DDBoost for SQL Server

We have a new backup tool EMC DDBoost for MSSQL Server which is connected to a data domain on servers. There are some steps involved in setting this tool on each SQL Server for taking regular database backups. The main feature of this tool is that it compresses the backups using the deduplication technique while improving the speed of back up process. The entire backup process is also simplified and quicker since we are directly backing up databases to a data domain appliance.

Contents

[Install the software: 2](#_Toc470013882)

[Configuring DDBoost for MSSQLServer 7](#_Toc470013883)

[Setting up SQL Server Agent Jobs : 15](#_Toc470013884)

[How to perform database backup using DDBoost SSMS plug in: 17](#_Toc470013885)

[How to perform Restore using DD Boost SSMS plug in: 21](#_Toc470013886)

[Maintenance of backup files in the data domain 27](#_Toc470013887)

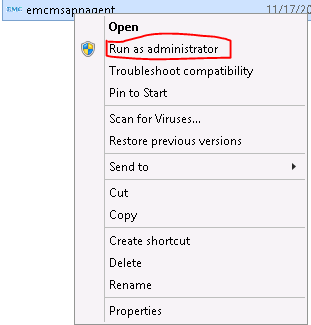
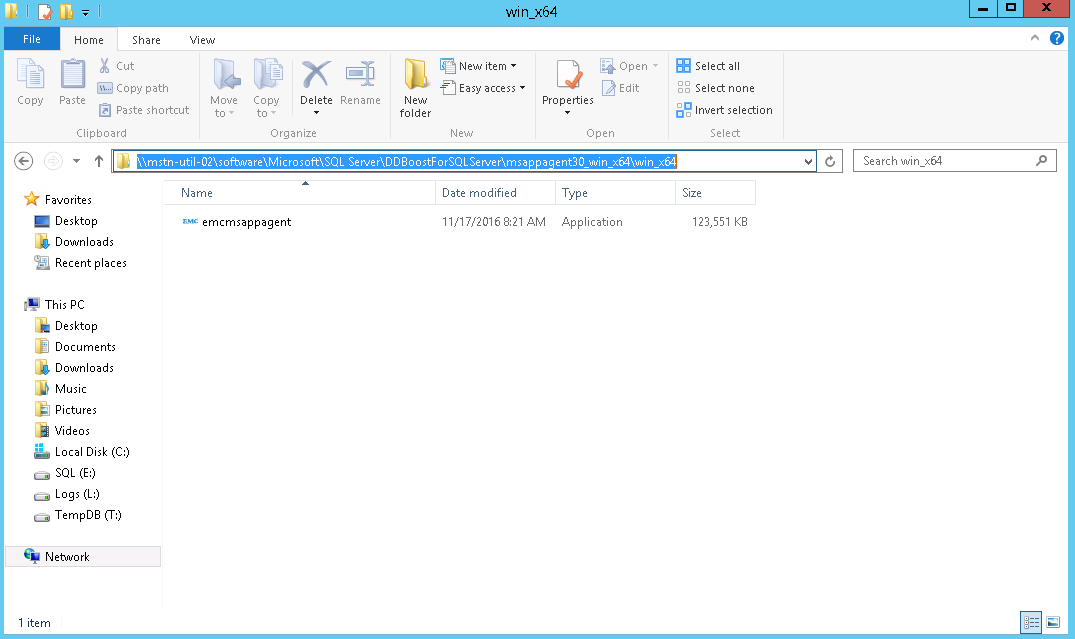
# Install the software:

V 3.0

[~~\\mstn-util-02\software\Microsoft\SQL Server\DDBoostForSQLServer\msappagent30\_win\_x64\win\_x64~~](file:///\\mstn-util-02\software\Microsoft\SQL%20Server\DDBoostForSQLServer\msappagent30_win_x64\win_x64)

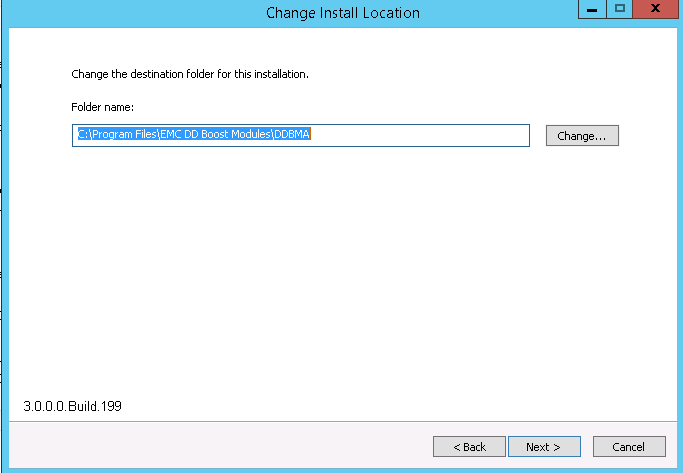
V3.5

\\mstn-util-02\software\Microsoft\SQL Server\DDBoostForSQLServer\msappagent35\_win\_x64\win\_x64

