

# Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

With the release of Windows 2000, Microsoft overhauled the built-in backup utility NTBackup and added media-management and scheduling capabilities. Although these updates are welcome, comments I have read in Web support forums and newsgroups suggest that many administrators and other users have been frustrated in their attempts to get NTBackup to work in their environments.

At my company, we previously used third-party backup software on Windows NT 4.0 because NT 4.0's NTBackup tool wasn't robust enough for our needs. When we upgraded to Windows Server 2003, I took a second look at NTBackup to determine whether the overhauled version would be robust enough and whether it could be easily automated. During this process, I discovered several shortcomings in NTBackup:

- There's no simple way to write a backup to an arbitrary tape unless you use the /um (unmanaged) option on the command line. This option lets you overwrite an arbitrary tape. However, when NTBackup overwrites a tape in a backup job, it always applies a new media label—either a label you specify or a generic label based on the current date and time. There's no built-in way to tell NTBackup to keep the current label but overwrite the tape.
- There's no simple way to append backup information to an inserted tape because you must use either the /t (tape name) or /g (globally unique identifier—GUID) options on the command line. Unmanaged mode won't work because you can only append to a specifically designated tape.
- NTBackup can't eject a tape after a backup.
- NTBackup can't email or print completed job logs.

To overcome these shortcomings, I created a set of scripts, which Web Table 1 lists. The set includes a main script and 13 supporting scripts. The main script, Backup.cmd, uses NTBackup to perform a backup that overwrites the current tape but doesn't change the tape's media label. The 13 supporting scripts perform various functions for Backup.cmd. Together, these scripts provide a backup solution that you can easily customize and use—no script-writing experience is necessary to run them. You simply follow these steps:

1. Understand what each supporting script does so you can customize a solution.
2. Understand how the main script works so that you can customize it.
3. Prepare your environment and the scripts.

Script Name	Function	Dependencies*
Backup.cmd	Performs the backup with the help of the other scripts.	All except TapePrep.cmd
Eject.cmd	Ejects the tape.	PhysicalMedia.cmd, Refresh.cmd, ram.exe, and SetupVars.cmd
Library.cmd	Outputs the media library's name and/or GUID.	Findstr.exe, RSMView.cmd, and SetupVars.cmd
MailLog.cmd	Emails the most recent NTBackup log file.	Cmd.exe, blat.exe, SetupVars.cmd, and ShowLog.cmd
MediaGUID.cmd	Outputs the tape's logical-media GUID.	Partition.cmd, PhysicalMedia.cmd, RSMView.cmd, and SetupVars.cmd
MediaName.cmd	Outputs the media label for the tape.	PhysicalMedia.cmd, RSMView.cmd, and SetupVars.cmd
Partition.cmd	Outputs the partition's name and/or GUID.	PhysicalMedia.cmd, RSMView.cmd, and SetupVars.cmd
PhysicalMedia.cmd	Outputs the tape's name and/or GUID.	Library.cmd, RSMView.cmd, and SetupVars.cmd
PrintLog.cmd	Prints the most recent NTBackup log file.	Notepad.exe, SetupVars.cmd, and ShowLog.cmd
Refresh.cmd	Refreshes the media library.	SetupVars.cmd, Library.cmd, ram.exe, and sleep.exe
RSMView.cmd	Handles the Ram View command in the scripts.	Findstr.exe, ram.exe, and SetupVars.cmd
SetupVars.cmd	Configures environment variables for the other scripts.	None
ShowLog.cmd	Outputs the most recent NTBackup log or its filename.	SetupVars.cmd
TapePrep.cmd	Labels and allocates a tape for use by NTBackup.	Cmd.exe, ntbackup.exe, PhysicalMedia.cmd, Refresh.cmd, and SetupVars.cmd

\* Blat.exe is available from <http://www.blat.net>. Cmd.exe, findstr.exe, notepad.exe, ntbackup.exe, and rsm.exe are part of the Windows OS (%SystemRoot%\system32). Sleep.exe is available in sleep.zip, which is in 44990.zip. It's also in various Windows resource kits.

# Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

## Step 1. Understand the Supporting Scripts' Functions

Backup.cmd is the skeleton of this backup solution, whereas the 13 supporting scripts are its muscles. Here's how the 13 scripts flex their muscles and contribute to the end result:

### RSMView.cmd

NTBackup manages tapes and other media by using the Removable Storage service, which has a database in the %SystemRoot%\system32\NtmsData directory. You can use a command-line tool called RSM to manage the Removable Storage service. One of RSM's most useful commands is Rsm View, which lets you view the objects in the Removable Storage service's database.

