



python



PowerShell

FPDF Library
PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Invoke-AsWorkflow'

PS C:\Users\wahid> Get-Help Invoke-AsWorkflow

NAME

Invoke-AsWorkflow

SYNOPSIS

Runs a command or expression as a Windows PowerShell Workflow.

SYNTAX

```
Invoke-AsWorkflow [-CommandName <System.String>] [-InputObject  
<System.Object>] [-Parameter <System.Collections.Hashtable>]  
[<CommonParameters>]
```

```
Invoke-AsWorkflow [-Expression <System.String>] [-InputObject <System.Object>]  
[<CommonParameters>]
```

DESCRIPTION

The `Invoke-AsWorkflow` workflow runs any command or expression as an inline script in a workflow. These workflows use the standard workflow semantics, have all workflow common parameters, and have all benefits of workflows, including the ability to stop, resume, and recover.

Workflows are designed for long-running commands that collect critical data, but can be used to run any command. For more information, see `about_Workflows` (`../PSWorkflow/About/about_Workflows.md`).

You can also add workflow common parameters to this command. For more information about workflow common parameters, see `about_WorkflowCommonParameters` (`../PSWorkflow/About/about_WorkflowCommonParameters.md`) This workflow is introduced in Windows PowerShell 3.0.

PARAMETERS

`-CommandName <System.String>`

Runs the specified cmdlet or advanced function as a workflow. Enter the cmdlet or function name, such as ``Update-Help``, ``Set-ExecutionPolicy``, or ``Set-NetFirewallRule``.

`-Expression <System.String>`

Specifies the expression that this cmdlet runs as a workflow. Enter the expression as a string, such as `"ipconfig /all"`. If the expression includes spaces or special characters, enclose the expression in quotation marks.

`-InputObject <System.Object>`

Used to allow pipeline input.

`-Parameter <System.Collections.Hashtable>`

Specifies the parameters and parameter values of the command that is specified in the ``CommandName`` parameter. Enter a hash table in which each key is a parameter name and its value is the parameter value, such as ``@{ExecutionPolicy="AllSigned"}``.

For information about hash tables, see `about_Hash_Tables` (`../Microsoft.PowerShell.Core/About/about_Hash_Tables.md`).

<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>).

----- Example 1: Run a cmdlet as a workflow -----

```
Invoke-AsWorkflow -PSComputerName (Get-Content Servers.txt) -CommandName  
Get-ExecutionPolicy
```

PSComputerName	PSSourceJobInstanceId
Value	
-----	-----

Server01	77b1cdf8-8226-4662-9067-cd2fa5c3b711
AllSigned	
Server02	a33542d7-3cdd-4339-ab99-0e7cd8e59462
Unrestricted	
Server03	279bac28-066a-4646-9497-8fcdcf9757e
AllSigned	
localhost	0d858009-2cc4-47a4-a2e0-da17dc2883d0
RemoteSigned	

This command runs the `Get-ExecutionPolicy` cmdlet as a workflow on hundreds of computers.

The command uses the `CommandName` parameter to specify the cmdlet that runs in the workflow. It uses the `PSComputerName` workflow common parameter to specify the computers on which the command runs. The value of the `PSComputerName`

parameter is a `Get-Content` command that gets a list of computer names from the Servers.txt file. The parameter value is enclosed in parentheses to direct Windows PowerShell to run the `Get-Command` command before using the value.

As with all remote commands, if the command runs on the local computer, (if the value of the PSComputerName parameter includes the local computer), you must start Windows PowerShell with the "Run as administrator" option.

----- Example 2: Run a cmdlet with parameters -----

```
$s = Import-Csv .\Servers.csv -Header ServerName, ServerID
Invoke-AsWorkflow -CommandName Get-ExecutionPolicy -Parameter
@{Scope="Process"} -PSComputerName {$s.ServerName} -PSConnectionRetryCount 5
```

The first command uses the `Import-Csv` cmdlet to create an object from the content in the Servers.csv file. The command uses the `Header` parameter to create a `ServerName` property for the column that contains the names of the target computers, also known as "remote nodes." The command saves the result in the `\$s` variable.

The second command uses the `Invoke-AsWorkflow` workflow to run a `Get-ExecutionPolicy` command on the computers in the Servers.csv file. The command uses the CommandName parameter of `Invoke-AsWorkflow` to specify the command to run in the workflow. It uses the `Parameter` parameter of `Invoke-AsWorkflow` to specify the `Scope` parameter of the `Get-ExecutionPolicy` cmdlet with a value of Process. The command also uses the `PSConnectionRetryCount` workflow common parameter to limit the command to five attempts on each computer and the `PSComputerName` workflow common parameter to specify the names of the remote nodes (target computers). The value of the `PSComputerName` parameter is an expression that gets the `ServerName` property of every object in the `\$s` variable.

These commands run a `Get-ExecutionPolicy` command as a workflow on hundreds of computers. The command uses the `Scope` parameter of the

`Get-ExecutionPolicy` cmdlet with a value of Process to get the execution policy in the current session.

----- Example 3: Run an expression as a workflow -----

```
Invoke-AsWorkflow -Expression "ipconfig /all" -PSComputerName (Get-Content DomainControllers.txt) -AsJob -JobName IPConfig
```

Id	Name	PSJobTypeName	State	HasMoreData	Location
	Command				
--	----	-----	----	-----	-----

2	IpConfig	PSWorkflowJob	Completed	True	Server01, Server01... Invoke-AsWorkflow

This command uses the `Invoke-AsWorkflow` workflow to run an Ipconfig command as a workflow job on the computers listed in the DomainControllers.txt file.

The command uses the `Expression` parameter to specify the expression to run. It uses the `PSComputerName` workflow common parameter to specify the names of the remote nodes (target computers).

The command also uses the `AsJob` and `JobName` workflow common parameters to run the workflow as a background job on each computer with the "Ipconfig" job name.

The command returns a `ContainerParentJob` object (`System.Management.Automation.ContainerParentJob`) that contains the workflow jobs on each computer.`

REMARKS

To see the examples, type: "get-help Invoke-AsWorkflow -examples".

For more information, type: "get-help Invoke-AsWorkflow -detailed".

For technical information, type: "get-help Invoke-AsWorkflow -full".

For online help, type: "get-help Invoke-AsWorkflow -online"

