



*Mastering Format and Type
Extensions*

Jeffery Hicks

The Problem

- `dir c:\work | Select Name,Length,@{Name="Owner";Expression={($_ | Get-ACL).owner }}`
- `Get-Process | Select name,id,path,starttime,@{name="Runtime";Expression={(get-date)-$_.Starttime }}`
- If I use these a lot, that's a lot of typing
- We need to take control of formatting

Formatting in a Nutshell

- What happens when you run a command?

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

- Out-Default --> Out-Host --> Format *
- PowerShell gets guidance from
 - Types.ps1xml
 - DotNetTypes.format.ps1xml

Demo

- Demo-FormattingNutshell



Formatting Rules

- If nothing else called for, check for object specific view
- If Format*, check for object specific view
- If no view and no formatting, then use default properties from types.ps1xml
- If nothing defined then return all properties
- Table or list depends on # of properties

The Solution

- Create a custom type and/or format XML file
 - Do NOT modify the Microsoft files
 - Copy and paste a starter section into a new file
 - Don't forget XML tags are cAsE Sensitive
- Add it into our PowerShell session
 - Update-TypeData
 - Update-FormatData
- Enjoy!

Common Entries

- AliasProperty
- ScriptProperty
- ScriptMethod
- PropertySet

Demos



- String Reverse
- FileSystem Enhancements
- Process Enhancements
- PowerShell Bingo Module

Caveats and Tips

- Create your own xml files
- Test with short paths
- Execution policy applies to ps1xml files
- Package with your modules
- Use an XML editor

Resources

- Windows PowerShell 2.0: TFM (SAPIEN Press 2010)
- PowerShell in Action (Manning 2011)
- `About_types.ps1xml`
- `About_format.ps1xml`

Thank You

- jhicks@jdhitsolutions.com
- <http://jdhitsolutions.com/blog>
- <http://twitter.com/jeffhicks>