Windows PowerShell Get-Help on Cmdlet 'Update-Help'

PS:\>Get-HELP Update-Help -Full

NAME

   Update-Help

SYNOPSIS

   Downloads and installs the newest help files on your computer.

SYNTAX

      Update-Help   [[-Module]   <System.String[]>]   [[-UICulture]   <System.Globalization.CultureInfo[]>]   [-Credential
<System.Management.Automation.PSCredential>] [-Force]

    [-FullyQualifiedModule   <Microsoft.PowerShell.Commands.ModuleSpecification[]>]   [-LiteralPath   <System.String[]>]
[-Recurse] [-UseDefaultCredentials] [-Confirm]

  [-WhatIf] [<CommonParameters>]

      Update-Help   [[-Module]   <System.String[]>]   [[-SourcePath]   <System.String[]>]   [[-UICulture]
<System.Globalization.CultureInfo[]>] [-Credential

               <System.Management.Automation.PSCredential>]   [-Force]   [-FullyQualifiedModule
<Microsoft.PowerShell.Commands.ModuleSpecification[]>] [-Recurse]

  [-UseDefaultCredentials] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

   The `Update-Help` cmdlet downloads the newest help files for PowerShell modules and installs them on your computer. You need not restart PowerShell to make the change

   effective. You can use the `Get-Help` cmdlet to view the new help files immediately.

   `Update-Help` checks the version of the help files on your computer. If you don't have help files for a module or if your help files are outdated, `Update-Help`

   downloads the newest help files. The help files can be downloaded and installed from the internet or a file share.

   Without parameters, `Update-Help` updates the help files for modules that support updateable help and are loaded in the session or installed in a location included in

   the `$env:PSModulePath`. For more information, see about_Updatable_Help (./About/about_Updatable_Help.md).

   `Update-Help` checks the version of the help installed. If `Update-Help` can't find updated help files for a module it continues silently without displaying an error

   message. Use the Force parameter to skip the version check. Use the Verbose parameter to see status and progress details. Use the Module parameter to update help

   files for a particular module.

   If the cultural settings of your operating system are configured for a language that's not available for updateable help, `Update-Help` continues silently without

   downloading any help. Use the UICulture parameter to download help files in a supported language. Help is always available for the `en-US` locale.

   You can also use `Update-Help` on computers that aren't connected to the internet. First, use the `Save-Help`cmdlet to download help files from the internet and save

   them in a shared folder that's accessible to the system not connected to the internet. Then use the SourcePath parameter of `Update-Help` to download the updated help

   files from the shared and install them on the computer.

   The `Update-Help` cmdlet was introduced in Windows PowerShell 3.0.

> [!IMPORTANT] > `Update-Help` requires administrative privileges. > > You must be a member of the Administrators group on the computer > to update the help files for

the core PowerShell modules. > > To download or update the help files for modules in the PowerShell > installation directory (`$PSHOME\Modules`), including the

PowerShell > Core modules, start PowerShell by using the Run as administrator option. > For example: `Start-Process powershell.exe -Verb RunAs`. > > You can also

update help files by using the Update Windows PowerShell Help > menu item in the Help menu in Windows PowerShell Integrated Scripting > Environment (ISE). > > The

Update Windows PowerShell Help item runs an `Update-Help` cmdlet > without parameters. > To update help for modules in the `$PSHOME` directory, > start Windows

PowerShell ISE by using the Run as administrator option.

PARAMETERS

  -Credential <System.Management.Automation.PSCredential>

    Specifies credentials of a user who has permission to access the file system location specified by SourcePath . This parameter is valid only when the SourcePath

    or LiteralPath parameter is used in the command.

    The Credential parameter enables you to run `Update-Help` commands with the SourcePath parameter on remote computers. By providing explicit credentials, you can

    run the command on a remote computer and access a file share on a third computer without encountering an access denied error or using CredSSP authentication to

    delegate credentials.

    Type a user name, such as User01 or Domain01\User01 , or enter a PSCredential object generated by the `Get-Credential` cmdlet. If you type a user name, you're

    prompted to enter the password.

    Credentials are stored in a PSCredential (/dotnet/api/system.management.automation.pscredential)object and the password is stored as a SecureString

    (/dotnet/api/system.security.securestring).

> [!NOTE] > For more information about SecureString data protection, see > How secure is SecureString? (/dotnet/api/system.security.securestring#how-secure-is-securestring).

Required?   false

Position?   named

Default value   Current user

Accept pipeline input?   False

Accept wildcard characters? false

-Force <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet doesn't follow the once-per-day limitation, skips version checking, and downloads files that exceed the 1 GB limit.

