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

PowerShell Get-Help on command 'Get-FileHash'

PS C:\Users\wahid> Get-Help Get-FileHash

NAME

    Get-FileHash

SYNOPSIS

    Computes the hash value for a file by using a specified hash algorithm.

SYNTAX

    Get-FileHash [-Algorithm {SHA1 | SHA256 | SHA384 | SHA512 | MACTripleDES | MD5
    | RIPEMD160}] -InputStream <System.IO.Stream> [<CommonParameters>]

    Get-FileHash [-Algorithm {SHA1 | SHA256 | SHA384 | SHA512 | MACTripleDES | MD5
    | RIPEMD160}] -LiteralPath <System.String[]> [<CommonParameters>]

    Get-FileHash [-Path] <System.String[]> [-Algorithm {SHA1 | SHA256 | SHA384 |
    SHA512 | MACTripleDES | MD5 | RIPEMD160}] [<CommonParameters>]

DESCRIPTION

    The `Get-FileHash` cmdlet computes the hash value for a file by using a
    specified hash algorithm. A hash value is a unique value that corresponds to

the content of the file. Rather than identifying the contents of a file by its file name, extension, or other designation, a hash assigns a unique value to the contents of a file. File names and extensions can be changed without altering the content of the file, and without changing the hash value. Similarly, the file's content can be changed without changing the name or extension. However, changing even a single character in the contents of a file changes the hash value of the file.

The purpose of hash values is to provide a cryptographically-secure way to verify that the contents of a file have not been changed. While some hash algorithms, including MD5 and SHA1, are no longer considered secure against attack, the goal of a secure hash algorithm is to render it impossible to change the contents of a file -- either by accident, or by malicious or unauthorized attempt -- and maintain the same hash value. You can also use hash values to determine if two different files have exactly the same content. If the hash values of two files are identical, the contents of the files are also identical.

By default, the `Get-FileHash` cmdlet uses the SHA256 algorithm, although any hash algorithm that is supported by the target operating system can be used.

PARAMETERS

-Algorithm <System.String>

Specifies the cryptographic hash function to use for computing the hash value of the contents of the specified file or stream. A cryptographic hash function has the property that it is infeasible to find two different files with the same hash value. Hash functions are commonly used with digital signatures and for data integrity. The acceptable values for this parameter are:

- SHA1

- SHA256

- SHA384

- SHA512

- MACTripleDES

- MD5

- RIPEMD160

If no value is specified, or if the parameter is omitted, the default value is SHA256.
For security reasons, MD5 and SHA1, which are no longer considered secure, should only be used for simple change validation, and should not be used to generate hash values for files that require protection from attack or tampering.

-InputStream <System.IO.Stream>
Specifies the input stream.

-LiteralPath <System.String[]>
Specifies the path to a file. Unlike the Path parameter, the value of the LiteralPath parameter is used exactly as it is typed. No characters are interpreted as wildcard characters. If the path includes escape characters, enclose the path in single quotation marks. Single quotation marks instruct PowerShell not to interpret characters as escape sequences.

-Path <System.String[]>
Specifies the path to one or more files as an array. Wildcard characters

are permitted.

<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: Compute the hash value for a file ---------

```
Get-FileHash $PSHOME\powershell.exe | Format-List


Algorithm : SHA256
Hash      : 908B64B1971A979C7E3E8CE4621945CBA84854CB98D76367B791A6E22B5F6D53
Path      : C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

------ Example 2: Compute the hash value for an ISO file ------

```
Get-FileHash C:\Users\user1\Downloads\Contoso8_1_ENT.iso -Algorithm SHA384 |
Format-List


Algorithm : SHA384
Hash      : 20AB1C2EE19FC96A7C66E33917D191A24E3CE9DAC99DB7C786ACCE31E559144FEAF
C695C58E508E2EBBC9D3C96F21FA3
Path      : C:\Users\user1\Downloads\Contoso8_1_ENT.iso
```

-------- Example 3: Compute the hash value of a stream --------

```
$wc = [System.Net.WebClient]::new()
$pkgurl = 'https://github.com/PowerShell/PowerShell/releases/download/v6.2.4/po
wershell_6.2.4-1.debian.9_amd64.deb'
```

```
$publishedHash =

'8E28E54D601F0751922DE24632C1E716B4684876255CF82304A9B19E89A9CCAC'

$FileHash = Get-FileHash -InputStream ($wc.OpenRead($pkgurl))

$FileHash.Hash -eq $publishedHash


True
```

----------- Example 4: Compute the hash of a string -----------

```
$stringAsStream = [System.IO.MemoryStream]::new()

$writer = [System.IO.StreamWriter]::new($stringAsStream)

$writer.write("Hello world")

$writer.Flush()

$stringAsStream.Position = 0

Get-FileHash -InputStream $stringAsStream | Select-Object Hash


Hash

----

64EC88CA00B268E5BA1A35678A1B5316D212F4F366B2477232534A8AECA37F3C
```

REMARKS

To see the examples, type: "get-help Get-FileHash -examples".

For more information, type: "get-help Get-FileHash -detailed".

For technical information, type: "get-help Get-FileHash -full".

For online help, type: "get-help Get-FileHash -online"