



Windows PowerShell Get-Help on Cmdlet 'New-IseSnippet'

PS:\>Get-HELP New-IseSnippet -Full

NAME

New-IseSnippet

SYNOPSIS

Creates a Windows PowerShell ISE code snippet.

SYNTAX

```
New-IseSnippet [-Title] <System.String> [-Description] <System.String> [-Text] <System.String> [-Author  
<System.String>] [-CaretOffset <System.Int32>] [-Force]  
[<CommonParameters>]
```

DESCRIPTION

The `New-IseSnippet` cmdlet creates a reusable text "snippet" for Windows PowerShell ISE. You can use snippets to add text to the Script pane or Command pane in

Windows PowerShell ISE. This cmdlet is available only in Windows PowerShell ISE.

Beginning in Windows PowerShell 3.0, Windows PowerShell ISE includes a collection of built-in snippets. The `New-IseSnippet` cmdlet lets you create your own snippets

to add to the built-in collection. You can view, change, add, delete, and share snippet files and include them in Windows PowerShell modules. To see snippets in

Windows PowerShell ISE, from the Edit menu, select Start Snippets or press <kbd>CTRL</kbd>+<kbd>J</kbd>.

The `New-ISESnippet` cmdlet creates a `<Title>.Snippets.ps1xml` file in the `$HOME\Documents\WindowsPowerShell\Snippets` directory with the title that you specify. To

include a snippet file in a module that you are authoring, add the snippet file to a Snippets subdirectory of your module directory.

You cannot use user-created snippets in a session in which the execution policy is Restricted or AllSigned .

This cmdlet was introduced in Windows PowerShell 3.0.

PARAMETERS

`-Author <System.String>`

Specifies the author of the snippet. The author field appears in the snippet file, but it does not appear when you click the snippet name in Windows PowerShell ISE.

Required?	false
-----------	-------

Position?	named
-----------	-------

Default value	None
---------------	------

Accept pipeline input?	False
------------------------	-------

Accept wildcard characters?	false
-----------------------------	-------

`-CaretOffset <System.Int32>`

Specifies the character of the snippet text that this cmdlet places the cursor on. Enter an integer that represents the cursor position, with "1" representing the

first character of text. The default value, 0 (zero), places the cursor immediately before the first character of text. This parameter does not indent the snippet

text.

Required?	false
Position?	named
Default value	0
Accept pipeline input?	False
Accept wildcard characters?	false

-Description <System.String>

Specifies a description of the snippet. The description value appears when you click the snippet name in Windows PowerShell ISE. This parameter is required.

Required?	true
Position?	2
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-Force <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet overwrites snippet files with the same name in the same location. By default, `New-ISESnippet` does not overwrite files.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

-Text <System.String>

Specifies the text value that is added when you select the snippet. The snippet text appears when you click the snippet name in Windows PowerShell ISE. This parameter is required.

Required?	true
Position?	3

Default value None
Accept pipeline input? False
Accept wildcard characters? false

-Title <System.String>

Specifies a title or name for the snippet. The title also names the snippet file. This parameter is required.

Required? true
Position? 1
Default value None
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

None
You can't pipe objects to this cmdlet.

OUTPUTS

None
This cmdlet returns no output.

NOTES

add them to a session in which the execution

policy is AllSigned or Restricted . In a Restricted or AllSigned session, you can create, get, and import unsigned user-created snippets, but you cannot use them in the session.

If you use the `New-IseSnippet` cmdlet in a Restricted or AllSigned session, the snippet is created, but an error message appears when Windows PowerShell tries to

add the newly created snippet to the session. To use the new snippet (and other unsigned user-created snippets), change the execution policy, and then restart

Windows PowerShell ISE.

For more information about Windows PowerShell execution policies, see `about_Execution_Policies` (`../Microsoft.PowerShell.Core/About/about_Execution_Policies.md`).

- To change a snippet, edit the snippet file. You can edit snippet files in the Script pane of Windows PowerShell ISE. -

To delete a snippet that you added, delete the snippet file.

- You cannot delete a built-in snippet, but you can hide all built-in snippets by using the

`"$psise.Options.ShowDefaultSnippets=$false"` command. - You can create a snippet that has the same name as a built-in snippet. Both snippets appear in the snippet menu in Windows PowerShell ISE.

----- Example 1: Create a Comment-Based help snippet -----

```
New-IseSnippet -Title Comment-BasedHelp -Description "A template for comment-based help." -Text "<#
```

```
.SYNOPSIS
```

```
.DESCRIPTION
```

```
.PARAMETER <Parameter-Name>
```

```
.INPUTS
```

```
.OUTPUTS
```

.EXAMPLE

.LINK

#>"

This command creates a Comment-BasedHelp snippet for Windows PowerShell ISE. It creates a file named `Comment-BasedHelp.snippets.ps1xml` in the user's Snippets

directory `\$HOME\Documents\WindowsPowerShell\Snippets`.

----- Example 2: Create a mandatory snippet -----

```
$M = @'
```

```
Param
```

```
(
```

```
[parameter(Mandatory=$true)]
```

```
[String[]]
```

```
$<ParameterName>
```

```
)
```

```
'@
```

```
New-ISESnippet -Text $M -Title Mandatory -Description "Adds a mandatory function parameter." -Author "Patti Fuller, Fabrikam Corp." -Force
```

This example creates a snippet named Mandatory for Windows PowerShell ISE. The first command saves the snippet text in the `\$M` variable. The second command uses the

`New-ISESnippet` cmdlet to create the snippet. The command uses the Force parameter to overwrite a previous snippet with the same name.

Example 3: Copy a mandatory snippet from a folder to a destination folder

```
Copy-Item "$HOME\Documents\WindowsPowerShell\Snippets\Mandatory.Snippets.ps1xml" -Destination "\\Server\Share"
```

This command uses the `Copy-Item` cmdlet to copy the Mandatory snippet from the folder where `New-ISESnippet` places it to the Server\Share file share.

https://learn.microsoft.com/powershell/module/ise/new-isesnippet?view=powershell-5.1&WT.mc_id=ps-gethelp

Get-IseSnippet

Import-IseSnippet