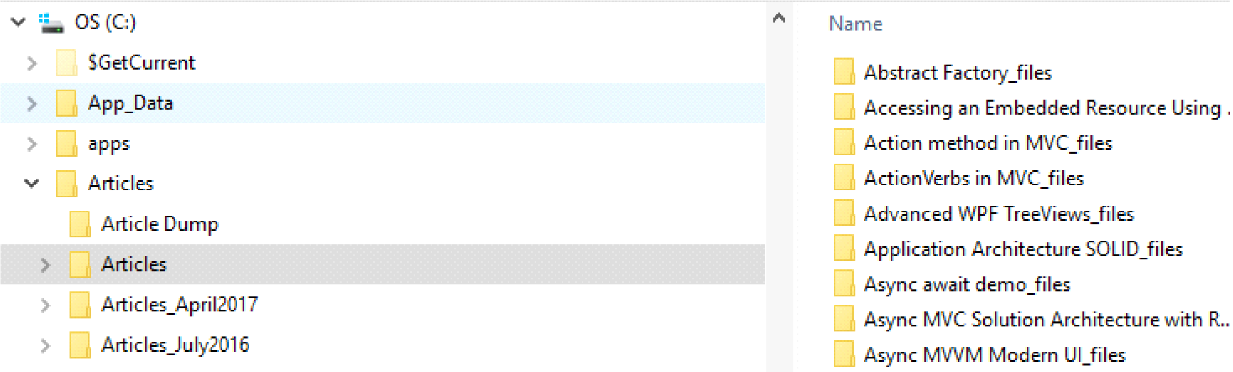
Application Setup

13 November 2017 10:14 AM

1- Set up a folder structure that is easily accessible
2- 2- I found the best way is to have two folders, one contains the files ready to be compiled, the other as a staging area for new documents. I usually just do a quick scan of the documents in the staging folder for things like correct file extensions, valid document names (no special characters)
3- Unzip the resources zip file, it contains a few images and 2 CSS stylesheets
4- These style sheets are essential for the way the chm component operates internally
5- Execute the install MSI, follow the prompts
6- The installer should install a start menu item, if not, the executable will be found under program files X86
7- Depending on how tight your user access rights are setup, you might have to give the executable rights to run under an admin account

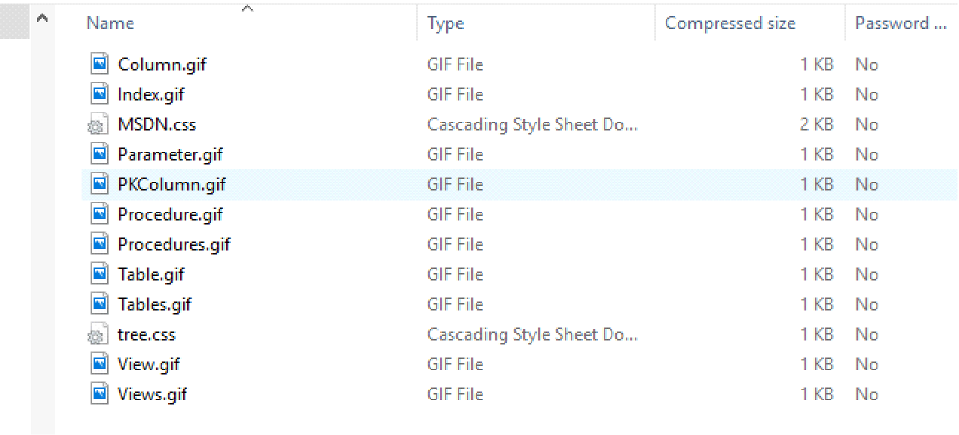


- OS (C:)
  - $GetCurrent
  - App_Data
  - apps
  - Articles
    - Article Dump
    - Articles
    - Articles_April2017
    - Articles_July2016

Name

- Abstract Factory_files
- Accessing an Embedded Resource Using .
- Action method in MVC_files
- ActionVerbs in MVC_files
- Advanced WPF TreeViews_files
- Application Architecture SOLID_files
- Async await demo_files
- Async MVC Solution Architecture with R..
- Async MVVM Modern UI_files

Proposed folder structure
Use **C:\Articles\Articles\** as the main location and **C:\Articles\Article Dump** for staging



