PowerShell Get-Help on command 'Format-Hex'

PS C:\Users\wahid> Get-Help Format-Hex

NAME

Format-Hex

SYNOPSIS

Displays a file or other input as hexadecimal.

SYNTAX

```
Format-Hex [-Encoding {ASCII | BigEndianUnicode | Unicode | UTF7 | UTF8 |
UTF32}] -InputObject <System.Object> [-Raw] [<CommonParameters>]

Format-Hex -LiteralPath <System.String[]> [<CommonParameters>]

Format-Hex [-Path] <System.String[]> [<CommonParameters>]
```

DESCRIPTION

The `Format-Hex` cmdlet displays a file or other input as hexadecimal values.
To determine the offset of a character from the output, add the number at the
leftmost of the row to the number at the top of the column for that character.

The `Format-Hex` cmdlet can help you determine the file type of a corrupted file or a file that might not have a filename extension. You can run this cmdlet, and then read the hexadecimal output to get file information.

When using `Format-Hex` on a file, the cmdlet ignores newline characters and returns the entire contents of a file in one string with the newline characters preserved.

PARAMETERS

-Encoding <System.String>

Specifies the encoding of the output. This only applies to `[string]` input. The parameter has no effect on numeric types. The default value is `ASCII`.

The acceptable values for this parameter are as follows:

- `Ascii` Uses ASCII (7-bit) character set.

- `BigEndianUnicode` Uses UTF-16 with the big-endian byte order.

- `Unicode` Uses UTF-16 with the little-endian byte order.

- `UTF7` Uses UTF-7.

- `UTF8` Uses UTF-8.

- `UTF32` Uses UTF-32 with the little-endian byte order.

Non-ASCII characters in the input are output as literal `?` characters resulting in a loss of information.

-InputObject <System.Object>

Specifies the objects to be formatted. Enter a variable that contains the objects or type a command or expression that gets the objects.

Only certain scalar (/powershell/scripting/learn/glossary#scalar-value)types and `[system.io.fileinfo]` are supported.

The supported scalar types are:

- `[string]`

- `[byte]`

- `[int]`, `[int32]`

- `[long]`, `[int64]`

-LiteralPath <System.String[]>

Specifies the complete path to a file. The value of LiteralPath is used exactly as it is typed. This parameter does not accept wildcard characters. To specify multiple paths to files, separate the paths with a comma. If the LiteralPath parameter includes escape characters, enclose the path in single quotation marks. PowerShell does not interpret any characters in a single quoted string as escape sequences. For more information, see about_Quoting_Rules (../Microsoft.Powershell.Core/About/about_Quoting_Rules.md).

-Path <System.String[]>

Specifies the path to files. Use a dot (`.`) to specify the current location. The wildcard character (` `) is accepted and can be used to specify all the items in a location. If the Path * parameter includes

    escape characters, enclose the path in single quotation marks. To specify
    multiple paths to files, separate the paths with a comma.

-Raw <System.Management.Automation.SwitchParameter>
    By default `Format-Hex` opts for compact output of numeric data types:
    single-byte or double-byte sequences are used if the value is small
    enough. The Raw parameter deactivates this behavior.

<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: Get the hexadecimal representation of a string --

```
'Hello World' | Format-Hex

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   48 65 6C 6C 6F 20 57 6F 72 6C 64                Hello World
```

The string Hello World is sent down the pipeline to the `Format-Hex` cmdlet.
The hexadecimal output from `Format-Hex` shows the values of each character in
the string.
----- Example 2: Find a file type from hexadecimal output -----

```
Format-Hex -Path .\File.t7f

Path: C:\Test\File.t7f

           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```

```
00000000   25 50 44 46 2D 31 2E 35 0D 0A 25 B5 B5 B5 B5 0D  %PDF-1.5..%????.
00000010   0A 31 20 30 20 6F 62 6A 0D 0A 3C 3C 2F 54 79 70  .1 0 obj..<</Typ
00000020   65 2F 43 61 74 61 6C 6F 67 2F 50 61 67 65 73 20  e/Catalog/Pages
```

The `Format-Hex` cmdlet uses the Path parameter to specify a filename in the current directory, `File.t7f`. The file extension `.t7f` is uncommon, but the hexadecimal output `%PDF` shows that it is a PDF file.

---------- Example 3: Display raw hexadecimal output ----------

```
PS> 1,2,3,1000 | Format-Hex

           Path:

           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   01 02 03 E8 03                                   ...?.



PS> 1,2,3,1000 | Format-Hex -Raw

           Path:

           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   01 00 00 00 02 00 00 00 03 00 00 00 E8 03 00 00  ............?...
```

Notice the difference in output. The Raw parameter displays the numbers as 4-byte values, true to their Int32 types.

REMARKS

To see the examples, type: "get-help Format-Hex -examples".

For more information, type: "get-help Format-Hex -detailed".

For technical information, type: "get-help Format-Hex -full".

For online help, type: "get-help Format-Hex -online"