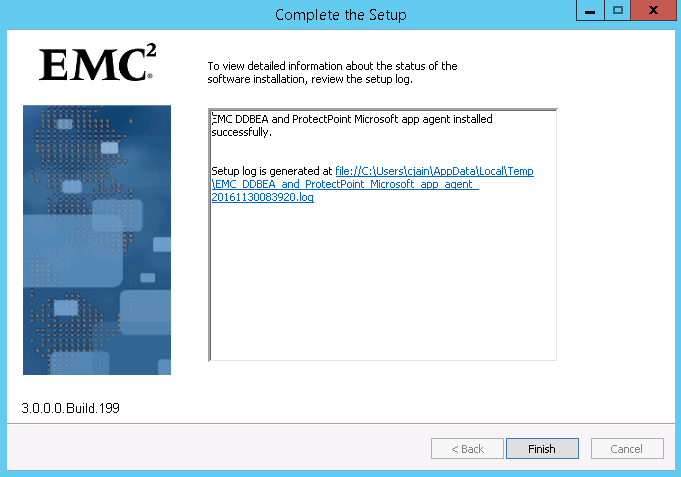
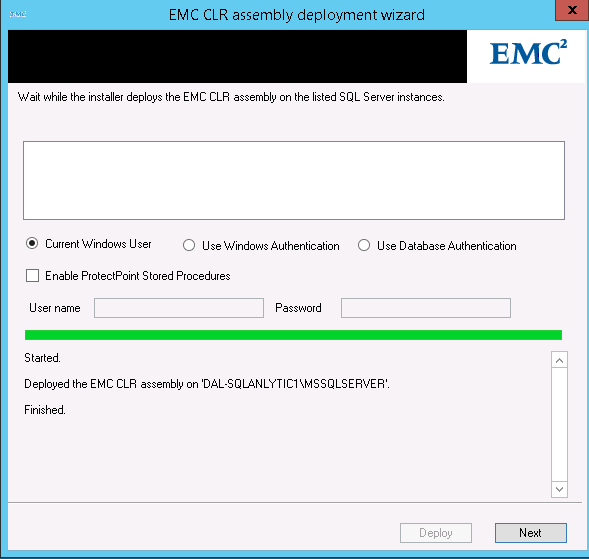
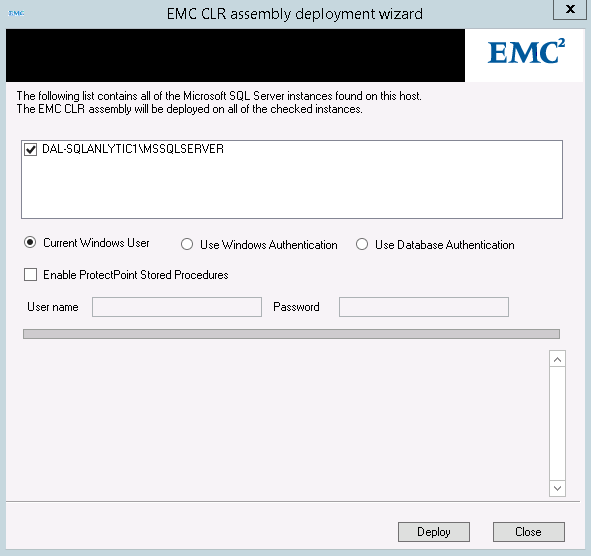
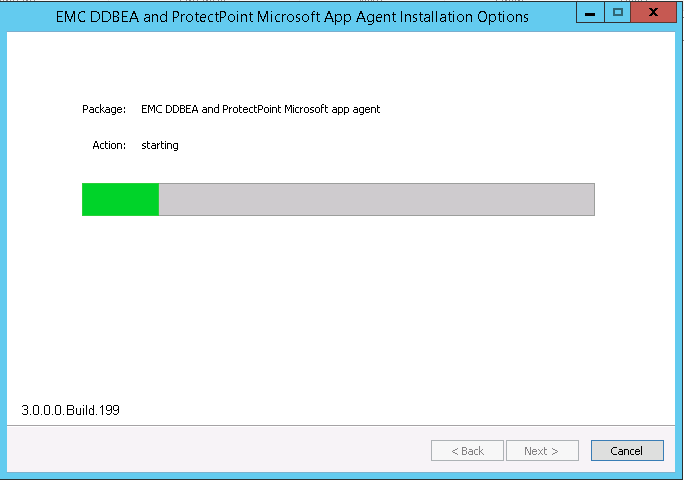
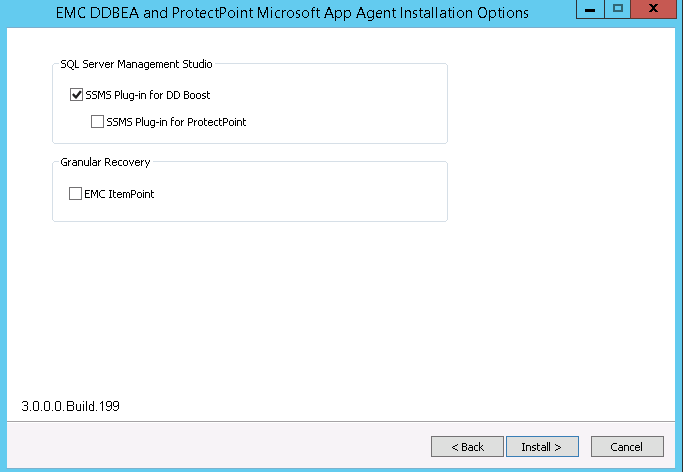


Right-click on **emcmsappagent** application file and select “Run as administrator”.





(V 3.5 WILL HAVE A DIFFERENT PATH, C:\Program Files\DPSAPPS\MSAPPAGENT)

 Hit Finish.

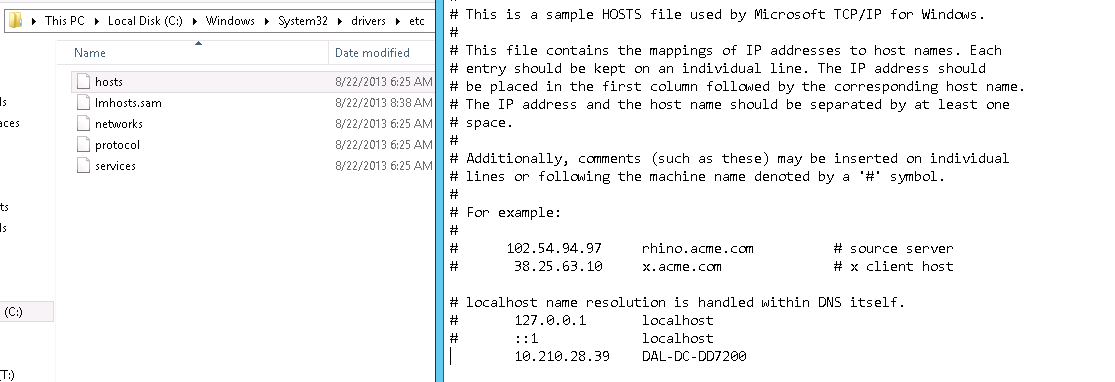
Installation is now complete!

