# Using the Markdown Conversion Tools

Leith McCombs, February 27, 2015

## Overview

2md.exe and md2hxs.exe are two console applications used to convert between markdown (open authoring) and .hxs files. 2md.exe converts .hxs files to markdown. Md2hxs.exe converts a directory of markdown files into an .hxs file. Both depend on a third party tool, Pandoc.exe, to convert between HTML and markdown, and on the HX Compiler, hxcomp.exe, to compile and decompile .hxs files. 2md.exe and md2hxs.exe handle the metadata, set TOC order, and clean up the pandoc output between conversions.

## Setup Requirements

You must have the following on your system to run the conversion tools.

1. Pandoc.exe. Install from <https://github.com/jgm/pandoc/releases/tag/1.13.2> (The Windows .msi installer is at the bottom of the page.)
2. HxComp.exe. Comes with the [Visual Studio 2008 SDK](http://www.microsoft.com/en-us/download/details.aspx?id=21827), which unfortunately requires that you have VS 2008 installed, at least temporarily.

Alternatively, you can try your luck with this [stand-alone installer from X-Tensive](http://tips.x-tensive.com/2009/07/separate-microsoft-help-20-sdk.html).

1. 2md.exe. Enlist in the CPUB.TOOLS branch at <https://microsoft.visualstudio.com/DefaultCollection/CPUB%20Tools/_git/CPUB.tools> and run from [branch]\MarkdownTools\2md.
2. Md2hxs.exe. Enlist in the CPUB.TOOLS branch at <https://microsoft.visualstudio.com/DefaultCollection/CPUB%20Tools/_git/CPUB.tools> and run from [branch]\MarkdownTools\md2hxs.

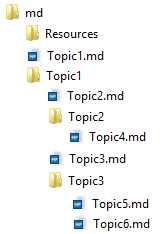
## Basic Scenarios

This section deals with basic usage and one-off conversions.

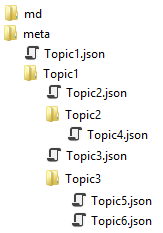
### Converting a directory tree of markdown files to an .hxs

#### Preparation

If your markdown directory tree is more than one level deep, each subfolder that contains markdown files must have the same name (not including extension) as a sibling file in the same folder. Ideally, the top level of the markdown tree should only have one markdown file, to act as the top level TOC node, with the rest of the markdown in subfolders. To compile with multiple markdown files in the root folder, you must use the –allowrootless switch. Other folders containing images and other non-markdown content can be anywhere in the tree.

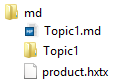


If you have metadata (generated by 2md or md2hxs) it should be in a separate folder outside of your markdown file tree. This folder will contain .json files with the same names as the markdown files, along with a global metadata file named global.json. Global.json contains the metadata values that are shared across the collection, such as Product and Locale. The other .json files contain only those fields specific to the individual topic, which are RLTitle, TOCTitle (where present), and AssetID, which are generated from file content.



If you do not have the topic-specific metadata files, you will be prompted and then md2hxs will create them. If you do not have global.json you will be prompted for the values. **It is very important that you not generate new metadata for topics that are already on MSDN**, as this will create new AssetIDs, which then causes MSDN to generate new copies of the topic.

If you have an .hxtx file (used to preserve TOC order), it will be in the root folder of the markdown directory tree by default, but can be anywhere except for the /compile folder generated by md2hxs as long as you reference it with the –hxtx switch.



#### Execution

The command syntax is **md2hxs.exe [source directory] [output file] [options]**, where [source directory] is the directory that contains the markdown files, [output file] is the full path of the .hxs file to create, and [options] refers to the set of optional switches described in the **Reference** section at the bottom of this document. The [output file] parameter is optional, and defaults to an .hxs file with a name based on the title of the top level markdown file in the parent directory of [source directory]. A log file is generated at [output file].log.

The following table shows some commands and their results.

|  |  |  |
| --- | --- | --- |
| **Command Line** | **Where** | **Result** |
| Md2hxs c:\sdk\md | C:\sdk\md\topic1.md is titled “SDK Overview” | Reads file tree at c:\sdk\md, outputs to c:\sdk\SDK\_Overview.hxs with TOC in alphabetical order. |
| Md2hxs c:\sdk\md c:\sdk\KinectSdk.hxs | “” | Same, but outputs to c:\sdk\KinectSdk.hxs. |
| Md2hxs c:\sdk\md c:\sdk\KinectSdk.hxs | “” AND there is an .hxtx file at c:\sdk\md\file.hxtx | Same, but writes TOC in same order as in .hxtx file. |
| Md2hxs c:\sdk\md c:\sdk\KinectSdk.hxs –hxtx c:\sdk\file.hxtx | “” AND there is an .hxtx file at c:\sdk\file.hxtx | Same, but gets TOC order from c:\sdk\file.hxtx. |

### Converting an .hxs to a directory tree of markdown files

2md created the markdown file tree and metadata file tree in folders named “md” and “meta” respectively, in the same parent directory as the .hxs. The markdown and metadata trees mirror the TOC order in their directory structure, in that each topic that has child nodes in the TOC has a folder in the markdown tree with the same name, and the child topics are in that folder, as in the diagrams above.

File names are derived from the RLTitle attributes of the topics, and then shortened as needed to prevent PathTooLong exceptions. You can adjust the threshold for maximum path length by using the –d switch, but this will change the names of the files, so be sure to clean out any previous versions in source control so you don’t end up with multiple copies of the same file checked in with different names.

If the TOC for your docset has a single top-level topic, you can specify a filename for that topic (which is also the name of the folder that contains the rest of the tree) by using the –n switch.

