Manipulating items directly

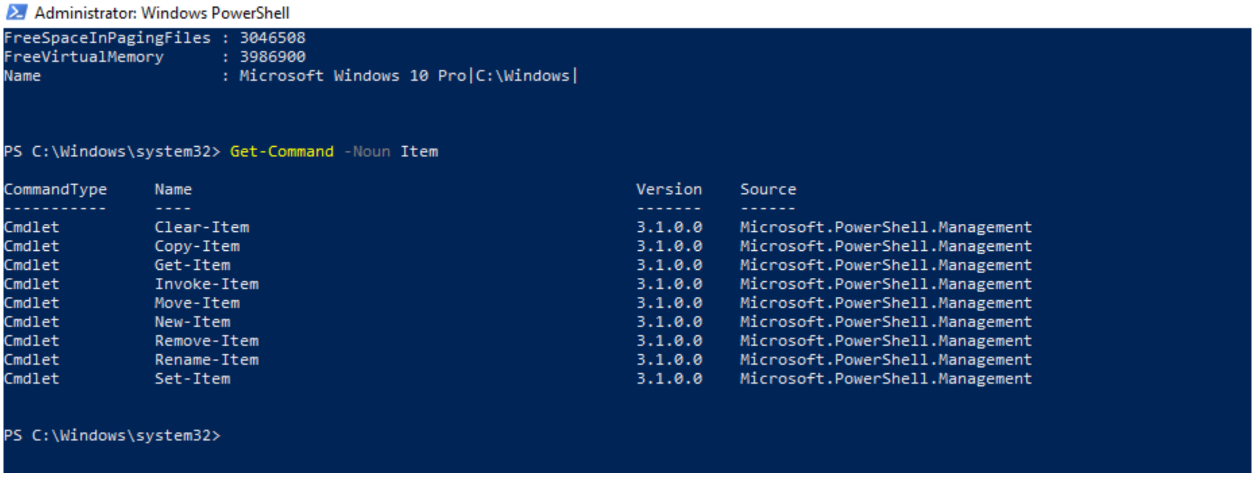
The elements that you see in Windows PowerShell drives, such as the files and folders in the file system drives, and the registry keys in the Windows PowerShell registry drives, are called **items** in Windows PowerShell. The cmdlets for working with them item have the noun **Item** in their names.

The output of the `Get-Command -Noun Item` command shows that there are nine PowerShell item cmdlets.

PowerShellCopy

```
Get-Command -Noun Item
```

Output :



```
FreeSpaceInPagingFiles : 3046508
FreeVirtualMemory      : 3986900
Name                   : Microsoft Windows 10 Pro|C:\Windows|


PS C:\Windows\system32> Get-Command -Noun Item

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          Clear-Item                                         3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Copy-Item                                          3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Get-Item                                           3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Invoke-Item                                        3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Move-Item                                          3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          New-Item                                           3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Remove-Item                                        3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Rename-Item                                        3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Set-Item                                           3.1.0.0    Microsoft.PowerShell.Management


PS C:\Windows\system32>
```

Creating new Items (New-Item)

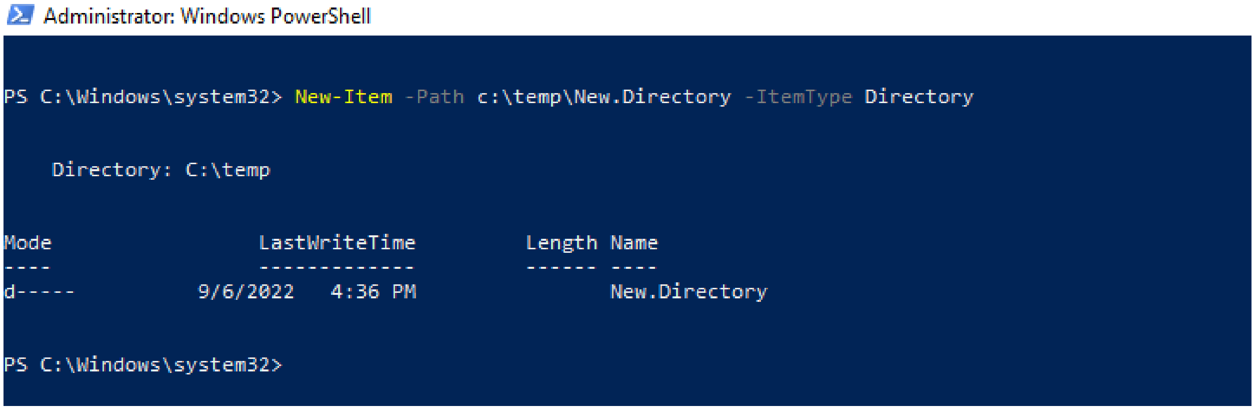
To create a new item in the file system, use the `New-Item` cmdlet. Include the **Path** parameter with path to the item, and the **ItemType** parameter with a value of `file` or `directory`.

