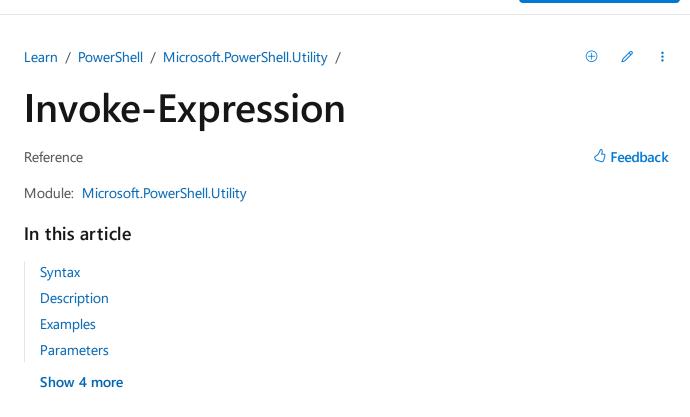


Download PDF



Sign in

Download PowerShell

Syntax

Runs commands or expressions on the local computer.



Description

The Invoke-Expression cmdlet evaluates or runs a specified string as a command and returns the results of the expression or command. Without Invoke-Expression, a string submitted at the command line is returned (echoed) unchanged.

Expressions are evaluated and run in the current scope. For more information, see about_Scopes.

⊗ Caution

Take reasonable precautions when using the Invoke-Expression cmdlet in scripts. When using Invoke-Expression to run a command that the user enters, verify that the command is safe to run before running it. In general, it is best to design your script with predefined input options, rather than allowing freeform input.

Examples

Example 1: Evaluate an expression



invoke-E	xpressio	n \$Command					
Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
296	4	1572	1956	20	0.53	1348	AdtAgent
270	6	1328	800	34	0.06	2396	alg
57	2	620	484	20	0.22	716	ati2evxx
1060	15	12904	11840	74	11.48	892	CcmExec
400	33	25280	37544	223	38.44	2564	communicator

This example demonstrates the use of Invoke-Expression to evaluate an expression. Without Invoke-Expression, the expression is printed, but not evaluated.

The first command assigns a value of Get-Process (a string) to the \$Command variable.

The second command shows the effect of typing the variable name at the command line. PowerShell echoes the string.

The third command uses Invoke-Expression to evaluate the string.

Example 2: Run a script on the local computer

```
PowerShell

Invoke-Expression -Command "C:\ps-test\testscript.ps1"

"C:\ps-test\testscript.ps1" | Invoke-Expression
```

These commands use Invoke-Expression to run a script, TestScript.ps1, on the local computer. The two commands are equivalent. The first uses the **Command** parameter to specify the command to run. The second uses a pipeline operator (|) to send the command string to Invoke-Expression.

Example 3: Run a command in a variable

```
PowerShell

$Command = 'Get-Process | where {$_.cpu -gt 1000}'
Invoke-Expression $Command
```

This example runs a command string that is saved in the \$Command variable.

The command string is enclosed in single quotation marks because it includes a variable, \$_, which represents the current object. If it were enclosed in double quotation marks, the \$_ variable would be replaced by its value before it was saved in the \$command variable.

Example 4: Get and run a cmdlet Help example

```
PowerShell

$Cmdlet_name = "Get-ComputerInfo"

$Example_number = 1

$Example_code = (Get-Help $Cmdlet_name).examples.example[($Example_number-1)].cc
Invoke-Expression $Example_code
```

This command retrieves and runs the first example in the Get-EventLog cmdlet Help topic.

To run an example of a different cmdlet, change the value of the \$Cmdlet_name variable to the name of the cmdlet. And, change the \$Example_number variable to the example number you want to run. The command fails if the example number is not valid.

① Note

If the example code from the help file has output in the example, PowerShell attempts to run the output along with the code and an error will be thrown.

Parameters

-Command

Specifies the command or expression to run. Type the command or expression or enter a variable that contains the command or expression. The **Command** parameter is required.

Expand table

Type:	String
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	True
Accept wildcard characters:	False

Inputs

String

You can pipe a string representing the expression to invoke to this cmdlet. Use the \$Input automatic variable to represent the input objects in the command.

PSObject

You can pipe an object representing the expression to invoke to this cmdlet. Use the \$Input automatic variable to represent the input objects in the command.

Outputs

None

This cmdlet returns no output of its own, but the invoked command may return output.

Notes

PowerShell includes the following aliases for Invoke-Expression:

- All platforms:
 - o iex

In most cases, you invoke expressions using PowerShell's call operator and achieve the same results. The call operator is a safer method. For more information, see about_Operators.

Related Links

- Invoke-Command
- about_Scopes

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see our contributor quide.



PowerShell feedback

PowerShell is an open source project. Select a link to provide feedback:

🖔 Open a documentation issue

Provide product feedback