



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

Windows PowerShell Get-Help on Cmdlet 'Get-Host'

PS:\>Get-HELP Get-Host -Full

NAME

Get-Host

SYNOPSIS

Gets an object that represents the current host program.

SYNTAX

Get-Host [<CommonParameters>]

DESCRIPTION

The `Get-Host` cmdlet gets an object that represents the program that is hosting Windows PowerShell.

The default display includes the Windows PowerShell version number and the current region and language settings that the host is using, but the host object contains a

wealth of information, including detailed information about the version of Windows PowerShell that is currently running and the current culture and UI culture of

Windows PowerShell. You can also use this cmdlet to customize features of the host program user interface, such as the text and background colors.

PARAMETERS

<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) (https://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

None

You can't pipe objects to this cmdlet.

OUTPUTS

System.Management.Automation.Internal.Host.InternalHost

This cmdlet returns an InternalHost object.

NOTES

The ``$Host`` automatic variable contains the same object that ``Get-Host`` returns, and you can use it in the same way. Similarly, the ``$PSCulture`` and ``$PSUICulture`` automatic variables contain the same objects that the `CurrentCulture` and `CurrentUICulture` properties of the host object contain. You can use these features interchangeably.

For more information, see [about_Automatic_Variables](#) (../Microsoft.PowerShell.Core/About/about_Automatic_Variables.md).

- Example 1: Get information about the PowerShell console host -

Get-Host

```
Name       : ConsoleHost
Version    : 2.0
Instanceld : e4e0ab54-cc5e-4261-9117-4081f20ce7a2
UI         : System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture : en-US
CurrentUICulture : en-US
PrivateData : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
IsRunspacePushed : False
Runspace   : System.Management.Automation.Runspaces.LocalRunspace
```

This command displays information about the PowerShell console, which is the current host program for PowerShell in this example. It includes the name of the host, the version of PowerShell that is running in the host, and current culture and UI culture.

The Version , UI , CurrentCulture , CurrentUICulture , PrivateData , and Runspace properties each contain an object with other useful properties. Later examples examine these properties.

----- Example 2: Resize the PowerShell window -----

```
$H = Get-Host
$Win = $H.UI.RawUI.WindowSize
$Win.Height = 10
$Win.Width = 10
$H.UI.RawUI.Set_WindowSize($Win)
```

This command resizes the Windows PowerShell window to 10 lines by 10 characters.

----- Example 3: Get the PowerShell version for the host -----

```
(Get-Host).Version
```

5 1 22621 963

This command gets detailed information about the version of Windows PowerShell running in the host. You can view, but not change, these values.

The Version property of `Get-Host` contains a System.Version object. This command uses a pipeline operator (|) to send the version object to the `Format-List`

cmdlet. The `Format-List` command uses the Property parameter with a value of all (*) to display all of the properties and property values of the version object.

----- Example 4: Get the current culture for the host -----

(Get-Host).CurrentCulture | Format-List

```
Parent           : en
LCID              : 1033
KeyboardLayoutId : 1033
Name              : en-US
IetfLanguageTag  : en-US
DisplayName       : English (United States)
NativeName       : English (United States)
EnglishName      : English (United States)
TwoLetterISOLanguageName : en
ThreeLetterISOLanguageName : eng
ThreeLetterWindowsLanguageName : ENU
CompareInfo      : CompareInfo - en-US
TextInfo         : TextInfo - en-US
IsNeutralCulture : False
CultureTypes     : SpecificCultures, InstalledWin32Cultures
NumberFormat     : System.Globalization.NumberFormatInfo
DateTimeFormat   : System.Globalization.DateTimeFormatInfo
Calendar         : System.Globalization.GregorianCalendar
OptionalCalendars : {System.Globalization.GregorianCalendar}
```

UseUserOverride : True
IsReadOnly : True

This command gets detailed information about the current culture set for Windows PowerShell running in the host. This is the same information that is returned by the

``Get-Culture` cmdlet.`

Similarly, the `CurrentUICulture` property returns the same object that ``Get-UICulture`` returns.

