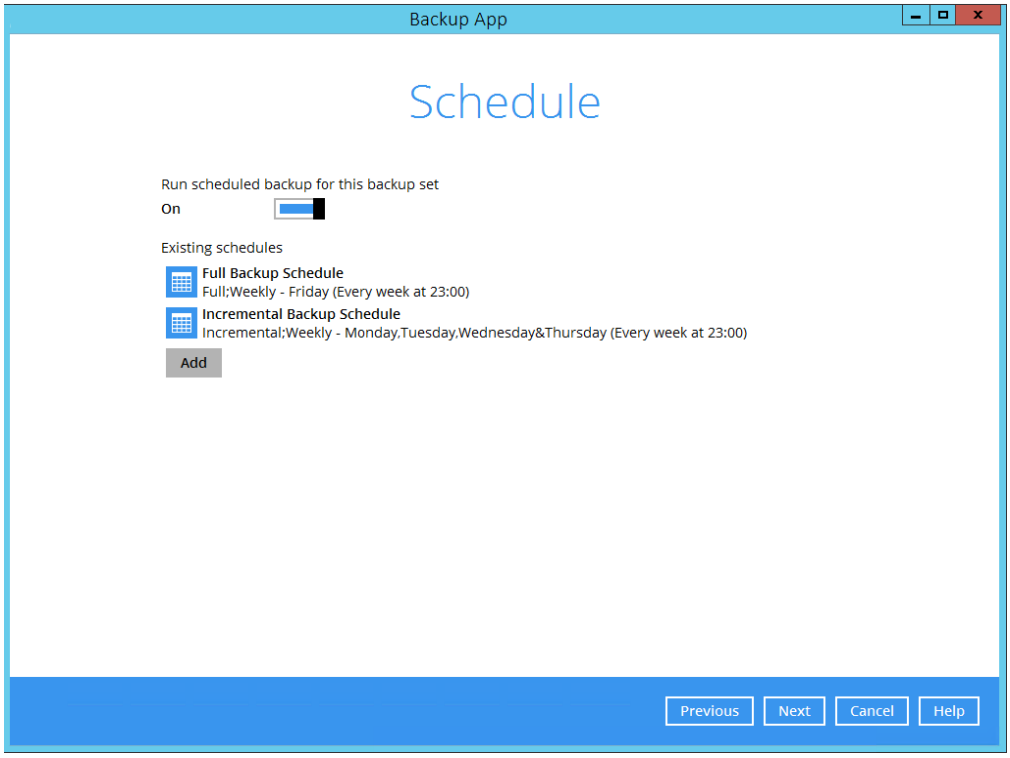
   If you have chosen to back up multiple SQL instances in the previous step, databases in all the chosen instances will be shown here.



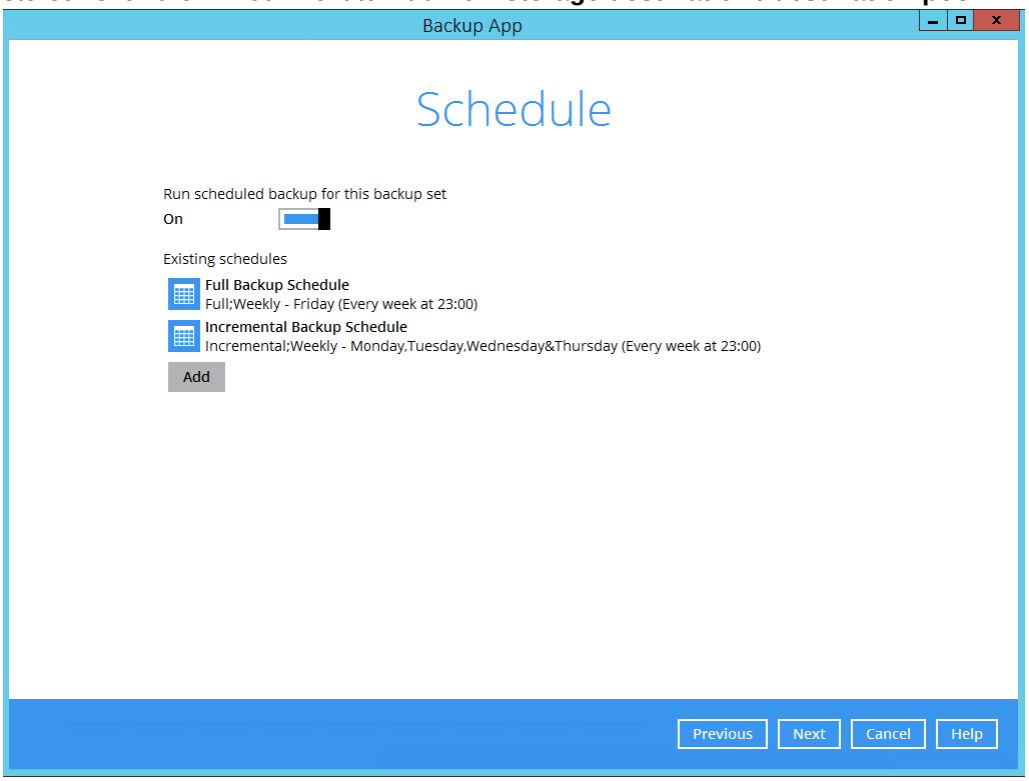
5. In the Schedule menu, you can configure a backup schedule for backup job to run automatically at your specified time interval. Click **Add** to add a new schedule, then click **Next** to proceed when you are done with the settings.

	VSS Mode	ODBC Mode
Name	Name of the Backup Schedule	
Backup set type	➢ Full ➢ Differential ➢ Incremental	➢ Full ➢ Differential ➢ Transaction Log
	Refer to Appendix A for details on the differences of the backup set type.	
Type	Choose frequency for this backup schedule to occur	
Start backup at	Choose a time for this backup schedule to start	
Run Retention Policy after backup	Check this box if you wish to enable the Retention Policy setting	
Default setting	➢ **Full Backup Schedule** Full Backup / Every Friday at 23:00 ➢ **Incremental Backup Schedule** Incremental Backup Type / Mon-Thu every week at	➢ **Full Backup Schedule** Full Backup / Every Friday at 23:00 ➢ **Transaction Log Backup Schedule** Transaction Log Backup Type / Mon-Thu every

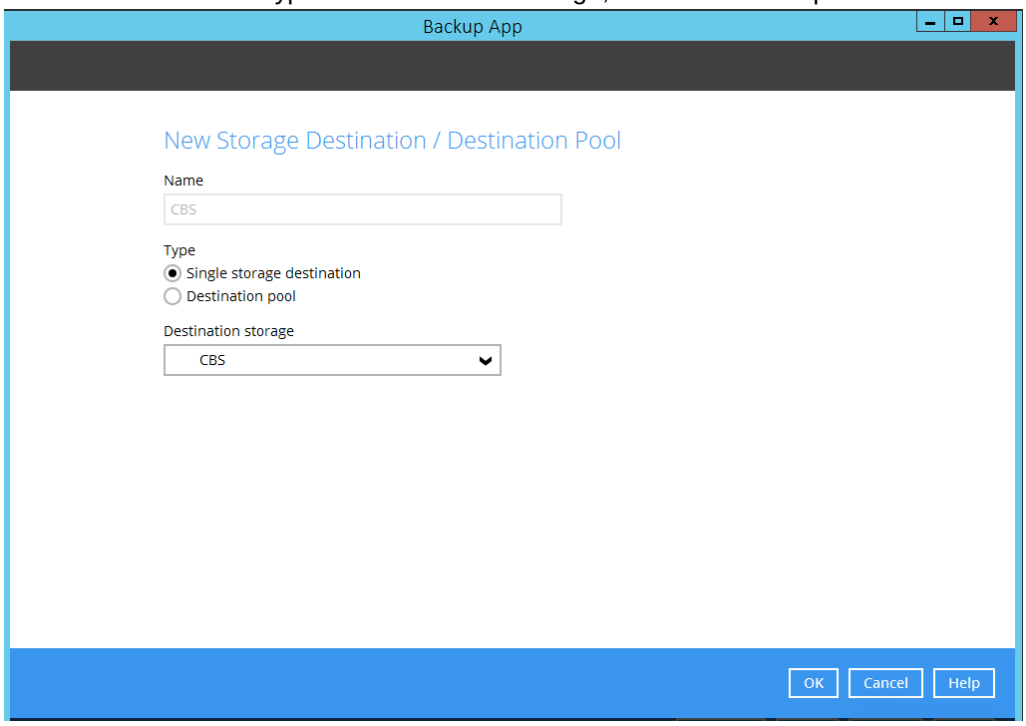
	23:00	week at 23:00



6. In the Destination menu, select a backup destination where the backup database will be stored. Click the "**+**" icon next to **Add new storage destination / destination pool**.

