



# MSSQL VSS Plugin

**Bacula Systems Documentation**

---

# Contents

<b>1 Overview</b>	<b>3</b>
<b>2 Bacula Windows VSS Plugin</b>	<b>3</b>
2.1 BMR and VSS . . . . .	4
<b>3 Backup</b>	<b>4</b>
<b>4 Restore</b>	<b>5</b>
4.1 Example . . . . .	8
4.2 Restoring master Databases . . . . .	10
<b>5 Plugin Notes</b>	<b>10</b>
5.1 Windows VSS Plugin Items to Note . . . . .	10
5.2 General Plugin Items to Note . . . . .	11
<b>6 Problem Resolution</b>	<b>11</b>

# Contents



- *Overview*
- *Bacula Windows VSS Plugin*
- *Backup*
- *Restore*
- *Plugin Notes*
- *Problem Resolution*

**Attention: The MSSQL VSS plugin is deprecated**

Support for the MSSQL VSS plugin will stop in September 2022. All instances of Microsoft SQL Server should use the mssqlvdi-plugin for future backups.

**Please do not run any differential backup with the MSSQL VSS plugin.**

In order to access the MSSQL VDI plugin, please contact our Support Team. The mssqlvss-to-mssqlvdi will guide you through the migration process.

# 1 Overview

This white paper presents how to use the Microsoft Windows VSS plugin's SQL Server support with **Bacula Enterprise** version 6.0 or newer. These solutions are not applicable to prior versions. This document is intended to be used by **Bacula Enterprise** administrators.

## 2 Bacula Windows VSS Plugin

**Bacula Systems** provides a single plugin for Bacula Enterprise named `vss-fd.dll` that permits you to backup a number of different components on Windows machines. One of those components is Microsoft SQL Server (MSSQL), which is the subject of this white paper.

Backing up and restoring MSSQL databases is supported with Full and Differential level backups. It is not possible to do Incremental backups because Microsoft does not support that backup level for the SQL Server product.

To activate the MSSQL component you have to put the following into the Include section of the File Set which will be used to back up the SQL Server data:

```
Plugin = "vss:/@MSSQL/"
```

This will back up all SQL server data except for those databases owned by Sharepoint, if that VSS plugin component is also specified.

The plugin directive must be specified exactly as shown above. A Job may have one or more of the vss plugin components specified.

You must ensure that the `vss-fd.dll` plugin is in the plugins directory on the FD doing the backup (done by default with the installer), and that the `Plugin Directory` directive line is present and enabled in the FD's configuration file `bacula-fd.conf`. An example configuration file is shown in figure *File Daemon Configuration Excerpt with "Plugin Directory" Line*. The status output of a client with the VSS plugin available is shown in figure *Status of a File Daemon with VSS Plugin Available*.

