RWExport-To-HTML

Version: 0.9.1b

Summary:
Documentation for the RWExport-To-HTML.ps1 PowerShell Script

Description:
RWExport-To-HTML.ps1 is a PowerShell script that converts the XML content in Realm Works export files into HTML content that can be printed, published to a web site, loaded into other applications, or otherwise shared, imported and edited.

Requirements
Realm Works (http://www.wolflair.com/realmworks) A version of Microsoft Windows that supports PowerShell. This script was written with PowerShell version 5.1.14393.693. I have not tested compatibility with older versions.

Credits:
RWExport-To-HTML.ps1 was written and developed by EightBitz. The code used for RWExportGUI.ps1 was modified and adapted from code that was generated by Sean Kearney using SAPIEN Technologies Primal Forms (Community Edition) 1.0.8.0. The example code I used is available here: https://blogs.technet.microsoft.com/heyscriptingguy/2011/07/24/create-a-simple-graphical-interface-for-a-powershell-script/

Legal:

the full license text later in this document. Windows and PowerShell are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Realm Works is a registered trademark of Lone Wolf Development.

Additional Notes:
Special shoutout to MaxSupernova for a kind donation. Thank you, Max! Thank you to Daplunk for his video, for his initial work on developing a GUI, and for actively and diligently testing new versions! And thank you to Parody for his suggestions on improving HTML and CSS. Thank you to tmliktoast for suggesting the -ExtractFiles option. This seems like an obvious option that should have been there from the start, but I didn't know how to do it at first and probably wouldn't have bothered to learn had it not been for his suggestion.

00. Table of Contents

Overview

---

01. Exporting from Realm Works

Overview

Before you do anyting with this script, you need to export all or part of your realm from Realm Works to an .rwexport file. This export must be done with the "Compact Output" option. This script is not designed to work with a "Full Export".

There is a great video on this topic by daplunk.

https://www.youtube.com/watch?v=kfL2gGSBBqQ&t=611s

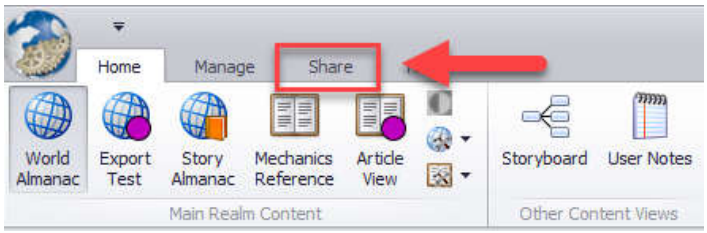
After my initial attempts to export were met with failure and confusion, I watched this video to glean understanding. If your the kind of person who would rather watch an actual demonstration, you might be better off watching that video then reading these instructions.

Nonetheless, I've documented the process below.

Exporting an Entire Realm

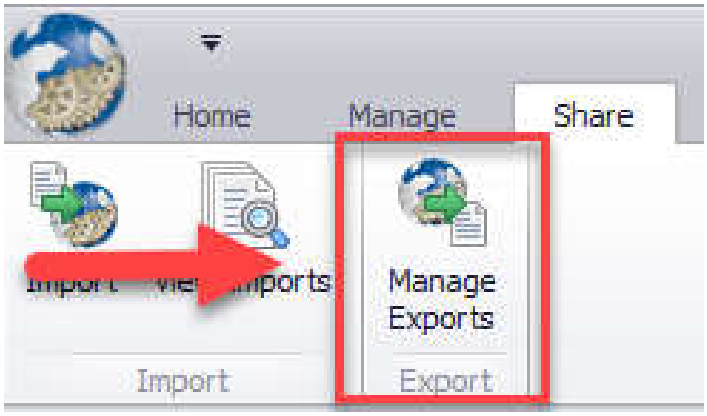
Open the realm you want to export, and you'll notice at the top of the window, there's a new option named "Share".

Share: [Picture]



Clicking that reveals a new option to manage exports.

Manage Exports: [Picture]

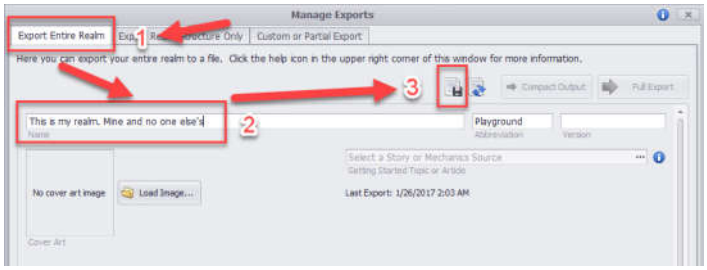


Clicking on Manage Exports gives you a new form to fill out. There are four things here that are of immediate importance. The first three are:

1. Make sure you're on the "Export Entire Realm" tab. You will be by default, but if you've clicked around to see what the other tabs do, make sure you come back to this one.
2. Enter a name for this export.
3. Click the save button.

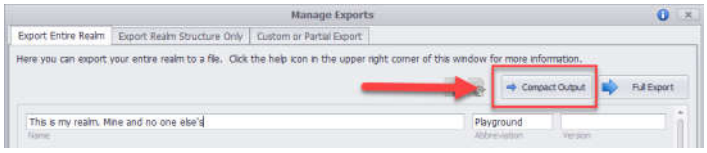
There's are other things you can fill out here, like a summary and a description, but this is all you need to get started. If you do fill in those other things, make sure you hit the save button before you proceed further.

Easy as 1, 2, 3: [Picture]



You may notice that the "Compact Output" button was disabled before. But now that you've entered a name for your export, and saved your preferences, you can click that button. This is, #4 on the list of important things, so go ahead and click "Compact Output".

Compact Output: [Picture]



You will see a pop-up window asking you to "Confirm Compact Output Export."
Here, you have the option to only export revealed content. You can check that option if you wish.
When you've decided, click the "Export" button.

You will get a file dialog window asking you to confirm (or change) the file name, and save.

Once you click save, Realm Works will do its thing and create your export file.

Exporting Selected Topics

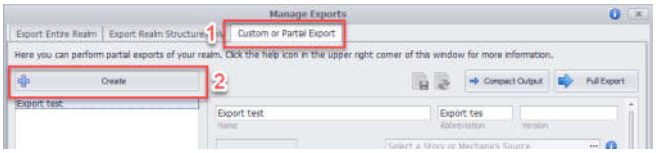
Creating the Export

If you wish to only export selected topics, you start out the same way.

1. Open the realm you wish to export from.
2. Click Share.
3. Click Manage Exports.

This time, you'll want to click on the "Custom or Partial Exports" tab, then click the "Create" button.

Custom or Partial Exports: [Picture]



Type a simple name for your new export (for this example, "Selected Topics"), then click the save button.

Name your export.: [Picture]

You will want to do another Compact Output, but before you do, you have to select which topics to export. So for now, click "Done", and go back to your topics.

Assigning Individual Topics to a Custom Export

Select the topic you wish to assign.

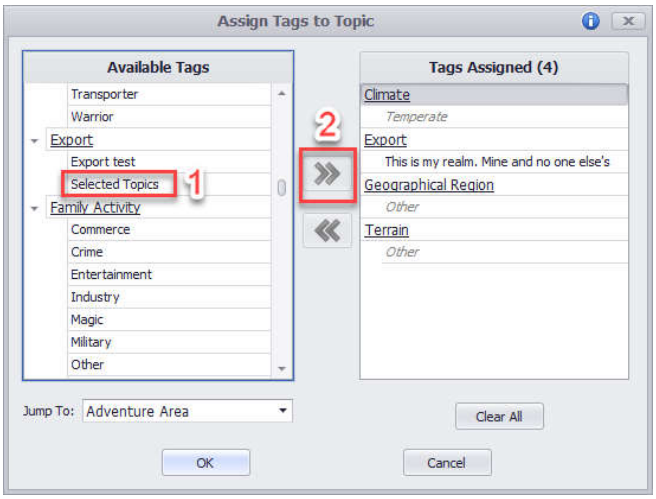
On the far-right side of your Realm Works window, you will see a Table of Contents with "Tags" listed at the bottom.
Click the options button for tags (the little hammer and wrench icon), and select "Modify Tag Assignments"

Modify Tag Assignments: [Picture]
Annotation: Pay no attention to the funky blue dots.

In the tag assignment window, scroll down until you see the "Export" tag domain. Under that, select the export you created earlier (in the case of this example, "Selected Topics", then click the arrows pointing right to assign the tag to the topic.

Click OK and save the topic.

Tag Assignment: [Picture]

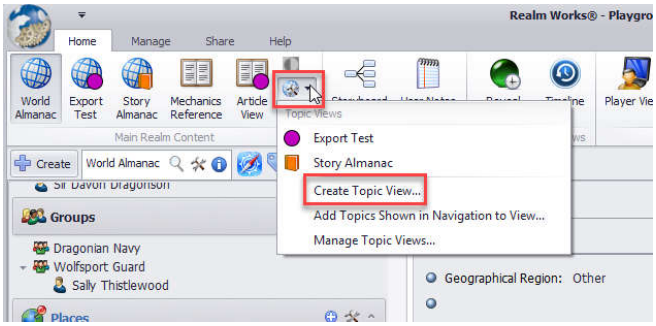


Assigning Tags to Several Topics at Once.

Instead of going topic by topic to assign the tag, you can assign it to several tags at once. You still have to go topic by topic to assign each topic to a vew, but if you have several topics to assign, instead of just a few, this method is still quicker.

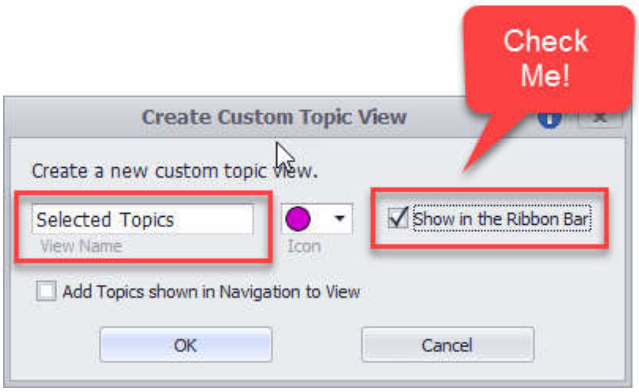
At the top of your Realm Works window, you will see a little icon that looks like a wrench on top of a globe. Click that, and select "Create Topic View".

Create Topic view: [Picture]



Enter the same name you used for your export (in our example, "Selected Topics"), click the box labeled "Show in the Ribbon Bar", then click OK.

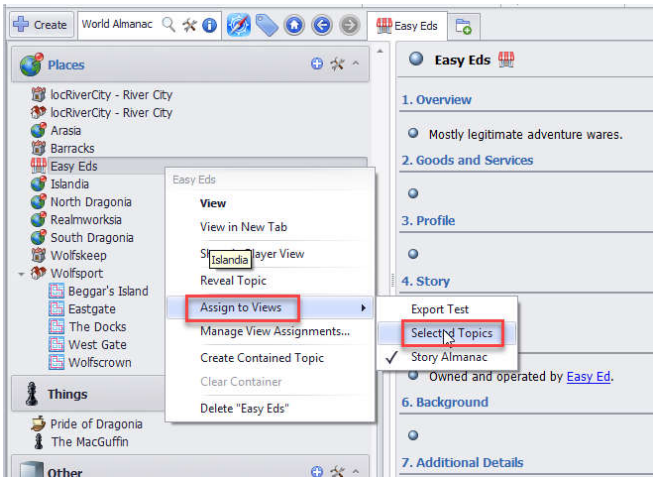
Customize the new topic.: [Picture]



Go to your World Almanac.

