CmdletJack 2.5 User's Guide

Contents

[New in 2.5: 2](#_Toc387214501)

[Requirements 2](#_Toc387214502)

[Installation 2](#_Toc387214503)

[Get Cmdlet Projects 2](#_Toc387214504)

[Usage 2](#_Toc387214505)

[Reports 2](#_Toc387214506)

[Searches 3](#_Toc387214507)

[Output 4](#_Toc387214508)

[Descriptions – CJ\_Descriptions.csv 6](#_Toc387214509)

[Parameters – CJ\_Parameters.csv 7](#_Toc387214510)

[Examples – CJ\_Examples.csv 8](#_Toc387214511)

[Links – CJ\_Links.csv 9](#_Toc387214512)

[Summary – CJ\_Summary.csv 10](#_Toc387214513)

[Warnings – Warnings.csv 11](#_Toc387214514)

[How to drill down on QA warnings 12](#_Toc387214515)

[List of QA warnings 12](#_Toc387214516)

[How to create custom QA reporting 13](#_Toc387214517)

**Welcome to CmdletJack 2.5.**

CmdletJack creates spreadsheets (.CSV files) of cmdlet documentation content as reports. It also provides quality assurance analysis and you can also search for text.

# New in 2.5:

* New report on Input and Output objects.
* Improved Summary report.
* Search, reporting, and other bug fixes.

# Requirements

CmdletJack requires the following:

* .NET Framework 4, .NET Framework 4 Client Profile, .NET Framework 4.5, or later.
* Windows PowerShell 3.0

# Installation

Copy **CmdletJack.exe** from the share below to a local directory on your computer.

You can determine the current version of CmdletJack by right-clicking the exe, chose Properties, and check the product version on the Details tab.

# **Note: This documentation is pertinent only to DxStudio and does not address CAPS.**

# Get Cmdlet Projects

1. Create a directory to contain the cmdlet projects you are interested in. The perform a DxReq request to get the psmrefmaml help file for a project, such as:

DxReq SC2012\_SPF\_Cmdlets /d:wswemdmainpsblue /t:psmaml

1. When the email from IXP Build arrives, click on the drop link.
2. Navigate to the **psmrefmaml** directory in the build.
3. Copy the XML file from that directory to your cmdlet directory, such as SC2012\_SPF\_Cmdlets.XML. You do not need the HelpInfo xml file.
4. Repeat for each cmdlet project you are interested in.

# Usage

## Reports

The usage is as follows:

CmdletJack <*path to cmdlet directory or specific file*>

For example, consider that you have a directory that contains three cmdlet projects as shown here:

C:\cmdlets  
- CmdletsA.XML  
- CmdletsB.XML  
- CmdletsC.XML

To create reports of all the projects, the syntax is simply:

CmdletJack c:\cmdlets

To create a report for just a specific cmdlet project:

CmdletJack c:\cmdlets\CmdletsB.XML

All reports overwrite any existing reports, unless followed by the append option: -a

For example:

CmdletJack c:\cmdlets -a

CmdletJack c:\cmdlets\CmdletsB.XML -a

The append option is available only for reports, not for searches.

You can just have one cmdlet project in the directory and still use the directory syntax. You can files of other types in the directory. CmdletJack processes all the XML files in the directory. You can have other XML files besides cmdlet projects, and CmdletJack will not process them but that should be avoided.

Here is an example shot of CmdletJack:

## 

## Searches

You can search for text in Cmdlets. CmdletJack will search for the provided string pattern in the synopsis, description, parameter description, example title, example remarks, and in example code. The syntax is as follows:

CmdletJack <*path directory or file*> search <pattern> [<matchcase|wholeword|both>]

The word 'search' is required after the path. Specifying the 'both' option will match the case and whole words. You need to put the pattern in quotes only if the pattern contains spaces.