The `CurrentCulture` property of the host object contains a `System.Globalization.CultureInfo` object. This command uses a pipeline operator (`|`) to send the `CultureInfo`

object to the ``Format-List`` cmdlet. The ``Format-List`` command uses the `Property` parameter with a value of `all`` to display all of the properties and property

values of the `CultureInfo * object.`

-- Example 5: Get the `DateTimeFormat` for the current culture --

```
(Get-Host).CurrentCulture.DateTimeFormat | Format-List
```

```
AMDesignator      : AM  
Calendar          : System.Globalization.GregorianCalendar  
DateSeparator     : /  
FirstDayOfWeek    : Sunday  
CalendarWeekRule  : FirstDay  
FullDateTimePattern : dddd, MMMM dd, yyyy h:mm:ss tt  
LongDatePattern   : dddd, MMMM dd, yyyy  
LongTimePattern   : h:mm:ss tt  
MonthDayPattern   : MMMM dd  
PMDesignator      : PM  
RFC1123Pattern    : ddd, dd MMM yyyy HH':'mm':'ss 'GMT'  
ShortDatePattern  : M/d/yyyy  
ShortTimePattern  : h:mm tt  
SortableDateTimePattern : yyyy'-MM'-'dd'T'HH':'mm':'ss  
TimeSeparator     : :
```

```

UniversalSortableDateTimePattern : yyyy'-MM'-'dd HH':'mm':'ss'Z'
YearMonthPattern                 : MMMM, yyyy
AbbreviatedDayNames              : {Sun, Mon, Tue, Wed...}
ShortestDayNames                 : {Su, Mo, Tu, We...}
DayNames                         : {Sunday, Monday, Tuesday, Wednesday...}
AbbreviatedMonthNames            : {Jan, Feb, Mar, Apr...}
MonthNames                       : {January, February, March, April...}
IsReadOnly                       : False
NativeCalendarName               : Gregorian Calendar
AbbreviatedMonthGenitiveNames    : {Jan, Feb, Mar, Apr...}
MonthGenitiveNames               : {January, February, March, April...}

```

This command returns detailed information about the `DateTimeFormat` of the current culture that is being used for Windows PowerShell.

The `CurrentCulture` property of the host object contains a `CultureInfo` object that, in turn, has many useful properties. Among them, the `DateTimeFormat` property contains a `DateTimeFormatInfo` object with many useful properties.

To find the type of an object that is stored in an object property, use the ``Get-Member`` cmdlet. To display the property values of the object, use the ``Format-List`` cmdlet.

----- Example 6: Get the `RawUI` property for the host -----

```
(Get-Host).UI.RawUI
```

```

ForegroundColor   : Gray
BackgroundColor   : Black
CursorPosition    : 0,28
WindowPosition    : 0,0
CursorSize        : 25
BufferSize        : 120,29
WindowSize        : 120,29

```

MaxWindowSize : 120,29

MaxPhysicalWindowSize : 1904,69

KeyAvailable : True

WindowTitle : PowerShell 7.3.3

Example 7: Set the background color for the PowerShell console

```
(Get-Host).UI.RawUI.BackgroundColor = "Black"
```

```
Clear-Host
```

This change is effective only in the current session. To change the background color of the console for all sessions, add the command to your PowerShell profile.

---- Example 8: Set the background color for error messages ----

```
$Host.PrivateData.ErrorBackgroundColor = "white"
```

This command changes the background color of error messages to white.

This command uses the ``$Host`` automatic variable, which contains the host object for the current host program. ``Get-Host`` returns the same object that ``$Host`` contains, so you can use them interchangeably.

This command uses the `PrivateData` property of ``$Host`` as its `ErrorBackgroundColor` property. To see all of the properties of the object in the ``$Host`.PrivateData` property, type ``$host.PrivateData | format-list *``.

RELATED LINKS

Online

Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/get-host?view=powershell-5.1&WT.mc_id=ps-gethelp

Clear-Host

Read-Host

