



Comparison Guide: Choosing Between Help Authoring Tools and CCMSs

Introduction

Every few years, the debate flares up over whether to use a HAT (help authoring tool) or a CMS (content management system) to create online Help and documentation. The gist of the debate is that HATs, which appeared in 1991, are outdated and should be replaced by a CMS or, today, by a CCMS (component content management system). This white paper, which revisits that debate, touches upon topics that may help your company avoid repeating the mistakes of the 1990s when companies bought HATs without first distinguishing between different HATs' feature sets and workflows.

Note: for the purposes of this white paper, Flare is categorized as a HAT. However, Flare, like all modern HATs, went far beyond simple help authoring long ago to become the single-sourcing, multiple output powerhouses of today. The comments on the HAT side apply to most HATs but focus on Flare. The comments about the CCMS side are generic.

Let's start with a few definitions.

- HATs let us create online Help and documentation in multiple online and print formats. A HAT stores the files that form a project – topics, content “fragments” like variables and snippets, plus control files like skins and stylesheets – as separate chunks using the standard Windows file structure.
- CMSs (content management systems) can also create online Help and documentation in online and print formats, although they have other uses as well. Like HATs, CMSs work with chunks. The definition of “chunk” depends on the CMS, but it's often a page or a document, such as a set of Word documents stored as individual chunks in a CMS. Unlike HATs, CMSs store content as entries in a database rather than as files.
- CCMSs (component content management systems) are conceptually the same as CMSs but work with more “granular” – e.g. smaller, chunks. For example, where CMSs store documents, a CCMS can store those documents in smaller chunks that are similar to the chunks used in HATs. This provides more flexibility than a CMS. Like a CMS, a CCMS stores content as entries in a database rather than as files in the Windows file structure. Because of the conceptual similarity between HATs and CCMSs, this white paper compares those two types of tools and ignores CMSs.

Comparing HATs and CCMSs

Let's compare HATs and CCMSs from three angles – features, cost, and miscellaneous but crucial issues like project size and access.

Features

When making the decision between two different software options, comparing and contrasting the features and characteristics of each offering is vital to consider. The following comparison points are based on a featured article from February 29, 2009 by Mike Heck, from InfoWorld Test Center Guide: Content Management Systems, and adapted to include points that address CCMSs. So, what are the major features of a (C)CMS and the HAT equivalents?

For Authoring

Easy creation of authoring templates

Flare has offered this capability for years. Defining what you want in a template can be difficult, but the mechanics of creating one and integrating it into the interface are easy.

Easy creation of topic-sized content using the authoring templates, or sub-topics like snippets and variables or other fragments

This has been a core feature of every HAT since the introduction of Doc-To-Help in 1991.

Digital asset management capability for image and multimedia files

Also a core feature of every HAT since the introduction of Doc-To-Help in 1991.

The ability to import data in a variety of formats

Flare can import Word, FrameMaker, and Excel files, plus various flavors of HTML and XML, along with DITA. (The latter offers an easy and inexpensive way to back out of DITA for companies that adopted it only to find that it was a bad decision.)

Link maintenance and management

Flare keeps track of folder and file renaming, moving, or deletion and can automatically check for and

offer to correct any resulting broken links. Note that the responsibility for creating effective folder and file names still falls on the author, unlike in a CCMS where much of the naming is handled by the CCMS engine itself.

Collaboration between authors and subject matter experts for content review and new content submission

MadCap Contributor “electronifies” the submission and review process to make it easy to pull this material into a Flare project and get rid of badly formatted Word comments or unreadable hand-written comments.

Note also that MadCap’s release of the 2018 version of Flare, with an updated version of [MadCap Central](#), adds another, cloud-based option to the author-subject matter workflow.

Metadata support

Depending on how you define “metadata”, this is one area of weakness in Flare. If you consider condition tags to be metadata, which they are, then Flare strongly supports metadata. If you define metadata to mean standards like RDF (the Resource Description Framework from the [WorldWide Web Consortium](#)), then Flare does not support it.

Plug-ins for desktop applications such as Word or FrameMaker®

Flare can automatically input Word and FrameMaker® files and generate output, letting it be used as an output generator for those tools rather than an authoring platform in its own right.

Accessibility of code for power users

Flare provides easy access to the underlying code in several ways. It’s usually best to not access the code unless you’re expert because working in code can lead to quirky results and is a fertile source of errors. But working in the code does offer capabilities that the GUI may hide.

Translation management

MadCap Lingo supports translation and localization.