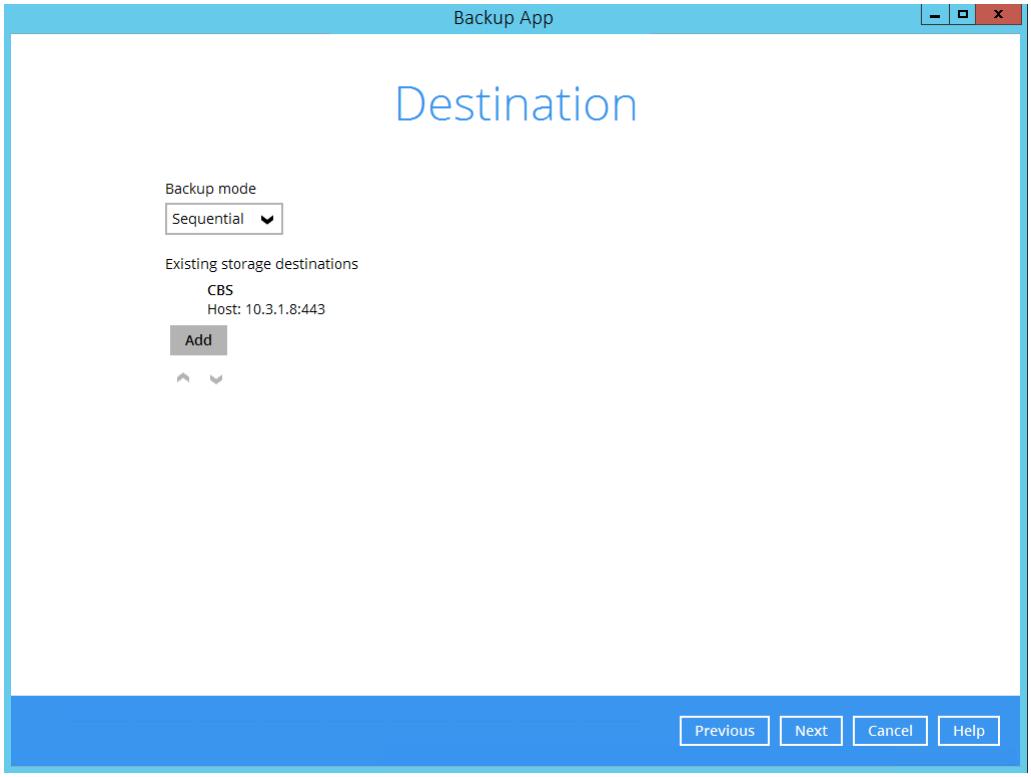


7. Select the destination type and destination storage, then click **OK** to proceed.

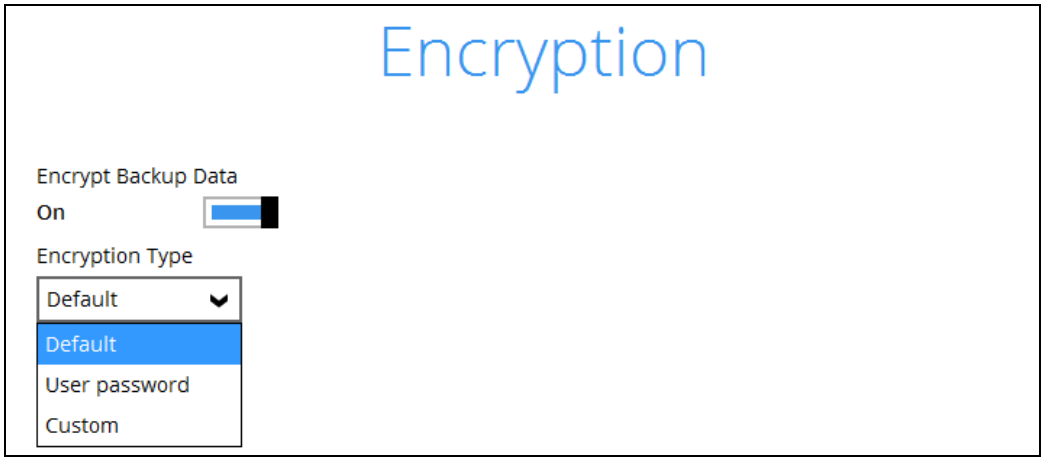


For more information regarding backing up to cloud storage destination, refer to Appendix C Cloud Storage as Backup Destination.

8. Click **Next** on the Destination menu page to proceed.

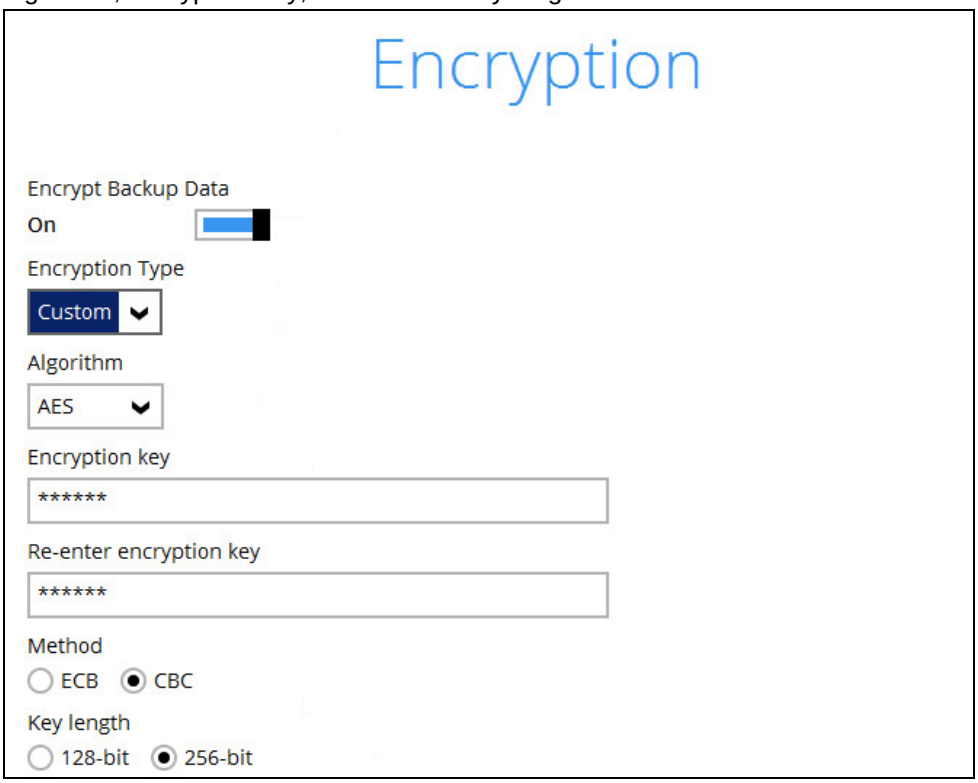


9. In the Encryption window, the default **Encrypt Backup Data** option is enabled with an encryption key preset by the system which provides the most secure protection.



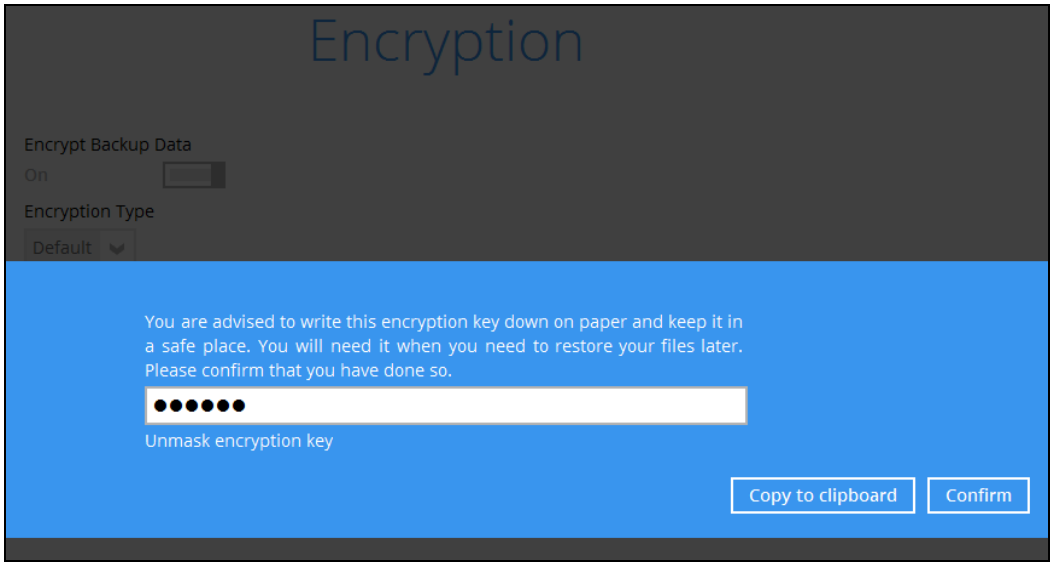
You can choose from one of the following three Encryption Type options:

- **Default** – an encryption key with 44 alpha numeric characters will be randomly generated by the system
- **User password** – the encryption key will be the same as the login password of your Backup App at the time when this backup is created. Please be reminded that if you change the Backup App login password later, the encryption keys of the backup sets previously created with this encryption type will remain unchanged.
- **Custom** – you can customize your encryption key, where you can set your own algorithm, encryption key, method and key length.



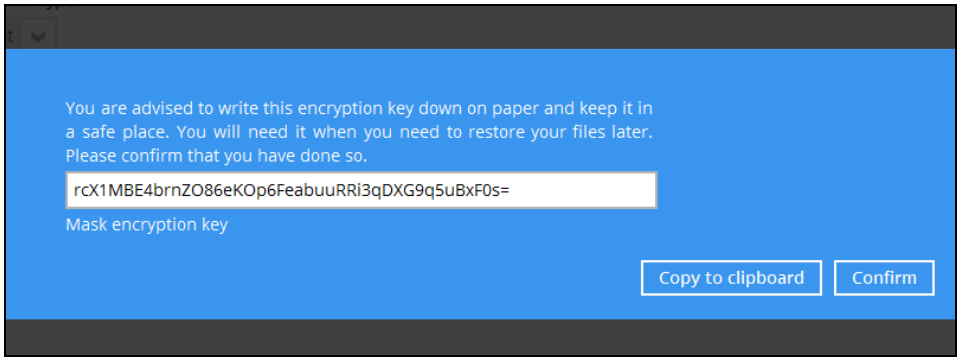
Click **Next** when you are done setting.

10. If you have enabled the Encryption Key feature in the previous step, the following pop-up window shows, no matter which encryption key you have selected.



The pop-up window has the following three options to choose from:

- **Unmask encryption key** – The encryption key is masked by default. Click this option to show the encryption key.