# 

# Configuring DDBoost for MSSQLServer

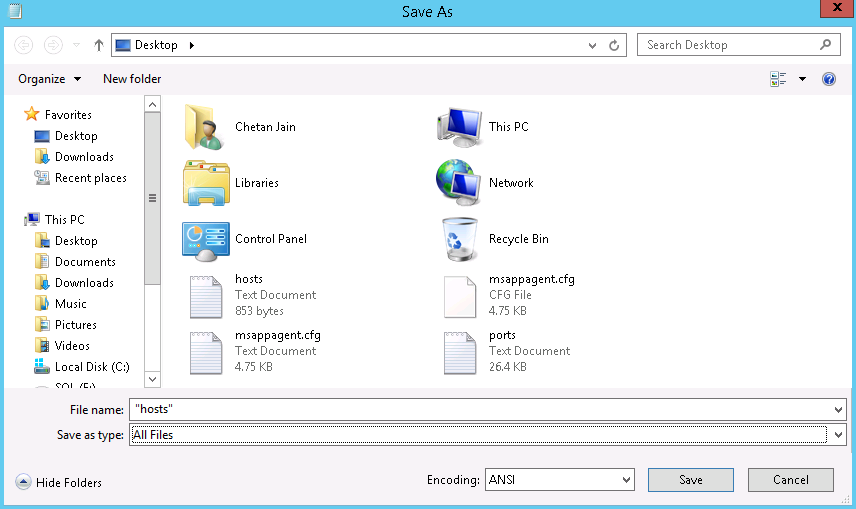
Open the **hosts** file using Notepad, located in C:\Windows\system32\drivers\etc

Make an entry as follows:

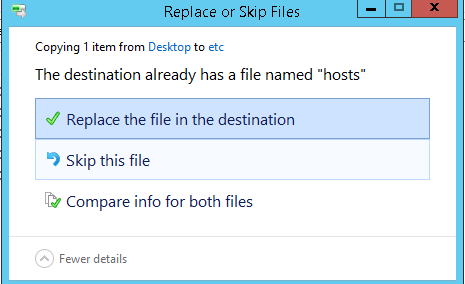


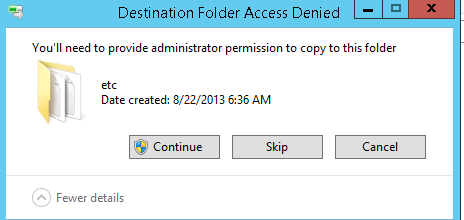
10.210.28.40 DALDataDomain

Use the “Save As” option to save the file to another location, example your Desktop. While doing save as ensure that you embed the file name in quotes and select “All Files” in the save as type. Refer screenshot below:



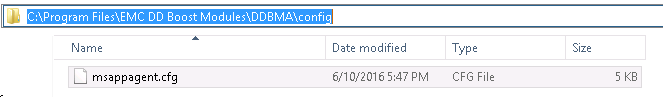
Go to Desktop Right click on the hosts file that you saved – Click Copy and Paste the file back into the C:\Windows\system32\drivers\etc folder. Respond to the windows prompts as follows:





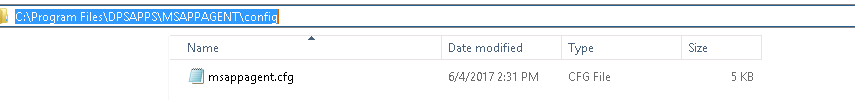
~~V 3.0~~

~~Go to C:\Program Files\EMC DD Boost Modules\DDBMA\config folder and locate msappagent.cfg file:~~

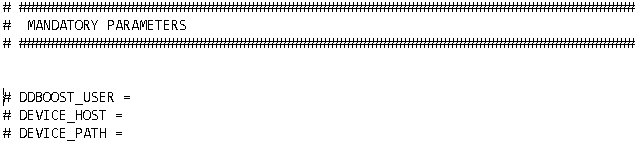


V 3.5

C:\Program Files\DPSAPPS\MSAPPAGENT\config



Open the file and locate the Mandatory parameters section



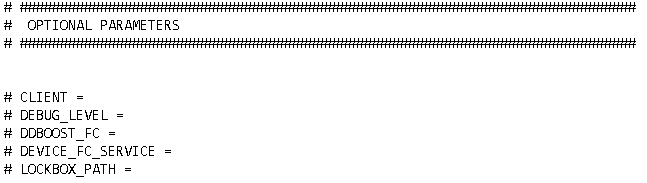
Edit the entries as shown below (do not forget to uncomment the changed parameters by removing the hash (#) from that line):

DDBOOST\_USER =ddboost

DEVICE\_HOST =DALDataDomain

DEVICE\_PATH =/DBbackups

Now, locate the optional parameters section in the same file



Edit only the below two entries as shown (do not forget to uncomment the changed parameters by removing the hash (#) from that line):

CLIENT = <FQDN of the server where you have installed the DDBoost agent)

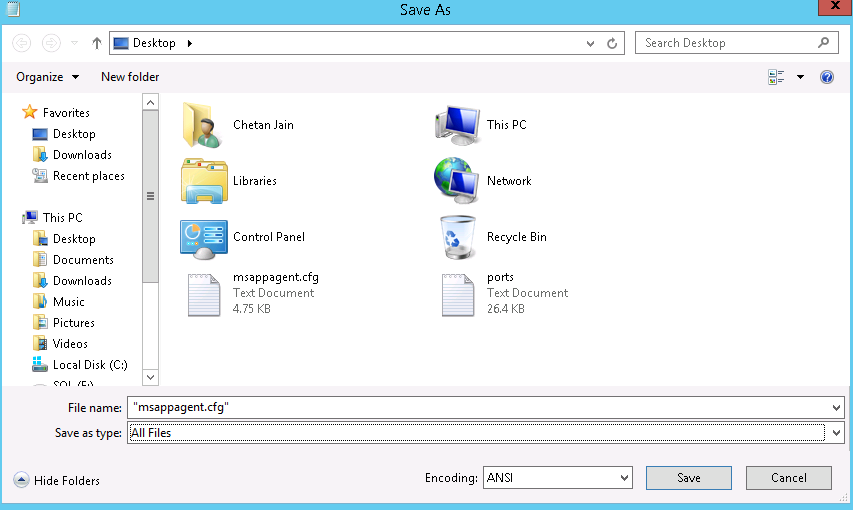
~~V 3.0~~

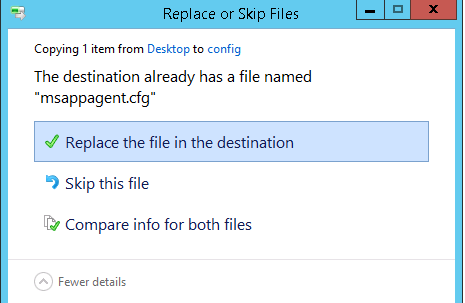
~~LOCKBOX\_PATH = C:\Program Files\EMC DD Boost Modules\DDBMA\config~~

