Windows PowerShell Blog

Automating the world one-liner at a time...

Viewing Junctions with 'dir'

PowerShell Team 9 Feb 2010 5:18 PM 7

One feature of the NTFS file system is the junction, which is similar to a short cut but works at the file system level. This lets you link one directory to another. There's a tool called 'junction' available here that lets you manipulate junctions.

When listing the contents of a directory, by default PowerShell doesn't tell you which sub-directories are junctions. I wanted to be able to tell which ones were junctions when doing a 'dir'.

The way I did that was to copy the built in file system formatting file, modify it so that junctions are indicated, then load it with Update-FormatData:

The file system formatting rules are in $pshome\FileSystem.Format.ps1xml. I copied this, then in the element [ViewDefinitions –> View –> TableControl –> TableRowEntries –> TableRowEntry –> TableColumnItems –> TableColumnItem] I changed the content of PropertyName with value of 'Mode' to the following:

```
<ScriptBlock>

   "$($_.Mode)$(if($_.Attributes -band [IO.FileAttributes]::ReparsePoint) {'J'})"
</ScriptBlock>
```

This does a bitwise AND on the DirectoryInfo object Attributes property ($_.Attributes) against the .Net System.IO.FileAttributes.ReparsePoint enum value. If the result is not zero, it displays a 'J' next to the other file mode attributes.

Next, load the new formatting file like this:

PS> Update-FormatData -PrependPath myFilesystem.format.ps1xml

The PrependPath parameter ensures that the new formatting file is loaded before the built-in formatting files.

Let's see what the output looks like:

```
PS> dir

    Directory: C:\tmp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----J         2/9/2010   3:51 PM            alink
d----          2/9/2010   3:51 PM            notAlink
```

Directory alink has a 'J' in the mode column, seems to work!

cheers, Nigel Sharples [MSFT]

Comments

9 Feb 2010 11:57 PM

FP

Oh, so if I understand correctly, "junctions" are just like the soft links Unix has had for, what, 30 years ? Where is the ln utility ? And why on Earth is it necessary to copy and paste (!) code from the equivalent of "ls" to see that some items are in fact soft links ?

10 Feb 2010 2:15 AM

Blake

This works, but it is not a good practice, because the it leaves the impression that the Mode property includes that trailing J when it doesn't.

The better practice is to add a new ScriptProperty to a local types.ps1xml file for both DirectoryInfo and FileInfo types. Mine looks like this:

```
<ScriptProperty>
 <Name>Attrib</Name>
 <GetScriptBlock>
  if ($this.Attributes -band [IO.FileAttributes]::ReparsePoint) {
    'l'+$this.Mode
  } else {
    '-'+$this.Mode
  }
 </GetScriptBlock>
</ScriptProperty>
```

Then use a local format.ps1xml to display that property for table formatting rather than the Mode property.

(I used an 'l' to mirror the unix-y look of the rest of usual Mode string, because ReparsePoint can be set for many things other than a directory junction, and because 'r' is already used.)

10 Feb 2010 12:25 PM

Klaus Graefensteiner

This is great stuff. I am working with Junctions a lot. Every deployment of our software needs to change junction points.

Klaus

10 Feb 2010 11:16 PM

n4cer

@FP - Junctions are hard links, evaluated on the server in networking scenarios, and available in NT since Windows 2000.

Symbolic links, introduced on Windows in Windows Vista along with support for them in SMB2, are soft links and evaluated clientside.

Both link types are implemented using NTFS reparse points.

The equivalent to ln on Windows Vista and above is mklink. On versions prior to Vista, you can use linkd (shipped seperately in a Resource Kit) or the Sysinternals tool linked in the post.

Reparse points can also be shown in CMD's dir using the /AL switch, and in Vista/7's Explorer if the related columns are made visible.

14 Feb 2010 2:43 PM

L.

@PowerShellTeam: Come on, you should support current NTFS features out of the box! Even cmd.exe knows about junctions and symlinks.

@Nigel Sharples [MSFT]: This will display a J for any kind of reparse point, though, not only for junctions. Is it really all right to confuse junctions with e.g. files moved to offline storage? (IIRC reparse points are used in this case to trigger fetching back the file).

@n4cer: You description of junctions vs symlinks is right, but junctions are not hard links. Hard links are yet another kind of beast (a file that has several names, possibly in different directories), and are also supported by NTFS.

9 Mar 2010 10:50 PM

jon

@L.

I agree. PowerShell really ought to support for listing and manipulating junctions, hard links and symlinks out of the box.

31 Jul 2011 5:53 PM

JFX

I agree with L. and Jon ... maybe in version 3 ?