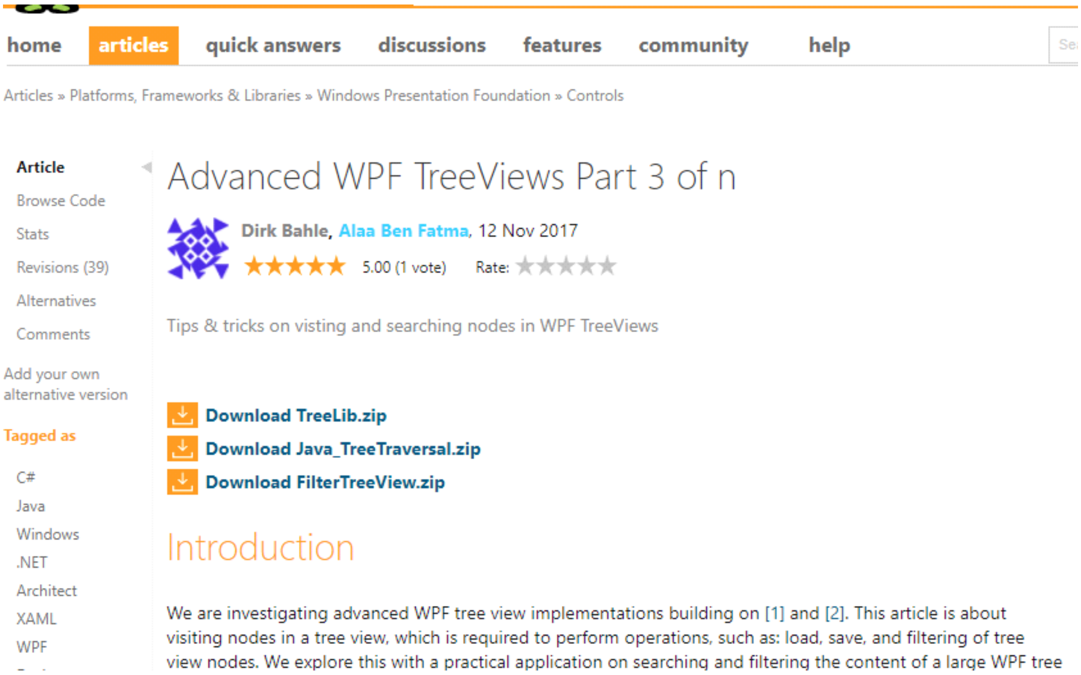
Name	Type	Compressed size	Password ...
Column.gif	GIF File	1 KB	No
Index.gif	GIF File	1 KB	No
MSDN.css	Cascading Style Sheet Do...	2 KB	No
Parameter.gif	GIF File	1 KB	No
PKColumn.gif	GIF File	1 KB	No
Procedure.gif	GIF File	1 KB	No
Procedures.gif	GIF File	1 KB	No
Table.gif	GIF File	1 KB	No
Tables.gif	GIF File	1 KB	No
tree.css	Cascading Style Sheet Do...	1 KB	No
View.gif	GIF File	1 KB	No
Views.gif	GIF File	1 KB	No

Contents of zipped resources file: extract them into your folder where all Your documents are ready to be compiled

Saving Site Content

13 November 2017 10:38 AM

1- Navigate to the website containing the information you're interested in



Articles » Platforms, Frameworks & Libraries » Windows Presentation Foundation » Controls