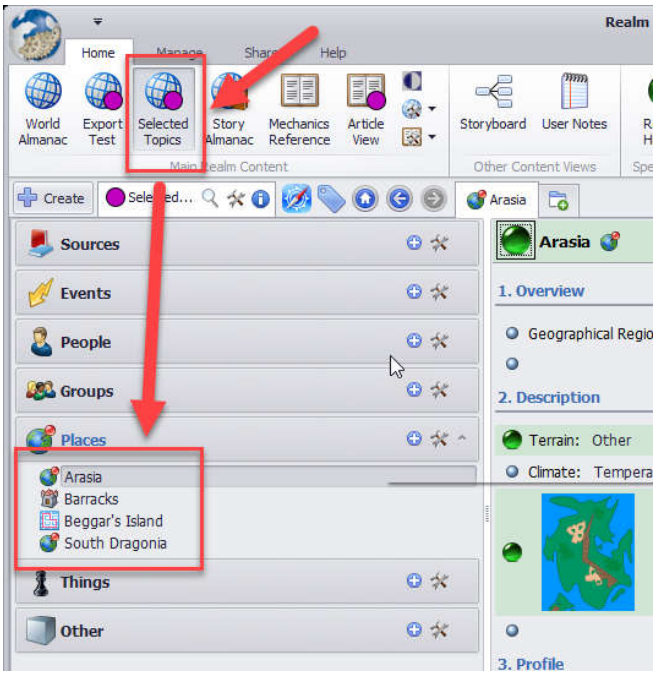
Right click each topic that you want to export, and click "Assign to Views".
In the sub-menu, select the view you created earlier (in our example, "Selected Topics").

Assign to Views: [Picture]



When you are finished assigning topics, click your new view at the top of your window, and you should see all the topics you assigned.

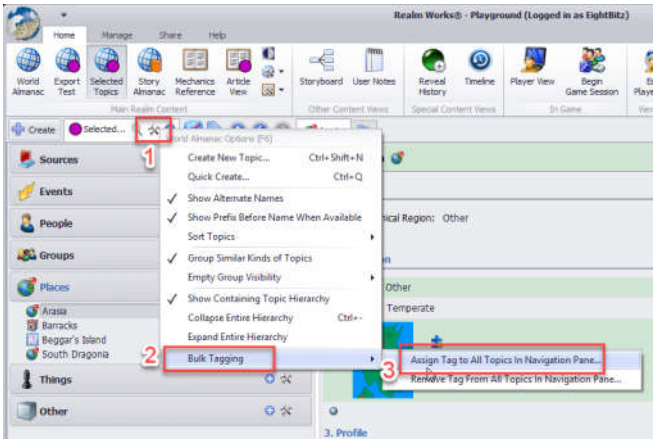
Selected Topics view.: [Picture]



Between the menu ribbon and the topics, there's a little options button. Again, it's the hammer and wrench icon. Click that, and select "Bulk Tagging".

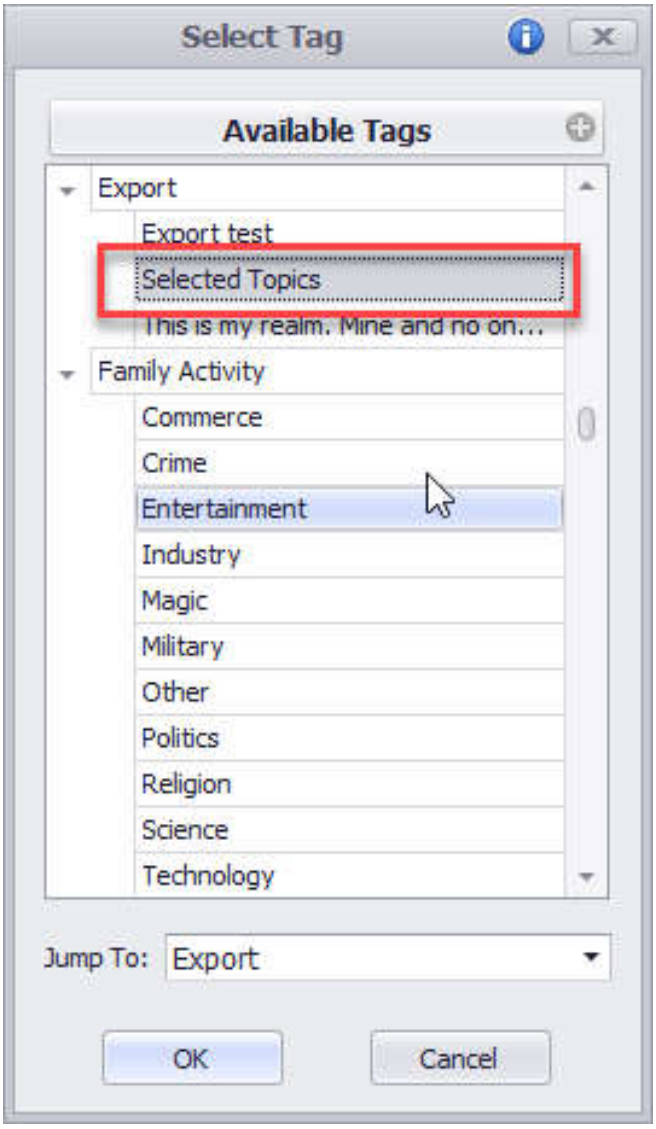
In the sub-menu, select "Assign Tag to All Topics in Navigation Pane".

Bulk Tagging: [Picture]



In the "Select Tag" window, scroll down to (or Jump To) the "Export" domain, and select the export you created earlier (in our example, "Selected Topics"), then click OK.

Select Tag: [Picture]



Export the Tagged Topics

After you've assigned the appropriate export tag to all the topics you wish to export, whether you did it individually or by bulk assignment, it's now time to export those topics.

Click Share
Click Manage Exports

Click the "Custom or Partial Export" tab.

Select the export name that you just tagged your topics with.

Fill in any of the other information you wish to fill in (Summary, Description, etc), and click the save button to save any changes.

When you've saved your changes, click the "Compact Output" button to export your topics.

---

02. Using PowerShell

Overview

PowerShell is a scripting language: an automation tool that runs a series of commands. There are different types of scripting languages. In the past, scripting languages have been rather simple, with limited functionality. Over the years, they have become more sophisticated, and for many purposes, today's scripting languages approach the sophistication of programming languages.

How do I get PowerShell?

PowerShell is a Microsoft product. If you are running Microsoft Windows, it is very likely already installed on your system. PowerShell has been built-in to Windows since Windows XP, service Pack 2. Unless you are very out of date with your operating system, you already have PowerShell.

Should I be concerned about security?

The answer to this question is always yes. Adamantly and absolutely yes.

PowerShell is a very powerful tool that can affect almost every aspect of your computer's configuration, its network activity and its file system. Never run a script you don't trust.

Fortunately, PowerShell has an execution policy that helps protect you from running rogue scripts. This feature will be discussed a bit further below.

PowerShell scripts are plain-text files, so you can view them and see what they do. If you don't understand what you're looking at, you can have someone else view it, or ask other users on the LWD forums (http://forums.wolflair.com) and ask other users of this script about their experiences and opinions. If you have any concerns about the safety of this script, I would encourage you to do that.

How do I use PowerShell?

Basically, there are three different ways to use PowerShell.

1. You can run pre-written scripts.
2. You can run built-in commands.
3. You can write your own scripts.

Option 3 is beyond the scope of this documentation. This documentation will focus primarily on option 1, and on just a few built-in commands that are relevant to option 1.

The four built-in commands you will need to know to use this script are:

- Update-Help
- Get-Help
- Get-ExecutionPolicy
- Set-ExecutionPolicy
- CD

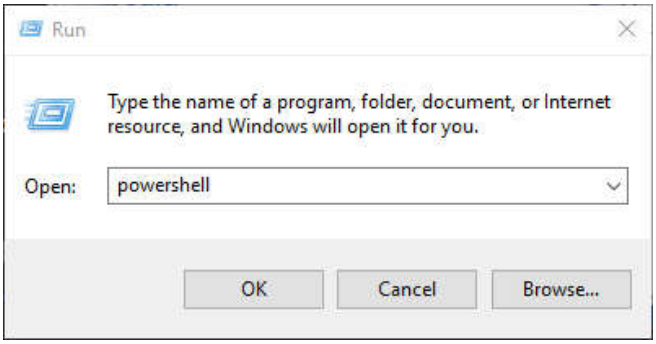
If you are interested in more information about what other built-in commands are available, or how to write your own scripts, you can search the internet for tutorials and books.

Starting PowerShell

To start PowerShell:

1. Right-click the Windows Start menu
2. Click run
3. In the text box, type "powershell"
4. Press ENTER or click OK.

Right-click the Start menu, then click "Run".: [Picture]
Annotation: Type "powershell" then press ENTER or click OK.



Update-Help

One of the most useful commands in PowerShell is Get-Help. The complete help files, though, are not installed by default. If you have never updated your help files, and you run Get-Help, you willl get some basic information, but you will also see the following line:

Get-Help cannot find the Help files for this cmdlet on this computer. It is displaying only partial help.

Fixing this is very easy, and you may as well do it sooner rather than later. Simply type the following command:

Update-Help

then press ENTER.

(Actually, this command is failing on my computer, and I'm not sure why. But don't worry, there's a workaround which will be detailed in the next section.)

CD (Change Directory)

When you open PowerShell, the command line prompt will always tell you what folder (or directory) you're working in. When you first start it up, for example, if your username is Joshua, you will see the prompt look like this:

```
PS C:\Users\Joshua> _
```

You don't always want to work from that directory. For example, if you have saved the RWExport-To-HTML script to a folder on your desktop named "RWExport", you would want to change the directory. For example:

```
CD C:\Users\Joshua\Desktop\RWExport
```

If that is indeed a valid folder, then your prompt will change to reflect it. It will now look like this:

```
PS C:\Users\Joshua\Desktop\RWExport> _
```

Or, since you're already in C:\Users\Joshua, you could just CD from where you are, without having to start with C:\. You could just type:

```
CD Desktop\RWExport
```

Say you have a space in the folder name, though. Instead of RWExport, you named it: RW Export. If you try to change direcotry to that folder as indicated above, you will get an error. For that, you would need to put the folder name in quotes. Not the command name itself, just the folder name. For example, either of the following two commands:

```
CD "Desktop\RW Export"
```

```
CD "C:\Users\Joshua\Desktop\RW Export"
```

Get-Help

If you know what command you want to use, but you are not sure how to use it, Get-Help will show you how. There are different ways to use it, each of which will give you different results.

1. Get-Help <command>
2. Get-Help <command> -examples
3. Get-Help <command> -detailed
4. Get-Help <command> -full
5. Get-Help <command> -online

Version 1 will give you basic information about the command.

Version 2 will give you basic information of the command with a few examples.

Version 3 will give you more detail about the command.

Version 4 will give you all the help that is available about the command.

Version 5 will only work for built-in commands, and you can use this version if your Update-Help command is also failing. This will open a web page that will show you all the help information that is online.

If a person were to write their own script, the output of the Get-Help command is dependent on that information being detailed in the script itself. For this script, I have included that information.

To use Get-Help with this script, change directory (as previously documented) to wherever you saved the script, then type any one of the following commands:

Get-Help .\RWExport-To-HTML.ps1

Get-Help .\RWExport-To-HTML.ps1 -examples

Get-Help .\RWExport-To-HTML.ps1 -detailed

Get-Help .\RWExport-To-HTML.ps1 -full

Notice the .\ before the script name. This specifieis the current directory. This tells the Get-Help command to look for the script right here. Right where we changed directory to. If you were not to include the .\ before the script name, Get-Help would not know where to find the script, and therefore, it would not be able to find the information it needs.