For Output

Support for single sourcing

Another core feature of every HAT since the introduction of Doc-To-Help in 1991.

The ability to output and multi-channel publish to a variety of formats

Flare can output to a wide variety of formats including responsive HTML5, Clean XHTML (which strips Flare-specific code out of the output in order to let you use the files outside a Flare target, such as for a wiki), plus a variety of older or niche formats like WebHelp, Eclipse Help, DotNet Help, and eBook formats. Users can also create print formats such as PDF, Word, FrameMaker®, and others.

The ability to personalize content to a specific audience

Flare has offered this capability since it was introduced through conditionality, variables and snippets, variables in snippets, style sheets, style sheet mediums, master pages, and more.

For Management

Version control capabilities such as rollback and differencing

MadCap Software offers a hosted version control system – MadCap Central – that integrates easily with Flare and offers various workflow management features. Flare also offers native support for popular third-party version control systems including Git, Subversion, Perforce, Microsoft’s Team Foundation Server and Visual SourceSafe. It’s also possible to add support for others.

Workflow management

MadCap Central doesn’t have the many governance and workflow control features of a CCMS (or CMS), but it does have a powerful initial set of features such as the ability to create and manage project related tasks, and a handy Project Checklist feature which makes it easy to create unique, customizable project milestones that can be tracked individually or as a team.

The ability to customize the interface via an open SDK

Flare lets you customize the interface and save the result as a “layout”. You might do this if you want to efficiently set up the interface for specific tasks, like indexing and linking. These interfaces can be called up as needed, without having to recreate them over again.

Based on these features, HATs, and Flare in particular, appear to hold their own well against the CCMS-based authoring tools.

Cost

To complicate the choice between a HAT and a CCMS-based authoring tool, they appear to be priced in different ways. For a HAT, you can go to the vendors' web site to find the cost, which will likely be per seat or a floating license, so you can easily figure out what you'll pay.

For a CCMS-based tool, the process is more complicated because, for a cloud-based tool, the vendor may first need information about variables like the number of authoring seats, data storage requirements, and setup requirements.

There's another aspect to tool cost - a willingness to put up with an inappropriate but expensive tool. The higher the cost (which includes the cost of training the users and converting legacy material), the more reluctant your company will be to admit a mistake and look for a more appropriate tool. In other words, it's a lot easier to be dissatisfied and switch tools if you're writing off \$1,000 rather than \$10,000 or \$100,000 or more.

Miscellaneous but Crucial Issues

In addition to the features and cost arguments, consider two other points.

Project size

One of the arguments for a CMS- or CCMS-based authoring tool has been the idea that as your projects get larger, Windows' file system won't be powerful enough to handle the number of files. This may be true, but a project has to have a huge number of files before that becomes an issue. In other words, unless you're working on some very unusual or very large projects, it's unlikely that you'll need a CCMS-based authoring tool.

Note: the author notes that his average project has 1,000 topics. The largest project he's worked on had 176,500 files. The largest project he knows of has over 800,000. But in each case, these projects ran in the Windows' file system.

Network speed and reliability

Another argument for a CMS-or CCMS-based authoring tool is the fact that a cloud-based tool is more convenient if you have multiple authors using the tool. There is an element of truth to this, but it's not that clear-cut. For example, if you're using a cloud-based, CCMS-based tool, you need fast and reliable network and internet access. If there are any speed or reliability problems, or if you're in a situation where you don't have network and internet access, you either won't be able to work or will

have to work on a local copy of the tool, effectively negating many of the benefits of working in a cloud-based environment.

Conclusion

The points above are meant for a broad audience and may not reflect your specific needs. Be sure to look at your specific requirements and goals when picking an authoring tool. However, based on the points above, modern HATs in general, and Flare in particular, are general-purposes powerhouses that should meet most online Help and documentation authoring needs for years to come.

About the Author

Neil has 39 years' experience in technical communication, with 32 in training, consulting, and development for various online formats and tools including WinHelp, HTML Help, CE Help, JavaHelp, WebHelp, Flare, and more. Neil is a frequent speaker at MadWorld and various professional groups and the author of several books about Flare and mobile app development.

Neil is MadCap certified for Flare and Mimic, ViziApps certified for the ViziApps mobile app development platform, and certified in other authoring tools. He provides training, consulting, and development for online help and documentation, Flare, Mimic, other authoring tools, mobile apps, XML, single-sourcing, topic-based and structured authoring, and content strategy. He can be reached at nperlin@concentric.net, www.hyperword.com.

Looking for More Resources?



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



Learn more at madcapsoftware.com/resources