Without this parameter, `Update-Help` runs only once in each 24-hour period. Downloads are limited to 1 GB of uncompressed content per module and help files are

only installed when they're newer than the existing files on the computer.

The once-per-day limit protects the servers that host the help files and makes it practical for you to add an `Update-Help` command to your PowerShell profile

without incurring the resource cost of repeated connections or downloads.

To update help for a module in multiple UI cultures without the Force parameter, include all UI cultures in the same command, such as:

`Update-Help -Module PSScheduledJobs -UICulture en-US, fr-FR, pt-BR`

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-FullyQualifiedModule <Microsoft.PowerShell.Commands.ModuleSpecification[]>

The value can be a module name, a full module specification, or a path to a module file.

When the value is a path, the path can be fully qualified or relative. A relative path is resolved relative to the script that contains the using statement.

When the value is a name or module specification, PowerShell searches the PSModulePath for the specified module.

A module specification is a hashtable that has the following keys.

- `ModuleName` - Required Specifies the module name. - `GUID` - Optional Specifies the GUID of the module. - It's also Required to specify at least one of the

three below keys. - `ModuleVersion` - Specifies a minimum acceptable version of the module. - `MaximumVersion` - Specifies the maximum acceptable version of

the module. - `RequiredVersion` - Specifies an exact, required version of the module. This can't be used with   the other Version keys.

You can't specify the FullyQualifiedModule parameter in the same command as a Module parameter.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   True (ByPropertyName)

Accept wildcard characters? false

-LiteralPath <System.String[]>

Specifies the folder for updated help files instead of downloading them from the internet. Use this parameter or SourcePath if you've used the `Save-Help` cmdlet

to download help files to a directory.

You can pipeline a directory object, such as from the `Get-Item` or `Get-ChildItem` cmdlets, to `Update-Help`.

Unlike the value of SourcePath , the value of LiteralPath is used exactly as it's typed. No characters are interpreted as

wildcard characters. If the path

includes escape characters, enclose it in single quotation marks. Single quotation marks tell PowerShell not to interpret any characters as escape sequences.

```
Required?              false
Position?              named
Default value          None
Accept pipeline input?     True (ByPropertyName)
Accept wildcard characters?  false
```

-Module <System.String[]>

Updates help for the specified modules. Enter one or more module names or name patterns in a comma-separated list, or specify a file that lists one module name on

each line. Wildcard characters are permitted. You can pipeline modules from the `Get-Module` cmdlet to the `Update-Help` cmdlet.

The modules that you specify must be installed on the computer, but they don't have to be imported into the current session. You can specify any module in the

session or any module that's installed in a location listed in the `$env:PSModulePath` environment variable.

A value of `*` (all) attempts to update help for all modules that are installed on the computer. Modules that don't support Updatable Help are included. This

value might generate errors when the command encounters modules that don't support Updatable Help. Instead, run `Update-Help` without parameters.

The Module parameter of the `Update-Help` cmdlet doesn't accept the full path of a module file or module manifest file. To update help for a module that isn't in

a `$env:PSModulePath` location, import the module into the current session before you run the `Update-Help` command.

```
Required?              false
Position?              0
Default value          None
Accept pipeline input?     True (ByPropertyName)
```

Accept wildcard characters? true

-Recurse <System.Management.Automation.SwitchParameter>

    Performs a recursive search for help files in the specified directory. This parameter is valid only when the command

uses the SourcePath parameter.

    Required?   false

    Position?   named

    Default value   False

    Accept pipeline input?   False

    Accept wildcard characters? false

-SourcePath <System.String[]>

    Specifies a file system folder where `Update-Help` gets updated help files, instead of downloading them from the

internet. Enter the path of a folder. Don't

        specify a file name or file name extension. You can pipeline a folder, such as one from the `Get-Item` or

`Get-ChildItem` cmdlets, to `Update-Help`.

    By default, `Update-Help` downloads updated help files from the internet. Use SourcePath when you've used the

`Save-Help` cmdlet to download updated help files to

    a directory.

    To specify a default value for SourcePath , go to Group Policy , Computer Configuration , and Set the default source

path for Update-Help . This Group Policy

    setting prevents users from using `Update-Help` to download help files from the internet. For more information, see

about_Group_Policy_Settings

    (./About/about_Group_Policy_Settings.md).

    Required?   false

    Position?   1

    Default value   None

    Accept pipeline input?   False

    Accept wildcard characters? false

-UICulture <System.Globalization.CultureInfo[]>

Specifies UI culture values that `Update-Help` uses to get updated help files. Enter one or more language codes, such as es-ES , a variable containing culture

objects, or a command that gets culture objects, such as a `Get-Culture` or `Get-UICulture` command. Wildcard characters aren't permitted and you can't submit a

partial language code, such as de .

