Windows PowerShell Get-Help on Cmdlet 'Invoke-TroubleshootingPack'

PS:\>Get-HELP Invoke-TroubleshootingPack -Full

NAME

Invoke-TroubleshootingPack

SYNOPSIS

Runs a troubleshooting pack.

SYNTAX

Invoke-TroubleshootingPack [-Pack] <DiagPack> [-AnswerFile <String>] [-Result <String>] [-Unattended] [<CommonParameters>]

DESCRIPTION

The Invoke-TroubleshootingPack cmdlet runs a troubleshooting pack in either interactive or unattended mode. A troubleshooting pack determines the root causes of

issues, resolves the issues, and verifies that the issues were resolved. The cmdlet can save reports that detail issues and resolutions.

In interactive mode, you can select the resolutions to use and provide input to interactions with the troubleshooting pack. In unattended mode, the troubleshooting

pack determines which resolutions to use at run time. While in unattended mode, if the troubleshooting pack requires input, you need to provide answers or specify an

answer file. To create an answer file, use the Get-TroubleshootingPack cmdlet.

You can save the result report and the debug report, along with XSL and any linked files. Both reports contain the issues and resolutions. The debug report contains

additional information.

PARAMETERS

-AnswerFile <String>

Specifies a path for an answer file. You can use an absolute path, a relative path, or a Universal Naming Convention (UNC) path. To generate an answer file, use

the Get-TroubleshootingPack cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Pack <DiagPack>

Specifies a DiagPack object. The DiagPack object defines a troubleshooting pack. To obtain a DiagPack object, use the Get-TroubleshootingPack cmdlet.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Result <String>

Specifies a path for the result report and the debug report. You can use an absolute path, a relative path, or a

Universal Naming Convention (UNC) path.

If you do not use this parameter, the cmdlet does not save reports.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Unattended [<SwitchParameter>]

Indicates that the troubleshooting pack runs in unattended mode. If you specify this parameter and the troubleshooting pack requires input, specify an answer file

in the AnswerFile parameter.

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

Microsoft.Windows.Diagnosis.DiagPack

The troubleshooting pack to run. To obtain a DiagPack object, use the Get-TroubleshootingPack cmdlet.

OUTPUTS

NOTES

---------- Example 1: Run a troubleshooting pack ----------

PS C:\> Get-TroubleshootingPack -Path "C:\Windows\Diagnostics\System\Audio" | Invoke-TroubleshootingPack

This example runs the Audio pack in interactive mode. This example does not save reports.

The command uses the Get-TroubleshootingPack cmdlet to get a DiagPack object and pipes it to the Invoke-TroubleshootingPack cmdlet.

---------- Example 2: Save troubleshooting reports ----------

PS C:\> $Audio = Get-TroubleshootingPack -Path "C:\Windows\Diagnostics\System\Audio"

PS C:\> Invoke-TroubleshootingPack -Pack $Audio -Result "C:\DiagResult"

This example runs the Audio pack in interactive mode and saves the results to a folder.

The first command uses the Get-TroubleshootingPack cmdlet to get a DiagPack object and stores it in the $Audio variable.

The second command invokes the troubleshooting pack stored in $Audio. The pack saves reports in the specified folder.

--- Example 3: Run a troubleshooting pack in unattended mode ---

PS C:\> $Audio = Get-TroubleshootingPack -Path "C:\Windows\Diagnostics\System\Audio"

PS C:\> Invoke-TroubleshootingPack -Pack $Audio -AnswerFile "AudioAnswerFile.xml" -Unattended

This example runs the Audio pack in unattended mode with a specified answer file.

The first command uses the Get-TroubleshootingPack cmdlet to get a DiagPack object and stores it in the $Audio

variable.

The second command invokes the troubleshooting pack stored in $Audio in unattended mode. The command specifies an answer file, previously created by using the

Get-TroubleshootingPack cmdlet.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/troubleshootingpack/invoke-troubleshootingpack?view=windowsserver2022-ps&wt.mc_id=ps-gethelp

Get-TroubleshootingPack