POWER SHELL MANAGEMENT IN A NUTSHELL

Jeffery Hicks
Windows PowerShell MVP
jhicks@jdhitsolutions.com
AGENDA

- What is PowerShell?
- Why does it Matter?
- Key Features
- Demos
- Resources
- Q&A
Who Am I?

- Windows PowerShell MVP
- PowerShell Author
  - Windows PowerShell 2.0: TFM (with Don Jones)
  - Managing Active Directory with Windows PowerShell: TFM (2nd Ed.)
  - PowerShell in Depth (with Don Jones and Richard Siddaway) in progress
  - MCPMag.com Prof. PowerShell
- IT trainer and consultant
- http://jdhitsolutions.com/blog
- http://twitter.com/jeffhicks
- https://plus.google.com/109354722869529171746
What is PowerShell?

- Windows PowerShell is an object-oriented management *engine* built on the .NET Framework.
- It is Microsoft's required server management tool.
- It is primarily exposed to IT Pros as a console.
- ...but it can be "hosted" by other applications.
- GUIs built on top of PowerShell.
What is PowerShell?

- 1.0 2006 Windows XP and later
- 2.0 2009 Added remoting, background jobs
- Requires .NET Framework 3.0 and later
- Now a core part of the operating system
- Growing adoption by Microsoft product teams
- Growing adoption by vendors
Why Does it Matter?

- In the past IT Pro's needed to learn different tools and techniques to manage different platforms.
- GUIs are good for one-off tasks but don't scale.
- Enterprise efficiency requires automation.
- PowerShell provides an interactive management console.
- ...that also supports batch-like scripting.
- It's not a matter of if you'll be using PowerShell only a matter of when.
**Key Features**

- PowerShell uses objects, not text.
- Objects are passed through a pipeline.
- Cmdlets are single purpose commands designed to work with objects in the pipeline.
- At the end of the pipeline is your result.
## Pipeline Example

PS C:\> Get-Service

<table>
<thead>
<tr>
<th>Status</th>
<th>Name</th>
<th>DisplayName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stopped</td>
<td>AeLookupSvc</td>
<td>Application Experience</td>
</tr>
<tr>
<td>Stopped</td>
<td>ALG</td>
<td>Application Layer Gateway Service</td>
</tr>
<tr>
<td>Stopped</td>
<td>AppIDSvc</td>
<td>Application Identity</td>
</tr>
<tr>
<td>Stopped</td>
<td>Appinfo</td>
<td>Application Information</td>
</tr>
<tr>
<td>Stopped</td>
<td>AppMgmt</td>
<td>Application Management</td>
</tr>
<tr>
<td>Stopped</td>
<td>aspnet_state</td>
<td>ASP.NET State Service</td>
</tr>
<tr>
<td>Running</td>
<td>AudioEndpointBu</td>
<td>Windows Audio Endpoint Builder</td>
</tr>
<tr>
<td>Stopped</td>
<td>BDESVC</td>
<td>BitLocker Drive Encryption Service</td>
</tr>
<tr>
<td>Running</td>
<td>BFE</td>
<td>Base Filtering Engine</td>
</tr>
<tr>
<td>Running</td>
<td>BITS</td>
<td>Background Intelligent Transfer Service</td>
</tr>
</tbody>
</table>

Text based shells would need to parse this output.
PS C:\> Get-Service | Where {$_.Status -eq "running"}

<table>
<thead>
<tr>
<th>Status</th>
<th>Name</th>
<th>DisplayName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>AudioEndpointBuilder</td>
<td>Windows Audio Endpoint Builder</td>
</tr>
<tr>
<td>Running</td>
<td>AudioSrv</td>
<td>Windows Audio</td>
</tr>
<tr>
<td>Running</td>
<td>BFE</td>
<td>Base Filtering Engine</td>
</tr>
<tr>
<td>Running</td>
<td>BITS</td>
<td>Background Intelligent Transfer Service</td>
</tr>
<tr>
<td>Running</td>
<td>bthserv</td>
<td>Bluetooth Support Service</td>
</tr>
<tr>
<td>Running</td>
<td>cmdAgent</td>
<td>COMODO Internet Security Helper Service</td>
</tr>
<tr>
<td>Running</td>
<td>ConfigFree</td>
<td>ConfigFree Service</td>
</tr>
<tr>
<td>Running</td>
<td>CryptSvc</td>
<td>Cryptographic Services</td>
</tr>
</tbody>
</table>

Get all Service objects

Only keep objects that are running
**KEY FEATURES - REMOTE MANAGEMENT**

- You can gather information and manage remote computers using .NET and WMI
- PowerShell v2.0 uses WinRM and the WSMAN protocol to execute commands *on* the remote machine
- You can run a command on your desktop that runs *simultaneously* on 1-1000 (more machines).
KEY FEATURES - MORE

- Background jobs
- Integrated help system
- Extended functionality through snapins and modules
- Simple scripting language
- Large and growing PowerShell community
Demos

- Demos are written mostly as one-liners.
- Focus on results not language or syntax
RESOURCES - BOOKS

- Windows PowerShell 2.0: TFM by Don Jones and Jeffery Hicks
- Learn PowerShell in a Month of Lunches by Don Jones
- Windows PowerShell Cookbook 2nd Ed. by Lee Holmes
- Windows PowerShell in Action 2nd Ed. by Bruce Payette
Resources - Web

- Windows PowerShell Team blog (http://blogs.msdn.com/powershell)
- The Lonely Administrator (http://jdhitsolutions.com/blog)
- Prof. PowerShell (http://mcpmag.com/articles/list/prof-powershell.aspx)
RESOURCES - TRAINING

- Windows PowerShell Fundamentals Training (TrainSignal)
- Windows Server 2008 PowerShell Training (TrainSignal)
- Conferences: TechEd, TechMentor, The Experts Conference
- Private and public training
Questions
THANK YOU

- http://jdhitsolutions.com/blog
- jhicks@jdhitsolutions.com