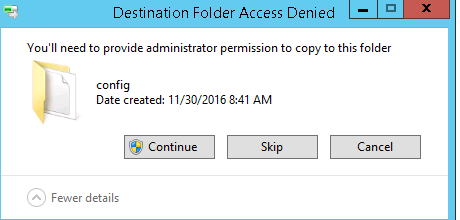
V 3.5

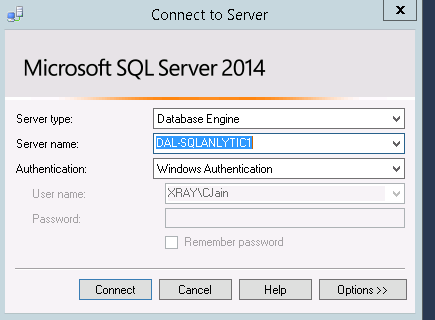
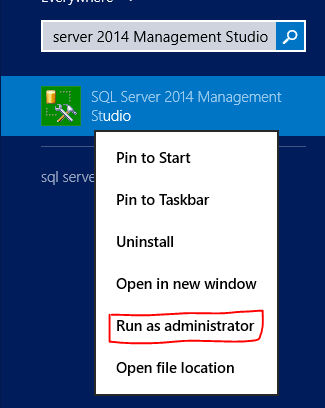
LOCKBOX\_PATH = “C:\Program Files\DPSAPPS\common\lockbox”

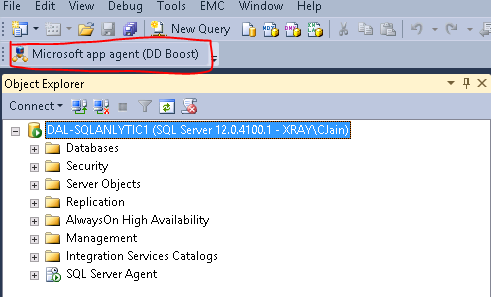
Save the file. In some cases, windows may not allow you to save the file in same location. In that case, Hit “Save As” option and save it to some other location. Copy the file from your saved location and paste it into the C:\Program Files\EMC DD Boost Modules\DDBMA\config folder. Choose to overwrite the existing file and allow administrator permissions

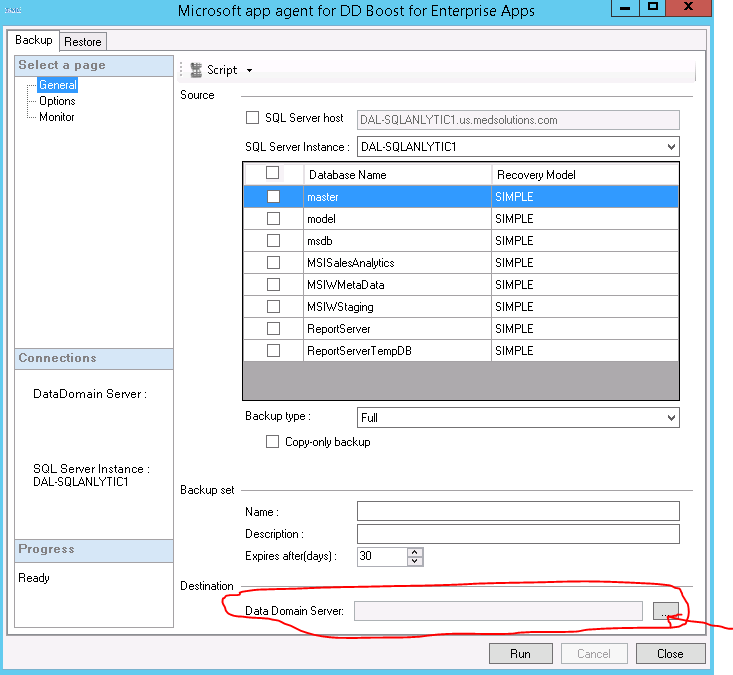
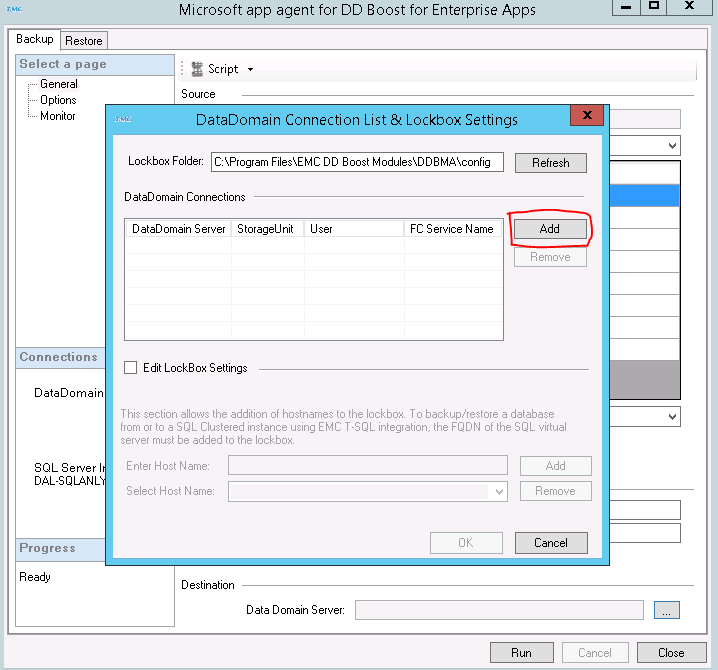