Examples:

CmdletJack c:\cmdlets search "virtual network"

CmdletJack c:\cmdlets search "VIRTUAL NETWORK" matchcase

CmdletJack c:\cmdlets search diskdrive wholeword

CmdletJack c:\cmdlets search RESTfull matchcase

CmdletJack c:\cmdlets search RESTfull both

# Output

Every time you run CmdletJack, it generates the following reports in the specified cmdlets directory:

* CJ\_Descriptions.csv
* CJ\_Examples.csv
* CJ\_Links.csv
* CJ\_InOut.csv
* CJ\_Parameters.csv
* CJ\_Summary.csv
* CJ\_Warnings.csv

If you perform a search, and if one or more matches are found, CmdletJack generates only this report:

* CJ\_SearchResults.csv

All information regarding quality assurance, such as no code examples and descriptions that do not start correctly are in the CJ\_Warnings.csv report.

If you have run CmdletJack with previous reports in the directory, the data will be appended to the same reports the next time you run CmdletJack. You can then filter reports for a specific run by using the TimeStamp column in the report that contains the time the report was generated. You can always safely delete any or all reports and run CmdletJack again.

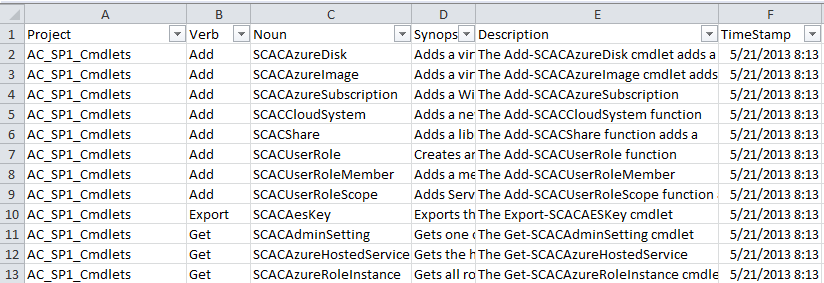
CmdletJack generates the data, but the data mining capabilities are in Excel where can employ filtering and sorting actions on the columns as desired. To increase that potential, the cmdlet verb is in a separate column from the cmdlet noun.

|  |  |
| --- | --- |
| **Tip** | If you run CmdletJack more frequently than one minute apart, it is better to delete the files and create new ones. Filtering the timestamp value with less than a minute is tricky.  Also, consider renaming the cmdlet project file to something shorter for easier manipulation in Excel. |

The following sections provide guidance on each of the reports.

|  |  |
| --- | --- |
| **Note** | Do not assume that because a cell is blank when you open a CSV report in Excel that there is no text. Expand the size of the cell or double-click on it. You may need to click the Word Wrap button.  In addition, some character formatting from DxStudio is rendered with special characters in the spreadsheet, such as "MicrosoftÂ SystemÂ CenterÂ". Always check the DxStudio or the topic on TechNet in these cases. |

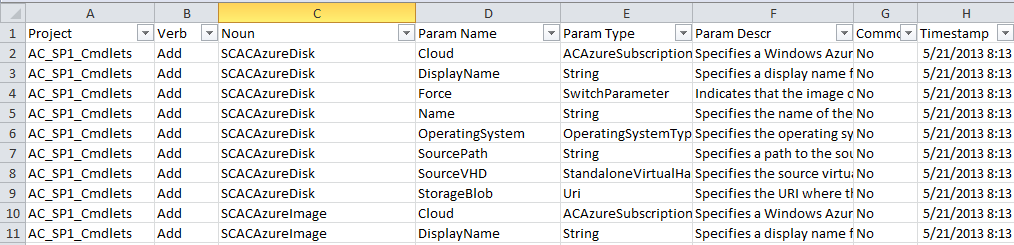
# Descriptions – CJ\_Descriptions.csv



The descriptions report contains the following columns:

* Project
* Verb
* Noun
* Synopsis (The shorter description of the cmdlet.)
* Description (The longer description of the cmdlet.)
* TimeStamp

# Parameters – CJ\_Parameters.csv



The parameters report contains the following columns:

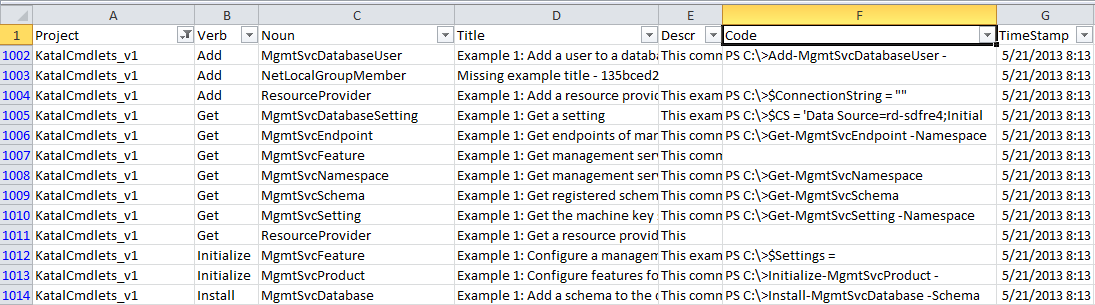
* Project
* Verb
* Noun
* Parameter Name
* Parameter Data Type
* Parameter Description
* Common
* TimeStamp

The Common column indicates Yes if the parameter is one of the following common parameters:

* Debug
* ErrorAction
* ErrorVariable
* OutVariable
* OutBuffer
* Verbose
* WarningAction
* WarningVariable
* WhatIf
* Confirm

You can decrease the size of the report if you filter the Common column to No. Descriptions for common parameters are automatically generated by DxStudio.

# Examples – CJ\_Examples.csv



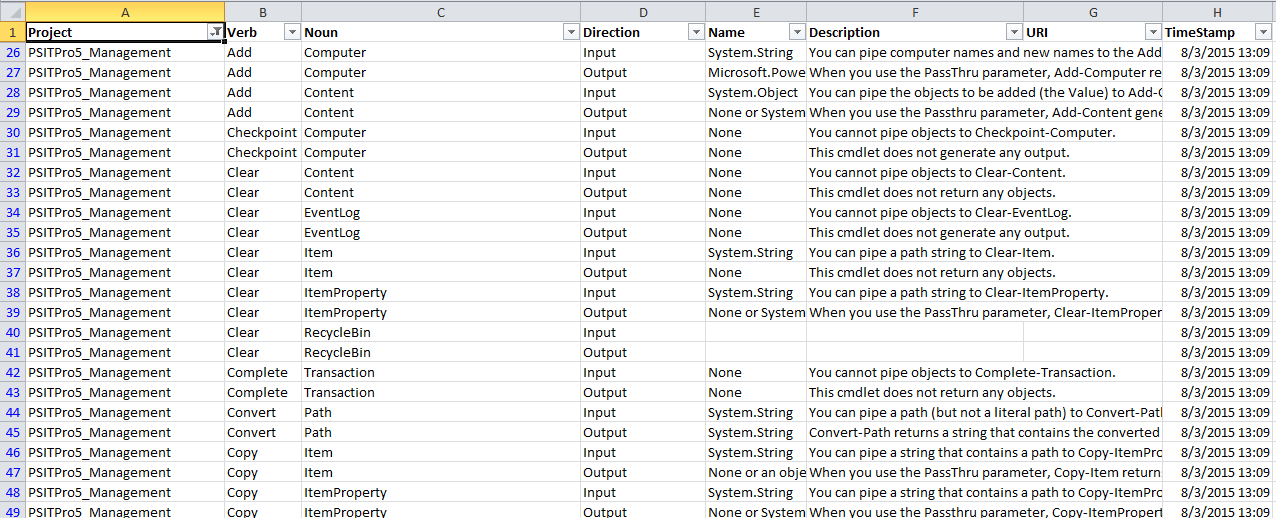
The Examples report contains the following columns:

* Project
* Verb
* Noun
* Title
* Description
* Code
* TimeStamp

If an example title is missing, CmdletJack will insert "Missing example title – " followed by a partial GUID for infrastructure purposes.

With Example reports, it is important to format the cells so that you get the complete content in the Title, Description, and Code columns.

