MadCap Flare Refresh: Reviewing the Basics to Work Smarter

BY KATE SCHNEIDER

WHEN YOU'RE GETTING ready to start a new job, you prepare yourself in many ways. You clean up your résumé, you study companies, you might even buy a new suit. But if you've held one technical writing position for a few years, and especially if you've been using a single authoring tool for a long time, it's easy to forget one thing that might be beneficial when you start your new job: brushing up on your authoring tool knowledge.

Consider these questions:

- Will you be responsible for different tasks in your new job?
- Will you be building a Help system from the ground up, or will you be managing a complex project?
- Will you be single-sourcing content in your new job?
- Has your authoring tool released new features recently? If so, have you used all of them?
- Do you use a set process at your current job? Have you experimented to see if there are more efficient ways to use your authoring tool (e.g., new features, different settings, single-sourcing options)?
- Were you responsible for setting up your current project? If so, when was the last time you looked at those setup files (e.g., page layouts, stylesheets, master pages, targets)?

If you said yes to any of these, you're not alone. However, you don't know what you'll be getting into with your new job—that's one of the things that makes new jobs exciting. This also means that you might need an entirely different authoring tool approach than you're using now.

When I decided to leave my technical writing job at MadCap Software—which makes MadCap Flare, an authoring tool, and other software for technical writers—I knew I had no small task ahead. I was used to working with fantastic documentation, and lots of it. I was a tech writer for tech writing software! And now I was tasked with starting from scratch: building a new Help system from nothing. So before I got started, I brushed up on my Flare basics. I'm glad I did, because it helped me get my project up and running efficiently, with long-term goals in mind. If you're starting a new job—or building out a new project at your existing job—I recommend reviewing project setup, project structure, and single-sourcing to help you get started.

Project Setup

Before I could write a word, I needed to start a new project. But first, I had to consider if I wanted to use a single project for all of my deliverables (or use a global project with linked child projects), and if I wanted to use a template or start from scratch.

Single Project vs. Linked Projects

Using a single project and using linked (master and child) projects both have their benefits. A single project is easy to manage because you don't have to worry about knowing where files are stored or accidentally editing linked files in the wrong project. Linked projects (known in Flare as global project linking) are easy to manage because all of your shared files (e.g., page layouts, stylesheets) are stored in a master project, and each deliverable is linked to that project in a child project; this keeps all of the projects smaller and cleaner.

For my new project, I will be working with lots of shared files, but I will also be working with multiple audiences and repurposing many of the same topics across deliverables. I chose to use a single project and then apply conditional tags for each of my deliverables (discussed later), because I had too many files that would eventually be shared.

If you have been using one system or the other, you should take some time to get familiar with how the other system works and what might work best for your new project. Additionally, global project linking is a feature specific to Flare, so if you are using a different authoring tool and coming over to Flare for a new position, I recommend familiarizing yourself with this on a small test project—in other words, before you have dozens child projects and thousands of linked files to worry about.

Scratch Project vs. Flare Templates

Once I decided to create a single project, I had to decide how to create it. I could create it from scratch, which would give me the most flexibility for pretty much everything: a blank Flare project comes with one starter topic, and that's pretty much it. Of course, this would mean I'd either be working with generic-looking topics for a while, or I'd have

to spend some time putting together company-tailored stylesheets and page layouts (or getting them from my company's design team). I planned to do this eventually, but I wasn't sure I wanted to do it on my first day.

On the other hand, I had several out-of-the-box Flare templates at my disposal for both printed and online outputs. These templates are great for new Flare users who want to have some basic settings in place (e.g., page layouts, stylesheets) or who want to get started quickly. They are also useful for experienced users who are less concerned with creating a custom project right away and just want to get words on paper.

I decided to cobble together my own project. I started with Flare's Online & Print Top Navigation and PDF Advanced template, because I wanted a TopNav home page set up in case I needed to show anyone sample HTML5 output. However, I knew that initially I'd be working with PDF outputs, and this template also gave me a full complement of page layouts to use. Next, I deleted everything I didn't need to declutter the template—including all of the starter topics—so I could structure my topics exactly how I liked. Finally, I opened the stylesheet and added several custom styles that I wanted in my project to format notes and examples. This all gave me enough to work with as well as make my project look nice (while allowing me to build outputs), while not being too company-specific. Later, when I am able to grab a few minutes of the graphic designer's time, I'll give him what I have so far and he can help me add company branding to my stylesheet and layouts.

