madcap software

WHITE PAPER

Project Essentials: 7 Best Practices for Starting Your First MadCap Flare Project

Table of Contents

Introduction

MadCap Flare is a very powerful tool. It has so many features and options it can be a challenge determining the best place to start, particularly on a project level. New Flare users often jump right in and begin importing or authoring content. But without proper planning, a small issue in your project now could turn into a much bigger problem later on that ripples down to future projects.

This guide presents seven best practices for starting your first project on a sound footing and staying on that footing for later projects. However, always consider your specific situation before following any suggestions.

The suggestions are:

- Use version control.
- Use a parent/child project structure.
- Create topic type templates
- Set up control file folders.
- Create a CSS.
- Create one or more table CSSs.
- Set standards.

Let's look at each of these in more detail.

Use Version Control

New Flare users often store projects directly on their local PC. This works but has risks and limitations.

- The major risk is that the project is on your local PC. If the hard disk fails or you spill coffee into the PC, you may lose the project and have to recreate it from scratch. You might get around this risk by backing up the project to a shared drive or a USB stick, if you can do this consistently.
- The limitation is that it's hard for multiple authors to work on a project. You can split the project into sub-projects, passing the sub-projects to the authors, and combining the sub-projects at the end in order to generate the output. This works but can get cumbersome.

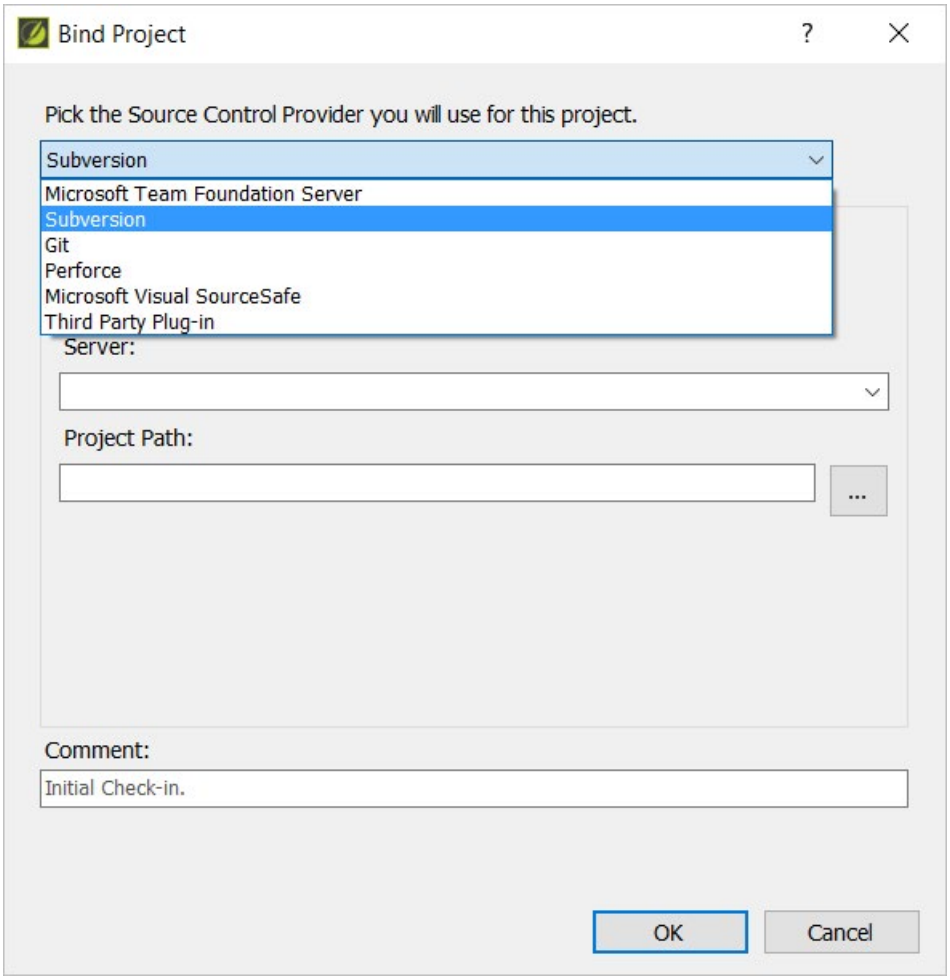
Version control eliminates the risk and fixes the problem.

Basically, a version control system stores the main copy of the project on a server. Individual authors can then download a copy of the project to their local PCs. The authors can then make changes on their local copies of the project and upload those changes back to the main, or master, copy on the server.

