mcpmag.com

Managing Restore Points with PowerShell -- Microsoft Certified Professional Magazine Online

By Jeffery Hicks02/21/2012

5-6 minutes

Prof. Powershell

Managing Restore Points with PowerShell

Windows System Restore can be managed through several PowerShell cmdlets. We cover some of them here.

For a number of years Microsoft Windows has had a nice feature for recovering from something "bad" using system restore points.

These system snapshots can be configured to happen automatically. Restore points are often also created when new updates are installed. You may have thought you always needed the GUI to manage restore points, but you can manage them with Powershell on Windows 7.

The first thing you might need to do is enable system restore:

```
PS C:\> Enable-ComputerRestore -drive "c:\"
```

If by chance you want to turn it off, then use the Disable-ComputerRestore cmdlet. To view existing restore points, use the Get-ComputerRestorePoint cmdlet:

```
PS C:\> Get-ComputerRestorePoint
PS C:\>
```

I don't have anything yet so I guess I'll create one using the Checkpoint-Computer cmdlet:

```
PS C:\> Checkpoint-Computer -description "My first checkpoint"
-restorepointtype "Modify_Settings"
```

The default restore point type is APPLICATION_INSTALL, but you

can also use: APPLICATION_UNINSTALL, DEVICE_DRIVER_ INSTALL, MODIFY_SETTINGS, and CANCELLED_OPERATION. Here's my result.

```
PS C:\> Get-ComputerRestorePoint

CreationTime        Description      SequenceNumber EventType
      RestorePointType
------------        -----------      -------------- ---------        ----------------
2/2/2012 12:36:17 PM My first check...  58
       BEGIN_SYSTEM_C... MODIFY...
```

The other critical point to know when using Checkpoint-Computer cmdlet is that you can only create a restore point with this cmdlet once every 24 hours. You can run the command again, but it will only keep the last restore point:

```
PS C:\> Checkpoint-Computer -Description "My 2nd checkpoint"
-RestorePointType "Modify_Settings"
PS C:\> Get-ComputerRestorePoint | format-list

__GENUS          : 2
```

```
__CLASS          : SystemRestore
__SUPERCLASS     :
__DYNASTY        : SystemRestore
__RELPATH        : SystemRestore.SequenceNumber=59
__PROPERTY_COUNT : 5
__DERIVATION     : {}
__SERVER         : CLIENT2
__NAMESPACE      : root\default
__PATH           : \\CLIENT2
\root\default:SystemRestore.SequenceNumber=59
CreationTime     : 20120202180537.316029-000
Description      : My 2nd checkpoint
EventType        : 100
RestorePointType : 12
SequenceNumber   : 59
```

I wish the cmdlet supported -WhatIf, but it doesn't so be careful.

Fortunately when it comes time to restore, -Whatif is supported:

```
PS C:\> Restore-Computer -RestorePoint 59 -whatif
```

What if: Performing operation "Restore-Computer" on Target "CLIENT2".

I'll go ahead and do the restore, which will automatically reboot the computer:

```
PS C:\> Restore-Computer -RestorePoint 59
```

When the computer returns, I can check the status of the last operation:

```
PS C:\> Get-ComputerRestorePoint -LastStatus
```

The computer has been restored to the specified restore point.

If you see the message, "The last attempt to restore the computer failed" this may not mean a real failure. You can get this response if you've never performed a system restore.

These are definitely commands you want to test with in a non-production environment. A virtual environment would be even better so you can use snapshots to rollback. I also encourage you to read full help and examples for all of these cmdlets.

About the Author

Jeffery Hicks is an IT veteran with over 25 years of experience, much of it spent as an IT infrastructure consultant specializing in Microsoft server technologies with an emphasis in automation and efficiency. He is a multi-year recipient of the Microsoft MVP Award in Windows PowerShell. He works today as an independent author, trainer and consultant. Jeff has written for numerous online sites and print publications, is a contributing editor at Petri.com, and a frequent speaker at technology conferences and user groups.