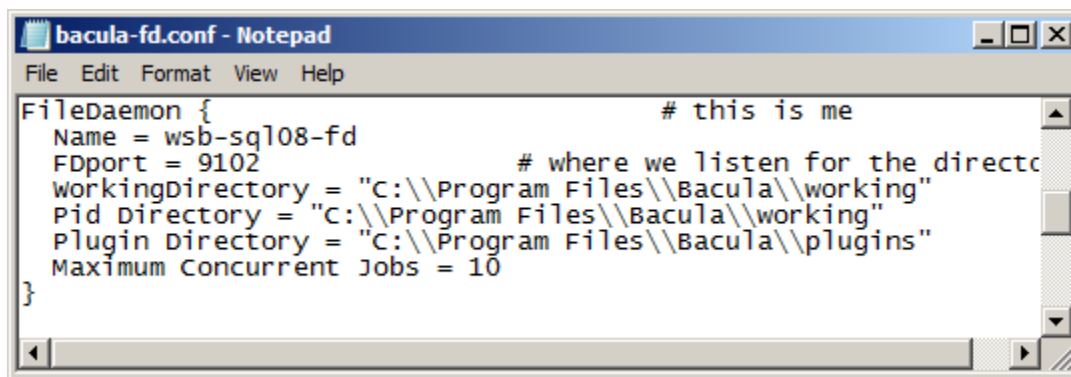


Fig. 1: File Daemon Configuration Excerpt with "Plugin Directory" Line

```

*status client=wsb-sql08-fd
Connecting to Client wsb-sql08-fd at wsb-sql08:9102

wsb-sql08-fd Version: 6.0.1 (02 Apr 2012) VSS Linux Cross-compile Win64
Daemon started 20-Apr-12 13:14. Jobs: run=15 running=0.
Microsoft Windows Server 2008 R2 Standard Edition Service Pack 1 (build ✓
-7601), 64-bit
Heap: heap=0 smbytes=1,061,455 ...
Sizes: boffset_t=8 size_t=8 debug=0 ...
Plugin: alldrives-fd.dll delta-fd.dll vss-fd.dll

```

Fig. 2: Status of a File Daemon with VSS Plugin Available

## 2.1 BMR and VSS

The VSS plugin will not work correctly during a Bare Metal Recovery because Microsoft does not include the VSS infrastructure in WinPE, the environment used during Bare Metal Recovery. As a consequence, any backup you do with the VSS plugin cannot be used for a Bare Metal Recovery. To have a good backup for BMR purposes, you must run the Bacula FD **without** using the `Plugin =` directive in your FileSet (i.e. the plugin must not be used).

## 3 Backup

If everything is set up correctly as above then the backup will include the SQL server data. The SQL server data files backed up will appear in a bconsole or bat restore like:

```

/@MSSQL/
...
etc

```

Only a backup of the complete SQL server data is supported, i. e. all database server instances, all databases and all tables are being backed up. It is not currently possible to back up only parts of the SQL server data like certain server instances, databases, or tables.

Both Full and Differential backups are supported. Microsoft does not support Incremental backups for MSSQL. If you run one, you will get errors.

Following the creation of a new database, you should run a Full backup of the SQL server data. A new database will not be backed up until a Full backup has been done, and Differential backups will fail on this specific new database.

As Windows does not update the time and date of modification of all SQL server files, you **must** use the **Accurate Job** option if you want correct Differential backups. If you do not use this option, the plugin will print a warning message, and restores will probably fail.

A complete example of the Job setup for MS SQL Server data looks as shown in figure *MS SQL Server Backup Job*. As for all VSS-enabled components, it is the administrator's responsibility to make sure that the required VSS snapshots are created by explicitly mentioning at least one file or directory for each drive where data that is handled by the plugin is stored. In the example, we use the file `c:/backmeup` to ensure this.

Note the inclusion of `c:/backmeup`, which is required to ensure that **Bacula** creates the required VSS snapshot of the drive the backed up data is stored on. If SQL Server data is also stored on other drives, you need to create similar File `=`-lines for these drives, too<sup>1</sup>.

<sup>1</sup> Starting with version 12.5, specifying the volumes is not mandatory anymore

```

File Set {
  Name = MSSQL
  Include {
    Options {
      Signature = SHA1
    }
    File = C:/backmeup
    Plugin = "vss:@MSSQL/"
  }
}

Job {
  Name = MSSQL08
  Accurate = Yes
  File Set = MSSQL
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Differential
}

```

Fig. 3: MS SQL Server Backup Job

```

File Set {
  Name = MSSQL-TestDB
  Include {
    Options {
      Signature = SHA1
    }
    File = C:/backmeup
    # backup only TestDB on the server
    Plugin = "vss:@MSSQL/ cinclude=*/TestDB cexclude=*"
  }
}

```

In this example, only the database TestDB will be included in the backup. Use of multiple `cinclude` parameters is possible in the Plugin command line.