# Input and Output objects – CJ\_InOut.csv



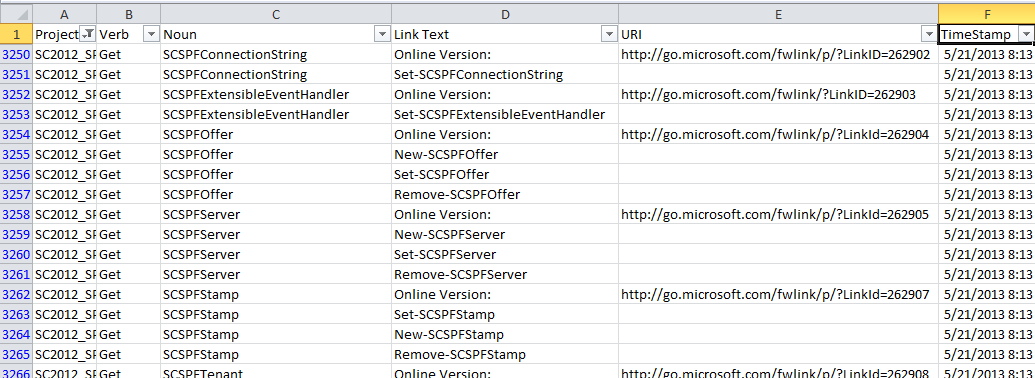
The InOut report contains the following columns:

* Project
* Verb
* Noun
* Direction
* Name
* Description
* URI
* TimeStamp

This report describes the Input objects and the Output objects (ReturnValue objects in the PSMAML). Every cmdlet has a record for an input object and an output object, whether or not they are required.

To filter to see only the input or output objects, filter on the Direction column.

# Links – CJ\_Links.csv



The Links report contains the following columns:

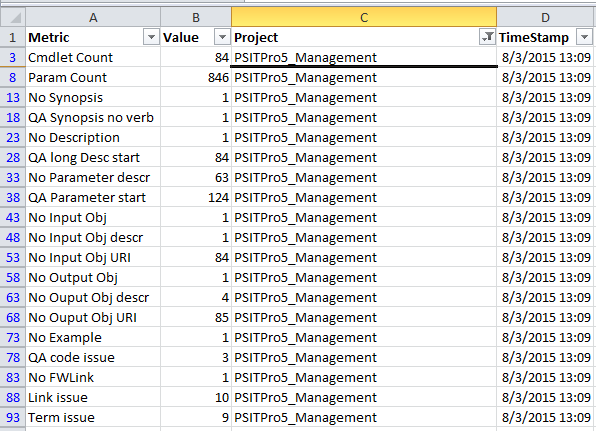
* Cmdlet Project
* Verb
* Noun
* Link Text
* URI
* TimeStamp

The values for Link Text column should only be "Online Version:" or a link to another cmdlet that has been formatted correctly in DxStudio.

The values for the URI should only be an FWLink to the TechNet cmdlet topic. To optimize the ranking of the topic in search results, include a "/p/" in the URI as shown here:

http://go.microsoft.com/fwlink/p/?LinkId=262905

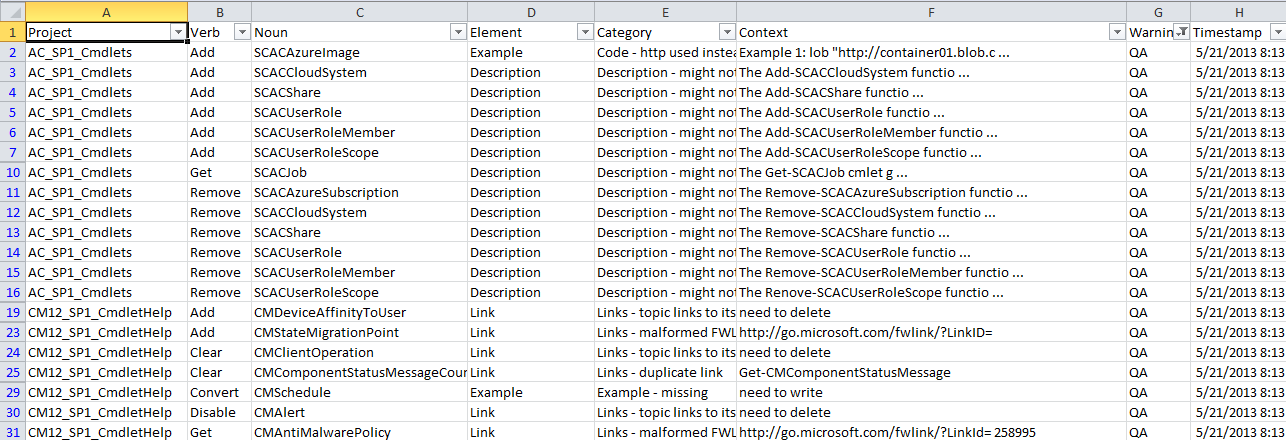
# Summary – CJ\_Summary.csv



The Summary report contains the following columns:

* Metric
* Value
* Project
* TimeStamp

# Warnings – Warnings.csv



The Warnings report contains the following columns:

* Project
* Parameter Count
* Verb
* Noun
* Element
* Category
* Context
* Warning
* Timestamp

The first thing to do when investigating the Warnings report is to filter by the Warning type, which are described in the following table. Writers need to be only concerned about QA warnings.

|  |  |
| --- | --- |
| **Warning** | **Description** |
| Element | These warnings are mostly about empty XML elements that have no consequence for the writer, such as empty example title, description, and code that are not being used. These warnings are mostly useful for investigating of the XML output in diagnosing structural issues. |
| QA | These warnings should be investigated by the writers. See a list of them below. |
| Schema | These warnings detect new schema elements being introduced and are mainly by CmdletJack to keep in sync with structural changes. |
| Internal | These warnings occur if a .NET Framework exception is not handled by CmdletJack. The goal is that they should never be generated.  **Important** - If you do see one, please email the row of the warning to v-bruham immediately. |

## How to drill down on QA warnings

Use the following filtering techniques to determine the warnings you are interested in.

1. Filter by the 'QA' warning type.
2. Filter by the Element column (not to be confused with the Element warning type). The Element column allows you to filter the parts of the cmdlet topic you are interested in and contains only the following values:
   * Description
   * Example
   * Link
   * Parameter
   * Synopsis
3. Choose the category. These are the various QA warnings that are generated.
4. Review the context to see where the error occurred. For parameter warnings, the parameter in question precedes the context separated by a colon. An ellipsis indicates only a portion of the relevant text is used.

## List of QA warnings

|  |  |
| --- | --- |
| **Element warning occurs** | **Warning** |
| Description | **Description - starts with a space**  **Description - might not start correctly** (Correct: 'The Get-WhatEver cmdlet' …)  **Description – missing**  **Description - starts correctly but is incomplete**  **Custom:** <text specified in CustomQA.txt> |
| Example | **Code - check company name in URL** (all company names should be from the LCA site: <http://lcaweb/CTP/Trademarks/Pages/FictitiousNames.aspx>)  **Code - http used instead of https** (might not be an error, but good to verify)  **Code - only has a prompt** (code only has 'PS C:\>')  **Description - starts with a space**  **Example – missing** (topic has no examples at all or if all code blocks just have the prompt)  **Custom:** <text specified in CustomQA.txt> |
| Link | **Links - duplicate link** (There is duplicated link to another cmdlet)  **Links - malformed FWLink**  **Links - missing FWLink**  **Links - topic links to itself** |
| Parameter | **Param descr - might not start with verb** (Excludes occurrences such as 'deprecated' or 'do not use')  **Param descr – missing**  **Custom:** <text specified in CustomQA.txt> |
| Synopsis | **Synopsis - might not start with a verb** (Excludes occurrences such as 'deprecated' or 'do not use')  **Synopsis - starts with a space**  **Custom:** <text specified in CustomQA.txt> |
| (any element) | **Branding - PowerShell not preceded by Windows**  **Branding - wrong PowerShell casing** |

## How to create custom QA reporting

If a file exists in the cmdlets directory named **CustomQA.txt**, and that file has one or more strings each on a line, then hits on those strings are reported in QA warnings.

First filter the CJ\_Warnings.csv spreadsheet by the warning type (column G) to 'QA' and then filter for all the "Custom: MySQL" (column E) and note the occurrences (column F).

For example, here are the results for "MySQL" on a line in CustomQA.txt. If you wanted to search for a whole word, search for " MySQL ". The reporting is case insensitive, so the casing in the text file does not matter.