Advanced WPF TreeViews Part 3 of n

Dirk Bahle, Alaa Ben Fatma, 12 Nov 2017

★★★★★ 5.00 (1 vote) Rate: ★★★★★

Tips & tricks on visting and searching nodes in WPF TreeViews

Download TreeLib.zip

Download Java_TreeTraversal.zip

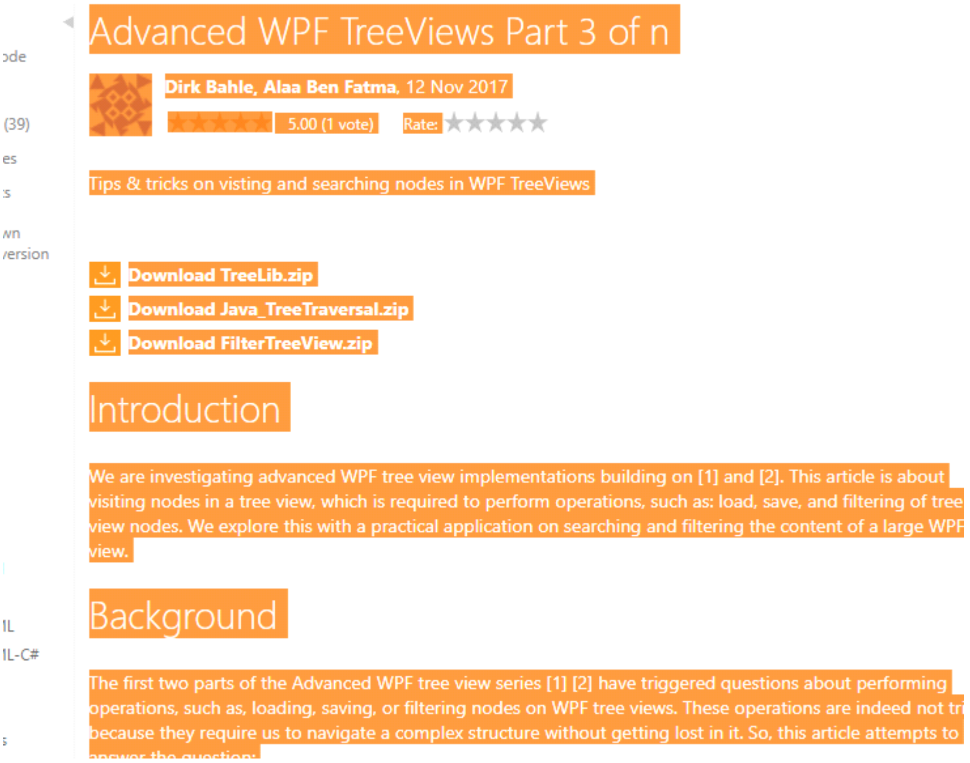
Download FilterTreeView.zip

Introduction

We are investigating advanced WPF tree view implementations building on [1] and [2]. This article is about visiting nodes in a tree view, which is required to perform operations, such as: load, save, and filtering of tree view nodes. We explore this with a practical application on searching and filtering the content of a large WPF tree

2- Select the text you want and highlight to copy

3- Try to just select just the information you require and exclude any embedded advertisements, navigation bars and other unnecessary content, it just adds to the size of the end product



Advanced WPF TreeViews Part 3 of n

Dirk Bahle, Alaa Ben Fatma, 12 Nov 2017

5.00 (1 vote) Rate:

Tips & tricks on visting and searching nodes in WPF TreeViews

Download TreeLib.zip

Download Java_TreeTraversal.zip

Download FilterTreeView.zip

Introduction

We are investigating advanced WPF tree view implementations building on [1] and [2]. This article is about visiting nodes in a tree view, which is required to perform operations, such as: load, save, and filtering of tree view nodes. We explore this with a practical application on searching and filtering the content of a large WPF view.

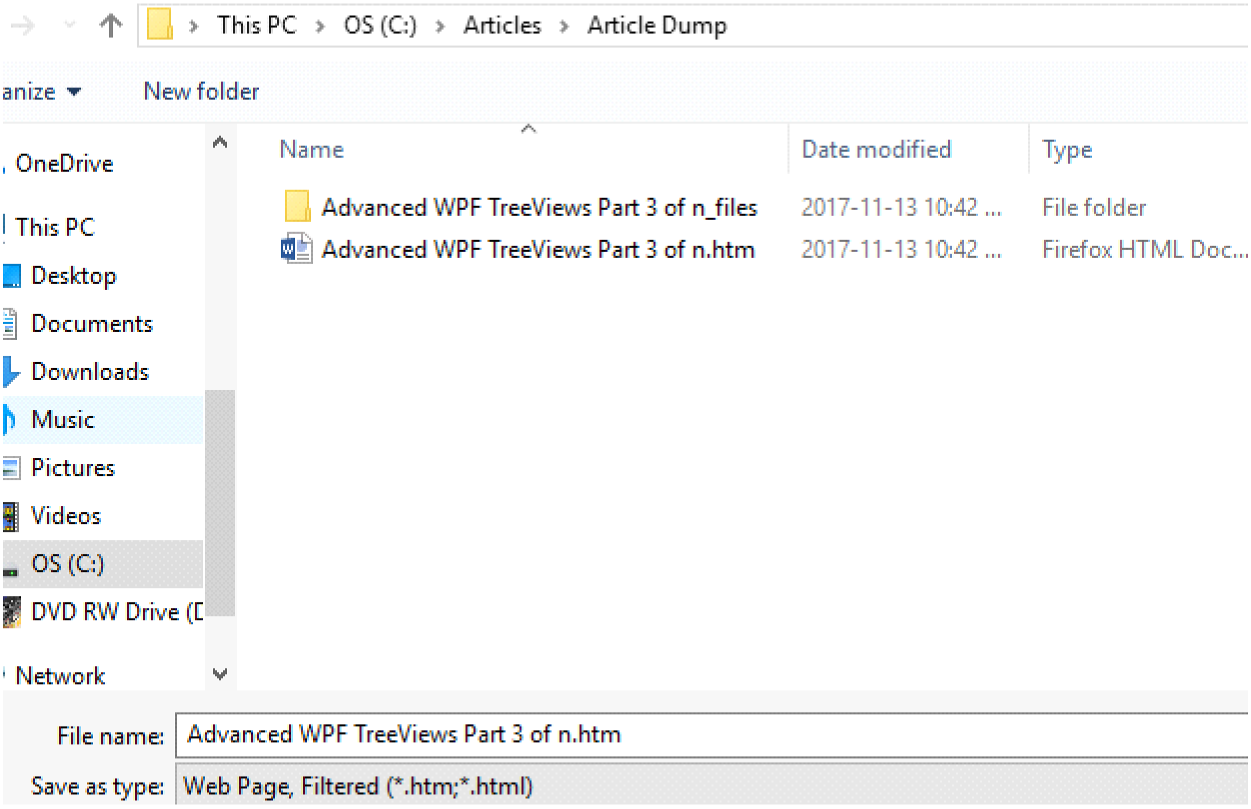
Background

The first two parts of the Advanced WPF tree view series [1] [2] have triggered questions about performing operations, such as, loading, saving, or filtering nodes on WPF tree views. These operations are indeed not tr because they require us to navigate a complex structure without getting lost in it. So, this article attempts to answer the question:

4- Open any word processor application and paste the content. Save the document into either your main folder location or your staging folder. (i.e. C:\Articles\Article Dump)

5- NB: Save the document as **web page filtered,** this is important for the CHM compiler component which uses web pages as the base document type in its compilation process.

6- Another reason for using this file type extension is that the document images are also saved in a separate sub folder which is picked up by the application and included in the chm when compiling.



Name	Date modified	Type
Advanced WPF TreeViews Part 3 of n_files	2017-11-13 10:42 ...	File folder
Advanced WPF TreeViews Part 3 of n.htm	2017-11-13 10:42 ...	Firefox HTML Doc...

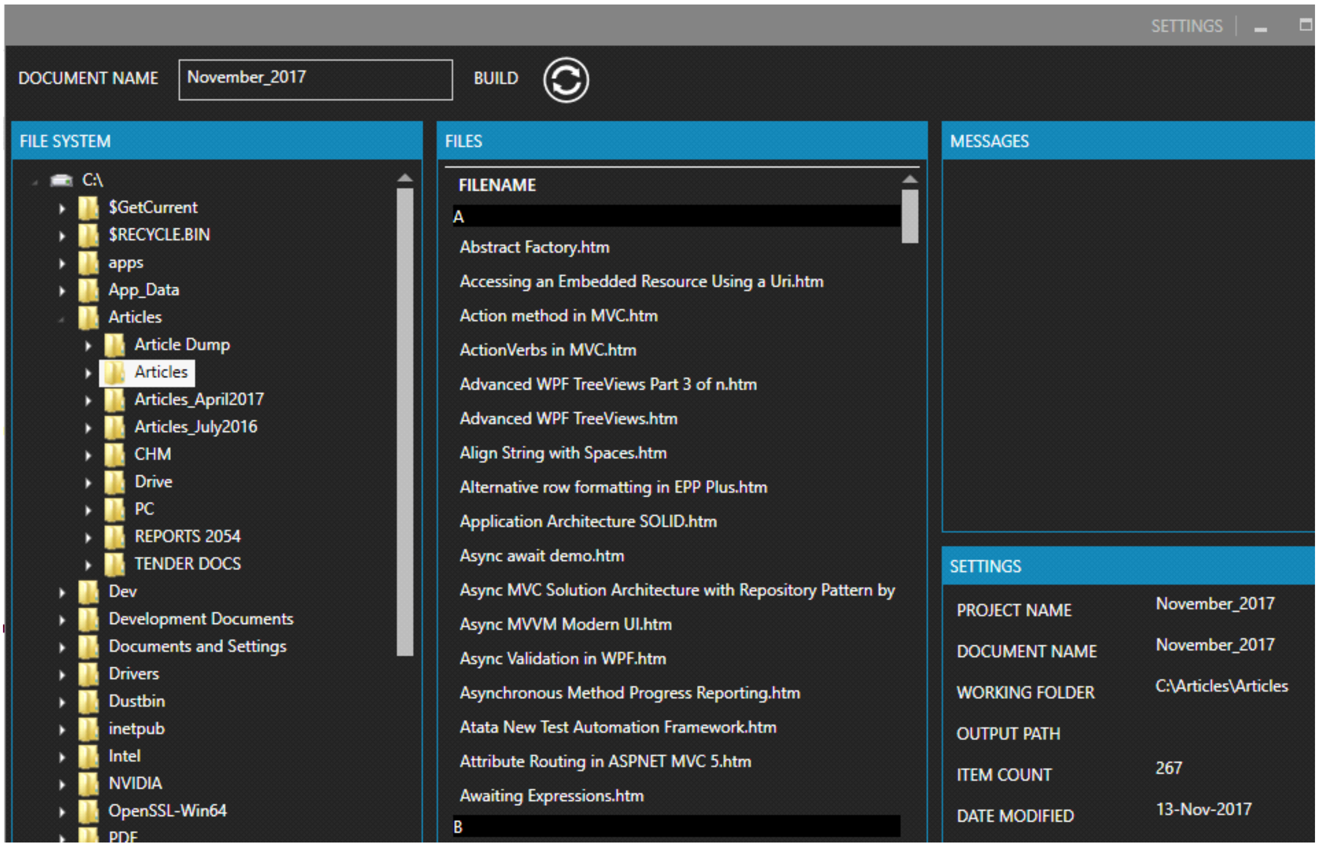
File name: Advanced WPF TreeViews Part 3 of n.htm