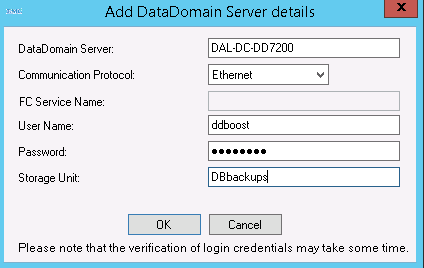










DataDomain Server: DALDataDomain

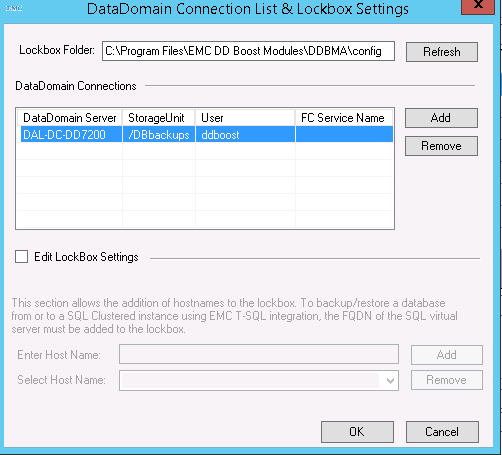
Communication Protocol: Ethernet

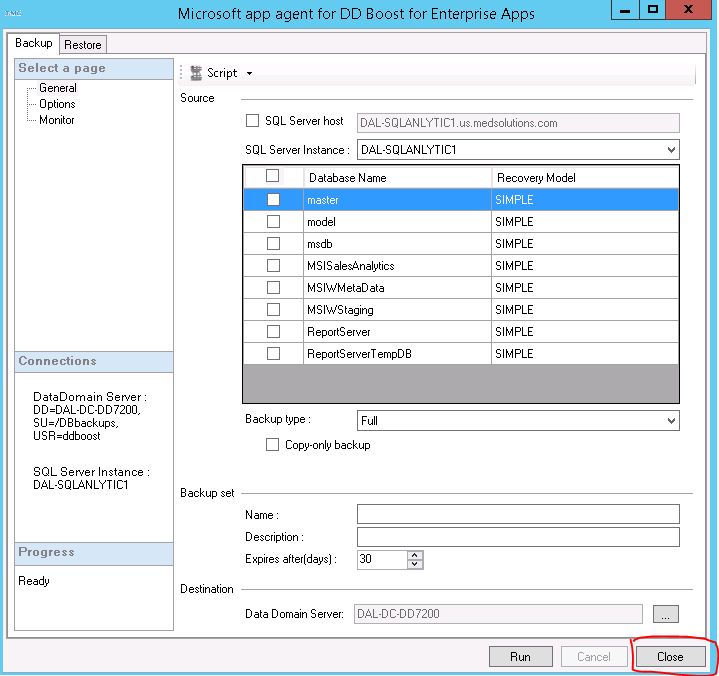
Username: ddboost

Pass: 3V!570r463

Storage unit: DBbackups

**These settings are case-sensitive so please make a note of upper and lower case used.**



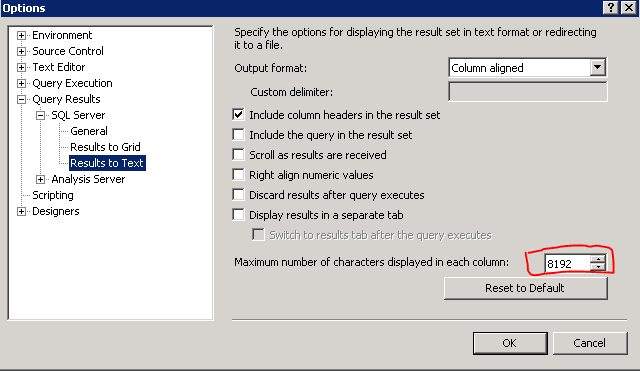


Configuration is now complete!

# Setting up SQL Server Agent Jobs:

* Now we are ready to setup SQL Agent jobs that will use DDBoost executable to run the backup directly to Data Domain server *DALDATADOMAIN* under a storage unit called as DBbackup.
* For most databases, a daily full backup should work i.e. we will try to get rid of the differential backups from our backup strategy since DDBoost is efficient enough to complete full backups in faster time and saving lot of space as well.
* We will have separate jobs one each for System and User database backups on each server. The script is designed to create either a System database or User database backup job.
* Each database will be added as one step into the system/user database backup job. If more flexibility is desired like some databases in our environment may have a backup schedule different than the others, then we can copy the steps into a separate job which can be created manually as needed.
* As of now, transaction log backup jobs are to be excluded from DDBoost backups. We will take some phased approach to make t-log backups for non-log shipped databases first. After that log shipped databases can be moved to DDBoost. Since log shipping is not a solution provided in DDBoost, we will need to come up with our own technique. This part will be done later.

1. Launch SSMS and set the max number of characters in each column to 8192 under Tools 🡪 Options:



Open New Query and Change the Result Mode to Text

1. Run the following script from SSMS on the instance that you wish to deploy backup jobs:

\\mstn-util-02\software\Microsoft\SQL Server\DDBoostForSQLServer\DDBoostBackupScriptsv3.5.sql

Execute this in a separate window

set nocount on;

exec DDBMA\_GenerateBackupJobScripts

@DBType = 1,

@DDHost = ‘DALDataDomain’,

@DDBoostUser = 'ddboost',

@DDStorageUnit = '/DBbackups',

@BackupLevel = 'Full',

@RetentionDays = 30,

@NumberOfStripes = 1

A script will be generated in the output window. Copy that script and execute it from a separate window

Repeat the command from step #2 with @DBType = 2

set nocount on;

exec DDBMA\_GenerateBackupJobScripts

@DBType = 2,

@DDHost = ‘DALDataDomain’,

@DDBoostUser = 'ddboost',

@DDStorageUnit = '/DBbackups',

@BackupLevel = 'Full',

@RetentionDays = 30,

@NumberOfStripes = 4

– for databases larger than 200 GB, its better to stripe them into 4 or higher number. This can be edited in the job steps after creating the job using stripes =1.

Again, a script will be generated in the output window. Copy that script and execute it from a separate window.

Transaction Log backups, for user databases in FULL recovery model:

set nocount on;

exec DDBMA\_GenerateBackupJobScripts

@DBType = 2,

@DDHost = ‘DALDataDomain’,

@DDBoostUser = 'ddboost',

@DDStorageUnit = '/DBbackups',

@BackupLevel = 'incr',

@RetentionDays = 30,

@NumberOfStripes = 1

Go to the SQL Agent from SSMS and expand the Jobs. There would be two new jobs created.

Please note that by default the backup jobs are scheduled at 12:00 AM via the script. However, they need not be kept default. In most cases, we should look at the existing backup jobs schedule on the server and change the schedules accordingly.

Disable the old backup jobs, if any

1. Schedule the cleanup job that will delete older backups

[\\mstn-util-02\software\Microsoft\SQL Server\DDBoostForSQLServer\CleanupExpiredBackupsets.sql](file:///\\mstn-util-02\software\Microsoft\SQL%20Server\DDBoostForSQLServer\CleanupExpiredBackupsets.sql)

**Additional tuning options:**

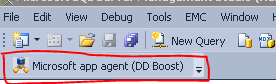
1. We can use the –S parameter in the ddbmsqlsv command to increase the number of stripes for the backup file. A database more than 200 GB can be considered for increasing number of stripes and we should increment the number of stripes in multiples of 4.
2. We can also specify multiple database names in one ddbmsqlsv command by specifying more than one database names <instancename>:<databasename> <instancename>:<databasename> <instancename>:<databasename>

Where instance name is only the instance part of the name. For default instance it will be MSSQL whereas for named instance it will be like SQL2012 (for example server name: DCB-DBA-SANDBOX\SQL2012).