## 4 Restore

To restore you can use all regular ways to start a restore. However, you must make sure that, if restoring differential data, the previous full backup is also restored. This happens automatically if you start the restore, in `bconsole`, using the restore options 5 or 12. In the file tree generated, you should either mark complete database instances or databases. Note that it is important **not** to include master databases in a restore as those have to be handled specially. We show an example of that below.

Additionally, a database must be restored to the MSSQL server from which it has been backed up; restoring to another machine is not possible.<sup>2</sup>

The **master** database cannot be restored while the MSSQL server is running. Thus you must take care not to select the master database for a regular restore. If you do, the restore will fail. Accordingly, we provide instructions how to

<sup>2</sup> The MS SQL plugin is not intended to be used to migrate data. To do that, you should use specialized tools. In the simplest case, a complete SQL data dump may be created, moved to the new machine, and fed to the database locally.

restore the **master** database of a MSSQL server instance, see chapter *Restoring master Databases*.

All other databases can be restored while the system is running. The Windows VSS writer will automatically unmount each database before it is restored, then apply all the outstanding log files if any exist, and finally remount the database.

It is important to understand that, using relocation of the restored files, data can be restored to a different and even a new database. This may be useful, but it may also be unintended. We recommend to make sure that any relocation is either turned off, or the effects are well understood. See figure *Relocated Database* for an example of how things may look when relocating to a default location.

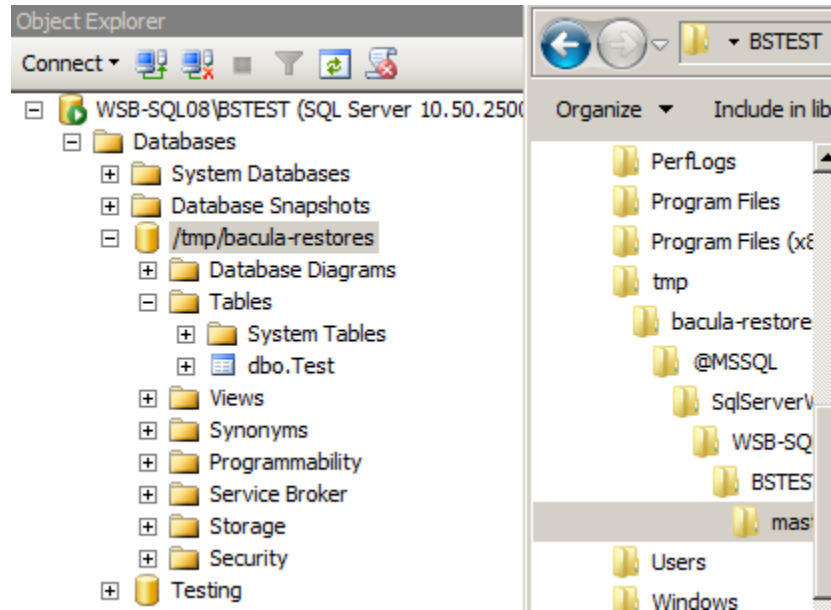


Fig. 4: Relocated Database

This result would probably not be what you expect – the result is a new database with a name that most likely does not match any reasonable naming scheme.

If you restore SQL Server data, it is important to understand that, during the restore process, **all** existing database log files in the database directory will be applied. For this reason, it can be important to remove old log files. This can only be done when the database is offline, so in this situation, the restore sequence may be like this:

- Take database offline. This can be done through the Microsoft SQL Server Management Studio by right-clicking on the database in the Explorer pane, clicking “Tasks”, then “Take Offline”. This can be seen in figure *Off-Line a Database in SQL Server Management Studio*.
- Delete, move or rename the database log files. An example is shown in figure *Renaming a Log File in Explorer*.
- Run the actual restore.
- Bring the database online again, similar to the procedure in step 1.

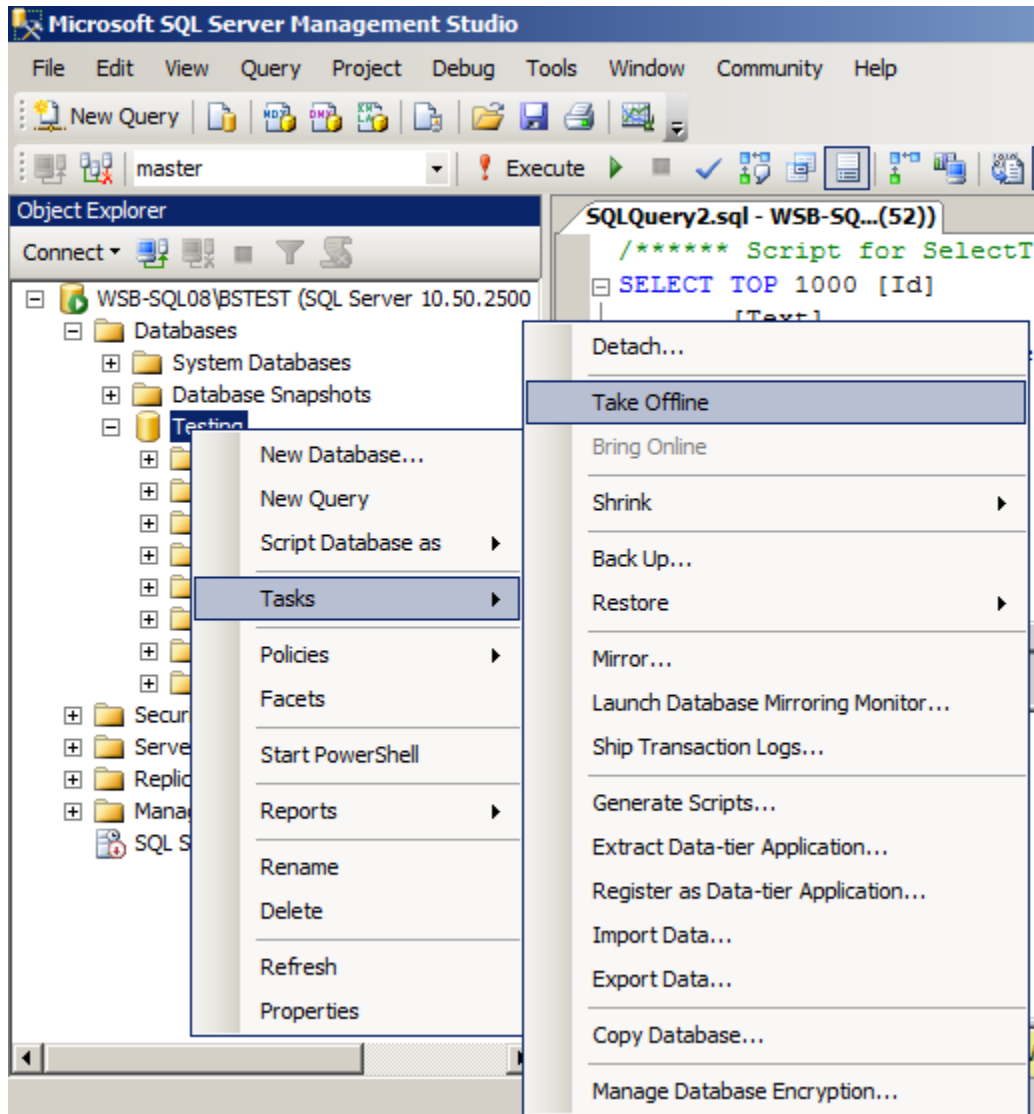


Fig. 5: Off-Line a Database in SQL Server Management Studio

QL Server ▾ MSSQL10\_50.BSTEST ▾ MSSQL ▾ DATA Search DATA

.. New folder

Name ^	Date modified	Type	Size
master	4/24/2012 10:35 AM	SQL Server Databa...	4,096 KB
mastlog	4/24/2012 10:35 AM	SQL Server Databa...	768 KB
model	4/24/2012 10:35 AM	SQL Server Databa...	1,280 KB
modellog	4/25/2012 12:41 PM	SQL Server Databa...	1,024 KB
MS_AgentSigningCertificate	4/19/2012 6:05 AM	Security Certificate	1 KB
MSDBData	4/24/2012 10:35 AM	SQL Server Databa...	15,104 KB
MSDBLog	4/24/2012 10:35 AM	SQL Server Databa...	5,184 KB
tempdb	4/24/2012 10:36 AM	SQL Server Databa...	8,192 KB
templog	4/24/2012 10:36 AM	SQL Server Databa...	512 KB
testing	4/30/2012 10:57 PM	SQL Server Databa...	2,304 KB
testing_log_old	4/30/2012 10:57 PM	SQL Server Databa...	504 KB

Fig. 6: Renaming a Log File in Explorer

## 4.1 Example

We assume that a correct backup of MSSQL data exists and you start the restore with option 5 in `bconsole`'s `restore` command, mark the complete tree of data backed up by the MSSQL component of the VSS plugin, then finally do `lsmark @MSSQL` to show all the files selected to be restored. Then the output you see should be similar to that presented in figure *Output for lsmark Command with SQL Server 2005 Data Marked*.

Note, the MSSQL files are generally under `@MSSQL/MSDEWriter` as in figure *Output for lsmark Command with SQL Server 2005 Data Marked* if you are running MSSQL 2005 on Windows Server 2003. If you are running MSSQL 2008 on Windows Server 2008, they will be under `@MSSQL/SqlServerWriter`.

Following that part of the data tree is the name of the MSSQL server host, in this case RUFUS-WIN2003, possibly the database server instance, and then the various databases.

In figure *Output for lsmark Command with SQL Server 2005 Data Marked* you can see that the name of the database master immediately follows RUFUS-WIN2003. The other databases, model, and msdb are at the same indentation level as master. So, for the restore to work, in the above example, you will need to unmark all the items that are associated with database master and below. If you have only selected database msdb for restoration, you would have output that looks like the one shown in figure *Marked Files to Restore Excluding master Database*.