The object types of interest here are the LIBRARY, PHYSICAL\_MEDIA, PARTITION, and LOGICAL\_MEDIA objects. (The object model for the Removable Storage database contains many other types of objects.) A LIBRARY object represents a physical device that uses removable media. The device we're interested in is the tape drive. One type of object the LIBRARY object can contain is the PHYSICAL\_MEDIA object, which represents the physical medium—in this case, a tape. The PHYSICAL\_MEDIA object, in turn, can contain a PARTITION object, which represents a partition. (The GUI uses the term side rather than partition, but the two terms refer to the same thing.) One object the PARTITION object can contain is the LOGICAL\_MEDIA object, which represents the allocated partition of a tape.

When you use the Rsm View command to display the objects in a container object (i.e. an object that contains other objects), you must specify the container object by its GUID. Determining the GUID is simply a matter of using one of RSM's options. When you use Rsm View with the /guiddisplay option, RSM outputs a list of objects and their associated GUIDs. With the GUID in hand, you can cut and paste the GUID into the Rsm View command. Cutting and pasting a couple of GUIDs isn't a problem, but having to cut and paste many of them can become quite tedious. Scripting is the perfect solution to eliminate this tedious task. So, the RSMView.cmd script handles the Rsm View command in the rest of the scripts. You don't need to cut and paste a single GUID.

### Library.cmd

Library.cmd obtains the tape drive's friendly name and GUID or just the tape drive's GUID. If you run Library.cmd by itself, it will display the tape drive's name and GUID. This information is helpful for troubleshooting if you have problems with the scripts. When Backup.cmd runs Library.cmd, Library.cmd retrieves only the GUID because Backup.cmd doesn't need the tape drive's name.

### PhysicalMedia.cmd

PhysicalMedia.cmd obtains the inserted tape's friendly name and physical-media GUID (i.e., the GUID for the physical tape) or just the tape's physical-media GUID. If you run PhysicalMedia.cmd by itself, it will display the tape's name and GUID. When Backup.cmd runs PhysicalMedia.cmd, PhysicalMedia.cmd retrieves only the tape's GUID.

### Partition.cmd

Partition.cmd obtains the partition's friendly name and GUID or just the partition's GUID. If you run Partition.cmd by itself, it will display the partition's name and GUID. When Backup.cmd runs Partition.cmd, Partition.cmd returns only the partition's GUID.

## Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

### MediaGUID.cmd

MediaGUID.cmd outputs the inserted tape's logical-media GUID (i.e., the GUID for the allocated partition of the tape), which is used with NTBackup's /g option. MediaGUID.cmd outputs the GUID in the format that NTBackup needs.

### MediaName.cmd

MediaName.cmd obtains the inserted tape's media label as it appears in the Name field on the Side tab of the tape's Properties dialog box in the Removable Storage console. Figure 1 shows a sample Properties dialog box. Backup.cmd uses the media label obtained by MediaName.cmd with NTBackup's /n option to reapply the existing media label when overwriting a tape.

Note that NTBackup uses the Info field when referring to tapes by name. Windows 2003 and Windows XP automatically set the Name field to the Info field. However, when I was testing Win2K, I discovered the Name field was left blank, and I had to copy the label in the Info field to the Name field. In any case, the Name field must match the Info field on the Side tab; otherwise, MediaName.cmd will fail or return incorrect results.

### Refresh.cmd

Depending on your hardware, RSM isn't always able to detect when tapes are inserted and removed. Thus, before running a backup job, it's important to perform a refresh operation to ensure that the database contains the tape drive's actual state. Refresh.cmd refreshes the tape drive. If the refresh operation is successful, Refresh.cmd uses sleep.exe to pause the script so that the database will return up-to-date information.

### Eject.cmd

Eject.cmd ejects the tape after the backup is complete. The script then calls Refresh.cmd to refresh the tape drive to ensure the Removable Storage service's database will return up-to-date information.

### ShowLog.cmd

ShowLog.cmd outputs the most recent NTBackup log file (i.e., the log you'd see in the NTBackup GUI) to the screen. If you run ShowLog.cmd with the /f option, it will output the log's full path and filename instead.

### PrintLog.cmd

PrintLog.cmd uses Notepad to print the most recent NTBackup log file. You need to configure a default printer before PrintLog.cmd will work. Summary logs (/L:s) are recommended if you use PrintLog.cmd.

