

BackupExec

BackupExec is Veritas's small business backup solution and depending on the version, allows **Volume** information to be extracted via:

1. Command line, or
2. Windows Power Shell, or
3. ODBC.

When **Volumes** are new, they will appear in the Scratch Media Set, but as they are used they are assigned to a specific Media Set.

Unlike Veritas's **NetBackup** product, when **Volumes** expire, by default, they are not moved back into the Scratch Media Set.

The Command Line Interface

The Command Line Interface was used in BackupExec 2010 and below.



When BackupExec is installed a number of predefined reports are created. The ID of each predefined report varies from one installation to the next.

Sample Command Line Syntax

```
bemcmd -o402 -r25 -ft:4 -f:"BE.csv" > "BE-CMD.txt" 2>&1
```

The Windows Powershell interface

The Windows Powershell interface replaced the Command Line Interface in BackupExec 2012.

Sample Powershell Script

```
cd "C:\Program Files\Symantec\Backup Exec\Modules\BEMCLI"
import-module BEMCLI

cd "C:\Program Files\TapeTrack\TapeTrack Sync\var"

#
# Get Media list from Backup Exec
#
$MediaList = Get-BEMedia
```

```
$Today = Get-Date

#
# Initialize the output array
#
$Records = @()

ForEach ($Media in $MediaList) {
    $Record = "" | Select-Object CartridgeLabel, MediaSetName, MediaVault,
RetentionHoursRemaining
    $Record.CartridgeLabel = $Media.Name
    $Record.MediaSetName = $Media.MediaSet
    $Record.MediaVault = $Media.MediaVault
    $Record.RetentionHoursRemaining =
[Int]($Media.OverwriteProtectedUntilDate - $Today).TotalHours
    $Records += $Record
}

$Records | Export-CSV -notype BE.csv
```

Synchronization



You will need to install the [TapeTrack Sync software](#) to complete these instructions.

Synchronization with TapeTrack is performed by calling the [TMSS10Sync](#) command line program, along with:

1. The CSV output file.
2. Command line arguments that instructs the program how to process volumes.
3. A synchronization definition file that instructs the program how to interpret the CSV output.

Example Command Line Arguments

Call Windows Powershell and run the BE-List script.

Call the TapeTrack Sync module and process the output created by the Powershell script.

```
powershell.exe -NoProfile -file "BE-List.ps1" -executionpolicy RemoteSigned
TMSS10Sync -S user:-password@server -a -d BE.ttidef < BE.csv
```

Where:

- -d is the path to the Synchronization Definition File.
- -a tells the program to add new tape volumes if they are encountered.
- -S tells the program what [Server](#) to connect to.
- BE.CSV is the output from the BE-List script.

Example Synchronization Definition

BE.ttidef

```
#  
# Set the Customer and Media as literal values as they never change  
#  
SetLiteral(CUSTOMER, "ACME");  
SetLiteral(MEDIA, "LT0");  
  
#  
# Set CSV delimiter  
#  
SetCSVDelimiter(",");  
  
#  
# Get the Volume-ID  
#  
Extract(VOLUME, 1, 10, 0);  
  
#  
# Get the Repository from a translated location Name  
#  
Extract(REPOSITORY, 0, 200, 0);  
AddTranslation(REPOSITORY, "*,WEEKLY_BACKUP*,[0-9]*", "OFFS");  
AddTranslation(REPOSITORY, "*", "LIBR");  
  
#  
# Set the Description to the Pool Name  
#  
Extract(DESCRIPTION, 2, 100, 0);  
RemoveSpaces(DESCRIPTION);  
#
```

[cookbook](#)

From:

<https://docs.tapetrack.com/> - **TapeTrack Documentation**

Permanent link:

<https://docs.tapetrack.com/cookbook/backupexec?rev=1626921995>

Last update: **2025/01/21 22:07**