Save as type: Web Page, Filtered (*.htm;*.html)

Using the Application

13 November 2017 10:42 AM

1- Start the Article Compiler application
2- Select the main folder location where all your documents are sitting
3- The documents will be listed only if they have a **.htm** file extension
4- The default name of the final product is the current month, but it can be renamed in the text box provided
5- If you are happy with the selected location, press build



DOCUMENT NAME: November_2017

FILE SYSTEM

- C:\
  - $GetCurrent
  - $RECYCLE.BIN
  - apps
  - App_Data
  - Articles
    - Article Dump
    - Articles
    - Articles_April2017
    - Articles_July2016
    - CHM
    - Drive
    - PC
    - REPORTS 2054
    - TENDER DOCS
  - Dev
  - Development Documents
  - Documents and Settings
  - Drivers
  - Dustbin
  - inetpub
  - Intel
  - NVIDIA
  - OpenSSL-Win64
  - PDF

FILENAME

A

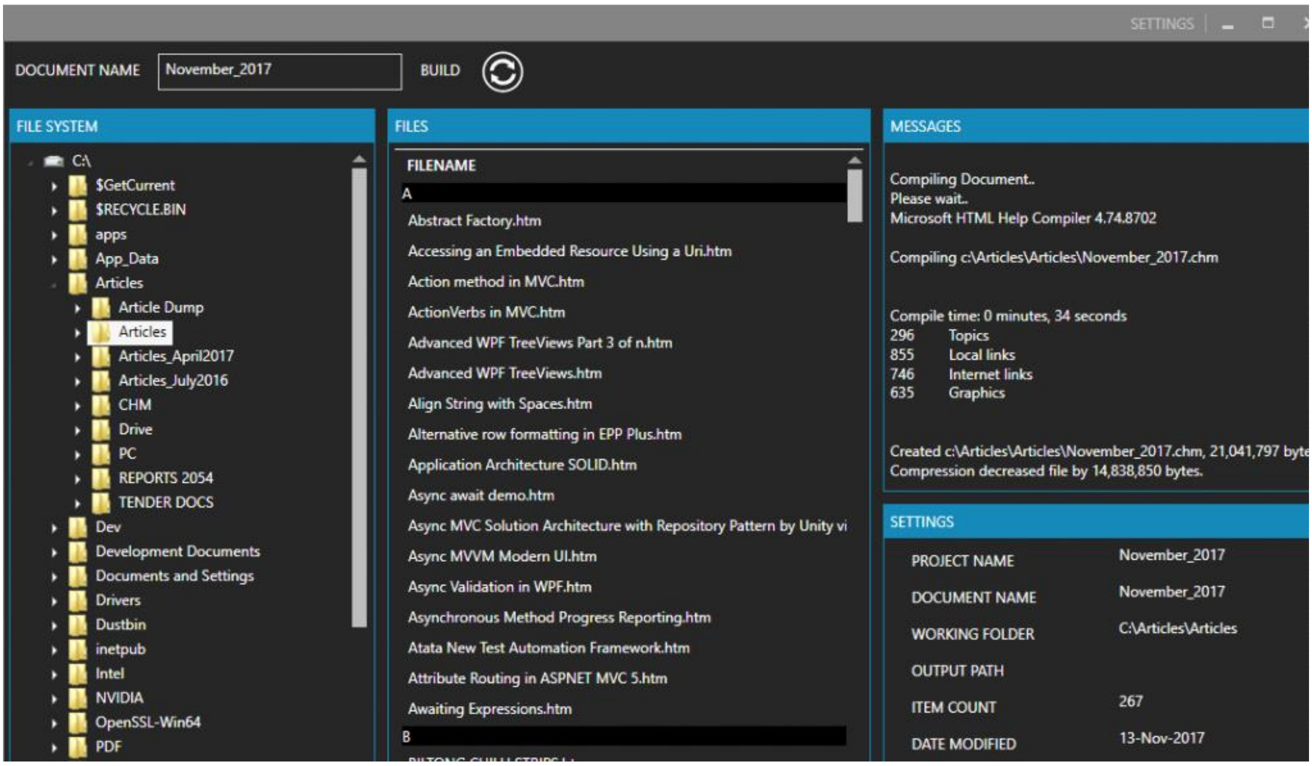
- Abstract Factory.htm
- Accessing an Embedded Resource Using a Uri.htm
- Action method in MVC.htm
- ActionVerbs in MVC.htm
- Advanced WPF TreeViews Part 3 of n.htm
- Advanced WPF TreeViews.htm
- Align String with Spaces.htm
- Alternative row formatting in EPP Plus.htm
- Application Architecture SOLID.htm
- Async await demo.htm
- Async MVC Solution Architecture with Repository Pattern by
- Async MVVM Modern UI.htm
- Async Validation in WPF.htm
- Asynchronous Method Progress Reporting.htm
- Atata New Test Automation Framework.htm
- Attribute Routing in ASPNET MVC 5.htm
- Awaiting Expressions.htm

B

PROJECT NAME	November_2017
DOCUMENT NAME	November_2017
WORKING FOLDER	C:\Articles\Articles
OUTPUT PATH	
ITEM COUNT	267
DATE MODIFIED	13-Nov-2017

6-Once the application has finished the process, a confirmation message will be displayed with the size and location of the final CHM file.

7- Depending on the size of the file folder, it should take about 30 seconds to compile about 250 documents



DOCUMENT NAME November_2017

```
Compiling Document..
Please wait..
Microsoft HTML Help Compiler 4.74.8702

Compiling c:\Articles\Articles\November_2017.chm


Compile time: 0 minutes, 34 seconds
296     Topics
855     Local links
746     Internet links
635     Graphics


Created c:\Articles\Articles\November_2017.chm, 21,041,797 byte
Compression decreased file by 14,838,850 bytes.
```

PROJECT NAME	November_2017
DOCUMENT NAME	November_2017
WORKING FOLDER	C:\Articles\Articles
OUTPUT PATH	
ITEM COUNT	267
DATE MODIFIED	13-Nov-2017

Troubleshooting

13 November 2017 10:44 AM

- 1-The Microsoft CHM compile component is usually part of the Windows OS, if it isn't, it can be downloaded from MSDN. Search for MS help Compiler

- 2-When saving your content, ensure that title of the document doesn't contain special characters

- Over time I found the ideal size of the chm is around 30megs (about 300/400 documents). It’s another way to catalogue your information and make searching for content a lot easier

How the Application Works

13 November 2017   11:13 AM

- The application takes the saved documents and rewrites them into a format that is understood by the CHM component like the TOC navigation bar
- The content and index pages are created by taking the names of the files, reformatted and inserted into custom Html pages with css classes which are specific to the CHM component's internal process of navigation
- These pages are then written out to the hard drive with all the supporting files needed by the CHM compiler to create the CHM document like folder structure, application settings etc
- Once the CHM has been complied, all the output of the process is removed and deleted.
- This is helpful for determining the source of any errors that have occurred during the compilation process.
- The whole point to this application was for the process to be quick and easy, so I used an alphabetic mechanism for cataloguing the documents. It is quite easy to make the indexing customizable, but that would defeat the purpose of a quick one-click process