Individual authors still have to coordinate in order to avoid changing the same material, but the version control system will track the changes and warn of any conflicts. Version control systems offer additional benefits, such as the ability to "roll back" to an earlier version of the project and the ability to highlight differences ("diffing") between two versions of the same file.

This eliminates the risk of storing the project on a local PC. If your hard disk crashes or you spill coffee in the PC, just get a new PC, download a copy of the project to the PC, and go back to work. This also lets you easily add an author to a project. It also eliminates difficulties of getting multiple authors working on the project.

Flare supports most major version control systems natively, including Subversion (SVN), Microsoft Team Foundation Server (TFS), Git, and others. MadCap Software also offers its own cloud-based system called MadCap Central, which combines version control features with project management features. You can see a list in the Bind Project dialog box shown below.



Do you need version control? No. You could simply back up the project yourself to a shared drive or USB drive, if you remember to do so regularly. But version control systems take a lot of that work off your hands, which is why version control is the absolute top priority recommendation for any new Flare site even if you are the lone author.

Use a Parent-Child Project Structure

Consistency is often desirable. To get the same look across multiple projects, you want to use the same CSS or table CSS. To get the same listing of copyrights, you want to use the same description. And so on. The question is how to get and enforce that consistency.

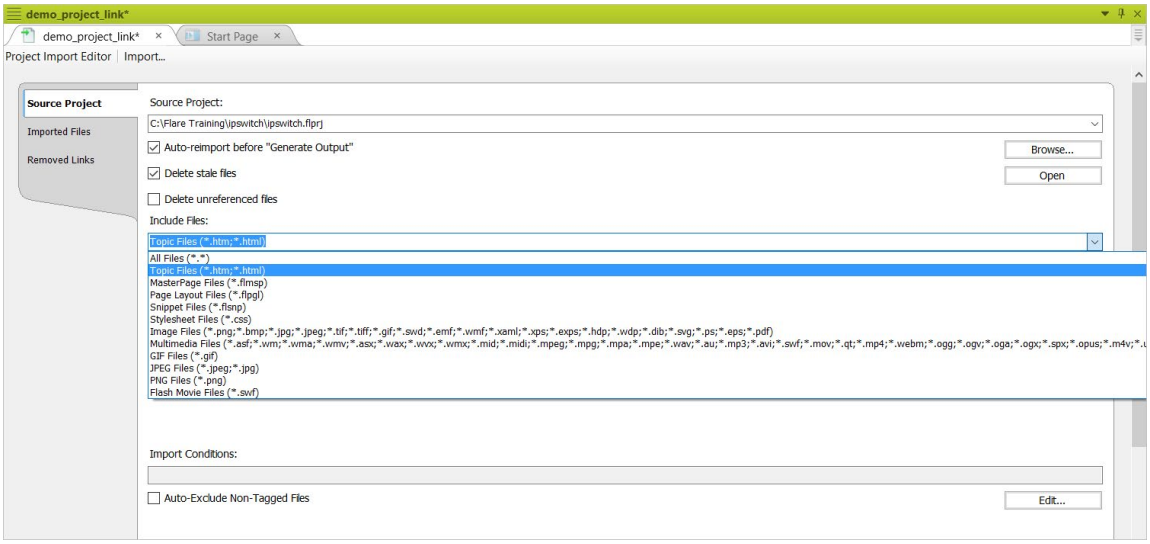
The typical approach is to create the files to be shared, such as the CSS or a copyrights topic, copy those files to each project, and tell each "child" project's author not to change the common files. This works but has two problems.

- If one of the common files changes, such as adding a style to the CSS or adding a copyright to a copyrights topic, someone has to copy the new versions of those files to the child projects.
- The owner of the common files may tell the child project authors not to modify the common files but has no way to prevent them from doing so.

The parent-child project structure, created using the Flare Project Import feature, solves both problems.

You create one project that you designate as the "parent" project and that contains the files like the CSS and common topics that you want to share across other projects.

You then create the "child" projects, link each child to the parent, and copy down the common files from the parent to the child. Now, each child will have different content but the same common files. You can be very specific as to which file types to copy down and even use conditions to control the process, as shown below.



The best part of this feature is that Flare maintains a link between a child project and the master. If one of the common files changes, because the child project author or the master project author changed them, the changed files are copied down from the master to the child projects, overwriting the previous versions in the child projects. In effect, you get invisible consistency – one of the most useful features in Flare.