You can also direct the output of any one of those commands to a text file instead of to your screen. Say you wanted to create a text file called RWExport-Quick-Reference.txt, and you wanted this file to have the output of the -full option. You would use this command:

Get-Help .\RWExport-To-HTML.ps1 -full > RWExport-Quick-Reference.txt

The > character tells PowerShell to send the output to the specified file instead to the screen.

Get-ExecutionPolicy

As mentioned earlier, PowerShell has an execution policy to help protect you from rogue scripts. This policy can viewed and modified with two commands. One to view the current setting, and one to modify it.

To view the policy, use the following command:

```
Get-ExecutionPolicy -List > Original-Policy-Settings.txt
```

We're sending the output to a text file, because we want to keep the information for later reference.

If you open that text file, you will see something that looks very similar to this:

```
        Scope ExecutionPolicy
        ----- ---------------
MachinePolicy Undefined
   UserPolicy Undefined
      Process Undefined
  CurrentUser Undefined
 LocalMachine Undefined
```

One or more of these might say "Restricted", but that's OK. Don't worry about that. Just so long as you keep this file around so you always know what the original settings are.

Set-ExecutionPolicy

Most of you, when you try to run this script, are giong to get an error that the script cannot run due to the execution policy setting. Unfortunately, we will have to temporarily change one of these settings to Unrestricted, run the script, then change the ExecutionPoligy back to it's default setting.

I want to make this very clear. This is normally **not** a good idea. Unfortunately, though, if you want to run this script, this is how it has to be for now, and I really hate saying that.

A very important thing to remember is that after you've finished running the script, you have the HTML output that you need, **change your settings back to what they originally**

were!

The command you need to run to use the script, and I grit my teeth as I type this is:

Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force

This is the point where you would run the script with your desired parameters. The details regarding this are discussed later in this document. For now, be aware that after you run the script, **change your execution policy settings back to what they originally were!** To do that, open the "Original-Policy-Settings.txt" file that you created earlier, and check the execution policy for the current user.

Note that you cannot set a policy back to undefined, so if it was Undefined before, change it to Restricted. If it was set to someting other than Undefined or Restricted, change it back to that. This is how:

Set-ExecutionPolicy Restricted -Scope CurrentUser -Force

If you want to read more detail on this command (and you should), you can type:

Get-Help Set-ExecutionPolicy -online

---

03. Running the RWExport-To-HTML.ps1 Script

Overview

The main script is named RWExport-To-HTML.ps1, and it must be run from within PowerShell.
It requires only two parameters to run: the name of the source file to convert, and the name of the destination file to create. There are, however, several other parameters that are optional, but very useful. These will be discussed futher below.
Before proceeding, make sure you have set your execution policy as discussed in the previous topic, "02. Using PowerShell"

Three Ways to Run the Script

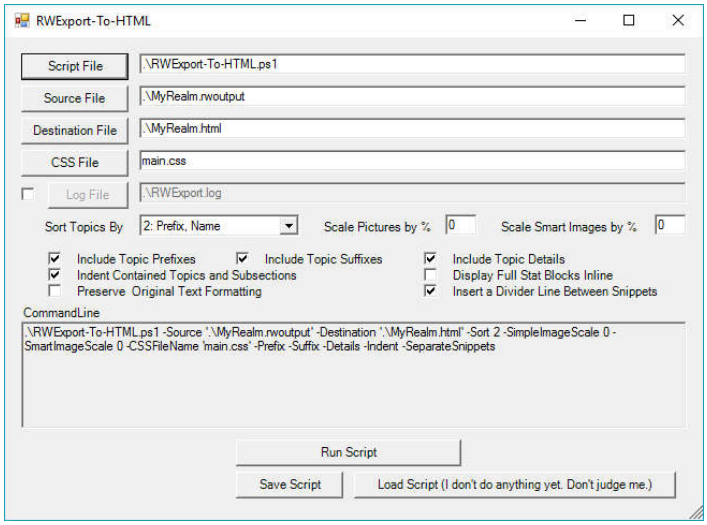
The most basic way to start the script is to right-click it, and select "Run with PowerShell".
You may be asked to confirm if you want to run the script. They only way it will work is if you click yes.
A new PowerShell window will open where you will be prompted for the name of a source file as well as for the name of a destination file. The source file is the file you exported from Realm Works.

The destination file is the file the script will create or overwrite as it converts the Realm Works eport file into HTML.
Once you enter the names of the source and destination files, the script will run. This method is quick and simple and easy, but it does not allow you to tailor the other parameters to your preference.

Use the Graphical User Interface

This is really just another PowerShell Script, but it's one that provides a graphical front end to the main script.
Right-click RWExportGUI.ps1, and select "Run with PowerShell".

Screenshot of RWExportGUI.ps1: [Picture]



Click the buttons, check boxes and drop-down lists to set everything according to your preferences, then either run the script or save it as its own script for later use.

Say, for example, that you always want the same options for your Pathfinder realm, and you always want the same options for your Savage Worlds realm, but you want the options to be different between Pathfinder and Savage Worlds.

You can set all the options you want for Pathfinder, then save that to a script called "Pathfinder-Exports.ps1". Then you can set all the options you want for Savage Worlds, then save it to something like "SavageWorlds-Exports.ps1"
Note that PowerShell scripts should always end with .ps1

After saving these two scripts, whenever you want to convert your Pathfinder export, you can right-click on "Pathfinder-Exports.ps1" and select "Run with PowerShell". And whenever you want to convert your Savage Worlds export, you can right-click on "SavageWorlds-Exports.ps1" and select "Run with PowerShell".

Since you've already defined the source and destination files and all the other parameters with the graphical interface, you will never be prompted for anything.

Do not save over the RWExport-To-HTML.ps1 script or the RWExportGUI.ps1 script!

From the PowerShell Window

This is the "under the hood" method. To tell you the truth, I'll probably be using the GUI myself from now on, but this documentation would not be complete without a full description of this method.

The Basic Script Command

As discussed in the previous topic, "02. Using PowerShell", open a PowerShell window, and change directory (CD) to where your script is saved. Then type the following command.

.\RWExport-To-HTML.ps1

You'll notice that the same thing happens as when you right-click and select "Run with PowerShell". You are prompted for a source file and a destination file. However, you can specity the source and destination file names on the same command line where you invoke the script. For example:

.\RWExport-To-HTML.ps1 -Source .\MyRealm.rwoutput -Destination .\MyRealm.html

In the case of this script, -Source and -Destination must always be the first two parameters listed.

There are, however, several other parameters you can use to define the nature and format of your output, and these can be included in any order after -Source and -Destination. Each command line parameter starts with a dash, and some require a value. If a command line parameter is listed as "-Source <filename>", that means it requires a file name as a value. Other types of values will be listed differently, but those will be explained for their respective parameters.
If a command line parameter is listed simply as "-Indent", that means it requires no value. Simply including the parameter activates the option.

Command Line Parameters

-Source <filename>

This specifies the name of the .rwexport file that you wish to convert to HTML. If there is a space in this filename or anywhere in its path, it should be enclosed in quotes. Enclosing the name in quotes anyway has no adverse effect, so enclosing all file names in quotes is perfectly valid.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport"

-Destination <filename>

This specifies the name of the .html file that you wish to create. If there is a space in this filename or anywhere in its path, it should be enclosed in quotes. Enclosing the name in quotes anyway has no adverse effect, so enclosing all file names in quotes is perfectly valid.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html"

-Sort <n>

Realm Works includes prefixes for topics. Prefixes can be used in any number of ways and can be helpful in organizing topics. Or not. With this option, that is entirely up to you. You can sort topics strictly by name, or by prefix and then by name. Or, if you wish to keep the same types of topics grouped together (all rooms or all people or whatever, this option can do that as well.

You can sort topics by:

1. Name
2. Prefix, Name
3. Category, Name
4. Category, Prefix, Name

If you do not specify an option, the default will be option 2 (Prefix, then Name).

Options 2 and 4 will sort by prefix, regardless of whether or not the -Prefix parameter is specified. Likewise, options 1 and 3 will sort by name regardless of whether or not the -Prefix switch is specified.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -Sort 3

This example will sort topics by category, then by name.

-Prefix

This parameter determines whether or not to include topic prefixes. Maybe you want to see them, and maybe you don't. If you do, include this on the command line.
For instance, if you had a topic named "Map Room" and you gave it a prefix of "D1" to indicate it was in dungeon #1, then without this parameter, the topic would be listed as "Map Room". If you include this parameter, the topic would be listed as "D1 - Map Room".

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html"

-Prefix

-Suffix

This parameter determines whether or not to include topic Suffixes. Maybe you want to see them, and maybe you don't. If you do, include this on the command line.
For instance, if you had a topic named "Map Room" and you gave it a suffix of "Topographical" to indicate it contained topographical maps, then without this parameter, the topic would be listed as "Map Room". If you include this parameter, the topic would be listed as "Map Room (Topographical)".

If you include both the Prefix and Suffix parameters, the topic would be listed as "D1 - Map Room (Topographical)"

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -Suffix

or

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -Prefix -Suffix

-Details

This parameter includes various topic details such as Category, Parent, Linkage, Tags, etc. This is basic information about each topic that can be useful for your own reference, but you would not necessarily want to include it in output that is intended for players' eyes.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -Details

-Indent

Realm Works allows or its own type of hierarchical structure. Topics can be contained within other topics which can be contained within other topics. Sections can also have subsections which can have their own subsections.

If you want your subtopics and subsections indented for an easy visual reference, you can use this parameter.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -Indent

-SeparateSnippets

Snippets can be long, or snippets can be short. In at least one case, there's a two-in-one snippet (GM Directions + Text). It can be difficult, sometimes, to tell where one snippet ends and another begins. If you want a visual cue for that, you can include this parameter to insert a thin separator line between each snippet.
If you have predominantly shorter snippets, using this option might make things look cluttered or visually distracting. But if you have predominantly longer snippets, this can add a nice aesthetic and cosmetic touch.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -SeparateSnippets

-InlineStats

By default, stat blocks are treated as attachments. You can view them by clicking a link, but they do not display inline. One the one hand, your output is less cluttered. On the other hand, if you want to print your document for offline reference, your document will be incomplete.

This parameter allows you to display stat blocks inline, in a unified document.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -InlineStats

-SimpleImgScale <n>

By default, images (in this case, simple pictures) are treated as attachments. They appear as thumbnails (with a clickable link to view them at full size), but they do not display inline except as a thumbnail. If you want to show them at larger scales, you can include this option to include the scale as a percentage of the width of the page.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -SimpleImgScale 50

This will show simple pictures at 50% of the width of the page, or less if the image is smaller.

-SmartImgScale <n>

By default, images (in this case, smart images) are treated as attachments. They appear as thumbnails (with a clickable link to view them at full size), but they do not display inline except as a thumbnail. If you want to show them at larger scales, you can include this option to include the scale as a percentage of the width of the page.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -SmartImgScale 25

This will show smart images at 25% of the width of the page, or less if the image is smaller. If you wanted to include scales for both simple pictures and smart images, you could do something like this:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -SimpleImgScale 50 -SmartImgScale 25

This will show simple pictures at 50% of the width of the page and smart images at 25%.

-KeepStyles

By default, inline style definitions are removed for font, font size, font color and font background color. This is done in order to provide more universal support for the CSS file to manage formatting. However, if you have your tables set up just the way you like them, and your bulleted lists are set up just the way you like them, then maybe you don't want this script stripping out your formatting.

Use this option to preserve your original formatting.

