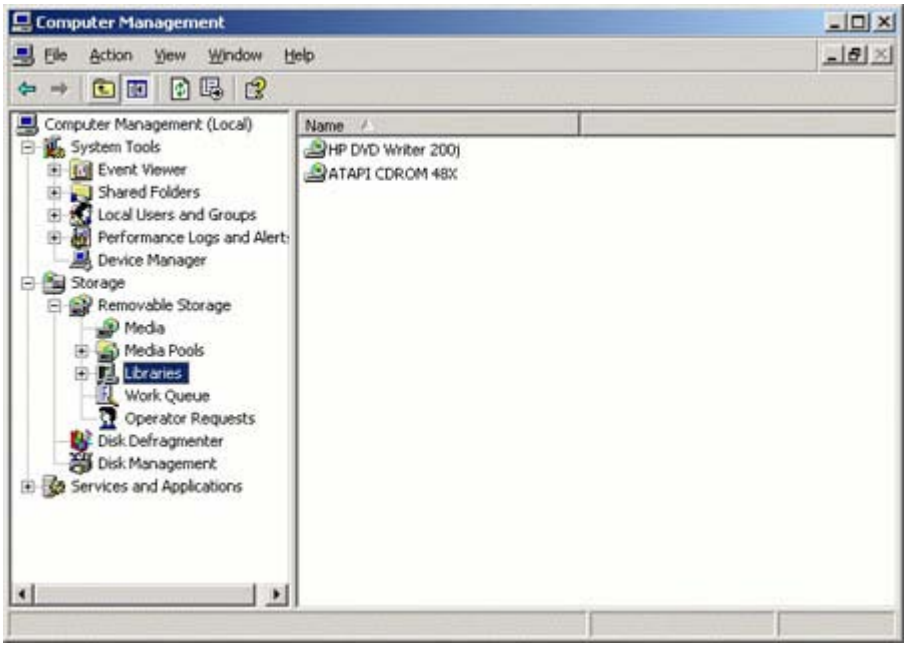
Automating Backups With NT Backup and the Removable Storage Manager

Brian Posey

The NTBACKUP utility may lack some of the features found in many third-party backup solutions -- for instance, its abilities to manage a library of tapes or to use a tape drive with an auto loader are, at best, limited. You can, however, create a script that leverages the power of the Windows Removable Storage Manager (RSM) interface in a way that gives you much more control over NTBACKUP.

But before you can create this script (part two of this article will show you how), you'll need a crash course in Windows Removable Storage Manager. Even though we won't be using RSM's GUI in our script, you'll need to understand what the RSM is all about before you can make use of it. To access the RSM, select the Computer Management command from Windows Administrative Tools menu. When the computer management console opens, you can find the Removable Storage console beneath the Storage container.

Basically, the RSM acts as a simple database that keeps track of backup-related resources. RSM is divided into five sections: Media, Media Pools, Libraries, Work Queue and Operator Requests (see Figure 1). This article, however, will look at only the first three: Media, Media Pools and Libraries. (The Work Queue and Operator Requests containers, which simply store instructions that are sent to the storage device, are irrelevant to what we will be doing.)