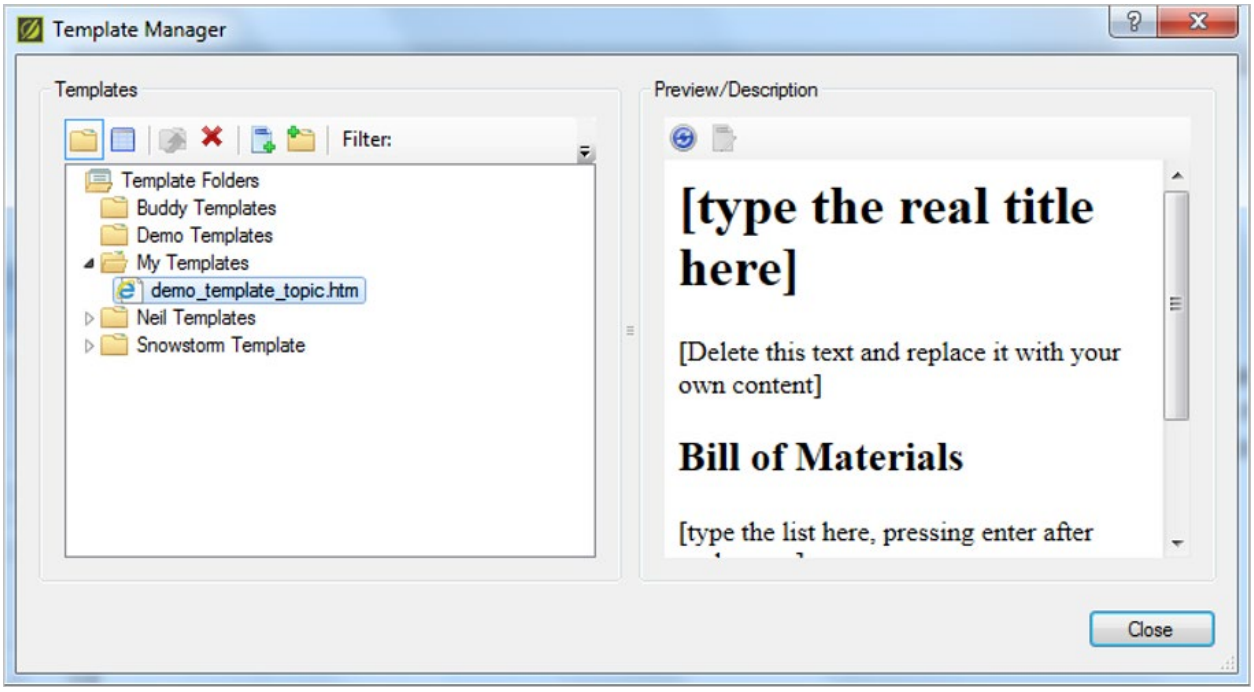
For more information, see “Import Flare Project Wizard” in Flare’s help at http://help.madcapsoftware.com/flare2017r2/Content/CSH-Only-Topics/Flare/Import-Flare-Project-Wizard.htm

