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

PowerShell Get-Help on command 'Get-Item'

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

NAME

Get-Item

SYNOPSIS

Gets the item at the specified location.

SYNTAX

Get-Item [-CodeSigningCert] [-Credential

<System.Management.Automation.PSCredential>] [-DnsName

<Microsoft.PowerShell.Commands.DnsNameRepresentation>]

[-DocumentEncryptionCert] [-Eku <System.String>] [-Exclude <System.String[]>]

[-ExpiringInDays <System.Int32>] [-Filter <System.String>] [-Force] [-Include

<System.String[]>] -LiteralPath <System.String[]> [-SSLServerAuthentication]

[-Stream <System.String[]>] [-UseTransaction] [<CommonParameters>]

Get-Item [-Path] <System.String[]> [-CodeSigningCert] [-Credential

<System.Management.Automation.PSCredential>] [-DnsName

<Microsoft.PowerShell.Commands.DnsNameRepresentation>]

[-DocumentEncryptionCert] [-Eku <System.String>] [-Exclude <System.String[]>]

[-ExpiringInDays <System.Int32>] [-Filter <System.String>] [-Force] [-Include

<System.String[]>] [-SSLServerAuthentication] [-Stream <System.String[]>]

[-UseTransaction] [<CommonParameters>]

DESCRIPTION

The `Get-Item` cmdlet gets the item at the specified location. It doesn't get

the contents of the item at the location unless you use a wildcard character

(`*`) to request all the contents of the item.

This cmdlet is used by PowerShell providers to navigate through different

types of data stores. Some parameters are only available for a specific

provider. For more information, see about_Providers

(../Microsoft.PowerShell.Core/About/about_Providers.md).

PARAMETERS

-CodeSigningCert <System.Management.Automation.SwitchParameter>

This is a dynamic parameter made available by the Certificate provider.

To get certificates that have `Code Signing` in their EnhancedKeyUsageList

property value, use the CodeSigningCert parameter.

For more information, see about_Certificate_Provider

(../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

-Credential <System.Management.Automation.PSCredential>

> [!NOTE] > This parameter isn't supported by any providers installed with

PowerShell. > To impersonate another user, or elevate your credentials

when running this cmdlet, > use Invoke-Command

(../Microsoft.PowerShell.Core/Invoke-Command.md).

-DnsName <Microsoft.PowerShell.Commands.DnsNameRepresentation>

This is a dynamic parameter made available by the Certificate provider.

Specifies a domain name or name pattern to match with the DNSNameList
property of certificates the cmdlet gets. The value of this parameter can
either be `Unicode` or `ASCII`. Punycode values are converted to Unicode.
Wildcard characters (`*`) are permitted.

This parameter was reintroduced in PowerShell 7.1

For more information, see about_Certificate_Provider
(../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

-DocumentEncryptionCert <System.Management.Automation.SwitchParameter>
    This is a dynamic parameter made available by the Certificate provider.

    To get certificates that have `Document Encryption` in their
    EnhancedKeyUsageList property value, use the DocumentEncryptionCert
    parameter.

    For more information, see about_Certificate_Provider
    (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

-Eku <System.String>
    This is a dynamic parameter made available by the Certificate provider.

    Specifies text or a text pattern to match with the EnhancedKeyUsageList
    property of certificates the cmdlet gets. Wildcard characters (` `) are
    permitted. The EnhancedKeyUsageList * property contains the friendly name
    and the OID fields of the EKU.

    This parameter was reintroduced in PowerShell 7.1

    For more information, see about_Certificate_Provider
    (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

-Exclude <System.String[]>

  Specifies, as a string array, an item or items that this cmdlet excludes

  in the operation. The value of this parameter qualifies the Path

  parameter. Enter a path element or pattern, such as ` .txt`. Wildcard

  characters are permitted. The Exclude * parameter is effective only when

  the command includes the contents of an item, such as `C:\Windows*`, where

  the wildcard character specifies the contents of the `C:\Windows`

  directory.

-ExpiringInDays <System.Int32>

  This is a dynamic parameter made available by the Certificate provider.

  Specifies that the cmdlet should only return certificates that are

  expiring in or before the specified number of days. A value of zero (`0`)

  gets certificates that have expired.