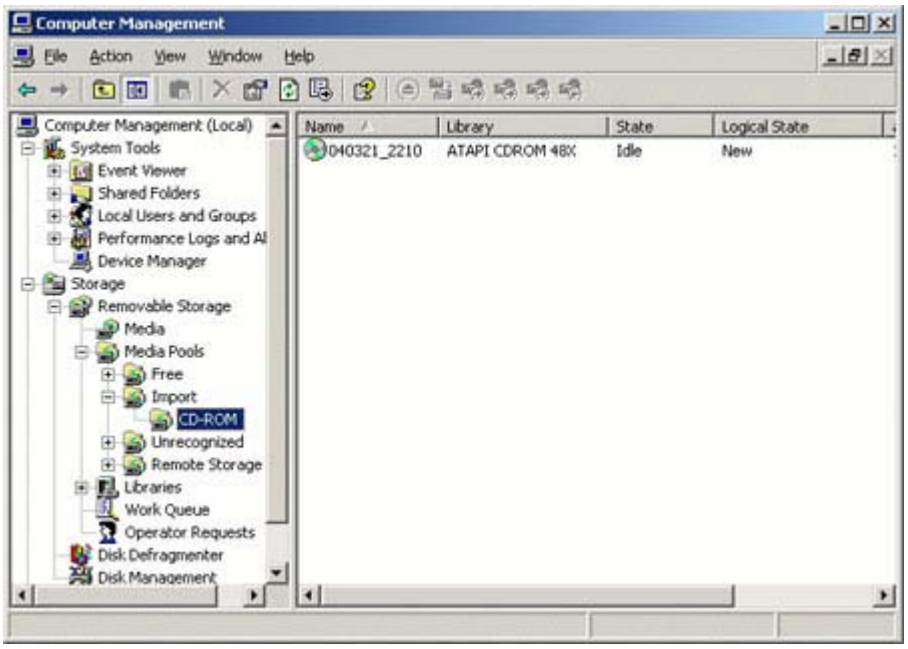


- **Libraries container:** RSM uses the word *library* to describe a physical, removable storage device, such as a tape drive or a CD-RW drive. In the RSM relational database, the Libraries container stores listings for removable storage devices.

- **Media container:** In the RSM, the word *media* describes anything that can be put into a removable storage device -- for example, tapes, CDs and DVDs. But if you open the media container, you'll see that it doesn't just contain generic representations of CDs and tapes; it also lists the specific media in the drive at that very moment. For instance, if you inserted the Windows Server 2003 installation CD into a CD drive, you would see that CD listed in the Media container.

- **Media Pools container:** This section has three containers by default -- Free, Import and Unrecognized -- but other containers can be added by various applications.

RSM is a database designed to keep track of backup-related resources. As you might expect, RSM keeps track of your backup tapes. When you insert a backup tape into the server's tape drive, the tape is assigned one of three categories (by default): Free, Import or Unrecognized. A listing for the tape is then displayed in the appropriate Media Pool. To be more precise, the Media Pool will list the device containing the tape and will display the tape just below that device. For example, my test machine does not have a tape drive, but I have inserted a CD into the machine, and RSM has placed the CD into the Import category. As Figure 2 shows, the CD is actually listed under Media Pools | Import | CD-ROM. The Media Pool lists the media on a per-device basis.



What do the various Media Pool categories mean? The Unrecognized Media container displays tapes that RSM considers to be unformatted. (Although most tapes you buy nowadays come preformatted, RSM will treat the tape as unformatted because RSM has not written any identifying information to the tape. When a tape is in the Unrecognized Media container, RSM can't do anything with the tape until the tape is prepared for use.)

Once RSM has prepared a tape for use, the tape is moved from the Unrecognized container to the Free container. The Free container is for blank tapes that have been prepared for use by RSM. Tapes in the Free container are ready to be used.

Let's say you insert a tape that contains data. Which container does RSM list the tape in? Actually, this is a trick question. Four paragraphs back I told you there were three containers by default, but other containers could be added by various applications. When you use an RSM-aware backup application for the first time, the application creates its own Media Pools container. For example, NTBACKUP creates its own Media Pool, as does Windows Remote Storage service. Third-party applications also create Media Pools, as long as the application is RSM-compliant.

Let's return to the question of where a tape containing data gets listed in the Media Pools. When you insert the tape, RSM analyzes its contents to see which application created the tape. If RSM recognizes the application, the tape is listed under that application's Media Pool. For example, if you

created a tape in NTBACKUP and then inserted that tape later on, RSM would see that the data on the tape was created by NTBACKUP and would associate the tape with the NTBACKUP Media Pool.

On the other hand, if RSM doesn't recognize the application that created the data on the tape, it will check to see if the tape has an RSM identification on it. Remember, RSM prepares a new tape for use, even if the tape is already formatted. If RSM finds an RSM identification on the tape, but does not recognize the application that wrote the data to the tape, it places the tape in the Import category -- a catch-all category.

In other words, the tape can't go into the Unrecognized or Free categories because it isn't blank. And RSM doesn't recognize the application that created the data on the tape, so the Import category is used as a sort of purgatory until an administrator installs an application that will recognizes the tape or performs an action that moves the tape into a different category.

