Windows PowerShell 4.0
For the IT Professional - Part 1

Module 6: Help System

Student Lab Manual

Version 2.0

Conditions and Terms of Use

Microsoft Confidential - For Internal Use Only

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

Contents

Lab 6: The Help System

Introduction

Windows PowerShell includes a comprehensive help system. Full help topics are not installed by default. To support frequent documentation updates all help files may be downloaded from online sources. In a typical organization, only the computers where scripts are written should require help files. Windows PowerShell also has Cmdlets for visual help.

Objectives

After completing this lab, you will be able to:

- Understand how to download help files from the internet.
- Distribute help files to other computers with no internet access.
- Use options to display help.

Prerequisites

Start all VMs provided for the workshop labs.

Logon to WIN8-WS as:

Username: **Contoso\Administrator**

Password: **PowerShell4**

Estimated time to complete this lab

30 minutes

NOTE: Of the virtual machines in the labs, **WIN8-WS** already has updated help. **2012R2-MS** does not have the help files installed, but does have an offline copy saved to **\\2012R2-MS\PowerShell-Help**.

These exercises use many Windows PowerShell commands. You can type these commands into the Windows PowerShell Integrated Scripting Environment (ISE) or the Windows PowerShell console. For some exercises, you can load pre-typed lab files into the Windows PowerShell ISE, allowing you to select and execute individual commands. Each lab has its own folder under **C:\PShell\Labs\** on **WIN8-WS**.

Some exercises in this workshop may require running the Windows PowerShell console or ISE as an elevated user (Run as Administrator).

We recommend that you connect to the virtual machines (VMs) for these labs through Remote Desktop rather than connecting through the Hyper-V console. This allows you to use copy and paste between VMs and the host machine. If you are using the online hosted labs, then you are already using Remote Desktop.

Exercise 6.1: Getting Help Files - Without Internet

Introduction

Many computers do not have an internet connection, but they can still update the Windows PowerShell help files from an offline copy. Most organizations will create a central share on an internal file server to host the latest Windows PowerShell help files. Then they will schedule a task from an internet-connected machine to keep these files updated. Even if all computers do have internet connectivity, this is still a good way to minimize internet traffic.

Objectives

After completing this exercise, you will be able to:

- Update Windows PowerShell help files from a local repository.

Prerequisites

Start all VMs provided for the workshop labs.

Logon to 2012R2-MS server as Contoso\Administrator using the password "PowerShell4".

Server 2012R2-MS has a folder containing Windows PowerShell help files previously downloaded with the Save-Help Cmdlet.

Task 6.1.1: Loading the saved help files (Internet not required)

1. Logon to 2012R2-MS server
2. Right-click the Windows PowerShell icon on the taskbar and select the "Run ISE as Administrator" option. This will launch the Windows PowerShell Integrated Scripting Environment.
3. List the available downloaded help files.

```
Get-Childitem C:\PowerShell-Help
```

4. Update the local help files:

```
Update-Help -Verbose -Force -SourcePath C:\PowerShell-Help
```

- Make sure you specify the correct **-SourcePath**.
- **-Force** and **-Verbose** are both optional.

> **Note:** If Update-Help reports an error, “Unable to connect to Help content”, then use this temporary work-around and then try again:
>
> https://connect.microsoft.com/PowerShell/feedback/details/777012/update-help-sourcepath-failure-on-computer-without-internet-connection#details

```
Get-Childitem $PSHome -Filter *helpinfo.xml -Recurse | Remove-Item -
Confirm:$false
```

5. Now verify that the Help System is working. Remember that you can use TAB completion when typing cmdlets names and about* help topic names.

```
Get-Help Get-Process -Full
Get-Help Get-NetAdapter -Examples
Get-Help about_Updatable_Help
```

Exercise 6.2: Graphical Help

Introduction

In this exercise, you will learn some new options to access help graphically. You will be able to locate help items, build parameters more easily and find keywords inside the help text.

Objectives

After completing this exercise, you will be able to:

- Use new cmdlets to get graphical help on cmdlets and parameters.
- Find text inside help.

Prerequisites

Start all VMs provided for the workshop labs.

Logon to **WIN8-WS** as Contoso\Administrator, using the password “PowerShell4”.

Task 6.2.1: Using Get-Help -ShowWindow

1. Launch the Windows PowerShell ISE on **WIN8-WS**.
2. Run this command.

   ```
   Get-Help Get-Process -ShowWindow
   ```

   - A separate window will open.
   - Scroll up and down.
   - Change the font size using the slider control at the bottom right corner.
3. Using the textbox at the top of the window, find all references to the text **MainWindowTitle** in Get-Process.
   - Notice that the number of matches will update as you type.
   - You should be able to find three matches. Use the **Next** button.
   - Use the **Settings** button on the right to filter the displayed information and to adjust the search options.

```
  “The second command gets the Windows PowerShell
process that is hosting the current session.
  -------------------------- EXAMPLE 10 --------
------------------
  PS C:\>get-process | where
{$_.mainWindowTitle} | format-table id, name,
mainwindowtitle -autosize”
```

Task 6.2.2: Using Show-Command

1. Launch the Windows PowerShell ISE on **WIN8-WS**.
2. Run this command.

```
Show-Command Get-Process
```

- You will get a different popup window with a separate tab for each one of the possible parameter sets for this Cmdlet.