Figure 1. Use standard templates to get basic files for project setup, then edit them to reflect your immediate needs.

Project Structure

When I started my new position, I knew that I would be creating multiple deliverables and writing for several audiences, but also that I would be creating PDF outputs now and online outputs later. I want to be able to put my project

together without having to restructure it in a few months, so I carefully considered my project targets and tables of contents.

Project Targets

I came into this position knowing that I'd be working mostly with PDF outputs to start, but I would also need a basic HTML5 output to use as a sample, since I'd be building this in the future. I had a short list of project deliverables: all PDF, with one end user guide and the rest designed for administrators.

I started by creating multiple PDF targets in Flare's Target Editor. What makes each target unique is its set of conditional tags based on audience. I'll discuss this in more depth later, but what you need to know for now is that each target uses tagging to define the content that appears (or does not appear) in the final output. I set these rules in the target before I even got started writing. This way I could get started with single-sourcing right off the bat and know that my final outputs would show the correct content.

Additionally, I enabled the "Use TOC Depth for Heading Levels" option on the Advanced tab of the Target Editor; this allowed me to use Heading 1s on all of my topics and have Flare automatically update the heading level based on the topic's location in the TOC (which is useful if I repurpose a topic in a different table of contents at a different level).
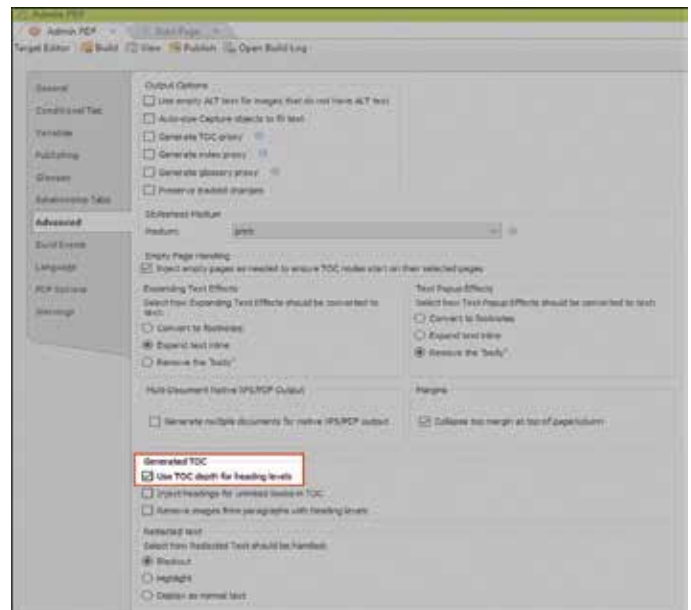

Figure 2. Enable "Use TOC Depth for Heading Levels" to make your heading levels reflect their location in the table of contents.

Tables of Contents

Because I was going to be starting with printed output, my initial table of contents was for one of my administrator PDF guides. I set up my TOC so each functional area of the system was a chapter, with related concept and task topics making up the bulk of the chapter's content.

I also set up a skeleton TOC for my future online output. This TOC was set up differently, with only a few first-level

topics ("Features," "Administration," and "Interface" for now; I'll add no more than two or three more as my project grows). All of my other topics fall as second- or third-level topics in one of these categories. Because I will be using a Top Navigation HTML5 structure for my online output, limiting to five or six top-level topics will keep my navigation uncluttered.

Single-Sourcing

As any writer who has used it knows, single-sourcing is a massive topic. I've been using single-sourcing for many years, but I always feel like I'm learning new tricks that help me find ways to repurpose my content just a little bit more effectively. I was excited to start a new project from nothing because it meant that I could really maximize my content reuse. With knowledge of my deliverables and audiences in hand, I could get a good grasp on single-sourcing right away. I focused on conditional tagging and reusable content.

Conditional Tagging

Conditional tagging allows you to exclude (or include) content from your project at the target, topic, or snippet (a type of reusable content) level. This is invaluable because if you are writing for multiple audiences or creating multiple deliverables, you can effectively filter a single document to show only the content that is related to the relevant audience or deliverable.

