PowerShell Get-Help on command 'Get-TypeData'

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

NAME

Get-TypeData

SYNOPSIS

Gets the extended type data in the current session.

SYNTAX

Get-TypeData [[-TypeName] <System.String[]>] [<CommonParameters>]

DESCRIPTION

The `Get-TypeData` cmdlet gets the extended type data in the current session.

This includes type data that was added to the session by `Types.ps1xml` file

and dynamic type data that was added by using the parameter of the

`Update-TypeData` cmdlet.

You can use the extended type data that `Get-TypeData` returns to examine the

type data in the session and send it to the `Update-TypeData` and

`Remove-TypeData` cmdlets.

Extended type data adds properties and methods to objects in PowerShell. You can use the added properties and methods in the same ways that you would use the properties and methods that are defined in the object type. However, when writing scripts, be aware that the added properties and methods might not be present in every PowerShell session.

For more information about `Types.ps1xml` files, see about_Types.ps1xml (../Microsoft.PowerShell.Core/About/about_Types.ps1xml.md). For more information about dynamic type data that the `Update-TypeData` cmdlet adds, see `Update-TypeData`.

This cmdlet was introduced in Windows PowerShell 3.0.

PARAMETERS

  -TypeName <System.String[]>

    Specifies type data as an array only for the types with the specified names. By default, `Get-TypeData` gets all types in the session.

    Enter type names or a name patterns. Full names, or name patterns with wildcard characters are required, even for types in the System namespace. Wildcards are supported and the parameter name TypeName is optional. You can also pipe type names to `Get-TypeData`.

  <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 all extended type data ------------

--------------- Example 2: Get type data by name ---------------

Get-TypeData -TypeName System.IO.*

TypeName	Members
System.IO.DirectoryInfo	{[Mode, System.Management.Automation.Runspaces.CodePropert.
System.IO.FileInfo	{[Mode, System.Management.Automation.Runspaces.CodePropert.

Example 3: Get the script block that creates a property value

(Get-TypeData *EventLogEntry*).Members.EventID

GetScriptBlock	SetScriptBlock	IsHidden	Name
$this.get_EventID() -band 0xFFFF		False	EventID

Example 4: Get the script block that defines a property for a specified object

```
(Get-TypeData -TypeName System.DateTime).Members["DateTime"].GetScriptBlock
if ((& { Set-StrictMode -Version 1; $this.DisplayHint }) -ieq  "Date") {
    "{0}" -f $this.ToLongDateString()
}
elseif ((& { Set-StrictMode -Version 1; $this.DisplayHint }) -ieq "Time") {
    "{0}" -f  $this.ToLongTimeString()
}
else {
```

```
    "{0} {1}" -f $this.ToLongDateString(), $this.ToLongTimeString()
}
```

The command uses the `Get-TypeData` cmdlet to get the extended type data for the System.DataTime type. The command gets the Members property of the TypeData object.

The Members property contains a hash table of properties and methods that are defined by extended type data. Each key in the Members hash table is a property or method name and each value is the definition of the property or method value.

The command gets the DateTime key in Members and its GetScriptBlock property value.

The output shows the script block that creates the value of the DateTime property of every System.DateTime object in PowerShell.

REMARKS

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

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

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

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