One of the first techniques I want to show you is a method for determining a tape's Globally Unique Identifier (GUID). Part one of this article discussed the concept of preparing a tape for use.

When NTBACKUP prepares a tape, it assigns the tape a GUID. This is important because NTBACKUP has a nasty habit of changing a tape's name from one day to the next, but the tape's GUID will remain consistent over the life of the tape (assuming you don't reformat the tape).

I'll be up front with you and tell you that neither the tape's name nor the tape's GUID are any fun to incorporate into a script. By default, NTBACKUP assigns tapes names such as "Backup Set Created 12/13/2005 at 12:15 PM." In contrast, a GUID is a hexadecimal number consisting of eight characters, a dash, four characters, a dash, four more characters, a dash, another four characters, a dash, and then 12 characters. Basically, it looks like this:

```
b456c789-abcd-1234-a1b4-ab56cd89ef56.
```

Obviously, you don't want to have to type a GUID every time you want your script to perform an operation against a tape. The good news is that, in many cases, you don't have to. You can simply write a bit of code that will tell the tape drive to read the currently inserted tape and return its GUID. Before I show you how to determine a tape's GUID, I want to remind you that the title of this series of articles is "Automating complex backups." Like any complicated task, the process becomes much easier if you break it down into smaller, more manageable jobs. Reading a tape's GUID won't do anything for you by itself, but it is a key step in achieving our ultimate goal.

First Steps: Know How To Read A Map and Build a GUID Bridge

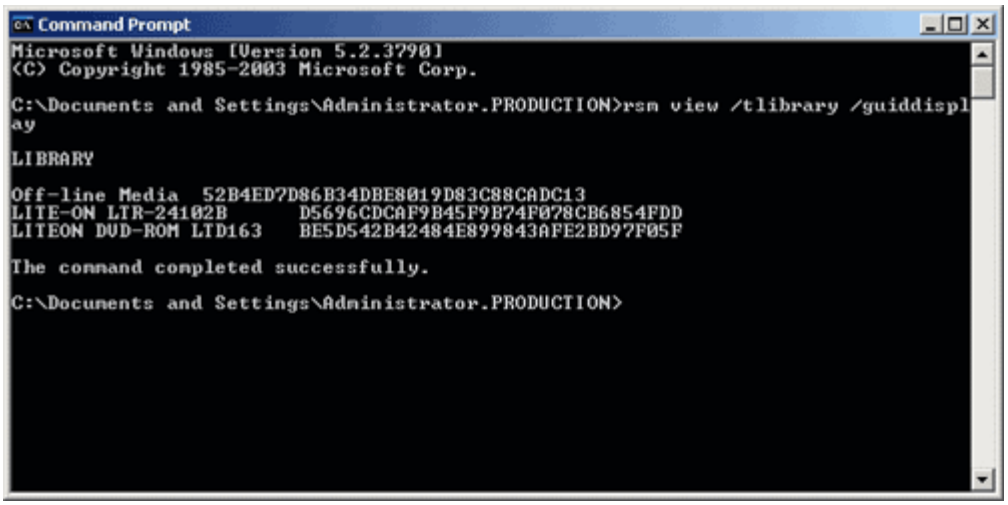
The first thing you need to know about the process for reading a tape's GUID is that you have to tell Windows which tape drive the tape is currently inserted into. Just as each tape has its own GUID, each tape drive attached to your system also has a unique GUID.

As you may have guessed, your script will have to reference each tape drive by its GUID. Unfortunately, there is no way to have your script dynamically read a tape drive's GUID every time the script runs. Even if it were possible to do that, it would be a bad idea because an automated script that dynamically reads a tape drive's GUID could potentially run an operation against the wrong drive. I recommend running a simple command to determine the tape drive's GUID and then hard-coding that GUID into the script that we are creating.

To determine a tape drive's GUID, open a Command Prompt window on the server to which the tape drive is attached and enter the following command:

```
rsm view /tlibrary /guiddisplay
```

This command gives you a way of interacting with the RSM via the command line. This will be important later on. If you notice the command's syntax, we are basically just telling the RSM to display the GUIDs associated with the tape library. When you enter the command, the output will look something like what you see in Figure 1, but your GUIDs will be different than mine.