I knew that I needed to write for administrator and end user audiences, so I focused my main conditional tag set on these audiences. As mentioned previously, I toggled the conditional tags in each target to reflect which audience should see what content. For example, if I tagged a topic for "Admin," that was included in my Administrator Guide, but excluded from my End User Guide.

I also created a number of snippet-level conditions to include or exclude content in my reusable snippets (in Flare, these conditional tags can be treated separately from your other conditions, and you can toggle them on and off for individual snippets). In previous jobs, I have based these on audience or target, but this got unwieldy, fast. But I learned a trick at MadCap that I implemented at my new job: I simply created a handful of numbered snippet conditions.

Then, I can turn them on or off as needed. Because they are not specifically tied to a target, they are easy to maintain and manage, and I can use them in any topic.
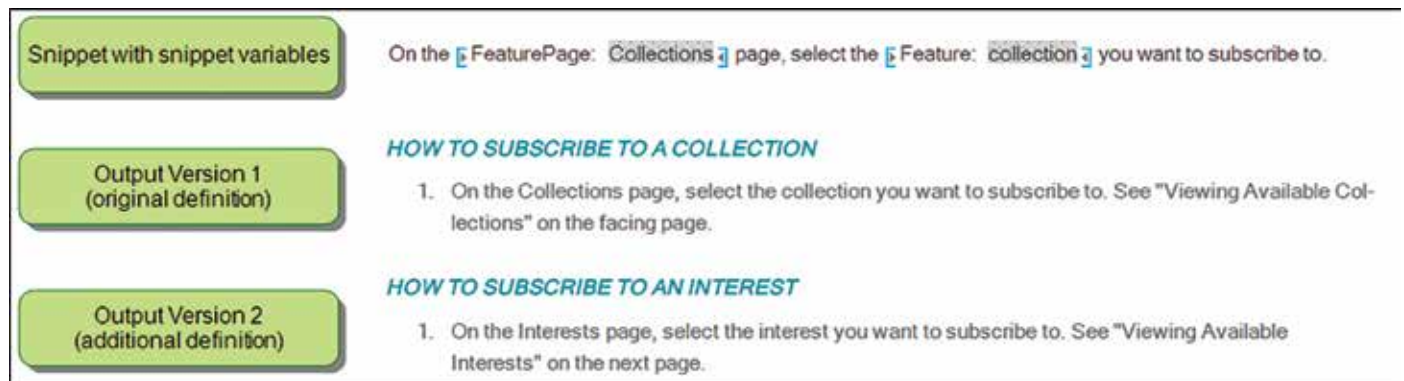
Reusable Content

My biggest focus was on reusable content: snippets and variables. I have rarely had a chance to start a project from scratch, so this was a thrill. From the first topic I wrote, I was on the lookout for content that I was using in more than one place. If it was used often enough or was complex, I immediately turned it into a reusable snippet to use again later. By creating reusable snippets (or variables, in the case of page names), I am able to make changes in one place that are reflected throughout my project. Additionally, I can use the snippet conditions that I created to enable or disable content within a snippet so it displays only where I want it.

Finally, there is one other feature that I was particularly excited to add to my project: snippet variables. But I thought they'd be a great addition to my project. I have several topics that have similar content, except for the page name or the name of a feature. I created multi-definition variables for this information (each definition is the different page name or feature), and added them into snippets for each step in my topic. Then, using Flare's snippet variable settings, I can change the variable definition that appears in the snippet. Same content, two— or three or six—results!

Getting started on any new project or with any new job is always difficult, but reviewing the basics and taking the time to think out your long-term goals before you get started will help you avoid re-work in the end.

If you want to know more about any of the topics I've discussed here, check out the MadCap Flare online Help at *http://help.madcapsoftware.com/flare.* ℹ️

KATE SCHNEIDER *(kateschneider42@gmail.com) is a Senior Technical Writer at Webinfinity. She has ten years of experience in the software industry, where she has background building online Help systems from the ground up, single-sourcing and restructuring complex documentation suites, and proposal writing. She holds an MS in Technical Communication from Northeastern University. She lives in Tucson, Arizona.*

Figure 3. Snippet variables give you additional flexibility when working with reusable content.