By default, `Update-Help` gets help files in the UI culture set for the operating system. If you specify the UICulture parameter, `Update-Help` looks for help

only for the specified UI culture.

Commands that use the UICulture parameter succeed only when the module provides help files for the specified UI culture. If the command fails because the

specified UI culture isn't supported, an error message is displayed.

Required?   false

Position?   2

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-UseDefaultCredentials <System.Management.Automation.SwitchParameter>

Indicates that `Update-Help` runs the command, including the internet download, using the credentials of the current user. By default, the command runs without

explicit credentials.

This parameter is effective only when the web download uses NT LAN Manager (NTLM), negotiate, or Kerberos-based authentication.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet isn't run.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.IO.DirectoryInfo

You can pipe a directory path object to this cmdlet.

System.Management.Automation.PSModuleInfo

You can pipe a module object to this cmdlet.

OUTPUTS

None

This cmdlet returns no output.

NOTES

To update help for the core PowerShell modules, that contain the commands that are installed with PowerShell, or any module in the `$PSHOME\Modules` directory,

start PowerShell with the option to Run as administrator .

Only members of the Administrators group on the computer can update help for the core PowerShell modules, the commands that are installed together with

PowerShell, and for modules in the `$PSHOME\Modules` folder. If you don't have permission to update help files, you can read the help files online. For example,

`Get-Help Update-Help -Online`.

Modules are the smallest unit of updatable help. You can't update help for a particular cmdlet. To find the module containing a particular cmdlet, use the

ModuleName property of the `Get-Command` cmdlet, for example, `(Get-Command Update-Help).ModuleName`.

Because help files are installed in the module directory, the `Update-Help` cmdlet can install updated help file only for modules that are installed on the

computer. However, the `Save-Help` cmdlet can save help for modules that aren't installed on the computer.

The `Update-Help` cmdlet was introduced in Windows PowerShell 3.0. It doesn't work in earlier versions of PowerShell. On computers that have both Windows

PowerShell 2.0 and Windows PowerShell 3.0, use the `Update-Help` cmdlet in a Windows PowerShell 3.0 session to download and update help files. The help files are

available to both Windows PowerShell 2.0 and Windows PowerShell 3.0.

The `Update-Help` and `Save-Help` cmdlets use the following ports to download help files: Port 80 for HTTP and port 443 for HTTPS.

`Update-Help` supports all modules and the core PowerShell snap-ins. It doesn't support any other snap-ins.

To update help for a module in a location that isn't listed in the `$env:PSModulePath` environment variable, import the module into the current session and then
run an `Update-Help` command. Run `Update-Help` without parameters or use the Module parameter to specify the module name. The Module parameter of the
`Update-Help` and `Save-Help` cmdlets doesn't accept the full path of a module file or module manifest file.

Any module can support Updatable Help. For instructions for supporting Updatable Help in the modules that you author, see Supporting Updatable Help
(/powershell/scripting/developer/module/supporting-updatable-help).

The `Update-Help` and `Save-Help` cmdlets aren't supported on Windows Preinstallation Environment (Windows PE).

--------- Example 1: Update help files for all modules ---------

Update-Help

------ Example 2: Update help files for specified modules ------

Update-Help -Module Microsoft.PowerShell*

Example 3: Updating help on a system not set to the en-US locale

Update-Help Microsoft.PowerShell.Utility -Force

Update-Help Microsoft.PowerShell.Utility -Force -UICulture en-GB

Update-Help : Failed to update Help for the module(s) 'Microsoft.PowerShell.Utility'

with UI culture(s) {en-GB} : The specified culture is not supported: en-GB. Specify a

culture from the following list: {en-US}.

At line:1 char:1

+ Update-Help Microsoft.PowerShell.Utility -Force -UICulture en-GB

+ ~~~~~~

    + CategoryInfo   : InvalidOperation: (:) [Update-Help], Exception

    + FullyQualifiedErrorId : HelpCultureNotSupported,Microsoft.PowerShell.Commands

   .UpdateHelpCommand

The help files are always published for the `en-US` locale. To download the English help, run `Update-Help` with the UICulture parameter and specify the `en-US`

locale.

---------- Example 4: Update help files automatically ----------