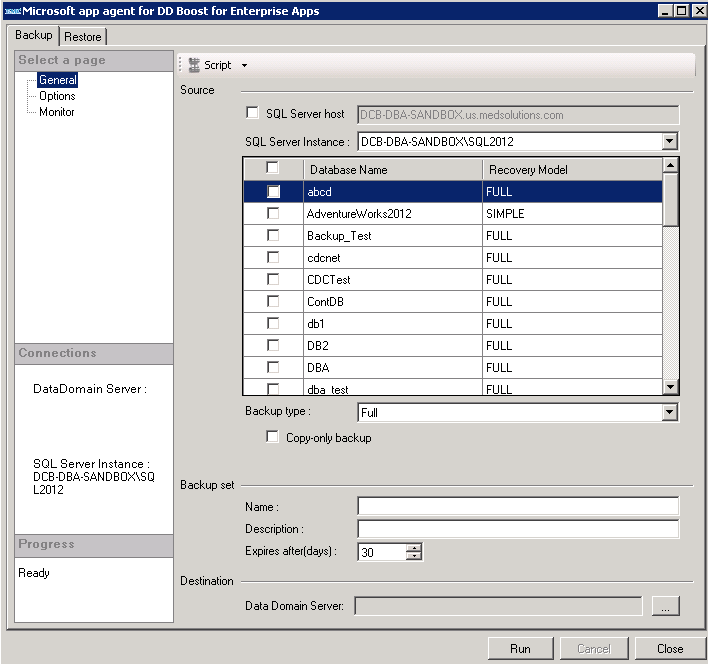
# How to perform database backup using DDBoost SSMS plug in:

From SSMS, connect to the SQL Server instance using Object Explorer.

You will see a Microsoft app agent (DD Boost) icon, click on that:



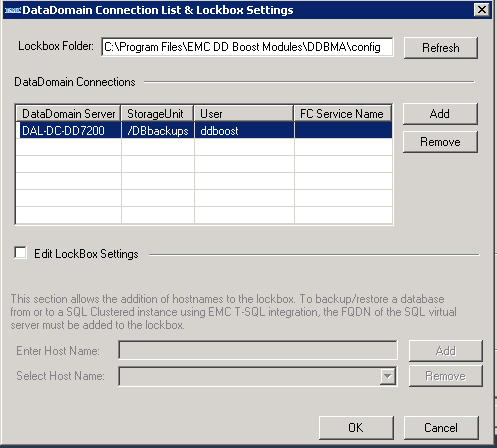
It will launch the below console:



Click on the below highlighted button:



Select the data domain server and click OK

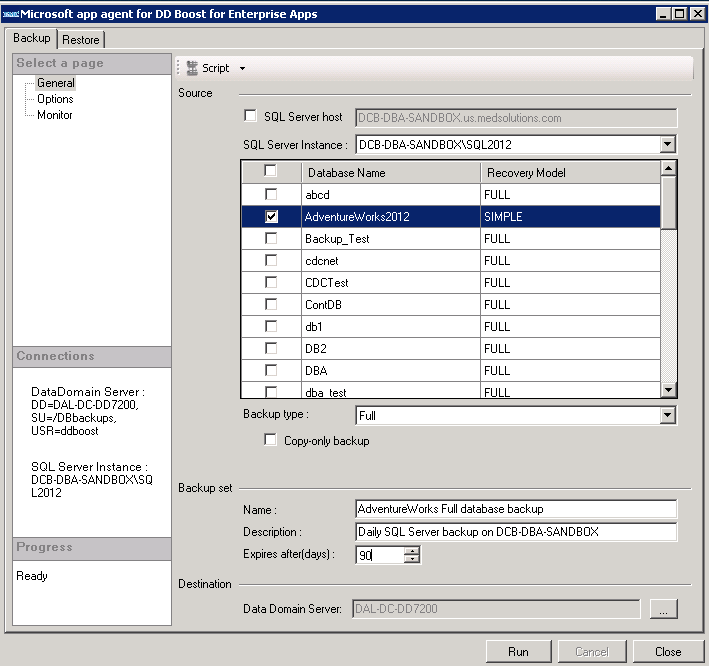


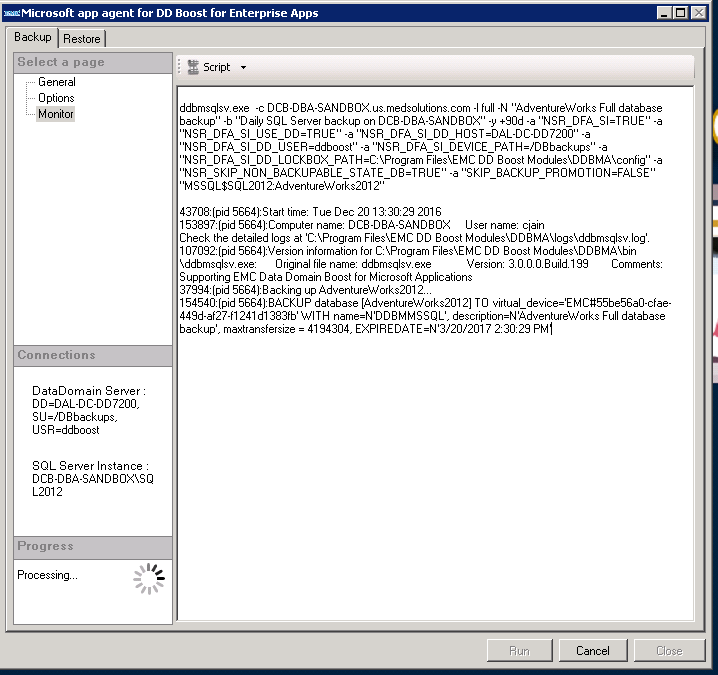
Select the host, instancename from the drop-down menu at top

Select database(s) that you wish to backup to data domain

Specify the details like Backup Type, whether you want copy\_only backup or not, name and description to identify the backupset, expiry date, etc.

If you wish to run the backup immediately, Click Run and the backup will start





Wait until backup is complete and a pop-up message appears. Hit OK on the pop-up window.

If you wish to run the backup at a later time, you can choose the Script option at the top of the console window. There are two options:

CLI script: it’s a command line script which can be invoked from cmd

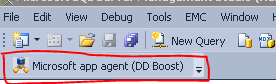
T-SQL script: It is a T-SQL command which can be run against the instance from where the backup needs to be taken.

**Note:** If we take a backup for any CR purpose, we can back up to a file share and name is appropriately so that the files do not get deleted until the purpose is over. While backing up to a data domain, specify appropriate retention period so that we do not happen to delete a file before its purpose is served.