Example:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -KeepStyles

-CSSFilename <filename>

The main.css file is included with with this script and is defined in the header of the HTML file. If you wish to define a different file name in the HTML header file, you can use this parameter. Note that this will not create a CSS file. If the file name you specify does not exist, your HTML file will lack proper formatting.

You could use this option if you wanted to have different formatting for different outputs. You

can assign one css file to a Pathfinder output and another to a Savage Worlds output.

Examples:

.\RWExport-To-HTML.ps1 -Source ".\My Realm.rwexport" -Destination ".\My Realm.html" -CSSFilename "main.css"

.\RWExport-To-HTML.ps1 -Source ".\Pathfinder.rwexport" -Destination ".\Pathfinder.html" -CSSFilename "Pathfinder.css"

.\RWExport-To-HTML.ps1 -Source ".\Savage.rwexport" -Destination ".\Savage.html" -CSSFilename "Savage.css"

-SplitTopics

This will save each topic as a separate HTML file. Note that if you use this option, then you must specify a folder name for **-Destination** instead of a filename.

-ExtractFiles

This will extract embedded files, and save them each in their original format. Note that if you use this option, then you must specify a folder name for **-Destination** instead of a filename. Additionally, that folder must have a subfolder named *realm_files*.

Important:

- The folder name should already exist and should be empty (*except for the relevant CSS file)*.
- If you keep exporting to the same folder without clearing it out first, the files will accumulate.
- Nothing will be overwritten.
- If you forget to copy the CSS file into the folder, there will be no formatting in place when you view them.

The file name for each topic will be in the format of: *Prefix-topicname(suffix).html*

The prefix and suffix are included automatically to help prevent duplicate file names if the topic names are otherwise identical. If there is still an issue, then a number will be appended to each subsequent file name. For instance, if you have the following topics in your realm:

- Mr. - John Smith (Esquire)
- Dr. - John Smith (Alien)
- Dr. - John Smith (Alien)
- John Smith
- John Smith
- Dr. John Smith (Human)

Your filenames will be:

- Mr.-John Smith(Esquire).html
- Dr.-John Smith(Alien).html
- Dr.-John Smith(Alien)-1.html
- John Smith.html
- John Smith-1.html
- Dr.-John Smith(Human).html

Characters that are invalid for filenames will be replaced with an underscore. So if you have a topic with a prefix of "Level 1:" and a name of "Map Room", the file name will be Level1_-Map Room.html.

I don't expect this option to be commonly used, but it was mentioned on the forums that one could transfer a realm to Obsidian Portal by exporting one topic at a time, and using this script to convert each topic to HTML. That sounds horrendously tedious, so for anyone wha wants to do this, I hope I've made things easier for you.

-Log <filename>

This function is not yet fully implemented.

---

04. Output from Get-Help

Overview

NAME

.\RWExport-To-HTML.ps1

SYNOPSIS

RWExport-To-HTML.ps1

by EightBitz

Version 0.9.1b

2017-01-29, 10:00 AM CST

RWExport-To-HTML.ps1 transforms a Realm Works export file into a formatted HTML file. The formatted HTML file relies on a

"RWExport_091b.css" file for formatting information.

LICENSE:

This script is now licensed under the Creative Commons Attribution + Non-Commercial license.

Basically, that means:

-You are free to share and adapt the script.

-When sharing the script, you must give appropriate credit and indicate if changes were made.

Summary: https://creativecommons.org/licenses/by/4.0/

Legal Code: https://creativecommons.org/licenses/by/4.0/legalcode

IMPORTANT:

You will likely have to change your execution policy to run this script. You can do that with the following command:

Set-ExecutionPolicy RemoteSigned

You may have to be logged in as a local administrator to do this, or run at least use the "Run as Administrator" option

when opening PowerShell (you can do this with a right-click).

Please do NOT set the execution policy to "Unrestricted". If you still have issues after setting it to "RemoteSigned",

then you can open this script as a text file, select all, copy, and paste it into a new text file. Once you do that,

delete the old file, and rename the new one to "RWExport-To-HTML.ps1".

IMPORTANT:

The export from Realm Works must done with the "Compact Output" option. This script will likely not work with a "Full

Export".

You can still export your full realm if you like, but make sure you do so with the "Compact Output" option.

IMPORTANT:

Make sure that the HTML file and the RWExport_091b.css file are in the same directory, otherwise, the HTML file will have

no formatting.

For more information about this, type:

Get-Help .\RWExport-To-HTML.ps1 -full

And see the NOTES section.

SYNTAX

.\RWExport-To-HTML.ps1 [-Source] <String> [-Destination] <String>

[-Sort <Int32>] [-Prefix] [-Suffix] [-Details] [-Indent] [-SeparateSnippets] [-InlineStats] [-SimpleImageScale <Int32>]

[-SmartImageScale <Int32>] [-KeepStyles] [-ExtractFiles] [-CSSFileName <String>] [-SplitTopics] [-Log <String>]

[<CommonParameters>]

DESCRIPTION

RWExport-To-HTML.ps1 loads the XML file exported from Realm Works and transforms it into formatted HTML so it can be

printed or imported/pasted into other programs. The export from Realm Works must done with the Compact Output option.

PARAMETERS

-Source <String>

Enter the full path and filename for the source file (the Realm Works export file).

IMPORTANT: This must be the first parameter on the command line.

Required? true

Position? 2

Default value

Accept pipeline input? false

Accept wildcard characters? false

-Destination <String>

Enter the full path and filename for the destination file (the HTML output).

IMPORTANT: This must be the second parameter on the command line.

(As long as the source and destination parameters are the first two, the remaining parameters are optional and can be

used in any order.)

Required? true

Position? 3

Default value

Accept pipeline input? false

Accept wildcard characters? false

-Sort <Int32>

Choose your preferred sort order for exported topics.

1 = Name

2 = Prefix, Name **Default**

3 = Category, Name

4 = Category, Prefix, Name

Required? false

Position? named

Default value 2

Accept pipeline input? false

Accept wildcard characters? false

-Prefix [<SwitchParameter>]

Include this parameter to display the prefix for each topic. If you've entered prefixes for your topics in Realm

Works, this option will add them to the display in the HTML file in the form of "Prefix - Topic Name".

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-Suffix [<SwitchParameter>]

Include this parameter to display the suffix for each topic. If you've entered suffixes for your topics in Realm

Works, this option will add them to the display in the HTML file in the form of "Topic Name (Suffix)".

If you include both the Prefix and Suffix parameters, the result will be "Prefix - Topic Name (Suffix)".

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-Details [<SwitchParameter>]

Include this parameter to include topic details (Category, Parent, Linkage, Tags, etc ...)

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-Indent [<SwitchParameter>]

Include this parameter to indent nested topics and section headers.

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-SeparateSnippets [<SwitchParameter>]

Include this parameter to display a line between snippets.

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-InlineStats [<SwitchParameter>]

Display Statblocks inline.

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-SimpleImageScale <Int32>

Include this parameter to scale the display size, by percentage, of embedded simple pictures.

If you omit this parameter or if you set it to 0, only the thumbnail will display.

Required? false

Position? named

Default value 0

Accept pipeline input? false

Accept wildcard characters? false

-SmartImageScale <Int32>

Include this parameter to scale the display size, by percentage, of embedded smart images (usually maps).

If you omit this parameter or if you set it to 0, only the thumbnail will display.

Required? false

Position? named

Default value 0

Accept pipeline input? false

Accept wildcard characters? false

-KeepStyles [<SwitchParameter>]

By default, this script strips some (not all) formatting from the imported data. It does this to allow formatting to

be controlled by the the CSS file. Otherwise, the inline formatting will override the settings in the CSS file.

If you want to keep the original formatting, though, you can use this option.

Right now, the format options that are stripped are: font, font size, font color and background color.

Note that the CSS file has different definitions for regular snippet text, bulleted lists, numbered lists and tables.

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-ExtractFiles [<SwitchParameter>]

By default, this script creates one, single HTML file, with any file attachments encoded and embedded in the same,

single HTML file.

If you wish, you can override this with the -ExtractFiles parameter. Including this on the command line will tell the

script to save attachments as separate files.

If you include this option on the command line, you must specify a folder name for the destination instead of a file

name. The HTML file will be stored in the folder you specify, and each file will be stored in a subfolder name

"realm_files".

Also note that both the destination folder and the "realm_files" subfolder must already exist. The script will not

create them for you.

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-CSSFileName <String>

By default, the HTML output file looks for "RWExport_091b.css" to define its style. You can use this option to

specify a different name.

If you have two realms called Realm1 and Realm2, and you want each HTML output to have different fonts, font sizes or

colors, you can specify a name or "realma.css" for one file and "realmb.css" for the other.

Not that this option does not create the file. It just tells the HTML file which filename to look for.

For now, at least, you will have to manually copy RWExport_091b.css or some other css file you like and make whatever

changes you wish.

Required? false

Position? named

Default value RWExport_091b.css

Accept pipeline input? false

Accept wildcard characters? false

-SplitTopics [<SwitchParameter>]

Save each topic as a separate HTML file.

If you include this option on the command line, you must specify a folder name for the destination instead of a file

name. Each topic will be stored as an individual file within the folder you specify.

Also note that the folder must already exist. The script will not create it for you.

Required? false

Position? named

Default value False

Accept pipeline input? false

Accept wildcard characters? false

-Log <String>

Specify the path for an optional log file.

[NOT WORKING YET]

Required? false

Position? named

Default value

Accept pipeline input? false

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (http://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

The .rwexport file from a Realm Works export that was created with the "Compact Output" option. This can an export of a

full realm or a custom or partial export, just so long as it's made with the "Compact Output" option.

OUTPUTS

An HTML file that uses an external style sheet for formatting. The external style sheet should be named

"RWExport_091b.css" and should be in the same folder as the HTML output. The RWExport_091b.css file will not be generated

by this script, but should already exist.

NOTES

SORTING:

The specified sort order will occur regardless of whether or not prefixes are included.

If you sort by prefix, but do not include the -Prefix parameter, your topics will still be sorted by prefix, even

though the prefix won't be displayed.

Likewise, if you sort by Name, but do include the -Prefix parameter, your topics will still be sorted by name, even

though the prefix will be displayed.

It appears that topics will always be sorted under their containers. In other words, topics that are not in

containers will be sorted relative to each other. Contained topics will be sorted relative to their peers within that

container.

TIPS ON MANAGING COMMANDLINE OPTIONS:

There are a several commandline options for this script, and that can probably be intimidating to some people. Here

is my recommended starting point:

c:\<full path>\RWExport-To-HTML.ps1 -Source C:\<full path>\MyExport.rwoutput -Destination c:\<full

path>\MyHTML.html -Indent -Prefix -Suffix -Sort 2

2 is the default sort value, so if that's what you want, you don't have to specify it, but I'm doing so anyway just

to be clear as to what's happening in this example.

If you like the output from this combination of options, you don't have to rememeber to type this out all the time.

You can save it as it's own PowerShell script. Paste the command into it's own .ps1 file. Name it something like

"MyExportOptions.ps1". (Make sure to replace <full path> with the actual path for where the relevant files are.)

But you don't have to stop there. Say you have three different realms, and you want to specify

different options for

each. You can do something like this:

c:\<full path>\RWExport-To-HTML.ps1 -Source C:\<full path>\Pathfinder-Realm.rwoutput -Destination c:\<full

path>\Pathfinder-Realm.html -Indent -Prefix -Suffix -Sort 2

c:\<full path>\RWExport-To-HTML.ps1 -Source C:\<full path>\SavageWorlds-Realm.rwoutput -Destination c:\<full

path>\SavageWorlds-Realm.html -Indent -InlineStats

c:\<full path>\RWExport-To-HTML.ps1 -Source C:\<full path>\FATE-Realm.rwoutput -Destination c:\<full

path>\FATE-Realm.html -SimpleImageScale 25 -SmartImageScale 50

You can put all three of those command lines in the same PowerShell script and call it "ConvertAllMyExports.ps1".

