Replicating (and Extending) the DIR Command

In its basic form the Get-ChildItem cmdlet provides functionality similar to the dir command. For example, if you simply type Get-ChildItem at the Windows PowerShell prompt you'll get back information about the objects in the current location:

[crayon-6a9d5f54bc58b069800896/]

That's all well and good, but you can do a lot more with Get-ChildItem than simply list the items found in the current location. For example, in the output above you might have noticed that there wasn't much to look at; that's because the current location happened to be a folder that contained only a handful of subfolders. Because of that you might have found it a bit more useful if Get-ChildItem had returned not only the names of those subfolders but also the contents of those subfolders; that is, you might want a list of all the files and folders in the subfolders. No problem; just add the -recurse parameter:

[crayon-6a9d5f54bc591025280142/]

Of course, you aren't limited to working with only the current location; in fact, you aren't limited to working with just files and folders. Would you like to see a list of all your environment variables? Then simply pass along the path to the environment variable «drive,» like so:

[crayon-6a9d5f54bc593606224018/]

What about all the registry subkeys found in HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall? Why not:

[crayon-6a9d5f54bc595920521562/]

Note. Get-ChildItem cannot be used to retrieve information about the registry values contained within a subkey. For that you need to use the Get-ItemProperty cmdlet.

We could do this all day. For example, the -include and -

exclude parameters make it easy to retrieve a specific set of items from a location. Suppose you want information about only the .txt and .log files found in the folder C:\Scripts? That's easy:

[crayon-6a9d5f54bc597639839858/]

As you can see, we ask for all the files (*.*) found in the folder C:\Scripts. We then tack on the -include parameter, specifying two file types: *.txt and *.log. (And separating the file types using a comma). What do we get back? We get back only .txt and .log files:

[crayon-6a9d5f54bc598366569203/]

If we wanted to get back everything except .txt and .log files then we'd simply use the -exclude parameter instead; this parameter tells Windows PowerShell which items should not be included in the returned dataset. Here's what the command looks like:

[crayon-6a9d5f54bc59a526034156/]

Give it a try and see what happens.

The information returned by Get-ChildItem can also be piped into the Sort-Object cmdlet, providing a way to sort the data by in some other format. Would you rather see files sorted by size (length) than by name? Then use this command:

[crayon-6a9d5f54bc59b686710815/]

Or, if you'd rather see the largest files listed first and the smallest files listed last, then add the -descending parameter:

[crayon-6a9d5f54bc59d647390542/]