### MailLog.cmd

MailLog.cmd uses blat.exe to email the most recent NTBackup log file to the specified person (e.g., the administrator). Blat.exe is a free utility that lets you send mail messages from a command script. SetupVars.cmd. SetupVars.cmd defines the settings of key environment variables used in the other scripts. This setup makes the scripts easily portable to different computers. You just need to modify the settings in one script (i.e., in SetupVars.cmd) rather than 13 scripts. (In Step 3, I discuss the settings you need to modify.)

### TapePrep.cmd

When you have one or more new tapes you want to label and allocate to NTBackup, you can use TapePrep.cmd. This script runs NTBackup in unmanaged mode so that NTBackup can use whatever tape happens to be inserted. The script labels the tape with the text you specify.

# Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

You run TapePrep.cmd at the command line and not in a script such as Backup.cmd because you need to use the script only once per tape. The command to launch TapePrep.cmd is:

```
TapePrep media_label
```

where media\_label is the text you want to use as the label. If the text contains spaces, you must enclose it in quotes.

## Step 2. Understand How the Main Script Works

As I mentioned previously, Backup.cmd runs a backup that overwrites the inserted tape, without changing the tape's media label. This script either directly or indirectly uses all the other scripts, except for TapePrep.cmd.

Listing 1 contains Backup.cmd. Callout A in Listing 1 shows the environment variables that direct its operation. To use this script, you need to customize these variables, which you'll do in Step 3.

### Listing 1: Backup.cmd

```
@Echo Off
Setlocal EnableExtensions EnableDelayedExpansion

Call SetupVars

Set BACKUP=C:\NTBackup
Set JOBNAME=NTBackup tools
Set SETDESC=NTBackup tools [%DTSTAMP%]
Set JOBINFO=NTBackup job '%JOBNAME%' on %COMPUTERNAME%
Set OPTIONS=/hc:on /l:s /m normal /r:yes /v:no
Set LOGFILE=%~dpn0.log
Set RC=0

Call Refresh > "%LOGFILE%"

Call PhysicalMedia > nul
If %ERRORLEVEL% NEQ 0 (Set RC=%ERR_PHYSICAL_MEDIA%
    Call :DIE "%JOBINFO% aborted: Media not detected (!RC!)"
    Goto :END)

Set MEDIANAME=
For /f "delims=" %n in ('MediaName /b') Do Set MEDIANAME=%n
If Not Defined MEDIANAME (Set RC=%ERR_PARTITION%
    Call :DIE "%JOBINFO% aborted: Unable to determine media name (!RC!)"
    Goto :END)
Echo Media name: %MEDIANAME% >> "%LOGFILE%"

Set MEDIAGUID=
For /f %g in ('MediaGUID /b') Do Set MEDIAGUID=%g
If Not Defined MEDIAGUID (Set RC=%ERR_LOGICAL_MEDIA%
    Call :DIE "%JOBINFO% aborted: Unable to determine media GUID (!RC!)"
    Goto :END)
Echo Media GUID: %MEDIAGUID% >> "%LOGFILE%"

Set COMMAND=ntbackup backup %BACKUP% /j "%JOBNAME%" /d "%SETDESC%"
```

## Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

```
/g %MEDIAGUID% /n "%MEDIANAME%" %OPTIONS%
Echo NTBackup command line: >> "%LOGFILE%"
Echo %COMMAND% >> "%LOGFILE%"
%COMMAND%
Set RC=%ERRORLEVEL%
Echo NTBackup exit code: %RC% >> "%LOGFILE%"

If /i "%EJECT%"=="YES" Call Eject >> "%LOGFILE%"
If /i "%MAILLOG%"=="YES" Call MailLog "%JOBINFO% [%DTSTAMP%] (%RC%)"
If /i "%PRINTLOG%"=="YES" Call PrintLog
Goto :END

:DIE
Setlocal
Set ERRORMSG=%~1
Echo %ERRORMSG% > "%NTBACKUP_DATA%\error.log"
Echo %ERRORMSG% >> "%LOGFILE%"
If /i "%MAILLOG%"=="YES" call MailLog "%ERRORMSG%"
If /i "%PRINTLOG%"=="YES" call PrintLog
Endlocal & Goto :EOF

:END
Endlocal & Exit /b %RC%
```

After defining the variables, Backup.cmd uses Refresh.cmd to refresh the tape drive. The script then performs three important operations. First, Backup.cmd uses PhysicalMedia.cmd to see whether a tape is inserted in the tape drive. Then, Backup.cmd uses MediaName.cmd to determine the tape's media label. Finally, Backup.cmd uses MediaGUID.cmd to determine the tape's logical-media GUID.