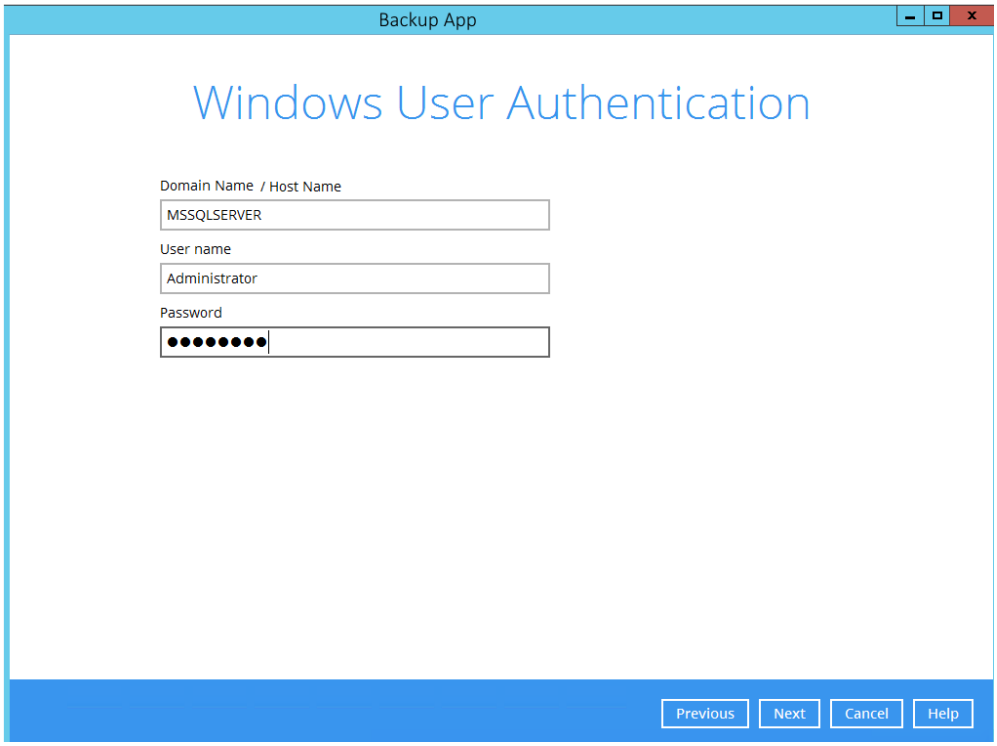


- **Copy to clipboard** – Click to copy the encryption key, then you can paste it in another location of your choice.

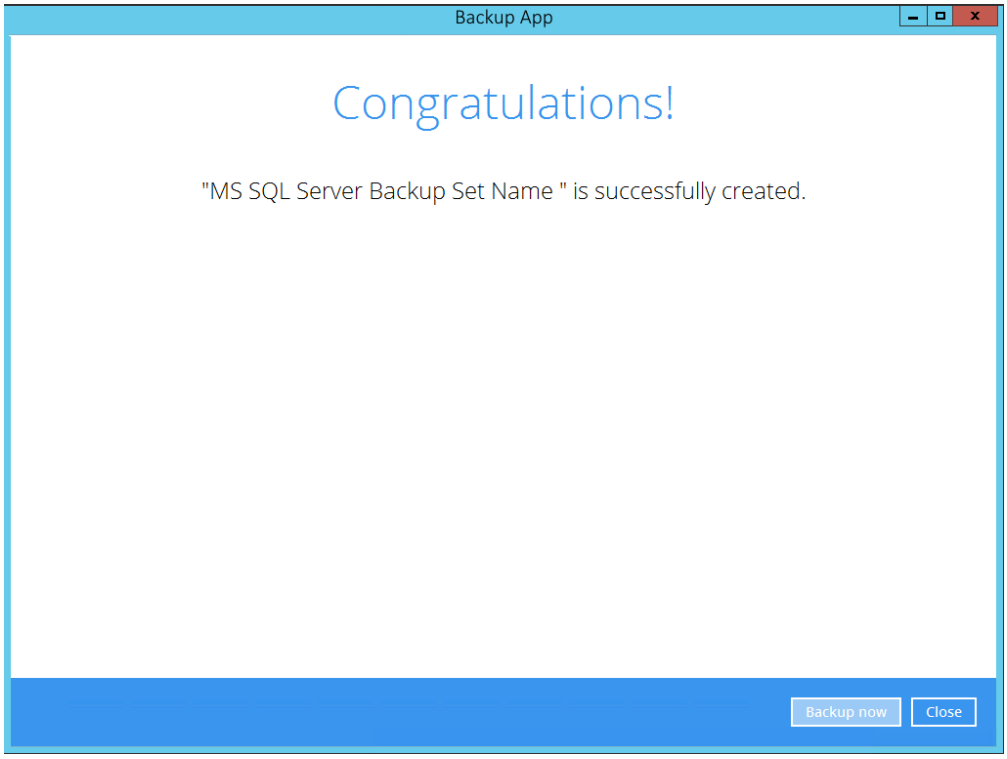
- **Confirm** – Click to exit this pop-up window and proceed to the next step.

11. Enter the Windows login credentials for user authentication. Click **Next** to proceed.

    Note: *This screen shows only if you have configured scheduled backup.*



12. The following screen shows when the new backup set is created successfully.

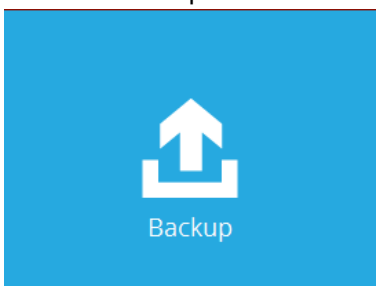


13. Click **Backup now** to start a backup immediately, or you can run a backup job later by following the instructions in Running Backup Job for Microsoft SQL Server.

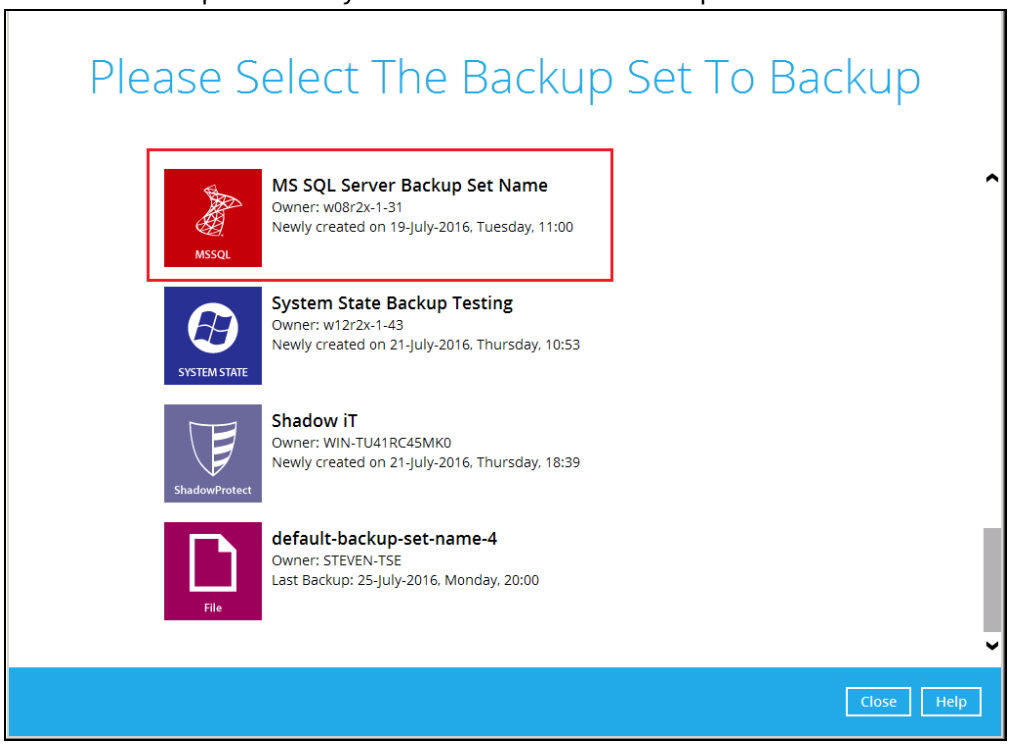
Running Backup Job for Microsoft SQL Server

1. Log in to Backup App.

2. Click the Backup icon on the main interface of Backup App.

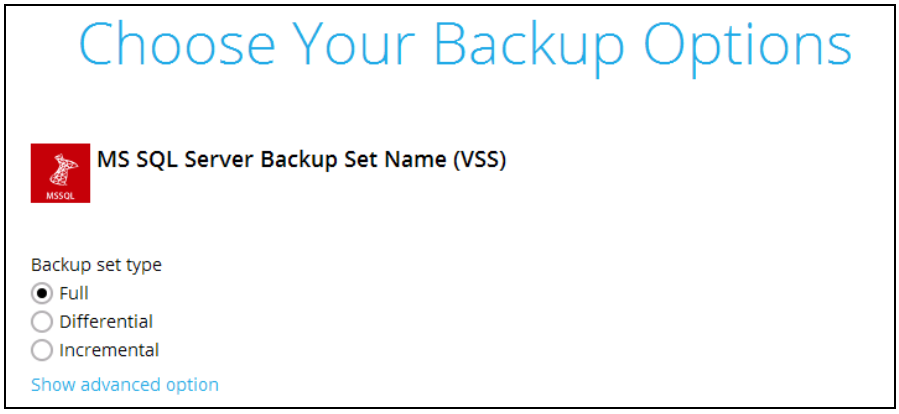


3. Select the backup set which you would like to start a backup for.

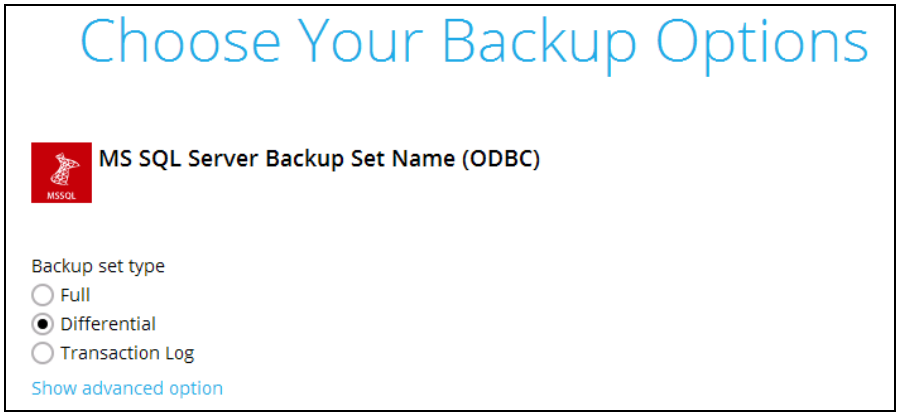


4. Select the Backup set type. For more details regarding the Backup set type & In-file delta type, refer to Appendix A Backup Set Type .