Or you can put each command line in its own script called "ConvertPathfinderExport.ps1",

"ConvertSavageWorldsExport.ps1" and "ConvertFATEExport.ps1"

Now, when you want convert a given export, you don't have to remember all the command line options, because you

already have your favorite ones ready to go. You just run your PowerShell script instead of this one, and yours will

invoke this one with your favorite options.

As a side note, you can set up a separate folder for each export, and put a different RWExport_091b.css file in each

folder, so you can give each export a completely different look (by editing the RWExport_091b.css file in each

folder).

AND SPEAKING OF RWExport_091b.css:

This script and its resulting HTML file assume the existence of a file named "RWExport_091b.css". If this file is not

in the same folder as the resulting HTML file, it will not display correctly.

---------- EXAMPLE 1 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html

This is the most basic example.

This would give you the most basic output. The text would be formatted according to Title, Topic, Section and Snippet,

and that's about it. Nothing much more than that.

Topics will be sorted by prefix first, then by name.

Topics will be listed by name only with no prefixes or suffixes.

Nested topics and sections will not be indented. Everything will be left-justified.

Statblocks will not be displayed inline.

Simple Pictures and Smart Images will be displayed as thumbnails.

---------- EXAMPLE 2 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html -Sort 1

Choose your preferred sort order for exported topics. The example above will sort topics by their names.

Valid options are:

1 = Sort by topic names

2 = Sort by topic prefixes first, then by topic names (this is the default value)

3 = Sort by topic category first, then by topic name

4 = Sort by topic category first, then by topic prefix, then by topic name

---------- EXAMPLE 3 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html -Prefix

Prepend topic prefixes to topic names.

If a topic has a prefix of "Dungeon 1" and a name of "Room 3", you would include the Prefix parameter to list the topic

in the output as "Dungeon 1 - Room 3"

---------- EXAMPLE 4 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html -Suffix

Append topic suffixes to topic names.

If a topic has a suffix of "Mess Hall" and a name of "Room 3", you would include the Suffix parameter to list the topic

in the output as "Room 3 (Mess Hall)"

If you include the prefix and suffix parameter, the topic would be displayed as "Dungeon 1 - Room 3 (Mess Hall)"

---------- EXAMPLE 5 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html -Indent

To give your output just a little more style, you can choose to indent subsections and contained topics.

This looks really nice for documents that have content that might be nested 2 or 3 levels in, but for documents that

might have 4, 5 or 6 levels of nested topics and subsections, it might not look so great.

---------- EXAMPLE 6 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html -InlineStats

By default, stat blocks will not display inline, but will instead offer a clickable link where you can then see the full

stat block.

If you want everything in one view, though, without having to click, you can use the -InlineStats option.

Inline stat blocks will not always look as nice as you might like, but this parameter gives you the option to include

them if you like.

Also, if you include both -Indent and -InlineStats, the stat blocks will NOT indent. That prospect was fraught with too

many headaches.

As far as "looking nice", the clickable links are the better option, but if you want everything viewable in one document,

this gives you that option.

---------- EXAMPLE 7 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html -SimpleImageScale 50

Display simple images inline at 50% of the width of the page.

---------- EXAMPLE 8 ----------

PS C:\>RWExport-To-HTML.ps1 -Source MyExport.rwoutput -Destination MyHTML.html -SmartImageScale 50

Display smart images inline at 50% of the width of the page.

RELATED LINKS

---

05. Formatting Your Document with CSS

Overview

CSS stands for Cascading Style Sheets, and is a relatively easy way to change the format, appearance and style of an HTML file. When I say it's relatively easy, I mean relative to editing the HTML file itself. A few simple changes in a CSS file can change the whole look and feel of an HTML file whereas without using CSS, those same edits might be time consuming and tedious.

Telling you everything about CSS is both beyond the scope of this document and beyond the scope of my knowledge. So I'm just going to cover some basics as they apply to this script and its output.

Using the Right Editor

CSS files are plain-text files, but they have a certain structure. Do ***not*** edit a CSS file with Word or WordPad or any kind of a full-featured word processor.

A good program to use would be Notepad++

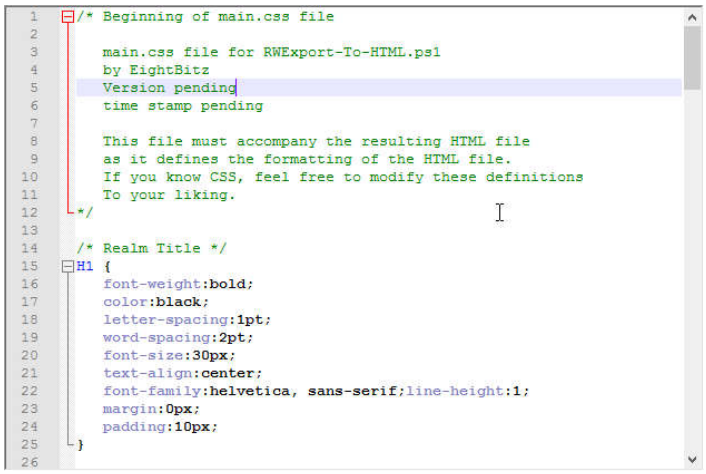
You can download it for free from here: https://notepad-plus-plus.org/

Download it, install it, and use it to edit CSS files.

Structure of the Included CSS File.

Once you open the included CSS file in Notepad++, it will look like this:

CSS Example in Notepad++: [Picture]



```
/* Beginning of main.css file

   main.css file for RWExport-To-HTML.ps1
   by EightBitz
   Version pending
   time stamp pending

   This file must accompany the resulting HTML file
   as it defines the formatting of the HTML file.
   If you know CSS, feel free to modify these definitions
   To your liking.
*/

/* Realm Title */
H1 {
   font-weight:bold;
   color:black;
   letter-spacing:1pt;
   word-spacing:2pt;
   font-size:30px;
   text-align:center;
   font-family:helvetica, sans-serif;line-height:1;
   margin:0px;
   padding:10px;
}
```

Notice the color-coding. This is one of the reasons why Notepad++ is such a nice program. It recognizes that this is a CSS file, understands its structure, and adds color codes for easy visual reference. None of the color coding is ever saved. It is only there for visual reference. If you make changes and save the file, the file will still be plain-text.

The first block of text, at the very top, the green block, starts with /* and ends with */. These characters indicate that anything between them is a comment, purely for informational purposes, and not to be interpreted as any type of format definition. That green block of text is there for your eyes only. Your computer ignores it.

After that block of text, the very next line is also a comment: /* Realm Title */
That comment is there to tell you what following format information applies to.

A block of format information starts with an HTML tag, then and open curly bracket, then a series of format specifications (each separated by a semi-colon), then a closed curly bracket.

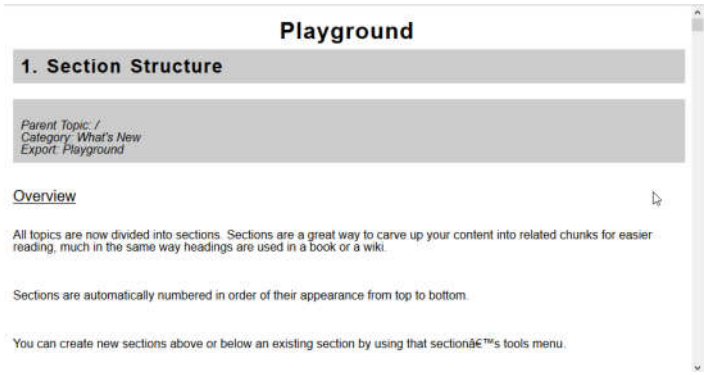
The HTML tag, in this case, is H1. That means that within the HTML file itself, any text between <H1> and </H1> will be formatted according to the information in this block. But H1 could mean anything. Without the /* Realm Title */ comment, you would have no specific way (without experimenting) to know that in the context of this script's output, H1 refers to the realm title.

You'll see similar comments if you scroll down, telling you which formatting blocks apply to topics, which apply to sections, and which apply to snippets. There are also separate blocks for bulleted lists, numbered lists, tables and stat blocks. They are all preceded by a comment line to let you know what type of thing a given format block applies to.

Making Changes

Let's look at an HTML export with the default format.

Default CSS Formatting: [Picture]



In all honesty, this isn't that exciting. The main reason I went with this is because my main interest is printing, and I don't have a color printer. But you may have a color printer. Or you may want to post this file on your web site. So let's play around a little.

I always like to keep the original settings before I change anything, so let's comment out what we're going to change. In the /* Realm Title */ title section, change the line that says "color:black;" to "/* color:black; */"

Now that's commented out. Add a couple of blank lines after it, and add these lines there:

color:white;

background-color:red;

Now find the following line: "font-size:30px;", and make that a comment as well, then add a couple of blank lines after it, and change it from 30px to 60px.

Your /* Realm Title */ section should now look something like this:

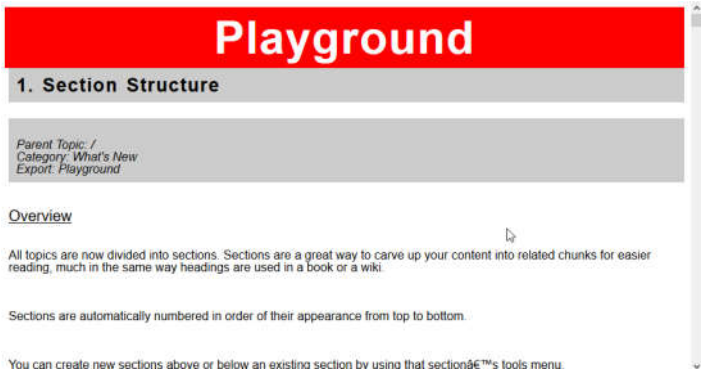
Edited CSS File: [Picture]

```
/* Realm Title */
H1 {
    font-weight:bold;
    /* color:black; */

    color:white;
    background-color:red;

    letter-spacing:1pt;
    word-spacing:2pt;
    /* font-size:30px; */

    font-size:60px;

    text-align:center;
    font-family:helvetica, sans-serif;line-height:1;
    margin:0px;
    padding:10px;
}
```

Note: Keeping things lined up with the same indentation is not functionally important, but it is visually helpful, so take a second or two to line things up.

Now save your changes and click your browser's reload button.

After our CSS edits.: [Picture]



You can clearly see the changes. Maybe the settings I chose came out to be a bit jarring. Let's try one more edit. Again, in the /* Realm Title */ section, change the following two lines:

color:white;

background-color:red;

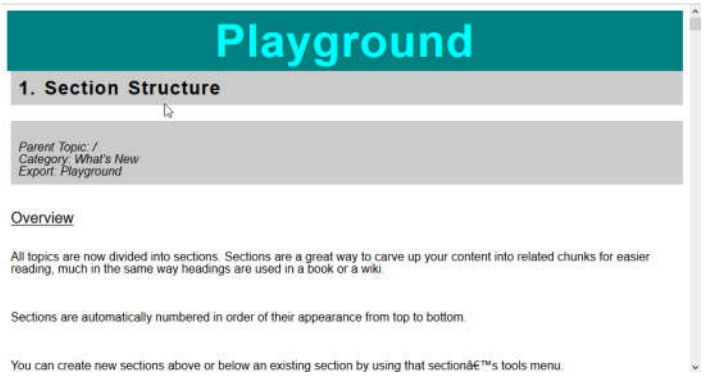
To

color:aqua;

background-color:teal;

Then save the CSS file and click your browser's reload button.