3. Using the **Show-Command** graphical user interface**,** build and then execute these commands with the given parameters:

```
Get-Process -Id 4 -ErrorAction SilentlyContinue
Get-Process -Id $pid -Module
```

- Notice that the enumerable options (like **-ErrorAction**) are displayed as lists.
- You can use the **Copy** button to send the command to the clipboard.

> **NOTE:** See `Help About_Automatic_Variables` to identify the **$PID** variable.

Contains the process identifier (PID) of the process that is hosting the current Windows PowerShell session.

4. In the **Show-Command Get-Process** window, in the **Id** tab, next to the **Id** parameter there is an asterisk (*) visible in the dialog. What does it mean?

Means this is a mandatory parameter.

> **HINT:** Mouse over the textbox for a clue.

5. Once built, execute the commands by clicking **Run**.

- The dialog will disappear and Windows PowerShell will execute the command. Verify its execution. Type this command.

```
Get-History
```

6. Run the following command.

```
Show-Command Get-Process
```

7. Click the **Help** icon (question mark) on the upper right corner of the Show-Command dialog. What do you get?

Help about the cmdlet Get-Process
"…**Description**   The Get-Process cmdlet gets the processes on a local or remote computer."

8. By default the line below does not store any value in the variable. Try this in the console:

```
Show-Command Get-Process –Outvariable result
$result
```

9. Add the **-PassThru** switch and try again:

```
Show-Command Get-Process -PassThru  -Outvariable result
```

Now the **Run** button is no longer available, but you have an **OK** button instead. When you click **OK** the full command is saved in a string variable and returned by the Cmdlet. The **$Result** variable should contain your command. You can now execute it by piping to **Invoke-Expression**.

```
$result | Invoke-Expression
```

Exercise 6.3: Getting Help Files - Online

Introduction

This exercise is provided for future reference and has been done for you.

Objectives

After completing this exercise, you will be able to download help files and stage them for offline help updates.

Scenario

You need to update your own help files and provide the latest help files for computers without internet access.

6.3.1 FOR FUTURE REFERENCE: Updating from the internet

> **NOTE**: The virtual machines used for this workshop do not have Internet access, so these steps are provided for you to follow when you return to your work environment.

1. Run Windows PowerShell ISE in elevated mode (Run As Administrator).
2. Check the help system status :

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

   If Windows PowerShell has been recently installed and does not have internet access, you will not have access to the full help system. Only automatically generated help will be displayed.

3. Update all help files:

```
PS C:\> Update-Help -Verbose
```

4. Repeat the help command from step 2. The result will contain much more information.

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

5. On the same computer, try to update all help files again, using the same Cmdlet and syntax . Note that the files will not download again. You need to wait 24 hours. This is meant to reduce inadvertent traffic in cases where this might be added to a script.

```
PS C:\> Update-Help -Verbose
```

6. Update all help files again but add the **-Force** parameter:

```
PS C:\> Update-Help -Verbose -Force
```

TIP: Help files are frequently updated. Remember to update your local help regularly. See **`help about_Updateable_Help`** for more information and for an example of scheduling a task to update help.

FOR FUTURE REFERENCE: Saving help files from the Internet

NOTE: The virtual machines used for this workshop do not have Internet access, so these steps are provided for you to follow when you return to your work environment.

Tip: It is best to run **Save-Help** as Administrator on an internet-connected machine that has all of the modules installed that will be used on the machines using the saved help files. A client with the Remote Server Administration Tools (RSAT) installed will already have a majority of the Windows PowerShell modules installed.

For example, if you want the ActiveDirectory module help files for a domain controller to update, then you should run **Save-Help** from a client that has the ActiveDirectory module installed from the RSAT. Windows PowerShell 3.0 has a limitation that **Save-Help** will only download help for modules that are installed on the local machine. Windows PowerShell 4.0 has addressed this limitation. For more information, see **`Help Save-Help -Full`**.

7. Launch the Windows PowerShell ISE as **Administrator**.
8. Create a destination folder.

```
New-Item -Path C:\TheHelpFiles –ItemType Directory
```

- The destination folder can be any location with approximately 5MB of free disk space, even an external USB drive.

9. View the internet path for the module help files:

```
Get-Module -ListAvailable | Format-Table Name, HelpInfoURI
```

10. Save a local copy of the help files:

```
Save-Help -Verbose -Force -DestinationPath C:\TheHelpFiles
```

- You do not specify a source for **Save-Help**. The files will always download from the internet.
- **-DestinationPath** is a mandatory parameter.
- **-Verbose** is an optional parameter.
- **-Force** has the same behavior we saw in **Update-Help**. Use it when you must download all files again within a 24-hour period.

11. Review the received files and their total size:

```
Get-Childitem C:\TheHelpFiles
```

12. Logon to the **WIN8-WS** client machine to review the saved help files for the labs:

```
Get-Childitem \\2012R2-MS\PowerShell-Help
```

These lab help files will not be completely current, but they will provide help for all Cmdlets covered in this course.