For VSS Backup Mode



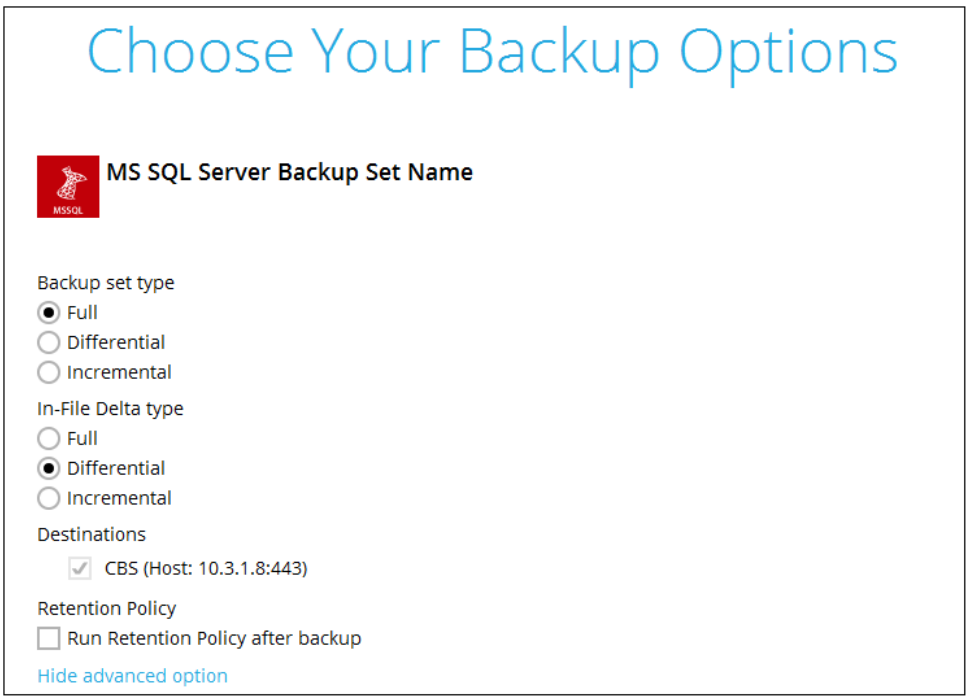
For ODBC Backup Mode



Important

Upon upgrade to Backup AppCBS v7 from Backup AppOBS v6, when attempting to run a transaction log backup for backup sets created on v6 for the **FIRST TIME**, a full backup will be performed instead. As the disk space required for running a full backup set may significantly be larger than running a transaction log backup, make sure the backup destination has enough quota to accommodate the full backup.

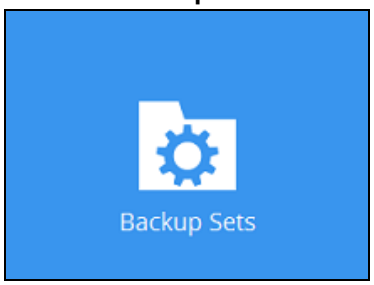
If you would like to modify the In-File Delta type (for Full backup set type only), Destinations and Retention Policy settings, click **Show advanced option**.



5. Click **Backup** to start the backup.

Configuring Backup Schedule for Automated Backup

1. Click the **Backup Sets** icon on the Backup App main interface.



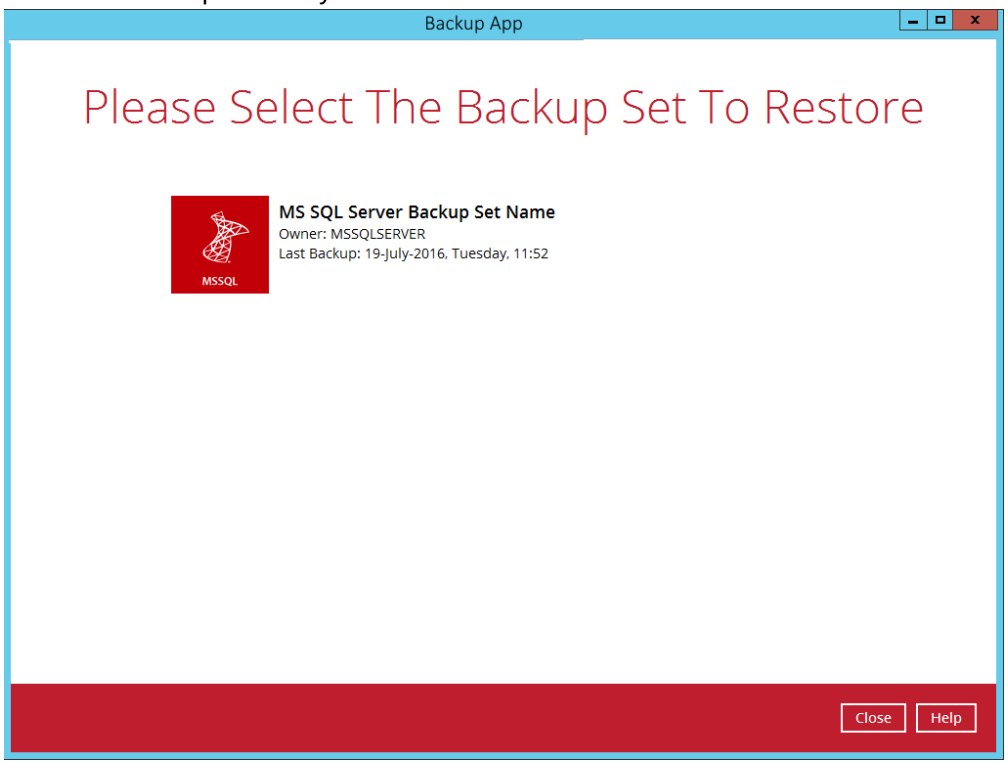
2. Select the backup set that you want to create a backup schedule for.
3. Click **Backup Schedule**, then create a new backup schedule by clicking **Add**.
4. Configure the backup schedule settings, then click **OK** to proceed.
5. Click **Save** to confirm your settings.

8 Restoring Backup for Microsoft SQL Server

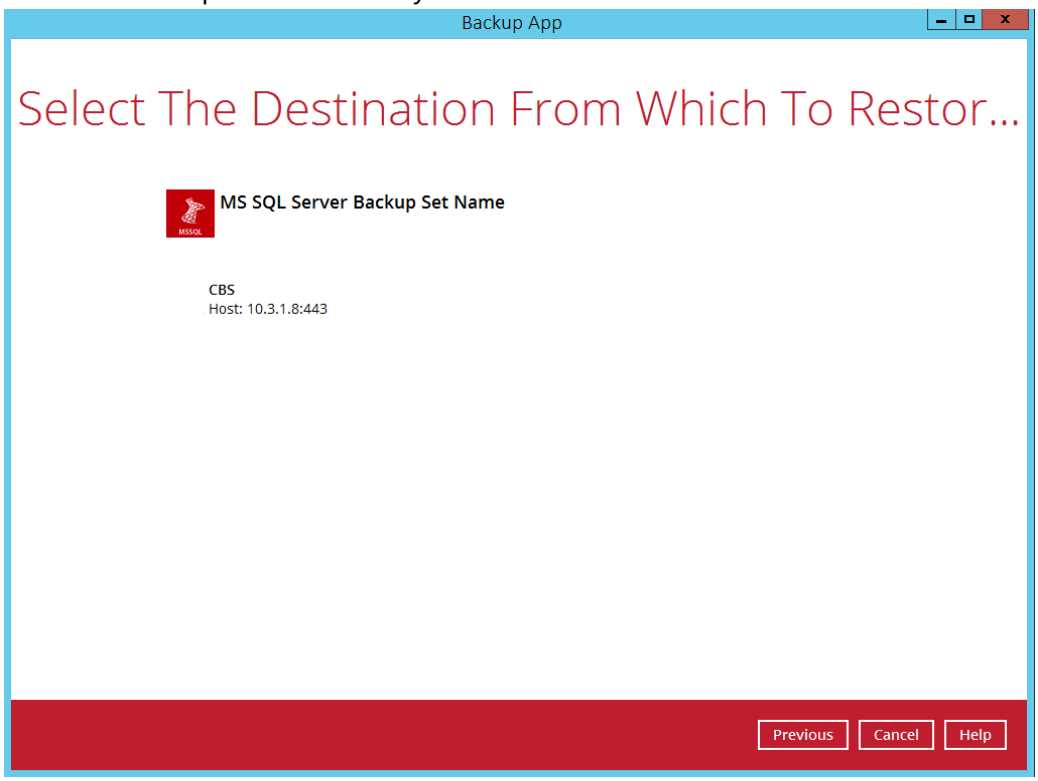
Restoring Backup for Microsoft SQL Server

