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

PowerShell Get-Help on command 'Move-Item'

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

```
NAME
    Move-Item


SYNOPSIS
    Moves an item from one location to another.



SYNTAX
    Move-Item [[-Destination] <System.String>] [-Credential
    <System.Management.Automation.PSCredential>] [-Exclude <System.String[]>]
    [-Filter <System.String>] [-Force] [-Include <System.String[]>] -LiteralPath
    <System.String[]> [-PassThru] [-UseTransaction] [-Confirm] [-WhatIf]
    [<CommonParameters>]

    Move-Item [-Path] <System.String[]> [[-Destination] <System.String>]
    [-Credential <System.Management.Automation.PSCredential>] [-Exclude
    <System.String[]>] [-Filter <System.String>] [-Force] [-Include
    <System.String[]>] [-PassThru] [-UseTransaction] [-Confirm] [-WhatIf]
    [<CommonParameters>]
```

DESCRIPTION

The `Move-Item` cmdlet moves an item, including its properties, contents, and child items, from one location to another location. The locations must be supported by the same provider.

For example, it can move a file or subdirectory from one directory to another or move a registry subkey from one key to another. When you move an item, it is added to the new location and deleted from its original location.

PARAMETERS

-Credential <System.Management.Automation.PSCredential>

> [!NOTE] > This parameter is not 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).

-Destination <System.String>

Specifies the path to the location where the items are being moved. The default is the current directory. Wildcards are permitted, but the result must resolve to a single location.

To rename the item being moved, specify a new name in the value of the Destination parameter.

-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.

-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 the use of filters. You can find the syntax for the FileSystem filter language in about_Wildcards (../Microsoft.PowerShell.Core/About/about_Wildcards.md). Filters are more efficient than other parameters, because the provider applies them when the cmdlet gets the objects rather than having PowerShell filter the objects after they are retrieved.

-Force <System.Management.Automation.SwitchParameter>

Forces the command to run without asking for user confirmation. Implementation varies from provider to provider. For more information, see about_Providers (../Microsoft.PowerShell.Core/About/about_Providers.md).

-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 the path to the current location of the items. The value of LiteralPath is used exactly as it is 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).

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object representing the moved item. By default, this cmdlet does not generate any output.

-Path <System.String[]>

Specifies the path to the current location of the items. The default is the current directory. Wildcard characters are permitted.

-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).

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

<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: Move a file to another directory and rename it --

```
Move-Item -Path C:\test.txt -Destination E:\Temp\tst.txt
```

Example 2: Move a directory and its contents to another directory

```
Move-Item -Path C:\Temp -Destination C:\Logs
```

Example 3: Move all files of a specified extension from the current directory to another directory

```
Move-Item -Path .\*.txt -Destination C:\Logs
```

Example 4: Recursively move all files of a specified extension from the current directory to another directory

```
Get-ChildItem -Path ".\*.txt" -Recurse | Move-Item -Destination "C:\TextFiles"
```

The command uses the `Get-ChildItem` cmdlet to get all of the child items in the current directory (represented by the dot (`.`)) and its subdirectories that have a `*.txt` file name extension. It uses the Recurse parameter to make the retrieval recursive and the Include parameter to limit the retrieval to `*.txt` files.

The pipeline operator (`|`) sends the results of this command to `Move-Item`, which moves the text files to the `TextFiles` directory.

If files that are to be moved to `C:\Textfiles` have the same name, `Move-Item` displays an error and continues, but it moves only one file with each name to `C:\Textfiles`. The other files remain in their original directories.

If the `Textfiles` directory (or any other element of the destination path) does not exist, the command fails. The missing directory is not created for you, even if you use the Force parameter. `Move-Item` moves the first item to a file called `Textfiles` and then displays an error explaining that the file already exists.

Also, by default, `Get-ChildItem` does not move hidden files. To move hidden files, use the Force parameter with `Get-ChildItem`.

> [!NOTE] > In Windows PowerShell 2.0, when using the Recurse parameter of the `Get-ChildItem` cmdlet, the > value of the Path parameter must be a container. Use the Include parameter to specify the > `*.txt` file name extension filter > (`Get-ChildItem -Path .\ -Include .txt -Recurse | Move-Item -Destination C:\TextFiles`).

--- Example 5: Move registry keys and values to another key ---

```
Move-Item "HKLM:\software\mycompany\*" "HKLM:\software\mynewcompany"
```

Example 6: Move a directory and its contents to a subdirectory of the specified directory

```
Move-Item -LiteralPath 'Logs[Sept`06]' -Destination 'Logs[2006]'
```

The LiteralPath parameter is used instead of Path , because the original directory name includes left bracket and right bracket characters (`[` and `]`). The path is also enclosed in single quotation marks (`'`), so that the backtick symbol (`` ` ``) is not misinterpreted.

The Destination parameter must also be enclosed in single quotation marks because it includes brackets that can be misinterpreted.

REMARKS

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

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

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

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