Now that we have the GUID for the tape drive, let's get the GUID for the tape that's in the drive. You can do this in as little as two lines of code. Again, you will be leveraging the power of the RSM to accomplish this. The code itself looks like this:

```
Set drvguid=d5696cdcaf9b45f9b74f078cb6854fdd FOR /F "usebackq" %%i IN ('rsm
view /tphysical_media /cg%drvguid% /b /guiddisplay') DO set x=%%i
```

Now that I've shown you the commands, let's go over what they actually do. The first line simply assigns the tape drive's GUID, which I looked up earlier, to a variable named DRVGUID.

The second line is more complicated; it is basically a "for" command that sets the variable X to equal the result of the code that falls in between the two %%i markers. That code tells the RSM to display the GUID of the tape currently loaded into the tape drive that you have specified. Notice that the line makes use of the %drvguid% variable, which was defined in the previous line. You could actually specify the tape drive's GUID in place of the variable, but assigning that GUID to a variable keeps you from having to retype the GUID in other lines later on. (Remember, this is only the first step in the process). You may also have noticed that the /B switch is being used with the RSM command. This tells the RSM that you don't want to see any information other than the tape's GUID. When the line executes, the tape's GUID is assigned to the variable X.

So far, we have used the tape drive's GUID in conjunction with a few commands to retrieve a tape's GUID. You might be wondering how we can verify that the command worked. In the coming parts of this article, I will show you a lot more you can do with these commands, but I want to leave you with

one more line of code you can use to verify that the code you have already produced is working correctly. The line is:

```
rsm eject /pg%x% /astart
```

This command tells the RSM to eject the tape that is currently in the drive. The reason you can use this line of code to verify that everything is working correctly is because the code references the tape by its GUID (notice the %x% variable reference). If the tape ejects successfully, it means that the code you produced is working properly.

As you may recall, the first line in the script hard-codes the tape drive's GUID.

```
Set drvguid=d5696cdcaf9b45f9b74f078cb6854fdd
FOR /F "usebackq" %%i IN
('rsm view /tphysical_media /cg%drvguid% /b /guiddisplay') DO set x=%%i
```

It was enough to simply identify the tape's GUID because all we were doing was ejecting the tape. But if you're going to be writing data to the tape, just knowing the tape's GUID may not be enough. A physical tape formatted for use with NTBACKUP will usually contain one or more partitions, and a partition will contain one or more pieces of logical media. When you automate NTBACKUP so it can interact with the RSM, you must usually direct NTBACKUP to write data to a specific piece of logical media, not to the tape itself. Since your script will have to reference each piece of logical media by its GUID, you will have to write some code to derive the GUID prior to executing a backup.

The process of determining the GUID for logical media is similar to finding the GUID for physical media. Simply remember the hierarchical nature of tapes: Physical tapes contain partitions, which contain logical media. You already have the code to get the GUID for the physical tape; you can use that to get the GUID for a partition. You can then use the partition's GUID to get the GUID of the logical media.

To make the code work, you *must* understand the lines of code displayed above. If you don't understand what those lines of code do, you should review part two before continuing. With that said, here are a few more lines of code that you can use to derive a tape's partition GUID and its logical media GUID:

```
@echo off
Set drvguid= d5696cdcaf9b45f9b74f078cb6854fdd
FOR /F "usebackq delims==" %%x IN

('rsm view /tphysical_media /cg%drvguid% /guiddisplay /b') DO set
tapeguid=%%x
FOR /F "usebackq delims==" %%x IN

('rsm view /tpartition /cg%tapeguid% /guiddisplay /b') DO set partguid=%%x
FOR /F "usebackq delims==" %%x IN

('rsm view /tlogicalmedia /cg%partguid% /guiddisplay /b') DO set
logguid=%%x
```

Notice I have added the line *@echo off* to the beginning of this block of code. This line makes the output a bit cleaner and prevents the operating system from displaying redundant information as the code runs. The second line of code is the same line we were already using to hard-code our tape drive's GUID.