1. In the Backup App main interface, click the **Restore** icon.

2. Select the backup set that you would like to restore.

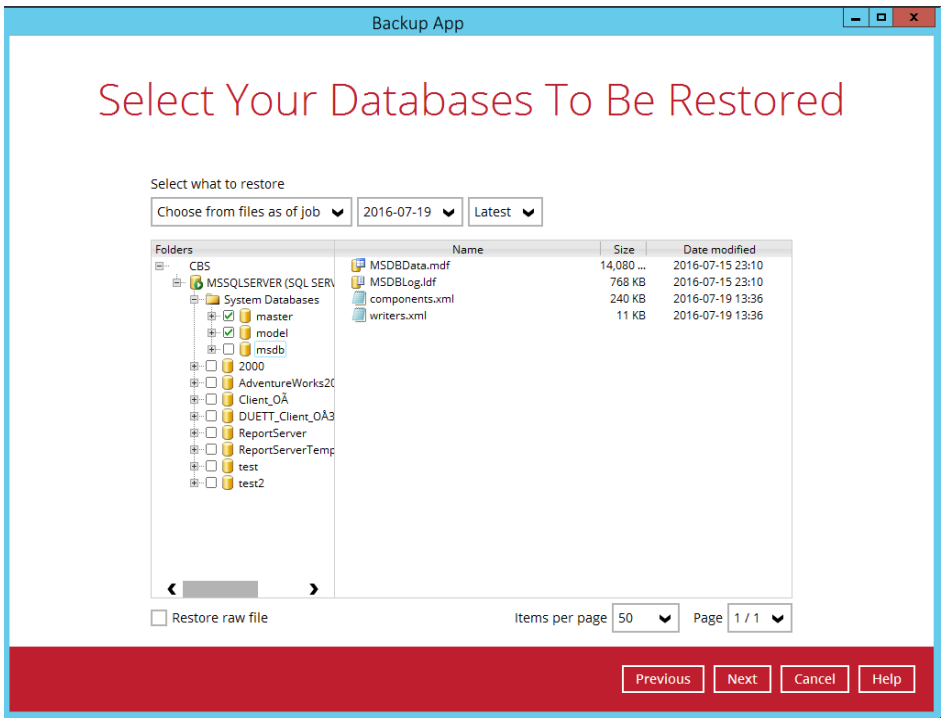


3. Select the backup destination that you would like to restore data from.

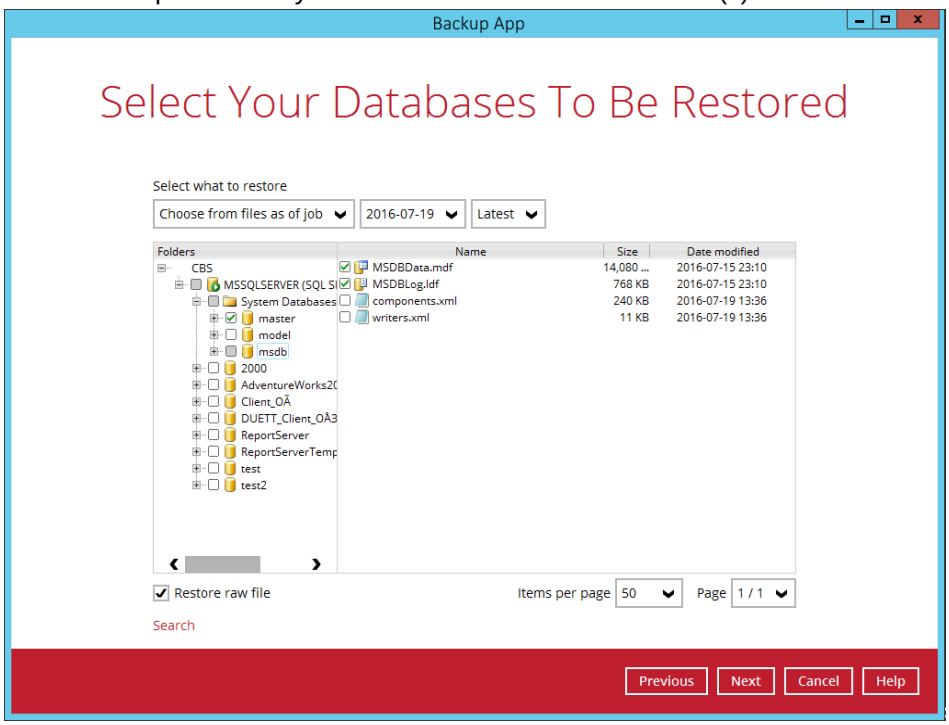


4. Select the database(s) or raw file(s) you would like to restore. You can also choose to restore backed up database or raw file from a specific backup job of your choice using the **Select what to restore** drop-down menu at the top. Click **Next** to proceed when you are done with the selection.

- **Restoring database** - expand the menu tree to select which database to restore. Follow **5a** below to select restoring to the original SQL server or an alternate SQL server.



- **Restoring raw file** - you can select individual raw database file to restore by clicking the **Restore raw file** checkbox at the left bottom corner. Follow **5b** below to select the path where you would like to restore the raw file(s) to.



Limitations:

- If you would like to restore database with the Alternate location option, you can only choose to restore one database at a time.
- If you would like to restore database to an alternate SQL server with the **Restore raw file** option, make sure you have checked the **Restore raw file** option.

5. Select the destination to restore. Refer to 5a or 5b below for steps to restore the database automatically (Restore database to Original/Alternate location) or manually (Restore raw file).

5a. Select to restore the database to its Original SQL server, or to an Alternate SQL server.

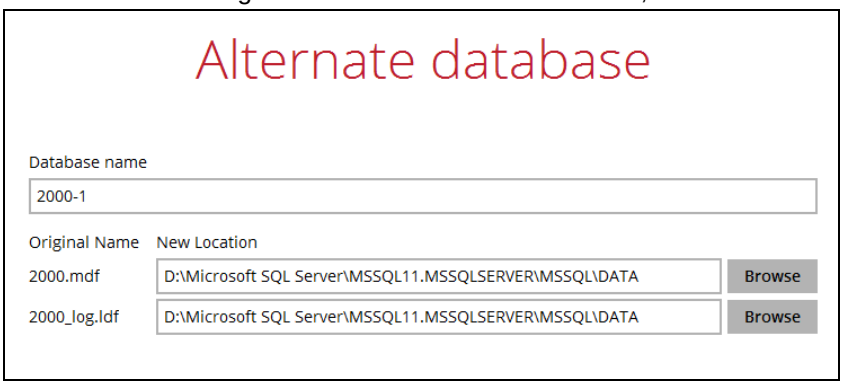
- **Restore to Original SQL server**

  Select the **Original location** option, then press **Next** to proceed.

- **Restore to Alternate SQL server (only for restoring raw file)**

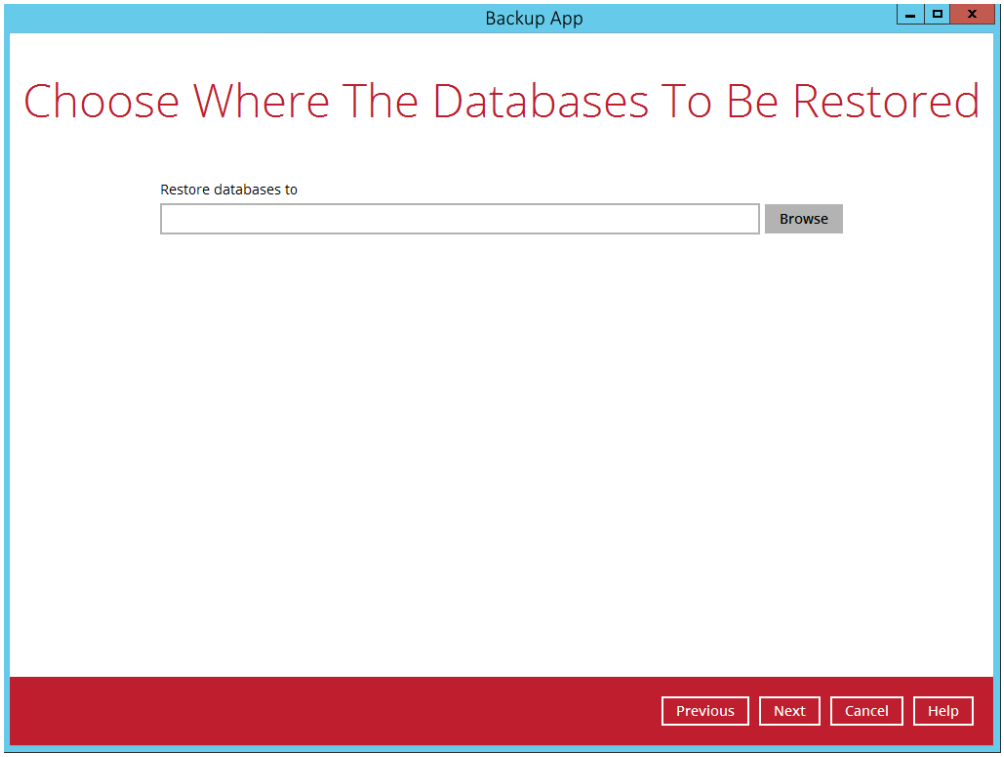
i. Select the **Alternate location** option, then press **Next**.

ii. Click **Browse** to select the locations where you would like to restore the database and log files to. Name the new database, then.



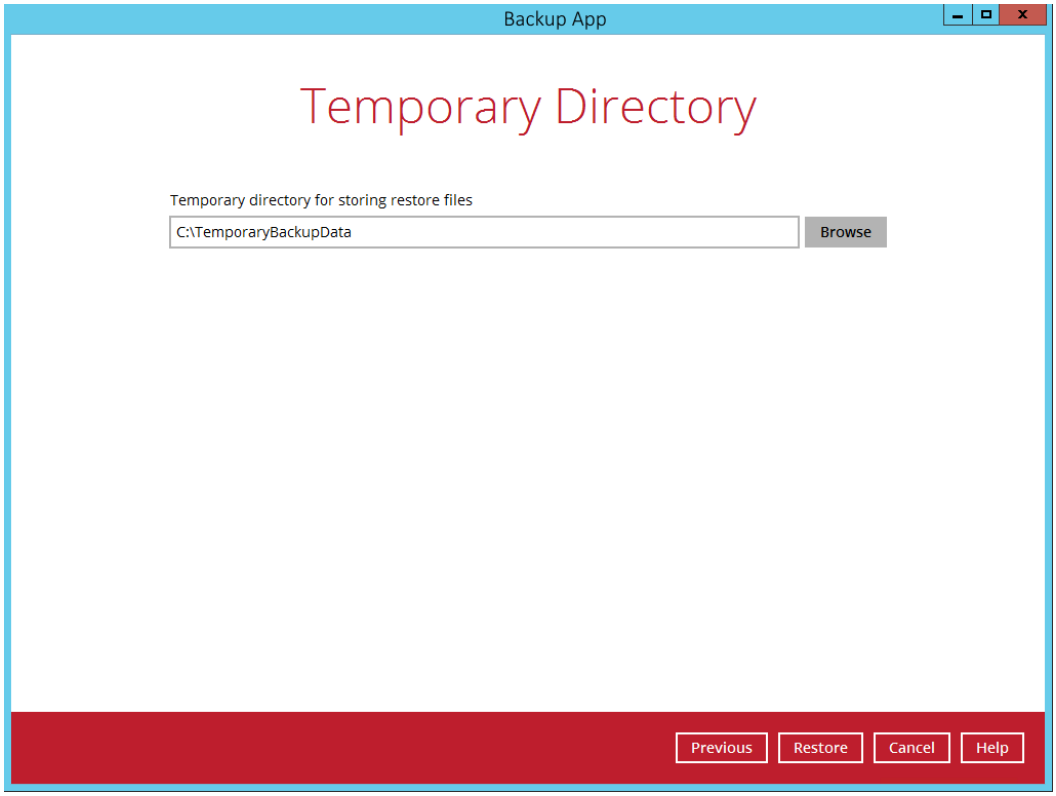
iii. Click **Next** to proceed when you are done with the settings.