  This parameter was reintroduced in PowerShell 7.1

  For more information, see about_Certificate_Provider

  (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

-Filter <System.String>

  Specifies a filter to qualify the Path parameter. The FileSystem

  (../Microsoft.PowerShell.Core/About/about_FileSystem_Provider.md)provider

  is the only installed PowerShell provider that supports filters. Filters

  are more efficient than other parameters. The provider applies filter when

  the cmdlet gets the objects rather than having PowerShell filter the

  objects after they're retrieved. The filter string is passed to the .NET

  API to enumerate files. The API only supports `*` and `?` wildcards.

-Force <System.Management.Automation.SwitchParameter>

  Indicates that this cmdlet gets items that can't otherwise be accessed,

such as hidden items. Implementation varies from provider to provider. For

more information, see about_Providers

(../Microsoft.PowerShell.Core/About/about_Providers.md). Even using the

Force parameter, the cmdlet can't override security restrictions.

-Include <System.String[]>

Specifies, as a string array, an item or items that this cmdlet includes

in the operation. The value of this parameter qualifies the Path

parameter. Enter a path element or pattern, such as ` .txt`. Wildcard

characters are permitted. The Include * parameter is effective only when

the command includes the contents of an item, such as `C:\Windows*`, where

the wildcard character specifies the contents of the `C:\Windows`

directory.

-LiteralPath <System.String[]>

Specifies a path to one or more locations. The value of LiteralPath is

used exactly as it's typed. No characters are interpreted as wildcards. If

the path includes escape characters, enclose it in single quotation marks.

Single quotation marks tell PowerShell not to interpret any characters 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 an item. This cmdlet gets the item at the specified

location. Wildcard characters are permitted. This parameter is required,

but the parameter name Path is optional.

Use a dot (`.`) to specify the current location. Use the wildcard

character (`*`) to specify all the items in the current location.

-SSLServerAuthentication <System.Management.Automation.SwitchParameter>   _Page 5/8_

This is a dynamic parameter made available by the Certificate provider.

To get certificates that have `Server Authentication` in their
EnhancedKeyUsageList property value, use the SSLServerAuthentication
parameter.

For more information, see about_Certificate_Provider
(../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

-Stream <System.String[]>
This is a dynamic parameter made available by the FileSystem provider.

Gets the specified alternate NTFS file stream from the file. Enter the
stream name. Wildcards are supported. To get all streams, use an asterisk
(`*`). This parameter isn't valid on folders.

This parameter was introduced in PowerShell 3.0.

For more information, see about_FileSystem_Provider
(../Microsoft.PowerShell.Core/About/about_FileSystem_Provider.md).

-UseTransaction <System.Management.Automation.SwitchParameter>
Includes the command in the active transaction. This parameter is valid
only when a transaction is in progress. For more information, see
about_Transactions
(../Microsoft.PowerShell.Core/About/about_Transactions.md).

<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 current directory ------------

Get-Item .

Directory: C:\

```
Mode            LastWriteTime     Length Name
----            -------------     ------ ----
d----       7/26/2006  10:01 AM          ps-test
```

---- Example 2: Get all the items in the current directory ----

Get-Item *

Directory: C:\ps-test

```
Mode            LastWriteTime     Length Name
----            -------------     ------ ----
d----       7/26/2006   9:29 AM          Logs
d----       7/26/2006   9:26 AM          Recs
-a---       7/26/2006   9:28 AM       80 date.csv
-a---       7/26/2006  10:01 AM       30 filenoext
-a---       7/26/2006   9:30 AM    11472 process.doc
-a---       7/14/2006  10:47 AM       30 test.txt
```

------- Example 3: Get the current directory of a drive -------

Get-Item C:

--------- Example 4: Get items in the specified drive ---------

Get-Item C:*

In PowerShell, use a single asterisk (` `) to get contents, instead of the
traditional ` .*`. The format is interpreted literally, so ` . ` wouldn't
retrieve directories or filenames without a dot.
----- Example 5: Get a property in the specified directory -----

(Get-Item C:\Windows).LastAccessTime

-------- Example 6: Show the contents of a registry key --------

Get-Item HKLM:\Software\Microsoft\Powershell\1\Shellids\Microsoft.Powershell\

-- Example 7: Get items in a directory that have an exclusion --

Get-Item C:\Windows*.* -Exclude "w*"

REMARKS

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

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

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

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