#### Preparation

Your .hxs file should generally be in a folder by itself so as to avoid duplicate folder names. If your docset has a deep TOC, you should consider giving it a short name with the –n switch and storing the resulting markdown files in a location with a short path to prevent PathTooLong exceptions.

#### Execution

The command syntax is **2md [source file] [options]**, where [source file] is the path to your .hxs file and [options] refers to the set of optional switches described in the **Reference** section at the bottom of this document.

The following table shows some commands and their results.

|  |  |
| --- | --- |
| **Command Line** | **Result** |
| 2md c:\sdk\kinectsdk.hxs | Decompiles c:\sdk\kinectsdk.hxs, creates markdown tree in TOC order at c:\sdk\md, .hxtx file at c:\sdk\md\kinectsdk.hxtx, and a matching metadata tree at c:\sdk\meta. |
| 2md c:\sdk\kinectsdk.hxs –n k2 | Same, but outputs to c:\sdk\k2.hxs. |
| 2md c:\sdk\kinectsdk.hxs –n k2 –d 120 | Same, but shortens the file names enough to copy the file trees to a 120 character host path. |

## Extended Scenarios

### Preserving TOC order

In an .hxs file, the TOC structure and order is defined in the .hxt file. When converting to markdown, 2md takes the information in the .hxt file and copies it to an .hxtx file, along with the corresponding markdown file paths and asset ids. While users viewing the markdown content through github will still see the contents of each directory in alphabetical order, the .hxtx file allows us to preserve the original topic order so that when new markdown content is added, it can be incorporated into the docset and added to MSDN without losing the order of existing topics. The suggested workflow for maintaining TOC order in a docset that on both github and MSDN is as follows.

1. Convert existing content and TOC to markdown with 2md
2. Share markdown tree (but not metadata tree or .hxtx file) on Github
3. Github content updated by community
4. Convert updated content with md2hxs

If there are new topics from Github, and you want to adjust the TOC order of the new topics:

* 1. Decompile the .hxs with HxComp
  2. Edit the .hxt file
  3. Recompile with HxComp

1. Publish to MSDN
2. If there was new content on either end, re-run 2md to preserve the metadata from the new content, and push to Github.

The .hxtx file is generated in the root of the markdown tree. This is fine as long as you don’t share it out when you share the markdown. But if you would rather keep it in an outside location, you can use the –hxtx [file path] switch in md2hxs to read it from a custom location. Just don’t put it in the /meta folder or any of the generated folders because it might be deleted before it can be read.

### Automation

2md.exe uses fixed file paths, while md2hxs has the flexibility to set specific paths. So to share markdown content on github without sharing metadata or TOC files, I recommend building your markdown tree with 2md in a private location, and then using your automation scripts to copy just the markdown tree (excluding the .hxtx file) to your github depot. You can then tell md2hxs to get its markdown content directly from the repo but get the metadata (-meta) and .hxtx file (-hxtx) from the private location.

If you are running scheduled builds and don’t want to be prompted in mid-execution, you can use the –q switch to run both 2md and md2hxs in quiet mode. This suppresses all prompts, and in the case of errors, causes the application to fail gracefully and write the error to the log file. If there is missing metadata, including for a new topic, running in quiet mode will cause md2hxs to fail out unless you also include the –generate switch, which allows it to generate metadata without prompting.

### Github markdown vs. strict

2md.exe converts HTML to Github-flavored markdown. Md2hxs converts from Github-flavored markdown by default, but can also convert from standard markdown by using the –strict switch. Contributors should be encouraged to use Github markdown, but if your compiled HTML suddenly has a bunch of manual line breaks in the text, that probably means it was authored in strict. In that case, the best approach is to convert it to html and then convert it back to github flavor, or use pandoc to convert it directly from strict to github.

## Reference

### 2md

Syntax: 2md.exe [inputfile] [options]

|  |  |
| --- | --- |
| [inputfile] | The absolute or relative path to the HxS or html file to convert. |
| - n [collection name] | Overrides the default name for the top level output file and directory. |
| - d [destination path length] | Causes 2md to shorten file and directory names enough to be able to host the content in a directory of the specified path length without throwing a PathTooLong exception. Default value is 75. |
| -q | Quiet mode. Skips all user prompts, continuing on warnings and exiting the program on errors. |
| -? | Displays this help text and then exits. |

### Md2hxs

Syntax: md2hxs.exe [source directory] [output file] [options]

|  |  |
| --- | --- |
| [source directory] | The directory containing the top level markdown file. |
| [output file] | Optional. The path and file name of the output HxS file. By default, the .hxs will be written to the parent of [source directory] and named for the title of the first markdown file in the tree. |
| -m [metadata root] | Specifies the root of the directory tree that contains the metadata for the doc set. By default, this is a directory named "\meta" in the same location as [source directory]. |
| -global [path] | Specifies the location of the global metadata for the docset. The default value is [metadata root]\\global.json. |
| -q | Quiet mode. Turns off all prompts except for emergencies. |
| -strict | Tells the program to interpret markdown as "markdown\_strict". The default flavor is "markdown\_github". |
| -generate | Tells the program to generate new metadata without prompting for any files that don't already have it. |
| -pd [path] | Explicitly sets the path to Pandoc.exe. By default, the program assumes that Pandoc.exe is in your %PATH% environment variable. |
| -hx [path] | Explicitly sets the path to HxComp.exe. By default, the program assumes that HxComp.exe is in your %PATH% environment variable. |
| -hxtx [path] | Points to an .hxtx file that that contains TOC information, or to where it should be created if there is not one. The default value is [source directory]\\[outputfilename].hxtx. |
| -? | Displays this help message and exits the program. |