If all three operations succeed, Backup.cmd defines the COMMAND variable, which contains the NTBackup command, as callout B in Listing 1 shows. The script records the NTBackup command in the script's log file named backup.log, then executes the command. After NTBackup completes the backup, Backup.cmd writes NTBackup's exit code to backup.log. Finally, Backup.cmd checks the EJECT, MAILLOG, and PRINTLOG variables, as the code at callout C in Listing 1 shows. If any of these variables are set to YES, Backup.cmd calls the appropriate scripts.

If any of the three operations fail (i.e., the script can't detect the tape's physical-media GUID, media label, or logical-media GUID), Backup.cmd calls the :DIE subroutine, which callout D in Listing 1 shows. This code displays an error message, creates the error.log file in the NTBackup data directory, and writes the same error message to error.log and backup.log. Depending on the values set for the MAILLOG and PRINTLOG variables, the code might also email and/or print the most recent NTBackup log file, which contains the error message.

### Step 3. Prepare the Environment and Scripts

You can find the 14 scripts on the Windows IT Pro Web site. Go to <http://www.windowitpro.com>, enter InstantDoc ID 44990 in the InstantDoc ID text box, then click the 44990.zip hotlink. After you create a directory to hold the scripts (e.g., C:\NTBackup), extract them into that directory. When you schedule a job that uses the scripts, make sure you specify this directory as the Start in directory for the scheduled task.

The 44990.zip file includes the sleep.zip file, which contains sleep.exe. Unzip this file and extract its contents into the directory you just created. If you want to email the most recent NTBackup log file to

## Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

your administrator after the backup is complete, you need to obtain blat.exe. You can download blat.exe from <http://www.blat.net>. This utility requires no installation, registry entries, or support files. Just place it in the same directory as the scripts.

After the scripts and utilities are in place, you need to customize SetupVars.cmd and Backup.cmd:

Customizing SetupVars.cmd. Open SetupVars.cmd, which Listing 2 shows, in Notepad or another text editor. In the code at callout A in Listing 2, you need to define at least the RSM\_LIBRARY and RSM\_REFRESH variables. If you plan to use TapePrep.cmd, you must also define the RSM\_POOL variable.

### Listing 2: SetupVars.cmd

```
Set RSM_LIBRARY=
Set RSM_REFRESH=
Set RSM_POOL=

Set EJECT=NO
Set MAILLOG=NO
Set PRINTLOG=NO

Set SMTPSERVER=
Set SENDER=
Set RECIPIENTS=

Set DTSTAMP=%DATE:~0,3% %DATE:~10,4%-%DATE:~4,2%-%
DATE:~7,2% %TIME:~0,5%

Set ERR_NOT_FOUND=2
Set ERR_LIBRARY=6
Set ERR_PHYSICAL_MEDIA=7
Set ERR_PARTITION=8
Set ERR_LOGICAL_MEDIA=9

Set NTBACKUP_DATA=%USERPROFILE%\Local Settings\Application
Data\Microsoft\Windows NT\NTBackup\Data
```

RSM\_LIBRARY defines the friendly name for your tape drive. If you're unsure of this name, you can open a command-shell window and run the command:

#### RSMView library

to obtain the list of libraries on your system. Then simply copy the friendly name (but not the GUID) into SetupVars.cmd. You don't need to enclose the library's friendly name in quotes if it contains spaces. To check whether you have specified the library name correctly, type the following command at a command prompt:

#### Library

You should see the library's friendly name followed by its GUID. You must make sure that the library's friendly name doesn't contain any of the following reserved shell characters:

```
( ) < > ^ & |
```

## Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

If your library's friendly name contains any of these characters, you need to remove them from the library's name. To do so, first open the Removable Storage console by selecting Start, Run, `ntsmmgr.msc`, OK. In the left pane, expand Libraries (in Windows 2003 and XP) or Physical Locations (in Win2K). Right-click the tape drive in the left pane and choose Properties. Remove any of the offending characters from the Name text box, and click OK.

You need to set `RSM_REFRESH` to the number of seconds that `Refresh.cmd` should pause after performing a device refresh. This value depends on your hardware; set it to the number of seconds it takes to insert and remove a tape. A good starting point is 30 seconds.

