PowerShell Providers

What is a PowerShell Provider?

Software that translates data storage into a PowerShell-friendly format

Typically hierarchical data structures

- FileSystem
- Registry
- Certificate store
- Active Directory

What is a PowerShell Provider?

But also configuration sources

- WSMan
- IIS
- Environment

Additional providers may be added via modules

Run Get-PSProvider to view installed providers

Read help topic about_providers

Using Providers

Common providers
- FileSystem
- Registry
- Certificate
- WSMan

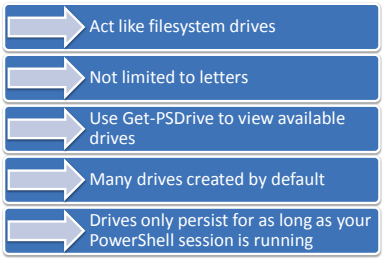
Data storage exposed as a PSDrive

Provider translates cmdlet actions to the "drive"

Providers may expose dynamic parameters

Providers may expose dynamic help examples

PSDrives




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


Creating a PSDrive

Use New-PSDrive
- Specify drive name
- Specify provider
- Specify path
- Options: credential, persistence

The drive name does not include the colon

Drives are scope specific

Drives not necessarily seen by Windows

Add drive settings to your PowerShell profile script

```
PS C:\> New-PSDrive IT FileSystem \\chi-fp01\IT
```

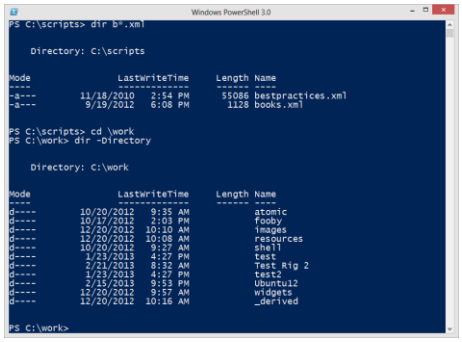
Navigating PSDrives

- Use the colon when accessing a drive
- Use same set of cmdlets
- Some providers require ItemProperty cmdlets
- Some PSDrives require elevated privileges

FileSystem

- Use traditional file system commands
- PowerShell should detect mapped Windows drives
- Create persistent drives
- External drives automatically mapped
- Tip: Create your own shortcut drives

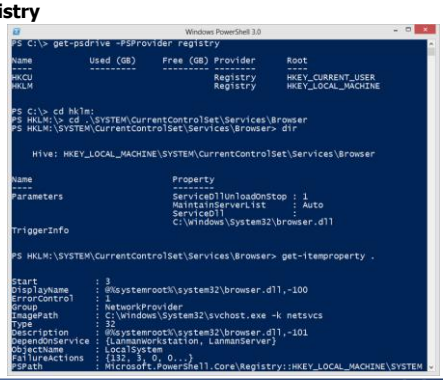
FileSystem



Registry

- Only works locally
- Default drives to HKLM and HKCU
- Technically no filtering…but there are ways
- Some information stored as an Item property

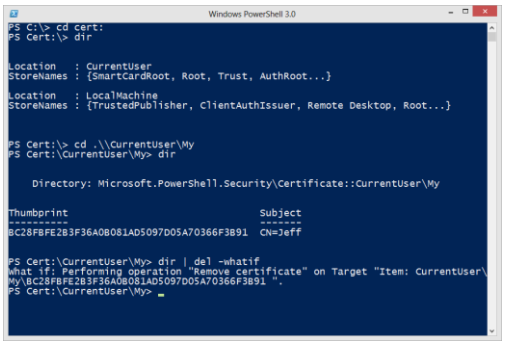
Registry



Certificate

- Access certificate store for local user and computer
- Hierarchical storage of certificate objects
- Certificates as “files”
- Objects can be deleted

Certificate



Environment

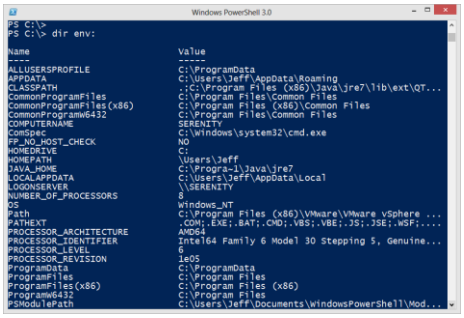
Access environmental variables

Use for read-only references

Access common settings as variables:

- $env:computername
- $env:userdomain
- $env:username
- $env:temp

Environment



Removing a PSDrive

- User defined drives end with session
- Use Remove-PSDrive to manually removal
- Can't remove Windows or fixed drives
- Don't try to remove your current drive
- Remove persistent drive mappings

```
PS C:\> Remove-PSDrive X
```


PowerShell Provider Demos

Provider Sources

Additional providers may be added from modules, server features, or third party tools

- Active Directory
- IIS
- SQL Server
- PowerShell Community Extensions
- VMware PowerCLI

Lab

1. Create a new PSDrive called TEMP that is mapped to the %TEMP% environmental variable

2. Change to the new drive and recursively list all files and folders

3. Using the registry find the value for ProductName under HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion