Windows PowerShell Get-Help on Cmdlet 'Test-FileCatalog'

PS:\>Get-HELP Test-FileCatalog -Full

NAME

Test-FileCatalog

SYNOPSIS

`Test-FileCatalog` validates whether the hashes contained in a catalog file (.cat) matches the hashes of the actual files in order to validate their authenticity.

This cmdlet is only supported on Windows.

SYNTAX

Test-FileCatalog [-CatalogFilePath] <System.String> [[-Path] <System.String[]>] [-Detailed] [-FilesToSkip <System.String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

`Test-FileCatalog` validates the authenticity of files by comparing the file hashes of a catalog file (.cat) with the hashes of actual files on disk. If it detects

any mismatches, it returns the status as ValidationFailed. Users can retrieve all this information by using the -Detailed parameter. It also displays signing status

of catalog in Signature property which is equivalent to calling `Get-AuthenticodeSignature` cmdlet on the catalog file. Users can also skip any file during validation

by using the -FilesToSkip parameter.

This cmdlet is only supported on Windows.

PARAMETERS

-CatalogFilePath <System.String>

A path to a catalog file (.cat) that contains the hashes to be used for validation.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-Detailed <System.Management.Automation.SwitchParameter>

Returns more information a more detailed `CatalogInformation` object that contains the files tested, their expected/actual hashes, and an Authenticode signature

of the catalog file if it's signed.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-FilesToSkip <System.String[]>

An array of paths that should not be tested as part of the validation.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Path <System.String[]>

A folder or array of files that should be validated against the catalog file.

Required? false

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

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 is not 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 `DirectoryInfo` object representing the path to the files that need to be validated.

System.String

You can pipe a string representing the path to the files that need to be validated.

OUTPUTS

System.Management.Automation.CatalogValidationStatus

By default, this cmdlet returns a CatalogValidationStatus object with a value of either `Valid` or `ValidationFailed`.

System.Management.Automation.CatalogInformation

When you use the Detailed parameter, the cmdlet returns a CatalogInformation object for each file, which can be used to analyze specific files that may or may not

have passed validation, which hashes were expected vs. found, and the algorithm used in the catalog.

NOTES

-------- Example 1: Create and validate a file catalog --------

New-FileCatalog -Path $PSHOME\Modules\Microsoft.PowerShell.Utility -CatalogFilePath \temp\Microsoft.PowerShell.Utility.cat -CatalogVersion 2.0

Test-FileCatalog -CatalogFilePath \temp\Microsoft.PowerShell.Utility.cat -Path

"$PSHome\Modules\Microsoft.PowerShell.Utility\"

Valid

--- Example 2: Validate a file catalog with detailed output ---

Test-FileCatalog -Detailed -CatalogFilePath \temp\Microsoft.PowerShell.Utility.cat -Path "$PSHome\Modules\Microsoft.PowerShell.Utility\"

Status : Valid
HashAlgorithm : SHA256
CatalogItems : {[Microsoft.PowerShell.Utility.psd1, A7028BD54018AE519381CDF5BF91F3B0417BD9345478086089ACBFAD05C899FC], [Microsoft.PowerShell.Utility.psm1, 1127E8151FB86BCB683F932E8F6538552F7195816ED351A28AE07A753B8F20DE]}
PathItems : {[Microsoft.PowerShell.Utility.psd1, A7028BD54018AE519381CDF5BF91F3B0417BD9345478086089ACBFAD05C899FC], [Microsoft.PowerShell.Utility.psm1, 1127E8151FB86BCB683F932E8F6538552F7195816ED351A28AE07A753B8F20DE]}
Signature : System.Management.Automation.Signature

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.security/test-filecatalog?view=powershell-5.1&WT.mc_id=ps-gethelp

New-FileCatalog

PowerShellGet