Create Topic Type Templates

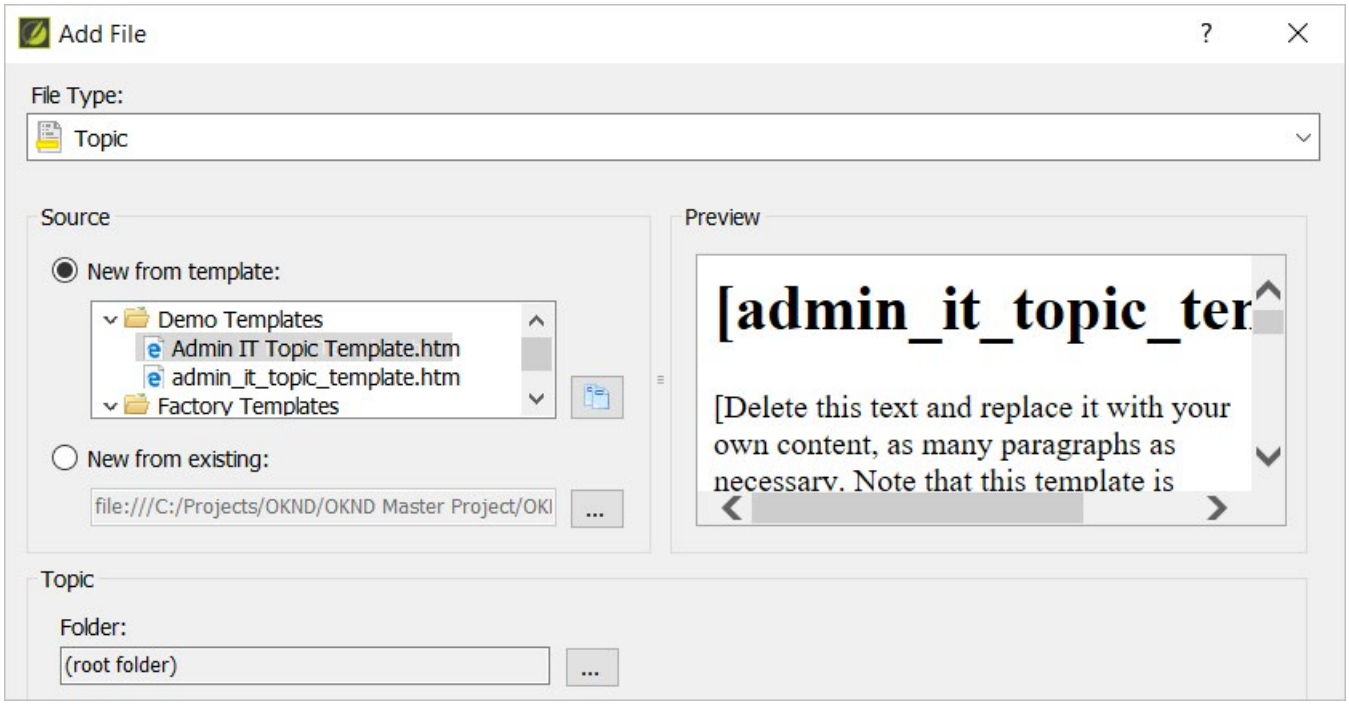
Consistent structures for similar topic types make it easy for authors to write and readers to understand the material. It's easy to create topics to use as templates but also easy to accidentally overwrite them. What's needed is topic templates that can't be overwritten. Flare's Template Manager feature lets you create them.

Creating a topic type template, for a concept, task description, or any topic type (and this is not DITA), is a pretty simple process.

1. Analyze your content to find standard types, such as concept, procedure, reference, or others.
2. Create a Flare topic to use as the template for each type. This topic might have prompts such as "[type the list of materials here]".
3. Move the template-to-be topic out of the project and into a temporary storage folder.
4. Add the templates to Flare's interface using the Template Manager feature, shown below.



5. When you create a new topic, you can now select the appropriate template from the list in the New From Template field. The result is a new topic but with a pre-defined structure, as shown below.