5b. i) If you have chosen to restore raw file, choose the location path where you would like the raw file(s) to be restored to. Click **Next** to proceed.

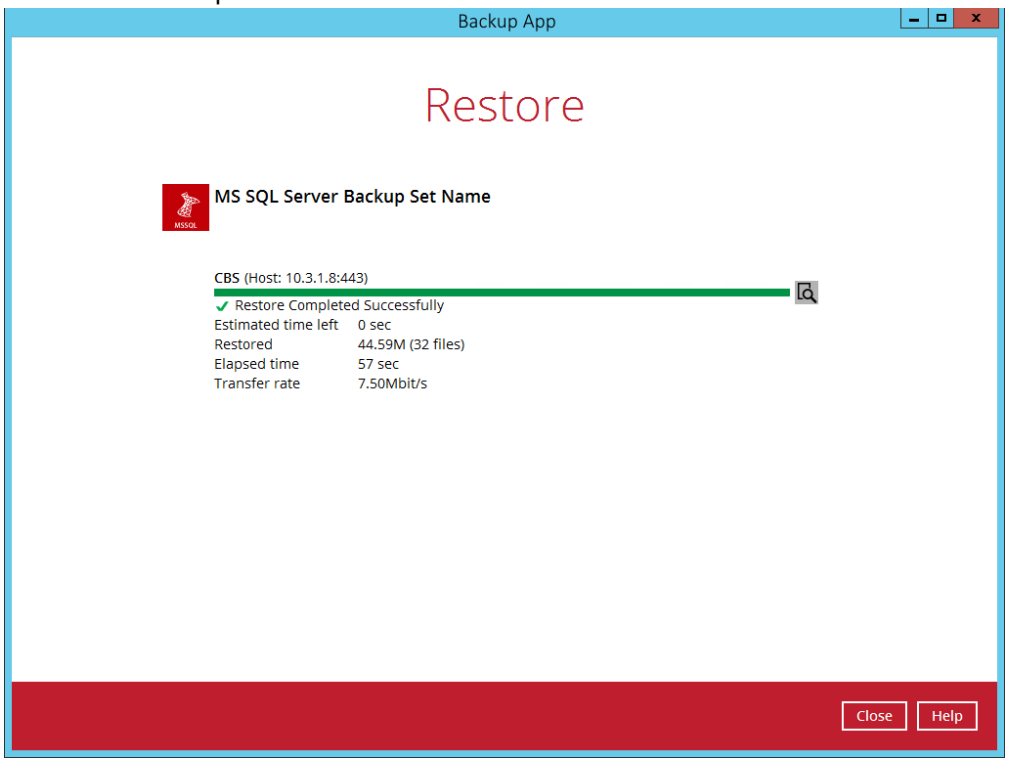


ii) Restore the database manually with the restored database file via the SQL Server Management Studio. Refer to the MS KB article below for instructions.
https://technet.microsoft.com/en-us/library/ms177429%28v=sql.110%29.aspx

6. Select the temporary directory for storing temporary files, such as delta files when they are being merged, click **Restore** to start the restoration.



7. The following screen with the text **Restore Completed Successfully** shows when the restoration is completed.

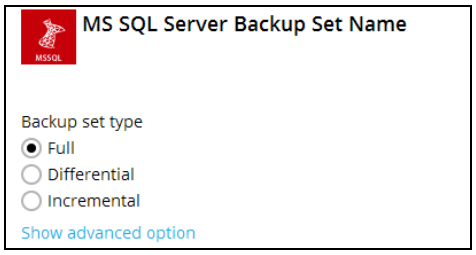


Appendix

Appendix A Backup Set Type

There are three kinds of backup set type to choose from, namely the full backup, differential backup and incremental backup. The information below gives you an overall idea of what each backup set type is like.

Full backup (with configurable in-file delta type)

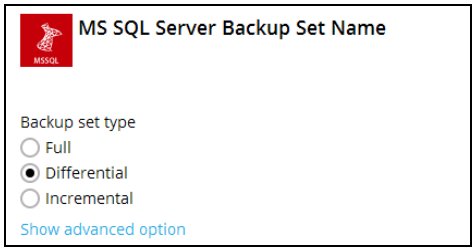


To perform a full backup, Backup App requests the SQL server to generate a Volume Shadow Copy Service (VSS) snapshot of the database. Backup App will back up the VSS snapshot generated by the SQL server directly. A full backup is required in order to run incremental or differential backups.

You can also decide how the full backup is run by selecting the desired in-file delta type (Full, Differential or Incremental).

For further details on this topic, refer to the URL below.
https://msdn.microsoft.com/en-us/library/ms175477.aspx

Differential backup



A differential backup of the SQL server saves changes to the database that have occurred since the last full backup. To perform a differential backup, Backup App requests the SQL server to generate a differential backup file of the database since the last full backup. At the back end, the SQL server performs the following:

1. Generate a VSS snapshot of the database of the current state.
2. Compare the VSS snapshot just generated by the SQL server with the one generated from the last full backup in order to produce a differential backup file.
3. The differential backup file being sent to Backup App for backup.

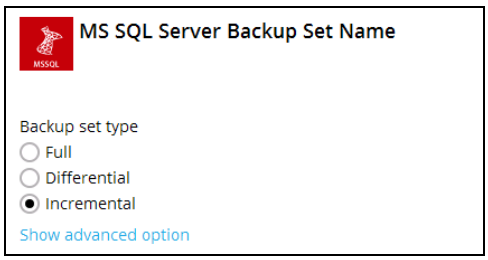
Using a differential backup file to recover a database requires the restoration of only two data sets - the last full backup and the most recent differential backup.

The disadvantage of using differential backups is that it duplicates the backed up data in each backup until a full backup is performed. If there are many differential backups taken between full backups, the storage space required can greatly exceed that required by the same number of incremental backups.

The SQL server does not allow a differential backup to occur when there has been no previous full backup to establish the starting point.

For further details on this topic, refer to the URL below.
https://msdn.microsoft.com/en-us/library/ms186289.aspx

Incremental backup



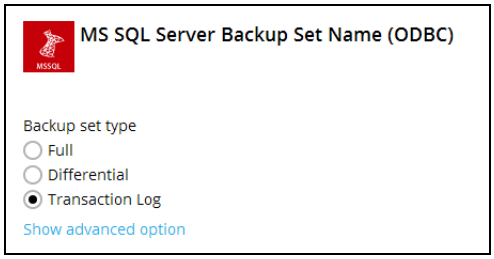
An incremental backup of the SQL server saves changes to the database that have occurred since the last full or incremental backup. To perform an incremental backup, Backup App requests the SQL server to generate a differential backup file of the database since the last full backup. At the back end, the SQL server performs the following:

1. Generate a VSS snapshot of the database of the current state.
2. Compare the VSS snapshot just generated with the one generated from the last full backup in order to produce a differential backup file.
3. The differential backup file being sent to Backup App.
4. Backup App performs an in-file delta check between the differential backup file just received from the SQL server and the one from the last backup.
5. Backup App will then be able to generate an incremental delta file which contains changes of the database files since last differential backup. Only this incremental delta file will be backed up.

Using an incremental backup to recover a database requires the restoration of at least two data sets - the last full backup and every incremental backup taken after the last Full backup. The benefit of using incremental backups is that the individual backups are much smaller than a full backup and individual incremental backups are frequently smaller than differential backups.

The disadvantage of using incremental backups is that if there are many incremental backups made between full backups, recovering the storage group may involve recovering many incremental backups. The SQL server does not allow an incremental backup to occur when there has been no previous full backup to establish the starting point.

Transaction log



Every SQL Server database has a transaction log that records all transactions and the database modifications made by each transaction. The transaction log is a critical component of the database. If there is a system failure, you will need that log to bring your database back to a consistent state.

If you have chosen to back up in ODBC mode, you can configure schedule backup to back up the transaction log regularly at a time interval of your choice.

Important

Upon upgrade to Backup AppCBS v7 from Backup AppOBS v6, when attempting to run a transaction log backup for backup sets created on v6 for the **FIRST TIME**, a full backup will be performed instead. As the disk space required for running a full backup set may significantly be larger than running a transaction log backup, make sure the backup destination has enough quota to accommodate the full backup.