```

$ mark @M*
14 files marked.
$ lsmark
*@MSSQL/
  *MSDEWriter/
    *RUFUS-WIN2003/
      *master/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *master.mdf
                    *mastlog.ldf
      *model/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *model.mdf
                    *modellog.ldf
      *msdb/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *msdbdata.mdf
                    *msdblog.ldf

```

Fig. 7: Output for lsmark Command with SQL Server 2005 Data Marked

```

$ lsmark
*@MSSQL/
  *MSDEWriter/
    *RUFUS-WIN2003/
      *msdb/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *msdbdata.mdf
                    *msdblog.ldf

```

Fig. 8: Marked Files to Restore Excluding master Database

## 4.2 Restoring master Databases

To restore the SQL Server **master** database, the only reliable way is to turn off plugins, restore the database files to some location, and copy the restored files to their original location.

To turn off the VSS plugin, use notepad to edit the **file daemon** (fd)'s configuration file, bacula-fd.conf, put a hash sign “#” in front of the plugin directory line, and re-start the **file daemon** (fd). A typical command sequence (in a command window with elevated privileges!) would look something like this:

```
net stop bacula-fd
notepad "C:\Program Files\Bacula\bacula-fd.conf"
net start bacula-fd
```

```
$ ls
master.mdf
mastlog.ldf
$ pwd
cmd is: /@MSSQL/SqlServerWriter/WSB-SQL08/BSTEST/master/c:/program files/microsoft
sql server/mssql10_50.bstest/mssql/data/
$ mark *
2 files marked.
```

This example shows SQL Server 2010 paths.

When restoring, you should navigate directly to the directory containing the data files, which will be represented in the virtual directory tree similarly to what we show in figure **fig:masterrestore**. In there, mark the data files (there should be two: master.mdf and mastlog.ldf), and continue the restore process. Using file relocation features to put the files into a new location is strongly recommended.