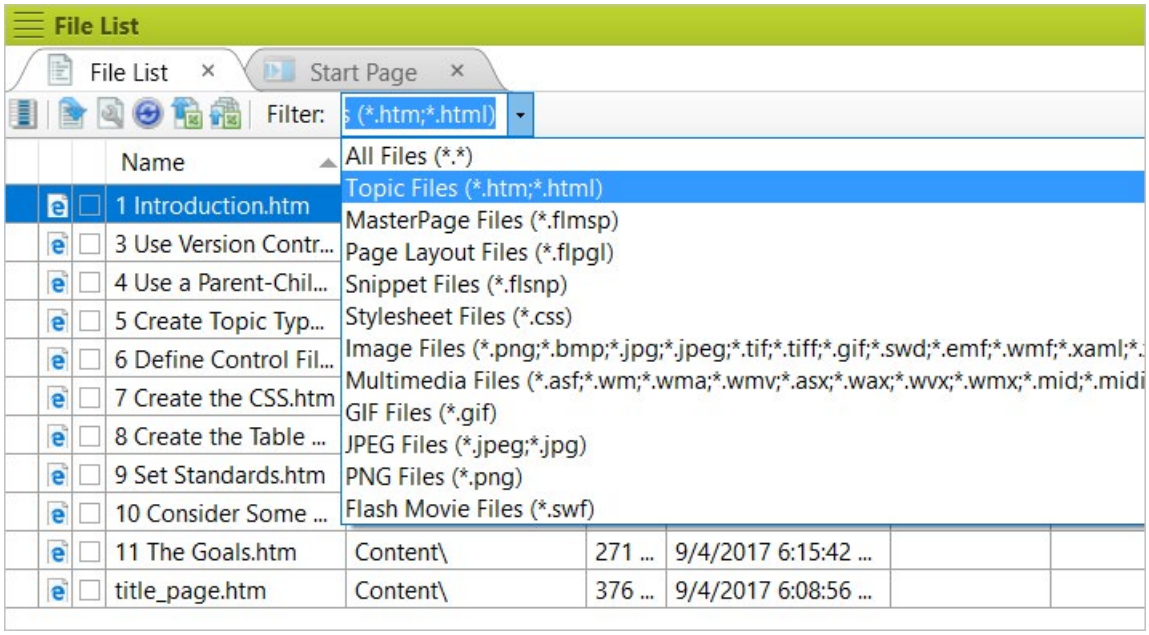


It's a surprisingly simple, quick task. You'll spend most of your time creating the template topics. Adding them to Flare's interface takes just a few seconds, and you can make them available over the network to all your Flare authors. For more information about this, see “Managing Templates” in Flare’s help at http://help.madcapsoftware.com/flare2017r2/Content/Templates/Flare/Managing-Templates.htm.

Set Up Control File Folders

When you create control files like CSSs and master pages, Flare automatically puts some in sub-folders under the Resources folder on the Content Explorer and others, like table CSSs, under the Content folder on the Content Explorer.

Flare doesn't care where these files are; it will find them wherever they're located. You can also look for them in the list of files on the Content Explorer or by using the filters to list only the type of file you're looking for in the File List, shown below.



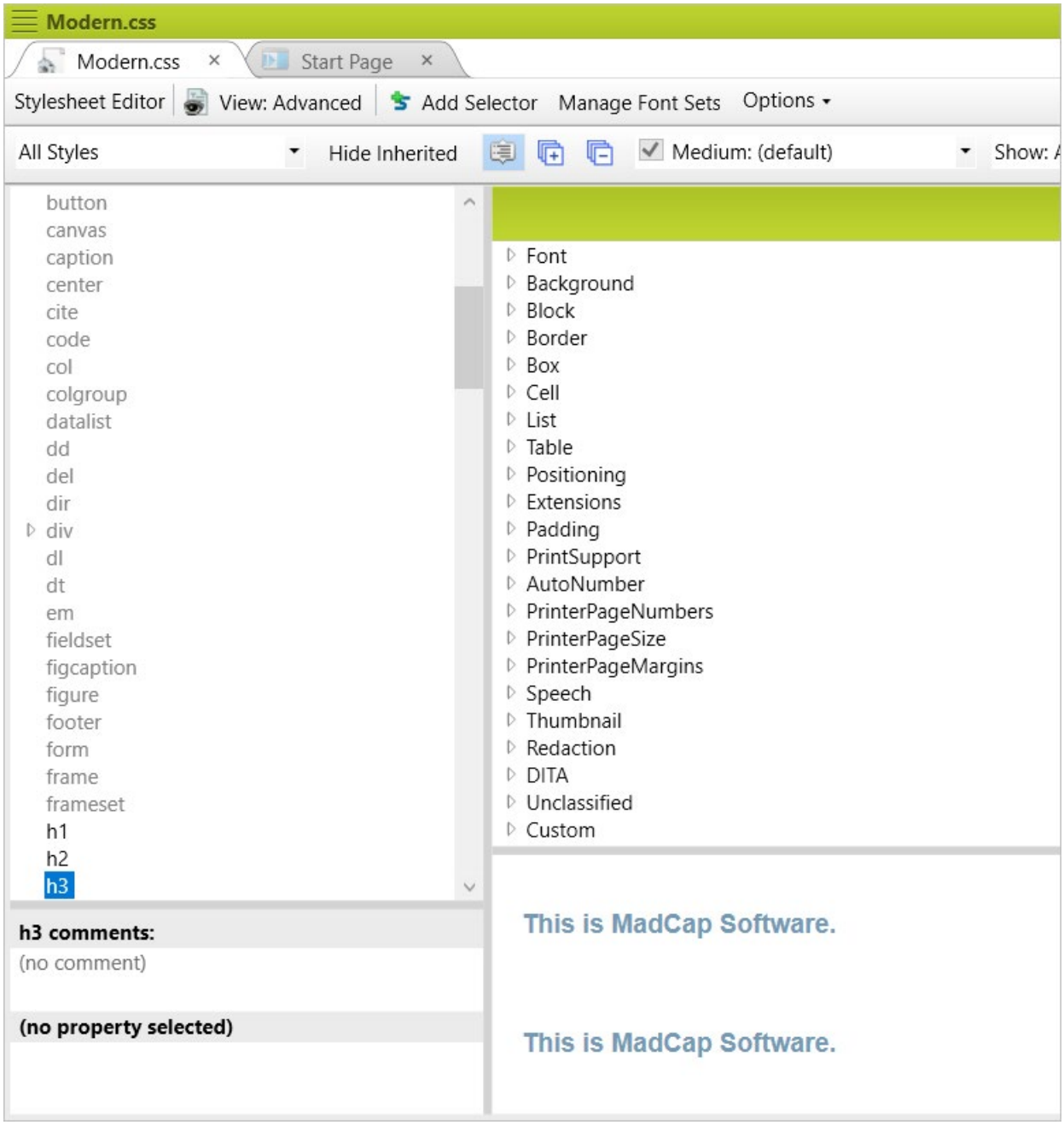
However, anything you do to simplify your projects can only help. One surprisingly simple thing is to create sub-folders to contain the specific types of control files.