The three lines following *@echo off* actually derive the GUIDs. The third line in the script gets the physical tape's GUID and assigns it to the variable TAPEGUID. As part two explained, the %drvguid% portion of the command fills in the GUID of the tape drive. The /guiddisplay switch tells the command that you want the GUID of the specified object (in this case, the physical media, specified by the tphysical_media parameter). The /b switch tells the line not to display anything other than the GUID. (Of course, we are not actually displaying the GUID, but rather assigning it to a variable.)

The fourth line works in the same way, except this time we are asking for the partition GUID. (We used the /tpartition object instead of the tphysical_media object.) Likewise, this command required us to tell it which tape we want the partition off of, so we are referencing the %tapeguid% variable as a way of giving the command the GUID of the physical tape that we want partition information for. When the command executes, the partition's GUID will be assigned to the variable PARTGUID.

The last line of the script works the same way as the previous two lines except that this time we are getting the logical media GUID (notice the /tlogicalmedia switch) by referencing the partition GUID. When the command executes, the logical media's GUID will be assigned to the variable LOGGUID.

As I explained earlier, when you script a backup operation, NTBackup expects you to supply the logical media GUID for the tape. We now have the logical media GUID assigned to a variable.

Unfortunately, the logical media GUID is not in a usable format. As you may have noticed, the RSM works with GUIDs as a continuous 32-character hexadecimal string. The problem is that NTBackup can't accept a GUID in this format. NTBackup expects the GUID to be hyphenated with eight characters, a dash, four characters, a dash, four more characters, a dash, another four characters, a dash and 12 characters. Therefore, we need to parse the logical media GUID in a way that will insert hyphens into the correct positions. You can accomplish this by appending the following lines of code to the end of the script:

```
Set part1=%logguid:~0,8%
Set part2=%logguid:~8,4%
Set part3=%logguid:~12,4%
Set part4=%logguid:~16,4%
Set part5=%logguid:~20,12%
Set bkupguid=%part1%-%part2%-%part3%-%part4%-%part5%
```

The first five lines of code in this block break down the logical media GUID (which was assigned to the LOGGUID variable) into five smaller chunks. Notice that each of these lines ends with two numbers. The first number tells the operating system what position to start in; the second tells it how many characters to capture. The captured characters are then assigned to a variable (part1, part2, etc.).

For example, in the first line in this block of code, the numbers used are zero and eight. The zero indicates that the operation should start at the beginning of the string. The eight indicates that the first eight characters after the starting point should be captured. If you recall the format of a GUID, there are eight characters before the first hyphen. After the first hyphen, there are four characters before the

second hyphen. If you look at the second line of code in this block, you will see that it starts at position eight, captures four characters and assigns those characters to a variable named part2.

The last line in this block of code puts all of the pieces together. I have created a variable named BKUPGUID that is a combination of part1, a hyphen, part2, a hyphen, part3, a hyphen, part4, a hyphen and part5. This is the variable that we will be using to provide NTBackup with a GUID in part four of this series.

Much of this four-part series on automating complex backups has focused on how you can interact with the Removable Storage Manager (RSM) via either a command prompt or a script. Part one provided a crash course on RSM. Part two explained how to create a script to leverage the power of RSM. Part three showed you how to take a GUID that you extracted from the RSM and put it into a format that The NTBACKUP command in Windows Server 2003 can use. Here in part four, I will show what to do with that GUID now that it is in the proper format.

The trick to scripting backup operations is in realizing that NTBackup can be called from the command line and that there are a bunch of switches you can use with it. Since this series discusses the backup tape's GUID in so much detail, I might as well let the cat out of the bag and say that you can supply NTBACKUP with the tape's GUID by using the /G switch. The GUID itself must be surrounded by quotation marks and must be hyphenated in the manner discussed in part three. Furthermore, the /G switch cannot be used in conjunction with the /P switch, because if you use the /P switch, NTBackup will expect you to supply the name of a media pool.

Now let's look at the other switches that NTBACKUP will accept. (Keep in mind that not every switch listed here is required.)