If you plan to use `TapePrep.cmd`, set `RSM_POOL` to the media pool that matches the type of media used by your tape drive. For example, if your tape drive uses Travan tapes, specify the Travan media pool:

```
Set RSM_POOL=Travan
```

If the value contains spaces (e.g., 4mm DDS), you don't need to enclose it in quotes.

The code at callout B in Listing 2 defines the post-backup behavior. Setting the `EJECT` variable to YES prompts `Eject.cmd` to eject the tape. Setting `MAILLOG` to YES causes `MailLog.cmd` to email the NTBackup log file to the specified recipient. Setting `PRINTLOG` to YES prompts `PrintLog.cmd` to print the NTBackup log file on the default printer.

If you set `MAILLOG` to YES, you need to set the variables in the code at callout C in Listing 2. (If you set `MAILLOG` to NO, you don't need to do anything with this code.) Set the `SMTPSERVER` variable to the DNS name or IP address of an SMTP server on your network. (In this example, no authentication is used. However, `blat.exe` supports SMTP authentication. For information about how to incorporate authentication, see the `blat.exe` documentation.) Set `SENDER` to the email address you want to appear in the message's From field. Set `RECIPIENTS` to the email address of the person you want to receive the message. You can specify more than one recipient. Simply separate the addresses with a comma; no spaces are allowed.

You don't need to do anything with the remaining code, which defines the `DTSTAMP` variable, `ERR` constants, and `NTBACKUP_DATAPATH` variable. `Backup.cmd` uses `DTSTAMP` to create a date and time stamp. The `ERR` constants define the exit codes for the scripts. And the `NTBACKUP_DATAPATH` variable points to NTBackup's data folder, which stores the backup selection (`.bks`) files and log files.

Customizing `Backup.cmd`. Open `Backup.cmd` in Notepad or another text editor. In the code at callout A in Listing 1, set the `BACKUP` variable to the directory or directories you want to back up. Separate directory names with spaces. If a directory contains spaces, enclose it in quotes. Alternatively, you can specify a `.bks` file. Enter the full pathname to the file, prefix it with the `@` symbol, and enclose it in quotes if it contains spaces. You can use the `NTBACKUP_DATAPATH` variable as well. For example, if your filename is `Weekly full.bks`, you'd set the `BACKUP` variable to:

```
"@%NTBACKUP_DATAPATH%\Weekly full.bks"
```

Set the `JOBNAME` variable, which is used with NTBackup's `/j` option, to the backup job's name. This name will appear in NTBackup's Backup Reports window. Set `SETDESC`, which is used with NTBackup's `/d` option, to the backup job's description. This description will appear in the log file and in

## Want a Flexible Backup Solution?

Bill Stewart

(Reprinted From WindowItPro Magazine)

the Description column on the Restore tab in NTBackup's UI. The optional DTSTAMP variable that appears at the end of the description will let you see at a glance when the backup job ran.

You can customize the JOBINFO variable, but it's not required. This variable specifies the text used in error messages and in the email subject line. Note that it references the JOBNAME and COMPUTERNAME variables so you'll know at a glance the job and computer that generated the message.

You use the OPTIONS variable to define NTBackup's command-line options. For example, suppose you want NTBackup to append the backup to whatever tape happens to be inserted. You simply add the /a option to the OPTIONS variable. (You also need to remove /n "%MEDIANAME%" in the Set COMMAND= line because you can't use /n with /a.)

You can specify any of NTBackup's command-line options, except /j, /d, /n, /g, /f, /t, /p, and /um. As I mentioned previously, /j and /d are already defined in JOBNAME and SETDESC, respectively. The script automatically determines the tape's media label (/n) and GUID (/g). The other options (/f, /t, /p, and /um) shouldn't be used because they create conflicts.

The LOGFILE variable exists for troubleshooting purposes. Each script's output is redirected to this file, along with the tape's name and logical-media GUID, the NTBackup command line, and that command's exit code. If you leave the LOGFILE variable set to %~dpn0.log, the file will be created in the same directory as Backup.cmd and will have the filename Backup.log.

### Ready, Test, Go

After you've completed these three steps, the backup scripts are ready to test in a nonproduction environment. Even though I tested them with Windows 2003, XP, and Win2K, you should always test any new scripts in a controlled lab environment before deploying them. After you've successfully tested them, you'll have a flexible backup solution to use.