Help Video: MVP Summit

With Adam Bertram and June Blender

**Welcome intro:** Help is so important to users, but it can be a drag for the scripter who just wants to get working on the next big thing.

**Goal: Let's find ways to make it easier.** And with better quality.

1. Comment-based help
2. XML help
3. Using prompts ("
4. Online help
5. Updatable help

**Questions:**

How do I write comment-based help?

-- Use your tools.

------ Snippets (In various tools)

------ PowerShell Studio:

---------- Generate comment-based help as a GUI

---------- PrimalSense

* It's a pain to write help. Do you have any suggestions to make it easier?

-- Use checklists. These contain prompts that help you write the content. They include answers to questions that users might have, but that were obvious to the author.

For parameters:

-- Reuse the description of the parameters that you call. Often you get a parameter value just to pass it to another function, like Get-ADUser or Get-WmiObject. In that case, reuse the parameter description of the cmdlet that you're calling.

-- Reuse good descriptions. (This helps you choose standard parameter names, too.)

Get-Command <Command> -Parameter <Name>

Get-Command <Command> -Parameter <Name> | foreach { Get-Help <Command> -Parameter Name}

Get-ParameterDescriptions.ps1 | Out-GridView

* I've got dozens of functions that I've been writing comment-based help in. I'm finding that I forget to update the help when I make a change. Is there an easies way to manage help for many functions at once like this?
* How do I write XML help?

-- Use your tools!

----- PowerShell HelpWriter

----- Use existing XML help as a template

* I wrote XML help, but how do I name my help file.

There are different rules for PS 3.0 and 4.0+. The 3.0 rules are more strict and they work everywhere.

* If it's a function, you can pick any name, but it's customary to use the module name + help.xml
* For other command types, the XML file must be named for the file in which the command is defined + help.xml
* In 4.0, you can put all of the XML help topics for all commands in a single module and name it for the module, e.g. MyModule-help.xml.
* Hmm, this isn't working. I still see my comment-based help.

Comment-based help takes precedence over XML help if you have both.

To use XML help for advanced functions, you need to add the .ExternalHelp comment keyword to the function. It works if you don't have comment-based help, and it takes precedence over other comment-based help keyword -- essentially, telling Get-Help to ignore the CBH and use XML help.

The .ExternalHelp comment keyword is required for all advanced functions in PS 3.0. In 4.0+, it's required only for advanced functions in script modules. It's not required for advanced functions in manifest modules.

* Now it's working. But if I update and then run Get-Help, I don't see my changes.

You need to restart your Windows PowerShell session. The help is cached at the beginning of each session.

* I ran Update-Help on this module, but I'm getting an error. What's up with that?

You can check to see if the module supports updatable help. If the module supports updatable help, the **HelpInfoUri** property of the module has a URL value.

function Test-UpdatableHelp

{

param

(

[Parameter(Mandatory = $true,

ValueFromPipeline = $true)]

[ValidateScript({ Get-Module -List $\_ })]

[string]$Module

)

$uri = (Get-Module $Module -ListAvailable).HelpInfoUri

return $Uri -ne "" -and $Uri -ne $null

}

PS C:\ps-test> (Get-Module PSScheduledJob -List).HelpInfoUri

http://go.microsoft.com/fwlink/?linkid=390816

PS C:\ps-test> (Get-Module PSCX -List).HelpInfoUri

Ah, so the PSCX module doesn't support Updatable Help, but the PSScheduledJob module does. So, if the module has a HelpInfoUri value and Update-Help returns an error, the help for module might not be ready. It's set up for updatable help, but the help isn't yet in place.

* Updatable Help work for the PSScheduledJob module, but when I go to that URL, I don't see the help. I get an error.

That's right. This is updatable help, not online help. In updatable help, the HelpInfoUri value is the online location of the HelpInfo XML file for the module. It's like an .inf file. It's not a web page.

* So, that's not online help. How do I tell if a module supports online help?

Look at the value of the HelpUri property of the CommandInfo object -- the one that Get-Command returns.

(Clarify. Updatable Help is a property of a module -- the Help**Info**Uri property. Online help is a property of a command. It's the HelpUri property.)

And you can look at that URL, because it's really the location of the online help for the command.

Start-Process (Get-Command <Command>).HelpUri

Start-Process (Get-Command Invoke-Command).HelpUri

* How do I support online help for my function.

First, you need a web page with the help for your function. The Convert-HelpToHtml filter (like a function) in Aaron Jensen's Silk module is a great start: https://bitbucket.org/splatteredbits/silk

Then, in your function, in the CmdletBinding attribute, add a HelpUri parameter. The value of HelpUri is the URL of the online help topic.

function Get-Hello

{

[CmdletBinding(HelpUri = 'http://help.sapien.com')]

Param ()

}

You can also put the HelpUri value in your comment-based help or XML help. It goes in the value of the first item in the Related Links section.

<#

.SYNOPSIS

Says 'hello' from comment-based help.

.DESCRIPTION

Says 'hello' from comment-based help.

.PARAMETER Name

Says 'hello' to the specified name.

.NOTES

Additional information about the function.

.LINK

http://help.sapien.com

#>

function Get-Hello

{

[CmdletBinding(HelpUri = 'http://help.sapien.com')]

Param ()

}

If you have both a HelpUri value and a first related link, the first related link takes precedence.

Most of the PowerShell core cmdlets have both. Not sure why, but in PowerShell 5.0, the HelpUri opens PS 4.0 help and the first related link opens PS 5.0 help.

PS C:\ps-test> (Get-Command Get-Process).HelpUri

http://go.microsoft.com/fwlink/?LinkID=113324

PS C:\ps-test> (Get-Help Get-Process).RelatedLinks[0].NavigationLink.Uri[0]

http://go.microsoft.com/fwlink/p/?linkid=290498