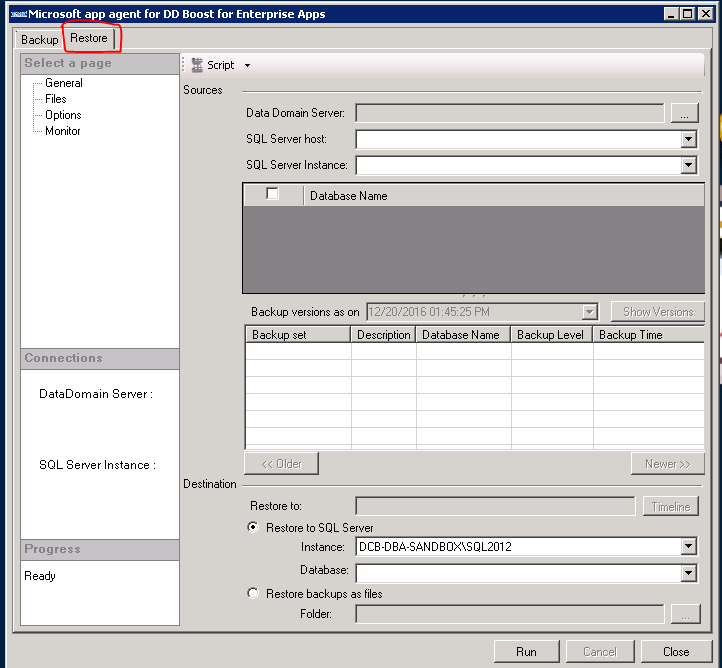
# How to perform Restore using DD Boost SSMS plug in:

From SSMS, connect to the SQL Server instance using Object Explorer.

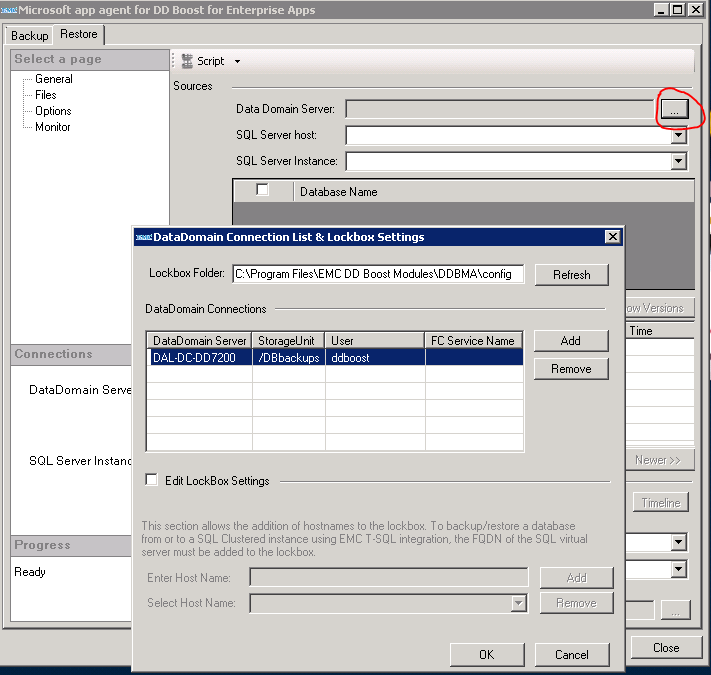
You will see a Microsoft app agent (DD Boost) icon, click on that:



It will launch the below console (make sure that you are on the second tab i.e. Restore:



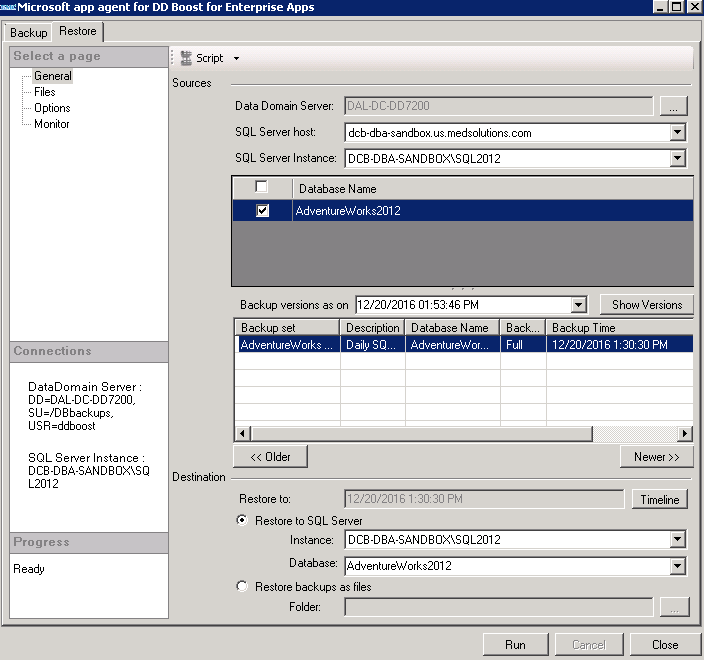
Click on the highlighted button to select the data domain server.



In the DataDomain Connection List & Lockbox Settings, select the datadomain server and click OK.

Sometimes, it may take some time at this step, so we just need to wait for a minute or two.

Select the appropriate input values like follows:

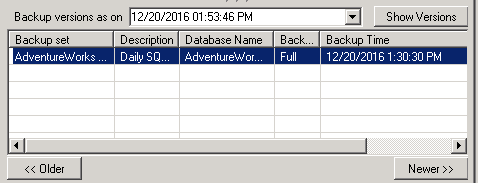


SQL Server is the host from where the backup was taken

Select the source SQL Server instance from where the backup was taken

Select the checkbox next to the database that you wish to restore.

The available backup set information will be populated in the table below



Select the backup. In this example, there is only one backup so only one record is visible.