For example, if you create a master page, Flare will list it as master_page_name.flmsp in the Content folder. To simplify the projects, add an additional folder for master pages, or any other type of control file, under the Resources folder. Simply right-click on the Resources folder, select New > Folder, and type the folder name, such as MasterPages. You'll see that folder name in the Folder field on the Add File dialog box and can select it with one click to specify where to store the new file.

This is not a major workflow task. It's a small one but, together with other small ones, can make Flare and your projects easier to navigate.

Create a CSS

If you're new to Flare, a CSS may seem like one more control file; it’s actually one of the most important. It lets you specify the formatting for a project and add features that support single sourcing and various types of file processing. As shown in the image below, Flare offers a very powerful CSS (stylesheet) feature.



Here are some overall suggestions for your CSS work:

- After defining your CSS, resist the urge to tinker with it.
- Try to put all style code in the CSS, instead of locally in the topics.
- Don't change de facto standards unless absolutely necessary. For example, you can format links as pink text on a purple background rather than the usual blue and underlined, but you'll have to explain what that format means since it won't be what users are accustomed to.
- Keep it simple and consider maintainability.
- Stay out of the code unless there's a good reason otherwise and you know what you're doing. Working directly in the code runs the risk of making typos or violating good coding practice.
- Document it.

The purpose of this paper is not to go into detail about the CSS, other than to recommend analyzing your content and direction and creating the CSS early in your authoring. You can create the CSS at any time, including during a project. However, CSS work can get detailed enough that it can be hard to focus on both the project and the CSS at the same time. Instead, do it as a separate task.

Create One or More Table CSSs

A table CSS is conceptually the same as a regular CSS but for tables. The same points apply. Creating one or more table CSSs, one for each style of table you use, can easily add a lot of consistency to your tables.

Set Standards

The more you standardize, the more consistent your projects will be and the less authors will have to guess about what setting to use for a given task or feature. You might standardize:

- Graphic file formats.
- Conditional build tag usage.
- Variable and snippet usage.
- Index entries.
- Wording (which you can control to a degree by using variables).
- Link types.
- hyperlinks vs. cross-references.
- File naming conventions.
- Anything else you can think of...

Additional Tips

Finally:

- Document your project settings. It's all well and good to set standards for your work but not very helpful if you forget what they were or can't transmit them to your successor.
- For additional learning, the basic/intermediate training courses are a great resource. Many people learn Flare on their own, but the training course can provide structure that can guide you through many of the best practices and advanced techniques.
- Join a user group if there's one near you. Someone in your local user group has probably already made that mistake that you're trying to fix and can tell you the answer.
- Don't be afraid to contact MadCap support, even if you think your question is silly. The support agents are there to help you make good use of Flare.

The Goals

Flare is a large, deep, and powerful tool and if you're new to it, it may not be obvious where to start or what decisions to make to start your projects. By following the suggestions in this post, you'll:

- Create controls and standards that are common to multiple projects and multiple authors. You can then focus on creating project-specific control files and content.
- Start your Flare work under control and keep it under control.

About the Author

Neil Perlin is an internationally known consultant, strategist, trainer, and developer for online content in all forms from online help to apps. He is MadCap Certified for Flare and Mimic and a long-time Flare and Mimic trainer and consultant. He is also ViziApps Certified for the ViziApps Rapid Mobile App Development (RMAD) platform. He is the author of books on Flare, Mimic, and mobile apps. Neil writes columns and articles for various industry publications and is a popular conference speaker, most recently at TCUK 2017 in Nottingham, England. You can reach him at nperlin@nperlin.cnc.net.

Looking for More Resources?

Browse our library of webinars, videos, white papers and more.

Learn more at **madcapsoftware.com/resources**