Appendix B Truncating Transaction Log

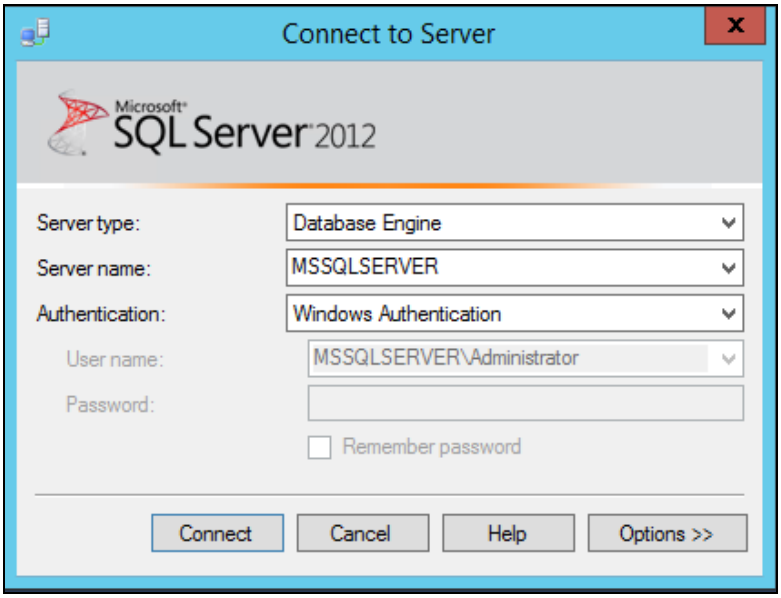
The instructions below only apply for database with full recovery model.

Since Backup App 7 utilize VSS-based backup, which does not support log backup (https://technet.microsoft.com/en-us/library/cc966520.aspx), transaction log of database in full / bulk-logging recovery model may eventually fill up all disk space available on the volume

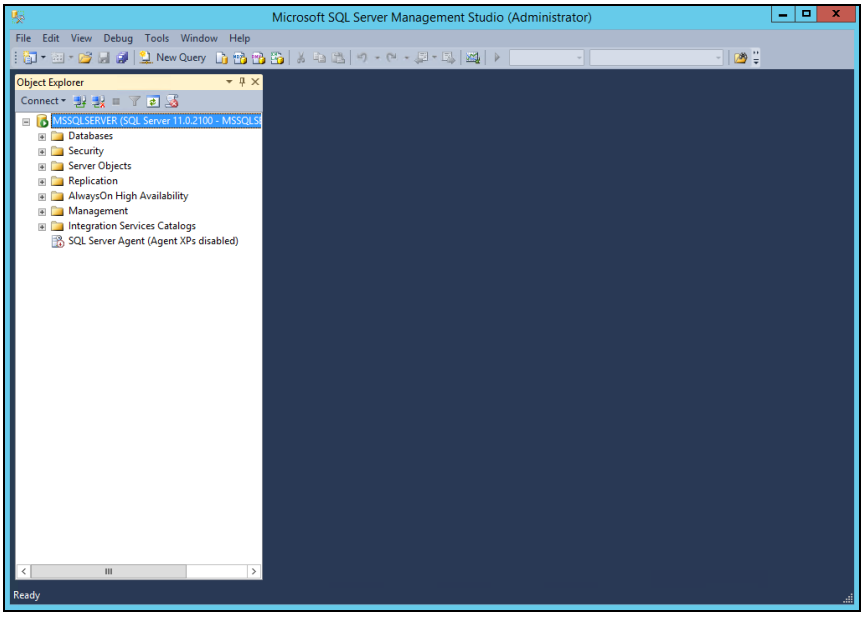
Below are steps to perform a log backup in the SQL Server Management Studio. For further details on this topic, refer to the URL below.
https://msdn.microsoft.com/en-us/library/ms179478.aspx

1. Launch SQL Server Management Studio in Windows.

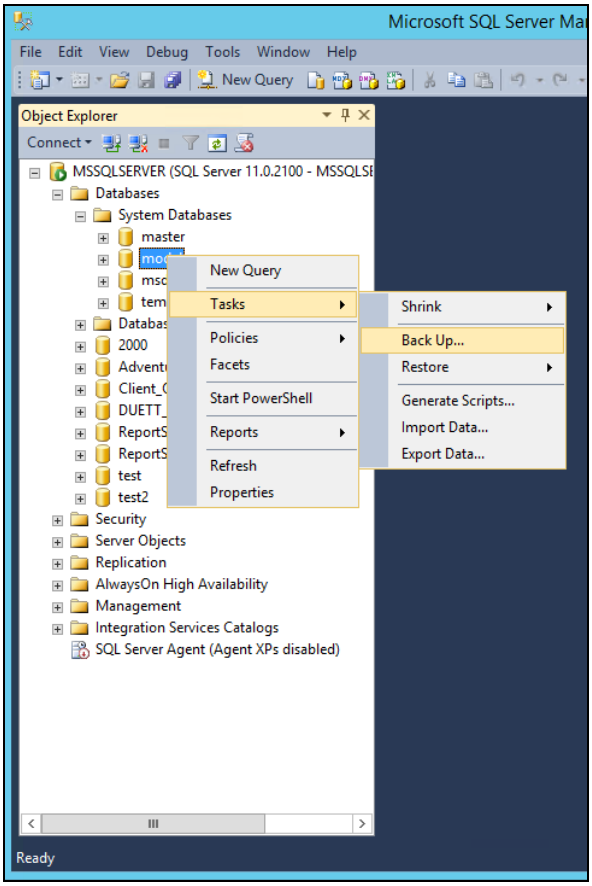
2. Select the SQL server you would like to connect to, and the corresponding authentication method, then click **Connect** to proceed.



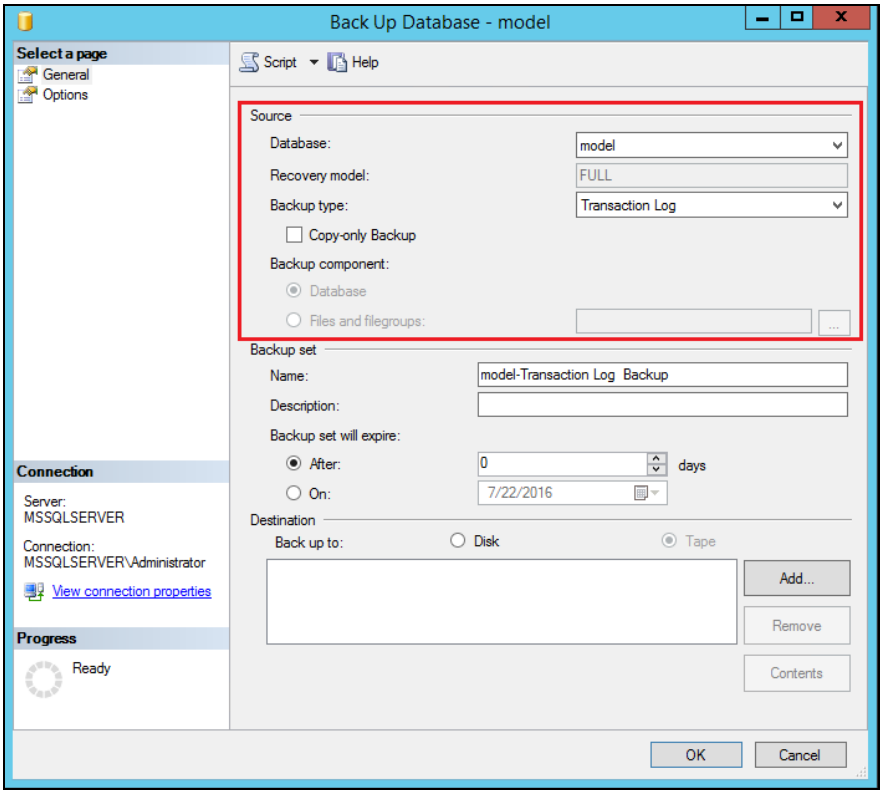
3. Expand the menu tree and select the desired database you would like to back up.



4. Right click the database name, then go to **Tasks** > **Back Up**. The Back Up Database dialog box shows.

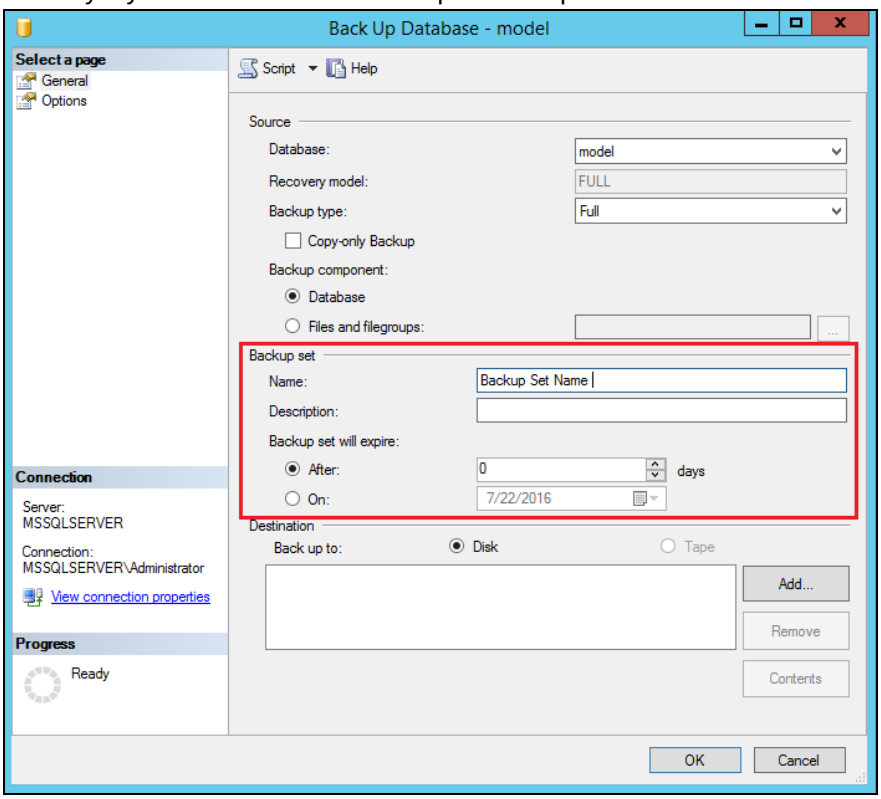


5. In the **Source** section, confirm the database name, then select Transaction Log in the **Backup type** drop-down menu.