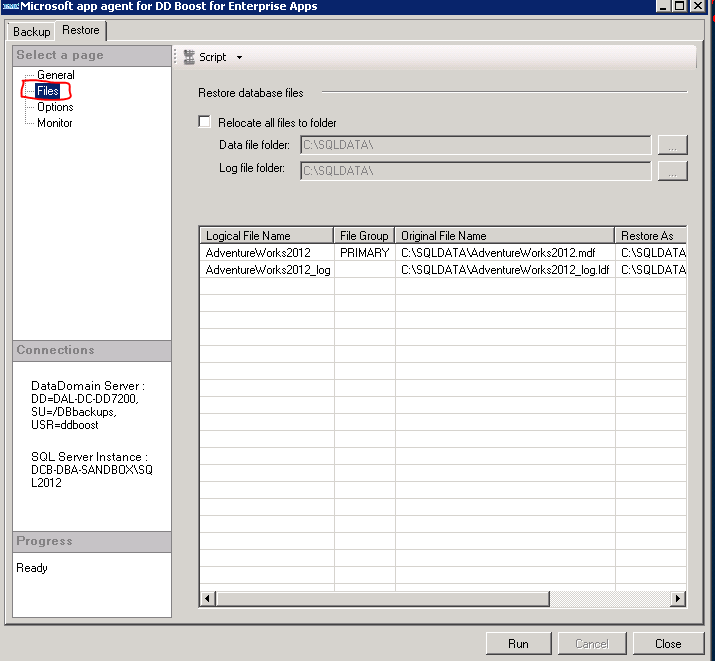
In the Destination section at the bottom, there are two options:

1. Restore to SQL Server: select Instance where we need to restore to

Select the database to restore to (specify a new name if you want to create a new database for restoration) else you can choose to overwrite an existing database.

1. Restore backups as files: This will restore the backup as a native format file in uncompressed state.

If you have selected option 1, you need to Files tab and specify appropriate data/log file locations.



Goto Options Tab and make appropriate selections like “restore with recovery” or “norecovery” or “standby”

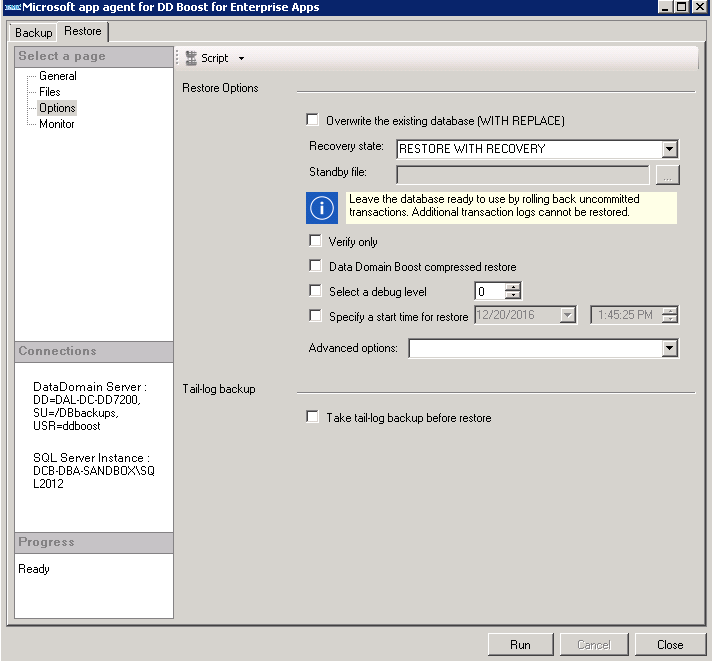
In the advanced options select keep replication or keep\_cdc (if applicable)

In most cases, we will keep Take tail log backup restore as unchecked.

Click Run or choose to generate a script which can be used for later. There are two different options like every other script:

T-SQL Script: T-SQL script can be run from SSMS or SQL Agent Job Step

CLI Script: Can be run from cmd or SQL Agent Job Step



If you have selected option #2, i.e. to restore the backup file to any location. After the restore is complete, use the backup file with native SSMS tools or T-SQL script to run a RESTORE DATABASE command.

# Maintenance of backup files in the data domain

Ddbmexptool can be used from command line to either view the historic backups or delete expired backups.

**Viewing the backups**

To view the backups that are within a specific save time range, run the following

command:

**ddbmexptool [-l] [-v -D 9 -b <start\_save\_time> -e**

**<end\_save\_time> -N <save\_set\_name>] {-z <config\_file> | -a**

**"parameter=value"} -n <application>**

Note: If you do not specify the -b and -e options, the command displays all the backups.

**Deleting the backups**

To delete the backups that are within a specific save time range and regardless of the

expiry date, run the following command:

**ddbmexptool -d [-v -D 9 -b <start\_save\_time> -e**

**<end\_save\_time>] {-z <config\_file> | -a "parameter=value"}**

**-n <application>**

Note: If you do not specify the -b and -e options, the command deletes all the

backups. To delete the backups that have been taken since yesterday, you must use

the interactive mode. When you use the non-interactive mode (-Y option), set the

value of the -e option to minimum 1 days ago to prevent the accidental deletion of the

active backups

**Deleting the expired backups**

To delete the expired backups that are within a specific save time range, run the

following command:

**ddbmexptool -k [-v -D 9 -b <start\_save\_time> -e**

**<end\_save\_time> -N <save\_set\_name>] {-z <config\_file> | -a**

**"parameter=value"} -n <application>**

Note: If you do not specify the -b and -e options, the command deletes all the expired

backups.

Note: The deletion commands do not check for the dependencies of the backups to

delete.

**Option Description**

**-d** Deletes the backups that are within a specific save time range.

Note: When you use the -d option with the -Y option, specify the value of the -e option as at least one day ago, which

does not delete the current backups.

**-k** Deletes the expired backups that are within a specific save time range.

**-v** Displays the verbose output on the console.

**-b** Specifies the lower boundary of the save time of the backup.

Specify the time in either the Hr (24 hour format):Min:Sec Month DD, YYYY format or the relative time format. The

supported relative time formats are now, <n> days ago, <n> months ago, <n> years ago, and -<n> days.

If you do not specify the -b option, the command considers 1/1/1970 as the lower boundary.

**-e** Specifies the upper boundary of the save time of the backup.

Specify the time in either the Hr (24 hour format):Min:Sec Month DD, YYYY format or the relative time format. The

supported relative time formats are now, <n> days ago, <n> months ago, <n> years ago, and -<n> days.

If you do not specify the -e option, the command considers the current time as the upper boundary.

The best practice is to set the value of -e to the same value as the expiration policy time period that was specified at the

backup time.

**-N** Specifies a filter on the save set name to display and delete.

-n Specifies the application, that is, SQL Server for this release.

Specify -n MSSQL.

**-a** Specifies a key-value pair, that is, -a "key = value" for options present in the .cfg (configuration) file.

For example, -a "DEVICE\_HOST = bu-ddb890.lss.emc.com"

**-z** Specifies the configuration filepath.

**-Y** Indicates your approval (yes) to delete the expired backups.

If you specify the -Y option, the message that asks for your approval to delete the expired backups does not appear.