**Viewing Object Structure (Get-Member)**

* Article
* 06/29/2022
* 2 minutes to read
* 3 contributors

Because objects play such a central role in PowerShell, there are several native commands designed to work with arbitrary object types. The most important one is the Get-Member command.

The simplest technique for analyzing the objects that a command returns is to pipe the output of that command to the Get-Member cmdlet. The Get-Member cmdlet shows you the formal name of the object type and a complete listing of its members. The number of elements that are returned can sometimes be overwhelming. For example, a process object can have over 100 members.

To see all the members of a Process object and page the output so you can view all of it, type:

PowerShellCopy

Get-Process | Get-Member | Out-Host -Paging

The output from this command will look something like this:

OutputCopy

TypeName: System.Diagnostics.Process

Name MemberType Definition

---- ---------- ----------

Handles AliasProperty Handles = Handlecount

Name AliasProperty Name = ProcessName

NPM AliasProperty NPM = NonpagedSystemMemorySize

PM AliasProperty PM = PagedMemorySize

VM AliasProperty VM = VirtualMemorySize

WS AliasProperty WS = WorkingSet

add\_Disposed Method System.Void add\_Disposed(Event...

...

We can make this long list of information more usable by filtering for elements we want to see. The Get-Member command lets you list only members that are properties. There are several forms of properties. The cmdlet displays properties of any type if we set the **MemberType** parameter to the value **Properties**. The resulting list is still very long, but a bit more manageable:

PowerShellCopy

Get-Process | Get-Member -MemberType Properties

OutputCopy

TypeName: System.Diagnostics.Process

Name MemberType Definition

---- ---------- ----------

Handles AliasProperty Handles = Handlecount

Name AliasProperty Name = ProcessName

...

ExitCode Property System.Int32 ExitCode {get;}

...

Handle Property System.IntPtr Handle {get;}

...

CPU ScriptProperty System.Object CPU {get=$this.Total...

...

Path ScriptProperty System.Object Path {get=$this.Main...

...

**Note**

The allowed values of MemberType are AliasProperty, CodeProperty, Property, NoteProperty, ScriptProperty, Properties, PropertySet, Method, CodeMethod, ScriptMethod, Methods, ParameterizedProperty, MemberSet, and All.

There are over 60 properties for a process. The reason PowerShell often shows only a handful of properties for any well-known object is that showing all of them would produce an unmanageable amount of information.

**Note**

PowerShell determines how to display an object type by using information stored in XML files that have names ending in .format.ps1xml. The formatting data for process objects, which are .NET System.Diagnostics.Process objects, is stored in DotNetTypes.format.ps1xml.

If you need to look at properties other than those that PowerShell displays by default, you will need to format the output data yourself. This can be done by using the format cmdlets.