```
$jobParams = @{
  Name = 'UpdateHelpJob'
  Credential = 'Domain01\User01'
  ScriptBlock = '{Update-Help}'
  Trigger = (New-JobTrigger -Daily -At "3 AM")
}
Register-ScheduledJob @jobParams
```


Id	Name	JobTriggers	Command	Enabled
1	UpdateHelpJob	1	Update-Help	True

The `Register-ScheduledJob` cmdlet creates a scheduled job that runs an `Update-Help` command. The command uses the Credential parameter to run `Update-Help` by using

the credentials of a member of the Administrators group on the computer. The value of the Trigger parameter is a `New-JobTrigger` command that creates a job trigger

that starts the job every day at 3:00 AM.

To run the `Register-ScheduledJob` command, start PowerShell by using the Run as administrator option. PowerShell prompts you for the password of the user specified

in the Credential parameter. The credentials are stored with the scheduled job. You aren't prompted when the job runs.

You can use the `Get-ScheduledJob` cmdlet to view the scheduled job, use the `Set-ScheduledJob` cmdlet to change it, and use the `Unregister-ScheduledJob` cmdlet to

delete it. You can also view and manage the scheduled job in Task Scheduler in the following path:

`Task Scheduler Library\Microsoft\Windows\PowerShell\ScheduledJobs`.

Example 5: Update help files on multiple computers from a file share

```
Save-Help -DestinationPath \\Server01\Share\PSHelp -Credential Domain01\Admin01
Invoke-Command -ComputerName (Get-Content Servers.txt) -ScriptBlock {
    Update-Help -SourcePath \\Server01\Share\PSHelp -Credential Domain01\Admin01
}
```

The `Save-Help` command downloads the newest help files for all modules that support Updatable Help. The DestinationPath parameter saves the files in the

`\Server01\Share\PSHelp` file share. The Credential parameter specifies a user who has permission to access the file share.

The `Invoke-Command` cmdlet runs remote `Update-Help` commands on multiple computers. The ComputerName parameter gets a list of remote computers from the Servers.txt

file. The ScriptBlock parameter runs the `Update-Help` command and uses the SourcePath parameter to specify the file share containing the updated help files. The

Credential parameter specifies a user who can access the file share and run the remote `Update-Help` command.

--------- Example 6: Get a list of updated help files ---------

```
Update-Help -Module Microsoft.PowerShell.Utility -Verbose
```

----- Example 7: Find modules that support Updatable Help -----

Get-Module -ListAvailable | Where-Object -Property HelpInfoUri

Directory: C:\program files\powershell\6\Modules

ModuleType Version   Name   PSEdition ExportedCommands

---------- -------   ----   --------- ----------------

Manifest   6.1.0.0   CimCmdlets   Core   {Get-CimAssociatedInstance... }

Manifest   1.2.2.0   Microsoft.PowerShell.Archive   Desk   {Compress-Archive... }

Manifest   6.1.0.0   Microsoft.PowerShell.Diagnostics   Core   {Get-WinEvent, New-WinEvent}

   Directory: C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules

ModuleType Version   Name   PSEdition ExportedCommands

---------- -------   ----   --------- ----------------

Manifest   2.0.1.0   Appx   Core,Desk {Add-AppxPackage, ... }

Script   1.0.0.0   AssignedAccess   Core,Desk {Clear-AssignedAccess, ... }

Manifest   1.0.0.0   BitLocker   Core,Desk {Unlock-BitLocker, ... }

----------- Example 8: Inventory updated help files -----------

Get-UpdateHelpVersion.ps1

Param(

   [parameter(Mandatory=$False)]

   [String[]]

   $Module

)

$HelpInfoNamespace = @{helpInfo='http://schemas.microsoft.com/powershell/help/2010/05'}

if ($Module) { $Modules = Get-Module $Module -ListAvailable | where {$_.HelpInfoUri} }

else { $Modules = Get-Module -ListAvailable | where {$_.HelpInfoUri} }

foreach ($mModule in $Modules)

```
    {

        $mDir = $mModule.ModuleBase


        if (Test-Path $mdir\*helpinfo.xml)

        {

            $mName=$mModule.Name

            $mNodes = dir $mdir\*helpinfo.xml -ErrorAction SilentlyContinue |

                Select-Xml -Namespace $HelpInfoNamespace -XPath "//helpInfo:UICulture"

            foreach ($mNode in $mNodes)

            {

                $mCulture=$mNode.Node.UICultureName

                $mVer=$mNode.Node.UICultureVersion


                [PSCustomObject]@{"ModuleName"=$mName; "Culture"=$mCulture; "Version"=$mVer}

            }

        }

    }
```

```
ModuleName              Culture              Version

----------              -------              -------

ActiveDirectory         en-US                3.0.0.0

ADCSAdministration       en-US                3.0.0.0

ADCSDeployment          en-US                3.0.0.0

ADDSDeployment          en-US                3.0.0.0

ADFS                    en-US                3.0.0.0
```


RELATED LINKS

Online   Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.core/update-help?view=powershell-5.1&WT.mc_id=ps-g

ethelp

Get-Help

Get-Module

Get-UICulture

Start-Job

Save-Help