After our second CSS edit.: [Picture]
Annotation: And that's much easier on the eyes.



Those are the basics. You can change the formatting in the other sections in the same manner so things are to your liking. Again, there's much more you can do with CSS formatting. If you want to delve in deeper, here's a great place to start learning: http://www.w3schools.com/cssref/default.asp

---

Div Tags and Variables

If you scroll down a bit in the CSS file, you will see the following section headers:

- .snippet
- .snippet P
- .snippet UL
- .snippet OL

and a few others.

The very first one, .snippet, contains default values for the following sections. For example, .snippet contains the following variable: --main-font-size:16px;

.snippet P, .snippet UL, and .snippet OL (as well as following .snippet sections) all contain the following statement:

font-size:var(--main-font-size);

This means that if you change the variable in the very first .snippet section to: --main-font-size:20px; that change is made for all of the other .snippet sections that refer to the variable. With one change, you can modify the look of all those sections, as long as they refer to that variable.

Tips for Tables

If you use the -KeepStyles option, or as it's labeled in the GUI, "Preserve original text formatting," your tables may not display correctly, particularly the font color and background color. You can fix this in the CSS file.

Scroll down to the ".snippet TABLE TR TD P" section, and you will see the following few lines:

```
color:var(--main-color);
background-color:var(--main-background-color);

/* If you use the -KeepStyles option, and your tables don't look right,
Replace the two color and background-color lines above with the two below. */

/* color:transparent; */
/* background-color:transparent; */
```

Remember that /* and */ surround comments. Anything included in between will not be counted as formatting statements even if they are valid formatting statements. So you can change those lines by reversing the commenting. When you're done, it should look like this:

/* color:var(--main-color); */

/* background-color:var(--main-background-color); */

/* If you use the -KeepStyles option, and your tables don't look right,

Replace the two color and background-color lines above with the two below. */

color:transparent;

background-color:transparent;

That should display the table with your original colors.

---

06. Default CSS File

Overview

As of version 0.9a, the default CSS file has been renamed from main.css to RWExport_09a.css. This is due to structural changes within the CSS file and to avoid compatibility issues with HTML files that were created with older versions of this script.

RWExport_091b.css

/* Beginning of main.css file

main.css file for RWExport-To-HTML.ps1

by EightBitz

Version 0.9.1b

2017-01-29, 10:00 AM CST

This file must accompany the resulting HTML file

as it defines the formatting of the HTML file.

If you know CSS, feel free to modify these definitions

To your liking.

*/

```
/* Realm Title */
H1 {
  font-weight:bold;
  color:white;
  background: #466368;
  background: radial-gradient(#648880, #293f50);
  letter-spacing:1pt;
  word-spacing:2pt;
  font-size:60px;
  text-align:center;
  font-family:helvetica, sans-serif;line-height:1;
  margin:0px;
  padding:10px;
}

/* Realm Details */
H2 {
  font-weight:normal;
  font-style:italic;
  color:black;
  background-color:white;
  letter-spacing:1pt;
  word-spacing:2pt;
  font-size:16px;
  text-align:center;
  font-family:helvetica, sans-serif;line-height:1;
  margin:0px 200px 0px 200px;
  padding:10px;
}
```

```
/* Topic Name */
H3 {
  font-weight:bold;
  color:white;
  background: #466368;
  background: linear-gradient(to right bottom, #648880, #293f50);
  letter-spacing:1pt;
  word-spacing:2pt;
  font-size:25px;
  text-align:left;
  font-family:helvetica, sans-serif;line-height:1;
  margin:0px;
  padding:10px;
}

/* Topic Details (Category, Parent, Linkage, Tags, etc ...) */
H4 {
  font-weight:normal;
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  font-style:italic;
  color:black;
  background-color:#CCCCCC;
  padding:10px;
}

/* Section Header */
H5 {
  font-weight:normal;
```

```
   text-decoration:underline;
   font-family:helvetica, sans-serif;line-height:1;
   font-size: 20px;
   background:white;
   color: black;
}

/* Snippet Text */
.snippet {
/* variables for all snippet elements */
   --main-font-family:helvetica, sans-serif;
   --main-font-size:16px;
   --main-font-weight:normal;
   --main-font-style:normal;
   --main-text-decoration:normal;
   --main-color:black;
   --main-background-color:white;
   --main-line-height:1;
   --main-padding-top:0px;
   --main-padding-right:0px;
   --main-padding-bottom:0px;
   --main-padding-left:0px;
}

.snippet P {
/* unformatted snippet text */
   font-family:var(--main-font-family);
   font-size:var(--main-font-size);
   font-weight:var(--main-font-weight);
   font-style:var(--main-font-style);
```

```
  text-decoration:var(--text-decoration);
  color:var(--main-color);
  background-color:var(--main-background-color);
  line-height:var(--main-line-height);
  padding-top:var(--main-padding-top);
  padding-right:var(--main-padding-right);
  padding-bottom:var(--main-padding-bottom);
  padding-left:var(--main-padding-left);
}

.snippet UL {
/* bulleted lists in snippets */
  /* If your bulleted lists are not indented to your liking,
     you can uncomment the following two lines. */

  list-style-position: inside;
  margin-left:20px;

  /* You can also change the 20px to your liking. */
  font-family:var(--main-font-family);
  font-size:var(--main-font-size);
  font-weight:var(--main-font-weight);
  font-style:var(--main-font-style);
  text-decoration:var(--text-decoration);
  color:var(--main-color);
  background-color:var(--main-background-color);
  line-height:var(--main-line-height);
  padding-top:var(--main-padding-top);
  padding-right:var(--main-padding-right);
  padding-bottom:var(--main-padding-bottom);
```

```
    padding-left:var(--main-padding-left);
}

.snippet OL {
/* numbered lists in snippets */
    list-style-position: inside;
    margin-left:20px;
    font-family:var(--main-font-family);
    font-size:var(--main-font-size);
    font-weight:var(--main-font-weight);
    font-style:var(--main-font-style);
    text-decoration:var(--text-decoration);
    color:var(--main-color);
    background-color:var(--main-background-color);
    line-height:var(--main-line-height);
    padding-top:var(--main-padding-top);
    padding-right:var(--main-padding-right);
    padding-bottom:var(--main-padding-bottom);
    padding-left:var(--main-padding-left);
}

.snippet TABLE {
/* tables within snippets */
    font-family:var(--main-font-family);
    font-size:var(--main-font-size);
    font-weight:var(--main-font-weight);
    font-style:var(--main-font-style);
    text-decoration:var(--text-decoration);
    color:var(--main-color);
    background-color:var(--main-background-color);
```

```
    line-height:0;
    padding-top:var(--main-padding-top);
    padding-right:var(--main-padding-right);
    padding-bottom:var(--main-padding-bottom);
    padding-left:var(--main-padding-left);
}

.snippet TABLE TR {
/* cells within snippet tables */
    font-family:var(--main-font-family);
    font-size:var(--main-font-size);
    font-weight:var(--main-font-weight);
    font-style:var(--main-font-style);
    text-decoration:var(--text-decoration);
    color:var(--main-color);
    background-color:var(--main-background-color);
    line-height:0;
    padding-top:var(--main-padding-top);
    padding-right:var(--main-padding-right);
    padding-bottom:var(--main-padding-bottom);
    padding-left:var(--main-padding-left);
}

.snippet TABLE TR TD {
/* text within table cells */
    font-family:var(--main-font-family);
    font-size:var(--main-font-size);
    font-weight:var(--main-font-weight);
    font-style:var(--main-font-style);
    text-decoration:var(--text-decoration);
```

```
    color:var(--main-color);
    background-color:var(--main-background-color);
    line-height:0;
    padding-top:var(--main-padding-top);
    padding-right:var(--main-padding-right);
    padding-bottom:var(--main-padding-bottom);
    padding-left:var(--main-padding-left);
}

.snippet TABLE TR TD P {
    font-family:var(--main-font-family);
    font-size:var(--main-font-size);
    font-weight:var(--main-font-weight);
    font-style:var(--main-font-style);
    text-decoration:var(--text-decoration);
    color:transparent;
    background-color:transparent;
    line-height:0;
    padding-top:var(--main-padding-top);
    padding-right:var(--main-padding-right);
    padding-bottom:var(--main-padding-bottom);
    padding-left:var(--main-padding-left);
}
/* The following items are for displaying inline stat blocks. */
.StatBlockLink {
    color:#0000FF;
    background-color:transparent;
    text-decoration: underline;

    font-family:var(--main-font-family);
```

```
    font-size:var(--main-font-size);
    font-weight:var(--main-font-weight);
    font-style:var(--main-font-style);
    line-height:var(--main-line-height);
    padding-top:var(--main-padding-top);
    padding-right:var(--main-padding-right);
    padding-bottom:var(--main-padding-bottom);
    padding-left:var(--main-padding-left);
}

.StatBlockSnippet {
    font-family:var(--main-font-family);
    font-size:var(--main-font-size);
    font-weight:var(--main-font-weight);
    font-style:var(--main-font-style);
    text-decoration:var(--text-decoration);
    color:var(--main-color);
    background-color:var(--main-background-color);
    line-height:var(--main-line-height);
    padding-top:var(--main-padding-top);
    padding-right:var(--main-padding-right);
    padding-bottom:var(--main-padding-bottom);
    padding-left:var(--main-padding-left);
}

.StatBlockDefault {
    text-align:left;
    text-indent:0pt;
    margin:0pt 0pt 0pt 0pt
}
```

```
.StatBlockBullet {
   text-align:left;
   text-indent:-7pt;
   margin:0pt 0pt 0pt 7pt}

.StatBlockEnumerated {
   text-align:left;
   text-indent:-12pt;
   margin:0pt 0pt 0pt 12pt
}

.StatBlock-td {
   border-style:solid;
   border-width: 1px;
   border-color: #a9a9a9;
   padding: 1.5pt 4pt;
}

.StatBlock-tr.naked td {border-style:none;}
.StatBlock-td.left {text-align:left;}
.StatBlock-td.center {text-align:center;}
.StatBlock-td.right {text-align:right;}

.StatBlock-p.footer {margin-left:0.75em;
   text-indent: -0.75em;
   margin-top: 0;
   margin-bottom: 0;
}
```

```
.StatBlock-p.smallgap {margin: 3pt 0 0 0;}
.StatBlock-p.mediumgap {margin: 6pt 0 0 0;}
/* End of main.css file */
```