- */J "job name"*: Tells NTBACKUP the name of the backup job.
- */T "tape name"*: Specifies the name of the tape that you want to use. But, as explained earlier in the series, if you are using a script, you will almost always want to use a GUID in place of a tape name.
- */N "tape name"*: Assigns a name to a new tape.
- */F "file name"*: Tells NTBACKUP to back up data to a file instead of to a tape.
- */D "description"*: Allows you to specify a backup description such as "full backup of server1 from 1-2-2006."
- */A*: Causes the backup operation to append the new backup to any data previously existing on the tape.
- */v:yes*: Tells NTBACKUP to verify the integrity of the backup.
- */R:yes*: Tells NTBACKUP that the backup tape should be accessible only to its owner and to members of the Administrators group.
- */L:f,s or n*: Tells NTBACKUP which logging level to use: (F)ull, (S)ummary or (N)one.

- */M backup type*: Specifies the backup type. Valid parameters for the /M switch include Normal, Copy, Incremental, Differential and Daily.
- */RS:yes*: Tells NTBACKUP that you want to back up the removable storage database.
- */hc:on*: Enables your tape drive's hardware compression.
- */SNAP:on*: Tells NTBACKUP to use Volume Shadow Copy should it need to backup an open file.

Note: If ON is listed as a parameter for a switch, you can achieve the opposite effect by specifying OFF instead. Likewise, if YES is used as a parameter for a switch, you can get the opposite effect by using NO.

There are three other parameters you need to know about: the Backup parameter, the BKS filename parameter and the SystemState parameter.

Specifying the *Backup parameter* after the NTBackup command places NTBackup in backup mode (as opposed to restore mode).

With the *BKS filename parameter,* prior to creating your script, you can open NTBackup and create a BKS file that says which files you want to include in the backup job. Suppose you wanted to back up the full contents of the C drive. You could open NTBACKUP, select the C drive and save the job as Backup.BKS. You could then call the job from within your script.

The *SystemState* parameter specifies that the computer's system state data should be backed up.

Now that I've gone through the syntax of the NTBACKUP command, let's discuss how you would use the command in a real-life situation. As mentioned earlier, most of the parameters described above are optional. Typical usage of the NTBACKUP command would look something like this:

```
NTBACKUP backup @c:\files\backup.bks /g "GUID" /a /v:yes /hc:on /m normal
```

After the NTBACKUP command is issued, the first thing I do is use the Backup parameter to put NTBACKUP into backup mode. Next, I specify the location of the Backup.BKS file, which tells NTBACKUP which files to back up. The next thing in the command is the /G "GUID" switch. In real life, you would replace the word GUID with the actual GUID. But since we are deriving *our* GUID from a script, you'll need to keep reading. The remaining switches tell NTBACKUP to append the job to whatever is on the backup media, verify the backup once it is completed, use hardware compression and perform a normal backup.

The script from part three of this article produced a variable called BKUPGUID, which contains the GUID that will be used by NTBACKUP. To make NTBACKUP use this GUID, simply replace the word GUID in the example above with the BKUPGUID variable. It would look something like this: /G "%bkupguid%" You can now complete the script by appending the NTBACKUP command to the end of the script as shown here:

```
@echo off
Set drvguid= d5696cdcaf9b45f9b74f078cb6854fdd
FOR /F "usebackq delims==" %%x IN ('rsm view /tphysical_media /cg%drvguid%
/guiddisplay /b') DO set tapeguid=%%x
```

```
FOR /F "usebackq delims==" %%x IN ('rsm view /tpartition /cg%tapeguid%
/guiddisplay /b') DO set partguid=%%x
FOR /F "usebackq delims==" %%x IN ('rsm view /tlogical_media /cg%partguid%
/guiddisplay /b') DO set logguid=%%x
Set part1=%logguid:~0,8%
Set part2=%logguid:~8,4%
Set part3=%logguid:~12,4%
Set part4=%logguid:~16,4%
Set part5=%logguid:~20,12%
Set bkupguid=%part1%-%part2%-%part3%-%part4%-%part5%
NTBACKUP backup @c:\files\backup.bks /g "%bkupguid%" /a /v:yes /hc:on /m
normal
```