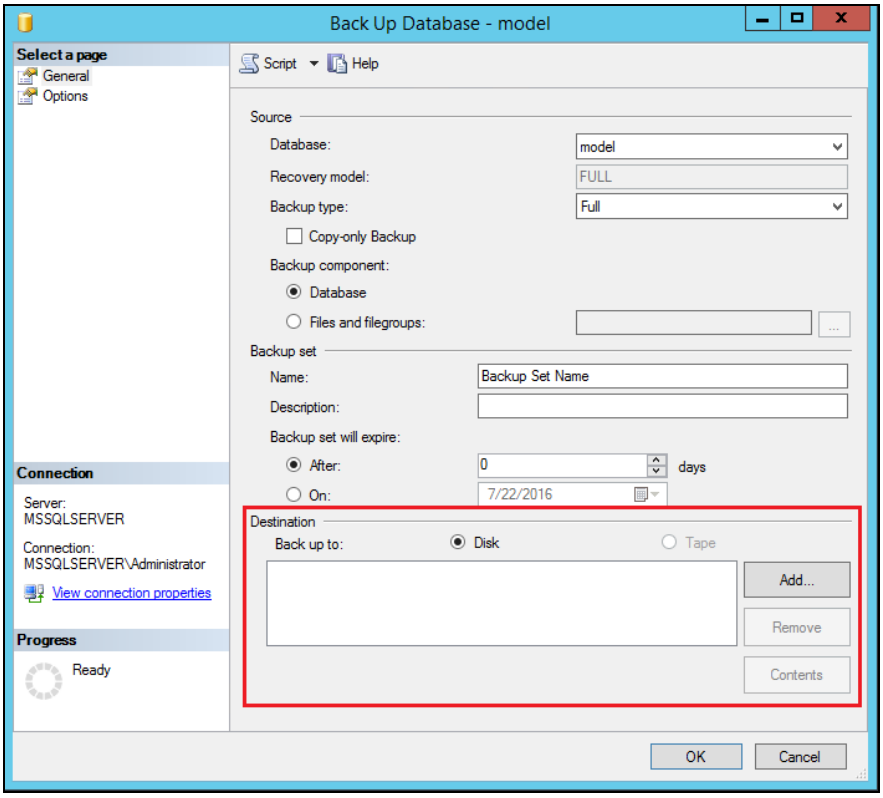


6. In the **Backup set** section, name the backup set and enter a description of the backup set if needed.

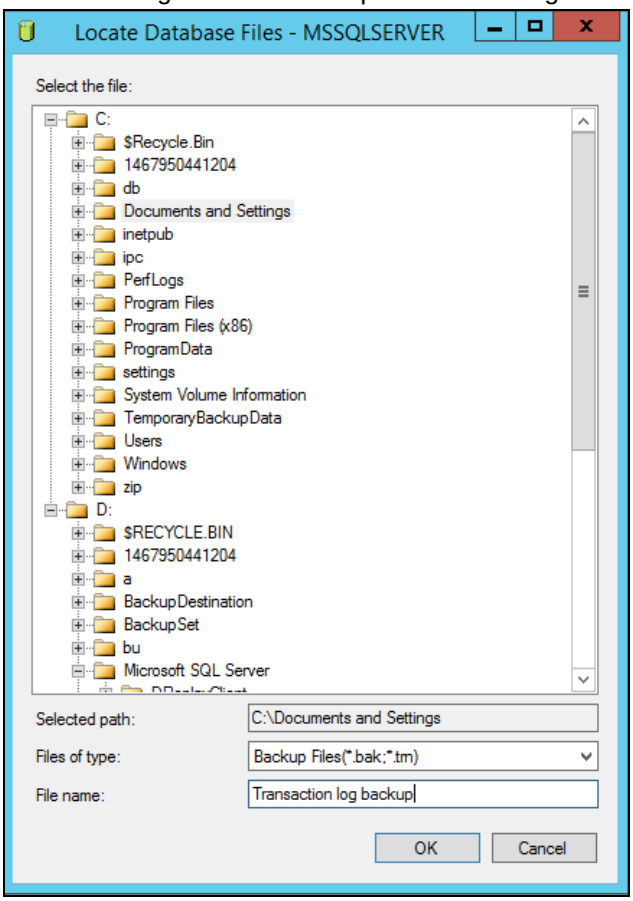
   Configure the Backup set to expire after a specified number of day or on a specified date. Set to 0 day if you do not want the backup set to expire



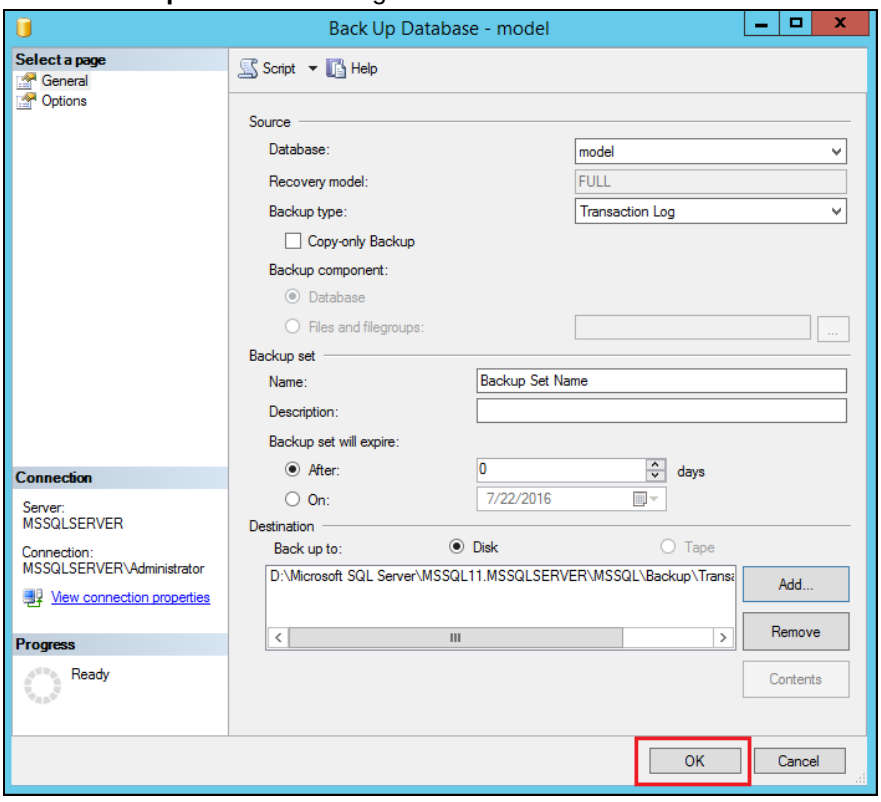
7. Select **Disk** or **Tape** as the destination of the backup, then click **Add** to select a destination path.



8. After selecting the destination path and naming the backup file, then click **OK** twice to proceed.



9. Click **OK** to start the transaction log backup when you are done with all the necessary settings in the **Back Up Database** dialog box.

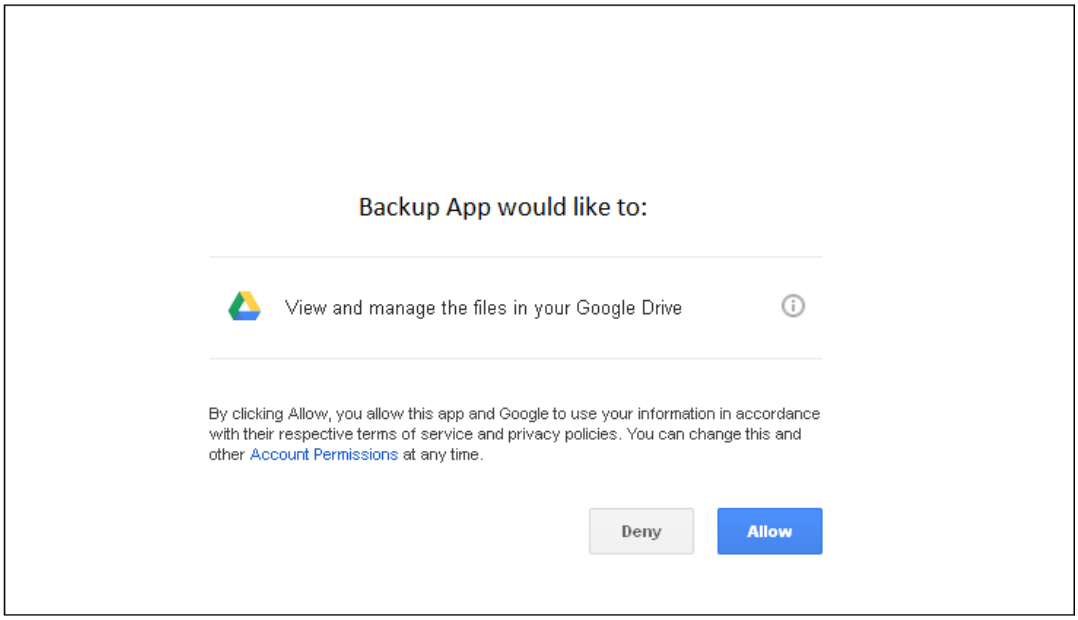


Appendix C Cloud Storage as Backup Destination

For most cloud storage provider (e.g. Dropbox, Google Drive ... etc.), you need to allow access Backup App to access the cloud destination. Click **OK** / **Test**, you will be prompted to log in to the corresponding cloud service.

Important: The authentication request will be opened in a new tab / window on the browser, ensure that the pop-up tab / window is not blocked (e.g. pop-up blocker in your browser).

Click **Allow** to permit Backup App to access the cloud storage.



Enter the authentication code returned in Backup App to complete the destination setup.

Note: *A backup destination can be set to a supported cloud storage, backup server, FTP / SFTP server, network storage, or local / removable drive on your computer.*

Multiple backup destinations can be configured for a single backup set. In fact it is recommended for you to setup at least 2 backup destinations for your backup set.