RWExport_09a.css

```
/* Beginning of main.css file

   RWExport_09a.css file for RWExport-To-HTML.ps1
   by EightBitz
   Version 0.9a
   2017-01-26, 4:00 AM CST

   This file must accompany the resulting HTML file
   as it defines the formatting of the HTML file.
   If you know CSS, feel free to modify these definitions
   To your liking.
*/

/* Realm Title */
H1 {
   font-weight:bold;
   color:black;
   letter-spacing:1pt;
   word-spacing:2pt;
   font-size:30px;
   text-align:center;
   font-family:helvetica, sans-serif;line-height:1;
   margin:0px;
   padding:10px;
```

```
}

/* Realm Details */
H2 {
   font-weight:normal;
   font-style:italic;
   color:black;
   background-color:white;
   letter-spacing:1pt;
   word-spacing:2pt;
   font-size:16px;
   text-align:center;
   font-family:helvetica, sans-serif;line-height:1;
   margin:0px 200px 0px 200px;
   padding:10px;
}

/* Topic Name */
H3 {
   font-weight:bold;
   color:black;
   background-color:#CCCCCC;
   letter-spacing:1pt;
   word-spacing:2pt;
   font-size:25px;
   text-align:left;
   font-family:helvetica, sans-serif;line-height:1;
   margin:0px;
   padding:10px;
}
```

```
/* Topic Details (Category, Parent, Linkage, Tags, etc ...) */
H4 {
  font-weight:normal;
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  font-style:italic;
  color:black;
  background-color:#CCCCCC;
  padding:10px;
}

/* Section Header */
H5 {
  font-weight:normal;
  text-decoration:underline;
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 20px;
  background:white;
  color: black;
}

/* Snippet Text */
P {
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  background:white;
  color:black;
}
```

```
/* Bulleted Lists */
UL {
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  background:white;
  color:black;
}

/* Numbered Lists */
OL {
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  background:white;
  color:black;
}

/* Text in Tables */
TABLE TR TD P {
  font-family:helvetica, sans-serif;line-height:1;
  background:white;
  color:black;
  font-size:16px;
}

/* The following items are for displaying inline stat blocks. */
.StatBlockLink {
  color:#0000FF;
  background-color:transparent;
  font-family:Tahoma;
  font-size:12pt;
```

```
    font-weight:normal;
    font-style:normal;
    text-decoration: underline;
}

.StatBlockSnippet {
    color:#000000;
    background-color:transparent;
    font-family:Tahoma;
    font-size:12pt;
    font-weight:normal;
    font-style:normal;
}

.StatBlockDefault {
    text-align:left;
    text-indent:0pt;
    margin:0pt 0pt 0pt 0pt
}

.StatBlockBullet {
    text-align:left;
    text-indent:-7pt;
    margin:0pt 0pt 0pt 7pt}

.StatBlockEnumerated {
    text-align:left;
    text-indent:-12pt;
    margin:0pt 0pt 0pt 12pt
}
```

```
.StatBlock-td {
  border-style:solid;
  border-width: 1px;
  border-color: #a9a9a9;
  padding: 1.5pt 4pt;
}

.StatBlock-tr.naked td {border-style:none;}
.StatBlock-td.left {text-align:left;}
.StatBlock-td.center {text-align:center;}
.StatBlock-td.right {text-align:right;}

.StatBlock-p.footer {margin-left:0.75em;
  text-indent: -0.75em;
  margin-top: 0;
  margin-bottom: 0;
}

.StatBlock-p.smallgap {margin: 3pt 0 0 0;}
.StatBlock-p.mediumgap {margin: 6pt 0 0 0;}
/* End of main.css file */
```

---

main.css

```
/* Beginning of main.css file

  main.css file for RWExport-To-HTML.ps1
  by EightBitz
  Versions up to, but not beyond 0.8a
```

```
   2017-01-24, 05:00 AM CST

   This file must accompany the resulting HTML file
   as it defines the formatting of the HTML file.
   If you know CSS, feel free to modify these definitions
   To your liking.
*/

/* Title */
H1 {
   font-weight:bold;
   color:black;
   letter-spacing:1pt;
   word-spacing:2pt;
   font-size:30px;
   text-align:center;
   font-family:helvetica, sans-serif;line-height:1;
   margin:0px;
   padding:10px;
}

/* Topic Name */
H2 {
   font-weight:bold;
   color:black;
   background-color:#CCCCCC;
   /* color:white;
   background-color:teal; */
   letter-spacing:1pt;
   word-spacing:2pt;
```

```
  font-size:25px;
  text-align:left;
  font-family:helvetica, sans-serif;line-height:1;
  margin:0px;
  padding:10px;
}

/* Topic Details (Category, Parent, Linkage, Tags, etc ...) */
H3 {
  font-weight:normal;
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  font-style:italic;
  color:black;
  background-color:#CCCCCC;
  /* color:white;
  background-color:blue; */
  padding:10px;
}

/* Section Header */
H4 {
  font-weight:normal;
  text-decoration:underline;
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 20px;
  background:white;
  color: black;
}
```

```
/* Snippet Text */
P {
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  background:white;
  color:black;
}

/* Bulleted Lists */
UL {
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  background:white;
  color:black;
}

/* Numbered Lists */
OL {
  font-family:helvetica, sans-serif;line-height:1;
  font-size: 16px;
  background:white;
  color:black;
}

/* Text in Tables */
TABLE TR TD P {
  font-family:helvetica, sans-serif;line-height:1;
  background:white;
  color:black;
  font-size:16px;
```

```
}

/* The following items are for displaying inline stat blocks. */
.StatBlockLink {
   color:#0000FF;
   background-color:transparent;
   font-family:Tahoma;
   font-size:12pt;
   font-weight:normal;
   font-style:normal;
   text-decoration: underline;
}

.StatBlockSnippet {
   color:#000000;
   background-color:transparent;
   font-family:Tahoma;
   font-size:12pt;
   font-weight:normal;
   font-style:normal;
}

.StatBlockDefault {
   text-align:left;
   text-indent:0pt;
   margin:0pt 0pt 0pt 0pt
}

.StatBlockBullet {
   text-align:left;
```

```
   text-indent:-7pt;
   margin:0pt 0pt 0pt 7pt}

.StatBlockEnumerated {
   text-align:left;
   text-indent:-12pt;
   margin:0pt 0pt 0pt 12pt
}

.StatBlock-td {
   border-style:solid;
   border-width: 1px;
   border-color: #a9a9a9;
   padding: 1.5pt 4pt;
}

.StatBlock-tr.naked td {border-style:none;}
.StatBlock-td.left {text-align:left;}
.StatBlock-td.center {text-align:center;}
.StatBlock-td.right {text-align:right;}

.StatBlock-p.footer {margin-left:0.75em;
   text-indent: -0.75em;
   margin-top: 0;
   margin-bottom: 0;
}

.StatBlock-p.smallgap {margin: 3pt 0 0 0;}
.StatBlock-p.mediumgap {margin: 6pt 0 0 0;}
/* End of main.css file */
```

07. Release Notes

Overview

Release Notes for RWExport-To-HTML.ps1

by EightBitz

Unnamed

RWExportGUI.ps1

- **There is a new checkbox for a new command line option.**
- Previously, after the "Run Script" button was clicked, the GUI window was set to inactive to indicate the script was running. When the script finished, the form was reactivated.
- Apparently, deactivating the whole form was causing some issues for at least one person, so now it's only the "Run Script" button that is deactivated and reactivated.
- Modified the "CSS File" button so it truncates the path name and just leaves the file name. You can still manually edit it to be whatever you wish.

New Command Line Option:

- -ExtractFiles

Changes to existing command line options:

- -Log

-ExtractFiles

This will extract embedded files, and save them each in their original format. Note that if you use this option, then you must specify a folder name for **-Destination** instead of a filename. Additionally, that folder must have a subfolder named *realm_files*.

-Log

This option now works. Add it to your command line, followed by a file name. The result will be a log file that contains version info, start time, the list of topics and sections that were processed, the completion time, and the total run time.

Most errors that occur while processing topics, sections and snippets will also be written to the log.

CSS Changes:

The CSS file has been modified to use div tags and to use variables. For more information, see "05. Formatting Your Document with CSS"

Other Improvements

- All embedded and extracted files for a known snippet type (including "Foreign Object") now have clickable links to open the embedded or extracted files.
- The above is a result of improved handling of MIME types when dealing with embedded files.
- Apparently, thumbnails for smart images were broken. I'm not sure if that happened in the previous version or if that happened as a result of changes I was making to this version, but if you've experienced that issue, it should now be fixed.
- As Parody informed me on the forums, I was missing a "<!DOCTYPE html>" at the beginning of my HTML output. That is now fixed.
- I removed a few chunks of obsolete code (much of which was made obsolete by the use of div tags in the CSS file).
- Error trapping for most errors that occur while processing topics, sections and snippets.
- Unknown snippet types, instead of being ignored, will now be listed as "[Unknown Snippet Type]". A warning will appear in the log as well.

Version 0.9a

RWExportGUI.ps1

- Applied a fix to enclose all file paths in quotes to avoid errors regarding spaces in folder names or file names.

Changes to existing command line options:

- -SimpleImgScale
- -SmartImgScale

When I first implemented these options, they were working fine...sort of. But somewhere between then and now, they stopped working at all.

Now I have them working again, but the way I had them working wasn't really doing what I intended. It was scaling the size of the photo, but it was still leaving white space all around it equivalent to the full size of the photo.

That seems rather pointless, so I changed how this works. It now scales the width relative to the size of the area available. In other words, -SimpleImageScale 50 will tell the image to take up 50% (or less, if the image is smaller) of the width of the page.

New command line option:

- -SplitTopics
- Updated RWExportGUI.ps1 to accommodate this new option.

-SplitTopics

This will save each topic as a separate HTML file. Note that if you use this option, then you must specify a folder name for **-Destination** instead of a filename.

Important:

- The folder name should already exist and should be empty (*except for the relevant CSS file)*.
- If you keep exporting to the same folder without clearing it out first, the files will accumulate.
- Nothing will be overwritten.
- If you forget to copy the CSS file into the folder, there will be no formatting in place when you view them.

The file name for each topic will be in the format of: *Prefix-topicname(suffix).html*

The prefix and suffix are included automatically to help prevent duplicate file names if the topic names are otherwise identical. If there is still an issue, then a number will be appended to each subsequent file name. For instance, if you have the following topics in your realm:

- Mr. - John Smith (Esquire)
- Dr. - John Smith (Alien)
- Dr. - John Smith (Alien)
- John Smith
- John Smith
- Dr. John Smith (Human)

Your filenames will be:

- Mr.-John Smith(Esquire).html
- Dr.-John Smith(Alien).html
- Dr.-John Smith(Alien)-1.html
- John Smith.html
- John Smith-1.html
- Dr.-John Smith(Human).html

Characters that are invalid for filenames will be replaced with an underscore. So if you have a topic with a prefix of "Level 1:" and a name of "Map Room", the file name will be Level1_-Map Room.html.

I don't expect this option to be commonly used, but it was mentioned on the forums that one could transfer a realm to Obsidian Portal by exporting one topic at a time, and using this script to convert each topic to HTML. That sounds horrendously tedious, so for anyone wha wants to do this, I hope I've made things easier for you.

Other CSS changes:

- I changed the structure of the default CSS file to accomodate a new section for Realm Details (see Other Improvements).

- I've renamed the default CSS file to RWExport_09a.css to avoid compatibility issues with existing HTML files created with older versions of this script.
- Any new HTML files created with this script will look for the new CSS file.

Other Improvements:

- Added Aliases to the Topic Details section.
- Added all the information you fill out when you export your realm (Summary, Description, Requirements, Credits, Legal Text, Additional Notes and even the Cover Art).

Version 0.8a (2017-01-24, 05:00 AM CST)

RWExportGUI.ps1:

- I've put together another script that serves as a GUI front end for the main script.
- The code for the GUI script was mostly borrowed and adapted. The original source had the following header:
- # Code Generated By: SAPIEN Technologies PrimalForms (Community Edition) v1.0.8.0
- # Generated On: 7/3/2011 11:35 AM
- # Generated By: sean.kearney

Licensing:

- RWExport-To-HTML.ps1 is now licensed under the Creative Commons Attribution license
- (I'm removing the non-commercial restriction.)
- Summary: https://creativecommons.org/licenses/by/4.0/
- Legal Code: https://creativecommons.org/licenses/by/4.0/legalcode

Basically, that means:

- -You are free to share and adapt the script.
- -When sharing the script, you must give appropriate credit and indicate if changes were made.

New command line options:

- -KeepStyles
- -CSSFileName

-KeepStyles

- By default, text formatting is partially stripped. This is done to allow more uniform controll of text formatting through the CSS file.
- If you wish to preserve your original formatting, include -KeepStyles on your command line.
- Right now, the affected formatting elements are: font, font size, font color and background color.
- Other formatting (bold, italic, underline, etc) will be preserved.