For example, to create a new directory named `New.Directory` in the `C:\Temp` directory, type:

PowerShellCopy

```
New-Item -Path c:\temp\New.Directory -ItemType Directory
```

Output :



```
PS C:\Windows\system32> New-Item -Path c:\temp\New.Directory -ItemType Directory


    Directory: C:\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          9/6/2022   4:36 PM                New.Directory


PS C:\Windows\system32>
```

To create a file, change the value of the **ItemType** parameter to `file`. For example, to create a file named `file1.txt` in the `New.Directory` directory, type:

PowerShellCopy

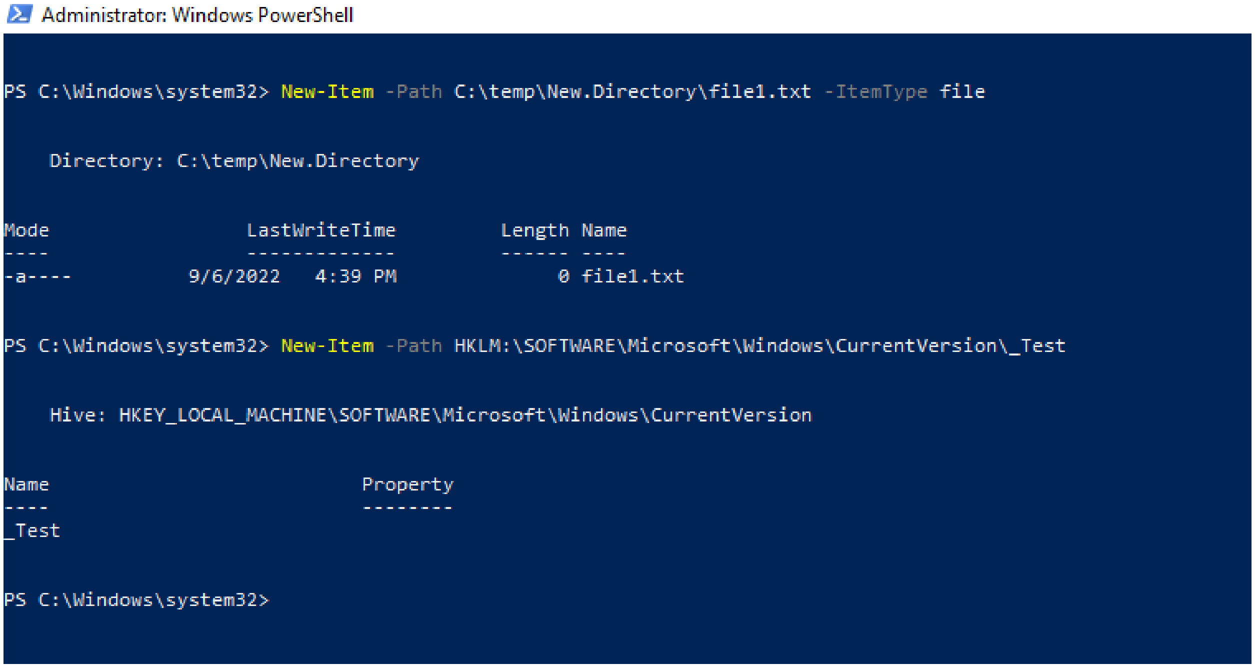
```
New-Item -Path C:\temp\New.Directory\file1.txt -ItemType file
```

You can use the same technique to create a new registry key. In fact, a registry key is easier to create because the only item type in the Windows registry is a key. (Registry entries are item **properties**.) For example, to create a key named "_Test" in the CurrentVersion subkey, type:

PowerShellCopy

```
New-Item -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\_Test
```

OutputCopy



```
PS C:\Windows\system32> New-Item -Path C:\temp\New.Directory\file1.txt -ItemType file


    Directory: C:\temp\New.Directory


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          9/6/2022   4:39 PM              0 file1.txt


PS C:\Windows\system32> New-Item -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\_Test


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion


Name                           Property
----                           --------
_Test


PS C:\Windows\system32>
```