After the restore of the data files, you have to shut down the SQL server instance, move the restored data files to their correct location, and start the server instance again.

Afterwards, make sure to turn on plugins for the again.

Restoring the **master** database while MSSQL server is running is not possible.

## 5 Plugin Notes

### 5.1 Windows VSS Plugin Items to Note

- One file from each drive needed by the plugins must be explicitly listed in File Set used. This is to ensure that the main **Bacula** code does a snapshot of all the required drives. At a later time, we will find a way to accomplish this automatically.
- When doing a backup that is to be used for Bare Metal Recovery, do **not** use the VSS plugin. The reason is that during a Bare Metal Recovery, VSS is not available nor are the writers from the various components that are needed to do the restore. You might do a full backup to be used with a Bare Metal Recovery once a month or once a week, and all other days, do a backup using the VSS plugin, but under a different Job name. Then to restore your system, use the last Full non-VSS backup during the bare metal restoration of your system, and after rebooting do a restore with the VSS plugin to get everything fully up to date.

## 5.2 General Plugin Items to Note

- The 'estimate' command does not handle plugins. When estimating a job that uses plugins, an error message regarding the plugin will be displayed. However, backup jobs will use the plugin.
- The File Set Include Option `CheckFileChanges = Yes` does not work with plugin-generated data. Thus, you must not use that Option in the Include section of the FileSet where you specify using the MSSQL plugin.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

## 6 Problem Resolution

Most problems that can happen are reported in the job report **Bacula** sends. In the following table, we have collected information about common problems and the suggested resolution.

Job Report Message	Cause	Resolution
Unable to do a Differential backup of MSSQL master. Database excluded.		The master table is only backed up at full level. This is desing limitation of MS SQL Server. <sup>3</sup>
Warning: VSS Writer "SqlServerWriter" has invalid sttate. ERR=The writer vetoed the shadow copy creation creation process during the backup preparation state.	Various	Check for other messages in the Job Report
VSS Writer (PrepareForBackup): "SqlServerWriter", State: 0x7 (VSS_WS_FAILED_AT_PREPARE_BACKUP)		
Warning: VSS Writer "SqlServerWriter" has invalid sttate. ERR=The writer vetoed the shadow copy creation creation process during the backup preparation state.	Various	Check for other messages in the Job Report
SQL writer error: Backup type 2 not supported.	Incremental	Do not run incremental use Full and Differential levels only
Error: Unstable writer state=13: Restore skipped for file: /@MSSQL/SqlServerWriter/... /... /master/:component_info_... Writer="SqlServerWriter" ... Sqlldb error: OLEDB Error encountered calling ICommandText::Execute. hr = 0x80040e14.		
SQLSTATE: 42000, Native Error: 3013 Error state: 1, Severity: 16 ... Error message: RESTORE master WITH SNAPSHOT is not supported. To restore master from a snapshot backup, stop the service and copy the data and log file.		The master table can not be restored with SQL Server running normally. Follow instructions given above
Error: Unstable writer state=0: Restore skipped for file: /@MSSQL/SqlServerWriter/... /... /master/:component_info_... Writer="SqlServerWriter" Ssqlldb error: OLEDB Error encountered calling IDBInitialize::Initialize. hr = 0x80004005 SQLSTATE: 42000, Native Error: 18461 Error state: 1, Severity: 14 Source: Microsoft SQL Server Native Client 10.0 Error message: Login failed for user 'NT AUTHORITY\SYSTEM'. Reason: Server is in single user mode. Only one administrator can connect at this time.		
DBPROP_INIT_DATASOURCE: WSB-SQL08BSTEST DBPROP_INIT_CATALOG: master DBPROP_AUTH_INTEGRATED: SSPI		Follow instructions above to restore master table

<sup>3</sup> An explanation can be found at a [Microsoft Web site](#)