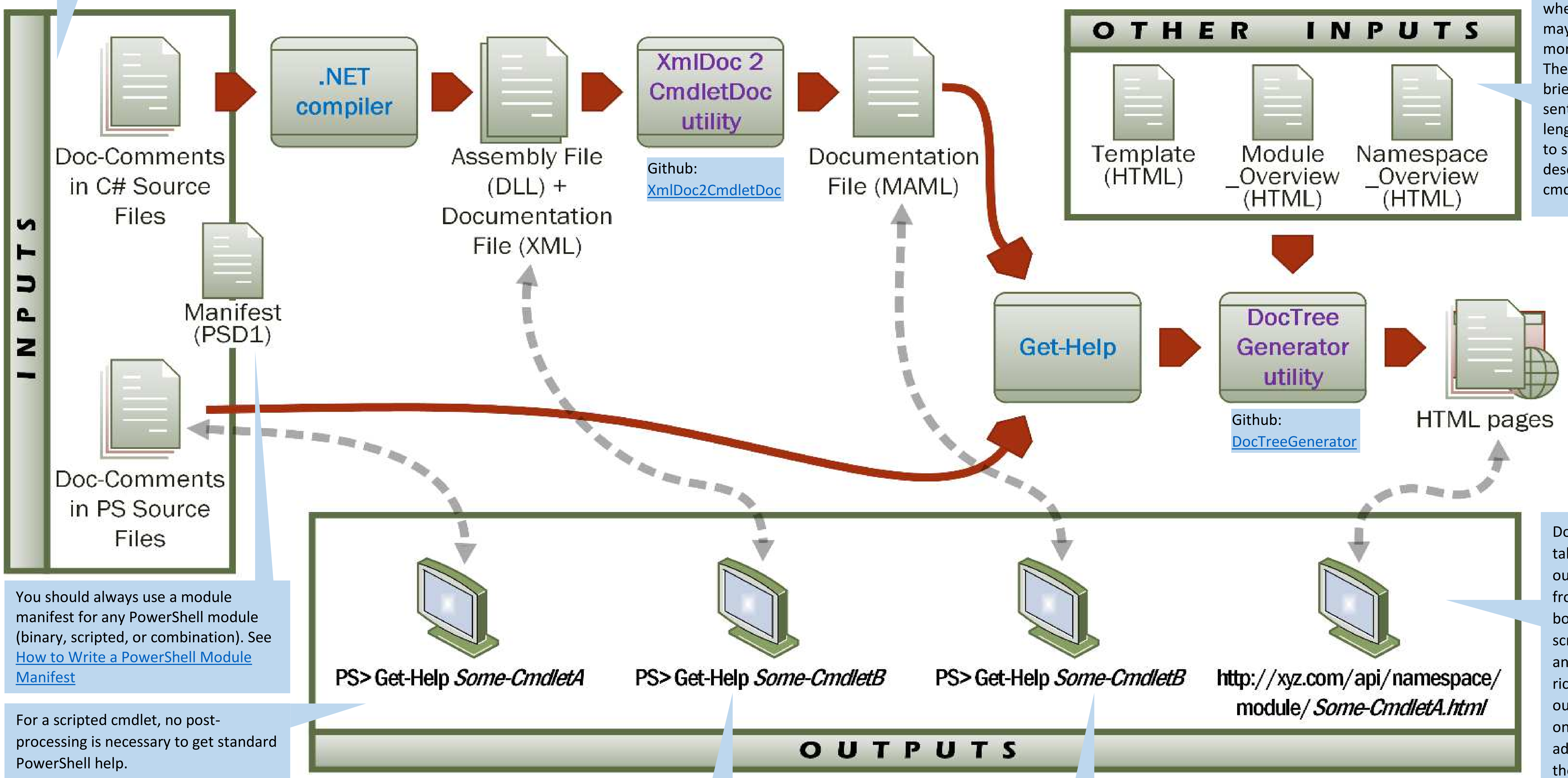
Documenting PowerShell Cmdlets End-to-End

PowerShell's Get-Help presents documentation for a single command at a time, though **you** need to supply its source material in the form of documentation comments ("doc-comments") instrumenting your code. Scripted cmdlets have long been easy to instrument, but not until the recent XmlDoc2CmdletDoc could you easily provide such doc-comments for your binary cmdlets, too. And like Javadoc for Java or Sandcastle for C#, DocTreeGenerator provides easy support for generating a hyperlinked, web-based documentation tree.

Copyright © 2016 Michael Sorens
2016.03.20 ■ Version 1.0.0
Published on Simple-Talk.com at
http://bit.ly/1WlvyOp

Start with instrumenting your source code—be it a set of cmdlets written in PowerShell or C#. Document the PS cmdlets per How To Document Your PowerShell Library and document the C# cmdlets per Documenting Your PowerShell Binary Cmdlets, both here on Simple-Talk.com.

For generating a web documentation set, you will need just a few other files: one module_overview.html per module, and one namespace_overview.html per namespace, where a namespace may contain one or more modules). These may be as brief as one sentence or as lengthy as you want to sufficiently describe your cmdlets.



INPUTS

Doc-Comments in C# Source Files

.NET compiler

Assembly File (DLL) + Documentation File (XML)

XmlDoc 2 CmdletDoc utility

Github: XmlDoc2CmdletDoc

Documentation File (MAML)

OTHER INPUTS

Template (HTML)

Module _Overview (HTML)

Namespace _Overview (HTML)

Manifest (PSD1)

Get-Help

DocTree Generator utility

Github: DocTreeGenerator

HTML pages

Doc-Comments in PS Source Files

PS> Get-Help *Some-CmdletA*

PS> Get-Help *Some-CmdletB*

PS> Get-Help *Some-CmdletB*

http://xyz.com/api/namespace/module/*Some-CmdletA.html*

OUTPUTS

You should always use a module manifest for any PowerShell module (binary, scripted, or combination). See How to Write a PowerShell Module Manifest

For a scripted cmdlet, no post-processing is necessary to get standard PowerShell help.
Get-Help directly processes the cmdlet doc-comments instrumenting your code; see about_Comment_Based_Help.

For a binary (or compiled) cmdlet, compile the cmdlets in your module into a DLL. Using the DLL, Get-Help can provide only a skeleton of a typical help entry, including the syntax, the pipeline-able inputs and outputs, and the standard properties of your parameters.

To get complete, standard help output for binary cmdlets you must instrument your C# source with doc-comments, then compile the code, and finally post-process with XmlDoc2CmdletDoc as described in Documenting Your PowerShell Binary Cmdlets.

DocTreeGenerator takes the full output available from Get-Help for both binary and scripted cmdlets and converts to rich, stylized output. It needs only a modicum of additional work in the form of module and namespace overview files as described above. You then have a hyperlinked interconnected web reference allowing you to drill down from namespace to module to function, as well as a full index giving you direct access to all your functions, modules, and namespaces.