-CSSFileName

- If you want to define a different CSS filename for each export, you can do that now.
- Include -CSSFileName on your command line, followed by the name of the file.
- Note that this option does NOT create the file. It merely specifies in the header of the HTML output. You will have to copy the main.css file, and rename the copy accordingly.

Stat Blocks:

- HTML stat blocks are now also controlled by CSS. I was able to figure how to strip the previously defined formatting without breaking all the other formatting.

Other CSS Changes:

- Tables are newly stripped of the previously defined formatting as well.
- A new section for tables has been added to the main.css file to accomodate.

Other Improvements:

- Addressed a cosmetic issue where "Label:" and "Annotation:" were displaying when there was no accompanying text.
- Now, if a label is empty, you won't see "Label:" on an otherwise empty line, and if an annotation is empty, you won't see "Annotation:" on an otherwise empty line.

Issues:

- I looked into an issue where Wingdings characters were not displaying properly, only to find out that supporting Wingdings is not part of the HTML Standard.
- There are unicode equivalents, but building that translation table, and parsing the XML code for where to place substititons is more work than I want to do right now, and I don't think it's going to be a common enough issue to make it worthwhile.
- The logging option is still not really functional.

Bug Fixes:

- It looks like the "Stream was not readable" error is indeed fixed. Since applying the fix in the last version, I have not seen it reappear.

Version 0.7a (2017-01-21, 06:40 AM CST)

Licensing:

- This script is now licensed under the Creative Commons Attribution + Non-Commercial license.
- Summary: https://creativecommons.org/licenses/by-nc/4.0/

- Legal Code: https://creativecommons.org/licenses/by-nc/4.0/legalcode

Basically, that means:

- You are free to share and adapt the script.
- When sharing the script, you must give appropriate credit and indicate if changes were made.
- You may not use the material for commercial purposes.

New command line options:

- -Details
- -SeparateSnippets

-Details

- Topic details (Category, Parent, Linkage, Tags, etc ...) are no longer included by default. The topic details are nice for reference, but the document looks more presentable without them. If you're planning to share or publish, you may not want them in there at all. However, if you do, you can use the -Details option to put them back in.

-SeparateSnippets

- Some snippets are longer, and some are shorter. Some are grouped and some are not. If you want a visual separator to show where one snippet ends and another begins, you can use the -SeparateSnippets option.

Stat Blocks:

- Inline statblocks are a lot cleaner now. They display the way they're supposed to display.
- For non-HTML stat blocks, you can change their formatting in the main.css file.
- HTML stat blocks are an exception, because they have inline CSS formatting. At this point I am concerned about the unintended consequences of programmatically stripping that out, regardless of how good my intentions may be.
- If I find a good way to include affet HTML stat blocks, I will make that change. That being said, HTML stat blocks display fine. They're just unaffected by changes in the CSS file.
- The original font size for stat blocks was 9 points. I had to squint to read that, so in the main.css file, I changed it to 12. If you click the link for the stat block, you'll see it in its original form, and you'll see the difference. If you like it at 9 points, you can easily make that change in the main.css file.

Other CSS Changes:

- I discovered that lists were not effected by the definition for regular snippets, so I added two new CSS definitions. One for bulleted lists and one for numbered lists

Other Improvements:

- I added input validation and error checking to make sure that the script does not attempt to run with invalid options. If an input file cannot be read, or an output file cannot be written, or a sort value is out of range, the script will report the error and exit.
- This is prelude for better error trapping overall and for proper logging of errors to aid in future troubleshooting and debugging. The error trapping and logging were what I was originally planning to include in this release, but one thing after another took hold, and I figured people would appreciate the new changes and functionality.

Bug fixes:

- Fixed an issue where snippets that contain both GM directions and regular text were not being properly handled.
- Fixed some superficial, but nagging bugs. If you previously got a stream of errors reporting that certain functions or methods could not be called on a null value, those should be fixed in this version.
- For display purposes, I was appending a colon (:) to snippets that had labels that did not already end with a colon. I fixed an issue were the colon was showing up even if there was no text in the label.
- I have hopefully fixed an intermittent "Stream was not readable" error. It looks like the fix worked, but since it was intermittent, I can't say for sure. All I can say is that I have not had the error since applying a suggested fix.

Get-Help:

- I have updated the Get-Help info as well, so you can see all the available command line options, what they do, and some examples of how to use them.
- To use Get-Help, open a PowerShell window, and CD to the folder where you've saved the script. Then type any one of the following commands:
- Get-Help .\RWExport-To-HTML.ps1
- Get-Help .\RWExport-To-HTML.ps1 -examples
- Get-Help .\RWExport-To-HTML.ps1 -detailed
- Get-Help .\RWExport-To-HTML.ps1 -full
- Furthermore, you can send the results of any one of those commands to a text file.
- Get-Help C:\<full path>\RWExport-To-HTML.ps1 -full > Full-Help-for-RWExport.txt
- That will create the indicated text file that will contain the results of the Get-Help command, and you can just keep that handy somewhere.

Version 0.6a (2017-01-19, 04:40 AM CST)

SHOW ALL THE SNIPPETS:

- Added at least a listing of all snippet types. If the snippet is there, you will at least see its name and type.

IMAGES:

- Added support for displaying images. Thumbnails display by default, but you can display a larger view with two new command line options.
- -SimpleImageScale 20
- -SmartImageScale 75
- These work by percentages. So with the above examples, simple pictures will display at 20%

of their full size, and smart images at 75%.
- In both cases, the image name will be followed by a clickable link to display the image in full size.

STAT BLOCKS:

- Stat blocks can now be viewed. By default, there is a clickable link after the name. If you would prefer, there is an option to display stat blocks inline.

-InlineStats

- It's not always going to look pretty, but it's there. The clickable links will usually display cleaner versions.

HELP! WHAT DO I DO! HELP!

- I polished the Get-Help info. Any time you want to know what options are available and how they work, you can use the following commands:
- Get-Help C:\<full path>\RWExport-To-HTML.ps1
- Get-Help C:\<full path>\RWExport-To-HTML.ps1 -examples
- Get-Help C:\<full path>\RWExport-To-HTML.ps1 -detailed
- Get-Help C:\<full path>\RWExport-To-HTML.ps1 -full
- Furthermore, you can send the results of any one of those commands to a text file.
- Get-Help C:\<full path>\RWExport-To-HTML.ps1 -full > Full-Help-for-RWExport.txt
- That will create the indicated text file that will contain the results of the Get-Help command, and you can just keep that handy somewhere.

Version 0.5a (RWExport-to-HTML.ps1)

These are the release notes for the HTML version.

- Most everything is the same, which is why I'm including the previous release notes for the text version.

The differences here are:

- The obvious: HTML formatting.
- With the HTML formatting comes and added CSS file, main.css.
- main.css should always be in the same folder as the HTML output, otherwise the output will not be properly formatted.
- There is also now an -Indent option which, if invoked, will indent nested topics and sections.
- If you lose your main.css file, see the sample below:

/* Beginning of main.css file */

```
/*
This file must accompany the resulting HTML file
as it defines the formatting of the HTML file.
If you know CSS, feel free to modify these definitions
To your liking.
*/

/* Title */
H1 {
   font-weight:bold;
   color:#000000;
   letter-spacing:1pt;
   word-spacing:2pt;
   font-size:30px;
   text-align:center;
   font-family:helvetica, sans-serif;line-height:1;
   margin:0px;
   padding:10px;
}

/* Topic Name */
H2 {
   font-weight:bold;
   color:#000000;
   background-color:#CCCCCC;
   letter-spacing:1pt;
   word-spacing:2pt;
   font-size:25px;
   text-align:left;
   font-family:helvetica, sans-serif;line-height:1;
```

```
   margin:0px;
   padding:10px;
}

/* Topic Details (Category, Parent, Linkage ...)*/
H3 {
font-weight:normal;
   font-family:helvetica, sans-serif;line-height:1;
   font-size: 16px;
   font-style: italic;
   color: black;
   background-color:#CCCCCC;
   padding:10px;
}

/* Section Header */
H4 {
font-weight:normal;
text-decoration:underline;
   font-family:helvetica, sans-serif;line-height:1;
   font-size: 20px;
   background: #ffffff;
   color: black;
}

/* Snippet */
P {
   font-family:helvetica, sans-serif;line-height:1;
   font-size: 16px;
   background: #ffffff;
```

```
   color: black;
}
/* End of main.css file */
```

Version 0.5a (RWExport-to-Text)

- Added a -Prefix switch to optionally include a topic's prefix.
- Added a -Suffix switch to optionally include a topic's suffix.
- Added a -Sort option to sort topics by:
- 1 = Name
- 2 = Prefix, Name **Default**
- 3 = Category, Name
- 4 = Category, Prefix, Name
- Choosing options 2 or 4 will sort by prefix, regardless of whether or not the -Prefix switch is specified. Likewise, choosing options 1 or 3 will sort by name, regardless of whether or not the -Prefix switch is specified.
- Added parentage for topics.
- Made the topic suffix parenthetical so it's consistent with the display in RW.
- Better (I hope) parsing of snippets, and support for more types.

Supported snippet types include:

- Text (Including lists and tables, but NOT including any formatting such as bold, italic, highlights, etc.)
- GM Directions
- Labeled Text
- Tags
- Calendar Date
- Calendar Date Range
- Numeric Value
- Tags (MultiDomain)

Unsupported snippet types include:

- Picture (Simple)
- Smart Image (Map)
- Statblock
- Hero Lab Portfolio
- Foreign Object
- Any snippet types listed under "Documents and Media"

Version 0.01a

> I am not well-versed in translating XML via XLST or CSS, so I'm working with what I know, and right now, that's PowerShell. I know there are probably 50 different ways that this could have been done better, and I know that there are a few people here who know all those ways, but alas, I'm stuck with what I know.

- In regards to the above, I'm not sharing this to impress anyone. I just want a simple, easy way to print, and I thought I would share this with anyone else who might want the same and who also isn't skilled with XLST and CSS.
- I'm mainly working on this for myself, because I want a way to print. If you like what you see, and you wish to request a change or a feature, I will do what I can, but understand that this is not my full-time job. In fact, I do not currently have a full-time job, nor any job, so my full-time job right now is to find a full-time job.
- This is a work in progress.
- I have tested multiple conditions, but only with small, simple exports.
- This will only process a compact export, not a full export. I'm not even going to mess with a full export.
- Right now, this is intended to process plain-text and tag-based snippets. It will not do anything with images, simple or smart. I have not tested how it handles tables or anything other than plain-text.
- The output, right now, is plain text. I would like to add CSS and HTML formatting, but right now, I wanted to get some basic functionality done first. And I still have to study up on CSS and HTML.
- The advantage with doing this through a procedural or scripting language is that I have finer control in what information to extract and how. The disadvantage is that it could take a while to run through a 5 GB realm. I'm not there yet, so I can't say for sure, though. But I'm guessing it will.
- I'm tired and I'm going to sleep. I hope this helps somebody. :-)
- Feel free to comment here or send me a PM, but just to preempt any questions of "Why did you do it this way when you could have done it this other and much better way?", the answer is, "Because this is the way I know how to do it."

---

08. License

Overview

---

09. Support

Overview

The best way to report issues is to send a private message to EightBitz on http://forums.wolflair.com

If that fails for any reason, you can send an email to kingmorganii@gmail.com, though I may not respond to that as quickly.

---

10. To Do

Overview

- Logging in the main script
- Error Trapping in the main script
- Input validation in the GUI script
- "Load Script" button in the GUI script
